

F O U R T H E D I T I O N



LINUX

CORTAFUEGOS

Mejorando la seguridad con nftable y más alla

S T E V E S U E H R I N G

Cortafuegos Linux®

Cuarta edición

Esta página se dejó en blanco intencionalmente



Libro descargado por helcurt
para la comunidad de
aprecking

<https://t.me/librosdehacking>

Contenido de un vistazo

Contenido ix

Prefacio xix

Sobre el autor xxi

I: Filtrado de paquetes y medidas básicas de seguridad 1

- 1** Conceptos preliminares subyacentes al filtrado de paquetes
Cortafuegos **3**
- 2** Conceptos de filtrado de paquetes **25**
- 3** iptables: la administración del firewall heredado de Linux
Programa **51**
- 4** nftables: la administración del cortafuegos de Linux
Programa **83**
- 5** Creación e instalación de un cortafuegos independiente **95**

II: Problemas avanzados, múltiples cortafuegos y perímetro

- Redes 143**
- 6** Optimización de cortafuegos **145**
- 7** Reenvío de paquetes **179**
- 8** NAT: traducción de direcciones de red **197**
- 9** Depuración de las reglas del cortafuegos **211**
- 10** Redes Privadas Virtuales **229**

III: Más allá de iptables y nftables 235

- 11** Detección y respuesta a intrusiones **237**
- 12** Herramientas de detección de intrusos **249**
- 13** Monitoreo de red y detección de ataques **263**
- 14** Integridad del sistema de archivos **295**

IV: Anexos 311

A Recursos de seguridad 313

B Ejemplos de cortafuegos y secuencias de comandos de soporte 315

C Glosario 351

D Licencia de Documentación Libre GNU 363

Índice 371

Contenido

Prefacio xix

Sobre el autor xxi

I: Filtrado de paquetes y medidas básicas de seguridad 1

1 Conceptos preliminares subyacentes al filtrado de paquetes Cortafuegos 3

El modelo de redes OSI 5

Sin conexión frente a orientado a la conexión
Protocolos 7

Próximos pasos 7

El Protocolo de Internet 7

Direccionamiento IP y división en subredes 8

Fragmentación de IP 11

Difusión y multidifusión 11

CIMP 12

Mecanismos de Transporte 14

UDP 14

PCT 14

No olvide el protocolo de resolución de direcciones 17

Nombres de host y direcciones IP 18

Direcciones IP y direcciones Ethernet 18

Enrutamiento: Obtener un paquete de aquí a allá 19

Puertos de servicio: la puerta a los programas en su
Sistema 19

Una conexión TCP típica: visitar un control remoto
Sitio web 20

Resumen 23

2 Conceptos de filtrado de paquetes 25

Un firewall de filtrado de paquetes 26

Elección de una política de filtrado de paquetes predeterminada 29

Rechazar versus denegar un paquete 31

Filtrado de paquetes entrantes 31

Filtrado de direcciones de origen remoto 31

Filtrado de direcciones de destino local 34

Filtrado de puertos de origen remotos 35

Filtrado de puertos de destino local 35

Filtrado de estado de conexión TCP entrante 35

Sondas y exploraciones 36

Ataques de denegación de servicio 39

Paquetes enrutados de origen 46

Filtrado de paquetes salientes 46

Filtrado de direcciones de origen local 47

Filtrado de direcciones de destino remoto 47

Filtrado de puerto de origen local 48

Filtrado de puertos de destino remoto 48

Filtrado de estado de conexión TCP saliente 48

Servicios de red privada versus pública 49

Protección de servicios locales no seguros 50

Selección de servicios para ejecutar 50

Resumen 50

3 iptables: la administración del firewall heredado de Linux

Programa 51

Diferencias entre IPFW y Netfilter Firewall

Mecanismos 51

Recorrido de paquetes IPFW 52

Paquete transversal de Netfilter 54

Sintaxis básica de iptables 54

Características de iptables 55

Características de la tabla NAT 58

Características de la tabla mangle 60

Sintaxis de iptables 61

Comandos de tabla filter 62

filtrar extensiones de destino de tabla 67

filtrar extensiones de coincidencia de tabla 68

Extensiones de destino de tabla nat 79

Comandos de tabla mangle 81

Resumen 82

4 nftables: la administración del cortafuegos de Linux

Programa 83

Diferencias entre iptables y nftables 83

Sintaxis básica de nftables 83

Funciones nftables 84

nftables Sintaxis 85

Sintaxis de tablas 85

Sintaxis de cadena 86

Sintaxis de la regla 87

Operaciones básicas de nftables 91 Sintaxis
del archivo nftables 92

Resumen 93

5 Creación e instalación de un cortafuegos independiente 95

Los programas de administración de firewall de Linux 96

Construir versus comprar: el kernel de Linux 97

Opciones de direccionamiento de origen y destino 98

Inicializar el cortafuegos 99

Constantes simbólicas utilizadas en el cortafuegos
Ejemplos 100

Habilitación del soporte de monitoreo de kernel 101

Eliminación de reglas preexistentes 103

Restablecimiento de políticas predeterminadas y
detención del firewall 104

Habilitación de la interfaz de bucle invertido 105

Definición de la política predeterminada 106

Uso del estado de conexión para omitir la regla

Comprobación 107

Suplantación de identidad de la dirección de origen y otras malas
Direcciones 108

Protección de servicios en puertos sin privilegios asignados 112

Servicios TCP locales comunes asignados a
puertos sin privilegios 113

Servicios UDP locales comunes asignados a puertos
sin privilegios 116

Habilitación de los servicios de Internet básicos requeridos 117

Permitir DNS (Puerto UDP/TCP 53) 118

Habilitación de servicios TCP comunes 122

Correo electrónico (TCP SMTP Puerto 25, POP Puerto 110,
Puerto IMAP 143) 123

SSH (puerto TCP 22) 128

FTP (Puertos TCP 21, 20) 130

Servicio TCP genérico 133

Habilitación de servicios UDP comunes	134
Acceso al servidor DHCP de su ISP (Puertos UDP 67, 68)	134
Acceso a servidores de hora de red remota (Puerto UDP 123)	136
Registro de paquetes entrantes descartados	138
Registro de paquetes salientes descartados	138
Instalación del cortafuegos	139
Sugerencias para depurar el script de firewall	139
Iniciar el cortafuegos en el arranque con Red Hat y SUSE	140
Iniciar el cortafuegos en el arranque con Debian	141
Instalación de un cortafuegos con una dirección IP dinámica	141
Resumen	141

II: Problemas avanzados, múltiples firewalls y redes perimetrales 143

6 Optimización de cortafuegos 145

Regla Organización	145
Comience con reglas que bloqueen el tráfico en las alturas Puertos	145
Use el módulo de estado para ESTABLECIDO y Partidos RELACIONADOS	146
Considere el Protocolo de Transporte	146
Coloque reglas de firewall para servicios muy utilizados lo antes posible	147
Utilice el flujo de tráfico para determinar dónde colocar las reglas para múltiples interfaces de red	147
Cadenas definidas por el usuario	148
Ejemplos optimizados	151
La secuencia de comandos de iptables optimizada	151
Inicialización del cortafuegos	153
Instalación de las cadenas	155
Construyendo la entrada EXT definida por el usuario y cadenas de salida EXT	157
tcp-estado-banderas	165
seguimiento de conexión	166
consulta-cliente-dhcp-local y respuesta -servidor-dhcp-remoto	166

- verificación de dirección de origen 167
- verificación de dirección de destino 168
- Registro de paquetes descartados con iptables 168
- La secuencia de comandos nftables optimizada 170
- Inicialización del cortafuegos 170
- Creación de los archivos de reglas 172
- Registro de paquetes descartados con nftables 175
- ¿Qué compró la optimización? 176
 - Optimización de iptables 176
 - Optimización de nftables 177
- Resumen 177

7 Reenvío de paquetes 179

- Las limitaciones de un firewall independiente 179
- Configuraciones básicas de firewall de puerta de enlace 181
- Problemas de seguridad de LAN 182
- Opciones de configuración para una LAN doméstica de confianza 183
 - Acceso LAN al cortafuegos Gateway 184
 - Acceso de LAN a otras LAN: reenvío de tráfico local entre varias LAN 186
- Opciones de configuración para una más grande o menos confiable LAN 188
 - Dividir el espacio de direcciones para crear múltiples Redes 188
 - Acceso interno selectivo por host, rango de direcciones o puerto 190
- Resumen 195

8 NAT: traducción de direcciones de red 197

- El trasfondo conceptual de NAT 197
- Semántica NAT con iptables y nftables 201
 - Fuente NAT 203
 - Destino NAT 205
- Ejemplos de SNAT y LAN privadas 206
 - Enmascaramiento del tráfico LAN a Internet 206
 - Aplicación de NAT estándar al tráfico LAN a Internet 208
- Ejemplos de DNAT, LAN y Proxies 209
 - Reenvío de host 209
- Resumen 210

9 Depuración de las reglas del cortafuegos 211

Sugerencias generales para el desarrollo de firewalls 211

Listado de las reglas del cortafuegos

213 Ejemplo de listado de tablas de iptables 213

Ejemplo de listado de tablas de nftables 216

Interpretación de los registros del sistema

217 Configuración de syslog 217

Mensajes de registro del cortafuegos: ¿qué significan? 220

Comprobación de puertos abiertos

223 netstat -a [-n -p -A inet] 224

Comprobación de un proceso vinculado a un puerto particular con fuser 226

Nmapa 227

Resumen 227

10 Redes Privadas Virtuales 229

Descripción general de las redes privadas virtuales 229

Protocolos VPN 229

PPTP y L2TP 229

IPsec 230

Productos Linux y VPN 232

Openswan/Libreswan 233

Open VPN 233

PPTP 233

VPN y cortafuegos 233

Resumen 234

III: Más allá de iptables y nftables 235

11 Detección y respuesta a intrusiones 237

Detección de intrusiones 237

Síntomas que sugieren que el sistema podría

Estar comprometido 238

Indicaciones de registro del sistema 239

Indicaciones de configuración del sistema 239

Indicaciones del sistema de archivos 240

Indicaciones de cuenta de usuario 240

Indicaciones de la herramienta de auditoría de seguridad 241

Indicaciones de rendimiento del sistema 241

Qué hacer si su sistema se ve comprometido 241 Informe de incidentes 243 ¿Por qué informar un incidente? 243 ¿Qué tipo de incidentes podría informar? 244

¿A quién informa un incidente? 246

¿Qué información proporciona? 246

Resumen 247

12 Herramientas de detección de intrusos 249

Kit de herramientas de detección de intrusos: herramientas de red 249

Conmutadores y concentradores y por qué le importa 250

Reloj ARP 251

Inspectores de rootkits 251

Ejecución de Chkrootkit 251

¿Qué pasa si Chkrootkit dice que la computadora está infectada? 253

Limitaciones de Chkrootkit y herramientas similares 253

Usar Chkrootkit de forma segura 254

¿Cuándo se debe ejecutar Chkrootkit? 255

Integridad del sistema de archivos 255

Supervisión de registros 256

muestra 256

Cómo no comprometerse 257

Seguro a menudo 257

Actualizar a menudo 258

Prueba a menudo 259

Resumen 261

13 Monitoreo de red y detección de ataques 263

Escuchando el Éter 263

Tres herramientas valiosas 264

TCPDump: una descripción general simple 265

Obtención e instalación de TCPDump 266

Opciones de volcado TCP 267

Expresiones de volcado TCP 269

Más allá de lo básico con TCPDump 272

Uso de TCPDump para capturar protocolos específicos 272

Uso de TCPDump en el mundo real 272

Ataques a través de los ojos de TCPDump 280

Grabación de tráfico con TCPDump 284

Monitoreo de intrusión automatizado con Snort 286

Obtención e instalación de Snort 287

Configuración de Snort 288

prueba de resoplido 289

Recibir alertas 290

Reflexiones finales sobre Snort 291

Monitoreo con ARPWatch 291

Resumen 293

14 Integridad del sistema de archivos 295

Definición de integridad del sistema de archivos 295

Integridad práctica del sistema de archivos 295

Instalación de AIDE 296

Configuración de AIDE 297

Creación de un archivo de configuración AIDE 297

Un ejemplo de archivo de configuración de AIDE 299

Inicializando la Base de Datos AIDE 300

Programación de AIDE para que se ejecute automáticamente 301

Monitoreo AIDE para cosas malas 301

Limpieza de la base de datos AIDE 302

Cambio de la salida del informe AIDE 303

Obtención de resultados más detallados 305

Definición de Macros en AIDE 306

Los tipos de cheques AIDE 307

Resumen 310

IV: Anexos 311

A Recursos de seguridad 313

Fuentes de información de seguridad 313

Documentos de referencia y preguntas frecuentes 314

B Ejemplos de cortafuegos y secuencias de comandos de soporte 315

Cortafuegos de iptables para un sistema independiente del

Capítulo 5 315

nftables Firewall para un sistema independiente del
Capítulo 5 328

Firewall de iptables optimizado del Capítulo 6 332 Firewall
de nftables del Capítulo 6 345

C Glosario 351

D Licencia de Documentación Libre GNU 363

- 0. Preámbulo 363
- 1. Aplicabilidad y definiciones 363 2.
Copia textual 365 3. Copia en cantidad
365
- 4. Modificaciones 366
- 5. Combinación de documentos 367
- 6. Colecciones de documentos 368
- 7. Agregación con obras independientes 368
- 8. Traducción 368
- 9. Terminación 369
- 10. Futuras revisiones de esta licencia 369
- 11. Renovación de licencias 370

Índice 371

Esta página se dejó en blanco intencionalmente

Prefacio

Bienvenido a la cuarta edición de *Linux® Firewalls*. El libro analiza lo que se necesita para construir un firewall usando una computadora con Linux. El material cubierto incluye algunos conceptos básicos de redes, IP y seguridad antes de saltar a iptables y nftables, el último software de firewall en Linux.

El lector de este libro debe tener una computadora con Linux, ya sea independiente o como firewall o puerta de enlace a Internet. El libro muestra cómo construir un firewall para una sola computadora cliente, como una computadora de escritorio, y también muestra cómo construir un firewall detrás del cual se pueden alojar varias computadoras en una red local.

La parte final del libro muestra aspectos de la seguridad informática y de redes más allá de iptables y nftables. Esto incluye la detección de intrusos, la supervisión del sistema de archivos y la escucha del tráfico de la red. El libro es en gran parte agnóstico de Linux, lo que significa que casi cualquier sabor popular de Linux funcionará con el material con poca o ninguna adaptación.

Expresiones de gratitud

Me gustaría agradecer a mi esposa, familia y amigos por su apoyo incondicional. Gracias también a Robert PJ Day y Andrew Prowant por revisar el manuscrito.

Esta página se dejó en blanco intencionalmente

Sobre el Autor

Steve Suehring es un arquitecto tecnológico especializado en sistemas y desarrollo Linux y Windows. Steve ha escrito varios libros y artículos de revistas sobre una amplia gama de tecnologías. Durante su mandato como editor en la revista *LinuxWorld*, Steve escribió y editó artículos y reseñas sobre seguridad y promoción de Linux, incluido un artículo destacado sobre el uso de Linux en las carreras de autos de Fórmula Uno.

Esta página se dejó en blanco intencionalmente

Filtrado de paquetes y seguridad básica Medidas

- 1 Conceptos preliminares subyacentes al filtrado de paquetes
cortafuegos
- 2 conceptos de filtrado de paquetes
- 3 iptables: el firewall heredado de Linux
Programa de Administración
- 4 nftables: la administración del cortafuegos de Linux
Programa
- 5 Construcción e instalación de un cortafuegos independiente

Esta página se dejó en blanco intencionalmente

Conceptos Preliminares

Filtrado de paquetes subyacente cortafuegos

A sitio pequeño puede tener acceso a Internet a través de varios medios, como una línea T1, un cable módem, DSL, inalámbrico, una conexión PPP o ISDN, o cualquier número de otros medios. La computadora conectada directamente a Internet es un punto de enfoque para los problemas de seguridad. Ya sea que tenga una computadora o una red de área local (LAN) de computadoras vinculadas, el enfoque inicial para un sitio pequeño estará en la máquina con la conexión directa a Internet. Esta máquina será la máquina cortafuegos.

El término *firewall* tiene varios significados según su implementación y propósito. En este punto inicial del libro, cortafuegos significa la máquina conectada a Internet. Aquí es donde se implementarán sus principales políticas de seguridad para el acceso a Internet. La tarjeta de interfaz de red externa de la máquina cortafuegos es el punto de conexión, o puerta de enlace, a Internet. El propósito de un firewall es proteger lo que está de su lado de esta puerta de enlace de lo que está del otro lado.

Una configuración simple de firewall a veces se denomina *firewall de bastión* porque es la principal línea de defensa contra ataques desde el exterior. Muchas de sus medidas de seguridad están montadas desde este defensor de su reino. En consecuencia, se hace todo lo posible para proteger este sistema.

Detrás de esta línea de defensa está su única computadora o su grupo de computadoras. El propósito de la máquina cortafuegos podría ser simplemente servir como punto de conexión a Internet para otras máquinas en su LAN. Es posible que esté ejecutando servicios privados locales detrás de este firewall, como una impresora compartida o sistemas de archivos compartidos. O quizás desee que todas sus computadoras tengan acceso a Internet. Una de sus máquinas podría albergar sus registros financieros privados. Es posible que desee tener acceso a Internet desde esta máquina, pero no quiere que nadie entre. En algún momento, es posible que desee ofrecer sus propios servicios a Internet. Una de las máquinas podría alojar su propio sitio web para Internet. Otro podría funcionar como su servidor de correo o puerta de enlace. Su configuración y objetivos determinarán sus políticas de seguridad.

El propósito del cortafuegos es hacer cumplir las políticas de seguridad que defina. Estas políticas reflejan las decisiones que ha tomado sobre qué servicios de Internet desea que estén accesibles

a sus computadoras, qué servicios desea ofrecer al mundo desde sus computadoras, qué servicios desea ofrecer a usuarios o sitios remotos específicos, y qué servicios y programas desea ejecutar localmente para su uso privado. Las políticas de seguridad tienen que ver con el control de acceso y el uso autenticado de servicios, programas y archivos privados o protegidos en sus computadoras.

Los sistemas domésticos y de pequeñas empresas no enfrentan todos los problemas de seguridad de un sitio corporativo más grande, pero las ideas y los pasos básicos son los mismos. Simplemente no hay tantos factores a considerar, y las políticas de seguridad a menudo son menos estrictas que las de un sitio corporativo. El énfasis está en proteger su sitio del acceso no deseado desde Internet. Un firewall de filtrado de paquetes es un enfoque común y una pieza de la seguridad de la red y el control del acceso hacia y desde el exterior.

Por supuesto, tener un firewall no significa que esté totalmente protegido. La seguridad es un proceso, no una pieza de hardware. Por ejemplo, incluso con un firewall instalado, es posible descargar spyware o adware o hacer clic en un correo electrónico creado con fines maliciosos, lo que abre la computadora y, por lo tanto, la red al ataque. Es tan importante contar con medidas para mitigar los ataques exitosos como gastar recursos en un firewall. El uso de las mejores prácticas dentro de su red ayudará a disminuir la posibilidad de una explotación exitosa y le dará resiliencia a su red.

Algo a tener en cuenta es que el paradigma de Internet se basa en la premisa de la transparencia de extremo a extremo. Las redes entre las dos máquinas que se comunican están destinadas a ser invisibles. De hecho, si falla un dispositivo de red en algún lugar de la ruta, la idea es que el tráfico entre las dos máquinas de punto final se redirija silenciosamente.

Idealmente, los cortafuegos deberían ser transparentes. Sin embargo, rompen el paradigma de Internet mediante la introducción de un único punto de falla dentro de las redes entre las dos máquinas de punto final. Además, no todas las aplicaciones de red utilizan protocolos de comunicación que pasan fácilmente a través de un simple cortafuegos de filtrado de paquetes. No es posible pasar cierto tráfico a través de un firewall sin soporte de aplicaciones adicionales o una tecnología de firewall más sofisticada.

Para complicar aún más el problema ha sido la introducción de la traducción de direcciones de red (NAT, o "enmascaramiento" en la jerga de Linux). NAT permite que una computadora actúe en nombre de muchas otras computadoras al traducir sus solicitudes y reenviarlas a su destino. El uso de NAT junto con las direcciones IP privadas RFC 1918 ha evitado de manera efectiva una inminente escasez de direcciones IPv4. La combinación de espacio de direcciones NAT y RFC 1918 hace que la transmisión de algunos tipos de tráfico de red sea difícil, imposible, compleja o costosa.

Nota

Muchos dispositivos de enrutador, especialmente aquellos para DSL, módems por cable e inalámbricos, se venden como firewalls, pero no son más que enrutadores habilitados para NAT. No realizan muchas de las funciones de un verdadero cortafuegos, pero separan lo interno de lo externo. Tenga cuidado al comprar un enrutador que dice ser un cortafuegos pero solo proporciona NAT. Aunque algunos de estos productos tienen algunas buenas características, las configuraciones más avanzadas a veces no son posibles.

Una complicación final ha sido la proliferación de protocolos multimedia y peer-to-peer (P2P) utilizados tanto en software de comunicación en tiempo real como en juegos populares en red. Estos protocolos son la antítesis de la tecnología de firewall actual. Hoy en día, se deben construir e implementar soluciones de software específicas para cada protocolo de aplicación. Las arquitecturas de firewall para el manejo fácil y económico de estos protocolos están en proceso en los grupos de trabajo de los comités de estándares.

Es importante tener en cuenta que la combinación de cortafuegos, DHCP y NAT presenta complejidades que hacen que los sitios tengan que comprometer la seguridad del sistema hasta cierto punto para utilizar los servicios de red que los usuarios desean. Las pequeñas empresas a menudo tienen que implementar varias LAN y configuraciones de red más complejas para satisfacer las distintas necesidades de seguridad de los hosts locales individuales.

Antes de entrar en los detalles del desarrollo de un cortafuegos, este capítulo presenta los conceptos y mecanismos subyacentes básicos en los que se basa un cortafuegos de filtrado de paquetes. Estos conceptos incluyen un marco de referencia general sobre qué es la comunicación de red, cómo se identifican los servicios basados en la red, qué es un paquete y los tipos de mensajes e información que se envían entre las computadoras en una red.

El modelo de redes OSI

El modelo OSI (Open System Interconnection) representa un marco de red basado en capas. Cada capa del modelo OSI proporciona una funcionalidad distinta en relación con las otras capas. El modelo OSI contiene siete capas, como se muestra en la Figura 1.1.

A veces se hace referencia a las capas por número, siendo la capa más baja (Física) la Capa 1 y la capa más alta (Aplicación) la Capa 7. Si escucha que alguien se refiere a una

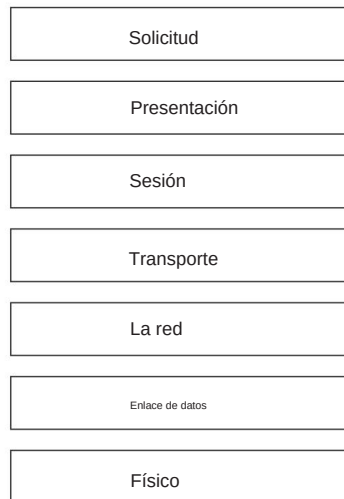


Figura 1.1 Las siete capas del modelo OSI

"Interruptor de capa 3", se refiere a la tercera capa del modelo OSI. Como persona interesada en la seguridad y la detección de intrusos, debe conocer las capas del modelo OSI para comprender completamente las rutas de ataque que podrían comprometer sus sistemas.

Cada capa del modelo OSI es importante. Los protocolos que usa todos los días, como IP, TCP, ARP y otros residen en las diversas capas del modelo. Cada capa tiene su propia función y papel distintos en el proceso de comunicación.

La capa física del modelo OSI está ocupada por los propios medios, como el cableado. y protocolos de señalización relacionados; en otras palabras, transfiriendo los bits. En su mayor parte, la capa física preocupa menos al analista de intrusiones en la red más allá de proteger los dispositivos y el cableado. Debido a que este libro realmente no habla mucho sobre la seguridad física (¿qué tan interesantes son las cerraduras de las puertas?), tampoco dedicaré más tiempo a la capa física del modelo OSI. Naturalmente, los pasos que toma para proteger los cables físicos son diferentes de los que tomaría para intentar proteger los dispositivos inalámbricos.

La siguiente capa por encima de la física es la capa de enlace de datos. La capa de enlace de datos transfiere los datos a través del medio dado y es responsable de cosas como la detección y recuperación de errores en la transmisión. La capa de enlace de datos también es la capa donde se definen las direcciones de hardware físico, como la dirección de control de acceso a medios (MAC) de una tarjeta Ethernet.

Por encima de la capa de enlace de datos, la capa de red es la tercera capa más importante en las redes IP. Esta capa es responsable del direccionamiento lógico y el enrutamiento de los datos. IP es un protocolo de capa de red, lo que significa que la capa de red es la capa en la que se utilizan las direcciones IP y las máscaras de subred. Los enrutadores y algunos conmutadores operan en la Capa 3, moviendo datos entre redes divididas tanto lógica como físicamente.

La cuarta capa, la capa de transporte, es la capa principal sobre la que se puede construir la confiabilidad. Los protocolos que existen en la capa de transporte incluyen TCP y UDP. La quinta capa es la capa de sesión, dentro de la cual se construyen sesiones entre puntos finales. La sexta capa, Presentación, es principalmente responsable de la comunicación con la capa de Aplicación por encima de ella, y también define cosas como el cifrado que se utilizará. Finalmente, la capa de Aplicación es responsable de mostrar los datos al usuario o aplicación.

Además del modelo OSI, existe otro modelo, el modelo DARPA, a veces llamado modelo de referencia TCP/IP, que tiene solo cuatro capas. El modelo OSI se ha convertido en el modelo tradicional o de facto en el que tienen lugar la mayoría de las discusiones en red.

A medida que los datos se mueven desde una aplicación hacia abajo en las capas del modelo OSI, el protocolo en la siguiente capa inferior puede agregar su propia información a los datos. Estos datos generalmente consisten en un encabezado que se antepone a los datos del siguiente nivel más alto, aunque a veces también se agrega un avance. Este proceso, llamado *encapsulación*, continúa hasta que los datos se transmiten a través del medio físico. En el caso de Ethernet, los datos se conocen como *trama* . cuando se transmite. Cuando la trama de Ethernet llega a su destino, la trama comienza el proceso de subir las capas del modelo OSI, y cada capa lee la información del encabezado (y posiblemente el tráiler) de la capa correspondiente del remitente. Este proceso se llama *demultiplexación*.

Protocolos sin conexión frente a protocolos orientados a la conexión

En algunas capas del modelo OSI, los protocolos se pueden definir en términos de una de sus propiedades, sin conexión u orientados a la conexión. Esta definición se refiere a los métodos que contiene el protocolo para proporcionar cosas tales como control de errores, control de flujo, segmentación de datos y reensamblaje de datos.

Piense en los protocolos orientados a la conexión en términos de una llamada telefónica. Generalmente hay un protocolo aceptable para hacer una llamada telefónica y tener una conversación. La persona que realiza la llamada, el iniciador de la comunicación, abre la comunicación marcando un número de teléfono. La persona (o máquina, como es el caso cada vez mayor) en el otro extremo recibe la solicitud para iniciar una conversación telefónica. La solicitud de iniciar una conversación telefónica se indica con frecuencia mediante el timbre del teléfono en el extremo del receptor. El receptor levanta el teléfono y dice "Hola" o alguna otra forma de saludo. El iniciador luego reconoce este saludo respondiendo del mismo modo. En este punto, es seguro decir que se ha iniciado la conversación o la configuración de la llamada. A partir de este punto, la conversación se produce. Durante la conversación, si algo sale mal, como un ruido en la línea, una de las partes puede pedirle a la otra que repita su última afirmación. La mayoría de las veces, cuando se completa una llamada, ambos lados indicarán que han terminado con la conversación diciendo "Adiós". La llamada termina poco después.

El ejemplo que se acaba de dar proporciona una imagen semirazonable de un protocolo orientado a la conexión como TCP. Hay excepciones a la regla, al igual que puede haber excepciones o errores con el protocolo TCP. Por ejemplo, a veces la llamada inicial falla por razones tecnológicas fuera del control de la persona que llama o del receptor.

Por otro lado, un protocolo sin conexión es más parecido a una postal enviada por correo. Después de que el remitente escribe un mensaje en la postal y lo coloca en el buzón, el remitente (presuntamente) pierde el control sobre ese mensaje. El remitente no recibe ningún reconocimiento directo de que la postal se haya entregado correctamente. Los ejemplos de protocolos sin conexión incluyen UDP e IP.

Próximos pasos

A partir de aquí, pasaré a una mirada más detallada al Protocolo de Internet (IP).

Sin embargo, le recomiendo encarecidamente que dedique más tiempo a aprender sobre el modelo OSI y los protocolos en sí. El conocimiento de los protocolos y el modelo OSI es vital para un profesional de la seguridad. Recomendando enfáticamente el libro *TCP/IP Illustrated, Vol ume 1, Second Edition*, de Kevin R. Fall y W. Richard Stevens como un libro indispensable en el escritorio de cualquier profesional de la computación.

El Protocolo de Internet

El Protocolo de Internet es la base sobre la que opera Internet. Junto con los protocolos de otras capas, la capa IP proporciona comunicaciones para innumerables aplicaciones. IP es un protocolo sin conexión que proporciona funciones de enrutamiento de capa 3.

Direccionamiento IP y división en subredes

Como ya sabe, pero me siento obligado a escribir, las direcciones IP para la versión 4 de IP (IPv4) consisten en cuatro números de 8 bits separados por puntos, conocidos como la notación "cuadrángulo punteado" o "decimal punteado". Para IP versión 6 (IPv6), las direcciones son de 128 bits y se muestran como ocho grupos de dígitos hexadecimales, cada uno separado por dos puntos. Aunque aparentemente todo el mundo entiende o al menos ha visto una dirección IP, ciertamente parece que cada vez menos entienden las subredes y las máscaras de subred que son una parte importante del esquema de direccionamiento IP. Esta sección analiza brevemente el direccionamiento IP y las subredes.

Las direcciones IPv4 se dividen en diferentes clases en lugar de ser una dirección completamente plana espacio. Las clases de direcciones IPv4 se muestran en la Tabla 1.1.

En la práctica, solo las direcciones de las Clases A a C son para uso general de Internet. Sin embargo, algunos lectores pueden tener experiencia con direcciones de Clase D, que se utilizan con frecuencia para multidifusión. La clase E es el rango experimental y no asignado.

Direcciones IP especiales

Hay tres casos especiales principales de direcciones IP:

- **Dirección de red 0:** como se indica en las direcciones de Clase A, la dirección de red 0 no es se utiliza como parte de una dirección enrutable para IPv4. Se representa como ::0 para IPv6. Cuando se usa como una dirección de origen, su único uso legal es durante la inicialización cuando un host intenta que un servidor le asigne dinámicamente su dirección IP. Cuando se usa como destino, solo la dirección 0.0.0.0 tiene significado, y luego solo para la máquina local como referencia a sí misma, o como una convención para referirse a una ruta predeterminada.
- **Dirección de red de bucle invertido 127:** como se indica en las direcciones de clase A, la red la dirección 127 no se utiliza como parte de una dirección enrutable. La dirección de loopback IPv6 se representa como 0:0:0:0:0:0:0:1 o, más típicamente, se reduce a ::1. Las direcciones de loopback se refieren a una interfaz de red privada compatible con el sistema operativo. La interfaz se utiliza como mecanismo de direccionamiento para los servicios basados en la red local. En otras palabras, los clientes de la red local lo usan para dirigirse a los servidores locales. El tráfico de loopback permanece completamente dentro del sistema operativo. Nunca se pasa a una interfaz de red física. Por lo general, 127.0.0.1 es la única dirección de loopback utilizada para IPv4 y ::1 para IPv6, en referencia al host local.
- **Direcciones de difusión:** las direcciones de difusión son direcciones especiales que se aplican a todos hosts en una red. Hay dos categorías principales de direcciones de difusión. Las transmisiones limitadas no se enrutan, pero se entregan a todos los hosts conectados al mismo segmento de red física. Todos los bits tanto en la red como en los campos de host de la dirección IP se establecen en uno, como 255.255.255.255. Las transmisiones dirigidas a la red se enrutan y se entregan a todos los hosts en una red específica. El campo de red de la dirección IP especifica una red. El campo host suele ser todos unos, como en 192.168.10.255. Alternativamente, a veces puede ver la dirección especificada como la dirección de red, como en 192.168.10.0. IPv6 no utiliza direcciones de difusión en este sentido, sino más bien utiliza multidifusión para comunicarse con grupos de hosts.

Tabla 1.1 Direcciones de Internet

Clase	Rango de direcciones
A	0.0.0.0 a 127.255.255.255
B	128.0.0.0 a 191.255.255.255
C	192.0.0.0 a 223.255.255.255
D	224.0.0.0 a 239.255.255.255
E y sin asignar 240.0.0.0 a 255.255.255.255	

El encabezado de IPv4 consta de varios campos y tiene un total de 20 bytes, sin incluir los campos de opciones opcionales que se pueden incluir como parte del encabezado. El encabezado de IPv6 es un encabezado de 320 bits. El encabezado de IPv4 se muestra en la Figura 1.2.

El encabezado de IPv4 comienza con 4 bits que indican la versión, actualmente la versión 4, seguido de 4 bits que indican la longitud del encabezado. El encabezado en sí es normalmente de 20 bytes más opciones opcionales. La longitud máxima del encabezado IPv4 es de 60 bytes. El siguiente campo, punto de código de servicios diferenciados (DSCP), tiene una longitud de 6 bits seguido de 2 bits para la notificación de congestión explícita (ECN).

El primer número de una dirección IP indica la clase de la dirección, porque cada número El número dentro de la notación decimal con puntos es de 8 bits, los valores posibles para cada número son de 0 a 255. La clase indica el número predeterminado de bits dedicados a la porción de red de la dirección versus el número de bits dedicados a la identificación del host con un número determinado. habla a. La división entre la porción de red de la dirección y la porción de host de la dirección es importante porque es la base del direccionamiento de subred.

Aparte de las clases, hay tres tipos de direcciones disponibles en Internet: unidifusión, multidifusión y difusión. Las direcciones de unidifusión corresponden a una única interfaz de red en Internet. Las direcciones de multidifusión corresponden a un grupo de hosts que solicitan ser incluidos dentro

Versión HDR	Len	TOS	Longitud total del datagrama	
identificación del paquete			Flags	Desplazamiento de fragmentos
TTL	Protocolo		Suma de comprobación del encabezado	
Dirección de la fuente				
Dirección de destino				
(Opciones de IP)				(Relleno)

Figura 1.2 El encabezado de IPv4

Tabla 1.2 Máscaras de subred predeterminadas

Clase	Máscara de subred predeterminada
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

ese grupo Las direcciones de difusión son utilizadas por hosts que desean enviar datos a cada host en una subred determinada.

Cada clase de dirección tiene una máscara de subred predeterminada que indica la división entre la porciones de red y host de una dirección dada. Eso es bastante complicado, así que daré ejemplos y luego habrá un cuestionario. ¡Es una broma!

Las máscaras de subred predeterminadas para las Clases A a C se proporcionan en la Tabla 1.2.

Sin duda, ha visto y escrito estos números al configurar los ajustes de red.

Como se indicó anteriormente, la máscara de subred indica la división entre la red y las porciones de host de una dirección IP. La parte no enmascarada, conocida como parte del host, de la dirección comprende la red lógica en la que reside un host determinado. En otras palabras, con una máscara de subred Clase C de 255.255.255.0, puede haber un total de 254 hosts en la red. Un lector astuto podría notar que en realidad hay 256 direcciones pero solo 254 hosts.

Dentro de una red IP lógica dada, hay dos direcciones especiales, la dirección de red y la dirección de transmisión. Esto es cierto independientemente del tamaño de la red. En el caso del ejemplo de subred Clase C, la dirección de red termina en .0 y la dirección de transmisión termina en .255.

Como ilustra la Tabla 1.2, del total de 32 bits en una dirección IPv4, una máscara de subred Clase A usa 8 bits, una máscara de subred Clase B usa 16 bits y una máscara de subred Clase C usa 24 bits.

Cuando una red se divide a lo largo de los límites de clases de direcciones tradicionales utilizando la máscara de subred predeterminada, se dice que es una red con *clase*. Como era de esperar, hay momentos en los que sería beneficioso utilizar una red mucho más pequeña. Por ejemplo, dos enrutadores IP que solo necesitan transmitir entre sí utilizarían una red Clase C completa utilizando subredes tradicionales con clase. Afortunadamente, también es posible dividir en subredes sin clase.

Al utilizar la división en subredes *sin clases*, oficialmente denominada enrutamiento entre dominios sin clases (CIDR), puede dividir las redes según las necesidades agregando o restando bits de la máscara de subred. Esto es útil para la conservación de direcciones porque permite que los administradores de la red trator para personalizar el tamaño de la red en función de la necesidad y la conveniencia más que en los límites de clase. Volviendo al ejemplo con dos enrutadores que se comunican únicamente entre sí, mediante CIDR, un administrador de red puede crear una red de solo dos hosts con la máscara de subred resultante 255.255.255.252.

Llevaré ese ejemplo un poco más lejos. Los dos enrutadores solo necesitan comunicarse entre sí dentro de esta red para que puedan enrutar el tráfico entre dos redes IP diferentes. El administrador de red asigna a un enrutador la dirección 192.168.0.1 y al otro enrutador la dirección 192.168.0.2 y les da a ambos una máscara de subred de 255.255.255.252. Dada esa máscara de subred, hay dos direcciones IP disponibles con las que se puede direccionar un host.

La dirección de red para esta red lógica es 192.168.0.0 y la dirección de transmisión es

192.168.0.3. Con CIDR, el administrador de la red ahora puede usar el resto de la red 192.168.0, siguiendo las reglas de CIDR, para otros hosts.

Con frecuencia verá una notación de subred a la que se hace referencia con /NN, siendo NN el número de bits a enmascarar. Por ejemplo, una Clase C tiene 24 bits para la porción de red de la dirección, lo que significa que podría denominarse /24. Una Clase B sería /16 y una Clase A sería /8. Volviendo al ejemplo de dos enrutadores, la notación CIDR para esta dirección es /30 porque la subred consume 30 bits de la dirección.

¿Por qué es importante dividir en subredes? La respuesta simple es que una subred define el espacio de transmisión más grande posible para una red determinada. Dentro de una subred determinada, un host puede enviar una transmisión a todos los demás hosts de esa subred. Sin embargo, en la práctica, las transmisiones están más limitadas por las limitaciones físicas que por las limitaciones lógicas que presentan las máscaras de subred. Puede conectar solo una cantidad limitada de dispositivos a un conmutador antes de que pueda (repito, puede) comenzar a ver una degradación del rendimiento y probablemente divida la red en secciones lógicas más pequeñas. Sin la división en subredes, tendríamos un espacio de direcciones muy grande y plano, que sería mucho más lento que el direccionamiento jerárquico que se usa actualmente.

Fragmentación de IP

Hay momentos en que un datagrama IP es más grande que el tamaño máximo permitido para el medio físico en el que viajará. Este tamaño máximo permitido se conoce como Unidad de transmisión máxima o MTU. Si un datagrama IP es mayor que la MTU del medio, será necesario dividir el datagrama en fragmentos más pequeños antes de transmitirlo. Para Ethernet, la MTU es de 1500 bytes. El proceso de dividir un datagrama IP en partes más pequeñas se llama *fragmentación*.

La fragmentación se maneja en la capa IP del modelo OSI y, por lo tanto, es transparente para los protocolos de capa superior, como TCP y UDP. Como administrador, debe preocuparse por la fragmentación en la medida en que puede afectar el rendimiento de la aplicación si se pierde uno de los fragmentos de un segmento grande. Además, como administrador de seguridad, debe comprender la fragmentación porque ha sido una ruta de ataque en el pasado. Tenga en cuenta, sin embargo, que cualquier enrutador intermediario u otros dispositivos dentro de la ruta de comunicación pueden causar fragmentación y es posible que ni siquiera lo sepa.

Radiodifusión y multidifusión

Cuando un dispositivo desea enviar datos a otros dispositivos en el mismo segmento de red, puede enviar los datos a una dirección especial conocida como dirección de transmisión para realizar esta tarea. Por otro lado, se envía un multicast a los dispositivos que pertenecen al grupo de multicast, algunas veces llamados *suscriptores*.

Imagine una red grande y plana en la que cada computadora y dispositivo estén conectados entre sí. En tal entorno, cada dispositivo de red ve el tráfico de todos los demás dispositivos de red. En este tipo de red, cada dispositivo ve el tráfico y determina si le importa el tráfico en cuestión. En otras palabras, busca ver si los datos están dirigidos a él o a otro dispositivo. Si los datos se dirigen al dispositivo, pasa los datos a las capas del modelo OSI. A nivel de interfaz para Ethernet, el dispositivo busca su MAC

o la dirección de hardware asociada con la propia interfaz de red. Recuerde que las direcciones IP solo son relevantes para los protocolos en las capas superiores del modelo OSI.

Además de las tramas dirigidas al propio dispositivo, existen dos casos especiales que pueden causar una interfaz para aceptar datos y pasarlos a capas superiores. Estos dos casos especiales son multicast y broadcast. La multidifusión es un método para transmitir datos a un subconjunto de dispositivos que se dice que están suscritos a esa multidifusión.

Por otro lado, las transmisiones están destinadas a ser procesadas por cada dispositivo que recibe ellos. Principalmente hay dos tipos de transmisiones disponibles: transmisión dirigida y transmisión limitada. Con mucho, las transmisiones dirigidas son las más comunes. Los dispositivos que intentan configurarse a través de DHCP, BOOTP u otro protocolo de configuración utilizan difusiones limitadas. Se envía una transmisión limitada a la dirección 255.255.255.255 y nunca debe pasar por un enrutador. Esta es una sugerencia clave para cualquier persona que controle un enrutador u otro dispositivo de enrutamiento, como un firewall de enrutamiento. Si recibe un paquete en la interfaz de su enrutador externo con acceso a Internet dirigido a 255.255.255.255, es probable que haya un dispositivo mal configurado o, más probablemente, que un posible atacante esté intentando sondear su red. Es posible que vea una transmisión limitada en una interfaz interna para un enrutador si tiene dispositivos que se configuran solos al arrancar usando DHCP.

Las transmisiones dirigidas son la forma más común de transmisión que verá en cualquier red. Esto se debe a que el protocolo de resolución de direcciones (ARP, que se analiza más adelante) utiliza las difusiones para determinar la dirección MAC de una dirección IP en una subred determinada. Una transmisión dirigida es una transmisión que está limitada por la red o subred en la que reside el dispositivo emisor. De manera predeterminada, cuando la interfaz de un enrutador encuentra una difusión dirigida, no la transmite a otras subredes a través del enrutador. La mayoría de los enrutadores se pueden configurar para permitir este comportamiento; sin embargo, se debe tener cuidado de no crear una tormenta de transmisión al reenviar transmisiones a través de un enrutador. Una transmisión de subred es una trama de datos dirigida a la dirección de transmisión en una subred determinada. Esta dirección de transmisión varía según la máscara de subred para la subred dada. En una subred Clase C (255.255.255.0 /24), la dirección de transmisión predeterminada es la dirección más alta disponible, por lo tanto, la que termina en .255. Por ejemplo, en la red 192.168.1.0/24, la dirección de transmisión es 192.168.1.255.

ICMP

Algunos dicen que ocupar un lugar especial dentro de la capa IP es ICMP. Probablemente esté familiarizado con ICMP porque ping usa ICMP. ICMP, o Protocolo de mensajes de control de Internet, tiene varios usos, incluido ser el protocolo subyacente para el comando ping. Hay 15 funciones dentro de ICMP, cada una indicada por un código de tipo. Por ejemplo, el tipo para una solicitud de eco ICMP (piense: ping) es 8; la respuesta a esa solicitud, acertadamente titulada respuesta de eco, es de tipo 0. Dentro de los diferentes tipos también pueden existir códigos para especificar la condición para el tipo dado. Los tipos y códigos de los mensajes ICMP se muestran en la Tabla 1.3.

El tipo y el código de un mensaje ICMP están contenidos en el encabezado ICMP, que se muestra en la Figura 1.3.

Tabla 1.3 Códigos y tipos de mensajes ICMP

Escribe	Código	Descripción
0	0	Respuesta de eco
3		Destino inalcanzable
	0	Red no disponible
	1	Host inalcanzable
	2	Protocolo inalcanzable
	3	Puerto inalcanzable
	4	Fragmentación necesaria y conjunto de DF
	5	Ruta de origen fallida
	6	Red de destino desconocida
	7	Host de destino desconocido
	8	Host de origen aislado
	9	Red de destino prohibida administrativamente
	10	Host de destino prohibido administrativamente
	11	Red inaccesible para el tipo de servicio
	12	Host inaccesible para el tipo de servicio
	13	Comunicación Prohibida Administrativamente
	14	Violación de precedencia de host
	15	Corte de precedencia en efecto
4		Source Quench (en desuso)
5		redirigir
	0	Redirección de red
	1	Redirección de host
	2	Tipo de servicio y redirección de red
	3	Tipo de servicio y redirección de host
8	0	Solicitud de eco
9	0	Anuncio de enrutador
10	0	Selección de enrutador
11		Tiempo excedido
	0	TTL (tiempo de vida) excedido en tránsito
	1	Se excedió el tiempo de reensamblaje del fragmento
12	0	Problema de parámetro
13	0	Solicitud de marca de tiempo
14	0	Respuesta de marca de tiempo
15	0	Solicitud de información (en desuso)
discontinuo	0	Respuesta de información (en desuso)
17	0	Solicitud de máscara de dirección (en desuso)
18	0	Respuesta de máscara de dirección (en desuso)

Tipo de mensaje	Código de subtipo	Suma de verificación
ID de mensaje		Secuencia de números
(Estructura de datos ICMP opcional)		

Figura 1.3 El encabezado ICMP

Mecanismos de Transporte

Protocolo de Internet define un protocolo de capa de red del modelo OSI. También hay otros protocolos de capa de red, pero me concentraré únicamente en IP porque es, con mucho, el protocolo de capa de red más popular que se usa en la actualidad. Por encima de la capa de Red en el modelo OSI está la capa de Transporte. Como era de esperar, la capa de transporte tiene su propio conjunto de protocolos. Dos de los protocolos de la capa de transporte son de interés: UDP y TCP. Esta sección examina cada uno de estos protocolos.

UDP

UDP, o Protocolo de datagramas de usuario, es un protocolo sin conexión que se utiliza para servicios tales como consultas de DNS, SNMP y RADIUS. Al no tener conexión, UDP es similar a un tipo de protocolo de "disparar y olvidar". El cliente envía un paquete UDP, a veces denominado datagrama, y asume que el servidor recibirá el paquete. Depende de un protocolo de capa superior ensamblar los paquetes en orden. El encabezado UDP, que se muestra en la Figura 1.4, tiene una longitud de 8 bytes.

El encabezado UDP comienza con el número de puerto de origen y el número de puerto de destino. El siguiente es la longitud de todo el paquete, incluidos los datos. Obviamente, debido a que el encabezado en sí tiene una longitud de 8 bytes, el valor mínimo para esta parte del encabezado es 8. La parte final del encabezado UDP es la suma de comprobación, que incluye tanto el encabezado como los datos.

TCP

TCP, una abreviatura de Transmission Control Protocol, es un protocolo orientado a la conexión que se usa frecuentemente con IP. Referirse a TCP como orientado a la conexión significa que proporciona un servicio confiable a las capas superiores. Recuerde la analogía de la conversación telefónica

Puerto de origen	Puerto de destino
Longitud del paquete UDP	Suma de verificación

Figura 1.4 El encabezado UDP

Puerto de origen			Puerto de destino		
Secuencia de números					
Número de acuse de recibo					
Datos Compensar	No usado	Banderas		Ventana	
Suma de verificación				puntero urgente	

Figura 1.5 El encabezado TCP

dado anteriormente en este capítulo. Como en esa analogía, dos aplicaciones que deseen comunicarse mediante TCP también deben establecer una conexión (a veces denominada sesión). El encabezado TCP se muestra en la Figura 1.5.

Como puede ver en la Figura 1.5, el encabezado TCP de 20 bytes es significativamente más complicado que los otros encabezados de protocolo que se muestran en este capítulo. Al igual que el encabezado UDP, el encabezado TCP comienza con los puertos de origen y de destino. La combinación de los puertos de origen y destino junto con las direcciones IP del remitente y el receptor identifica la conexión. El encabezado TCP tiene un número de secuencia de 32 bits y un reconocimiento de 32 bits. Recuerde que TCP es un protocolo orientado a la conexión y brinda un servicio confiable. Los números de secuencia y reconocimiento son los mecanismos principales (pero no los únicos) que se utilizan para proporcionar esa confiabilidad. A medida que los datos pasan a la capa de Transporte, TCP divide los datos en lo que cree que es el tamaño más apropiado. Estas piezas se conocen como *segmentos TCP*. A medida que TCP envía datos por la pila de protocolos, crea un número de secuencia que indica el primer byte de datos para el segmento dado. En el extremo opuesto de la comunicación, el receptor envía un acuse de recibo indicando que se ha recibido el segmento. El remitente mantiene un temporizador en marcha y, si no se recibe un acuse de recibo a tiempo, el segmento se volverá a enviar.

Otro mecanismo de confiabilidad que proporciona TCP es una suma de verificación tanto en el encabezado y los datos. Si la suma de verificación establecida en el encabezado por el remitente no coincide con la suma de verificación calculada por el receptor, el receptor no enviará un acuse de recibo. Si un acuse de recibo se pierde en tránsito, es probable que el remitente envíe otro segmento con el mismo número de secuencia. En tal caso, el receptor simplemente descartará el segmento repetido.

Se utiliza un campo de 4 bits para la longitud del encabezado, incluidas las opciones proporcionadas como parte del encabezado. Hay varios indicadores de bits individuales dentro del encabezado TCP: URG, ACK, PSH, RST, SYN, FIN, NS, CWR y ECE. En la Tabla 1.4 se incluye una descripción de estas banderas.

El campo Ventana de 16 bits se utiliza para proporcionar un mecanismo de ventana deslizante. El receptor establece el número de ventana para indicar el tamaño que el receptor está listo para recibir, comenzando con el número de reconocimiento. Esta es una forma de control de flujo para TCP.

Tabla 1.4 Indicadores de encabezado TCP

Bandera	Descripción
URG	Indica que se debe examinar la parte del puntero urgente del encabezado.
POSB DE RI	Indica que se debe examinar el número de acuse de recibo.
Persiana	Indica que el receptor debe entregar estos datos a la siguiente capa lo antes posible.
PRIMERA	Indica que se debe restablecer la conexión.
VISIÓN	Inicia una conexión.
NS	Protección de ocultamiento de nonce ECN.
CWR	Ventana de congestión Reducida para indicar que se configuró un paquete con el indicador ECE y respondió el control de congestión.
CEPE	Si el indicador SYN se establece en 1, este indicador indica que el par TCP es compatible con ECN. Si SYN se establece en 0, este indicador indica que se estableció un indicador de congestión experimentada en un encabezado IP.
FINAL	Indica que el remitente (puede estar en cualquier lado de la conexión) ha terminado de enviar datos.

El puntero urgente de 16 bits indica el desplazamiento desde el número de secuencia donde urgente los datos terminan. Esto permite al remitente indicar que hay datos que deben ser tratados con urgencia. Este puntero también se puede utilizar junto con el indicador PSH .

Ahora que tiene una idea del encabezado TCP, es hora de examinar cómo funciona TCP. las conexiones se establecen y terminan.

Conexiones TCP

Mientras que UDP es un protocolo sin conexión, TCP es un protocolo orientado a la conexión. Con UDP no existe el concepto de conexión; solo hay un emisor y un receptor de un datagrama UDP. Con TCP, por otro lado, cualquier lado de la conexión puede enviar o recibir datos, posiblemente haciendo ambas cosas al mismo tiempo. Esto es lo que hace que TCP sea un protocolo full-duplex. El proceso de establecer una conexión TCP a veces se denomina protocolo *de enlace de tres vías*; verá por qué en breve.

Con un protocolo orientado a la conexión, existe un conjunto específico de procedimientos que se llevan a cabo para establecer una conexión TCP. Durante este proceso, existen varios estados para la conexión TCP. A continuación se detallan los procedimientos de establecimiento de la conexión y sus estados correspondientes.

El lado de la comunicación que desea iniciar la conexión (cliente) envía un TCP segmento con el indicador SYN establecido, así como un número de secuencia inicial (ISN) y el número de puerto para la conexión al otro lado, normalmente conocido como el lado del servidor de la conexión. Esto se denomina con frecuencia paquete SYN o segmento SYN , y se dice que la conexión está en el estado SYN_SENT .

El lado del servidor de la conexión responde con un segmento TCP con el indicador SYN establecido, así como el indicador ACK establecido. Además, el servidor establece el ISN con un valor uno superior a

el ISN enviado por el cliente. Esto se conoce con frecuencia como un paquete SYN-ACK o SYN -ACK segmento, y se dice que la conexión está en el estado SYN_RCVD .

Luego, el cliente reconoce el SYN-ACK enviando otro segmento con el ACK conjunto de banderas e incrementando el ISN en uno. Esto completa el protocolo de enlace de tres vías y se dice que la conexión está en un estado ESTABLECIDO .

Al igual que el protocolo de inicio de conexión, también existe un protocolo de conexión terminación. El protocolo para terminar una conexión TCP consta de cuatro pasos, a diferencia de los tres para el establecimiento de la conexión. El paso adicional se debe a la naturaleza de dúplex completo de una conexión TCP en la medida en que cualquiera de los lados puede estar enviando datos en un momento dado.

El cierre de una conexión en un lado se logra cuando ese lado envía un segmento TCP con la bandera FIN puesta. Cualquiera de los lados de la conexión puede enviar un FIN para indicar que ha terminado de enviar datos. El otro lado aún puede enviar datos. Sin embargo, en la práctica, después de recibir un FIN , normalmente comenzará la secuencia de terminación de la conexión. Para esta discusión, llamaré al lado que desea terminar la conexión, el lado del cliente.

El proceso de finalización comienza cuando el cliente envía un segmento con el indicador FIN establecido, conocido como estado CLOSE_WAIT en el lado del servidor y FIN_WAIT_1 en el lado del cliente. Después de que el servidor recibe el FIN , el servidor envía un ACK de regreso al cliente, incrementando el número de secuencia en uno. En este punto, el cliente entra en FIN_WAIT_2 estado. El servidor también indica a sus propios protocolos de capa superior que la conexión ha terminado. A continuación, el servidor cierra la conexión, lo que hace que se envíe un segmento con el indicador FIN al cliente, lo que a su vez hace que el servidor entre en un estado LAST_ACK mientras que el cliente entra en un estado TIME_WAIT . Finalmente, el cliente reconoce este FIN con un ACK e incrementa el número de secuencia en uno, lo que hace que la conexión entre en estado CERRADO . Debido a que las conexiones TCP pueden ser terminadas por cualquiera de los lados, una conexión TCP puede existir en un modo semicerrado en el que un extremo ha iniciado la secuencia FIN pero el otro lado no lo ha hecho.

Las conexiones TCP también se pueden terminar enviando un segmento con el indicador de reinicio (RST) establecido. Esto le dice al otro lado que use un método de liberación abortivo. Esto es lo opuesto a la terminación normal de una conexión TCP, a veces denominada liberación ordenada.

Una parte opcional de la secuencia de conexión TCP es el establecimiento del Tamaño Máximo de Segmento (MSS). El MSS es la porción máxima de datos que el extremo respectivo de la comunicación puede recibir. Debido a que el MSS es el tamaño máximo que puede recibir un extremo determinado de la conexión, está perfectamente bien enviar una porción de datos más pequeña que el MSS. En general, debe considerar que un MSS más grande es bueno, teniendo en cuenta que la fragmentación debe evitarse porque agrega sobrecarga (los bytes adicionales para cada encabezado IP y TCP requeridos para los paquetes fragmentados).

No olvide el protocolo de resolución de direcciones

El Protocolo de resolución de direcciones, o ARP, es el protocolo utilizado para vincular un dispositivo físico, como una tarjeta de red, a una dirección IP. Los dispositivos de red utilizan una dirección de 48 bits (conocida como dirección MAC) que es única en todos los dispositivos de un segmento determinado. Aunque a veces los dispositivos tienen la misma dirección MAC, esto es bastante raro dentro del mismo segmento de red.

Al capturar tráfico en una red, encontrará paquetes ARP en diferentes frecuencias a medida que los dispositivos se ubican entre sí a medida que pasan el tráfico. Las solicitudes ARP se transmiten para que todos los dispositivos las vean. Sin embargo, la mayoría de las respuestas ARP son de unidifusión, por lo que solo el dispositivo solicitante verá la respuesta. El tráfico ARP normalmente no pasa entre los segmentos de la red. Por lo tanto, un enrutador se puede configurar para proporcionar un servicio de proxy ARP para que pueda responder a las solicitudes de ARP en múltiples segmentos de red.

Nombres de host y direcciones IP

A la gente le gusta usar palabras para nombrar cosas, como dar nombres a las computadoras como `mycomputer.mydomain.example.com`. Técnicamente, no es la computadora la que recibe el nombre, sino la interfaz de red en la computadora. Si la computadora tiene varias tarjetas de red, cada tarjeta tendrá normalmente un nombre y una dirección diferentes y lo más probable es que esté en una red diferente en un subdominio diferente.

Los elementos del nombre de host están separados por puntos. En el caso de `mycomputer.mydomain.example.com`, el elemento más a la izquierda, `mycomputer`, es el nombre de host. `.mydomain`, `.example` y `.com` son elementos de los dominios de los que es miembro esta tarjeta de red. Los dominios de red son árboles jerárquicos. ¿Qué es un dominio? Es una convención de nombres. El árbol de dominio jerárquico representa la naturaleza jerárquica de la base de datos del Servicio de nombres de dominio (DNS) global. Los mapas de DNS entre los nombres simbólicos que las personas dan a las computadoras y las redes y las direcciones numéricas que usa la capa IP para identificar de manera única las interfaces de red.

Mapas de DNS en ambas direcciones: dirección IP a nombre y nombre a dirección IP. Cuando hace clic en una URL en su navegador web, se consulta la base de datos DNS para encontrar la dirección IP única asociada con ese nombre de host. La dirección IP se pasa a la capa IP para usarla como dirección de destino en el paquete.

Direcciones IP y direcciones Ethernet

Mientras que la capa IP identifica los hosts de la red por su dirección IP de 32 o 128 bits, la capa de enlace o subred identifica la tarjeta Ethernet por su dirección Ethernet única de 48 bits o dirección MAC, que el fabricante graba en la tarjeta y también puede ser configurado por el usuario. Las direcciones IP se pasan entre los hosts de puntos finales para identificarse. Las direcciones Ethernet se pasan entre hosts y enrutadores adyacentes.

Por lo general, la dirección Ethernet podría ignorarse en una discusión sobre el firewall. La dirección Ethernet del hardware de capa 2 no es visible para el nivel de IP de capa 3 o el nivel de transporte de capa 4. Como verá en capítulos posteriores, el programa de administración de firewall de Linux tiene la capacidad extendida para acceder y filtrar la dirección MAC. Existen usos especializados para esta funcionalidad de firewall, pero es importante recordar que las direcciones Ethernet no pasan de extremo a extremo a través de la red. Las direcciones Ethernet se pasan entre interfaces de red adyacentes o hosts y enrutadores. No se pasan a través de un enrutador sin cambios.

Enrutamiento: Obtener un paquete de aquí a allá

Es probable que ni un sitio residencial ni la mayoría de las empresas ejecuten protocolos de enrutamiento como RIP u OSPF. En estos casos, las tablas de enrutamiento se configuran de forma estática, a mano. Hay una pista ahí. Si está ejecutando un protocolo de enrutamiento como RIP, es probable que no sea necesario; podría operar una red más eficiente sin esa sobrecarga innecesaria. Por lo general, la mayoría de los sitios tienen un dispositivo de puerta de enlace predeterminado, que es la red a la que se envían los paquetes de interfaz cuando se desconoce la ruta de la dirección de destino. El proveedor de servicios suele proporcionar una sola dirección de enrutador, que es la puerta de enlace de Internet predeterminada para la red local del sitio.

Puertos de servicio: la puerta a los programas en su sistema

Los servicios basados en red son programas que se ejecutan en una máquina a la que pueden acceder otras computadoras en la red. Los puertos de servicio identifican los programas y las sesiones o conexiones individuales que tienen lugar. Los puertos de servicio son los nombres numéricos de los diferentes servicios basados en la red. También se usan como identificadores numéricos para los extremos de una conexión particular entre dos programas. Los números de puerto de servicio oscilan entre 0 y 65535.

Los programas de servidor (es decir, los *demonios*) escuchan las conexiones entrantes en un puerto de servicio asignado a ellos. Por convención histórica, a los principales servicios de red se les asignan números de puerto bien conocidos o famosos en el rango inferior del 1 al 1023. Estos números de puerto : las asignaciones a servicios están coordinadas por la Autoridad de Números Asignados de Internet (IANA) como un conjunto de convenciones o estándares acordados universalmente.

Un servicio anunciado es simplemente un servicio disponible a través de Internet desde su puerto asignado. Si su máquina no ofrece un servicio en particular y alguien intenta conectarse al puerto asociado con ese servicio, no sucederá nada. Alguien está llamando a la puerta, pero nadie vive allí para responder. Por ejemplo, HTTP se asigna al puerto 80 (aunque, de nuevo, no hay ninguna razón por la que no pueda ejecutarlo en el puerto 8080, 20943 o cualquier otro puerto disponible). Si su máquina no está ejecutando un servidor web basado en HTTP y alguien intenta conectarse al puerto 80, el programa cliente recibe un mensaje de cierre de conexión como un mensaje de error de su máquina que indica que el servicio no se ofrece.

Los números de puerto más altos de 1024 a 65535 se denominan *puertos sin privilegios*. Sirven para un doble propósito. En su mayor parte, estos puertos se asignan dinámicamente al extremo del cliente de una conexión. La combinación de pares de números de puerto de cliente y servidor, junto con sus respectivas direcciones IP de host y el protocolo de transporte utilizado, identifica de manera única la conexión.

Además, los puertos en el rango 1024 a 49151 están registrados en la IANA. Estos puertos se pueden usar como parte del grupo general sin privilegios, pero también están asociados con servicios particulares como SOCKS o servidores X Window. Originalmente, la idea era que los servicios ofrecidos en los puertos superiores no se ejecutaran con privilegios de root . Ellos

eran para uso de programas no privilegiados a nivel de usuario. La convención puede o no ser válida en cualquier caso individual.

Asignaciones de nombre de servicio a número de puerto

Las distribuciones de Linux se suministran con una lista de números de puerto de servicio comunes. La lista se encuentra en el archivo `/etc/services`.

Cada entrada consta de un nombre simbólico para un servicio, el número de puerto asignado, el protocolo (TCP o UDP) sobre el que se ejecuta el servicio y cualquier apodo opcional para el servicio.

La Tabla 1.5 enumera algunas asignaciones comunes de nombre de servicio a número de puerto, tomadas de Red Hat Linux.

Tabla 1.5 Asignaciones comunes de nombre de servicio a número de puerto

Nombre del puerto	Número de puerto/Protocolo	Alias
ftp	21/tcp	-
ssh	22/tcp	-
SMTP	25/tcp	correo
dominio	53/tcp	nombre del servidor
dominio	53/udp	nombre del servidor
http	80/tcp	www www-http
pop3	110/tcp	pop-3
nntp	119/tcp	leernoticias untp
ntp	123/udp	-
https	443/tcp	-

Tenga en cuenta que los nombres simbólicos asociados con los números de puerto varían según la distribución y versión de Linux. Los nombres y alias difieren; los números de puerto no.

También tenga en cuenta que los números de puerto están asociados con un protocolo. La IANA ha intentado asignar el mismo número de puerto de servicio a los protocolos TCP y UDP, independientemente de si un servicio en particular usa ambos modos de transporte. La mayoría de los servicios utilizan un protocolo u otro. El Servicio de Nombres de Dominio usa ambos.

Una conexión TCP típica: visitar un sitio web remoto

A modo de ilustración, una conexión TCP común va a un sitio web a través de su navegador (conectándose a un servidor web). Esta sección ilustra los aspectos del establecimiento de la conexión y la comunicación continua que serán relevantes para el filtrado de paquetes IP en capítulos posteriores.

¿Lo que sucede? Como se muestra en la Figura 1.6, un servidor web se está ejecutando en una máquina en algún lugar, esperando una solicitud de conexión en el puerto de servicio TCP 80. Hace clic en el enlace de una URL en su navegador web. Parte de la URL se analiza en un nombre de host; el nombre de host se traduce a la dirección IP del servidor web; y a su navegador se le asigna un puerto sin privilegios (por ejemplo, el puerto TCP 14000) para la conexión. Se construye un mensaje HTTP para el servidor web. Se encapsula en un mensaje TCP, se envuelve en un encabezado de paquete IP y se envía. Para nuestros propósitos, el encabezado contiene los campos que se muestran en la Figura 1.6.

Se incluye información adicional en el encabezado que no es visible en el filtrado de paquetes. Sin embargo, describir los números de secuencia asociados con SYN y ACK flags ayuda a aclarar lo que sucede durante el apretón de manos de tres vías. Cuando el programa cliente envía su primer mensaje de solicitud de conexión, el indicador SYN va acompañado de un número de secuencia de sincronización. El cliente solicita una conexión con el servidor y pasa un número de secuencia que utilizará como punto de partida para numerar el resto de los bytes de datos que enviará el cliente.

El paquete se recibe en la máquina servidor. Se envía al puerto de servicio 80. El servidor está escuchando el puerto 80, por lo que recibe una notificación de una solicitud de conexión entrante (el indicador de solicitud de sincronización de conexión SYN) de la dirección IP de origen y el par de puertos (su dirección IP, 14000). El servidor asigna un nuevo socket en su extremo (dirección IP del servidor web, 80) y lo asocia con el socket del cliente.

El servidor web responde con un acuse de recibo (ACK) al mensaje SYN, junto con su propia solicitud de sincronización (SYN), como se muestra en la Figura 1.7. La conexión ahora está medio abierta.

En el encabezado SYN-ACK se incluyen dos campos que no son visibles para el nivel de filtrado de paquetes. Junto con el indicador ACK, el servidor incluye el número de secuencia del cliente incrementado por

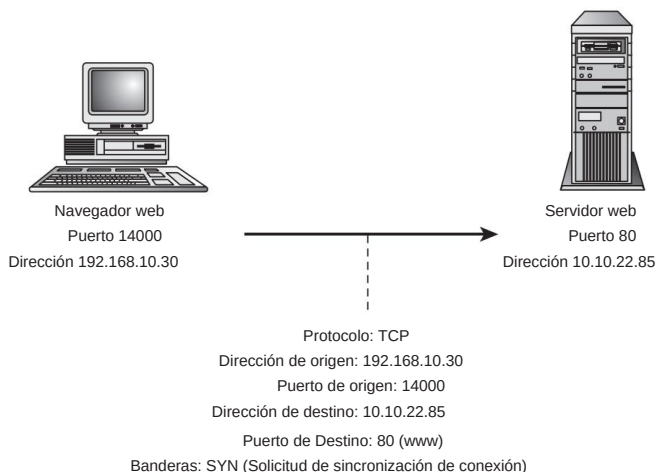


Figura 1.6 Solicitud de conexión de un cliente TCP

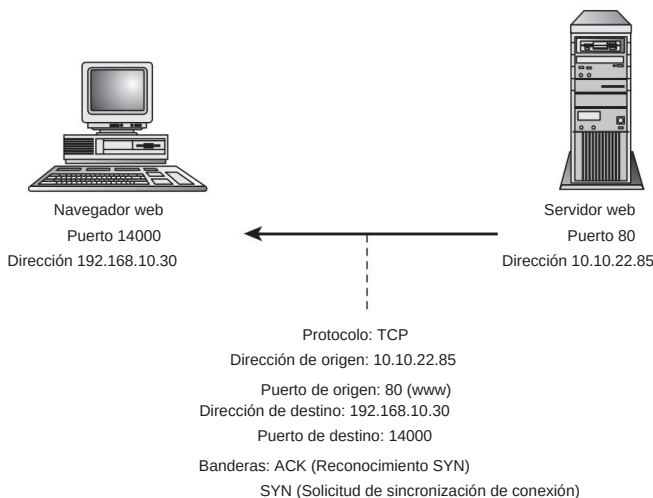


Figura 1.7 Confirmación de solicitud de conexión de un servidor TCP

el número de bytes de datos contiguos recibidos. El propósito del reconocimiento es reconocer los datos a los que se refirió el cliente por su número de secuencia. El servidor reconoce esto incrementando el número de secuencia del cliente, diciendo efectivamente que recibió los datos, y el número de secuencia más uno es el siguiente byte de datos que el servidor espera recibir.

El cliente es libre de tirar su copia del mensaje SYN original ahora que el servidor ha acusado recibo del mismo.

El servidor también establece el indicador SYN en su primer mensaje. Al igual que con el primer mensaje del cliente, el indicador SYN va acompañado de un número de secuencia de sincronización. El servidor pasa su propio número de secuencia inicial para su mitad de la conexión.

Este primer mensaje es el único mensaje que el servidor enviará con el indicador SYN establecido. Este y todos los mensajes subsiguientes tienen el indicador ACK establecido. La presencia del indicador ACK en todos los mensajes del servidor, en comparación con la falta de un indicador ACK en el primer mensaje del cliente, será una diferencia crítica cuando lleguemos a la información disponible para construir un firewall.

Su máquina recibe este mensaje y responde con su propio reconocimiento, después de lo cual se establece la conexión. La Figura 1.8 muestra una representación gráfica de esto.

A partir de aquí, tanto el cliente como el servidor establecen el indicador ACK. El indicador SYN no se establecerá de nuevo por ninguno de los programas.

Con cada confirmación, los programas cliente y servidor incrementan el número de secuencia de su proceso asociado por la cantidad de bytes de datos contiguos recibidos, más uno, lo que indica la recepción de esa cantidad de bytes de datos e indica el siguiente byte de datos en el flujo que el programa espera recibir.

A medida que su navegador recibe la página web, su máquina recibe mensajes de datos del servidor web con encabezados de paquetes, como se muestra en la Figura 1.9.

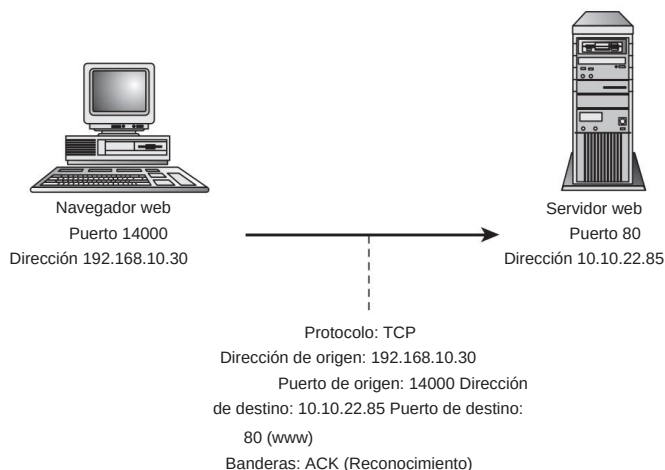


Figura 1.8 Establecimiento de conexión TCP

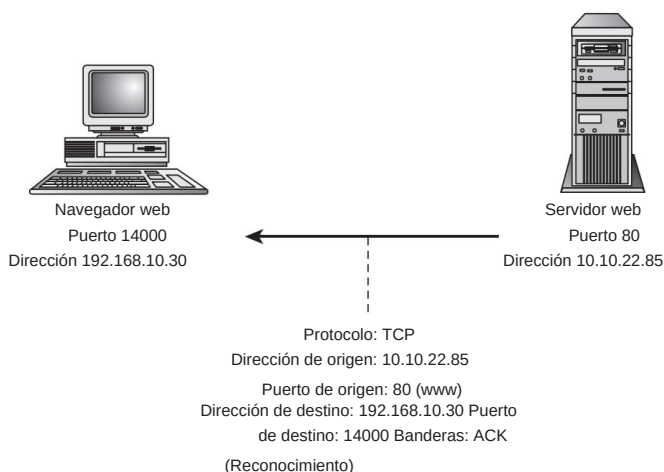


Figura 1.9 Una conexión TCP de servidor a cliente en curso

Resumen

Los ejemplos sencillos de este capítulo ilustran la información en la que se basan los cortafuegos de filtrado de paquetes IP. El Capítulo 2, "Conceptos de filtrado de paquetes", se basa en esta introducción y describe cómo se utilizan los tipos de mensajes ICMP, UDP y TCP y los números de puerto de servicio para definir un cortafuegos de filtrado de paquetes.

Esta página se dejó en blanco intencionalmente

Conceptos de filtrado de paquetes

¿Qué es un cortafuegos? A lo largo de los años, el término ha cambiado de significado. De acuerdo a RFC 2647, "Terminología de evaluación comparativa para el rendimiento del cortafuegos", un cortafuegos es "un dispositivo o grupo de dispositivos que aplica una política de control de acceso entre redes". Esta definición es muy amplia, y de hecho lo es a propósito. Un firewall puede abarcar muchas capas del modelo OSI y puede referirse a un dispositivo que filtra paquetes, realiza inspección y filtrado de paquetes, implementa una política en una aplicación en una capa superior, o hace cualquiera de estos y más.

Un firewall sin estado, o sin estado, generalmente realiza algún filtrado de paquetes basado únicamente en la capa IP (Capa 3) del modelo OSI, aunque a veces los protocolos de capa superior están involucrados en este tipo de firewall. Un ejemplo de este tipo de dispositivo podría incluir un enrutador de borde que se ubica en el borde de una red e implementa una o más listas de acceso para evitar que varios tipos de tráfico malicioso ingresen a la red. Algunos podrían argumentar que este tipo de dispositivo no es un cortafuegos en absoluto. Sin embargo, ciertamente parece encajar dentro de la definición de RFC.

Una lista de acceso de enrutadores fronterizos podría implementar muchas políticas diferentes según la interfaz en la que se recibió el paquete. Es típico filtrar ciertos paquetes en el borde de la red que se conecta a Internet. Estos paquetes se analizan más adelante en este capítulo.

A diferencia de un firewall sin estado, un firewall con estado es aquel que realiza un seguimiento de los paquetes vistos previamente dentro de una sesión determinada y aplica la política de acceso a los paquetes en función de lo que ya se ha visto para la conexión dada. Un firewall con estado también implica las capacidades básicas de filtrado de paquetes de un firewall sin estado. Un firewall con estado, por ejemplo, hará un seguimiento de las etapas del protocolo de enlace de tres vías TCP y rechazará los paquetes que aparecen fuera de secuencia para ese protocolo de enlace. Al no tener conexión, UDP es un poco más complicado de manejar para un firewall con estado porque no hay estado del que hablar. Sin embargo, un firewall con estado rastrea los intercambios UDP recientes para garantizar que un paquete que se ha recibido se relacione con un paquete saliente reciente.

Una *puerta de enlace de nivel de aplicación* (ALG), a veces denominada *puerta de enlace de capa de aplicación*, es otra forma más de firewall. A diferencia del cortafuegos sin estado, que tiene conocimiento de la red y posiblemente de las capas de transporte, un ALG maneja principalmente la capa 7, la capa de aplicación del modelo OSI. Los ALG suelen tener un conocimiento profundo de los datos de la aplicación.

está pasando y puede así buscar cualquier desviación del tráfico normal para la aplicación en cuestión.

Un ALG normalmente residirá entre el cliente y el servidor real y, para todos los efectos, imitará el comportamiento del servidor real para el cliente. En efecto, el tráfico local nunca sale de la LAN y el tráfico remoto nunca ingresa a la LAN.

ALG a veces también se refiere a un módulo o pieza de software que ayuda a otro cortafuegos. Muchos cortafuegos vienen con un FTP ALG para admitir el canal de datos del modo de puerto de FTP, donde el cliente le dice al servidor a qué puerto local conectarse para que pueda abrir el canal de datos. El servidor inicia la conexión del canal de datos entrante (mientras que, por lo general, el cliente inicia todas las conexiones). Con frecuencia, se requieren ALG para pasar protocolos multimedia a través de un firewall porque las sesiones multimedia a menudo usan múltiples conexiones iniciadas desde ambos extremos y generalmente usan TCP y UDP juntos.

ALG es un proxy. Otra forma de proxy es un *proxy a nivel de circuito*. Los proxies a nivel de circuito no por lo general tienen conocimientos específicos de la aplicación, pero pueden aplicar políticas de autorización y acceso, y sirven como puntos de terminación en lo que de otro modo sería una conexión de extremo a extremo. SOCKS es un ejemplo de un proxy a nivel de circuito. El servidor proxy actúa como punto de terminación para ambos lados de la conexión, pero el servidor no tiene ningún conocimiento específico de la aplicación.

En cada uno de estos casos, el propósito del cortafuegos es hacer cumplir las políticas de control de acceso o seguridad que defina. Las políticas de seguridad son esencialmente sobre el control de acceso: quién puede y no puede realizar qué acciones en los servidores y redes bajo su control.

Aunque no es necesariamente específico de un firewall, los firewalls muchas veces realizan tareas adicionales, algunas de las cuales pueden incluir traducción de direcciones de red (NAT), verificación de antivirus, notificación de eventos, filtrado de URL, autenticación de usuarios y cifrado de capa de red.

Este libro cubre las ideas detrás de un firewall de filtrado de paquetes, tanto estático como dinámico, o sin estado y con estado. Cada uno de los enfoques mencionados controla a qué servicios se puede acceder y por quién. Cada enfoque tiene sus fortalezas y ventajas basadas en la información diferente disponible en las distintas capas del modelo de referencia OSI.

Se presentó el Capítulo 1, "Conceptos preliminares subyacentes a los cortafuegos de filtrado de paquetes". los conceptos y la información en los que se basa un cortafuegos. Este capítulo presenta cómo se utiliza esta información para implementar reglas de firewall.

Un firewall de filtrado de paquetes

En su nivel más básico, un firewall de filtrado de paquetes consta de una lista de reglas de aceptación y denegación. Estas reglas definen explícitamente qué paquetes se permitirán y no se permitirán a través de la interfaz de red. Las reglas del cortafuegos usan los campos de encabezado del paquete descritos en el Capítulo 1 para decidir si reenviar un paquete a su destino, desechar el paquete silenciosamente o bloquearlo y devolver una condición de error a la máquina emisora. Estas reglas se pueden basar en una amplia gama de factores, incluidas las direcciones IP de origen o destino, los puertos de origen y (más comúnmente) de destino, y partes de paquetes individuales, como las banderas de encabezado TCP, los tipos de protocolo, la dirección MAC, y más.

El filtrado de direcciones MAC no es común en los cortafuegos conectados a Internet. Usando el filtrado MAC, el firewall bloquea o permite solo ciertas direcciones MAC. Sin embargo, es muy probable que solo vea una dirección MAC, la del enrutador justo antes de su firewall. Esto significa que cada host en Internet parecerá tener la misma dirección MAC hasta donde su firewall pueda ver. Un error común entre los nuevos administradores de cortafuegos es intentar utilizar el filtrado MAC en un cortafuegos de Internet.

Usando un híbrido del modelo de referencia TCP/IP, un cortafuegos de filtrado de paquetes funciona en las capas de Red y Transporte, como se muestra en la Figura 2.1.

La idea general es que debe controlar con mucho cuidado lo que pasa entre Internet y la máquina que ha conectado directamente a Internet. En la interfaz externa a Internet, usted filtra individualmente lo que entra y lo que sale de la máquina de la manera más exacta y explícita posible.

Para una configuración de una sola máquina, podría ser útil pensar en la interfaz de red como un par de E/S. El cortafuegos filtra de forma independiente lo que entra y lo que sale a través de la interfaz. El filtrado de entrada y el filtrado de salida pueden, y probablemente lo hagan, tener reglas completamente diferentes. La Figura 2.2 muestra el procesamiento de reglas en un diagrama de flujo.

Esto suena bastante poderoso, y lo es; pero no es un mecanismo de seguridad infalible. Es solo una parte de la historia, solo una capa en el enfoque de varias capas para la seguridad de los datos. No todos los protocolos de comunicación de aplicaciones se prestan al filtrado de paquetes. Este tipo de

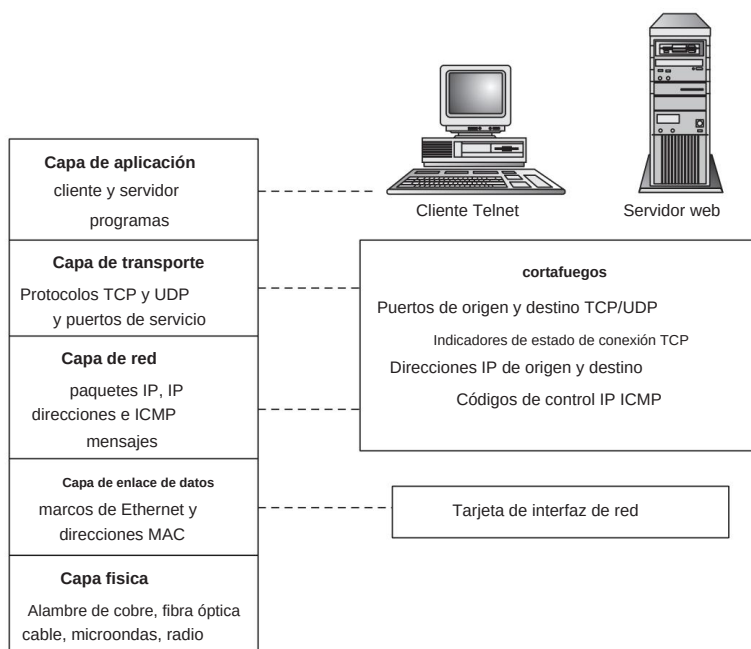


Figura 2.1 Ubicación del cortafuegos en el modelo de referencia TCP/IP

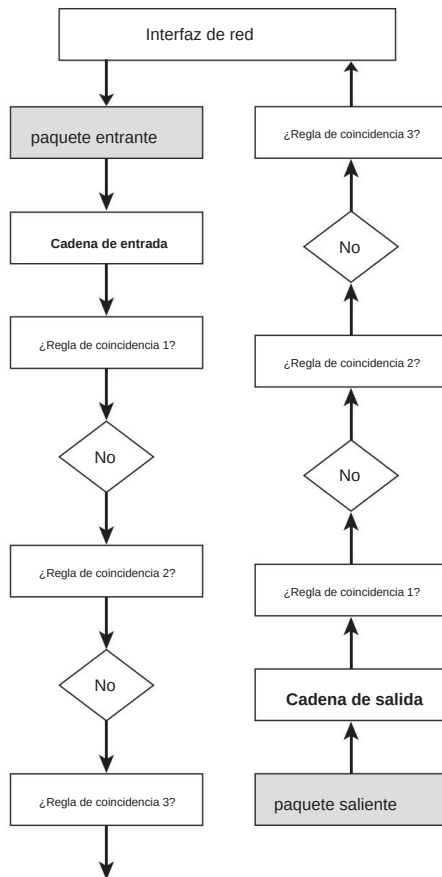


Figura 2.2 Diagrama de flujo de entrada y salida

el filtrado es de un nivel demasiado bajo para permitir una autenticación y un control de acceso detallados. Estos servicios de seguridad deben prestarse en niveles superiores. IP no tiene la capacidad de verificar que el remitente es quien dice ser. La única información de identificación disponible en este nivel es la dirección de origen en el encabezado del paquete IP. La dirección de origen se puede modificar con poca dificultad. Un nivel más arriba, ni la capa de red ni la capa de transporte pueden verificar que los datos de la aplicación sean correctos. No obstante, el nivel de paquete permite un control mayor y más simple sobre el acceso directo al puerto, el contenido del paquete y los protocolos de comunicación correctos que los que se pueden realizar fácil o convenientemente en niveles superiores.

Sin el filtrado a nivel de paquetes, el filtrado de alto nivel y las medidas de seguridad de proxy están paralizados o son potencialmente ineficaces. Hasta cierto punto, al menos, deben confiar en la corrección del protocolo de comunicación subyacente. Cada capa en la pila de protocolos de seguridad agrega otra pieza que otras capas no pueden proporcionar fácilmente.

Elección de una política de filtrado de paquetes predeterminada

Como se indicó anteriormente en este capítulo, un firewall es un dispositivo para implementar una política de control de acceso.

Una gran parte de esta política es la decisión sobre una política de firewall predeterminada.

Hay dos enfoques básicos para una política de firewall predeterminada:

• Denegar todo de forma predeterminada y permitir explícitamente el paso de los paquetes seleccionados.

• Aceptar todo de forma predeterminada y denegar explícitamente el paso de los paquetes seleccionados.

Sin duda, la política de negarlo todo es el enfoque recomendado. Este enfoque facilita la configuración de un firewall seguro, pero cada servicio y transacción de protocolo relacionado que desee debe habilitarse explícitamente (consulte la Figura 2.3). Esto significa que usted

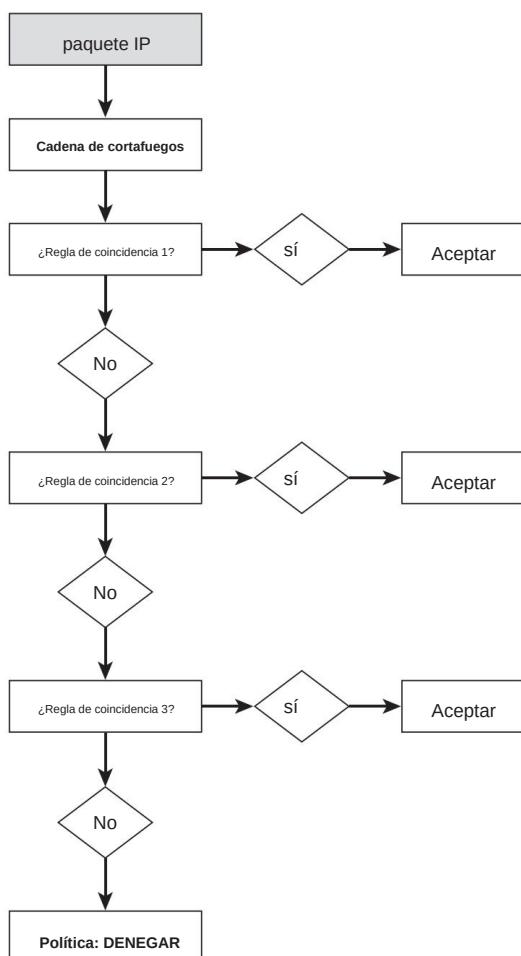


Figura 2.3 La política de negar todo por defecto

debe comprender el protocolo de comunicación para cada servicio que habilite. El enfoque de negar todo requiere más trabajo por adelantado para habilitar el acceso a Internet. Algunos productos comerciales de cortafuegos solo admiten la política de denegación de todo.

La política de aceptar todo hace que sea mucho más fácil empezar a trabajar de inmediato, pero lo obliga a anticipar cada tipo de acceso concebible que desee deshabilitar (vea la Figura 2.4). El peligro es que no anticipará un tipo de acceso peligroso hasta que sea demasiado tarde, o que luego habilitará un servicio inseguro sin bloquear primero el acceso externo a él. Al final, desarrollar un firewall seguro que acepte todo es mucho más trabajo, mucho más difícil, casi siempre mucho menos seguro y, por lo tanto, mucho más propenso a errores.

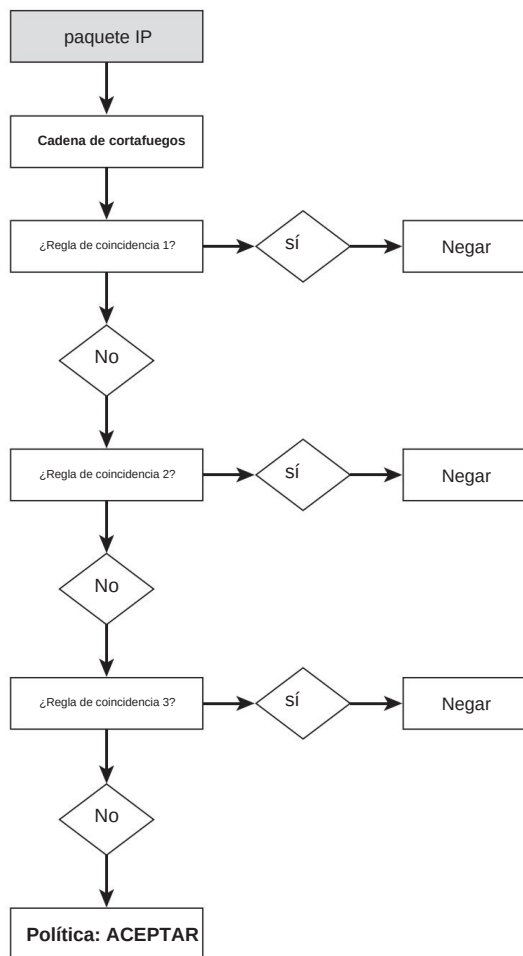


Figura 2.4 La política de aceptar todo por defecto

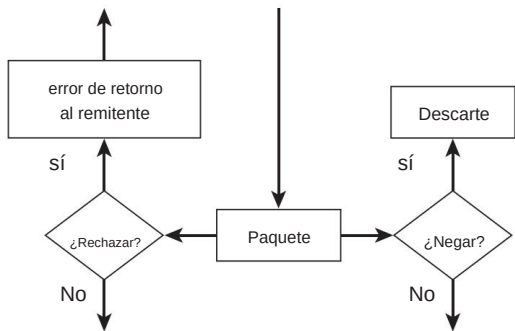


Figura 2.5 Rechazar versus denegar un paquete

Rechazar versus denegar un paquete

El mecanismo de firewall de Netfilter en iptables y nftables le brinda la opción de rechazar o descartar paquetes. ¿Cual es la diferencia? Como se muestra en la Figura 2.5, cuando se rechaza un paquete, el paquete se descarta y se devuelve un mensaje de error ICMP al remitente. Cuando se descarta un paquete, el paquete simplemente se descarta sin ninguna notificación al remitente.

La eliminación silenciosa del paquete es casi siempre la mejor opción, por tres razones. Primero, enviar una respuesta de error duplica el tráfico de red. La mayoría de los paquetes descartados se descartan porque son malévolos, no porque representen un intento inocente de acceder a un servicio que no ofrees. En segundo lugar, un paquete al que responde se puede utilizar en un ataque de denegación de servicio (DoS). En tercer lugar, cualquier respuesta, incluso un mensaje de error, proporciona información potencialmente útil al posible atacante.

Filtrado de paquetes entrantes

El lado de entrada del par de E/S de la interfaz externa, el conjunto de reglas de entrada, es el más interesante en términos de seguridad de su sitio. Como se mencionó anteriormente, puede filtrar según la dirección de origen, la dirección de destino, el puerto de origen, el puerto de destino, las marcas de estado de TCP y otros criterios. Aprenderá sobre todas estas piezas de información en un momento u otro en las siguientes secciones.

Filtrado de direcciones de origen remoto

A nivel de paquete, el único medio de identificar al remitente del paquete IP es la dirección de origen en el encabezado del paquete. Este hecho permite la posibilidad de suplantación de la dirección de origen, en la que el remitente coloca una dirección incorrecta en lugar de su propia dirección en el campo de origen. La dirección puede ser una dirección inexistente o puede ser una dirección legítima que pertenece a otra persona. Esto puede permitir que tipos desagradables entren en su

sistema al aparecer como tráfico local confiable; aparecer como usted mientras ataca otros sitios; fingiendo ser otra persona mientras te ataca; mantener su sistema atascado respondiendo a direcciones inexistentes; o de otra manera engañarlo en cuanto a la fuente de los mensajes entrantes.

Es importante recordar que, por lo general, no puede detectar direcciones falsificadas. La dirección podría ser legítimo y enrutable, pero podría no pertenecer al remitente del paquete. La siguiente sección describe las direcciones falsificadas que puede detectar.

Suplantación de identidad de la dirección de origen y direcciones ilegales

Hay varias clases principales de direcciones de origen que debe denegar en su interfaz externa en todos los casos. Estos son paquetes entrantes que afirman ser de lo siguiente:

- **Su dirección IP:** nunca verá paquetes entrantes legales que afirmen ser de su máquina. Debido a que la dirección de origen es la única información disponible y se puede modificar, esta es una de las formas de suplantación de direcciones legítimas que puede detectar en el nivel de filtrado de paquetes. Los paquetes entrantes que dicen ser de su máquina son falsificados. No puede estar seguro de si otros paquetes entrantes provienen de donde dicen ser. (Tenga en cuenta que algunos sistemas operativos fallan si reciben un paquete en el que tanto la dirección de origen como la de destino pertenecen a la interfaz de red del host).
- **Las direcciones de su LAN:** rara vez verá paquetes entrantes legales en la interfaz externa de Internet que afirmen ser de su LAN. Es posible ver dichos paquetes si la LAN tiene múltiples puntos de acceso a Internet, pero probablemente sea una señal de una red local mal configurada. En la mayoría de los casos, dicho paquete sería parte de un intento de obtener acceso a su sitio mediante la explotación de sus relaciones de confianza locales.
- **Direcciones IP privadas de clase A, B y C:** estos tres conjuntos de direcciones en el los rangos históricos de Clase A, B y C están reservados para su uso en LAN privadas. No están destinados para su uso en Internet. Como tal, estas direcciones pueden ser utilizadas por cualquier sitio internamente sin necesidad de comprar direcciones IP registradas. Su máquina nunca debería ver los paquetes entrantes de estas direcciones de origen:
 - A las direcciones privadas de Clase A se les asigna el rango de 10.0.0.0 a 10.255.255.255.
 - A las direcciones privadas de Clase B se les asigna el rango de 172.16.0.0 a 172.31.255.255.
 - A las direcciones privadas de Clase C se les asigna el rango de 192.168.0.0 a 192.168.255.255.
- **Direcciones IP de multidifusión de Clase D:** las direcciones IP en el rango de Clase D se configuran a un lado para su uso como direcciones de destino cuando se participa en una transmisión de red de multidifusión, como una transmisión de audio o una transmisión de video. Van desde 224.0.0.0 hasta 239.255.255.255. Su máquina nunca debería ver paquetes de estas direcciones de origen.

• **Direcciones IP reservadas de Clase E:** las direcciones IP en el rango de Clase E se reservaron para uso futuro y experimental y no se asignan públicamente. Van desde 240.0.0.0 hasta 247.255.255.255. Su máquina nunca debería ver paquetes de estas direcciones de origen, y lo más probable es que no lo haga. (Debido a que todo el rango de direcciones está reservado permanentemente hasta 255.255.255.255, el rango de Clase E se puede definir de manera realista como 240.0.0.0 a 255.255.255.255. De hecho, algunas fuentes definen el rango de direcciones de Clase E exactamente así).

• **Direcciones de interfaz de bucle invertido:** la interfaz de bucle invertido es una interfaz de red privada utilizada por el sistema Linux para servicios locales basados en la red. En lugar de enviar tráfico local a través del controlador de la interfaz de red, el sistema operativo toma un atajo a través de la interfaz de bucle invertido como una mejora del rendimiento. Por definición, el tráfico de loopback está dirigido al sistema que lo genera. No sale por la red. El rango de direcciones de loopback es de 127.0.0.0 a 127.255.255.255.

Por lo general, lo verá denominado 127.0.0.1, localhost o la interfaz de bucle invertido, lo.

• **Direcciones de difusión con formato incorrecto :** las direcciones de difusión son direcciones especiales que se aplican a todas las máquinas de una red. La dirección 0.0.0.0 es una dirección de fuente de transmisión especial. Una dirección de fuente de transmisión legítima será 0.0.0.0 o una dirección IP normal. Los clientes y servidores DHCP verán los paquetes de difusión entrantes desde la dirección de origen 0.0.0.0. Este es el único uso legal de esta dirección de origen. No es una dirección de origen de unidifusión punto a punto legítima. Cuando se ve como la dirección de origen en un paquete normal, punto a punto, que no se transmite, la dirección se falsifica o el remitente no está completamente configurado.

• **Direcciones de red 0 de clase A:** como se sugirió anteriormente, cualquier dirección de origen en el rango de 0.0.0.0 a 0.255.255.255 es ilegal como dirección de unidifusión.

• **Direcciones de red local de enlace:** los clientes DHCP a veces se asignan enlazar la dirección local cuando no pueden obtener una dirección de un servidor. Estas direcciones van desde 169.254.0.0 a 169.254.255.255.

• **NAT de nivel de operador:** hay direcciones IP marcadas para uso de proveedores de Internet que nunca deberían aparecer en una red pública, la Internet pública. Sin embargo, estas direcciones se pueden usar en escenarios de nube y, por lo tanto, si su servidor está alojado en un proveedor de nube, es posible que vea estas direcciones. Las direcciones NAT de nivel de operador van de 100.64.0.0 a 100.127.255.255.

• **Direcciones TEST-NET:** el espacio de direcciones de 192.0.2.0 a 192.0.2.255 está reservado para redes de prueba.

Bloqueo de sitios problemáticos

Otro esquema de filtrado de direcciones de origen común, pero que se usa con menos frecuencia, es bloquear todos los accesos desde una máquina seleccionada o, más típicamente, desde el bloque de direcciones IP de una red completa. Así es como la comunidad de Internet tiende a lidiar con los sitios problemáticos y los ISP que

no controle a sus usuarios. Si un sitio desarrolla una reputación como un mal vecino de Internet, otros sitios tienden a bloquearlo en todos los ámbitos.

A nivel individual, bloquear todo acceso desde redes seleccionadas es conveniente cuando las personas en la red remota habitualmente se molestan. Históricamente, esto se ha utilizado como un medio para combatir el correo electrónico no solicitado, y algunas personas llegaron a bloquear el rango de direcciones IP de todo un país.

Limitación de paquetes entrantes a hosts remotos seleccionados

Es posible que desee aceptar ciertos tipos de paquetes entrantes solo de sitios o individuos externos específicos. En estos casos, las reglas del cortafuegos definirán direcciones IP específicas o un rango limitado de direcciones IP de origen desde las que se aceptarán estos paquetes.

La primera clase de paquetes entrantes proviene de servidores remotos que responden a sus solicitudes. Aunque se puede esperar que algunos servicios, como los servicios web o FTP, provengan de cualquier lugar, otros servicios provendrán legítimamente solo de su ISP o de hosts de confianza especialmente elegidos. Ejemplos de servidores que probablemente se ofrecen solo a través de su ISP son el servicio de correo POP, las respuestas del servidor de nombres del Servicio de nombres de dominio (DNS) y posibles asignaciones de direcciones IP dinámicas o DHCP.

La segunda clase de paquetes entrantes proviene de clientes remotos que acceden a los servicios ofrecidos desde su sitio. Una vez más, aunque se puede esperar que algunas conexiones de servicios entrantes, como las conexiones a su servidor web, provengan de cualquier lugar, otros servicios locales se ofrecerán solo a unos pocos usuarios o amigos remotos de confianza. Ejemplos de servicios locales restringidos pueden ser ssh y ping.

Filtrado de direcciones de destino local

Filtrar los paquetes entrantes en función de la dirección de destino no es un gran problema.

En funcionamiento normal, su tarjeta de interfaz de red ignora los paquetes regulares que no están dirigidos a ella. La excepción son los paquetes de transmisión, que se transmiten a todos los hosts de la red.

La dirección IPv4 255.255.255.255 es la dirección de destino de difusión general. Se refiere a todos los hosts en el segmento de red físico inmediato y se denomina *transmisión limitada*. Una dirección de difusión se puede definir de manera más explícita como la dirección más alta en una subred determinada de direcciones IP. Por ejemplo, si la dirección de red de su ISP es 192.168.10.0 con una máscara de subred de 24 bits (255.255.255.0) y su dirección IP es 192.168.10.30, verá paquetes de transmisión dirigidos a 192.168.10.255 desde su ISP. Por otro lado, si tiene un rango más pequeño de direcciones IP, digamos /30 (255.255.255.252), entonces tiene un total de cuatro direcciones: una red, dos para hosts y la transmisión. Por ejemplo, considere la red 10.3.7.4/30. En esta red, 10.3.7.4 es la dirección de red, los dos hosts serían 10.3.7.5 y 10.3.7.6, y la dirección de transmisión sería 10.3.7.7. Este tipo de configuración de subred /30 generalmente se usa entre enrutadores, aunque las direcciones reales pueden variar. La única forma de saber cuál será la dirección de transmisión para una subred determinada es conocer tanto la dirección IP dentro de la subred como la máscara de subred. Estos tipos de transmisiones se denominan *transmisiones de subred dirigidas* y se envían a todos los hosts de esa red.

La dirección de transmisión a destino 0.0.0.0 es similar a la situación de los paquetes punto a punto que afirman ser de la dirección de origen de transmisión mencionada anteriormente, en la sección "Suplantación de identidad de la dirección de origen y direcciones ilegales". Aquí, los paquetes de difusión se dirigen a la dirección de origen 0.0.0.0 en lugar de a la dirección de destino, 255.255.255.255. En este caso, hay pocas dudas sobre la intención del paquete. Este es un intento de identificar su sistema como una máquina Linux. Por razones históricas, el código de red derivado de BSD UNIX devuelve un mensaje de error ICMP Tipo 3 en respuesta al uso de 0.0.0.0 como dirección de destino de difusión. Otros sistemas operativos descartan silenciosamente el paquete. Como tal, este es un buen ejemplo de por qué descartar versus rechazar un paquete hace la diferencia. En este caso, el mensaje de error en sí es lo que busca la sonda.

Filtrado de puertos de origen remotos

Las solicitudes entrantes y las conexiones de clientes remotos a sus servidores locales tendrán un puerto de origen en el rango sin privilegios. Si aloja un servidor web, todas las conexiones entrantes a su servidor web deben tener un puerto de origen entre 1024 y 65535. (La intención es que el puerto del servidor identifique el servicio, pero no la garantía. No puede estar seguro de que el servidor expect se está ejecutando en el puerto que espera).

Las respuestas entrantes y las conexiones de los servidores remotos con los que se comunicó tendrán el puerto de origen asignado al servicio en particular. Si se conecta a un sitio web remoto, todos los mensajes entrantes del servidor remoto tendrán el puerto de origen establecido en 80 (o cualquier puerto que haya especificado el cliente local), el número de puerto del servicio http .

Filtrado de puertos de destino locales

El puerto de destino en los paquetes entrantes identifica el programa o servicio en su computadora para el que está destinado el paquete. Al igual que con el puerto de origen, todas las solicitudes entrantes de clientes remotos a sus servicios generalmente siguen el mismo patrón y todas las respuestas entrantes de servicios remotos a sus clientes locales siguen un patrón diferente.

Las solicitudes entrantes y las conexiones de clientes remotos a sus servidores locales establecerán el puerto de destino en el número de servicio que asignó al servicio en particular. Por ejemplo, un paquete entrante destinado a su servidor web local normalmente tendría el puerto de destino establecido en 80, el número de puerto del servicio http .

Las respuestas entrantes de los servidores remotos con los que se comunicó tendrán un puerto de destino en el rango sin privilegios. Si se conecta a un sitio web remoto, todos los mensajes entrantes del servidor remoto tendrán un puerto de destino entre 1024 y 65535.

Filtrado de estado de conexión TCP entrante

Las reglas de aceptación de paquetes TCP entrantes pueden hacer uso de los indicadores de estado de conexión asociados con las conexiones TCP. Todas las conexiones TCP se adhieren al mismo conjunto de estados de conexión. Estos estados difieren entre el cliente y el servidor debido al protocolo de enlace de tres vías durante el establecimiento de la conexión. Como tal, el firewall puede distinguir entre el tráfico entrante de clientes remotos y el tráfico entrante de servidores remotos.

Los paquetes TCP entrantes de clientes remotos tendrán el indicador SYN establecido en el primer paquete recibido como parte del protocolo de enlace de establecimiento de conexión de tres vías. La primera solicitud de conexión tendrá el indicador SYN establecido, pero no el indicador ACK .

Los paquetes entrantes de servidores remotos siempre serán respuestas a la solicitud de conexión inicial iniciada desde su programa de cliente local. Cada paquete TCP recibido de un servidor remoto tendrá el indicador ACK establecido. Las reglas de firewall de su cliente local requerirán que todos los paquetes entrantes de servidores remotos tengan el indicador ACK establecido. Los servidores normalmente no intentan iniciar conexiones a programas cliente.

Sondas y exploraciones

Una *sonda* es un intento de conectarse u obtener una respuesta de un puerto de servicio individual. Un *escaneo* es una serie de sondeos a un conjunto de diferentes puertos de servicio. Los escaneos a menudo están automatizados.

Desafortunadamente, las sondas y los escaneos rara vez son inocentes. Es muy probable que sean la fase inicial de recopilación de información, en busca de vulnerabilidades interesantes antes de lanzar un ataque. Las herramientas de escaneo automatizado están muy extendidas y los esfuerzos coordinados de grupos de piratas informáticos son comunes. La seguridad, o la falta de seguridad, de muchos hosts en Internet, junto con la proliferación de gusanos, virus y máquinas zombis, hace que los escaneos sean un problema constante en Internet.

Exploraciones generales de puertos

Los escaneos generales de puertos son sondeos indiscriminados en un gran bloque de puertos de servicio, posiblemente en todo el rango (consulte la Figura 2.6). Estos escaneos son algo menos frecuentes, o, al menos, menos obvios, a medida que se encuentran disponibles herramientas ocultas más sofisticadas y dirigidas.

Exploraciones de puertos dirigidas

Los escaneos de puertos dirigidos buscan vulnerabilidades específicas (consulte la Figura 2.7). Las herramientas más nuevas y sofisticadas intentan identificar las versiones de hardware, sistema operativo y software. Estas herramientas están diseñadas para identificar objetivos que podrían ser propensos a una vulnerabilidad específica.

Destinos de puertos de servicio comunes

Los objetivos comunes a menudo se prueban y escanean individualmente. El atacante podría estar buscando una vulnerabilidad específica, como un servidor de correo no seguro, un servidor web sin parches o un daemon de mapa de puerto de llamada a procedimiento remoto (RPC) abierto .

Se puede encontrar una lista más extensa de puertos en <http://www.iana.org/assignments/port-numbers>. Solo se mencionan algunos puertos comunes aquí, para darle una idea:

• Los paquetes entrantes del puerto reservado 0 siempre son falsos. Este puerto no se usa legítimamente

• Las sondas de los puertos TCP 0 a 5 son una firma del programa ssan.

• Los ssh (22/tcp), smtp (25/tcp), dns (53/tcp/udp), pop-3 (110/tcp), imap (143/tcp) y snmp (161/udp), son los puertos de destino favoritos. Representan algunas de las aperturas potencialmente más vulnerables a un sistema, ya sea intrínsecamente, debido a

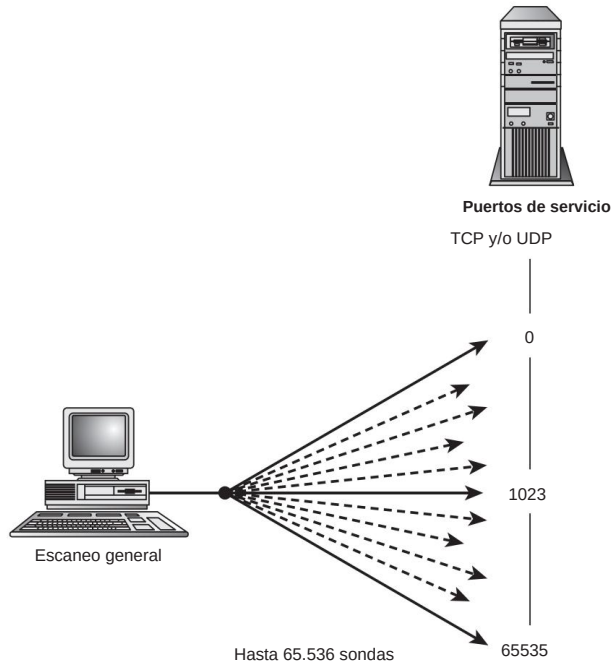


Figura 2.6 Un escaneo general de puertos

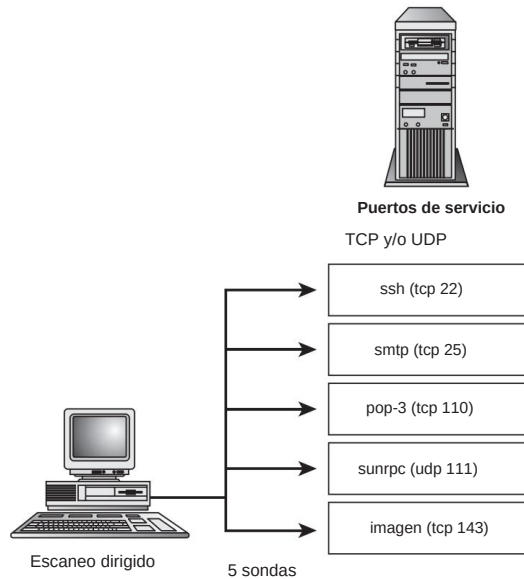


Figura 2.7 Un escaneo de puerto dirigido

errores comunes de configuración, o debido a fallas conocidas en el software. Debido a que estos servicios son tan comunes, son buenos ejemplos de por qué no desea ofrecerlos al mundo exterior u ofrecerlos con mucho cuidado con acceso externo controlado. Las sondas NetBIOS (137-139/tcp/udp) y Server Message Block (SMB) en Windows (445/tcp) son tediosas y comunes. Por lo general, no representan una amenaza para un sistema Linux a menos que se use Samba en el sistema. El objetivo típico es un sistema Windows, en este caso, pero los escaneos son muy comunes.

Escaneos sigilosos

Los escaneos de puertos Stealth, por definición, no están destinados a ser detectables. Se basan en cómo responde la pila de protocolos TCP a paquetes inesperados o paquetes con combinaciones de indicadores de estado ilegales. Por ejemplo, considere un paquete entrante que tiene el indicador ACK establecido pero no tiene una conexión relacionada. Si el ACK se enviara a un puerto con un servidor de escucha adjunto, la pila TCP no encontraría una conexión relacionada y devolvería un mensaje TCP RST para indicarle al remitente que restablezca la conexión. Si el ACK se enviara a un puerto no utilizado, el sistema simplemente devolvería un mensaje TCP RST como indicación de error, al igual que el firewall podría devolver un mensaje de error ICMP de forma predeterminada.

El problema se complica aún más porque algunos cortafuegos solo prueban el indicador SYN o el indicador ACK. Si no se establece ninguno, o si el paquete contiene alguna otra combinación de indicadores, la implementación del cortafuegos podría pasar el paquete al código TCP. Según la combinación de indicadores de estado de TCP y el sistema operativo que recibe el paquete, el sistema responderá con un RST o con un silencio. Este mecanismo se puede utilizar para ayudar a identificar el sistema operativo que está ejecutando el sistema de destino. En cualquiera de estos casos, es poco probable que el sistema receptor registre el evento.

Inducir a un host de destino para que genere un paquete RST de esta manera también se puede usar para mapear una red, determinando las direcciones IP de los sistemas que escuchan en la red. Esto es especialmente útil si el sistema de destino no es un servidor y su firewall se ha configurado para descartar silenciosamente los paquetes no deseados.

Evitar la paranoia: responder a los escaneos de puertos

Los registros del cortafuegos normalmente muestran todo tipo de intentos fallidos de conexión. Las sondas son lo más común que verá informado en sus registros.

¿La gente está probando su sistema con tanta frecuencia? Sí lo son. ¿Tu sistema está comprometido? No, no lo es. Bueno, no necesariamente. Los puertos están bloqueados. El cortafuegos está haciendo su trabajo. Estos son intentos fallidos de conexión que el cortafuegos denegó.

¿En qué momento decide personalmente denunciar una investigación? ¿En qué punto es lo suficientemente importante como para tomarse el tiempo para informarlo? ¿En qué momento dices que ya es suficiente y sigues con tu vida, o deberías estar escribiendo abuse@some.system cada vez?

No existen respuestas correctas. La forma en que responda es una cuestión de juicio personal y depende en parte de los recursos disponibles para usted, qué tan confidencial es la información en su sitio y qué tan crítica es la conexión a Internet para su sitio. Para sondeos y escaneos obvios, no hay una respuesta clara. Depende de su propia personalidad y nivel de comodidad cómo define personalmente una sonda seria y su conciencia social.

Con eso en mente, estas son algunas pautas viables.

Los intentos más comunes son una combinación de sondeo automatizado, errores, intentos legítimos basados en el historial de Internet, ignorancia, curiosidad y software que se comporta mal.

Casi siempre puede ignorar con seguridad los intentos individuales, aislados y de conexión única a telnet, ssh, ftp, finger o cualquier otro puerto para un servicio común que no está proporcionando. Las sondas y los escaneos son una realidad en Internet, son demasiado frecuentes y, por lo general, no representan un riesgo. Son como vendedores de puerta en puerta, llamadas telefónicas comerciales, números de teléfono incorrectos y correo postal no deseado. Para mí, al menos, no hay suficiente tiempo en el día para responder a cada uno.

Por otro lado, algunos sondeadores son más persistentes. Puede decidir agregar un firewall reglas para bloquearlos por completo, o posiblemente incluso todo su espacio de direcciones IP.

Los escaneos de un subconjunto de los puertos que se sabe que son posibles agujeros de seguridad suelen ser el precursor de un ataque si se encuentra un puerto abierto. Los escaneos más inclusivos suelen ser parte de un escaneo más amplio de aperturas en un dominio o subred. Las herramientas de piratería actuales prueban un subconjunto de estos puertos uno tras otro.

Ocasionalmente, verá intentos de piratería serios. Este es sin duda un momento para tomar acción. Escríbalos. Reportalos. Vuelva a comprobar su seguridad. Observa lo que están haciendo. Bloquearlos. Bloquear su bloque de dirección IP.

Algunos administradores de sistemas se toman en serio cada suceso porque, incluso si su máquina es segura, es posible que las máquinas de otras personas no lo sean. Es posible que la siguiente persona ni siquiera tenga la capacidad de saber que está siendo investigada. Informar sobre las investigaciones es socialmente responsable, por el bien de todos.

¿Cómo debe responder a los escaneos de puertos? Si escribe a estas personas, a su administrador de correo, a su centro de operaciones de red (NOC) del proveedor de servicios de enlace ascendente o al coordinador del bloque de direcciones de red, intente ser cortés. Dales el beneficio de la duda. Las reacciones exageradas están fuera de lugar la mayoría de las veces. Lo que podría parecerle un intento serio de piratería informática es a menudo un niño curioso que juega con un nuevo programa. Una palabra cortés al abusador, raíz o administrador de correos a veces puede resolver el problema. Es necesario educar a más personas sobre Netiqueta que cancelar sus cuentas de red. Y podrían ser inocentes de cualquier cosa. Con la misma frecuencia, el sistema de la persona está comprometido y esa persona no tiene idea de lo que está pasando y estará agradecida por la información.

Sin embargo, las sondas no son el único tráfico hostil que verá. Aunque las sondas son inofensivas en sí mismos, los ataques DoS no lo son.

Ataques de denegación de servicio

Los ataques DoS se basan en la idea de inundar su sistema con paquetes para interrumpir o degradar gravemente su conexión a Internet, bloqueando los servidores locales hasta el punto de que las solicitudes legítimas no pueden cumplirse o, en el peor de los casos, colapsando su sistema por completo. Los dos resultados más comunes son mantener el sistema demasiado ocupado para hacer algo útil y ocupar recursos críticos del sistema.

No puede protegerse completamente contra los ataques DoS. Pueden tomar tantas formas diferentes como lo permita la imaginación del atacante. Cualquier cosa que resulte en una respuesta de su sistema, cualquier cosa que resulte en que su sistema asigne recursos (incluido el registro del ataque), cualquier cosa que induzca a un sitio remoto a dejar de comunicarse con usted, todo puede usarse en un ataque DoS.

Más sobre ataques de denegación de servicio

Para obtener más información sobre los ataques DoS, consulte el documento "Denegación de servicio" disponible en <http://www.cert.org>.

Estos ataques generalmente involucran uno de varios patrones clásicos, incluida la inundación TCP SYN, inundación de ping, inundación de UDP, bombas de fragmentación, desbordamiento de búfer y bombas de redirección de enrutamiento ICMP.

Inundación TCP SYN

Un ataque de inundación TCP SYN consume los recursos de su sistema hasta que ya no son posibles más conexiones TCP entrantes (vea la Figura 2.8). El ataque hace uso del protocolo básico de protocolo de enlace de tres vías TCP durante el establecimiento de la conexión, junto con la suplantación de identidad de la dirección de origen IP.

El atacante falsifica su dirección de origen como una dirección privada e inicia una conexión a uno de sus servicios basados en TCP. Pareciendo ser un cliente que intenta abrir una conexión TCP, el atacante le envía un mensaje SYN generado artificialmente. Su máquina responde enviando un reconocimiento, un SYN-ACK. Sin embargo, en este caso, el

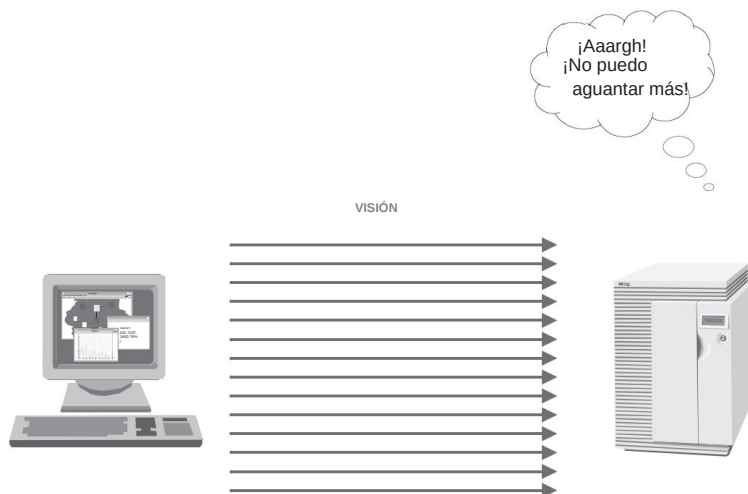


Figura 2.8 Una inundación TCP SYN

la dirección a la que está respondiendo no es la dirección del atacante. De hecho, debido a que la dirección es privada, no hay nadie para responder. El host falsificado no devolverá un mensaje RST para interrumpir la conexión a medio abrir.

La etapa final del establecimiento de la conexión TCP, recibir un ACK en respuesta, nunca sucederá. En consecuencia, se consumen recursos de conexión de red finitos. La conexión permanece en un estado semiabierto hasta que se agota el tiempo de espera del intento de conexión. El atacante inunda su puerto con solicitud de conexión tras solicitud de conexión, más rápido que los tiempos de espera de TCP que liberan los recursos. Si esto continúa, todos los recursos estarán en uso y no se podrán aceptar más solicitudes de conexión entrantes. Esto se aplica no solo al servicio que se está probando, sino también a todas las conexiones nuevas.

Varias ayudas están disponibles para los usuarios de Linux. El primero es el filtrado de direcciones de origen, descrito anteriormente. Esto filtra las direcciones de origen falsificadas más utilizadas, pero no hay garantía de que la dirección falsificada se encuentre dentro de las categorías que puede anticipar y filtrar.

El segundo es habilitar el módulo de cookies SYN de su núcleo, un retardador específico para el agotamiento de recursos causado por la inundación SYN. Cuando la cola de conexión comienza a llenarse, el sistema comienza a responder a las solicitudes SYN con cookies SYN en lugar de SYN-ACK, y libera la ranura de la cola. Por lo tanto, la cola nunca se llena por completo. La cookie tiene un tiempo de espera breve; el cliente debe responder dentro de un período breve antes de que el host de servicio responda con el SYN-ACK esperado. La cookie es un número de secuencia que se genera en función del número de secuencia original en el SYN, las direcciones y puertos de origen y destino, y un valor secreto. Si la respuesta a la cookie coincide con el resultado del algoritmo hash, el servidor está razonablemente seguro de que el SYN es válido.

Dependiendo de la versión en particular, es posible que deba o no habilitar el SYN protección de cookies dentro del kernel usando el comando `echo 1 > /proc/sys/net/ipv4/tcp_syncookies`. Algunas distribuciones y versiones del kernel requieren que configure explícitamente la opción en el kernel usando `make config`, `make menuconfig` o `make xconfig` y luego vuelva a compilar e instalar el nuevo kernel.

Inundación SYN y falsificación de IP

Para obtener más información sobre la inundación de SYN y la suplantación de IP, consulte el Aviso CA-96.21 de CERT, "Ataques de suplantación de IP e inundación de SYN de TCP", en <http://www.cert.org>.

inundación de ping

Cualquier mensaje que obtenga una respuesta de su máquina puede usarse para degradar su conexión de red al obligar al sistema a dedicar la mayor parte de su tiempo a responder. El mensaje de solicitud de eco ICMP enviado por ping es un culpable común. Un ataque llamado *Smurf*, y sus variantes, obliga a un sistema a gastar sus recursos procesando respuestas de eco. Un método para lograr esto es falsificar la dirección de origen de la víctima y transmitir una solicitud de eco a toda una red de hosts. Un solo mensaje de solicitud falsificado puede dar lugar a que se envíen cientos o miles de respuestas a la víctima. Otra forma de lograr un resultado similar es instalar troyanos en hosts comprometidos a través de Internet y cronometrarlos.

para enviar solicitudes de eco al mismo host simultáneamente. Finalmente, una inundación de ping simple en la que el atacante envía más solicitudes de eco e inunda la conexión de datos es otro método para un DoS, aunque se está volviendo menos común. Una inundación de ping típica se muestra en la Figura 2.9.

Ping de la muerte

Un exploit más antiguo llamado *Ping of Death* involucraba el envío de paquetes de ping muy grandes. Como resultado, los sistemas vulnerables podrían bloquearse. Linux no es vulnerable a este exploit, como tampoco lo son muchos otros sistemas operativos UNIX actuales. Si su firewall está protegiendo sistemas antiguos o computadoras personales, esos sistemas podrían ser vulnerables.

El exploit Ping of Death da una idea de cómo el hacker creativo puede utilizar los protocolos más simples y las interacciones de mensajes. No todos los ataques son intentos de entrar en su computadora. Algunos son simplemente destructivos. En este caso, el objetivo es bloquear la máquina.

(Los bloqueos del sistema también pueden ser un indicador de que necesita verificar su sistema en busca de programas troyanos instalados. Es posible que lo hayan engañado para que cargue un programa troyano, pero el programa en sí puede requerir un reinicio del sistema para activarse).

ping es una herramienta de red básica muy útil. Es posible que no desee desactivar el ping por completo. En el entorno de Internet actual, la gente conservadora recomienda deshabilitar el ping entrante o al menos limitar severamente de quién acepta solicitudes de eco. Debido al historial de participación de ping en ataques DoS, muchos sitios ya no responden a ping externo . solicitudes de cualquier fuente excepto las seleccionadas. Esto siempre ha parecido ser una reacción exagerada a la amenaza relativamente pequeña de un DoS basado en ICMP en comparación con las amenazas más ubicuas y peligrosas contra las aplicaciones y otros protocolos dentro de la pila.

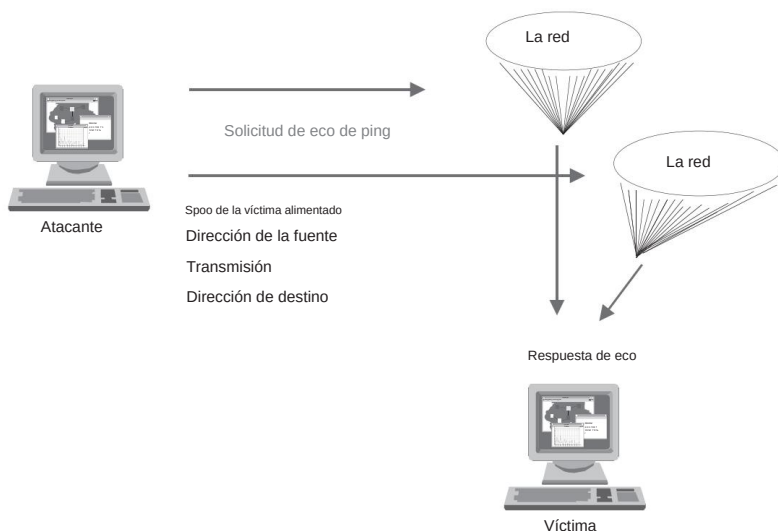


Figura 2.9 Una inundación de ping

Sin embargo, eliminar las solicitudes de ping no es una solución para el host de la víctima. Independientemente de cómo reaccione el destinatario de la inundación a los paquetes, el sistema (o la red) aún puede verse abrumado en el proceso de inspección y eliminación de una avalancha de solicitudes.

Inundaciones UDP

El protocolo UDP es especialmente útil como herramienta DoS. A diferencia de TCP, UDP no tiene estado. Los mecanismos de control de flujo no están incluidos. No hay indicadores de estado de conexión. No se utilizan números de secuencia de datagramas. No se mantiene información sobre qué paquete se espera a continuación. No siempre hay una forma de diferenciar el tráfico del cliente del tráfico del servidor en función de los números de puerto. Sin estado, no hay forma de distinguir una respuesta entrante esperada de un paquete no solicitado que llega inesperadamente. Es relativamente fácil mantener un sistema tan ocupado respondiendo a las sondas UDP entrantes que no quede ancho de banda para el tráfico de red legítimo.

Debido a que los servicios UDP son susceptibles a este tipo de ataques (a diferencia de los servicios TCP orientados a la conexión), muchos sitios deshabilitan todos los puertos UDP que no son absolutamente necesarios. Como se mencionó anteriormente, casi todos los servicios comunes de Internet están basados en TCP. El cortafuegos que construiremos en el Capítulo 5, "Construir e instalar un cortafuegos independiente", limita cuidadosamente el tráfico UDP solo a aquellos hosts remotos que brindan los servicios UDP necesarios.

El clásico ataque de inundación UDP involucra a dos máquinas víctimas o funciona de la misma manera que lo hace la inundación de ping de Smurf (consulte la Figura 2.10). Un solo paquete falsificado desde el puerto de eco UDP del atacante, dirigido al puerto de carga UDP de un host, puede generar un bucle infinito de tráfico de red. Los servicios echo y chargen son servicios de prueba de red. chargen genera una cadena ASCII. echo devuelve los datos enviados al puerto.

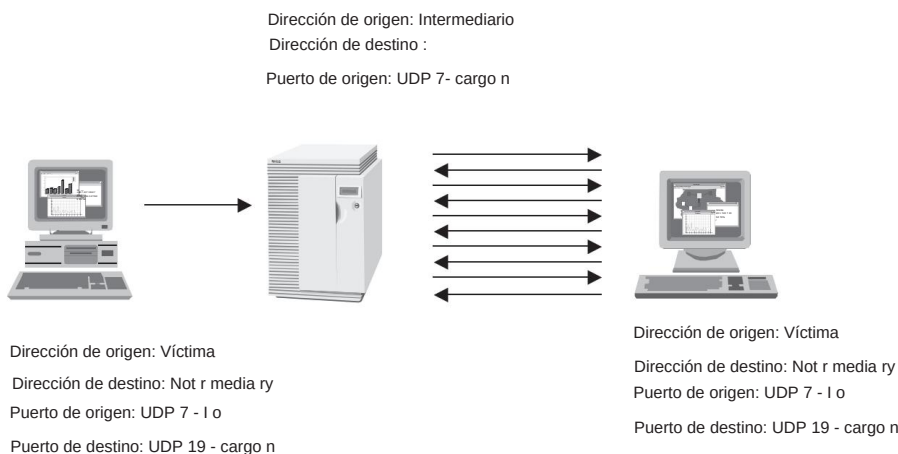


Figura 2.10 Una inundación UDP

Ataques de denegación de servicio de puerto UDP

Para obtener una descripción más completa de un exploit DoS que usa estos servicios UDP, consulte el Aviso CERT CA-96.01, "Ataque de denegación de servicio de puerto UDP", en <http://www.cert.org>.

Bombas de fragmentación

Diferentes tecnologías de red subyacentes (como Ethernet, modo de transferencia asíncrono [ATM] y token ring) definen diferentes límites en el tamaño de la trama de capa 2. A medida que un paquete pasa de un enrutador al siguiente a lo largo de la ruta desde la máquina de origen hasta la máquina de destino, es posible que los enrutadores de puerta de enlace de red necesiten cortar el paquete en partes más pequeñas, llamadas *fragmentos*, antes de pasarlos a la siguiente red. En una fragmentación legítima, el primer fragmento contiene los números de puerto de origen y destino habituales contenidos en el encabezado de transporte UDP o TCP. Los siguientes fragmentos no.

Por ejemplo, aunque la longitud teórica máxima del paquete es de 65 535 bytes, el tamaño máximo de trama de Ethernet (Unidad máxima de transmisión o MTU) es de 1500 bytes.

Cuando se fragmenta un paquete, los enrutadores intermedios no vuelven a ensamblar el paquete. los los paquetes se vuelven a ensamblar en el host de destino o por su enrutador adyacente.

Debido a que la fragmentación intermedia es, en última instancia, más costosa que el envío de paquetes no fragmentados más pequeños, los sistemas actuales a menudo realizan el descubrimiento de MTU con el host de destino al comienzo de una conexión. Esto se hace enviando un paquete con la opción Don't Fragment establecida en el campo de opciones de encabezado de IP (el único uso actual generalmente legítimo del campo de opciones de IP). Si un enrutador intermedio debe fragmentar el paquete, descarta el paquete y devuelve un error ICMP 3, se requiere fragmentación.

Un tipo de ataque de fragmentación implica la construcción artificial de paquetes muy pequeños. Los paquetes de un byte bloquean algunos sistemas operativos. Los sistemas operativos actuales generalmente prueban esta condición.

Otro uso de los fragmentos pequeños es construir el fragmento inicial para que el UDP o Los puertos TCP de origen y destino están contenidos en el segundo fragmento. (Los tamaños de MTU de todas las redes son lo suficientemente grandes como para transportar una IP estándar de 40 bytes y un encabezado de transporte). Los cortafuegos de filtrado de paquetes a menudo permiten el paso de estos fragmentos porque la información que filtran no está presente. Esta forma de ataque es útil para pasar paquetes a través del cortafuegos que de otro modo no estarían permitidos.

El exploit Ping of Death mencionado anteriormente es un ejemplo del uso de la fragmentación para transportar un mensaje ICMP ilegalmente grande. Cuando se reconstruye la solicitud de ping, el tamaño total del paquete supera los 65 535 bytes, lo que provoca que algunos sistemas se bloqueen.

Un ejemplo clásico de explotación de fragmentación es el ataque Teardrop. El método se puede utilizar para eludir un cortafuegos o bloquear un sistema. El primer fragmento se construye para ir a un servicio permitido. (Muchos cortafuegos no inspeccionan los fragmentos después del primer paquete). Si está permitido, el host de destino pasará y volverá a ensamblar los fragmentos posteriores. Si se descarta el primer paquete, los paquetes subsiguientes pasarán a través del firewall, pero el host final no tendrá nada que reconstruir y eventualmente descartará el paquete parcial.

Los campos de compensación de datos en los fragmentos posteriores se pueden modificar para sobrescribir el puerto información en el primer fragmento para acceder a un servicio no permitido. El desplazamiento también se puede modificar para que los desplazamientos utilizados en el reensamblaje de paquetes resulten ser números negativos. Debido a que las rutinas de copia de bytes del kernel generalmente usan números sin signo, el valor negativo se trata como un número positivo muy grande; la copia resultante destruye la memoria del núcleo y el sistema falla.

Las máquinas de firewall y las máquinas que realizan NAT para otros hosts locales deben configurarse para volver a ensamblar los paquetes antes de entregarlos al destino local. Algunas de las funciones de iptables requieren que el sistema vuelva a ensamblar los paquetes antes de reenviarlos al host de destino, y el reensamblaje se realiza automáticamente.

Desbordamientos de búfer

Los exploits de desbordamiento de búfer no pueden ser protegidos por un firewall de filtrado. Las hazañas se dividen en dos categorías principales. El primero es simplemente hacer que un sistema o servidor se bloquee al sobrescribir su espacio de datos o pila de tiempo de ejecución. El segundo requiere experiencia técnica y conocimiento del hardware y el software del sistema o la versión del servidor que se ataca. El propósito del desbordamiento es sobrescribir la pila de tiempo de ejecución del programa para que la pila de retorno de llamada contenga un programa y un salto a él. Este programa generalmente inicia un shell con root privilegio.

Muchas de las vulnerabilidades actuales en los servidores son el resultado de desbordamientos de búfer. Es importante instalar y mantener actualizados todos los parches y revisiones de software más recientes.

Bombas de redirección ICMP

El mensaje de redirección ICMP Tipo 5 le dice al sistema de destino que cambie sus tablas de enrutamiento en memoria a favor de una ruta más corta. Los enrutadores envían redireccionamientos a sus hosts adyacentes. Su intención es informar al host que hay una ruta más corta disponible (es decir, el host y el nuevo enrutador están en la misma red, y el nuevo enrutador es el enrutador al que el original enrutaría el paquete como su próximo salto).

Los redireccionamientos llegan casi a diario. Rara vez se originan en el enrutador adyacente. Para sitios residenciales o comerciales conectados a un ISP, es muy poco probable que su enrutador adyacente genere un mensaje de redirección.

Si su host usa enrutamiento estático y respeta los mensajes de redirección, es posible que alguien engañe a su sistema haciéndole creer que una máquina remota es una de sus máquinas locales o una de las máquinas de su ISP, o incluso que engañe a su sistema para que reenvíe todo el tráfico a alguna otro host remoto.

Ataques de denegación de servicio y otros recursos del sistema

La conectividad de la red no es la única preocupación en los ataques DoS. Estos son algunos ejemplos de otras áreas a tener en cuenta al configurar su sistema:

- Su sistema de archivos puede desbordarse si su sistema se ve obligado a escribir una gran cantidad de mensajes en los registros de errores, o si su sistema está inundado con muchas copias de mensajes de correo electrónico de gran tamaño. Es posible que desee configurar límites de recursos y configurar una partición separada para sistemas de archivos que crecen o cambian rápidamente.

Exploits de denegación de servicio de correo electrónico

Para obtener una descripción de un exploit DoS usando el correo electrónico, consulte "Bombeo de correo electrónico y spam" en <http://www.cert.org>.

La memoria del sistema, las ranuras de la tabla de procesos, los ciclos de CPU y otros recursos pueden agotado por invocaciones repetidas y rápidas de servicios de red. Puede hacer poco al respecto aparte de establecer límites configurables para cada servicio individual, habilitar las cookies SYN y denegar en lugar de rechazar los paquetes enviados a puertos de servicio no admitidos.

Paquetes enrutados de origen

Los paquetes enrutados en origen emplean una opción de IP que rara vez se usa y que permite que el originador defina la ruta tomada entre dos máquinas, en lugar de dejar que los enrutadores intermedios determinen la ruta. Al igual que con los redireccionamientos ICMP, esta característica puede permitir que alguien engañe a su sistema para que piense que está hablando con una máquina local, una máquina ISP o algún otro host confiable, o para crear el flujo de paquetes necesario para un intermediario. ataque.

El enrutamiento de origen tiene pocos usos legítimos en las redes actuales. Algunos enrutadores ignoran la opción. Algunos cortafuegos descartan los paquetes que contienen la opción.

Filtrado de paquetes salientes

Si su entorno representa un entorno de confianza, es posible que el filtrado de paquetes salientes no parezca tan importante como el filtrado de paquetes entrantes. Su sistema no responderá a los mensajes entrantes que el cortafuegos no atraviese. Los sitios residenciales a menudo adoptan este enfoque. Sin embargo, incluso para sitios residenciales, el filtrado simétrico es importante, en particular si el firewall protege las máquinas con Microsoft Windows. Para los sitios comerciales, el filtrado saliente es indiscutiblemente importante.

Si su firewall protege una LAN de sistemas Microsoft Windows, el control del tráfico saliente se vuelve mucho más importante. Históricamente, las máquinas Windows comprometidas se han utilizado (y continúan utilizándose) en ataques DoS coordinados y otros ataques salientes.

Por esta razón especialmente, es importante filtrar lo que sale de su red.

El filtrado de mensajes salientes también le permite ejecutar servicios de LAN sin filtrarse en Internet, donde estos paquetes no pertenecen. No se trata solo de prohibir el acceso externo a los servicios LAN locales. También es una cuestión de no transmitir información del sistema local a Internet. Ejemplos de esto serían si estuviera ejecutando un DHCPD local, NTP, SMB u otro servidor para uso interno. Otros servicios desagradables pueden ser la difusión de mensajes de pared o syslogd .

Una fuente relacionada es parte del software de la computadora personal, que a veces ignora los protocolos de puerto de servicio de Internet y las asignaciones reservadas. Este es el equivalente en computadora personal de ejecutar un programa diseñado para uso LAN en una máquina conectada a Internet.

Una razón final es simplemente mantener local el tráfico local que no está destinado a salir de la LAN pero que posiblemente podría hacerlo. Mantener el tráfico local local es una buena idea desde el punto de vista de la seguridad, pero también como un medio para conservar el ancho de banda.

Filtrado de direcciones de origen local

Es fácil filtrar los paquetes salientes en función de la dirección de origen. Para un sitio pequeño o una sola computadora conectada a Internet, la dirección de origen siempre es la dirección IP de su computadora durante el funcionamiento normal. No hay motivo para permitir que un paquete saliente tenga otra dirección de origen, y el cortafuegos debería imponer esto.

Para las personas cuya dirección IP es asignada dinámicamente por su ISP, existe una breve excepción durante la asignación de direcciones. Esta excepción es específica de DHCP y es el único caso en el que un host transmite mensajes utilizando 0.0.0.0 como dirección de origen.

Para las personas con una LAN cuya máquina de firewall tiene una dirección IP asignada dinámicamente, Es obligatorio limitar los paquetes salientes para que contengan la dirección de origen de la dirección IP de la máquina del cortafuegos. Lo protege de varios errores de configuración bastante comunes que aparecen como casos de suplantación de direcciones de origen o direcciones de origen ilegales para hosts remotos.

Si sus usuarios o su software no son 100 % confiables, es importante asegurarse de que el tráfico local contenga solo direcciones locales legítimas para evitar participar en ataques DoS mediante la suplantación de direcciones de origen.

Este último punto es especialmente importante. RFC 2827, "Filtrado de ingreso de red: Derrotar los ataques de denegación de servicio que emplean la suplantación de dirección IP de origen" (actualizado por RFC 3704, "Filtrado de ingreso para redes de múltiples terminales"), es un documento actual de "mejores prácticas" que habla exactamente de este punto. Idealmente, cada enrutador debería filtrar las direcciones de origen obviamente ilegales y asegurarse de que el tráfico que sale de la red local contenga solo direcciones de origen enrutables que pertenecen a esa red.

Filtrado de direcciones de destino remoto

Al igual que con los paquetes entrantes, es posible que desee permitir que ciertos tipos de paquetes salientes se dirijan solo a redes remotas específicas o máquinas individuales. En estos casos, las reglas del cortafuegos definirán direcciones IP específicas o un rango limitado de direcciones IP de destino a las que se permitirán estos paquetes.

La primera clase de paquetes salientes para filtrar por dirección de destino son los paquetes destinados a servidores remotos con los que se ha puesto en contacto. Aunque se puede esperar que algunos paquetes, como los que van a servidores web o FTP, estén destinados a cualquier parte de Internet, otros servicios remotos se ofrecerán legítimamente solo desde su ISP o hosts de confianza especialmente elegidos.

Algunos ejemplos de servicios que probablemente se ofrecen únicamente a través de su ISP son los servicios de correo como SMTP o POP3, los servicios de DNS, la asignación dinámica de direcciones IP de DHCP y el servicio de noticias de Usenet.

La segunda clase de paquetes salientes para filtrar por dirección de destino son los paquetes destinados a clientes remotos que acceden a un servicio ofrecido desde su sitio. Nuevamente, aunque algunas conexiones de servicios salientes, como las respuestas de su servidor web local, pueden ser

se espera que vaya a cualquier parte, otros servicios locales se ofrecerán solo a unos pocos sitios remotos de confianza o amigos. Ejemplos de servicios locales restringidos pueden ser telnet, SSH, servicios basados en Samba y servicios RPC a los que se accede a través de portmap. Las reglas del cortafuegos no solo denegarán las conexiones entrantes generales a estos servicios, sino que las reglas tampoco permitirán respuestas salientes de estos servicios a cualquiera.

Filtrado de puerto de origen local

Definir explícitamente qué puertos de servicio en su red se pueden usar para conexiones salientes tiene dos propósitos: uno para sus programas de cliente y otro para sus programas de servidor. Especificar los puertos de origen permitidos para sus conexiones salientes ayuda a garantizar que sus programas se comporten correctamente y protege a otras personas de cualquier tráfico de red local que no pertenezca a Internet.

Las conexiones salientes de sus clientes locales casi siempre se originarán en un puerto de origen sin privilegios. Limitar a sus clientes a los puertos sin privilegios en las reglas del firewall ayuda a proteger a otras personas de posibles errores de su parte al garantizar que los programas de sus clientes se comporten como se espera.

Los paquetes salientes de los programas de su servidor local siempre se originarán en su puerto de servicio asignado y serán en respuesta a una solicitud recibida. Limitar sus servidores a sus puertos asignados a nivel de firewall garantiza que los programas de su servidor funcionen correctamente a nivel de protocolo. Más importante aún, ayuda a proteger cualquier servicio de red local privado que pueda estar ejecutando desde el exterior. También ayuda a proteger los sitios remotos para que no se vean afectados por el tráfico de red que debería permanecer confinado a sus sistemas locales.

Filtrado de puertos de destino remoto

Sus programas de clientes locales están diseñados para conectarse a servidores de red que ofrecen sus servicios desde sus puertos de servicio asignados. Desde esta perspectiva, limitar sus clientes locales para que se conecten solo al puerto de servicio de su servidor asociado garantiza la corrección del protocolo. Limitar las conexiones de sus clientes a puertos de destino específicos también sirve para otros propósitos. En primer lugar, ayuda a proteger contra los programas de clientes de redes privadas locales que intentan acceder a servidores en Internet sin darse cuenta. En segundo lugar, hace mucho para rechazar errores salientes, escaneos de puertos y otras travesuras que potencialmente se originan en su sitio.

Los programas de su servidor local casi siempre participarán en conexiones que se originen en puertos sin privilegios. Las reglas del cortafuegos limitan el tráfico saliente de sus servidores solo a los puertos de destino sin privilegios.

Filtrado de estado de conexión TCP saliente

Las reglas de aceptación de paquetes TCP salientes pueden hacer uso de los indicadores de estado de conexión asociados con las conexiones TCP, tal como lo hacen las reglas entrantes. Todas las conexiones TCP se adhieren al mismo conjunto de estados de conexión, que difiere entre el cliente y el servidor.

Los paquetes TCP salientes de los clientes locales tendrán el indicador SYN establecido en el primer paquete enviado como parte del protocolo de enlace de establecimiento de conexión de tres vías. La solicitud de conexión inicial tendrá el indicador SYN establecido, pero no el indicador ACK. Las reglas de firewall de su cliente local permitirán paquetes salientes con el indicador SYN o ACK establecido.

Los paquetes salientes de los servidores locales siempre serán respuestas a una solicitud de conexión inicial iniciada desde un programa de cliente remoto. Cada paquete enviado desde sus servidores tendrá el indicador ACK establecido. Las reglas de firewall de su servidor local requerirán que todos los paquetes salientes de sus servidores tengan el indicador ACK establecido.

Servicios de red privados versus públicos

Una de las formas más fáciles de permitir inadvertidamente intrusiones no invitadas es permitir el acceso externo a los servicios locales que están diseñados solo para el uso de LAN. Algunos servicios, si se ofrecen localmente, nunca deben cruzar la frontera entre su LAN e Internet más allá.

Algunos de estos servicios molestan a sus vecinos, algunos brindan información que es mejor que se guarde para usted y algunos representan agujeros de seguridad evidentes si están disponibles fuera de su LAN.

Algunos de los primeros servicios de red, en particular los comandos basados en r-*, se diseñaron para compartir localmente y facilitar el acceso a través de varias máquinas de laboratorio en un entorno confiable. Algunos de los servicios posteriores estaban destinados al acceso a Internet, pero se diseñaron en una época en la que Internet era básicamente una comunidad extendida de académicos e investigadores. Internet era un lugar relativamente abierto y seguro. A medida que Internet creció hasta convertirse en una red global que incluía el acceso del público en general, se convirtió en un entorno completamente desconfiado.

Muchos servicios de red de Linux están diseñados para proporcionar información local sobre cuentas de usuario en el sistema, qué programas se están ejecutando y qué recursos están en uso, estado del sistema, estado de la red e información similar de otras máquinas conectadas a través de la red. No todos estos servicios de información representan agujeros de seguridad en sí mismos. No es que alguien pueda usarlos directamente para obtener acceso no autorizado a su sistema. Es que brindan información sobre su sistema y cuentas de usuario que pueden ser útiles para alguien que busca vulnerabilidades conocidas. También pueden proporcionar información como nombres de usuario, direcciones, números de teléfono, etc., que no desea que esté disponible para todos los que preguntan.

Algunos de los servicios de red más peligrosos están diseñados para proporcionar acceso LAN a sistemas de archivos y dispositivos compartidos, como una impresora en red o una máquina de fax.

Algunos servicios son difíciles de configurar correctamente y otros son difíciles de configurar de forma segura. Libros enteros están dedicados a configurar algunos de los servicios Linux más complicados. La configuración de servicios específicos está más allá del alcance de este libro.

Algunos servicios simplemente no tienen sentido en un hogar o en una oficina pequeña. Algunos están destinados a administrar grandes redes, brindar servicios de enrutamiento de Internet, proporcionar servicios de información de grandes bases de datos, admitir cifrado y autenticación bidireccionales, etc.

Protección de servicios locales no seguros

La forma más fácil de protegerse es no ofrecer el servicio. Pero, ¿qué sucede si necesita uno de estos servicios localmente?

No todos los servicios se pueden proteger adecuadamente en el nivel de filtrado de paquetes. El software para compartir archivos, los servicios de mensajería instantánea y los servicios RPC basados en UDP son notoriamente difíciles de proteger a nivel de filtrado de paquetes.

Una forma de proteger su computadora es no alojar servicios de red en la máquina de firewall que no pretenda para uso público. Si el servicio no está disponible, no hay nada a lo que pueda conectarse un cliente remoto. Que los cortafuegos sean cortafuegos.

Un firewall de filtrado de paquetes no ofrece una seguridad completa. Algunos programas requieren medidas de seguridad de mayor nivel que las que se pueden proporcionar en el nivel de filtrado de paquetes. Algunos programas son demasiado problemáticos para correr el riesgo de ejecutarse en una máquina con firewall, incluso en un host residencial menos seguro.

Los sitios pequeños, como los que se encuentran en el hogar, a menudo no tendrán un suministro de computadoras disponibles para hacer cumplir las políticas de seguridad de acceso mediante la ejecución de servicios privados en otras máquinas. Se deben hacer compromisos, particularmente para los servicios requeridos que son proporcionados únicamente por Linux. No obstante, los sitios pequeños con una LAN no deberían ejecutar archivos compartidos u otros servicios de LAN privados en el firewall, como Samba. La máquina no debe tener cuentas de usuario innecesarias. El software del sistema que no sea necesario debe eliminarse del sistema. La máquina no debe tener otra función que la de una puerta de enlace de seguridad.

Selección de servicios para ejecutar

Cuando todo está dicho y hecho, solo usted puede decidir qué servicios necesita o desea. El primer paso para proteger su sistema es decidir qué servicios y demonios desea ejecutar en la máquina del cortafuegos, así como detrás del cortafuegos en la LAN privada. Cada servicio tiene sus propias consideraciones de seguridad. Cuando se trata de seleccionar servicios para ejecutar bajo Linux o cualquier otro sistema operativo, la regla general es ejecutar solo los servicios de red que necesita y comprende. Es importante comprender un servicio de red, qué hace y para quién está destinado, antes de ejecutarlo, especialmente en una máquina conectada directamente a Internet.

Resumen

Entre este y el capítulo anterior, se han establecido los conceptos básicos de redes y firewalls. El siguiente capítulo profundiza en iptables .

iptables: el legado de Linux

Programa de administración de cortafuegos

El Capítulo 2, "Conceptos de filtrado de paquetes", cubrió las ideas y conceptos básicos detrás de un firewall de filtrado de paquetes. Cada lista integrada de reglas, a veces denominada *cadena de reglas*, tiene su propia política predeterminada. Cada regla puede aplicarse no solo a una cadena individual, sino también a una interfaz de red específica, un tipo de protocolo de mensaje (como TCP, UDP o ICMP) y un puerto de servicio o un número de tipo de mensaje ICMP. Las reglas de aceptación, denegación y rechazo individuales se definen para la *cadena de ENTRADA* y la *cadena de SALIDA*, así como para la *cadena de ENVÍO*, sobre la cual aprenderá al final de este capítulo y en el Capítulo 7, "Reenvío de paquetes".

Este capítulo cubre el programa de administración del cortafuegos iptables que se usa para construir un cortafuegos Netfilter. El programa de administración de iptables es parte del código de firewall heredado en el kernel de Linux. A partir de la versión 3.13 del kernel, se agregó un nuevo mecanismo de filtrado llamado nftables. El siguiente capítulo analiza nftables. Este capítulo se centra en las iptables heredadas programa de administración porque todavía se usa ampliamente en los sistemas Linux. Para aquellos de ustedes que están familiarizados o acostumbrados a los programas más antiguos ipfwadm e ipchains utilizados con la tecnología IPFW, iptables se verá muy similar a esos programas. Sin embargo, es mucho más rico en funciones y flexible, y es muy diferente en niveles sutiles.

De hecho, existe una diferencia entre iptables y Netfilter, aunque a menudo escuchará que los términos se usan indistintamente. Netfilter es el código del programa del espacio del kernel de Linux para implementar un firewall dentro del kernel de Linux, ya sea compilado directamente en el kernel o incluido como un conjunto de módulos. Por otro lado, iptables es el programa de usuario utilizado para la administración del cortafuegos de Netfilter. A lo largo de este texto, me referiré a iptables que incluye tanto Netfilter como iptables, a menos que se indique lo contrario.

Diferencias entre IPFW y Netfilter

Mecanismos de cortafuegos

Debido a que iptables es tan diferente de las ipchains anteriores, este libro no intentará cubrir la implementación anterior.

La siguiente sección está escrita para el lector que está familiarizado con ipchains o lo está usando actualmente. Si iptables es su primera introducción al firewall de Linux, puede pasar directamente a la sección "Netfilter Packet Traversal".

Si está convirtiendo desde ipchains, notará varias diferencias menores en la sintaxis de iptables, sobre todo que las interfaces de red de entrada y salida se identifican por separado. Otras diferencias incluyen:

• iptables está altamente modularizado, y los módulos individuales ocasionalmente deben ser cargado explícitamente.

• El registro es un objetivo de regla en lugar de una opción de comando.

• Se puede mantener el seguimiento del estado de la conexión. La dirección y la traducción del puerto son logi funciones separadas del filtrado de paquetes.

• Se implementa la traducción completa de direcciones de origen y destino.

• Enmascaramiento es un término que se utiliza para referirse a una forma especializada de dirección de origen NAT. El reenvío de puertos y la traducción de direcciones de destino se admiten directamente sin necesidad de soporte de software de terceros, como ipmasqadm.

Enmascaramiento en versiones anteriores de Linux

Para aquellos de ustedes que son nuevos en Linux, la traducción de direcciones de red (NAT) está completamente implementada en iptables. Antes de esto, NAT se denominaba enmascaramiento en Linux. Una implementación simple y parcial de Source Address Translation, el enmascaramiento fue utilizado por los propietarios de sitios que tenían una sola dirección IP pública y que querían que otros hosts en su red privada pudieran acceder a Internet. Los paquetes salientes de estos hosts internos tenían su dirección de origen enmascarada con la dirección IP pública enrutable.

La diferencia más importante está en cómo se enrutan o reenvían los paquetes a través del sistema operativo, lo que genera diferencias sutiles en cómo se construye el conjunto de reglas del firewall.

Para los usuarios de ipchains, comprender las diferencias en el cruce de paquetes que se analizan en las próximas dos secciones es muy importante. iptables e ipchains se parecen mucho en la superficie, pero son muy diferentes en la práctica. Es muy fácil escribir reglas de iptables sintácticamente correctas que tengan un efecto diferente al que tendría una regla similar en ipchains. Puede ser confuso. Si ya conoces ipchains, debes tener en cuenta las diferencias.

Paquete transversal IPFW

Bajo IPFW (ipfwadm e ipchains), se utilizaron tres cadenas de filtros integradas. Todos los paquetes que llegaban a una interfaz se filtraban contra la cadena INPUT. Si se aceptaba el paquete, se pasaba al módulo de enrutamiento. La función de enrutamiento determinaba si el paquete debía entregarse localmente o reenviarse a otra interfaz de salida. El flujo de paquetes IPFW se muestra en la Figura 3.1.

Si se reenvía, el paquete se filtra por segunda vez contra la cadena FORWARD. Si el paquete fue aceptado, se pasó a la cadena OUTPUT.

Tanto los paquetes salientes generados localmente como los paquetes reenviados se pasaron al Cadenas de SALIDA. Si se aceptaba el paquete, se enviaba por la interfaz.

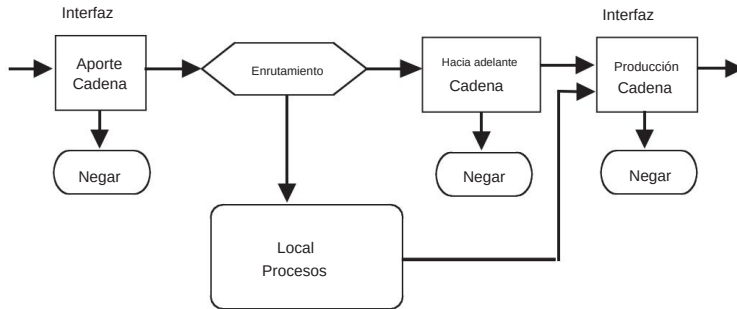


Figura 3.1 Travesía de paquetes IPFW

(Figura basada en "Linux IPCHAINS-HOWTO", de Rusty Russel, v1.0.8.)

Los paquetes locales (bucle invertido) recibidos y enviados pasan a través de dos filtros. Paquete reenviado Ets pasó a través de tres filtros.

La ruta de loopback involucró dos cadenas. Como se muestra en la Figura 3.2, cada paquete de loopback pasó a través del filtro de salida antes de "salir" de la interfaz de bucle invertido, donde luego se entregó a la interfaz de entrada del bucle invertido. Luego se aplicó el filtro de entrada.

Tenga en cuenta que la ruta de loopback demuestra por qué una sesión de X Window se bloquea cuando se inicia un script de firewall que no permite el tráfico de loopback o falla antes de hacerlo cuando se usa una política de denegación predeterminada.

En el caso de que los paquetes de respuesta fueran desenmascarados antes de reenviarlos a la LAN, se aplicaron los filtros de entrada. En lugar de pasar por la función de enrutamiento, el paquete se entregó directamente a la cadena de filtros de SALIDA . Por lo tanto, los paquetes entrantes desenmascarados se filtraron dos veces. Los paquetes enmascarados salientes se filtraron tres veces.

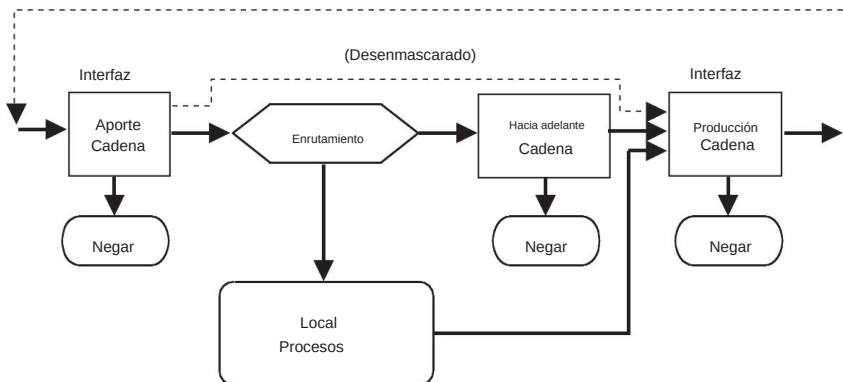


Figura 3.2 Loopback de IPFW y recorrido de paquetes enmascarados

(Figura basada en "Linux IPCHAINS-HOWTO", de Rusty Russel, v1.0.8.)

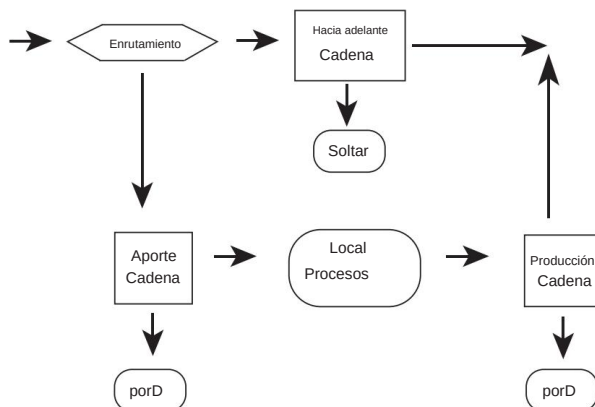


Figura 3.3 Travesía de paquetes de Netfilter

(Figura basada en “Linux 2.4 Packet Filtering HOWTO,” por Rusty Russel, v1.0.1.)

Paquete transversal de Netfilter

En Netfilter (iptables), se utilizan las cadenas de filtros INPUT, OUTPUT y FORWARD integradas.

Los paquetes entrantes pasan a través de la función de enrutamiento, que determina si entregar el paquete a la cadena INPUT del host local o a la cadena FORWARD. El flujo de paquetes de Netfilter se muestra en la Figura 3.3.

Si las reglas de la cadena INPUT aceptan un paquete con destino local, el paquete se entrega localmente.

Si las reglas de la cadena FORWARD aceptan un paquete con destino remoto, el paquete se envía a través de la interfaz apropiada.

Los paquetes salientes de los procesos locales se pasan a las reglas de la cadena de SALIDA. Si se acepta el paquete, se envía a través de la interfaz apropiada. Por lo tanto, cada paquete se filtra una vez (excepto los paquetes de bucle invertido, que se filtran dos veces).

Sintaxis básica de iptables

Los cortafuegos construidos con Netfilter se construyen a través del comando de administración de cortafuegos iptables. El comando iptables implementa las políticas de firewall que crea y administra el comportamiento del firewall. Los cortafuegos de Netfilter tienen tres tablas individuales: filter, nat y mangle. Dentro de estas tablas, los cortafuegos se construyen a través de cadenas, y cada enlace individual de la cadena es un comando de iptables individual.

Dentro de la tabla de filtro predeterminada, hay una cadena para la entrada o los datos que ingresan al firewall, una cadena para salida o datos que salen del firewall, una cadena para reenvío o datos que se envían a través del firewall, y otras cadenas que incluyen cadenas nombradas y configuradas por el usuario, comúnmente (y apropiadamente) llamadas *cadenas definidas por el usuario*. Las tablas nat y mangle

tienen cadenas especiales que se discutirán más adelante. Por ahora, es suficiente saber que la **tabla de filtro** es la tabla predeterminada para implementar un firewall básico, la **tabla nat** se usa para proporcionar NAT y funciones relacionadas, y la **tabla mangle** se usa cuando el firewall alterará el paquete.

Los comandos de iptables se emiten con una sintaxis muy específica. Muchas veces, el orden de las opciones dadas a iptables marca la diferencia entre un comando exitoso y un error de sintaxis. Los comandos emitidos a iptables fallan, por lo que un comando que permite ciertos paquetes que sigue a un comando que niega esos mismos paquetes hará que el firewall elimine los datos.

La sintaxis básica para un comando de iptables comienza con el propio comando de iptables , seguido de una o más opciones, una cadena, un conjunto de criterios de coincidencia y un objetivo o disposición. El diseño del comando depende en gran medida de la acción a realizar.

Considere esta sintaxis:

```
iptables <opción> <cadena> <criterios coincidentes> <objetivo>
```

Al construir un firewall, la opción suele ser -A para agregar una regla al final de la regla. colocar. Naturalmente, hay varias opciones según el objetivo y la operación que se está realizando. Este capítulo cubre la mayoría de esas opciones.

Como se indicó anteriormente, la cadena puede ser una cadena de ENTRADA , una cadena de SALIDA , una cadena de ADELANTE cadena o una cadena definida por el usuario. Además, la cadena también podría ser una cadena especial contenida en las tablas nat o mangle .

Los criterios de coincidencia en un comando de iptables establecen las condiciones para que se aplique la regla. Por ejemplo, los criterios de coincidencia se usarían para decirle a iptables que todo el tráfico TCP destinado al puerto 80 puede ingresar al firewall.

Finalmente, el objetivo establece la acción a realizar en un paquete coincidente. El objetivo puede ser algo tan simple como DROP para descartar silenciosamente el paquete, o puede enviar el paquete coincidente a una cadena definida por el usuario, o puede realizar cualquier otra acción configurada en iptables.

Las siguientes secciones de este capítulo muestran ejemplos prácticos que usan iptables para implementar reglas del mundo real para varias tareas. Algunos de los ejemplos incluyen sintaxis y opciones que aún no se han introducido. Si se pierde, consulte esta sección o la página del manual de iptables para obtener más información sobre la sintaxis que se utiliza.

Características de iptables

iptables usa el concepto de tablas de reglas separadas para diferentes tipos de funciones de procesamiento de paquetes. Estas tablas de reglas se implementan como módulos de tablas funcionalmente separados. Los tres módulos principales son la tabla de filtro de reglas, la tabla nat NAT y la tabla mangle especializada en el manejo de paquetes . Cada uno de estos tres módulos de tabla tiene sus propias extensiones de módulo asociadas que se cargan dinámicamente cuando se hace referencia por primera vez, a menos que las haya integrado directamente en el kernel. Otras tablas incluyen raw y security, que tienen usos especiales.

La **tabla de filtros** es la tabla predeterminada. Las otras tablas se especifican mediante una línea de comandos. opción. Las funciones básicas de la **tabla de filtros** incluyen las siguientes:

- Operaciones relacionadas con la cadena en las tres cadenas integradas (ENTRADA, SALIDA y ADELANTE) y en cadenas definidas por el usuario
- Ayuda
- Disposición de destino (ACCEPT o DROP)
- Operaciones de coincidencia de campo de encabezado IP para protocolo, dirección de origen y destino, entrada e interfaces de salida, y manejo de fragmentos
- Operaciones de coincidencia en los campos de encabezado TCP, UDP e ICMP

La **tabla de filtros** tiene dos tipos de extensiones de funciones: extensiones de *destino* y *coincidencia* extensiones Las extensiones de destino incluyen la disposición del paquete REJECT ; los objetivos BALANCE, MIRROR, TEE, IDLETIMER, AUDIT, CLASSIFY y CLUSTERIP ; y las funcionalidades CONNMARK, TRACE y LOG y ULOG , entre otras. Las extensiones de coincidencia admiten la coincidencia en lo siguiente:

- El estado de conexión actual
- Listas de puertos (compatibles con el módulo multipuerto)
- La dirección MAC de origen Ethernet del hardware o el dispositivo físico
- El tipo de dirección, tipo de paquete de capa de enlace o rango de direcciones IP
- Varias partes de los paquetes IPsec o la política IPsec
- El tipo ICMP
- La longitud del paquete
- La hora en que llegó el paquete
- Cada enésimo paquete o paquetes aleatorios
- El usuario, grupo, proceso o ID de grupo de procesos del remitente del paquete
- El campo Tipo de servicio (TOS) del encabezado IP (posiblemente establecido por la tabla mangle)
- La sección TTL del encabezado IP
- El campo de marca de iptables (establecido por la tabla mangle)
- Coincidencia de paquetes con velocidad limitada

La **tabla mangle** tiene dos extensiones de destino. El módulo MARK admite la asignación de un valor al campo de marca del paquete que mantiene iptables . El módulo TOS admite la configuración del valor del campo TOS en el encabezado IP.

La **tabla nat** tiene módulos de extensión de destino para transacciones de direcciones de origen y destino. lación y para la traducción portuaria. Estos módulos admiten estas formas de NAT:

- SNAT — NAT de origen
- DNAT: NAT de destino

• **MASQUERADE:** una forma especializada de NAT de origen para conexiones que son asignado una dirección IP temporal, modificable y asignada dinámicamente (como una conexión telefónica de acceso telefónico)

• **REDIRECT:** una forma especializada de NAT de destino que redirige el paquete al host local, independientemente de la dirección en el campo de destino del encabezado IP.

Se pueden inspeccionar todos los indicadores de estado de TCP y se pueden tomar decisiones de filtrado en función de los resultados iptables puede verificar escaneos sigilosos, por ejemplo.

TCP puede especificar opcionalmente el tamaño máximo del segmento que el remitente está dispuesto a aceptar a cambio. Filtrar en esta única opción de TCP es un caso muy especializado. La sección TTL del encabezado IP también se puede combinar y también es un caso especializado.

Se puede mantener el estado de la conexión TCP y la información de intercambio UDP en curso, permitiendo el reconocimiento de paquetes de forma continua en lugar de sin estado, paquete por paquete. Aceptar paquetes reconocidos como parte de una conexión establecida permite evitar la sobrecarga de verificar la lista de reglas para cada paquete. Cuando se acepta la conexión inicial, los paquetes subsiguientes pueden ser reconocidos y permitidos.

Generalmente, el campo TOS es de interés histórico solamente. El campo TOS se ignora o se usa con las definiciones de servicios diferenciados (DS) más recientes por parte de los enrutadores intermedios. El filtrado IP TOS tiene usos para la priorización de paquetes locales: enrutamiento y reenvío entre hosts locales y el enrutador local.

Los paquetes entrantes se pueden filtrar por la dirección MAC de origen. Esto tiene usos limitados y especializados para la autenticación local porque las direcciones MAC se transmiten solo entre hosts y enrutadores adyacentes.

Los mensajes de registro de filtros individuales pueden tener como prefijo cadenas definidas por el usuario. A los mensajes se les pueden asignar niveles de registro del kernel como se define en la configuración del daemon de registro del sistema. Esto permite activar y desactivar el registro y definir los archivos de salida del registro para un sistema determinado. Además, existe una opción ULOG que envía el registro a un demonio del espacio del usuario, ulogd, para permitir que se registren más detalles sobre el paquete.

Las coincidencias de paquetes se pueden limitar a una tasa de ráfaga inicial, después de lo cual se impone un límite por el número de coincidencias permitidas por segundo. Si la limitación de coincidencias está habilitada, el valor predeterminado es que, después de una ráfaga inicial de cinco paquetes coincidentes, se impone un límite de velocidad de tres coincidencias por hora. En otras palabras, si el sistema estuviera inundado con paquetes de ping, por ejemplo, los primeros cinco ping coincidirían. Después de eso, un solo paquete de ping podría coincidir 20 minutos más tarde y otro podría coincidir 20 minutos después, independientemente de cuántas solicitudes de eco se hayan recibido. La disposición de los paquetes, ya sea que se registren o no, dependería de cualquier regla posterior con respecto a los paquetes.

El destino REJECT puede especificar opcionalmente qué mensaje de error ICMP (o RST para TCP) devolver. El estándar IPv4 requiere que TCP acepte RST o ICMP como una indicación de error, aunque RST es el comportamiento predeterminado de TCP. El valor predeterminado de iptables es no devolver nada (DROP) o devolver un error ICMP (REJECT).

Junto con REJECT, otro objetivo de propósito especial es QUEUE. Su propósito es entregar el paquete a través del dispositivo netlink a un programa de espacio de usuario para su manejo. Si no hay ningún programa en espera, el paquete se descarta.

RETURN es otro objetivo de propósito especial. Su propósito es regresar de un definido por el usuario cadena antes de que se complete la coincidencia de reglas en esa cadena.

Los paquetes salientes generados localmente se pueden filtrar en función del usuario, grupo, proceso o ID de grupo de proceso del programa que genera el paquete. Por lo tanto, el acceso a los servicios remotos se puede autorizar en el nivel de filtrado de paquetes por usuario. Esta es una opción especializada para hosts multiusuario y multipropósito porque los enrutadores de firewall no deberían tener cuentas de usuario.

La coincidencia se puede realizar en varias partes del encabezado IPsec, incluidos los SPI (índices de parámetros de seguridad) del AH (Encabezado de autenticación) y ESP (Carga de pago de seguridad encapsulada).

El tipo de paquete, ya sea de difusión, unidifusión o multidifusión, es otra forma de coincidencia. Esto se hace en la capa de enlace.

Un rango de puertos así como un rango de direcciones también son coincidencias válidas con iptables. El tipo de dirección también es otra coincidencia válida. Relacionado con la coincidencia de tipos está el tipo de paquete ICMP. Recuerde que hay varios tipos válidos de paquetes ICMP. iptables puede coincidir con estos tipos.

La longitud del paquete es una coincidencia válida, al igual que la hora de llegada del paquete. Esta vez el emparejamiento es interesante. Usando las coincidencias de tiempo, puede configurar el firewall para rechazar cierto tráfico después del horario comercial o permitirlo solo durante ciertas horas del día.

Una buena coincidencia para la auditoría, una coincidencia de paquetes aleatorios también está disponible con iptables. Usando esta coincidencia, puede capturar cada enésimo paquete y registrarlo. Este sería un método para auditar las reglas del firewall sin registrar demasiada información.

Características de la tabla NAT

Hay tres formas generales de NAT:

• **NAT de salida unidireccional tradicional** : se utiliza para redes que utilizan direcciones.

• **NAT básica** : solo traducción de direcciones. Por lo general, se usa para mapear una fuente privada local direcciones a uno de un bloque de direcciones públicas.

• **NAPT (traducción de direcciones y puertos de red)**: por lo general, se utiliza para asignar direcciones de origen privadas locales a una única dirección pública (por ejemplo, enmascaramiento de Linux).

• **NAT bidireccional** : la traducción de direcciones bidireccional permite conexiones entrantes. Un uso de esto es el mapeo de direcciones bidireccional entre espacios de direcciones IPv4 e IPv6.

• **Dos veces NAT**: la traducción bidireccional de direcciones de origen y destino permite conexiones entrantes y salientes. Se puede usar Twice NAT cuando los espacios de direcciones de las redes de origen y destino chocan.

Esto podría ser el resultado de que un sitio utilice por error direcciones públicas asignadas a otra persona. Twice NAT también puede

usarse como una conveniencia cuando un sitio se volvió a numerar o se asignó a un nuevo bloque de dirección pública y el administrador del sitio no quería administrar la nueva dirección asignaciones localmente en ese momento.

iptables NAT admite NAT de origen (SNAT) y NAT de destino (DNAT). La tabla nat permite modificar la dirección de origen o la dirección y el puerto de destino de un paquete. Tiene tres cadenas incorporadas:

- La cadena PREROUTING especifica los cambios de destino de los paquetes entrantes antes de pasar el paquete a la función de enrutamiento (DNAT). Los cambios en la dirección de destino pueden ser en el host local (proxy transparente, redirección de puertos) o en un host diferente para el reenvío de host (funcionalidad ipmasqadm, *reenvío de puertos* en la jerga de Linux) o carga compartida.
- La cadena OUTPUT especifica los cambios de destino de los paquetes salientes generados localmente antes de que se haya tomado la decisión de enrutamiento (DNAT, REDIRECT). Esto generalmente se hace para redirigir de manera transparente un paquete saliente a un proxy local, pero también se puede usar para reenviar puertos a un host diferente.
- La cadena POSTROUTING especifica los cambios de origen en los paquetes salientes que se enrutan a través de la caja (SNAT, MASQUERADE). Los cambios se aplican después de que se ha tomado la decisión de enrutamiento.

Enmascaramiento en iptables

En iptables, el enmascaramiento es un caso especializado de NAT de origen en el sentido de que el estado de la conexión enmascarada se olvida inmediatamente si se pierde la conexión. Está diseñado para usarse con conexiones en las que la dirección IP se asigna temporalmente (por ejemplo, acceso telefónico). A un usuario que se vuelve a conectar inmediatamente probablemente se le asignará una dirección IP diferente a la de la conexión anterior. (Este no suele ser el caso con muchos proveedores de servicios de cable módem y ADSL. A menudo, después de una pérdida de conexión, se asigna la misma dirección IP al volver a conectarse).

Con SNAT normal, el estado de la conexión se mantiene durante un período de tiempo de espera. Si se restableciera una conexión lo suficientemente rápido, los programas actuales relacionados con la red podrían continuar sin ser perturbados porque la dirección IP no ha cambiado y el tráfico TCP interrumpido se retransmitiría.

La distinción entre MASQUERADE y SNAT es un intento de evitar una situación que ocurría en implementaciones anteriores de Linux NAT/MASQUERADE. Cuando se perdía una conexión de acceso telefónico y el usuario se volvía a conectar de inmediato, se le asignaba una nueva dirección IP. La nueva dirección no se pudo usar de inmediato porque la dirección IP anterior y la información de NAT todavía estaban en la memoria hasta que expiró el período de tiempo de espera.

La figura 3.4 muestra las cadenas NAT en relación con la función de enrutamiento y las cadenas INPUT, OUTPUT y FORWARD.

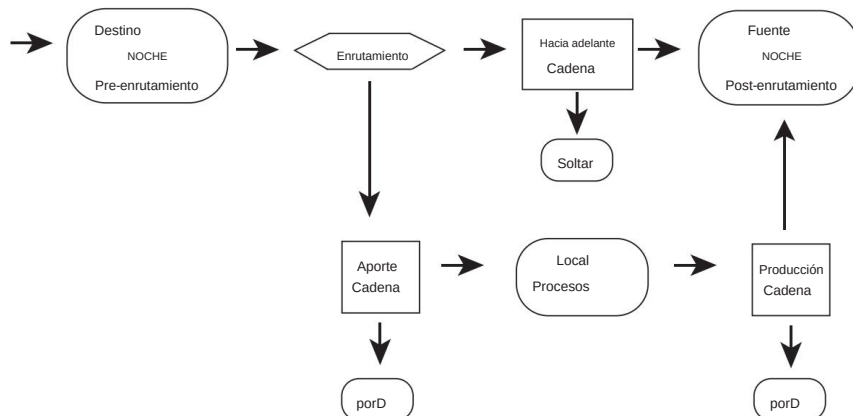


Figura 3.4 Travesía de paquetes NAT

(Figura basada en "Linux 2.4 Packet Filtering HOWTO," v1.0.1, y "Linux 2.4 NAT HOWTO," v1.0.1.)

Tenga en cuenta que, para los paquetes salientes, la función de enrutamiento está implícita entre el proceso local y la cadena de SALIDA. El enrutamiento estático se usa para determinar a qué interfaz saldrá el paquete, antes de que se apliquen las reglas de filtro de la cadena de SALIDA.

Características de la mesa mangle

La *tabla mangle* permite *marcar* o asociar un valor mantenido por Netfilter con el paquete, así como realizar cambios en el paquete antes de enviarlo a su destino. La *mesa de mangle* tiene cinco cadenas incorporadas:

- La *cadena PREROUTING* especifica cambios en los paquetes entrantes a medida que llegan a una interfaz, antes de que se tome una decisión de enrutamiento o entrega local.
- La *cadena INPUT* especifica los cambios en los paquetes a medida que se procesan, pero después de la *Se recorre la cadena PREROUTING*.
- La *cadena POSTROUTING* especifica los cambios en los paquetes a medida que salen del firewall, después de atravesar la *cadena de SALIDA*.
- La *cadena FORWARD* especifica cambios en los paquetes que se reenvían a través del cortafuegos.
- La *cadena OUTPUT* especifica los cambios en los paquetes salientes generados localmente.

Para el campo TOS, el enrutador Linux local se puede configurar para cumplir con los indicadores TOS establecidos por la *tabla mangle* o según lo establecido por los hosts locales.

Hay poca información disponible sobre el marcado de paquetes en la documentación de iptables, más allá de que es utilizada por la implementación de Calidad de servicio (QoS) de Linux y que está pensada como un indicador de comunicación entre los módulos de iptables.

Las secciones anteriores proporcionaron una descripción general de las funciones disponibles en iptables y la estructura general y la funcionalidad de los módulos individuales de la mesa. Las siguientes secciones presentan la sintaxis utilizada para invocar estas características.

Sintaxis de iptables

Como se presentó anteriormente, iptables usa el concepto de tablas de reglas separadas para diferentes funciones de procesamiento de paquetes. Las tablas no predeterminadas se especifican mediante una opción de línea de comandos.

Hay tres tablas principales disponibles, y otras, como la seguridad y la materia prima, tienen un propósito especializado.

Las tres tablas principales son:

• **filter:** la **tabla de filtros** es la tabla predeterminada. Contiene el filtro de firewall real

reglas de ing. Las cadenas integradas incluyen estas:

• ENTRADA

• SALIDA

• ADELANTE

• **nat:** la **tabla nat** contiene las reglas para la dirección y el puerto de origen y destino.

Traducción. Estas reglas son funcionalmente distintas de las reglas de filtrado del cortafuegos. Las cadenas integradas incluyen estas:

• ENRUTAMIENTO PREVIO—DNAT/REDIRECT

• SALIDA—DNAT/REDIRECT

• POSTERIOR AL ENRUTAMIENTO: SNAT/MASQUERADE

• **mangle:** la **tabla mangle** contiene reglas para configurar el enrutamiento de paquetes especializado.

banderas Estas banderas son luego inspeccionadas por reglas en la tabla de filtros. Las cadenas integradas incluyen estas:

• **PREROUTING—** Paquetes enrutados

• INPUT: paquetes que llegan al firewall pero después de la cadena PREROUTING

• FORWARD: cambia los paquetes que se enrutan a través del firewall.

• **POSTROUTING:** **cambia los** paquetes justo antes de que abandonen el cortafuegos, después de que Cadena de SALIDA

• SALIDA: paquetes generados localmente

Convenciones de formato de sintaxis

Las convenciones utilizadas para presentar las opciones de sintaxis de la línea de comandos son bastante estándar en el mundo de la informática. Para aquellos de ustedes que son nuevos en Linux o en la documentación informática en general, la Tabla 3.1 muestra las convenciones utilizadas en las próximas descripciones de sintaxis.

Tabla 3.1 Convenciones que representan las opciones de sintaxis de la línea de comandos

Elemento	Descripción
	Un símbolo de barra o tubería separa las opciones de sintaxis alternativas. Por ejemplo, la mayoría de los comandos de iptables tienen una forma corta y larga, como -L y --list, por lo que se enumerarían como opciones alternativas porque usaría uno u otro de -L o --list .
<valor>	Los corchetes angulares indican un valor proporcionado por el usuario, como una cadena o un valor numérico.
[]	Los corchetes indican que el comando, la opción o el valor adjuntos son opcionales. Por ejemplo, la mayoría de los operadores de coincidencia pueden tomar un operador de negación, !, que coincide con cualquier valor que no sea el especificado en la coincidencia. El operador de negación generalmente se coloca entre el operador de coincidencia y el valor que debe coincidir.
<valor>:<valor>	Los dos puntos indican un rango de valores. Los dos valores definen los valores mínimo y máximo dentro del rango. Debido a que los rangos en sí mismos son opcionales, la convención se presenta más a menudo como <valor>[:<valor>].

Filtrar comandos de tabla

Los comandos de la tabla de filtro son proporcionados por el módulo `ip_tables` . La funcionalidad se habilita cargando el módulo, lo que se hace automáticamente con la primera invocación del comando `iptables` , o se puede compilar en el kernel mismo, lo que significa que no necesita preocuparse por la carga de los módulos.

Filtrar operaciones de tabla en cadenas completas

La Tabla 3.2 muestra las operaciones de iptables en cadenas completas.

El comando `-h` `help` obviamente no es una operación en una cadena, ni tampoco lo es `--modprobe=<comando>`, pero no sabía dónde más enumerar los comandos.

El comando `list` acepta opciones adicionales, como se muestra en la Tabla 3.3.

Filtrar operaciones de tabla en una regla

Los comandos más utilizados para crear o eliminar reglas dentro de una cadena se muestran en la Tabla 3.4.

Operaciones básicas de coincidencia de tabla de filtro

Las operaciones básicas de coincidencia de filtros admitidas en la tabla de filtros predeterminada de iptables se enumeran en la Tabla 3.5.

Opción	Descripción
-N --nueva-cadena <cadena>	Crea una cadena definida por el usuario.
-F --flush [<cadena>]	Limpia la cadena o todas las cadenas si no se especifica ninguna.
-X --delete-chain [<cadena>]	Elimina la cadena definida por el usuario o todas las cadenas si no se especifica ninguna.
-P --política <cadena> <política>	Define la política predeterminada para una de las cadenas integradas, INPUT, OUTPUT o FORWARD. La política es ACCEPT o DROP.
-L --lista [<cadena>]	Enumera las reglas de la cadena o todas las cadenas si no se especifica ninguna.
-S --list-rules [<cadena>]	Imprime las reglas en la cadena especificada en formato de guardado de iptables.
-Z --cero	Restablece los contadores de paquetes y bytes asociados con cada cadena.
-h <algún comando> -h	Enumera los comandos y opciones de iptables o, si está precedido por un comando de iptables, enumera la sintaxis y las opciones para ese comando.
--modprobe=<comando>	Use <comando> para cargar los módulos necesarios al agregar o insertar una regla en una cadena.
-E --rename-chain <cadena antigua>	Cambia el nombre de la cadena definida por el usuario <cadena antigua> a la cadena definida por el usuario <nueva cadena>.

Opción	Descripción
-L -n --numérico	Enumera las direcciones IP y los números de puerto numéricamente, en lugar de por nombre
-L -v --verboso	Muestra información adicional sobre cada regla, como los contadores de bytes y paquetes, las opciones de reglas y la interfaz de red relevante.
-L -x --exacto	Muestra los valores exactos del contador, en lugar de los valores redondeados
-L --números de línea	Enumera la posición de la regla dentro de su cadena

Tabla 3.4 Cadena de Comandos en Reglas Individuales

Mando	Descripción
-A --append <cadena> <especificación de regla>	Agrega una regla al final de un cadena
-yo --insert <cadena> [<número de regla>] <especificación de regla>	Inserta una regla al principio de la cadena.
-R --replace <cadena> <número de regla> <especificación de regla>	Reemplaza una regla en la cadena
-D --delete <cadena> <número de regla> <especificación de regla>	Elimina la regla en la posición <número de regla> dentro de una cadena o la regla que coincide con la regla especificada
-C --check <cadena> <especificación de regla>	Examina la cadena para ver si un la regla coincide con la especificación

Tabla 3.5 Operaciones de regla de tabla de **filtro**

Opción	Descripción
-yo --en-interfaz [!] [<interfaz>]	Para los paquetes entrantes en las cadenas INPUT o FORWARD, o sus subcadenas definidas por el usuario, especifica el nombre de interfaz al que se aplica la regla. Si no se especifica ninguna interfaz, todas las interfaces están implícitas.
-o --out-interface [!] [<interfaz>]	Para paquetes salientes en el Las cadenas OUTPUT o FORWARD, o sus subcadenas definidas por el usuario, especifican el nombre de la interfaz que la regla se aplica a. Si no se especifica ninguna interfaz, todas las interfaces están implícitas.
-p --protocolo [!] [<protocolo>]	Especifica el protocolo IP al que se aplica la regla. Los protocolos incorporados son tcp, udp, icmp y todos. El valor del protocolo puede ser el nombre o el valor numérico, como se indica en /etc/protocolos
-s --fuente --src [!] <dirección>[</máscara>]	Especifica la dirección de origen del host o de la red en el encabezado IP.
-d --destino --dst [!] <dirección>[</máscara>]	Especifica el host o la red. dirección de destino en el encabezado IP.

Tabla 3.5 Operaciones de regla de tabla de **filtros** (continuación)

Opción	Descripción
-j --saltar <objetivo>	Especifica la disposición de destino del paquete si coincide con la regla. Los destinos predeterminados incluyen los destinos integrados, una extensión o una cadena definida por el usuario.
-g --goto <cadena>	Especifica que el procesamiento debe continuar en la cadena especificada, pero no devuelve el procesamiento (como lo hace la opción de salto).
-m --match <coincidencia> [!] -f --fragmento	Utiliza una extensión para probar una coincidencia. Especifica paquetes fragmentados segundos y adicionales. La versión negada de esto especifica paquetes no fragmentados.
-c --set-counters <paquetes> <bytes>	Inicializa los contadores de paquetes y bytes.

Los destinos de las reglas son opcionales

Si el paquete coincide con una regla que no tiene una disposición de destino, los contadores de paquetes se actualizan, pero continúa el recorrido de la lista.

operaciones de coincidencia de tabla de filtro tcp

Las opciones de coincidencia del encabezado TCP se enumeran en la Tabla 3.6. También puede ver estas opciones agregando el indicador -h después de la opción -p tcp , como en iptables -p tcp -h.

Tabla 3.6 Operaciones de coincidencia de tabla de **filtro tcp**

Opción -p tcp	Descripción
--fuente-puerto --sport [!] <puerto>[:<puerto>]]	Este comando especifica los puertos de origen.
--destino-puerto --dport <puerto>[:<puerto>]	Este comando especifica los puertos de destino [!].
--tcp-flags [!] <máscara> [,<máscara>] <conjunto>[:<conjunto>]	Este comando prueba los bits en el lista de máscaras, de la cual se deben configurar los siguientes bits para que coincidan.
[!] -visión	El indicador SYN debe configurarse como una solicitud de conexión inicial.
--tcp-option [!] <número>	La única opción tcp legal es el tamaño máximo de paquete que el host emisor está dispuesto a aceptar.

Tabla 3.7 Operaciones de coincidencia de tabla de **filtro udp**

Opción -pudp	Descripción
--fuente-puerto --sport [!] <puerto>[:<puerto>]	Especifica los puertos de origen
--destino-puerto --dport [!] <puerto>[:<puerto>]	Especifica los puertos de destino

Tabla 3.8 Operación de coincidencia de tabla de **filtro icmp**

Opción	Descripción
--icmp-tipo [!] <tipo>	Especifica el nombre o número del tipo ICMP. El tipo ICMP se utiliza en lugar de un puerto de origen.

Operaciones de coincidencia de tabla de filtro UDP

Las opciones de coincidencia del encabezado UDP se enumeran en la Tabla 3.7. También puede ver estas opciones agregando el indicador **-h** después de la opción **-p udp**, como en **iptables -p udp -h**.

Operación de coincidencia de tabla de filtro icmp

La opción de coincidencia de encabezado ICMP se enumera en la Tabla 3.8.

Los principales nombres de tipo ICMP y valores numéricos admitidos son los siguientes:

ÿ respuesta de eco (0)

ÿ destino inalcanzable (3)

ÿ red inalcanzable

ÿ host inalcanzable

ÿ protocolo inalcanzable

ÿ puerto inalcanzable

ÿ fragmentación necesaria

ÿ red desconocida

ÿ host desconocido

ÿ red prohibida

ÿ host prohibido

ÿ fuente-extinción (4)

ÿ redirigir (5)

ÿ solicitud de eco (8)

ÿ tiempo excedido (10)

ÿ problema de parámetros (11)

Compatibilidad adicional con ICMP e ICMP6

iptables admite varios tipos y subtipos de mensajes ICMP adicionales, menos comunes o específicos del enrutador y también puede funcionar con paquetes ICMP IPv6 a través de la extensión icmp6.

Para ver la lista completa, use los siguientes comandos de ayuda de iptables:

```
iptables -p icmp -h
iptables -p ipv6-icmp -h
```

filtrar las extensiones de destino de la tabla

Las extensiones de destino de la tabla de filtro incluyen funcionalidad de registro y la capacidad de rechazar un paquete en lugar de descartarlo.

La Tabla 3.9 enumera las opciones disponibles para el objetivo LOG . La Tabla 3.10 enumera la única opción disponible para el objetivo REJECT . Al igual que con otras opciones, puede agregar el indicador -h después de -j <OBJETIVO> para ver más opciones, así: iptables -j <OBJETIVO> -h.

La extensión de destino de la tabla ULOG

Relacionado con el objetivo LOG está el objetivo ULOG , que envía el mensaje de registro a un programa de espacio de usuario para el registro. Detrás de escena para ULOG, el kernel multidifunde el paquete

Tabla 3.9 Extensión de destino de LOG

-j Opción REGISTRO	Descripción
--log-level <nivel de registro del sistema>	El nivel de registro es la prioridad de inicio de sesión numérica o simbólica, como se indica en /usr/include/sys/syslog.h. Estos son los mismos niveles de registro utilizados en /etc/syslog.conf. Los niveles son emerg (0), alert (1), crit (2), err (3), warn (4), Notice (5), info (6), memerg (0), alert (1), crit (2), err (3), advertir (4), aviso (5), info (6) y depuración (7).
--log-prefix <"cadena descriptiva">	El prefijo es una cadena entrecomillada que será impreso al comienzo del mensaje de registro de la regla.
--log-ip-opciones	Este comando incluye cualquier opción de encabezado de IP en la salida del registro.
--log-tcp-secuencia	Este comando incluye el número de secuencia del paquete TCP en la salida del registro.
--log-tcp-opción	Este comando incluye cualquier opción de encabezado TCP en la salida del registro.
--log-uid	Este comando incluye el ID de usuario que generó el paquete en la salida del registro.

Tabla 3.10 Extensión de destino **REJECT**

-j Opción RECHAZAR	Descripción
--rechazar-con <ICMP tipo 3>	De manera predeterminada, un paquete rechazado genera un mensaje ICMP tipo 3 icmp-port-unreachable que se devuelve al remitente. En su lugar, se pueden devolver otros mensajes de error de tipo 3, incluidos icmp net-unreachable, icmp-host-unreachable, icmp-proto-unreachable, icmp-net prohibido e icmp-host-prohibited.
--rechazar-con tcp-restablecer	Los paquetes TCP entrantes se pueden rechazar con el mensaje TCP RST más estándar, en lugar de un mensaje ICMP mensaje de error.

a través de un socket de enlace de red de su elección (el valor predeterminado es el socket 1). El daemon del espacio de usuario leería el mensaje del socket y haría con él lo que le placiera. El destino ULOG se usa normalmente para proporcionar un registro más extenso que el que es posible con el destino LOG estándar .

Al igual que con el objetivo LOG , el procesamiento continúa después de las coincidencias en una regla objetivo ULOG . El objetivo ULOG tiene cuatro opciones de configuración, como se describe en la Tabla 3.11.

filtrar extensiones de coincidencia de tabla

Las extensiones de coincidencia de la tabla de filtro brindan acceso a los campos en los encabezados TCP, UDP e ICMP, así como a las funciones de coincidencia disponibles en iptables, como el mantenimiento del estado de conexión, las listas de puertos, el acceso a la dirección de origen MAC del hardware y el acceso a el campo IP TOS .

Tabla 3.11 Extensión de destino de **ULOG**

Opción	Descripción
--ulog-nlgrupo>grupo>	Define el grupo de enlace de red que recibirá el paquete. El grupo predeterminado es 1.
--ulog-prefix <prefijo>	Los mensajes tendrán como prefijo este valor, de hasta 32 caracteres de longitud.
--ulog-cprange <tamaño>	El tamaño en bytes para enviar al socket de enlace de red. El valor predeterminado es 0, que envía el paquete completo.
--ulog-qthreshold <tamaño>	El tamaño en paquetes para poner en cola dentro del kernel. El valor predeterminado es 1, lo que significa que se envía un paquete por mensaje al socket de enlace de red.

Sintaxis de coincidencia

Las extensiones de coincidencia requieren el comando `-m` o `--match` para cargar el módulo, seguido de cualquier opción de coincidencia relevante.

Extensión de coincidencia de tabla de filtro multipuerto

las listas de puertos multipuerto pueden incluir hasta 15 puertos por lista. No se permiten espacios en blanco. No puede haber espacios en blanco entre las comas y los valores del puerto. Los intervalos de puertos no se pueden intercalar en la lista. Además, el comando `-m` multipuerto debe seguir exactamente el especificador `-p <protocolo>` .

La Tabla 3.12 enumera las opciones disponibles para la extensión de coincidencia multipuerto .

La sintaxis multipuerto puede ser un poco complicada. Aquí se incluyen algunos ejemplos y precauciones. La siguiente regla bloquea los paquetes entrantes que llegan a la interfaz `eth0` destinados a los puertos UDP asociados con NetBIOS y SMB, puertos comunes que se explotan en las computadoras con Microsoft Windows y objetivos de los gusanos:

```
iptables -A ENTRADA -i eth0 -p udp \
-m multipuerto --puerto-destino 135,136,137,138,139 -j DROP
```

La siguiente regla bloquea las solicitudes de conexión salientes enviadas a través de la interfaz `eth0` para puertos altos asociados con los servicios TCP NFS, SOCKS y squid:

```
iptables -A SALIDA -o eth0 -p tcp \
-m multipuerto --destino-puerto 2049,1080,3128 --syn -j RECHAZAR
```

Lo que es importante tener en cuenta en este ejemplo es que el comando multipuerto debe seguir exactamente la especificación del protocolo. Habría resultado un error de sintaxis si `--syn` se colocaron entre `-p tcp` y `-m multipuerto`.

Para mostrar un ejemplo similar de colocación de `--syn` , lo siguiente es correcto:

```
iptables -A INPUT -i <interfaz> -p tcp \
-m multipuerto --puerto-fuente 80,443 ! --syn -j ACEPTAR
```

Sin embargo, esto provoca un error de sintaxis:

```
iptables -A INPUT -i <interfaz> -p tcp ! --syn \
-m multipuerto --puerto-origen 80,443 -j ACEPTAR
```

Tabla 3.12 Extensión de coincidencia **multipuerto**

metro opción multipuerto <code>--match</code>	Descripción
<code>--source-port <puerto>[,<puerto>]</code>	Especifica los puertos de origen.
<code>--destination-port <puerto>[,<puerto>]</code>	Especifica los puertos de destino.
<code>--port <puerto> [, <puerto>]</code>	Los puertos de origen y de destino son iguales y coinciden con un puerto de la lista.

Además, la ubicación de los parámetros de origen y destino no es obvia. los

Las siguientes dos variaciones son correctas:

```
iptables -A INPUT -i <interfaz> -p tcp -m multipuerto \
--fuente-puerto 80,443 \
! --syn -d $IPADDR --dport 1024:65535 -j ACCEPTAR
```

y

```
iptables -A INPUT -i <interfaz> -p tcp -m multipuerto \
--fuente-puerto 80,443 \
-d $IPADDR! --syn --dport 1024:65535 -j ACCEPTAR
```

Sin embargo, esto provoca un error de sintaxis:

```
iptables -A INPUT -i <interfaz> -p tcp -m multipuerto \
--fuente-puerto 80,443 \
-d $IPADDR --dport 1024:65535 ! --syn -j ACCEPTAR
```

Este módulo tiene algunos efectos secundarios de sintaxis sorprendentes. Cualquiera de los dos anteriores correct rules produce un error de sintaxis si se elimina la referencia al indicador SYN :

```
iptables -A INPUT -i <interfaz> -p tcp -m multipuerto \
--fuente-puerto 80,443 \
-d $IPADDR --dport 1024:65535 -j ACCEPTAR
```

El siguiente par de reglas, sin embargo, no:

```
iptables -A SALIDA -o <interfaz> \
-p tcp -m multipuerto --destino-puerto 80,443 \
! --syn -s $IPADDR --deporte 1024:65535 -j ACCEPTAR
```

```
iptables -A SALIDA -o <interfaz> \
-p tcp -m multipuerto --destino-puerto 80,443 \
--syn -s $IPADDR --deporte 1024:65535 -j ACCEPTAR
```

Tenga en cuenta que el argumento `--destination-port` del módulo `multipuerto` no es lo mismo que el argumento `--destination-port` o `--dport` del módulo que realiza la coincidencia para los argumentos `-p tcp`.

Extensión de coincidencia de tabla de filtro de límite

La coincidencia de velocidad limitada es útil para reducir la cantidad de mensajes de registro que se generarían durante una avalancha de paquetes registrados.

La Tabla 3.13 enumera las opciones disponibles para la extensión de coincidencia de límite.

La tasa de ráfaga define el número de coincidencias iniciales que se aceptarán. El valor predeterminado es cinco coincidencias. Cuando se alcanza el límite, las coincidencias adicionales se limitan al límite de velocidad. El límite predeterminado es de tres coincidencias por hora. Los especificadores de marco de tiempo opcionales incluyen `/segundo`, `/minuto`, `/hora` y `/día`.

En otras palabras, de forma predeterminada, cuando se alcanza la tasa de ráfaga inicial de cinco coincidencias dentro de el límite de tiempo, como máximo tres paquetes más coincidirán durante la próxima hora, uno cada 20 minutos, independientemente de cuántos paquetes se reciban. Si no se produce una coincidencia dentro del límite de velocidad, la ráfaga se recarga en uno.

Tabla 3.13 Extensión de coincidencia **límite**

-m --Opción de límite de coincidencia	Descripción
--limit <tasa>	Número máximo de paquetes para coincidir dentro del marco de tiempo dado
--limit-burst <número>	Número máximo de paquetes iniciales para hacer coincidir antes de aplicar el límite

Es más fácil demostrar la coincidencia de velocidad limitada que describirla con palabras. La siguiente regla limitará el registro de coincidencias de mensajes de ping entrantes a una por segundo cuando se reciban cinco solicitudes de eco iniciales en un segundo dado:

```
iptables -A ENTRADA -i eth0 \
    -p icmp --petición de eco de tipo icmp \
    -m límite --límite 1/segundo -j REGISTRO
```

También es posible aceptar paquetes con velocidad limitada. Las dos reglas siguientes, en combinación, limitará la aceptación de los mensajes de ping entrantes a uno por segundo cuando se reciban cinco solicitudes de eco iniciales en un segundo determinado:

```
iptables -A ENTRADA -i eth0 \
    -p icmp --petición de eco de tipo icmp \
    -m límite --límite 1/segundo -j ACEPTAR
```

```
iptables -A ENTRADA -i eth0 \
    -p icmp --petición de eco de tipo icmp -j DROP
```

La siguiente regla limita la cantidad de mensajes de registro generados en respuesta a mensajes de redirección ICMP descartados . Cuando se hayan registrado cinco mensajes iniciales en un período de 20 minutos, se generarán como máximo tres mensajes de registro más durante la siguiente hora, uno cada 20 minutos:

```
iptables -A ENTRADA -i eth0 \
    -p icmp --redirección de tipo icmp \
    -m límite -j REGISTRO
```

La suposición en el ejemplo final es que el paquete y cualquier otro paquete de redirección no coincidente se eliminan silenciosamente por la política DROP predeterminada para la cadena INPUT .

Extensión de coincidencia de tabla de filtro de estado

Los filtros estáticos analizan el tráfico paquete por paquete únicamente. La combinación particular de cada paquete de direcciones y puertos de origen y destino, el protocolo de transporte y la combinación de indicadores de estado TCP actual se examina sin referencia a ningún contexto en curso.

Los mensajes ICMP se tratan como eventos de capa 3 IP fuera de banda no relacionados.

La extensión de estado proporciona tecnología adicional de monitoreo y grabación para aumentar la tecnología de filtrado de paquetes estáticos y sin estado. La información de estado se registra cuando se inicia una conexión TCP o un intercambio UDP. Los paquetes subsiguientes no se examinan

solo en función de la información de tupla estática, sino también dentro del contexto del intercambio en curso. En otras palabras, parte del conocimiento contextual generalmente asociado con la capa de transporte TCP superior, o la capa de aplicación UDP, se reduce a la capa de filtro.

Una vez iniciado y aceptado el intercambio, los paquetes subsiguientes se identifican como parte del intercambio establecido. Los mensajes ICMP asociados se identifican como relacionados con un intercambio en particular.

Nota

En la terminología informática, una colección de valores o atributos que juntos identifican de forma única un evento u objeto se denomina tupla. Un paquete UDP o TCP se identifica únicamente por la combinación de tuplas de su protocolo, UDP o TCP, las direcciones de origen y destino, y los puertos de origen y destino.

Para el monitoreo de sesiones, las ventajas de mantener la información de estado son menos obvias para TCP porque TCP mantiene la información de estado por definición. Para UDP, la ventaja inmediata es la capacidad de distinguir las respuestas de otros datagramas. En el caso de una solicitud de DNS saliente, que representa un nuevo intercambio UDP, el concepto de una sesión establecida permite un datagrama de respuesta UDP entrante desde el host y el puerto al que se envió el mensaje original, dentro de una determinada ventana de tiempo. No se permiten los datagramas UDP entrantes de otros hosts o puertos. No son parte del estado establecido para este intercambio en particular. Cuando se aplica a TCP y UDP, los mensajes de error ICMP se aceptan si el mensaje de error está relacionado con la sesión en particular.

Al considerar el rendimiento del flujo de paquetes y la complejidad del firewall, las ventajas son más obvias para los flujos TCP. Los flujos son principalmente una tecnología de rendimiento y optimización del cortafuegos. El objetivo principal de los flujos es permitir eludir la ruta de inspección del firewall para un paquete. En algunos casos, se obtiene un manejo de paquetes TCP mucho más rápido porque los filtros de firewall restantes se pueden omitir si el paquete TCP se reconoce inmediatamente como parte de una conexión en curso permitida. Para las conexiones TCP, el estado de flujo puede ser una gran ventaja en términos de rendimiento de filtrado. Además, las reglas estándar del protocolo de aplicación TCP se pueden contraer en una sola regla de permiso inicial. El número de reglas de filtro se reduce (teóricamente, pero no necesariamente en la práctica, como verá más adelante en el libro).

La principal desventaja es que mantener una tabla de estado requiere más memoria que las reglas de firewall estándar por sí solas. Los enrutadores con 70 000 conexiones simultáneas, por ejemplo, requerirían enormes cantidades de memoria para mantener las entradas de la tabla de estado para cada conexión. El mantenimiento del estado a menudo se realiza en hardware por razones de rendimiento, donde las búsquedas de tablas asociativas se pueden realizar simultáneamente o en paralelo. Ya sea que se implementen en hardware o software, los motores de estado deben ser capaces de revertir un paquete a la ruta tradicional si no hay memoria disponible para la entrada de la tabla de estado.

Además, la creación, búsqueda y eliminación de tablas lleva tiempo en el software. El proceso adicional. Los gastos generales son una pérdida en muchos casos. El mantenimiento del estado es una victoria para los intercambios en curso, como una transferencia FTP o una sesión multimedia de transmisión UDP. Ambos tipos de flujo de datos

representan cantidades potencialmente grandes de paquetes (y pruebas de coincidencia de reglas de filtro). Sin embargo, el mantenimiento del estado no es una ganancia en el rendimiento del firewall para un simple intercambio de cliente/servidor de DNS o NTP. La creación y el desmantelamiento de estados pueden requerir fácilmente tanto procesamiento (y más memoria) que simplemente atravesar las reglas de filtrado para estos paquetes.

Las ventajas también son cuestionables para los cortafuegos que filtran principalmente el tráfico web. Web los intercambios cliente/servidor tienden a ser breves y efímeros.

Las sesiones Telnet y SSH están en un área gris. En enrutadores con mucho tráfico con muchas sesiones de este tipo, la sobrecarga de mantenimiento del estado puede ser una ganancia al pasar por alto la inspección del firewall. Sin embargo, para sesiones bastante inactivas, es probable que la entrada del estado de conexión se agote y se deseché. La entrada de la tabla de estado se volverá a crear cuando llegue el siguiente paquete, después de que haya superado las reglas de firewall tradicionales.

La Tabla 3.14 enumera la opción disponible para la extensión de coincidencia estatal .

El estado de la conexión TCP y la información de intercambio UDP en curso se pueden mantener, lo que permite filtrar los intercambios de red como NUEVO, ESTABLECIDO, RELACIONADO o NO VÁLIDO:

• **NUEVO** es equivalente a la solicitud TCP SYN inicial o al primer paquete UDP.

• **ESTABLECIDO** se refiere a los mensajes TCP ACK en curso después de que se inicia la conexión, a los datagramas UDP subsiguientes intercambiados entre los mismos hosts y puertos, ya los mensajes de respuesta de eco ICMP enviados en respuesta a una solicitud de eco anterior.

• **RELACIONADO** actualmente se refiere únicamente a los mensajes de error de ICMP. Las conexiones secundarias FTP son administradas por el módulo adicional de soporte de seguimiento de conexiones FTP. Con la adición de ese módulo, el significado de RELACIONADO se amplía para incluir la conexión FTP secundaria.

• Un ejemplo de un paquete **NO VÁLIDO** es un mensaje de error ICMP entrante que no fue una respuesta a una sesión actual, o una respuesta de eco que no fue una respuesta a una solicitud de eco anterior.

Idealmente, el uso de la coincidencia ESTABLECIDA permite que el par de reglas de firewall para un servicio sea colapsado en una sola regla que permite el primer paquete de solicitud. Por ejemplo, al usar la coincidencia ESTABLECIDA , una regla de cliente web requiere permitir solo el SYN saliente inicial solicitud. Una solicitud de cliente DNS requiere solo la regla que permite el paquete de solicitud saliente UDP inicial.

Tabla 3.14 Extensión de coincidencia **estatal**

-m --Opción de estado de coincidencia	Descripción
--state <estado>[,<estado>]	Coincide si el estado de conexión es uno en el lista. Los valores legales son NUEVO, ESTABLECIDO, RELACIONADO y NO VÁLIDO.

Con una política de entrada denegada por defecto, el seguimiento de la conexión se puede usar (teóricamente) para reemplazar todos los filtros específicos del protocolo con dos reglas generales que permiten la entrada y salida de paquetes que son parte de una conexión establecida, o paquetes relacionados con la conexión. Se requieren reglas específicas de la aplicación solo para el paquete inicial.

Si bien una configuración de firewall de este tipo podría funcionar muy bien para un sitio pequeño o residencial en la mayoría de los casos, es poco probable que funcione adecuadamente para un sitio más grande o un firewall que maneja muchas conexiones simultáneamente. El motivo se remonta al caso de los tiempos de espera de entrada de la tabla de estado, en los que se reemplaza una entrada de estado para una conexión inactiva debido al tamaño de la tabla y las limitaciones de memoria. El siguiente paquete que habría sido aceptado por la entrada de estado eliminada requiere una regla para permitir el paquete, y la entrada de la tabla de estado debe reconstruirse.

Un ejemplo simple de esto es un par de reglas para un servidor DNS local que funciona como caché y reenviar servidor de nombres. Un servidor de nombres de reenvío de DNS utiliza la comunicación de servidor a servidor. El tráfico de DNS se intercambia entre los puertos de origen y destino 53 en ambos hosts. La relación cliente/servidor UDP se puede hacer explícita. Las siguientes reglas permiten explícitamente solicitudes salientes (NUEVAS), respuestas entrantes (ESTABLECIDAS) y cualquier (RELACIONADO)

Mensajes de error ICMP:

```
iptables -A INPUT -m estado \
--estado ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

```
iptables -A SALIDA --out-interface <interfaz> -p udp \
-s $IPADDR --puerto-origen 53 -d $SERVIDOR_NOMBRE --puerto-destino 53 \
-m estado --estado NUEVO, RELACIONADO -j ACCEPTAR
```

DNS utiliza un protocolo simple de consulta y respuesta. Pero ¿qué pasa con una aplicación que ¿Puede mantener una conexión continua durante períodos prolongados, como una sesión de control de FTP o una sesión de telnet o SSH? Si la entrada de la tabla de estado se borra prematuramente por algún motivo, los paquetes futuros no tendrán una entrada de estado con la que compararse para identificarse como parte de un intercambio ESTABLECIDO .

Las siguientes reglas para una conexión SSH permiten esa posibilidad:

```
iptables -A INPUT -m estado \
--estado ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

```
iptables -A SALIDA -m estado \
--estado ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

```
iptables -A SALIDA --out-interface <interfaz> -p tcp \
-s $IPADDR --source-port $UNPRIVPORTS \
-d $REMOTE_SSH_SERVER --destination-port 22 \
-m estado --estado NUEVO, -j ACCEPTAR
```

```
iptables -A SALIDA --out-interface <interfaz> -p tcp ! --syn \
-s $IPADDR --source-port $UNPRIVPORTS \
-d $REMOTE_SSH_SERVER --destination-port 22 \
-j ACCEPTAR
```

```
iptables -A INPUT --in-interface <interfaz> -p tcp ! --syn \
-s $SERVIDOR_SSH_REMOTO --puerto-origen 22 \
-d $IPADDR --destination-port $UNPRIVPORTS \
-j ACEPTAR
```

Extensión de coincidencia de tabla de filtro mac

La Tabla 3.15 enumera la opción disponible para la extensión de coincidencia de mac .

Recuerde que las direcciones MAC no cruzan los límites del enrutador (o segmentos de red).

Recuerde también que solo se pueden especificar las direcciones de origen. La extensión mac solo se puede usar en una interfaz interna, como las cadenas INPUT, PREROUTING y FORWARD .

La siguiente regla permite conexiones SSH entrantes desde un solo host local:

```
iptables -A INPUT -i <interfaz local> -p tcp \
-m mac --mac-fuente xx:xx:xx:xx:xx:xx \
--fuente-puerto 1024:65535 \
-d <IPADDR> --dport 22 -j ACEPTAR
```

filtro de propietario Extensión de coincidencia de tabla

La Tabla 3.16 enumera las opciones disponibles para la extensión de coincidencia de propietario .

La coincidencia se refiere al creador del paquete. La extensión se puede utilizar en la SALIDA cadena solamente.

Estas opciones de coincidencia no tienen mucho sentido en un enrutador de firewall; hacen más sentido en un host final.

Entonces, digamos que tiene una puerta de enlace de firewall con un monitor, quizás, pero sin teclado.

La administración se realiza desde un host multiusuario local. Una sola cuenta de usuario puede iniciar sesión

Tabla 3.15 Extensión de coincidencia mac

-m --coincidir con la opción mac	Descripción
--mac-source [!] <dirección>	Coincide con la dirección de origen del hardware Ethernet de Capa 2, especificada como xx:xx:xx:xx:xx:xx., en la trama Ethernet entrante

Tabla 3.16 Extensión del partido del propietario

-m --Opción de propietario de coincidencia	Descripción
--uid-propietario <ID de usuario>	Coincidencias en el UID del creador
--gid-propietario <id de grupo>	Coincidencias en el GID del creador
--pid-propietario <id de proceso>	Coincidencias en el PID del creador
--sid-propietario <id de sesión>	Coincidencias en el SID o PPID del creador
--cmd-propietario <nombre>	Coincidencias en un paquete creado por un proceso con comando nombre <nombre>

en el cortafuegos desde este host. En el host multiusuario, el acceso administrativo al firewall podría filtrarse localmente como se muestra aquí:

```
iptables -A SALIDA -o eth0 -p tcp \
-s <IPADDR> --sport 1024:65535 \
-d <fw IPADDR> --dport 22 \
-m propietario --uid-propietario <id de usuario administrador> \
--gid-propietario <id de grupo de administración> -j ACCEPTAR
```

filtro de marca Extensión de coincidencia de tabla

La Tabla 3.17 enumera la opción disponible para la extensión de coincidencia de marcas .

El valor de la marca y la máscara son valores largos sin signo. Si se especifica una máscara, el valor y la máscara se combinan con AND .

En el ejemplo, suponga que un paquete de cliente telnet entrante entre una fuente específica y el destino había sido marcado previamente:

```
iptables -A ADELANTE -i eth0 -o eth1 -p tcp \
-s <alguna dirección src> --sport 1024:65535 \
-d <alguna dirección de destino> --dport 23 \
-m marca --marca 0x00010070 \
-j ACCEPTAR
```

El valor de la marca que se está probando aquí se estableció en algún punto anterior del procesamiento del paquete. El valor de la marca es una bandera que indica que este paquete debe manejarse de manera diferente a otros paquetes.

Extensión de coincidencia de tabla de filtro tos

La Tabla 3.18 enumera la opción disponible para la extensión de coincidencia de tos .

El valor tos puede ser una cadena o valores numéricos:

- ÿ minimizar-retraso, 16, 0x10
- ÿ maximizar el rendimiento, 8, 0x08
- ÿ maximizar la confiabilidad, 4, 0x04
- ÿ minimizar el costo, 2, 0x02
- ÿ servicio normal, 0, 0x00

Tabla 3.17 Extensión de partido de **marca**

-m --Opción de marca de coincidencia	Descripción
--mark <valor>[/<máscara>]	Coincide con los paquetes que tienen el valor de marca asignado por Netfilter

Tabla 3.18 Prórroga del partido **tos**

-m --coincidir con la opción	Descripción
--tos <valor>	Coincidencias en la configuración IP TOS

El campo TOS se ha redefinido como el campo Servicios diferenciados (DS) para que lo utilice el punto de código de servicios diferenciados (DSCP).

Para obtener más información sobre los servicios diferenciados, consulte estas fuentes:

- RFC 2474, "Definición del campo de servicios diferenciados (campo DS) en los encabezados de IPv4 e IPv6"
- RFC 2475, "Una arquitectura para servicios diferenciados"
- RFC 2990, "Próximos pasos para la arquitectura IP QoS"
- RFC 3168, "La adición de notificación de congestión explícita (ECN) a IP"
- RFC 3260, "Nueva terminología y aclaraciones para Diffserv"

Filtro impuro Table Match Extension

Las comprobaciones de validez de paquetes específicas realizadas por el módulo no limpio no están documentadas. El módulo se considera experimental.

La siguiente línea muestra la sintaxis del módulo sucio . El módulo no toma argumentos:

```
-m | --coincidir impuro
```

La extensión sucia podría ser "benedicida" para cuando se publique este libro. Mientras tanto, el módulo se presta a un ejemplo de las opciones de LOG :

```
iptables -A ENTRADA -p ! tcp -m impuro \
    -j LOG --log-prefix "Paquete SIN LIMPIAR: " \
    --log-ip-opciones

iptables -A INPUT -p tcp -m impuro \
    -j REGISTRO --prefijo de registro "TCP SIN LIMPIAR: " \
    --log-ip-opciones \
    --log-tcp-secuencia --log-tcp-opciones
```

```
iptables -A INPUT -m impuro -j DROP
```

Extensión de coincidencia de tabla de filtro de tipo de dirección

La extensión de coincidencia addrtype se utiliza para hacer coincidir los paquetes según el tipo de dirección utilizada, como unidifusión, difusión y multidifusión. Los tipos de direcciones incluyen los enumerados en la Tabla 3.19.

Se utilizan dos comandos con la coincidencia de tipo de dirección , como se muestra en la Tabla 3.20.

Coincidencia de tabla de filtro iprange

A veces, definir un rango de direcciones IP mediante la notación CIDR no es suficiente para sus necesidades. Por ejemplo, si necesita limitar un cierto rango de direcciones IP que no caen en un límite de subred o cruzan ese límite por solo un par de direcciones, el tipo de coincidencia iprange hará el trabajo.

Tabla 3.19 Tipos de direcciones utilizados con **addrtype** Match

Nombre	Descripción
CUALQUIER TRANSMISIÓN	Un paquete anycast
AGUJERO NEGRO	Una dirección de agujero negro
TRANSMISIÓN	Una dirección de transmisión
LOCAL	Una dirección local
MULTIDIFUSIÓN	Una dirección de multidifusión
PROHIBIR	Una dirección prohibida
UNICASTRO	Una dirección de unidifusión
INALCANZABLE	Una dirección inalcanzable
UNSPEC	Una dirección no especificada

Tabla 3.20 Comandos de coincidencia **addrtype**

Opción	Descripción
--src-tipo <tipo>	Coincidencias de direcciones con una fuente de tipo <tipo>
--dst-tipo <tipo>	Coincidencias de direcciones con un destino de tipo <tipo>

Tabla 3.21 Comandos de coincidencia **de iprange**

Mando	Descripción
[!] --src-range <ip address-ip address> Especifica (o niega) el rango de IP	direcciones para hacer coincidir. El rango se da con un solo guión y sin espacios.
[!] --dst-range <ip address-ip address> Especifica (o niega) el rango de IP	direcciones para hacer coincidir. El rango se da con un solo guión y sin espacios.

Con la coincidencia de `iprange` , especifica un rango arbitrario de direcciones IP para que la coincidencia surta efecto. La coincidencia de `iprange` también se puede negar. La Tabla 3.21 enumera los comandos para la coincidencia de `iprange` .

Filtro de longitud Coincidencia de tabla

La coincidencia de la tabla de filtros de longitud examina la longitud del paquete. Si la longitud del paquete coincide con el valor dado u opcionalmente cae dentro del rango dado, se invoca la regla.

La Tabla 3.22 enumera el único comando relacionado con la coincidencia de longitud .

Tabla 3.22 Comando de coincidencia de longitud

Mando	Descripción
--longitud <longitud>[:<longitud>]	Coincide con un paquete de <longitud> o dentro del rango <longitud:longitud>

Extensiones de destino de tabla nat

Como se mencionó anteriormente, iptables admite cuatro tipos generales de NAT: NAT de origen (SNAT); NAT de destino (DNAT); enmascaramiento (MASQUERADE), que es un caso especializado de la implementación SNAT; y la dirección del puerto local (REDIRECT) al host local. Como parte de la tabla nat , cada uno de estos objetivos está disponible cuando una regla especifica la tabla nat mediante el especificador de tabla -t nat .

Extensión de destino de tabla nat SNAT

Network Address and Port Translation (NAPT) es el tipo de NAT con el que las personas están más familiarizadas. Como se muestra en la figura 3.5, la traducción de la dirección de origen se realiza después de tomar la decisión de enrutamiento. SNAT es un objetivo legal solo en la cadena POSTROUTING . Debido a que SNAT se aplica inmediatamente antes de que se envíe el paquete, solo se puede especificar una interfaz de salida.

Algunos documentos se refieren a esta forma de NAT de origen (la forma más común) como NAPT, para reconocer la modificación del número de puerto. La otra forma de NAT unidireccional tradicional es la NAT básica, que no toca el puerto de origen. Ese formulario se usa cuando está traduciendo entre la LAN privada y un grupo de direcciones públicas.

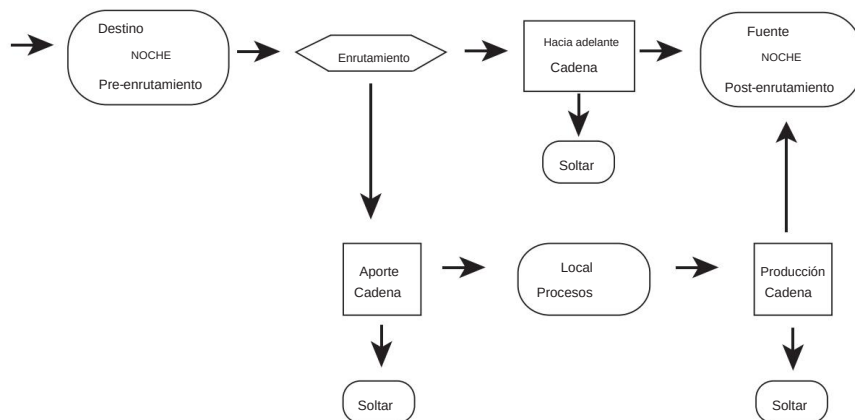


Figura 3.5 Travesía de paquetes NAT

NAPT se usa cuando tiene una sola dirección pública. El puerto de origen se cambia a un puerto libre en el cortafuegos/máquina NAT porque se está traduciendo para cualquier cantidad de computadoras internas, y es posible que la máquina NAT ya esté usando el puerto que la máquina interna está usando. Cuando regresan las respuestas, el puerto es todo lo que la máquina NAT tiene para determinar que el paquete realmente está destinado a una computadora interna en lugar de a sí mismo y luego determinar a qué computadora interna está destinado el paquete.

La sintaxis general para SNAT es la siguiente:

```
iptables -t nat -A POSTROUTING --out-interface <interfaz> ... \
-j SNAT --to-source <dirección> [- <dirección>] [: <puerto> - <puerto>]
```

La dirección de origen se puede asignar a un rango de posibles direcciones IP, si hay más de una disponible.

El puerto de origen se puede asignar a un rango específico de puertos de origen en el enrutador.

Extensión de destino de mesa natural MASQUERADE

La traducción de direcciones de origen se ha implementado de dos maneras diferentes en iptables, como SNAT y como MASQUERADE. La diferencia es que la extensión de destino de MASQUERADE está diseñada para usarse con conexiones en interfaces con direcciones IP asignadas dinámicamente, particularmente en el caso en que la conexión sea temporal y es probable que la asignación de la dirección IP sea diferente en cada nueva conexión. Como se discutió anteriormente, en la sección "Características de la tabla NAT", MASQUERADE puede ser útil para IP dinámica o conexiones móviles en particular.

Debido a que el enmascaramiento es un caso especializado de SNAT, también es un objetivo legal solo en la cadena POSTROUTING, y la regla solo puede referirse a la interfaz de salida. A diferencia del SNAT más generalizado, MASQUERADE no toma un argumento que especifique la dirección de origen para aplicar al paquete. La dirección IP de la interfaz de salida se utiliza automáticamente.

La sintaxis general de MASQUERADE es la siguiente:

```
iptables -t nat -A POSTROUTING --out-interface <interfaz> ... \
-j MASQUERADE [--to-ports <puerto>[:<puerto>]]
```

El puerto de origen se puede asignar a un rango específico de puertos de origen en el enrutador.

Extensión de destino de tabla nat DNAT

La traducción de direcciones y puertos de destino es una forma altamente especializada de NAT. Es más probable que un sitio residencial o de una pequeña empresa encuentre útil esta función si su dirección IP pública se asigna dinámicamente o si el sitio tiene una sola dirección IP y el administrador del sitio desea reenviar las conexiones entrantes a servidores internos que no son públicos. visible. En otras palabras, las características de DNAT se pueden usar para reemplazar el software de reenvío de puertos de terceros requerido anteriormente, como ipmasqadm.

Volviendo a la Figura 3.5, la traducción de dirección y puerto de destino se realiza antes se toma la decisión de enrutamiento. DNAT es un blanco legal en el PREROUTING y OUTPUT cadenas En la cadena PREROUTING, DNAT puede ser un objetivo cuando se especifica la interfaz de entrada. En la cadena de SALIDA, DNAT puede ser un objetivo cuando se especifica la interfaz de salida.

La sintaxis general para DNAT es la siguiente:

```
iptables -t nat -A PREROUTING --in-interface <interfaz> ... \
-j DNAT --to-destination <dirección>[<dirección>]:<puerto>-<puerto>]
iptables -t nat -A SALIDA --out-interface <interfaz> ... \
-j DNAT --to-destination <dirección>[<dirección>]:<puerto>-<puerto>]
```

La dirección de destino se puede asignar a un rango de posibles direcciones IP, si hay más de una disponible.

El puerto de destino se puede asignar a un rango específico de puertos alternativos en el anfitrión de destino.

Extensión de destino de tabla natural REDIRECT

La redirección de puertos es un caso especializado de DNAT. El paquete se redirige a un puerto en el host local. Los paquetes entrantes que de otro modo se reenviarían se redireccionan a la cadena INPUT de la interfaz entrante. Los paquetes salientes generados por el host local se redirigen a un puerto en la interfaz de loopback del host local.

REDIRECT es simplemente un alias, una conveniencia, para el caso especializado de redirigir un paquete a este host. No ofrece ningún valor funcional adicional. DNAT podría usarse fácilmente para causar el mismo efecto.

REDIRECT también es un objetivo legal solo en las cadenas PREROUTING y OUTPUT. En la cadena PREROUTING, REDIRECT puede ser un destino cuando se especifica la interfaz de entrada. En la cadena OUTPUT, REDIRECT puede ser un destino cuando se especifica la interfaz de salida.

La sintaxis general para REDIRECT es la siguiente:

```
iptables -t nat -A PREROUTING --in-interface <interfaz> ... \
-j REDIRECT [--to-ports <puerto>[<puerto>]]
iptables -t nat -A SALIDA --out-interface <interfaz> ... \
-j REDIRECT [--to-ports <puerto>[<puerto>]]
```

El puerto de destino se puede asignar a un puerto diferente o a un rango específico de alter puertos nativos en el host local.

Comandos de tabla mangle

Los destinos y extensiones de la tabla mangle se aplican a las cadenas OUTPUT y PREROUTING.

Recuerde, la tabla de filtros está implícita de forma predeterminada. Para usar las funciones de la tabla mangle, debe especificar la tabla mangle con la directiva -t mangle.

mark mangle Extensión de destino de tabla

La Tabla 3.23 enumera las extensiones de destino disponibles para la tabla mangle.

Tabla 3.23 Extensiones de destino de mangle

Opción -t mangle	Descripción
-j MARK --set-mark <valor>	Establece el valor de la marca de Netfilter para este paquete
-j TOS --set-tos <valor>	Establece el valor de TOS en el encabezado de IP

Hay dos extensiones de destino de tabla mangle : MARK y TOS. MARK contiene la funcionalidad para establecer el valor de marca larga sin firmar para el paquete mantenido por iptables mesa de mangle .

Un ejemplo de uso sigue:

```
iptables -t mangle -A PREROUTING --in-interface eth0 -p tcp \  
-s <alguna dirección src> --sport 1024:65535 \  
-d <alguna dirección de destino> --dport 23 \  
-j MARCA --set-marca 0x00010070
```

TOS contiene la funcionalidad para configurar los bits de TOS en el encabezado de IP. Un ejemplo de uso sigue:

```
iptables -t mangle -A SALIDA ... -j TOS --set-tos <tos>
```

Los posibles valores de TOS son los mismos valores disponibles en la coincidencia de TOS de la tabla de filtros . módulo de extensión.

Resumen

Este capítulo cubrió muchas funciones disponibles en iptables, sin duda, las funciones más utilizadas. He tratado de dar una idea general de las diferencias entre Netfilter e IPFW, aunque solo sea para darle un "aviso" sobre las diferencias de implementación que aparecerán en los siguientes capítulos. Se presentaron las divisiones de implementación modular de tres tablas principales separadas: filtro, mangle y nat. Dentro de cada una de estas divisiones principales, las funciones se dividieron en módulos que brindan extensiones de destino y módulos que brindan extensiones de coincidencia.

El Capítulo 5, "Creación e instalación de un cortafuegos independiente", presenta un ejemplo sencillo de cortafuegos independiente. Se presentan reglas básicas contra suplantación de identidad, denegación de servicio y otras reglas fundamentales. El propósito de este capítulo no es presentar un cortafuegos general para que la gente pueda cortar y pegar para uso práctico, sino demostrar la sintaxis presentada en este capítulo de una manera funcional. El próximo capítulo, Capítulo 4, "nftables: el programa de administración de firewall de Linux", presenta el nuevo sistema Netfilter Tables que servirá como reemplazo de iptables. Los capítulos posteriores cubren más detalles, como cadenas definidas por el usuario, optimización de firewall, LAN, NAT y hosts multitarjeta.

nftables: el cortafuegos de Linux

Programa de Administración

Capítulo 3, "iptables: el programa de administración de firewall heredado de Linux", examinado iptables, el programa de administración desde hace mucho tiempo para los cortafuegos de Linux. Allí se cubrieron la sintaxis y muchas de las opciones dentro de iptables . Este capítulo examina el nuevo programa Net filter Tables (nftables) . El programa nftables estuvo disponible como parte del núcleo principal de Linux a partir de la versión 3.13.

Diferencias entre iptables y nftables

Dentro del kernel, nftables representa una desviación significativa del sistema de filtrado de iptables . nftables reemplaza la funcionalidad no solo en iptables sino también en ip6tables para IPv6, arptables para filtrado ARP y ebtables para filtrado de puente Ethernet. La sintaxis de los comandos para nftables e iptables es diferente, y nftables permite el uso de capacidades de secuencias de comandos adicionales. El programa de administración de nftables se llama nft, y es a partir de ese comando que se construyen los cortafuegos.

A diferencia de iptables, nftables no incluye tablas integradas. depende del administrador istrator para determinar qué tablas se necesitan y agregar esas tablas seguidas de las reglas para el procesamiento. El resto de este capítulo analiza la sintaxis de nftables y su uso para crear un firewall.

Sintaxis básica de nftables

El comando nft proporciona el programa administrativo que se usa para construir un firewall. La sintaxis básica de un comando nftables comienza con el propio programa nft , seguido de un comando y un subcomando y varios argumentos y expresiones. Aquí hay un ejemplo:

```
nft <comando> <subcomando> <cadena> <definición de regla>
```

Los comandos típicos son

• y agregar

• y lista

- insertar

- borrar

- al ras

Los subcomandos típicos son

- mesa

- cadena

- regla

Características de nftables

nftables incluye algunas capacidades similares a un lenguaje de programación de alto nivel, como la capacidad de definir variables e incluir archivos externos. nftables también se puede usar para filtrar y procesar varias familias de direcciones. Estas familias de direcciones incluyen

- ip: direcciones IPv4

- ip6: direcciones IPv6

- inet: direcciones IPv4 e IPv6

- arp: direcciones del protocolo de resolución de direcciones (ARP)

- puente: procesamiento de paquetes puenteados

Cuando no se especifica, la familia de direcciones predeterminada es IP. La capacidad de procesar diferentes familias significa que nftables pretende reemplazar otros mecanismos de filtrado como ebtables y arptables.

La arquitectura de procesamiento general para nftables es determinar la familia de direcciones a la que se aplicará la regla. Luego, nftables usa una o más tablas, que contienen una o más cadenas, que a su vez contienen las reglas de procesamiento. Reglas de procesamiento para nftables se componen de expresiones como la dirección, la interfaz, los puertos u otros datos contenidos en el paquete que se está procesando actualmente y declaraciones como descartar, poner en cola y continuar.

Propina

Las tablas contienen cadenas; las cadenas contienen reglas.

Ciertas familias de direcciones contienen ganchos que permiten que nftables obtenga acceso a paquetes como atraviesan la pila de red en Linux. Esto significa que puede realizar una operación en un paquete antes de que se pase para el enrutamiento o después de que se haya procesado. Para las familias ip, ip6 e inet se aplican los siguientes ganchos:

- **preenrutamiento: paquetes** que acaban de llegar y aún no han sido enrutados o procesados por otras partes de nftables.

- **entrada:** paquetes entrantes que se han recibido y enviado a través de la ruta previa anzuelo .

• **reenviar:** si el paquete se enviará a un dispositivo diferente, estará disponible a través del gancho delantero .

• **salida:** paquetes salientes de procesos en el sistema local.

• **posenrutamiento:** justo antes de salir del sistema, el gancho de posenrutamiento hace que el paquete esté disponible para su posterior procesamiento.

La familia de direcciones ARP usa solo los ganchos de entrada y salida .

Sintaxis de nftables

El comando nft en sí tiene algunas opciones que están disponibles desde la línea de comando y no están directamente relacionadas con la definición de reglas de filtrado. Estas opciones de línea de comandos incluyen:

• **--debug <nivel, [nivel]>:** agrega depuración en <nivel> , como escáner, analizador, eval, netlink, mnl, segtree, proto-ctx o todos.

• **-h | --help:** muestra la ayuda básica.

• **-v | --version:** muestra el número de versión de nft.

• **-n | --numeric:** muestra la dirección y la información del puerto mediante números en lugar de realizar la resolución de nombres.

• **-a | --handle:** mostrar identificadores de reglas.

• **-l | --includepath <directorio>:** agrega <directorio> a la ruta de búsqueda de archivos incluidos.

• **-f | --file <nombre de archivo>:** incluye el contenido de <nombre de archivo>.

• **-i | --interactive:** lee la entrada desde la línea de comando.

Como se indicó anteriormente, no hay tablas predefinidas en nftables. Como tal, depende de usted definir las tablas que desea usar en un sistema nftables . Los comandos disponibles para definir una regla determinada dependen de si está trabajando con una tabla, una cadena o una regla.

Sintaxis de la tabla

Hay cuatro comandos disponibles cuando se trabaja con una tabla:

• **add:** agregue una tabla.

• **eliminar:** elimine una tabla.

• **lista:** muestra todas las cadenas y reglas de una tabla.

• **flush:** borra todas las cadenas y reglas de una tabla.

Puede enumerar qué tablas están disponibles con el siguiente comando (ejecutar como root):

tablas de lista nft

Recuerde, no hay tablas predeterminadas para nftables como las hubo para iptables.

Por lo tanto, el comando list tables no puede devolver nada si no se han definido tablas.

Este sería el comportamiento esperado si acaba de configurar nftables y aún no ha definido un firewall con él. Puede definir una tabla que contenga cadenas y reglas de cortafuegos normales de la siguiente manera:

```
nft agregar filtro de tabla
```

Una vez que se haya agregado la tabla de firewall, el comando `list tables` devolverá el nombre de la tabla:

```
filtro de tabla
```

Se puede obtener más información sobre la tabla con este comando:

```
filtro de tabla de lista nft
```

Al hacerlo, se mostrará información sobre la tabla, incluidas las cadenas definidas en la tabla:

```
filtro de ip de tabla{
}
```

Como muestra el ejemplo, la **tabla de filtros** usa la familia IP y actualmente está vacía.

La tabla de este ejemplo se denomina **filtro**, pero en su lugar podría llamarse cualquier nombre, como **cortafuegos**. Sin embargo, el uso común y el que se muestra en las nftables documentación y ejemplos es llamar a este filtro de tabla como se muestra aquí.

Al enumerar las reglas, es muy útil agregar la opción `-a` para mostrar los números de identificador. Los `handle` se puede usar para modificar o eliminar una regla con bastante facilidad. Este uso se demostrará más adelante en este capítulo cuando se agreguen reglas al firewall.

Al enumerar las reglas de firewall, nftables realizará la resolución de direcciones y puertos. Este comportamiento se modifica con la opción `-n`. Se pueden agregar dos opciones `-n` para evitar la resolución de la dirección y del puerto, como se muestra a continuación:

```
filtro de tabla de lista nft -nn
```

Sintaxis de la cadena

Cuando se opera en una cadena, hay seis comandos disponibles:

• `add`: agrega una cadena a una tabla.

• `create`: cree una cadena dentro de una tabla a menos que ya exista una cadena con el mismo nombre.
existe

• `delete`: eliminar una cadena.

• `flush`: borra todas las reglas de una cadena.

• `list`: muestra todas las reglas de una cadena.

• `rename`: cambie el nombre de una cadena.

Al agregar una cadena, se puede definir el gancho, discutido anteriormente. Además, un `priority` se puede agregar una prioridad opcional a la definición de la cadena.

Hay tres tipos de cadenas base que pueden contener reglas y también tener conectados los ganchos descritos anteriormente. Es necesario definir el tipo de cadena y el tipo de gancho.

durante la creación de la cadena y son vitales para la operación de la cadena en un escenario de firewall normal. Si el tipo de cadena y el tipo de enlace no están definidos, los paquetes no se enrutarán a la cadena.

Los tres tipos de cadenas base son:

• **filtro:** se utiliza para el filtrado de paquetes

• **ruta:** se utiliza para el enrutamiento de paquetes

• **nat:** se utiliza para la traducción de direcciones de red (NAT)

Se pueden agregar y usar otras cadenas para agrupar reglas similares. A medida que los paquetes atraviesan una cadena base, el procesamiento se puede enrutar hacia una o más de las cadenas definidas por el usuario para un procesamiento adicional.

Al agregar una cadena, se debe especificar la tabla dentro de la cual se definirá la cadena. Por ejemplo, este comando agrega una cadena de entrada a la tabla de filtros (definida en la sección anterior):

```
nft agregar entrada de filtro de cadena { tipo filtro gancho entrada prioridad 0 \; }
```

El comando indica que se agregará una cadena llamada entrada a la tabla llamada filtro. El tipo de cadena será una cadena base de filtro y se adjuntará al enlace de entrada con una prioridad de 0. Cuando se ingresa este comando desde la línea de comando, se debe agregar un solo espacio seguido de un punto y coma entre las llaves. Cuando este comando se usa dentro de un script nft nativo, se pueden omitir el espacio y la barra invertida.

Agregar una cadena de salida es similar, simplemente intercambiando entrada por salida cuando corresponda:

```
nft agregar salida de filtro de cadena { tipo filtro gancho salida prioridad 0 \; }
```

Sintaxis de la regla

Las reglas son donde tiene lugar la acción de filtrar. Hay tres comandos cuando se trabaja con reglas:

• **add:** agregue una regla.

• **insertar:** anteponga una regla en la cadena, ya sea al principio o en la ubicación especificado.

• **eliminar:** elimine una regla.

Dentro de las reglas, usted especifica los criterios de coincidencia para una regla determinada y un veredicto o decisión sobre lo que debería suceder con un paquete que coincida con esa regla. nftables y las reglas creadas en ellas utilizan varias declaraciones y expresiones para crear la definición.

Las declaraciones de nftables son similares a las declaraciones de iptables y generalmente afectan cómo el paquete se procesará, ya sea deteniendo el procesamiento, enviando el procesamiento a otra cadena o simplemente registrando el paquete. Las declaraciones y veredictos incluyen lo siguiente:

• **accept:** acepta el paquete y detiene el procesamiento.

• **continuar:** continúa procesando el paquete.

• **drop:** detiene el procesamiento y descarta silenciosamente el paquete.

• **goto:** envía el procesamiento a la cadena especificada pero no regresa a la cadena de llamada.

• **jump:** Envía el procesamiento a la cadena especificada y regresa a la cadena de llamada cuando done o cuando se ejecuta una declaración de retorno.

• **limit:** procesa el paquete de acuerdo con la regla si se alcanzó el límite de paquetes recibidos coincidentes.

• **log:** registra el paquete y continúa con el procesamiento.

• **cola:** detener el procesamiento y enviar el paquete al proceso de espacio de usuario.

• **rechazar:** **detener el** procesamiento y rechazar el paquete.

• **return:** **envía el** procesamiento de vuelta a la cadena de llamada.

Las expresiones de nftables pueden ser específicas para una familia de direcciones o un tipo de paquete que se está procesando. nftables usa expresiones de carga útil y metaexpresiones. Las expresiones de carga útil son aquellas que se recopilan a partir de la información del paquete. Por ejemplo, hay ciertas expresiones de encabezado como sport y dport (puerto de origen y puerto de destino, respectivamente) que se aplican a los paquetes TCP y UDP y no tienen sentido en la capa IPv4 e IPv6, ya que esas capas no usan puertos. Las metaexpresiones se pueden usar para reglas que se aplican ampliamente o están vinculadas a propiedades comunes de paquete o interfaz.

La Tabla 4.1 describe las metaexpresiones disponibles.

Las expresiones de seguimiento de conexión (a veces denominadas conntrack) funcionan con los metadatos del paquete para proporcionar información para el procesamiento posterior de reglas. Las expresiones Conntrack se incluyen con la palabra clave ct seguida de una de estas opciones:

• **ip** _

• **direction**

• **expiration**

• **helper**

• **l3proto**

• **mark**

• **protocol**

• **proto-src**

• **proto-dst**

• **saddr**

• **state**

• **state**

La expresión de estado es importante en el uso de cortafuegos. Inspección de paquete normal El procesamiento de ciones y reglas no tiene estado, lo que significa que el procesamiento no sabe nada sobre el paquete procesado previamente. Cada paquete es inspeccionado de acuerdo a sus características únicas.

Tabla 4.1 Metaexpresiones en nftables

Expresión	Descripción
si si	Índice de la interfaz que recibió el paquete
iifname	Nombre de la interfaz en la que se recibió el paquete
si tipo	Tipo de interfaz en la que se recibió el paquete
longitud	Longitud del paquete en bytes
marca	la marca del paquete
oif	Índice de la interfaz que generará el paquete
oifname	Nombre de la interfaz en la que se enviará el paquete
prioridad de	Tipo de interfaz por la que se enviará el paquete
tipo fijo	La prioridad del paquete TC
protocolo	El protocolo EtherType
idclasert	Reino de enrutamiento para el paquete
skgid	Identificador de grupo del socket de origen
skuid	Identificador de usuario del socket de origen

características de las direcciones de origen y destino, puertos y otros criterios. La expresión de estado, que se enumera a continuación, permite que se registre información sobre el paquete para que la regla de procesamiento tenga un contexto sobre el intercambio en curso del tráfico relacionado.

• **nuevo:** un nuevo paquete que llega al cortafuegos, un paquete TCP con el indicador SYN establecido, por ejemplo

• **establecido:** un paquete que forma parte de una conexión que ya se está procesando o rastreado

• **no válido:** un paquete que no se ajusta a las reglas del protocolo

• **relacionado:** un paquete relacionado con una conexión para un protocolo que no utiliza otros medios para rastrear su estado, como ICMP o FTP pasivo.

• **sin seguimiento:** un estado administrativo que se utiliza para omitir el seguimiento de la conexión, typi solo se utiliza en casos especiales

En la práctica, los estados nuevo, relacionado y establecido se usan con frecuencia y el estado inválido se usa cuando corresponde. Por ejemplo, a continuación se muestra una regla para permitir conexiones SSH establecidas y relacionadas. Permitir conexiones relacionadas es importante en caso de que se vacíe la memoria de estado, negando así cualquier estado de conexión establecido.

nft agregar entrada de filtro de regla tcp dport 22 ct estado establecido, relacionado aceptar

La sección titulada “Extensión de coincidencia de tabla de filtro de estado ” en el Capítulo 3, “iptables: El programa de administración de firewall de Linux heredado”, discutió el mecanismo estatal en detalle.

Las expresiones de carga útil se utilizan para crear reglas que coinciden con ciertos criterios específicos y son estrechamente relacionado con el tipo de paquete que se procesa.

La Tabla 4.2 describe las expresiones para los encabezados de IPv4.

La Tabla 4.3 describe las expresiones para los encabezados de IPv6.

La Tabla 4.4 describe las expresiones para los encabezados TCP.

UDP es generalmente un protocolo más simple y, como tal, hay menos expresiones para los encabezados UDP, como se describe en la Tabla 4.5.

La Tabla 4.6 muestra las expresiones de encabezado disponibles para ARP.

Tabla 4.2 Expresiones de carga útil para IPv4

Expresión	Descripción
suma de control	Suma de comprobación del encabezado IP
dest	Dirección IP de destino
frag	Compensación de fragmentación
len	Longitud del encabezado IP, incluidas las opciones
id	identificador de IP
total len	Longitud total del paquete
proto	Protocolo en uso en la capa por encima de IP
src	Dirección IP origen
tos	Valor del tipo de servicio
ttl	Valor de tiempo de vida
ver	Versión de cabecera IP, que siempre será 4 para expresiones IPv4

Tabla 4.3 Expresiones de encabezado de IPv6

Expresión	Descripción
dest	Dirección IP de destino
flow label	Etiqueta de flujo
hop limit	Límite de salto
len	Longitud de la carga útil
next header	protocolo next header
priority	Valor de prioridad
src	Dirección IP origen
ver	Versión de cabecera IP, que siempre será 6 para expresiones IPv6

Tabla 4.4 Expresiones de encabezado TCP

Expresión	Descripción
acuse de recibo	Número de acuse de recibo
suma de control	Suma de comprobación del paquete
quitarse	Compensación de datos
puerto deportivo	Puerto al que va destinado el paquete
banderas	Indicadores de TCP
secuencia	Secuencia de números
deporte	Puerto desde el que se originó el paquete
Urgptr	Valor de puntero urgente
ventana	Valor de la ventana TCP

Tabla 4.5 Expresiones de encabezado UDP

Expresión	Descripción
suma de control	Suma de comprobación del paquete
puerto deportivo	Puerto al que va destinado el paquete
longitud	Longitud total del paquete
deporte	Puerto desde el que se originó el paquete

Tabla 4.6 Expresiones de encabezado ARP

Expresión	Descripción
helenia	Longitud de dirección de hardware
tipoh	Tipo de equipo ARP
en	Operación
completo	Longitud de la dirección del protocolo
tipo p	Tipo de éter

Operaciones básicas de nftables

Al agregar una regla, la tabla y la cadena se especifican junto con los criterios de coincidencia. Por ejemplo, agregar una regla para aceptar conexiones SSH desde un host específico se vería así. Esta regla se agrega a la cadena de entrada creada anteriormente de la tabla de filtros :

```
nft agregar entrada de filtro tcp dport 22 aceptar
```

Las diversas declaraciones como aceptar, descartar, rechazar, registrar y otras (enumeradas anteriormente en esta sección) se denominaron *extensiones* en iptables. Muchas de las mismas opciones y modos de

La operación que funcionó para esas extensiones ahora también funciona con nftables. Por ejemplo, para registrar las conexiones entrantes, se utiliza la declaración de registro . Esta declaración se puede combinar con el seguimiento de conexiones de modo que solo se registren las nuevas conexiones al puerto 22 . Además, se puede agregar un límite para que el mecanismo de registro no se vea abrumado.

Iniciar sesión en nftables requiere los módulos del kernel nfnetlink_log o xt_LOG o soporte del núcleo. Además, debe habilitar el registro haciendo eco de "ipt_LOG" en la entrada de proceso nf_log :

```
echo "ipt_LOG" > /proc/sys/net/netfilter/nf_log/2
```

El comando final de nftables para registrar nuevas conexiones SSH (velocidad limitada) se ve así:

```
nft agregar filtro entrada tcp dport 22 ct estado nuevo límite velocidad 3/segundo registro
```

Las metaexpresiones, como las que seleccionan la interfaz entrante o saliente, se utilizan como selectores adicionales dentro de una regla. Por ejemplo, para registrar nuevas conexiones que llegan a la interfaz eth0 , el comando se ve así:

```
nft agregar entrada de filtro iif eth0 ct estado nuevo límite tasa 10/minuto registro
```

El capítulo 3 contiene reglas de sintaxis y opciones para las diversas expresiones.

Sintaxis del archivo nftables

Una de las mejores características de nftables es la capacidad de leer archivos externos que contienen reglas de nftables . Estos archivos permiten importar y utilizar conjuntos de reglas guardados sin necesidad de crear scripts de shell largos y complejos. Dicho esto, un script de shell sigue siendo útil como contenedor principal para los archivos de reglas de firewall, importándolos en el momento adecuado.

Los archivos se importan con la opción -f a nftables. Por ejemplo, este archivo crea un Cortafuegos de filtrado básico que registra nuevos paquetes SSH (tasa limitada):

```
filtro de tabla {
    entrada de cadena {
        tipo filtro gancho entrada prioridad 0;
        tcp dport 22 ct estado nueva velocidad límite 3/segundo prefijo de registro "NUEVO y paquete:"
    }

    salida en cadena {
        tipo filtro gancho entrada prioridad 0;
    }
}
```

Suponiendo que el archivo se guardó con el nombre firewall.nft, podría cargarse con el siguiente comando:

```
nft -f cortafuegos.nft
```


Resumen

nftables es similar a iptables en la medida en que las reglas y opciones generalmente se traducen bien cuando se construye un firewall. nftables utiliza tablas, que contienen cadenas, que a su vez contienen reglas. Las reglas le dicen a nftables qué hacer con el paquete que se está procesando. Al igual que iptables, nftables puede aceptar, descartar, rechazar, registrar y realizar acciones similares en un paquete. nftables también puede incluir procesamiento basado en estado. nftables reemplaza arptables, iptables y ebtables.

Debido a que muchas de las reglas y gran parte del funcionamiento de nftables son similares a las de iptables, puede usar el Capítulo 3 como referencia para aquellas expresiones que no se trataron explícitamente en este capítulo.

Esta página se dejó en blanco intencionalmente

Construcción e instalación de un Cortafuegos independiente

El Capítulo 2, "Conceptos de filtrado de paquetes", cubrió las ideas y conceptos básicos detrás de un firewall de filtrado de paquetes. Cada cadena de reglas de firewall tiene su propia política predeterminada. Cada regla no solo se aplica a una cadena de ENTRADA o SALIDA individual, sino que también se puede aplicar a una interfaz de red específica, tipo de protocolo de mensaje (como TCP, UDP o ICMP) y número de puerto de servicio. Las reglas individuales de aceptación, denegación y rechazo se definen para la ENTRADA cadena y la cadena de SALIDA, así como para la cadena FORWARD, sobre la que aprenderá en el Capítulo 7, "Reenvío de paquetes". Este capítulo reúne esas ideas para demostrar cómo construir un firewall simple de un solo sistema para su sitio.

El cortafuegos que construirá en este capítulo se basa en una política de denegación de todo por defecto. Todo el tráfico de red está bloqueado de forma predeterminada. Los servicios se habilitan individualmente como excepciones a la política.

Una vez construido el cortafuegos de un solo sistema, el Capítulo 7 y el Capítulo 8, "NAT: traducción de direcciones de red", demuestran cómo ampliar el cortafuegos independiente a un cortafuegos de doble base. Un cortafuegos multitarjeta tiene al menos dos interfaces de red. Aísla una LAN interna de la comunicación directa con Internet. Protege su LAN interna al aplicar reglas de filtrado de paquetes en las dos interfaces de reenvío y, con la adición de Traducción de direcciones de red (NAT), al actuar como una puerta de enlace de proxy entre la LAN e Internet. NAT no es un servicio proxy, en el sentido de que no proporciona un punto de terminación intermedio para la conexión. NAT es similar a un proxy en el sentido de que los hosts locales están ocultos de la Internet pública.

Los cortafuegos de un solo sistema y de doble base son las formas menos seguras de arquitectura de cortafuegos. Si el host del cortafuegos se viera comprometido, cualquier máquina local estaría expuesta a ataques. Como firewall independiente, es una propuesta de todo o nada. Un host de alojamiento único se encuentra con mayor frecuencia en una zona desmilitarizada (DMZ) que alberga un servicio público de Internet o en un entorno residencial.

En el caso de un hogar con un solo sistema o una pequeña empresa, la suposición es que la mayoría de los usuarios tienen una sola computadora o dispositivo conectado a Internet o una sola máquina de firewall que protege una pequeña LAN privada. La suposición es que estos sitios simplemente no tienen los recursos para extender el modelo a una arquitectura con niveles adicionales de firewalls.

Sin embargo, el término *menos seguro* no implica necesariamente un cortafuegos inseguro. Estos firewalls son menos seguros que las arquitecturas más complicadas que involucran varias máquinas.

La seguridad es un compromiso entre los recursos disponibles y los rendimientos decrecientes del próximo dólar gastado. El Capítulo 7 presenta configuraciones alternativas que permiten una seguridad interna adicional que protege configuraciones de servidores y LAN más complicadas que las que puede ofrecer un firewall de un solo sistema.

Los programas de administración de firewall de Linux

Este libro está basado en la serie del kernel de Linux 3.14. La mayoría de las distribuciones actuales de Linux vienen con el mecanismo de firewall Netfilter presentado en el Capítulo 3, "iptables: el programa de administración de firewall heredado de Linux". Este mecanismo suele denominarse iptables, el nombre de su programa de administración. Las distribuciones de Linux más antiguas usaban el mecanismo IPFW anterior. Ese mecanismo de firewall generalmente se conoce como ipfwadm o ipchains, los nombres del programa de administración de la versión anterior. A medida que las distribuciones se actualicen a kernels posteriores a la 3.13, es probable que se incluya el nuevo mecanismo de cortafuegos de Netfilter, nftables o nft (como se llama su programa de administración).

Como programa de administración de cortafuegos, iptables crea las reglas de filtrado de paquetes individuales para las cadenas de ENTRADA y SALIDA que componen el cortafuegos. nftables no crea tablas, cadenas o reglas predeterminadas, por lo que las tablas para filtrar paquetes que contienen cadenas que están conectadas a los ganchos de ENTRADA y SALIDA deben crearse manualmente.

Uno de los aspectos más importantes de la definición de reglas de firewall es el orden en que se definen las reglas. Por lo general, las reglas de filtrado de paquetes se almacenan en una tabla o tablas de filtro del kernel, dentro de las cadenas INPUT, OUTPUT o FORWARD, en el orden en que se definen. Las reglas individuales se insertan al comienzo de la cadena o se agregan al final de la cadena. Todas las reglas se adjuntan en los ejemplos de este capítulo (con una excepción al final del capítulo). El orden en que define las reglas es el orden en que se agregarán a las tablas del núcleo y, por lo tanto, el orden en que se compararán las reglas con cada paquete.

A medida que cada paquete de origen externo llega a la interfaz de red, sus campos de encabezado se comparan con cada regla en la cadena INPUT de la interfaz hasta que se encuentra una coincidencia. Por el contrario, a medida que se envía cada paquete generado localmente, sus campos de encabezado se comparan con cada regla en la cadena de SALIDA de la interfaz hasta que se encuentra una coincidencia. En cualquier dirección, cuando se encuentra una coincidencia, la comparación se detiene y se aplica la disposición del paquete de la regla: ACCEPT, DROP u, opcionalmente, REJECT. Si el paquete no coincide con ninguna regla de la cadena, se aplica la política predeterminada para esa cadena. La conclusión es que gana la primera regla coincidente.

Los números de puerto de servicio numérico, en lugar de sus nombres simbólicos, como se indica en /etc/services, se utilizan en todos los ejemplos de filtros de este capítulo. iptables y nftables admiten los nombres de puerto de servicio simbólico. Los ejemplos de este capítulo usan valores numéricos porque los nombres simbólicos no son consistentes entre las distribuciones de Linux, o incluso entre una versión y la siguiente. Puede utilizar los nombres simbólicos para mayor claridad en sus propias reglas, pero recuerde que su cortafuegos podría romperse con el siguiente sistema.

potenciar. He encontrado que es mucho más confiable usar los propios números de puerto. Lo último que desea en un firewall es la ambigüedad, que es justo lo que se introduce al usar nombres en lugar de números para los puertos.

La mayoría de las distribuciones de Linux implementan iptables como un conjunto de módulos de programa cargables. La mayoría o todos los módulos se cargan dinámica y automáticamente en el primer uso. Si elige crear su propio kernel, lo que casi siempre hago, y lo que deberá hacer si su distribución aún no tiene soporte para nftables, deberá compilar para que sea compatible con Netfilter, ya sea como módulos o directamente. en el núcleo.

Cuando trabaje con iptables, el comando iptables debe invocarse una vez para cada regla de firewall individual que defina. Esto se hace inicialmente desde un script de shell. Este capítulo creará y usará un script llamado rc.firewall para un firewall. La ubicación en la que se debe colocar este script depende del tipo de Linux donde se utilizará el script. En la mayoría de los sistemas, incluidos Red Hat/CentOS/Fedora y Debian, la ubicación correcta sería dentro de `/etc/init.d/`. En los casos en los que la semántica del shell difiera, los ejemplos están escritos en Bourne (sh) o Bourne Again (bash) semántica de shell.

El script de shell de iptables establece una serie de variables. El principal de ellos es la ubicación del propio comando iptables. Es importante configurar esto en una variable para que se ubique explícitamente. No hay excusa para la ambigüedad con un script de firewall. La variable utilizada para representar el comando iptables en este capítulo es `$IPT`. Si ve `$IPT`, es un sustituto del comando iptables. Podría ejecutar fácilmente los comandos desde el shell escribiendo iptables en lugar de `$IPT`. Sin embargo, para usar en un script (que es la intención de este capítulo), establecer esta variable es una buena idea.

La secuencia de comandos debe comenzar con la línea "shebang" que invoca al shell como intérprete de la secuencia de comandos. En otras palabras, ponga esto como la primera línea del script:

```
#!/bin/sh
```

Los ejemplos no están optimizados. Se detallan para mayor claridad. Optimización de cortafuegos y las cadenas definidas por el usuario se analizan por separado en el Capítulo 6, "Optimización de firewall".

El script de firewall de nftables comienza como un script de shell, pero también incluye varias reglas directas de nftables como archivos separados e incluidos. Estos archivos se llamarán cuando se utilicen más adelante en este capítulo.

El resto del capítulo analiza la creación de un firewall y muestra ejemplos del uso de iptables y nftables para cada elemento.

Construir versus comprar: el kernel de Linux

Existe un gran debate sobre si es aconsejable compilar un kernel personalizado o quedarse con el kernel "estándar" que viene con una distribución de Linux determinada. El debate también incluye si es intrínsecamente mejor compilar un kernel monolítico (en el que todo se compila en el kernel) o usar un kernel modular. Como con cualquier debate, hay ventajas y desventajas para cada método. Por un lado, están aquellos que siempre (o casi siempre) construyen su propio kernel, a veces llamado "rolling your own". Por otro lado, están aquellos que rara vez o nunca lanzan su propio kernel. Hay quienes siempre construyen núcleos monolíticos y otros que utilizan núcleos modulares.

Construir un kernel personalizado tiene algunas ventajas. La primera es la capacidad de compilar solo los controladores exactos y las opciones necesarias para que la computadora funcione. Esto es excelente para un servidor como un firewall porque el hardware rara vez cambia, si es que alguna vez lo hace. Otra ventaja de compilar un kernel personalizado, si elige un kernel monolítico, es la capacidad de prevenir por completo algunos tipos de ataques contra la computadora. Aunque los ataques contra núcleos monolíticos son posibles, son menos comunes que los ataques contra núcleos modulares. Además, cuando implementa su propio kernel, no está limitado a la versión del kernel utilizada por la distribución. Esto le permite utilizar el último y mejor kernel, que puede incluir correcciones de errores para su hardware. Finalmente, con un kernel personalizado puede aplicar mejoras de seguridad adicionales al propio kernel.

La construcción de un núcleo personalizado no deja de tener su propio conjunto de trampas. Después de rodar el tuyo kernel, ya no puede usar las actualizaciones del kernel de la distribución. En realidad, puede volver al kernel de la distribución y usar las actualizaciones, pero es probable que la distribución use una versión anterior del kernel que puede volver a introducir errores que se corrigieron en su versión personalizada. El uso de un kernel estándar también facilita la obtención de soporte del proveedor para problemas con el kernel.

Como se mencionó anteriormente, casi siempre implemento mi propio kernel para máquinas de servidor de producción. Las situaciones en las que el apoyo directo es un requisito absoluto son las únicas excepciones. Estos son pocos y distantes entre sí. Creo que la capacidad de personalizar el kernel para la computadora y agregar mayor seguridad a través de parches adicionales supera con creces la necesidad de usar actualizaciones oficiales del kernel de la distribución.

Opciones de direccionamiento de origen y destino

La dirección de origen y la dirección de destino de un paquete se pueden especificar en una regla de firewall.

Solo los paquetes con esa dirección de origen y/o destino específica coinciden con la regla.

Las direcciones pueden ser una dirección IP específica, un nombre de host completamente calificado, un nombre o dirección de red (dominio), un rango limitado de direcciones o todo incluido.

Direcciones IP expresadas como nombres simbólicos

Las redes y hosts remotos pueden especificarse como nombres de host o nombres de red completamente calificados. El uso de un nombre de host es especialmente conveniente para las reglas de firewall que se aplican a un host remoto individual. Esto es particularmente cierto para los hosts cuya dirección IP puede cambiar o que representan de forma invisible varias direcciones IP, como a veces lo hacen los servidores de correo ISP. En general, sin embargo, las direcciones remotas se expresan mejor en notación cuádruple con puntos debido a la posibilidad de suplantación de nombre de host DNS.

Los nombres de host simbólicos no se pueden resolver hasta que el tráfico de DNS esté habilitado en las reglas del firewall. Si se usan nombres de host en las reglas del firewall, esas reglas deben seguir las reglas que habilitan el tráfico de DNS, a menos que `/etc/hosts` contenga entradas para los nombres de host.

Además, algunas distribuciones usan un entorno de inicio que instala las reglas del firewall antes de iniciar la red o cualquier otro servicio, incluido BIND. Si se utilizan nombres simbólicos de host y red en la secuencia de comandos del cortafuegos, esos nombres deben tener entradas en `/etc/hosts` para que se resuelvan.

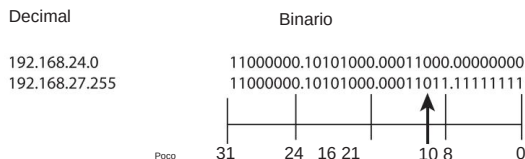


Figura 5.1 Los primeros 22 bits coincidentes en el rango de direcciones IP enmascaradas 192.168.24.0/22

Tanto iptables como nftables permiten que las direcciones tengan como sufijo un especificador de máscara de bits. El valor de la máscara puede variar de 0 a 32, indicando el número de bits a enmascarar. Como se discutió en el Capítulo 1, "Conceptos preliminares subyacentes a los cortafuegos de filtrado de paquetes", los bits se cuentan desde la izquierda, o el bit más significativo. Este especificador de máscara indica cuántos de los bits iniciales de la dirección deben coincidir exactamente con la dirección IP especificada.

Una máscara de 32, /32, significa que todos los bits deben coincidir. La dirección debe coincidir exactamente con lo que ha definido en la regla. Especificar una dirección como 192.168.10.30 es lo mismo que especificar la dirección como 192.168.10.30/32. La máscara /32 está implícita de forma predeterminada; no es necesario especificarlo.

Un ejemplo de enmascaramiento es permitir que las conexiones a un servicio en particular se realicen solo entre su máquina y las máquinas del servidor de su ISP. Digamos que su ISP usa direcciones en el rango de 192.168.24.0 a 192.168.27.255 para el espacio de direcciones de su servidor. En este caso, el par dirección/máscara sería 192.168.24/22. Como se muestra en la figura 5.1, los primeros 22 bits de todas las direcciones en este rango son idénticos, por lo que cualquier dirección que coincida con los primeros 22 bits coincidirá. Efectivamente, está diciendo que permitirá conexiones al servicio solo cuando se ofrezca desde máquinas en el rango de direcciones 192.168.24.0 a través del 192.168.27.255.

Una máscara de 0, /0, significa que no se requieren bits en la dirección para que coincidan. En otras palabras, debido a que no es necesario que coincidan los bits, usar /0 es lo mismo que no especificar una dirección. Cualquier dirección de unidifusión coincide. iptables tiene un alias integrado para 0.0.0.0/0, any/0. Tenga en cuenta que any/0, ya sea implícito o declarado, no incluye direcciones de transmisión.

Inicialización del cortafuegos

Un firewall se implementa como una serie de reglas de filtrado de paquetes definidas por opciones proporcionadas en la línea de comandos de iptables o nftables.

Las invocaciones de reglas deben realizarse desde un script de shell ejecutable, no directamente desde la línea de comando. Debe invocar el script de shell de firewall completo. No intente invocar reglas específicas desde la línea de comandos porque esto podría hacer que su firewall acepte o descarte paquetes de manera inapropiada. Cuando las cadenas se inicializan y la política de eliminación predeterminada está habilitada, todos los servicios de red se bloquean hasta que se definan los filtros de aceptación para permitir el servicio individual.

100 Capítulo 5 Creación e instalación de un cortafuegos independiente

Idealmente, debe ejecutar el script de shell desde la consola. Solo los valientes ejecutan el script de shell del firewall desde una máquina remota o desde una sesión xterm de X Window. No solo se bloquea el tráfico de la red remota, sino que también se bloquea el acceso a la interfaz de bucle invertido local utilizada por X Windows hasta que se vuelve a habilitar explícitamente el acceso a la interfaz. Idealmente, XWindows no debería estar ejecutándose o incluso instalado en un firewall. Es un ejemplo típico de software que no es necesario y se ha utilizado como medio para explotar servidores en el pasado.

Como alguien que administra computadoras Linux que son geográficamente cientos de miles de kilómetros de distancia, solo puedo activar un script de firewall desde una ubicación remota. En estos casos, es recomendable hacer dos cosas. Primero, cambie la política predeterminada a ACEPTAR para la primera o las primeras ejecuciones del script de firewall. Haga esto para depurar la sintaxis del script en sí, no las reglas. Después de que el script sea sintácticamente correcto, cambie esa política nuevamente a DROP.

Un segundo e igualmente importante consejo para trabajar con scripts de cortafuegos desde ubicaciones remotas es crear un trabajo cron para detener el firewall en algún momento en el futuro cercano. Si lo hace, le permitirá activar el cortafuegos y realizar algunas pruebas, pero también le permitirá volver a acceder a la computadora si se bloquea debido a reglas fuera de lugar (o faltantes). Por ejemplo, al depurar un script de firewall, crearé una entrada cron para desactivar el firewall cada 2 minutos. Entonces puedo ejecutar con seguridad la secuencia de comandos del cortafuegos y averiguar si he bloqueado mi sesión SSH. Si de hecho me he bloqueado, simplemente espero unos minutos a que se ejecute la secuencia de comandos del cortafuegos y lo apago, lo que me da la oportunidad de corregir la secuencia de comandos y volver a intentarlo.

Además, recuerda que los filtros del cortafuegos se aplican en el orden en que los hayas definido. Las reglas se agregan al final de su cadena en el orden en que las define. La primera regla coincidente gana. Debido a esto, las reglas del cortafuegos deben definirse en el orden correcto, desde las reglas más específicas hasta las más generales.

La inicialización del cortafuegos se utiliza para cubrir una gran cantidad de terreno, incluida la definición de constantes globales utilizadas en el script de shell, la activación de los servicios de soporte del kernel (cuando sea necesario), la eliminación de cualquier regla existente en las cadenas del cortafuegos, la definición de políticas predeterminadas para las cadenas de ENTRADA y SALIDA, volver a habilitar la interfaz de bucle invertido para el funcionamiento normal del sistema, denegar el acceso desde cualquier host o red específicos que haya decidido bloquear y definir algunas reglas básicas para proteger contra direcciones incorrectas y para proteger ciertos servicios que se ejecutan en puertos sin privilegios.

Constantes simbólicas utilizadas en los ejemplos de cortafuegos

Un script de shell de firewall es más fácil de leer y mantener si se utilizan constantes simbólicas para nombres y direcciones recurrentes. Las siguientes constantes se utilizan en los ejemplos de este capítulo o son constantes universales definidas en los estándares de red. Este ejemplo también incluye la línea de interpretación "shebang" de arriba como un recordatorio amistoso:

```
#!/bin/sh
INTERNET="eth0"
LOOPBACK_INTERFACE="lo"
IPADDR="mi.dirección.ip"

# Interfaz conectada a Internet
# Sin embargo, su sistema lo nombra
# Su dirección IP
```



```

MY_ISP="mi.isp.dirección.rango"           # Servidor ISP y rango de direcciones NOC
SUBNET_BASE="mi.subred.red"               # La dirección de red de su subred
SUBNET_BROADCAST="my.subnet.bcast" # Dirección de transmisión de su subred
LOOPBACK="127.0.0.0/8"                  # Rango de direcciones de loopback reservado
CLASE_A="10.0.0.0/8"                     # Redes privadas clase A
CLASE_B="172.16.0.0/12"                  # Redes privadas clase B
CLASE_C="192.168.0.0/16"                 # Redes privadas clase C
CLASE_D_MULTICAST="224.0.0.0/4"          # Direcciones de multidifusión de clase D
CLASS_E_RESERVED_NET="240.0.0.0/5" # Direcciones reservadas Clase E
TRANSMITIR_SRC = "0.0.0.0"               # Dirección de origen de la transmisión
BROADCAST_DEST = "255.255.255.255"       # Dirección de destino de transmisión
PRIVPORTS="0:1023"                       # Rango de puertos privilegiado y bien conocido
UNPRIVPORTS="1024:65535"                # Rango de puertos sin privilegios

```

nftables e iptables definen rangos de puertos de manera diferente. Por lo tanto, las variables de rango de puertos deben definirse de manera diferente para cada uno. Para un firewall de iptables, la siguiente declaración funciona:

```

PRIVPORTS="0:1023"                       # Rango de puertos privilegiado y bien conocido
UNPRIVPORTS="1024:65535"                 # Rango de puertos sin privilegios

```

Sin embargo, para nftables, los dos puntos deben cambiarse a un guión, como se muestra aquí:

```

PRIVPORTS="0-1023"                       # Rango de puertos privilegiado y bien conocido
UNPRIVPORTS="1024-65535"                 # Rango de puertos sin privilegios

```

Las constantes que no se enumeran aquí se definen en el contexto de las reglas específicas en las que se utilizan con. Se necesita una constante adicional para iptables o nftables. Si va a utilizar iptables, defina lo siguiente:

```

IPT="/sbin/iptables"                     # Ubicación de iptables en su sistema

```

Si va a utilizar nftables, defina lo siguiente:

```

NFT = "/usr/local/sbin/nft"              # Ubicación de nft en su sistema

```

Habilitación del soporte de monitoreo de kernel

El soporte del sistema operativo para varios tipos de verificación de paquetes a menudo se superpone con lo que puede probar el firewall. En caso de duda, busque la redundancia o la defensa en profundidad.

De los comandos que se muestran en las siguientes líneas, `icmp_echo_ignore_broadcasts` indica al núcleo que descarte mensajes de solicitud de eco ICMP dirigidos a direcciones de difusión o multidifusión. (Otra instalación, `icmp_echo_ignore_all`, descarta cualquier solicitud de eco entrante mensaje. Cabe señalar que los ISP a menudo se basan en ping para ayudar a diagnosticar problemas de la red local, y DHCP a veces se basa en la solicitud de eco para evitar la colisión de direcciones).

```

# Habilitar la protección de eco de transmisión
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

```

El enrutamiento de origen rara vez se usa legítimamente en la actualidad. Los cortafuegos suelen descartar todos los paquetes enrutados de origen. Este comando deshabilita los paquetes enrutados en origen:

```

# Deshabilitar paquetes enrutados de origen
echo "0" > /proc/sys/net/ipv4/conf/all/accept_source_route

```

Las cookies TCP SYN son un mecanismo para intentar una detección y una recuperación más rápidas de las inundaciones SYN. Este comando habilita las cookies SYN:

```
# Habilitar la protección de cookies TCP SYN
eco 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Los enrutadores adyacentes envían mensajes de redirección ICMP a los hosts. Su propósito es informar al host que hay disponible una ruta más corta. Es decir, el host y ambos enrutadores están en la misma red, y el nuevo enrutador es el enrutador al que el original enviaría el paquete como su próximo salto.

Los enrutadores generan mensajes de redirección para los hosts; los anfitriones no. Los hosts deben cumplir con los redireccionamientos y agregar la nueva puerta de enlace a su caché de ruta, excepto en los casos indicados en RFC 1122, "Requisitos para hosts de Internet: capas de comunicación".

Sección 3.2.2.2: "Un mensaje de redirección DEBE descartarse silenciosamente si la nueva dirección de puerta de enlace que especifica no está en la misma (sub)red conectada a través de la cual llegó la redirección [INTRO:2, Apéndice A], o si la fuente de la redirección no es la puerta de enlace de primer salto actual para el destino especificado (consulte la Sección 3.3.1)". Estos comandos deshabilitan redirecciones:

```
# Deshabilitar la aceptación de redirección ICMP
echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects
# No envíe mensajes de redirección
echo "0" > /proc/sys/net/ipv4/conf/all/send_redirects
```

rp_filter intenta implementar la validación de la dirección de origen como se describe en RFC 1812, "Requisitos para enrutadores IP versión 4", Sección 5.3.8. En resumen, los paquetes se descartan silenciosamente si su dirección de origen es tal que la tabla de reenvío del host no enrutaría un paquete con esa dirección de destino por la misma interfaz en la que se recibió el paquete. De acuerdo con RFC 1812, si se implementa, los enrutadores deben habilitar esta función de manera predeterminada. Esta forma de validación de direcciones a menudo no está habilitada en los enrutadores, por lo que estos comandos la deshabilitan:

```
# Soltar paquetes falsificados que llegan a una interfaz que, si se responde,
# daría como resultado que la respuesta saliera de una interfaz diferente.
para f en /proc/sys/net/ipv4/conf/* rp_filter; hacer
    echo "1" > $f
hecho
```

log_martians registra los paquetes recibidos con direcciones imposibles, como se define en RFC 1812, Sección 5.3.7. Las direcciones de origen imposible incluyen direcciones de difusión o multidifusión, direcciones en las redes 0 y 127 y el espacio reservado de Clase E. Las direcciones de destino imposible incluyen la dirección 0.0.0.0, el host 0 en cualquier red, cualquier host en la red 127 y las direcciones de Clase E.

Actualmente, el código de red de Linux busca las direcciones mencionadas anteriormente. No comprueba las direcciones de clases privadas (ni podría hacerlo sin el conocimiento de la red a la que se conectó una determinada interfaz). log_martians no afecta la verificación de la validez del paquete; simplemente afecta el registro, que se establece aquí:

```
# Registra paquetes con direcciones imposibles.
echo "1" > /proc/sys/net/ipv4/conf/all/log_martians
```

Eliminación de cualquier regla preexistente

Lo primero que debe hacer al definir un conjunto de reglas de filtrado es eliminar las reglas existentes de sus cadenas. De lo contrario, cualquier regla nueva que defina se agregará al final de las reglas existentes. Los paquetes podrían coincidir fácilmente con una regla preexistente antes de llegar al punto de la cadena que está definiendo.

La eliminación se llama *lavado* de la cadena.

Para iptables, el siguiente comando vacía todas las reglas de todas las cadenas a la vez:

```
# Eliminar cualquier regla existente de todas las cadenas
$IPT --vaciar
```

Luego, las tablas específicas se pueden vaciar usando la opción -t <table> :

```
$IPT -t natural --vaciar
$IPT -t mangle --vaciar
```

Para nftables, se debe especificar el nombre de la tabla y, al hacerlo, eliminará las reglas de todas las cadenas dentro de la tabla:

```
tabla de descarga nft <nombre de la tabla>
```

Si está utilizando la convención de nomenclatura estándar de la comunidad, tendrá una tabla de filtro y posiblemente una tabla nat que se puede vaciar con los siguientes comandos:

```
filtro de mesa de descarga nft nat de
mesa de descarga nft
```

Otras tablas definidas por el usuario deben vaciarse según lo necesite su firewall específico. Para nftables, este ciclo se puede usar para vaciar todas las cadenas en todas las tablas:

```
para i en 'tablas de lista de $ NFT | awk '{imprimir $2}'
```

hacer

```
echo "Enjuagar ${i}"
tabla de descarga $NFT ${i}
```

hecho

Un mejor método es eliminar no solo las reglas, sino también las cadenas y luego las propias tablas. Esto se logra con dos bucles for a través del script de shell nftables :

```
para i en 'tablas de lista de $ NFT | awk '{imprimir $2}'
```

hacer

```
echo "Enjuagar ${i}"
$NFT vaciar la tabla ${i} para j en
'$NFT list table ${i} | cadena grep | awk '{imprimir $2}'
```

hacer

```
echo "...Eliminando la cadena ${j} de la tabla ${i}"
$ Cadena de eliminación de NFT $ {i} $ {j}
```

hecho

```
echo "Eliminando ${i}"
$ Tabla de eliminación de NFT $ {i}
```

hecho

Vaciar las cadenas no afecta el estado de la política predeterminada actualmente en vigor.

104 Capítulo 5 Creación e instalación de un cortafuegos independiente

Para iptables, el siguiente paso sería eliminar cualquier cadena definida por el usuario. Ellos pueden ser eliminado con los siguientes comandos:

```
$IPT-X  
$IPT -t nacional -X  
$IPT -t destrozar -X
```

Con nftables, todas las tablas y cadenas están definidas por el usuario, por lo que en realidad no se aplica la misma sintaxis. En cualquier caso, el vaciado de las cadenas no afecta el estado de la política predeterminada actualmente en vigor.

En este punto, tiene un script básico que define algunas variables y borra tablas y cadenas, si se han definido.

Restablecimiento de políticas predeterminadas y detención del firewall

Hasta ahora, el cortafuegos ha establecido algunos valores predeterminados que se pueden usar independientemente del estado del cortafuegos del filtro de red. Antes de establecer las políticas predeterminadas en DROP, primero restableceré las políticas predeterminadas en ACCEPT. Esto es útil para detener el cortafuegos por completo, como verá en breve. Estas líneas establecen la política predeterminada:

```
# Restablecer la política predeterminada  
$IPT --política ENTRADA ACEPTADA  
$IPT --POLITICA SALIDA ACEPTADA  
$IPT --política ADELANTE ACEPTAR  
$IPT -t nat --policy ACEPTACIÓN DE ENRUTAMIENTO PREVIO  
$IPT -t nat --policy SALIDA ACEPTAR  
$IPT -t nat --policy ACEPTAR POSTERIOR ENRUTAMIENTO  
$IPT -t mangle --policy ACEPTACIÓN DE ENRUTAMIENTO PREVIO  
$IPT -t mangle --policy SALIDA ACEPTAR
```

Con nftables no existe una política predeterminada para una cadena en el mismo sentido que existe para iptables, y las cadenas y tablas ya se han eliminado. El efecto de esto es establecer la política en ACEPTAR ya que no hay un firewall en ejecución. Por lo tanto, no hay nada que hacer por nftables aquí.

Aquí hay una adición final a lo que considero que es el comienzo de la secuencia de comandos del cortafuegos, es decir, el código para permitir que el cortafuegos se detenga fácilmente. Con este código colocado debajo del código anterior, cuando llame a la secuencia de comandos con un argumento de "detener", la secuencia de comandos vaciará, borrará y restablecerá las políticas predeterminadas y el firewall se detendrá de manera efectiva:

```
si ["$1" = "parar" ]  
luego  
echo "¡Firewall completamente detenido! ADVERTENCIA: ESTE HOST NO TIENE FIREWALL EN FUNCIONAMIENTO."  
salida 0  
ser
```

Antes de continuar con la configuración de nftables, es necesario volver a crear las tablas base. Esta se puede lograr usando un archivo de reglas de nftables que llamaré tablas de configuración. El contenido del archivo de reglas setup-tables es:

```
filtro de tabla {  
    entrada de cadena {  
        tipo filtro gancho entrada prioridad 0;  
    }  
}
```

```

        salida en cadena {
            tipo filtro gancho salida prioridad 0;
        }
    }
}

```

Este archivo se carga con el siguiente comando. Este comando debe agregarse a la secuencia de comandos del firewall después del condicional para detener el firewall:

```
$NFT -f tablas de configuración
```

Habilitación de la interfaz de bucle invertido

Debe habilitar el tráfico de bucle invertido sin restricciones. Esto le permite ejecutar los servicios basados en la red local que elija, o de los que dependa el sistema, sin tener que preocuparse por especificar todas las reglas del firewall.

Los servicios locales se basan en la interfaz de red de bucle invertido. Una vez que se inicia el sistema, la política predeterminada del sistema es aceptar todos los paquetes. El lavado de cualquier cadena preexistente no tiene ningún efecto. Sin embargo, si el cortafuegos se está reiniciando y había utilizado previamente una política de denegación predeterminada, la política de descartar aún estaría en vigor. Sin ninguna regla de firewall de aceptación, la interfaz de bucle invertido aún sería inaccesible.

Debido a que la interfaz de loopback es una interfaz interna local, el firewall puede permitir loops tráfico de regreso inmediatamente. Estos son los comandos para el script de iptables :

```

# Tráfico ilimitado en la interfaz loopback
$IPT -A ENTRADA -i lo -j ACEPTAR
$IPT -A SALIDA -o lo -j ACEPTAR

```

Para nftables, los comandos se parecen a los que se muestran después de este texto. Los comandos se pueden agregar al script principal rc.firewall que está creando o se pueden agregar a un archivo de reglas de política de host local. Agregar las reglas a un archivo de política de host local separado tiene el siguiente aspecto. El archivo asume que las reglas contenidas en las tablas de configuración

El archivo (mostrado anteriormente) ya se ha agregado al firewall. Si no se ha agregado el archivo de reglas setup-tables , no se realizará ningún procesamiento.

El archivo de política localhost contiene lo siguiente:

```

filtro de tabla {
    entrada de cadena {
        iifname lo acepto
    }

    salida en cadena {
        oifname para aceptar
    }
}

```

Luego, este archivo se carga agregando el siguiente comando a rc.firewall:

```
$NFT -f política de host local
```

Alternativamente, si lo agregará al script rc.firewall , las siguientes dos líneas harán el trabajo:

```

$ NFT agregar entrada de filtro de regla iifname para aceptar
$ NFT agregar salida de filtro de regla oifname para aceptar

```

Definición de la política predeterminada

De forma predeterminada, desea que el cortafuegos descarte todo. Las dos opciones disponibles para las cadenas integradas en iptables son ACCEPT y DROP. REJECT no es una política legal en iptables para una cadena, pero puede usarse como objetivo, como ha visto antes. A las cadenas definidas por el usuario y las cadenas nftables no se les pueden asignar políticas predeterminadas.

Usar una política predeterminada de DROP, a menos que se defina una regla para permitir o rechaza un paquete coincidente, los paquetes se descartan silenciosamente. Lo que probablemente desee es descartar silenciosamente los paquetes entrantes no deseados, pero rechazar los paquetes salientes y devolver un mensaje de error ICMP al remitente local. La diferencia para el usuario final es que, por ejemplo, si alguien en un sitio remoto intenta conectarse a su servidor web, el navegador de esa persona se bloquea hasta que su sistema devuelve una condición de tiempo de espera de TCP. No hay ninguna indicación de si existe su sitio o su servidor web. Por otro lado, si intenta conectarse a un servidor web remoto, su navegador recibe una condición de error inmediata que indica que la operación no está permitida:

```
# Establecer la política predeterminada para descartar
$IPT --política ENTRADA DROP
$IPT --política CAÍDA DE SALIDA
$IPT --política FORWARD DROP
```

Como se indicó anteriormente, no hay políticas predeterminadas para una cadena en nftables. Se puede establecer un valor predeterminado al final de la cadena dada para nftables.

Es importante tener en cuenta que, en este punto, se bloquea todo el tráfico de red que no sea el tráfico de bucle invertido local. Si está trabajando en este cortafuegos a través de la red, su conexión ya no estará activa y es posible que no pueda acceder a la computadora en la que se está construyendo el cortafuegos.

Reglas de política predeterminadas y la primera regla coincidente gana

Dentro de iptables, las políticas predeterminadas parecen ser excepciones al escenario de victorias de la regla de la primera coincidencia. Los comandos de política predeterminados no dependen de la posición. No son reglas, per se. La política predeterminada de una cadena se aplica después de que un paquete se haya comparado con cada regla de la cadena sin coincidencias. Esto es notablemente diferente para nftables donde la primera regla coincidente siempre gana y no hay una política predeterminada.

Para iptables, las políticas predeterminadas se definen primero en la secuencia de comandos para definir la disposición de paquetes predeterminada antes de que se definan reglas contrarias. Si los comandos de la política se ejecutaron al final de la secuencia de comandos, y si la secuencia de comandos del cortafuegos contenía un error de sintaxis que hizo que se cerrara prematuramente, la política predeterminada de aceptar todo podría estar en vigor. Si un paquete no coincidía con una regla (y las reglas suelen ser reglas de aceptación en un cortafuegos que niega todo de forma predeterminada), el paquete caería al final de la cadena y se aceptaría de forma predeterminada. Las reglas del firewall no estarían logrando nada útil.

Para nftables, se puede agregar una regla de eliminación para el tráfico entrante al final de la cadena y se puede agregar una regla de rechazo al final de la cadena de filtro de SALIDA. Esto tendrá el mismo efecto general que las políticas predeterminadas de iptables. Pero es importante tener en cuenta que estas reglas deben agregarse al final de la secuencia de comandos del firewall y solo después de que se hayan creado otras reglas para permitir el tráfico por encima de ellas. De lo contrario, todo el tráfico se eliminará o rechazará desde la computadora donde se ejecuta el firewall, incluida posiblemente su sesión SSH para configurar el firewall.

Uso del estado de conexión para omitir la verificación de reglas

Especificar la coincidencia de estado para intercambios previamente iniciados y aceptados le permite omitir las pruebas de firewall para el intercambio en curso. Sin embargo, la solicitud inicial del cliente permanece controlada por los filtros específicos del servicio.

Tenga en cuenta que tanto los filtros de ENTRADA como los de SALIDA son necesarios para eludir las reglas en ambas direcciones. El módulo de estado no trata una conexión como un intercambio bidireccional y no se genera una regla dinámica simétrica.

Debido a que el módulo de estado puede requerir más RAM que las máquinas de firewall Linux más antiguas, los ejemplos de iptables desarrollados en este capítulo generalmente brindan las reglas para ambas alternativas, con y sin el módulo de estado. Las reglas de nftables asumen el uso del módulo de estado de conexión, ya que nftables normalmente se ejecutaría desde computadoras más nuevas.

Incluyendo reglas de iptables estáticas y dinámicas

Los límites de recursos en términos de escalabilidad y tiempos de espera de la tabla de estado pueden requerir que se utilicen tanto las reglas estáticas como las dinámicas. El límite superior es un punto de venta con grandes firewalls comerciales.

El problema de la escalabilidad surge en los grandes firewalls diseñados para manejar de 50 000 a 100 000 conexiones simultáneas, eso es mucho estado. Los recursos del sistema se agotan en algún momento y no se puede realizar el seguimiento de la conexión. O se debe interrumpir la nueva conexión o el software debe volver al modo sin estado.

También está el tema de los tiempos de espera. El estado de la conexión no se mantiene para siempre. Las conexiones lentas e inactivas pueden borrar fácilmente su información de estado para dejar espacio para otras conexiones más activas. Cuando llega un paquete más tarde, la información de estado debe reconstruirse. Mientras tanto, el flujo de paquetes tiene que volver al modo sin estado mientras la pila de transporte busca la información de conexión e informa al módulo de estado que el paquete forma parte de un intercambio establecido:

```
$IPT -A ENTRADA -m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
$IPT -A SALIDA -m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
# Usando el módulo de estado solo, INVALID romperá los protocolos que usan
# conexiones bidireccionales o múltiples conexiones o centrales,
# a menos que se proporcione un ALG para el protocolo.

$IPT -A INPUT -m estado --estado NO VÁLIDO -j REGISTRO \
--log-prefix "Entrada NO VÁLIDA:"
$IPT -A INPUT -m estado --estado NO VÁLIDO -j DROP

$IPT -A SALIDA -m estado --estado NO VÁLIDO -j REGISTRO \
--log-prefix "Salida NO VÁLIDA:"
$IPT -A SALIDA -m estado --estado NO VÁLIDO -j DROP
```

Para nftables, se agregan las siguientes reglas al script de firewall:

```
$NFT agregar regla filtro entrada ct estado establecido, relacionado aceptar
```

```
$NFT agregar regla filtrar entrada ct estado prefijo de registro no válido "Entrada NO VÁLIDA: " límite velocidad 3/segundo caída
```

```
$NFT agregar regla filtro salida ct estado establecido, relacionado aceptar
```

```
$NFT añadir filtro de regla salida ct estado prefijo de registro no válido "salida NO VÁLIDA: " y tasa de límite 3/segundo caída
```

Suplantación de direcciones de origen y otras direcciones incorrectas

Esta sección establece algunos filtros de cadena de ENTRADA basados en direcciones de origen y destino.

Estas direcciones nunca se verán en un paquete entrante legítimo de Internet.

En el nivel de filtrado de paquetes, uno de los pocos casos de falsificación de direcciones de origen que puede identificar con certeza como una falsificación es su propia dirección IP. Esta regla elimina los paquetes entrantes que afirman ser suyos:

```
# Rechazar paquetes falsificados que fingen ser de
# la dirección IP de la interfaz externa
$IPT -A ENTRADA -i $INTERNET -s $IPADDR -j DROP
```

La regla es similar para nftables porque aprovecha las variables definidas dentro del script de shell en lugar de las reglas nativas de nftables :

```
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $IPADDR
```

No es necesario bloquear los paquetes salientes destinados a usted mismo. No regresarán, alegando ser de usted y aparentando ser falsificados. Recuerde, si envía paquetes a su propia interfaz externa, esos paquetes llegan a la cola de entrada de la interfaz loopback, no a la cola de entrada de la interfaz externa. Los paquetes que contienen su dirección como dirección de origen nunca llegan a la interfaz externa, incluso si envía paquetes a la interfaz externa.

Registro de cortafuegos

El objetivo -j LOG habilita el registro de paquetes que coinciden con la regla. Cuando un paquete coincide con la regla, el evento se registra en `/var/log/messages`, o donde haya definido que se registren los mensajes de la prioridad especificada.

Como se explica en el Capítulo 1 y el Capítulo 2, las direcciones IP privadas se reservan en cada uno de los rangos de direcciones Clase A, B y C para usar en LAN privadas. No están destinados para su uso en Internet. Se supone que los enrutadores no deben enrutar paquetes con direcciones de origen privadas. Sin embargo, algunos enrutadores reenvían paquetes que contienen direcciones de origen privadas por error.

Además, si alguien en la subred de su ISP (es decir, en su lado del enrutador que comparte) está filtrando paquetes con direcciones IP privadas, los verá incluso si el enrutador no los reenvía. Las máquinas en su propia LAN también podrían filtrar direcciones privadas si su configuración NAT o proxy está configurada incorrectamente.

Los siguientes tres conjuntos de reglas no permiten paquetes entrantes que contengan direcciones de origen de cualquiera de las direcciones de red privada Clase A, B o C. Ninguno de estos paquetes debe verse en una red pública:

```
# Rechazar paquetes que afirman ser de una red privada Clase A
$IPT -A ENTRADA -i $INTERNET -s $CLASE_A -j DROP
```



```
# Rechazar paquetes que afirman ser de una red privada Clase B
$IPT -A ENTRADA -i $INTERNET -s $CLASE_B -j DROP
```

```
# Rechazar paquetes que afirman ser de una red privada Clase C
$IPT -A ENTRADA -i $INTERNET -s $CLASE_C -j DROP
```

La siguiente regla no permite paquetes con una dirección de origen en la red de bucle invertido:

```
# Rechazar paquetes que afirman ser de la interfaz loopback
$IPT -A ENTRADA -i $INTERNET -s $LOOPBACK -j DROP
```

El equivalente de nft se ve similar:

```
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $CLASS_A drop
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $CLASS_B drop
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $CLASS_C drop
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $LOOPBACK drop
```

Debido a que las direcciones de bucle invertido se asignan a una interfaz de software local interna, cualquier paquete que afirme ser de dicha dirección se falsifica intencionalmente.

Al igual que con las direcciones reservadas para su uso en LAN privadas, se supone que los enrutadores no deben reenviar paquetes que se originan en el rango de direcciones de loopback. Un enrutador no puede reenviar un paquete con una dirección de destino de bucle invertido.

Las siguientes dos reglas sirven principalmente para registrar paquetes coincidentes. La política predeterminada del cortafuegos es denegar todo. Como tal, las direcciones de transmisión se eliminan de forma predeterminada y deben habilitarse explícitamente si se desean:

```
# Rechazar paquetes de difusión mal formados
$IPT -A ENTRADA -i $INTERNET -s $BROADCAST_DEST -j REGISTRO
$IPT -A ENTRADA -i $INTERNET -s $BROADCAST_DEST -j DROP
```

```
$IPT -A ENTRADA -i $INTERNET -d $BROADCAST_SRC -j REGISTRO
$IPT -A ENTRADA -i $INTERNET -d $BROADCAST_SRC -j DROP
```

El primer par de reglas registra y niega cualquier paquete que afirme provenir de 255.255.255.255, la dirección reservada como dirección de destino de transmisión. Un paquete nunca se originará legítimamente en la dirección 255.255.255.255.

El segundo par de reglas registra y niega cualquier paquete dirigido a la dirección de destino 0.0.0.0, la dirección reservada como dirección de origen de transmisión. Tal paquete no es un error; es una sonda específica destinada a identificar una máquina UNIX que ejecuta software de red derivado de BSD. Debido a que la mayor parte del código de red del sistema operativo UNIX se deriva de BSD, esta sonda está diseñada para identificar máquinas que ejecutan UNIX.

El equivalente de nftables se ve similar; observe cómo las declaraciones de registro y eliminación aparecen en la misma regla con nftables:

```
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $BROADCAST_DEST límite de registro
yvelocidad 3/segundo caída
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $BROADCAST_SRC límite de registro
yvelocidad 3/segundo caída
```

Aclaración del significado de la dirección IP 0.0.0.0

La dirección 0.0.0.0 está reservada para su uso como dirección de fuente de transmisión. los

La convención de Netfilter de especificar una coincidencia en cualquier dirección, cualquier /0, 0.0.0.0/0 o 0.0.0.0/0.0.0.0, no coincide con la dirección de origen de la transmisión. El motivo es que un paquete de difusión tiene un bit establecido en el encabezado de trama de Capa 2 que indica que es un paquete de difusión destinado a todas las interfaces de la red, en lugar de un paquete de unidifusión punto a punto destinado a un destino particular. Los paquetes de difusión se manejan de manera diferente a los paquetes que no son de difusión. No hay una dirección IP legítima que no sea de transmisión 0.0.0.0.

Las siguientes dos reglas bloquean dos formas de transmisiones dirigidas:

```
# Rechazar transmisiones dirigidas
# Se utiliza para mapear redes y en ataques de denegación de servicio
$IPT -A ENTRADA -i $INTERNET -d $SUBNET_BASE -j DROP
$IPT -A ENTRADA -i $INTERNET -d $SUBNET_BROADCAST -j DROP
```

Las reglas de nftables se ven así:

```
$NFT agregar entrada de filtro de regla si $INTERNET ip daddr $SUBNET_BASE drop
$NFT agregar entrada de filtro de regla si $INTERNET ip daddr $SUBNET_BROADCAST soltar
```

Con la política de denegación predeterminada y las reglas del cortafuegos que aceptan explícitamente paquetes basados en parte en la coincidencia de la dirección de destino, el cortafuegos no aceptará ninguno de estos mensajes de difusión dirigidos. Estas reglas se vuelven más críticas en configuraciones más grandes en las que la LAN usa direcciones del mundo real.

Con el uso de prefijos de red de longitud variable, la red de un sitio y los campos de host pueden o no caer en un límite de bytes. En aras de la simplicidad, SUBNET_BASE es su dirección de red, como 192.168.1.0. SUBNET_BROADCAST es la dirección de difusión de su red, como en 192.168.1.255 .

Al igual que con los mensajes de difusión dirigidos, las difusiones limitadas, confinadas a su segmento de red local, tampoco se aceptan con la política de denegación predeterminada y las reglas del firewall que aceptan explícitamente paquetes basados en parte en la coincidencia de la dirección de destino. Nuevamente, la siguiente regla se vuelve más crítica en configuraciones más grandes en las que la LAN usa direcciones del mundo real:

```
# Rechazar transmisiones limitadas
$IPT -A ENTRADA -i $INTERNET -d $BROADCAST_DEST -j DROP
```

La regla de nftables se ve así:

```
$NFT agregar entrada de filtro de regla iif $INTERNET ip daddr $BROADCAST_DEST drop
```

Cabe señalar que se debe hacer una excepción en capítulos posteriores para los clientes DHCP. Las direcciones de origen y destino de difusión se utilizan inicialmente entre el cliente DHCP y los puertos del servidor.

Las direcciones de multidifusión son legales solo como direcciones de destino. La siguiente regla cae falsificada paquetes de red de multidifusión:

```
# Rechazar direcciones de multidifusión Clase D
# Ilegal como dirección de origen
$IPT -A ENTRADA -i $INTERNET -s $CLASS_D_MULTICAST -j DROP
```

Aquí está la regla de nftables para lo mismo:

```
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $CLASS_D_MULTICAST soltar
```

Los paquetes de multidifusión legítimos son siempre paquetes UDP. Como tal, los mensajes de multidifusión son enviado punto a punto, al igual que cualquier otro mensaje UDP. La diferencia entre los paquetes de unidifusión y multidifusión es la clase de dirección de destino utilizada (y el indicador de protocolo que se incluye en el encabezado de Ethernet). La siguiente regla niega los paquetes de multidifusión que llevan un protocolo que no es UDP:

```
$IPT -A ENTRADA -i $INTERNET ! -p udp -d $CLASS_D_MULTICAST -j DROP
```

La versión nftables de este comando se ve así:

```
$NFT agregar entrada de filtro de regla si $INTERNET ip daddr $CLASS_D_MULTICAST ip protocol != udp drop
```

La funcionalidad de multidifusión es una opción configurable cuando compila el kernel y su La tarjeta de interfaz de red se puede inicializar para reconocer direcciones de multidifusión. La funcionalidad está habilitada de forma predeterminada en el kernel predeterminado de muchas distribuciones más nuevas de Linux. Es posible que desee habilitar estas direcciones si se suscribe a un servicio de conferencias en red que proporciona transmisiones de video y audio de multidifusión. (La multidifusión también se usa a veces en la red local para el descubrimiento de recursos globales, como con DHCP o enrutamiento).

Por lo general, no verá las direcciones de destino de multidifusión a menos que se haya registrado usted mismo. como destinatario. Los paquetes de multidifusión se envían a objetivos múltiples, pero específicos, por acuerdo previo. Sin embargo, he visto paquetes de multidifusión enviados desde máquinas en la subred local de mi ISP. La política predeterminada elimina los paquetes de multidifusión, incluso si se ha registrado como destinatario. Debe definir una regla para aceptar la dirección de multidifusión. La siguiente regla permite la entrada de paquetes de multidifusión en aras de la integridad:

```
$IPT -A ENTRADA -i $INTERNET -p udp -d $CLASE_D_MULTICAST -j ACEPTAR
```

Y aquí está la versión de nftables :

```
$NFT agregar entrada de filtro de regla iif $INTERNET ip daddr $CLASS_D_MULTICAST ip protocol ==udp accept
```

El enrutamiento y el registro de multidifusión es un proceso complicado administrado por su propio protocolo de control de capa IP, el Protocolo de administración de grupos de Internet (IGMP, protocolo 2). Para obtener más información sobre la comunicación de multidifusión, consulte el "CÓMO de multidifusión sobre TCP/IP" en <http://www.tldp.org/HOWTO/Multicast-HOWTO.html>. Los recursos adicionales incluyen RFC 1458, "Requisitos para protocolos de multidifusión"; RFC 1112, "Extensiones de host para multidifusión IP" (actualizado por RFC 2236, "Protocolo de administración de grupos de Internet, versión 2"); y RFC 2588, "IP Multicast y Firewalls".

Las direcciones IP de clase D van desde 224.0.0.0 hasta 239.255.255.255. La CLASE_D_ La constante MULTICAST , 224.0.0.0/4, se define para que coincida con los primeros 4 bits de la dirección.

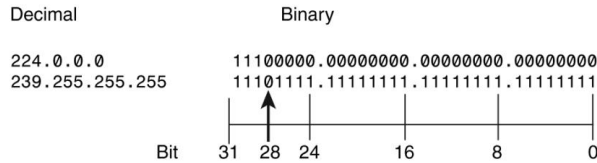


Figura 5.2 Los primeros 4 bits coincidentes en el rango de direcciones de multidifusión Clase D enmascaradas

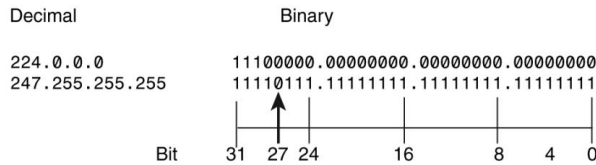


Figura 5.3 Los primeros 5 bits coincidentes en el rango de direcciones reservadas Clase E enmascaradas 240.0.0.0/5

Como se muestra en la Figura 5.2, en binario, los valores decimales 224 (11100000B) a 239 (11101111B) son idénticos en los primeros 4 bits (1110B).

La siguiente regla en esta sección elimina los paquetes que afirman ser de una red reservada Clase E:

```
# Rechazar direcciones IP reservadas de Clase E
$IPT -A ENTRADA -i $INTERNET -s $CLASS_E_RESERVED_NET -j DROP
```

El equivalente de nftables es

```
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $CLASS_E_RESERVED_NET drop
```

Las direcciones IP de clase E van desde 240.0.0.0 hasta 247.255.255.255. La constante `CLASS_E_RESERVED_NET`, `240.0.0.0/5`, se define para coincidir con los primeros 5 bits de la dirección. Como se muestra en la Figura 5.3, en binario, los valores decimales 240 (11110000B) a 247 (11110111B) son idénticos en los primeros 5 bits (11110B).

La IANA gestiona en última instancia la asignación y el registro de las direcciones IP del mundo. espacio. Para obtener más información sobre las asignaciones de direcciones IP, consulte <http://www.iana.org/assigns/ipv4-address-space/ipv4-address-space.xhtml>. Algunos bloques de direcciones están definidos como reservados por la IANA. Estas direcciones no deben aparecer en la Internet pública.

Protección de servicios en asignados Puertos sin privilegios

Los servicios destinados al uso local o privado, en particular, a menudo se ejecutan en puertos no privilegiados. Para los servicios basados en TCP, un intento de conexión a uno de estos servicios puede ser

se distingue de una conexión en curso con un cliente que utiliza uno de estos puertos sin privilegios a través del estado de los bits SYN y ACK. Bloquear las solicitudes de conexión es suficiente.

Los servicios basados en UDP simplemente deben bloquearse a menos que se use el módulo de estado.

Debe bloquear los intentos de conexión entrantes a estos puertos para su propia protección. Desea bloquear los intentos de conexión salientes para protegerse a sí mismo y a los demás de errores de su parte y para registrar posibles problemas de seguridad interna. Es más seguro bloquear estos puertos en todos los ámbitos y enrutar el tráfico relacionado de manera excepcional, caso por caso.

Asignaciones de número de puerto de servicio oficial

Los números de puerto son asignados y registrados por la IANA. La información se mantuvo originalmente como RFC 1700, "Números asignados". Ese RFC ahora está obsoleto. La IANA mantiene dinámicamente la información oficial en <http://www.iana.org/assignments/port-numbers>.

¿De qué tipo de errores podría necesitar protección? El peor error es ofrecer servicios peligrosos al mundo, ya sea sin darse cuenta o intencionalmente. Un error común es ejecutar servicios de red local que se filtran a Internet y molestan a otras personas.

Otro es permitir el tráfico saliente cuestionable, como escaneos de puertos, ya sea que este tráfico se genere por accidente o sea enviado intencionalmente por alguien en su máquina. Una política de firewall denegar todo por defecto lo protege de muchos errores de este tipo.

El problema con los escaneos de puertos

Los escaneos de puertos no son dañinos en sí mismos. Son generados por herramientas de análisis de red. El problema con los escaneos de puertos hoy en día es que generalmente son generados por personas con intenciones menos que honorables. Están "analizando" su red, no la suya propia. Desafortunadamente, esto deja a los meramente curiosos luciendo culpables también.

Una política de firewall denegar todo de forma predeterminada le permite ejecutar muchos servicios privados detrás del firewall sin riesgo indebido. Estos servicios deben permitirse explícitamente a través del firewall para que sean accesibles a los hosts remotos. Sin embargo, esta generalización es sólo una aproximación a la realidad. Si bien los servicios TCP en puertos privilegiados están razonablemente a salvo de todos menos de un hacker hábil y decidido, los servicios UDP son inherentemente menos seguros y algunos servicios están asignados para ejecutarse en puertos no privilegiados. Los servicios RPC, que generalmente se ejecutan sobre UDP, son aún más problemáticos. Los servicios basados en RPC están vinculados a algún puerto, a menudo un puerto sin privilegios. El daemon portmap mapea entre el número de servicio RPC y el número de puerto real. Un escaneo de puertos puede mostrar dónde están vinculados estos servicios basados en RPC sin pasar por el daemon portmap. Afortunadamente, el uso de portmap se está volviendo cada vez menos común, por lo que esto no es tan preocupante como lo era hace algunos años.

Servicios TCP locales comunes asignados a puertos sin privilegios

Algunos servicios, generalmente servicios LAN, se ofrecen a través de un puerto no privilegiado bien conocido y registrado oficialmente. Además, algunos servicios, como FTP e IRC, utilizan más

protocolos de comunicación complejos que no se prestan bien al filtrado de paquetes. Las reglas descritas en las siguientes secciones impiden que los programas de clientes locales o remotos inicien una conexión a uno de estos puertos.

FTP es un buen ejemplo de cómo la política de denegación predeterminada no siempre es suficiente para cubrir todos los casos posibles. El protocolo FTP se trata más adelante en este capítulo. Por ahora, la idea importante es que FTP permita conexiones entre dos puertos sin privilegios. Debido a que algunos servicios escuchan en puertos sin privilegios registrados, y debido a que la solicitud de conexión entrante a estos servicios se origina en un puerto de cliente sin privilegios, las reglas que permiten FTP pueden permitir sin darse cuenta conexiones entrantes a estos otros servicios locales también. Esta situación también es un ejemplo de cómo las reglas de firewall son lógicamente jerárquicas y dependen del orden. Las reglas que protegen explícitamente un servicio local privado que se ejecuta en un puerto sin privilegios deben preceder a las reglas de FTP que permiten el acceso a todo el rango de puertos sin privilegios.

Como resultado, algunas de estas reglas parecen ser redundantes y lo serán para algunas personas. Para otras personas que ejecutan otros servicios, las siguientes reglas son necesarias para proteger los servicios privados que se ejecutan en puertos locales sin privilegios.

No permitir conexiones a puertos de servidor sin privilegios TCP comunes

Las conexiones a los servidores remotos de X Window deben realizarse a través de SSH, que admite automáticamente las conexiones de X Window. Al especificar el indicador `--syn`, que indica el SYN bit, solo se bloquea el establecimiento de la conexión con el puerto del servidor. Otras conexiones iniciadas utilizando el puerto como puerto de cliente no se ven afectadas.

La asignación de puertos de X Window comienza en el puerto 6000 con el primer servidor en ejecución. Si se ejecutan servidores adicionales, cada uno se asigna al siguiente puerto incremental. Como un sitio pequeño, probablemente ejecutará un solo servidor X, por lo que su servidor escuchará solo en el puerto 6000. El puerto 6063 suele ser el puerto asignado más alto, lo que permite que 64 administradores de ventanas X independientes se ejecuten en una sola máquina, aunque varía hasta 6255 y 6999 también se ven a veces:

```
XWINDOW_PORTS="6000:6063" # (TCP) Ventana X
```

La primera regla garantiza que ninguna conexión saliente intente acceder a X Window de forma remota. Los administradores se hacen desde su máquina:

```
# X Establecimiento de conexión de ventana
$IPT -A SALIDA -o $!INTERNET -p tcp --syn \
--destination-port $XWINDOW_PORTS -j REJECT
```

La sintaxis para los rangos de puertos es diferente para `nftables` y, por lo tanto, `XWINDOW_`. La variable `PUERTOS` debe definirse en consecuencia:

```
XWINDOW_PORTS="6000-6063"
$NFT agregar salida de filtro de regla oif $!INTERNET ct state new tcp dport $XWINDOW_PORTS ¿reject
```

La siguiente regla bloquea los intentos de conexión entrantes a su administrador de X Window.

Las conexiones locales no se ven afectadas porque las conexiones locales se realizan a través de la interfaz de bucle invertido:

```
# Ventana X: intento de conexión entrante
$IPT -A ENTRADA -i $INTERNET -p tcp --syn \
    --destination-port $XWINDOW_PORTS -j DROP
```

Aquí está la entrada de nftables :

```
$NFT agregar entrada de filtro de regla si $INTERNET ct estado nuevo tcp dport $XWINDOW_PORTS ydrop
```

Los servicios basados en TCP restantes se pueden bloquear con una sola regla mediante el uso de la extensión de coincidencia multipuerto para iptables. No es necesario bloquear las conexiones entrantes si la máquina no está ejecutando el servicio, pero es más seguro a largo plazo, en caso de que luego decida ejecutar el servicio localmente.

El sistema de archivos de red (NFS) generalmente se une al puerto UDP 2049 pero puede usar TCP. No debería estar ejecutando NFS en una máquina con firewall, pero si lo está, se deniega el acceso externo.

No se deben permitir las conexiones a los administradores de Open Window. Linux no se distribuye con el administrador de ventanas abiertas. No es necesario bloquear las conexiones entrantes al puerto 2000 .

(Este no será el caso más adelante, cuando las reglas FORWARD del cortafuegos estén protegiendo a otros hosts locales).

squid es un caché web y un servidor proxy. squid usa el puerto 3128 de forma predeterminada, pero se puede configurar para usar un puerto diferente.

La siguiente regla impide que los clientes locales inicien una solicitud de conexión a un remoto Servidor NFS, administrador de ventanas abiertas, servidor proxy SOCKS o servidor de caché web squid :

```
NFS_PORT="2049"                # (TCP) NFS
CALCETINES_PORT="1080"         # (TCP) calcetines
OPENWINDOWS_PORT="2000"        # (TCP) Ventanas abiertas
SQUID_PORT="3128"              # (TCP) calamar
# Establecer una conexión sobre TCP a NFS, OpenWindows, SOCKS o squid
```

```
$IPT -A SALIDA -o $INTERNET -p tcp \
    -m multipuerto --puerto-destino \
    $NFS_PORT,$OPENWINDOWS_PORT,$SOCKS_PORT,$SQUID_PORT \
    --syn -j RECHAZAR
```

```
$IPT -A ENTRADA -i $INTERNET -p tcp \
    -m multipuerto --puerto-destino \
    $NFS_PORT,$OPENWINDOWS_PORT,$SOCKS_PORT,$SQUID_PORT \
    --syn -j CAÍDA
```

Para nftables, las mismas variables se pueden usar y colocar en reglas que se ven así:

```
$NFT agregar salida de filtro de regla oif $INTERNET \ tcp dport \
```

```
{$NFS_PORT,$SOCKS_PORT,$OPENWINDOWS_PORT,$SQUID_PORT} \
ct estado nuevo rechazo
$NFT agregar entrada de filtro de regla si $INTERNET \
puerto tcp \
{$NFS_PORT,$SOCKS_PORT,$OPENWINDOWS_PORT,$SQUID_PORT} \
ct estado nueva caída
```

Servicios UDP locales comunes asignados a puertos sin privilegios

Las reglas del protocolo TCP se pueden manejar con mayor precisión que las reglas del protocolo UDP debido al protocolo de establecimiento de conexión de TCP. Como servicio de datagramas, UDP no tiene un estado de conexión asociado. A menos que se utilice el módulo de estado, el acceso a los servicios UDP simplemente debe bloquearse. Se hacen excepciones explícitas para acomodar DNS y cualquiera de los pocos otros servicios de Internet basados en UDP que podría usar. Afortunadamente, los servicios de Internet UDP comunes suelen ser del tipo que se utiliza entre un cliente y un servidor específico. Las reglas de filtrado a menudo pueden permitir intercambios con un host remoto específico.

NFS es el principal servicio UDP de UNIX que debe tener en cuenta y también es uno de los más frecuentemente explotada. NFS se ejecuta en el puerto 2049 sin privilegios. A diferencia de los servicios anteriores basados en TCP, NFS es principalmente un servicio basado en UDP. Puede configurarse para ejecutarse como un servicio basado en TCP, pero normalmente no lo es.

Asociado con NFS está el demonio de bloqueo RPC, lockd, para NFS. lockd se ejecuta en el puerto UDP 4045:

```
NFS_PORT="2049"                                #NFS
LOCKD_PORT="4045"                             # RPC bloqueado para NFS

# NFS y bloqueo
$IPT -A SALIDA -o $INTERNET -p udp \
    -m multipuerto --puerto-destino $NFS_PORT,$LOCKD_PORT \
    -j RECHAZAR

$IPT -A ENTRADA -i $INTERNET -p udp \
    -m multipuerto --puerto-destino $NFS_PORT,$LOCKD_PORT \
    -j SOLTAR
```

Las reglas de nftables se ven así:

```
$NFT agregar salida de filtro de regla oif $INTERNET udp dport \
    {$NFS_PORT,$LOCKD_PORT} rechazar
$NFT agregar entrada de filtro de regla si $INTERNET udp dport \
    Caída de {$NFS_PORT,$LOCKD_PORT}
```

Las tablas de protocolo de servicio TCP y UDP

El resto de este capítulo está dedicado a definir reglas para permitir el acceso a servicios específicos. La comunicación cliente/servidor, tanto para los servicios basados en TCP como en UDP, implica algún tipo de comunicación bidireccional mediante un protocolo específico del servicio. Como tal, las reglas de acceso siempre se representan como un par de E/S. El programa cliente realiza una consulta y el servidor envía una respuesta. Las reglas para un servicio determinado se clasifican como reglas de cliente o reglas de servidor. La categoría de cliente representa la comunicación requerida para que sus clientes locales accedan a servidores remotos. La categoría de servidor representa la comunicación requerida para que los clientes remotos accedan a los servicios alojados desde sus máquinas.

Los mensajes de la aplicación se encapsulan en mensajes de protocolo de transporte TCP o UDP.

Debido a que cada servicio utiliza un protocolo de aplicación específico para sí mismo, las características particulares del intercambio TCP o UDP son, hasta cierto punto, únicas para el servicio dado.

El intercambio entre el cliente y el servidor está explícitamente descrito por las reglas del cortafuegos. Parte del propósito de las reglas de firewall es garantizar la integridad del protocolo a nivel de paquete. Sin embargo, las reglas de firewall, expresadas en la sintaxis de iptables o nftables, no son especialmente legibles para los humanos. En cada una de las siguientes secciones, el protocolo de servicio en el nivel de filtrado de paquetes se presenta como una tabla de información de estado, seguida de iptables y nftables

reglas que expresan esos estados.

Cada fila de la tabla enumera un tipo de paquete involucrado en el intercambio de servicios. Se define una regla de firewall para cada tipo de paquete individual. La tabla se divide en columnas:

- Descripción contiene una breve descripción de si el paquete se origina en el cliente o el servidor, y el propósito del paquete.
- Protocolo es el protocolo de transporte en uso, TCP o UDP, o el control del protocolo IP. mensajes, ICMP.
- Dirección remota es la dirección legal, o rango de direcciones, que el paquete puede contener en el campo de dirección remota.
- Puerto remoto es el puerto legal, o rango de puertos, que el paquete puede contener en el campo de puerto remoto.
- Entrada/Salida describe la dirección del paquete, es decir, si entra en el sistema desde una ubicación remota o si va desde el sistema a una ubicación remota.
- Dirección local es la dirección legal, o rango de direcciones, que el paquete puede contener en el campo de dirección local.
- Puerto local es el puerto legal, o rango de puertos, que el paquete puede contener en el puerto local. campo del puerto.
- Los paquetes del protocolo TCP contienen una columna final, TCP Flag, que define el SYN-ACK legal. indica que el paquete puede tener.

Finalmente, en los pocos casos en que el protocolo de servicio involucra mensajes ICMP, observe que los paquetes ICMP de la capa de red IP no están asociados con el concepto de un puerto de origen o destino, como es el caso de los paquetes TCP o UDP de la capa de transporte. En su lugar, los paquetes ICMP utilizan el concepto de un tipo de mensaje de control o de estado. Los mensajes ICMP no se envían a programas vinculados a puertos de servicio particulares. En cambio, los mensajes ICMP se envían de una computadora a otra. (El paquete ICMP contiene una copia de al menos parte del paquete original que generó el mensaje de error. El host receptor identifica el proceso al que se refiere el error al examinar el paquete transportado en el área de datos del paquete ICMP).

En consecuencia, las pocas entradas de paquetes ICMP que se presentan en las tablas usan la columna del puerto de origen para contener el tipo de mensaje. Para los paquetes ICMP entrantes, la columna del puerto de origen es la columna del puerto remoto. Para los paquetes ICMP salientes, la columna del puerto de origen es la columna del puerto local.

Habilitación de los servicios de Internet básicos requeridos

Solo se requiere un servicio: el Servicio de Nombres de Dominio (DNS). DNS traduce entre nombres de host y sus direcciones IP asociadas. Por lo general, no puede ubicar un host remoto sin DNS a menos que el host esté definido localmente.

Permitir DNS (Puerto UDP/TCP 53)

DNS utiliza un protocolo de comunicación que se basa tanto en UDP como en TCP. Los modos de conexión incluyen conexiones regulares de cliente a servidor, tráfico de igual a igual entre servidores de reenvío y servidores completos, y conexiones de servidor de nombres primario y secundario.

Las solicitudes de búsqueda de consultas normalmente se realizan a través de UDP, tanto para búsquedas de cliente a servidor y para búsquedas de servidores peer-to-peer. La comunicación UDP puede fallar en una búsqueda si la información que se devuelve es demasiado grande para caber en un solo paquete DNS UDP. El servidor establece un bit indicador en el encabezado del mensaje DNS que indica que los datos están truncados. En este caso, el protocolo permite un reintento a través de TCP. La figura 5.4 muestra la relación entre UDP y TCP durante una búsqueda de DNS. En la práctica, normalmente no se necesita TCP para las consultas. TCP se usa convencionalmente para transferencias de zonas administrativas entre primaria y secundaria

servidores de nombres

Las transferencias de zona son la transferencia de la información completa de un servidor de nombres sobre una red, o la parte (zona) de una red, para la cual el servidor tiene autoridad (es decir, el servidor oficial). El servidor de nombres autorizado se denomina servidor de nombres principal.

Los servidores de nombres secundarios o de respaldo pueden solicitar periódicamente transferencias de zona desde su servidor primario para mantener sus cachés DNS actualizadas.

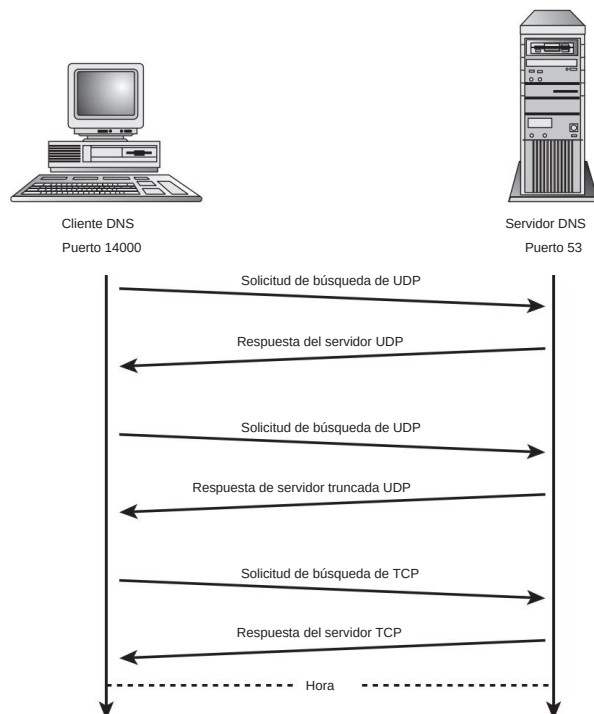


Figura 5.4 Búsqueda de cliente a servidor de DNS

Por ejemplo, uno de los servidores de nombres de su ISP es el servidor autorizado principal para el espacio de direcciones del ISP. Los ISP a menudo tienen varios servidores DNS para equilibrar la carga, así como para la redundancia de copias de seguridad. Los demás servidores de nombres son servidores de nombres secundarios y actualizan su información desde la copia maestra en el servidor principal.

Las transferencias de zona requieren un cuidadoso control de acceso entre los servidores principal y secundario. No es probable que un sistema pequeño sea un servidor de nombres autorizado para el espacio de nombres de un dominio público, ni es probable que sea un servidor de respaldo público para esa información. Los sitios más grandes podrían albergar fácilmente servidores primarios y secundarios. Se debe tener cuidado de que las transferencias de zona solo se permitan entre estos hosts. Numerosos ataques han tenido éxito porque el atacante pudo obtener una copia de una zona DNS completa y conocer la topología de la red para dirigir el ataque a los activos más valiosos.

La Tabla 5.1 enumera el protocolo DNS completo que tienen en cuenta las reglas del cortafuegos.

Tabla 5.1 Protocolo DNS

Descripción	Protocolo	Remoto	Local		TCP
		Habla a	Entrada/salida de puerto remoto	Dirección Puerto local	Bandera
Consulta de cliente local	UDP	SERVIDOR DE NOMBRES 53	Salida IPADDR 1024:65535 —		
Servidor remoto					
respuesta	UDP	SERVIDOR DE NOMBRES 53	En	IPADDR 1024:65535 —	
Consulta de cliente local	TCP	SERVIDOR DE NOMBRES 53	Salida IPADDR 1024:65535 Cualquiera		
Servidor remoto					
respuesta	TCP	SERVIDOR DE NOMBRES 53	En	IPADDR 1024:65535 RECONOCIMIENTO	
Servidor local					
consulta	UDP	SERVIDOR DE NOMBRES 53	Fuera	IPADDR 53	—
Servidor remoto					
respuesta	UDP	SERVIDOR DE NOMBRES 53	En	IPADDR 53	—
zona local					
solicitud de transferencia	TCP	Primario	53	Salida IPADDR 1024:65535 Cualquiera	
Zona remota					
solicitud de transferencia	TCP	Primario	53	En IPADDR 1024:65535 RECONOCIMIENTO	
cliente remoto					
consulta	UDP	Cliente DNS 1024:65535	En	IPADDR 53	—
Servidor local					
respuesta	UDP	Cliente DNS 1024:65535	Salida IPADDR 53		
cliente remoto					
consulta	TCP	Cliente DNS 1024:65535	En	IPADDR 53	Ninguna
Servidor local					
respuesta	UDP	cliente DNS	53	Fuera IPADDR 53	—
Zona remota					
solicitud de transferencia	TCP	Secundaria 1024:65535	En	IPADDR 53	
zona local					
respuesta de transferencia	TCP	Secundaria 1024:65535	Salida IPADDR 53		

PÓRTE DE 64

Permitir búsquedas de DNS como cliente

El cliente de resolución de DNS no es un programa específico. El cliente se incorpora al código de la biblioteca de red que utilizan los programas de red. Cuando un nombre de host requiere una búsqueda, la resolución solicita la búsqueda de un servidor DNS. La mayoría de las computadoras están configuradas solo como un cliente DNS. El servidor se ejecuta en una máquina remota. Para un usuario doméstico, el servidor de nombres suele ser una máquina propiedad de su ISP.

Como cliente, se supone que su máquina no está ejecutando un servidor DNS local; si es así, debe asegurarse de que realmente necesita ejecutar el servidor de nombres. ¡No hay necesidad de ejecutar servicios adicionales! Cada búsqueda de cliente pasa por el resolver y luego se envía a uno de los servidores de nombres remotos configurados en `/etc/resolv.conf`. En general, es mejor instalar las reglas del cliente incluso si se usa un servidor local. Evitará algunos problemas confusos que de otro modo podrían surgir en algún momento.

Estas reglas deben instalarse en las tablas del cortafuegos antes de que cualquier otra regla pueda especificar correctamente un host remoto por nombre, en lugar de por dirección IP, a menos que el host remoto tenga una entrada en el archivo local `/etc/hosts`.

DNS envía una solicitud de búsqueda como un datagrama UDP:

```
NAMESERVER ="mi.nombre.servidor"                # (TCP/UDP) DNS
si ["$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p udp \
        -s $IPADDR --deporte $UNPRIVPORTS \
        -d $SERVIDOR DE NOMBRES --dport 53 \
        -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p udp \
    -s $IPADDR --deporte $UNPRIVPORTS \
    -d $SERVIDOR DE NOMBRES --dport 53 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p udp \
    -s $SERVIDOR DE NOMBRES --deporte 53 \
    -d $IPADDR --dport $UNPRIVPORTS -j ACEPTAR
```

Las reglas de nftables se ven así:

```
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR udp sport $UNPRIVPORTS yip daddr $NAMESERVER udp dport 53 ct
state new accept
$NFT agregar entrada de filtro de regla iif $INTERNET ip daddr $IPADDR udp dport $UNPRIVPORTS yip saddr $NAMESERVER udp sport 53
acceptar
```

Si se produce un error porque los datos devueltos son demasiado grandes para caber en un datagrama UDP, el cliente DNS vuelve a intentarlo mediante una conexión TCP.

Las siguientes dos reglas se incluyen para las raras ocasiones en que la respuesta de búsqueda no encaja en un datagrama DNS UDP. No se utilizarán en las operaciones normales del día a día. Podría ejecutar su sistema sin problemas durante meses sin las reglas de TCP.

Desafortunadamente, de vez en cuando sus búsquedas de DNS se cuelgan sin estas reglas. Más típicamente, estas reglas son utilizadas por un servidor de nombres secundario que solicita una transferencia de zona desde su servidor de nombres principal:

```

si ["$CONEXIÓN_SEGUIMIENTO" = "1"]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \
        -s $IPADDR --deporte $UNPRIVPORTS \
        -d $SERVIDOR DE NOMBRES --dport 53 \
        -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \
    -s $IPADDR --deporte $UNPRIVPORTS \
    -d $SERVIDOR DE NOMBRES --dport 53 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \
    -s $SERVIDOR DE NOMBRES --deporte 53 \
    -d $IPADDR --dport $UNPRIVPORTS -j ACEPTAR

```

Las reglas de nftables se ven así:

```

$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $UNPRIVPORTS yip daddr $NAMESERVER tcp dport 53 ct
state new accept
$NFT agregar entrada de filtro de regla iif $INTERNET ip daddr $IPADDR tcp dport $UNPRIVPORTS yip saddr $NAMESERVER tcp sport 53
tcp flags != syn accept

```

Permitir sus búsquedas de DNS como un servidor de reenvío

La configuración de un servidor de nombres de reenvío local puede ser una gran ganancia de rendimiento. Como se muestra en la Figura 5.5, cuando BIND está configurado como un servidor de nombres de almacenamiento en caché y reenvío, funciona como un servidor local y como un cliente para un servidor DNS remoto. La diferencia entre un intercambio directo de cliente a servidor y un intercambio reenviado de servidor a servidor está en los puertos de origen y destino utilizados. En lugar de iniciar un intercambio desde un puerto sin privilegios, BIND inicia el intercambio desde su propio puerto DNS 53. (El puerto de origen de la consulta ahora se puede configurar. En las versiones más recientes de BIND, el servidor local realiza su solicitud desde un puerto sin privilegios, de forma predeterminada.) Una segunda diferencia es que las búsquedas de servidor de reenvío de este tipo siempre se realizan a través de UDP. (Si la respuesta es demasiado grande para caber en un paquete UDP DNS, el servidor local debe volver al comportamiento estándar de cliente/servidor para iniciar la solicitud TCP).

Uso del puerto DNS BIND

Históricamente, los servidores DNS usaban el puerto UDP 53 como puerto de origen cuando se comunicaban con otros servidores. Este distinguía el tráfico del cliente del tráfico iniciado por el servidor porque el cliente siempre usa un puerto alto y sin privilegios como origen. Las versiones posteriores de BIND permiten configurar el puerto de origen de servidor a servidor y utilizar los puertos sin privilegios de forma predeterminada. Todos los ejemplos de este libro asumen que BIND se ha configurado para usar el puerto UDP 53, en lugar de un puerto sin privilegios, para consultas de servidor a servidor.

Las solicitudes de clientes locales se envían al servidor DNS local. La primera vez, BIND no tendrá la información de búsqueda, por lo que reenvía la solicitud a un servidor de nombres remoto. BIND almacena en caché la información devuelta y la pasa al cliente. La próxima vez que se solicite la misma información, BIND la encuentra en su caché local (según el tiempo de vida [TTL] del registro) y no realiza una solicitud remota.

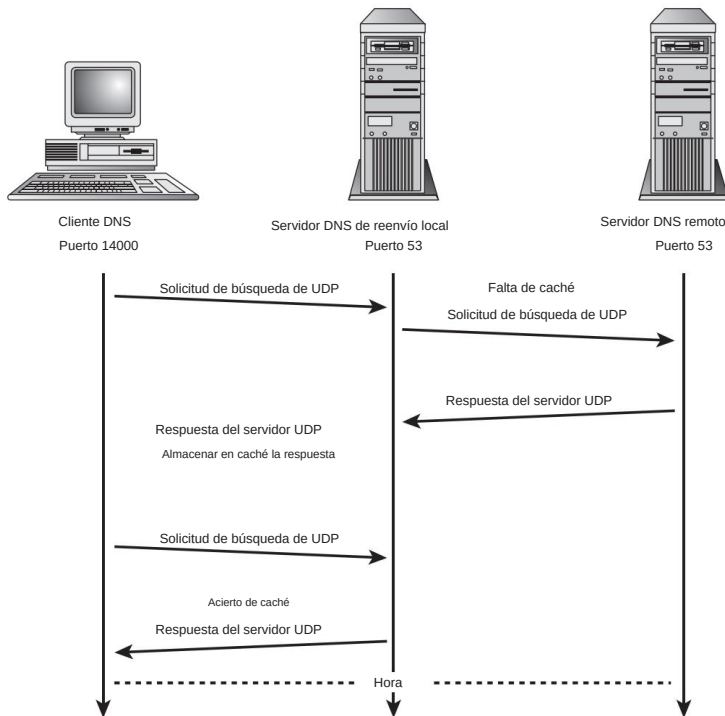


Figura 5.5 Una búsqueda de servidor de reenvío de DNS

Si la búsqueda falla debido al tamaño del paquete UDP, el servidor recurrirá a una búsqueda en modo cliente TCP. Si la búsqueda falla porque el servidor remoto no tiene la información, el servidor local consultará al servidor de caché raíz. Debido a esto, las reglas del cliente deberían permitir el tráfico de DNS a cualquier servidor, en lugar de a los servidores específicos enumerados en la configuración local.

La alternativa es configurar BIND no solo como servidor de reenvío, sino también como esclavo a los servidores remotos especificados en el archivo de configuración BIND, `named.conf`. Como esclavo, no se requieren las reglas generales de UDP del cliente.

Habilitación de servicios TCP comunes

Es probable que nadie quiera habilitar todos los servicios enumerados en esta sección, pero la mayoría querrá habilitar algún subconjunto de ellos. Estos son los servicios más utilizados en Internet en la actualidad. Como tal, esta sección es más una sección de referencia que cualquier otra cosa. Esta sección proporciona reglas para lo siguiente:

y Correo electrónico

• SSH

• FTP

• Servicio TCP genérico

Hay muchos otros servicios disponibles que no están cubiertos aquí. Algunos de ellos se utilizan en servidores especializados, algunos son utilizados por grandes empresas y organizaciones, y algunos están diseñados para su uso en redes privadas locales. Los servicios adicionales de LAN y DMZ se tratan en el Capítulo 7.

Correo electrónico (TCP SMTP Puerto 25, POP Puerto 110, IMAP Puerto 143)

El correo electrónico es un servicio que casi todo el mundo quiere. La configuración del correo depende de su ISP, su tipo de conexión y sus propias elecciones. El correo electrónico se envía a través de la red mediante el protocolo SMTP asignado al puerto de servicio TCP 25. El correo electrónico suele recibirse localmente a través de uno de los tres protocolos diferentes: SMTP, POP o IMAP, según los servicios que proporciona su ISP y su configuración local.

SMTP es el protocolo general de entrega de correo. El correo se entrega a la máquina host de destino, según lo define más comúnmente el registro MX en el DNS para el dominio dado.

El servidor de correo del extremo determina si el correo se puede entregar (dirigido a una cuenta de usuario válida en la máquina) y luego lo entrega al buzón de correo local del usuario.

POP e IMAP son servicios de recuperación de correo. POP se ejecuta en el puerto TCP 110. IMAP se ejecuta en Puerto TCP 143. Los protocolos POP e IMAP actuales normalmente se ejecutan sobre una capa de conexión segura (SSL) para el cifrado. POP/S e IMAP/S se ejecutan en los puertos 995 y 993 respectivamente.

Los ISP comúnmente ponen a disposición de sus clientes el correo entrante utilizando uno o ambos de estos dos servicios. Ambos servicios suelen estar autenticados por nombre de usuario y contraseña. En lo que respecta a la recuperación de correo, la diferencia entre SMTP y POP o IMAP es que SMTP recibe el correo entrante y lo pone en cola en el buzón de correo local del usuario. POP e IMAP recuperan correo en el programa de correo local del usuario desde el ISP del usuario, donde el correo se había puesto en cola de forma remota en el buzón SMTP del usuario en el ISP. La Tabla 5.2 enumera los protocolos de conexión cliente/servidor completos para SMTP, POP e IMAP. SMTP también utiliza mecanismos de entrega especializados que podría utilizar su red local, como ETRN, que transfiere de forma eficaz todo el correo de un dominio determinado para su procesamiento local.

Envío de correo a través de SMTP (puerto TCP 25)

El correo se envía a través de SMTP. Pero, ¿de quién es el servidor SMTP que utiliza para recopilar su correo y enviarlo? Los ISP ofrecen servicio de correo SMTP a sus clientes. El servidor de correo del ISP actúa como puerta de enlace de correo. Sabe cómo recopilar su correo, encontrar el host del destinatario y retransmitir el correo. Con UNIX, puede alojar su propio servidor de correo local, si lo desea. Su servidor será responsable de enrutar el correo a su destino.

Tabla 5.2 Protocolos de correo SMTP, POP e IMAP

Descripción	Protocolo	Remoto		Local		TCP
		Habla a	Entrada/salida de puerto remoto	Dirección	Puerto local	Bandera
Enviar correo saliente	TCP	EN CUALQUIER LUGAR	25	Salida	IPADDR 1024:65535 Cualquiera	
Servidor remoto respuesta	TCP	EN CUALQUIER LUGAR	25	En	IPADDR 1024:65535 RECONOCIMIENTO	
Recibir correo entrante	TCP	EN CUALQUIER LUGAR	1024:65535 En		IPADDR 25	Ninguna
Servidor local respuesta	TCP	EN CUALQUIER LUGAR	1024:65536 Salida	IPADDR 25		POBRE DE MÍ
Consulta de cliente local	TCP	SERVIDOR POP	110 o 995 Salida	IPADDR 1024:65535 Cualquiera		
Servidor remoto respuesta	TCP	SERVIDOR POP	110 o 995 In		IPADDR 1024:65535 RECONOCIMIENTO	
cliente remoto consulta	TCP	CLIENTE POP	1024:65535 En		IPADDR 110 o 995 Cualquiera	
Servidor local respuesta	TCP	CLIENTE POP	1024:65535 Salida	IPADDR 110 o 995 ACK		
Consulta de cliente local	TCP	SERVIDOR IMAP	143 o 993 Salida	IPADDR 1024:65535 Cualquiera		
Servidor remoto respuesta	TCP	SERVIDOR IMAP	143 o 993 En		IPADDR 1024:65535 RECONOCIMIENTO	
cliente remoto consulta	TCP	CLIENTE IMAP	1024:65535 En		IPADDR 143 o 993 Cualquiera	
Servidor local respuesta	TCP	CLIENTE IMAP	1024:65535 Salida	IPADDR 143 o 993 ACK		

Retransmisión de correo saliente a través de un servidor SMTP de puerta de enlace externo (ISP)

Cuando retransmite el correo saliente a través de un servidor de puerta de enlace de correo externo, el programa de correo de su cliente envía todo el correo saliente al servidor de correo de su ISP. Su ISP actúa como su puerta de entrada de correo al resto del mundo. Su sistema no necesita saber cómo ubicar sus destinos de correo o las rutas hacia ellos. La puerta de enlace de correo del ISP sirve como su relevo.

Las siguientes dos reglas le permiten retransmitir correo a través de la puerta de enlace SMTP de su ISP:

```
SMTP_GATEWAY="my.isp.server" if # Servidor de correo externo o retransmisión
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \
        -s $IPADDR --deporte $UNPRIVPORTS \
        -d $SMTP_GATEWAY --dport 25 -m estado --estado NUEVO -j ACEPTAR
ser
```



```
$IPT -A SALIDA -o $INTERNET -p tcp \
    -s $IPADDR --deporte $UNPRIVPORTS \
    -d $SMTP_GATEWAY --dport 25 -j ACCEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \
    -s $SMTP_GATEWAY --deporte 25 \
    -d $IPADDR --dport $UNPRIVPORTS -j ACCEPTAR
```

Los comandos de nftables se ven así:

```
$NFT agregar salida de filtro de regla oif $INTERNET ip daddr $SMTP_GATEWAY tcp dport 25 ip ysaddr $IPADDR tcp sport $UNPRIVPORTS
acceptar
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $SMTP_GATEWAY tcp sport 25 ip ydaddr $IPADDR tcp dport $UNPRIVPORTS
tcp flags != syn accept
```

Envío de correo a cualquier servidor de correo externo

Alternativamente, puede omitir el servidor de correo de su ISP y alojar el suyo propio. Su servidor local es responsable de recopilar su correo saliente, realizar la búsqueda de DNS en el nombre de host de destino y enviar el correo a su destino. El programa de correo de su cliente apunta a su servidor SMTP local en lugar de al servidor del ISP.

Las siguientes dos reglas le permiten enviar correo directamente a los destinos remotos:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \
        -s $IPADDR --deporte $UNPRIVPORTS \
        --dport 25 -m estado --estado NUEVO -j ACCEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \
    -s $IPADDR --deporte $UNPRIVPORTS \
    --dport 25 -j ACCEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \
    --deporte 25 \
    -d $IPADDR --dport $UNPRIVPORTS -j ACCEPTAR
```

Los comandos de nftables se ven así:

```
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $UNPRIVPORTS ytcp dport 25 aceptar

$NFT agregar entrada de filtro de regla iif $INTERNET ip daddr $IPADDR tcp sport 25 tcp dport y$UNPRIVPORTS tcp flags != syn accept
```

Recibiendo correo

La forma en que recibe el correo depende de su situación. Si ejecuta su propio servidor de correo local, puede recopilar el correo entrante directamente en su máquina Linux. Si recupera su correo de su cuenta de ISP, puede o no recuperar el correo como un cliente POP o IMAP, según cómo haya configurado su cuenta de correo electrónico de ISP y según los servicios de entrega de correo que ofrece el ISP.

Recepción de correo como servidor SMTP local (puerto TCP 25)

Si desea recibir correo enviado directamente a sus máquinas locales desde cualquier parte del mundo, debe ejecutar Sendmail, Gmail o algún otro programa de servidor de correo. Estas son las reglas del servidor local:

```

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A ENTRADA -i $INTERNET -p tcp \
        --deporte $UNPRIVPORTS \
        -d $IPADDR --dport 25 \
        -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A ENTRADA -i $INTERNET -p tcp \
    --deporte $UNPRIVPORTS \
    -d $IPADDR --dport 25 -j ACEPTAR

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \
    -s $IPADDR --deporte 25 \
    --dport $UNPRIVPORTS -j ACEPTAR

```

Los comandos para el script nftables se ven así:

```

$NFT agregar entrada de filtro de regla si $INTERNET tcp sport $UNPRIVPORTS ip daddr $IPADDR ytcp dport 25
aceptar
$NFT agregar salida de filtro de regla oif $INTERNET tcp sport 25 ip saddr $IPADDR tcp dport y$UNPRIVPORTS tcp
flags != syn accept

```

Alternativamente, si prefiere mantener su cuenta de correo electrónico local relativamente privada y usar su cuenta de correo electrónico del ISP o del trabajo como su dirección pública, puede configurar sus cuentas de correo del ISP y del trabajo para reenviar el correo a su servidor local. En este caso, podría reemplazar el par de reglas único anterior, aceptando conexiones desde cualquier lugar, con reglas específicas separadas para cada reenviador de correo.

Recuperación de correo como cliente POP (puerto TCP 110 o 995)

Conectarse a un servidor POP es un medio muy común de recuperar correo de un ISP remoto o una cuenta de trabajo. Si su ISP utiliza un servidor POP para la recuperación de correo del cliente, debe permitir las conexiones salientes de cliente a servidor.

La dirección del servidor será un nombre de host o una dirección específica en lugar del especificador ANYWHERE global e implícito. Las cuentas POP son cuentas de usuario asociadas a un usuario y contraseña específicos:

```

POP_SERVER="my.isp.pop.server" if # Servidor pop externo, si lo hay
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \
        -s $IPADDR --deporte $UNPRIVPORTS \
        -d $SERVIDOR_POP --dport 110 -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \
    -s $IPADDR --deporte $UNPRIVPORTS \
    -d $SERVIDOR_POP --dport 110 -j ACEPTAR

```

```
$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \
-s $SERVIDOR_POP --deporte 110 \
-d $IPADDR --dport $UNPRIVPORTS -j ACCEPTAR
```

Los comandos para nftables tienen el siguiente aspecto; sustituya 110 en lugar de 995 si su servidor de correo usa POP regular sin SSL:

```
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR ip daddr $POP_SERVER ytcp sport
$UNPRIVPORTS tcp dport 995 acceptar
$NFT agregar entrada de filtro de regla iif $INTERNET ip saddr $POP_SERVER tcp sport 110 ip ydaddr $IPADDR
tcp dport $UNPRIVPORTS tcp flags != syn accept
```

Recepción de correo como cliente IMAP (puerto TCP 143 o 993)

Conectarse a un servidor IMAP es otro medio común de recuperar correo de un ISP remoto o cuenta de trabajo. Si su ISP utiliza un servidor IMAP para la recuperación de correo del cliente, debe permitir las conexiones salientes de cliente a servidor.

La dirección del servidor será un nombre de host o una dirección específicos en lugar del especificador global implícito \$ANYWHERE . Las cuentas IMAP son cuentas de usuario asociadas a un usuario y contraseña específicos:

```
IMAP_SERVER="my.isp.imap.server" if # Servidor imap externo, si lo hay
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \
        -s $IPADDR --deporte $UNPRIVPORTS \
        -d $SERVIDOR_IMAP --dport 143 -m estado --estado NUEVO -j ACCEPTAR
```

ser

```
$IPT -A SALIDA -o $INTERNET -p tcp \
-s $IPADDR --deporte $UNPRIVPORTS \
-d $SERVIDOR_IMAP --dport 143 -j ACCEPTAR
```

```
$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \
-s $SERVIDOR_IMAP --deporte 143 \
-d $IPADDR --dport $UNPRIVPORTS -j ACCEPTAR
```

Las reglas de nftables se parecen a las siguientes; sustituya 143 en lugar de 993 si su El servidor IMAP no usa SSL:

```
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $UNPRIVPORTS yip daddr
$IMAP_SERVER tcp dport 993 acceptar
$NFT agregar entrada de filtro de regla iif $INTERNET ip saddr $IMAP_SERVER tcp sport 995 ip ydaddr $IPADDR
tcp dport $UNPRIVPORTS tcp flags != syn accept
```

Alojamiento de un servidor de correo para clientes remotos

El alojamiento de servicios POP o IMAP públicos es inusual para un sistema pequeño. Puede hacer esto si ofrece servicios de correo remoto a algunos amigos, por ejemplo, o si el servicio de correo de su ISP no está disponible temporalmente. En cualquier caso, es importante limitar los clientes de los que su sistema aceptará conexiones, tanto a nivel de filtrado de paquetes como a nivel de configuración del servidor.

Alojamiento de un servidor POP para clientes remotos

Los servidores POP son uno de los puntos de entrada más comunes y exitosos para las vulnerabilidades de piratería. Las reglas de firewall pueden ofrecer cierta cantidad de protección, en muchos casos. Por supuesto, también limitaría el acceso al nivel de configuración del servidor. Como siempre, y quizás particularmente con el software del servidor de correo, es fundamental mantenerse actualizado con las actualizaciones de seguridad del software.

Si utiliza un sistema local como servidor de correo central y ejecuta un servidor POP3 local para proporcionar acceso de correo a máquinas locales en una LAN, no necesita las reglas del servidor en este ejemplo.

Las conexiones entrantes de Internet deben interrumpirse. Si necesita alojar un servicio POP para un número limitado de personas remotas, las siguientes dos reglas permiten conexiones entrantes a su servidor POP. Las conexiones están limitadas a las direcciones IP de sus clientes específicos:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A ENTRADA -i $INTERNET -p tcp \
        -s <my.pop.clients> --sport $UNPRIVPORTS \
        -d $IPADDR --dport 110 \
        -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A ENTRADA -i $INTERNET -p tcp \
    -s <my.pop.clients> --sport $UNPRIVPORTS \
    -d $IPADDR --dport 110 -j ACEPTAR

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \
    -s $IPADDR --deporte 110 \
    -d <my.pop.clients> --dport $UNPRIVPORTS -j ACCEPT
```

Las reglas de nftables se ven así:

```
nft añadir entrada de filtro de regla iif $INTERNET ip saddr <POP_CLIENTS> tcp sport y$UNPRIVPORTS ip daddr $IPADDR tcp
dport 995 aceptar
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport 995 ip daddr y<POP_CLIENTS> tcp dport $UNPRIVPORTS
tcp flags != syn accept
```

Si su sitio fuera un ISP, podría usar el enmascaramiento de direcciones de red para limitar qué fuente direcciones de las que aceptaría solicitudes de conexión POP:

```
CLIENTES_POP="192.168.24.0/24"
```

Si el suyo es un sitio residencial con un puñado de clientes POP remotos, las direcciones del cliente tendría que indicarse explícitamente, con un par de reglas separadas para cada dirección de cliente.

Alojar un servidor IMAP para clientes remotos

Los servidores IMAP son uno de los puntos de entrada más comunes y exitosos para las vulnerabilidades de piratería. Las reglas de firewall pueden ofrecer cierta cantidad de protección, en muchos casos. Por supuesto, también limitaría el acceso al nivel de configuración del servidor. Como siempre, y quizás particularmente con el software del servidor de correo, es fundamental mantenerse actualizado con las actualizaciones de seguridad del software.

SSH (Puerto TCP 22)

Con la expiración de la patente RSA en el año 2000, OpenSSH, shell seguro, se incluye en las distribuciones de Linux. También está disponible gratuitamente en los sitios de software del

Internet. Se considera que SSH es mucho más preferible que usar telnet para el acceso de inicio de sesión remoto porque ambos extremos de la conexión usan claves de autenticación tanto para hosts como para usuarios, y porque los datos están encriptados. Además, SSH es más que un servicio de inicio de sesión remoto. Puede dirigir automáticamente conexiones X Window entre sitios remotos, y FTP y otras conexiones basadas en TCP se pueden dirigir a través de la conexión SSH más segura.

Siempre que el otro extremo de la conexión permita conexiones SSH, es posible enrutar todas las conexiones TCP a través del firewall usando SSH. Como tal, SSH es una especie de red privada virtual (VPN) de un hombre pobre.

Los puertos utilizados por SSH son altamente configurables. De forma predeterminada, las conexiones se inician entre el puerto sin privilegios de un cliente y el puerto de servicio 22 asignado del servidor. El cliente SSH utiliza exclusivamente los puertos sin privilegios. Las reglas de este ejemplo se aplican al uso del puerto SSH predeterminado:

```
SSH_PORTS="1024:65535"           # autenticación RSA
0
SSH_PORTS="1020:65535"          # Autenticación de tueste
```

Las reglas de cliente y servidor aquí permiten el acceso desde y hacia cualquier lugar. En la práctica, limitaría las direcciones externas a un subconjunto selecto, particularmente porque ambos extremos de la conexión deben configurarse para reconocer cada cuenta de usuario individual para la autenticación. La Tabla 5.3 enumera el protocolo de conexión cliente/servidor completo para el servicio SSH.

Tabla 5.3 Protocolo SSH

		Remoto		Local	TCP
Descripción	Protocolo	Habla a	Entrada/salida de puerto remoto	Dirección Puerto local	Bandera
cliente local					
solicitud	TCP	EN CUALQUIER LUGAR 22	Fuera	IPADDR 1024:65535 Cualquiera	
Servidor remoto					
respuesta	TCP	EN CUALQUIER LUGAR 22	En	IPADDR 1024:65535 RECONOCIMIENTO	
cliente local					
solicitud	TCP	EN CUALQUIER LUGAR 22	Fuera	IPADDR 513:1023	Ninguna
Servidor remoto					
respuesta	TCP	EN CUALQUIER LUGAR 22	En	IPADDR 513:1023	POBRE DE MI
cliente remoto					
solicitud	TCP	Clientes SSH 1024:65535 En		IPADDR 22	Ninguna
Servidor local					
respuesta	TCP	Clientes SSH 1024:65535 Fuera		IPADDR 22	POBRE DE MI
cliente remoto					
solicitud	TCP	Clientes SSH 513:1023		En IPADDR 22	Ninguna
Servidor local					
respuesta	TCP	Clientes SSH 513:1023		Fuera IPADDR 22	POBRE DE MI

Permitir el acceso de clientes a servidores SSH remotos

Estas reglas le permiten conectarse a sitios remotos usando SSH:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR
        --sport $SSH_PORTS \ --dport 22 -m estado
        --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR
    --sport $SSH_PORTS \ --dport 22 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \ --sport 22 \ -d
    $IPADDR --dport $SSH_PORTS -j ACEPTAR
```

Las reglas de nftables se parecen a las siguientes:

```
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $SSH_PORTS ytcp dport 22 accept
$NFT agregar entrada de filtro de regla iif $INTERNET tcp sport 22 ip daddr $IPADDR tcp dport y$SSH_PORTS tcp
flags != syn accept
```

Permitir el acceso remoto del cliente a su servidor SSH local

Estas reglas permiten conexiones entrantes a su servidor SSH:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A INPUT -i $INTERNET -p tcp \ --sport
        $SSH_PORTS \ -d $IPADDR --dport 22
        \ -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A ENTRADA -i $INTERNET -p tcp \ --sport
    $SSH_PORTS \ -d $IPADDR --dport 22
    -j ACEPTAR

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \ -s $IPADDR --
    sport 22 \ --dport $SSH_PORTS -j ACEPTAR
```

Las reglas de nftables se parecen a las siguientes:

```
$NFT agregar entrada de filtro de regla iif $INTERNET tcp sport $SSH_PORTS ip daddr $IPADDR ytcp dport 22
accept $NFT agregar entrada de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport 22 tcp dport
y$SSH_PORTS tcp flags != syn accept
```

FTP (Puertos TCP 21, 20)

FTP sigue siendo uno de los medios más comunes para transferir archivos entre dos máquinas en red. Las interfaces de navegador basadas en web para FTP también se han vuelto comunes.

Al igual que Telnet, FTP envía tanto las credenciales de autenticación como la comunicación de datos sin formato.

texto a través de la red. Por lo tanto, FTP también se considera un protocolo inherentemente inseguro. SFTP y SCP ofrecen mejoras a FTP en este sentido.

FTP se usa como el ejemplo clásico de un protocolo que no es compatible con firewall o NAT. Las aplicaciones tradicionales de cliente/servidor que se comunican a través de TCP funcionan todas de la misma manera. El cliente inicia la solicitud para conectarse al servidor.

La Tabla 5.4 enumera el protocolo de conexión cliente/servidor completo para el servicio FTP.

Tabla 5.4 Protocolo FTP

Descripción	Protocolo	Remoto	Local		TCP
		Habla a	Entrada/salida de puerto remoto	Dirección Puerto local	Bandera
Consulta de cliente local TCP		EN CUALQUIER LUGAR 21	Salida IPADDR 1024:65535	Cualquiera	
Servidor remoto					
respuesta	TCP	EN CUALQUIER LUGAR 21	En	IPADDR 1024:65535 RECONOCIMIENTO	
Servidor remoto					
canal de datos del puerto					
solicitud	TCP	EN CUALQUIER LUGAR 20	En	IPADDR 1024:65535 Cualquiera	
Canal de datos del					
puerto del cliente local					
respuesta	TCP	EN CUALQUIER LUGAR 20	Salida IPADDR 1024:65535	ACK	
cliente local					
solicitud de canal					
de datos pasivo TCP		EN CUALQUIER LUGAR 1024:65535	Salida IPADDR 1024:65535	Cualquiera	
Servidor remoto					
respuesta de					
canal de datos pasivo TCP		EN CUALQUIER LUGAR 1024:65535	En	IPADDR 1024:65535 RECONOCIMIENTO	
cliente remoto					
solicitud	TCP	EN CUALQUIER LUGAR 1024:65535	En	IPADDR 21	Ninguna
Servidor local					
respuesta	TCP	EN CUALQUIER LUGAR 1024:65535	Salida IPADDR 21		POBRE DE MI
Canal de datos del					
puerto del servidor local					
respuesta	TCP	EN CUALQUIER LUGAR 1024:65535	Salida IPADDR 20		Ninguna
cliente remoto					
canal de datos del puerto					
respuesta	TCP	EN CUALQUIER LUGAR 1024:65535	En	IPADDR 20	POBRE DE MI
cliente remoto					
solicitud de canal					
de datos pasivo TCP		EN CUALQUIER LUGAR 1024:65535	En	IPADDR 1024:65535	Cualquiera
Servidor local					
respuesta de					
canal de datos pasivo TCP		EN CUALQUIER LUGAR 1024:65535	Salida IPADDR 1024:65535	ACK	

FTP se desvía de este TCP estándar, modelo de comunicación cliente/servidor. FTP se basa en dos conexiones separadas, una para el flujo de control o comando y otra para pasar los archivos de datos y otra información, como las listas de directorios. El flujo de control se transmite a través de una conexión TCP tradicional. El cliente se vincula a un puerto alto sin privilegios y envía una solicitud de conexión al servidor FTP, que está vinculado al puerto 21. Esta conexión se utiliza para pasar comandos.

En términos de la segunda conexión de flujo de datos, FTP tiene dos modos alternativos para intercambiar datos entre un cliente y un servidor: modo puerto y modo pasivo. El modo de puerto es el mecanismo predeterminado original. El cliente le dice al servidor en qué puerto secundario sin privilegios escuchará. El servidor inicia la conexión de datos desde el puerto 20 al puerto sin privilegios que especificó el cliente.

Esta es la desviación del modelo cliente/servidor estándar. El servidor está iniciando la conexión secundaria con el cliente. Esta es la razón por la que FTP es un protocolo que requiere compatibilidad con ALG tanto para el cortafuegos como para NAT. El cortafuegos debe dar cuenta de una conexión entrante desde el puerto 20 a un puerto local sin privilegios. NAT debe tener en cuenta la dirección de destino utilizada para la conexión del flujo de datos secundario. (El cliente no tiene conocimiento de que su tráfico de red está siendo NAT. El puerto y la dirección que envió al servidor eran su puerto y dirección locales pre-NAT).

El modo pasivo es similar al modelo tradicional cliente/servidor en que el cliente inicia la conexión secundaria para el flujo de datos. Nuevamente, el cliente inicia la conexión desde un puerto alto y sin privilegios. Sin embargo, el servidor no está vinculado al puerto 20 para la conexión de datos. En cambio, el servidor le ha dicho al cliente a qué puerto alto y sin privilegios el cliente debe dirigir la solicitud de conexión. El flujo de datos se transporta entre puertos sin privilegios tanto en el cliente como en el servidor.

En términos de filtrado de paquetes tradicional, el cortafuegos debe permitir el tráfico TCP entre todos los puertos sin privilegios. El seguimiento del estado de la conexión y la compatibilidad con ALG permiten que el cortafuegos asocie la conexión secundaria con un flujo de control de FTP en particular. NAT no es un problema en el lado del cliente porque el cliente inicia ambas conexiones.

Permitir el acceso de clientes salientes a servidores FTP remotos

Es casi un hecho que la mayoría de los sitios querrán acceso de cliente FTP a repositorios de archivos remotos. La mayoría de la gente querrá habilitar las conexiones de clientes salientes a un servidor remoto.

Solicitudes FTP salientes a través del canal de control

Las siguientes dos reglas permiten una conexión de control saliente a un servidor FTP remoto:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \
        -s $IPADDR --deporte $UNPRIVPORTS \
        --dport 21 -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \
    -s $IPADDR --deporte $UNPRIVPORTS \
    --dport 21 -j ACEPTAR
```



```
$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \
--deporte 21 \
-d $IPADDR --dport $UNPRIVPORTS -j ACCEPTAR
```

Las reglas de nftables se ven así:

```
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $UNPRIVPORTS ytcp dport 21 accept
```

```
$NFT agregar entrada de filtro de regla si $INTERNET ip daddr $IPADDR tcp sport 21 tcp dport y$UNPRIVPORTS acceptar
```

Canales de datos FTP en modo puerto

Las siguientes dos reglas permiten la conexión de canal de datos estándar, en la que el servidor remoto vuelve a llamar para establecer la conexión de datos desde el puerto 20 del servidor a un puerto sin privilegios especificado por el cliente:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A ENTRADA -i $INTERNET -p tcp \
        --deporte 20 \
        -d $IPADDR --dport $UNPRIVPORTS \
        -m estado --estado NUEVO -j ACCEPTAR
```

ser

```
$IPT -A ENTRADA -i $INTERNET -p tcp \
--deporte 20 \
-d $IPADDR --dport $UNPRIVPORTS -j ACCEPTAR
```

```
$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \
-s $IPADDR --deporte $UNPRIVPORTS \
--dport 20 -j ACCEPTAR
```

Este comportamiento inusual de devolución de llamada, con el servidor remoto estableciendo la conexión secundaria con su cliente, es parte de lo que hace que FTP sea difícil de asegurar en el nivel de filtrado de paquetes. Las reglas para nftables asumen el uso del módulo de estado ct y, por lo tanto, no son necesarias.

Servicio TCP genérico

La mayoría de las reglas que se muestran en esta sección son similares entre sí. En lugar de intentar proporcionar reglas para cada tipo de servicio basado en TCP, es más útil simplemente aprender una forma general de agregar una regla según sea necesario para cualquier servicio que necesite proporcionar.

Las siguientes reglas genéricas se aplican a cualquier servicio TCP al que necesite realizar una conexión. Sustituya el puerto de servicio de destino por <SU PUERTO AQUÍ>:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \
        -s $IPADDR --deporte $UNPRIVPORTS \
        --dport <SU PUERTO AQUÍ> -m estado --estado NUEVO -j ACCEPTAR
```

ser

```
$IPT -A SALIDA -o $INTERNET -p tcp \
-s $IPADDR --deporte $UNPRIVPORTS \
--dport <SU PUERTO AQUÍ> -j ACCEPTAR
```

134 Capítulo 5 Creación e instalación de un cortafuegos independiente

```
$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \
--sport <TU PUERTO AQUÍ> \
-d $IPADDR --dport $UNPRIVPORTS -j ACEPTAR
```

Las reglas de nft se ven así:

```
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $UNPRIVPORTS ytcp dport <SU PUERTO AQUÍ> aceptar
```

```
$NFT agregar entrada de filtro de regla iif $INTERNET tcp sport <SU PUERTO AQUÍ> ip daddr y$IPADDR tcp dport $UNPRIVPORTS
aceptar
```

Las siguientes reglas se aplican para habilitar una conexión TCP entrante en cualquier puerto que sea necesario para el servicio dado:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A ENTRADA -i $INTERNET -p tcp \
        --deporte $UNPRIVPORTS \
        -d $IPADDR --dport <SU PUERTO AQUÍ> \
        -m estado --estado NUEVO -j ACEPTAR
ser
```

```
$IPT -A ENTRADA -i $INTERNET -p tcp \
--deporte $UNPRIVPORTS \
-d $IPADDR --dport <SU PUERTO AQUÍ> -j ACEPTAR
```

```
$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \
-s $IPADDR \
--dport $UNPRIVPORTS -j ACEPTAR
```

Las reglas para nftables son las siguientes:

```
nft agregar entrada de filtro de regla iif $INTERNET tcp sport $UNPRIVPORTS ip daddr $IPADDR ytcp dport <SU PUERTO AQUÍ> aceptar
```

```
nft añadir salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport <SU PUERTO Y AQUÍ> tcp dport $UNPRIVPORTS aceptar
```

Habilitación de servicios UDP comunes

El protocolo UDP sin estado es inherentemente menos seguro que el protocolo TCP basado en conexión. Debido a esto, muchos sitios conscientes de la seguridad deshabilitan por completo, o limitan tanto como sea posible, todo acceso a los servicios UDP. Obviamente, los intercambios de DNS basados en UDP son necesarios, pero los servidores de nombres remotos se pueden especificar explícitamente en las reglas del firewall. Como tal, esta sección proporciona reglas para solo dos servicios:

- Protocolo de configuración dinámica de host (DHCP)
- Protocolo de tiempo de red (NTP)

Acceso al servidor DHCP de su ISP (Puertos UDP 67, 68)

Los intercambios de DHCP, si los hay, entre su sitio y el servidor de su ISP serán necesariamente intercambios de cliente local a servidor remoto. En la mayoría de los casos, los clientes DHCP reciben direcciones IP temporales o semipermanentes asignadas dinámicamente desde un servidor central que administra los ISP.

espacio de direcciones IP del cliente. El servidor también suele proporcionar a su host local otra información de configuración, como la máscara de subred de la red; la red MTU; las direcciones predeterminadas del enrutador de primer salto; el nombre de dominio; y el TTL predeterminado.

Si tiene una dirección IP asignada dinámicamente desde su ISP, necesita ejecutar un demonio de cliente DHCP en su máquina.

La Tabla 5.5 enumera las descripciones de los tipos de mensajes DHCP, según se cita en RFC 2131, "Protocolo de configuración dinámica de host".

En esencia, cuando el cliente DHCP se inicializa, transmite una consulta DHCPDISCOVER para descubrir si hay algún servidor DHCP disponible. Cualquier servidor que reciba la consulta puede responder con un mensaje DHCPOFFER indicando su voluntad de funcionar como servidor para este cliente; incluyen los parámetros de configuración que tienen para ofrecer. El cliente transmite un mensaje DHCPREQUEST para aceptar uno de los servidores e informar a los servidores restantes que ha optado por rechazar sus ofertas. El servidor elegido responde con un mensaje de difusión DHCPACK, indicando la confirmación de los parámetros que ofreció originalmente. La asignación de direcciones está completa en este punto. Periódicamente, el cliente envía al servidor un mensaje DHCPREQUEST solicitando la renovación de la concesión de la dirección IP.

Si se renueva la concesión, el servidor responde con un mensaje DHCPACK de unidifusión. De lo contrario, el cliente vuelve al proceso de inicialización. La Tabla 5.6 enumera el protocolo de intercambio cliente/servidor completo para el servicio DHCP.

Tabla 5.5 Tipos de mensajes DHCP

Mensaje DHCP	Descripción
DHCPDESCUBRIR	Difusión de cliente para localizar servidores disponibles
DHCPOFFERTA	Servidor a cliente en respuesta a DHCPDISCOVER con oferta de parámetros de configuración
DHCPREQUEST	Mensaje del cliente a los servidores (a) solicitando parámetros ofrecidos de un servidor y rechazando implícitamente las ofertas de todos los demás; (b) confirmar la exactitud de la dirección previamente asignada después de, por ejemplo, reiniciar el sistema; o (c) extender el contrato de arrendamiento en una dirección de red en particular
DHCPPACK	Servidor a cliente con parámetros de configuración, incluida la dirección de red comprometida
DHCPNAK	Servidor a cliente que indica que la noción de dirección de red del cliente es incorrecta (por ejemplo, el cliente se ha movido a una nueva subred) o la concesión del cliente ha expirado
DHCPDECLINE	Cliente a servidor indicando que la dirección de red ya está en uso
LIBERARDHCP	Cliente a servidor renunciando a la dirección de red y cancelando el arrendamiento restante
DHCPINFORMAR	Cliente a servidor, solicitando solo parámetros de configuración locales; el cliente ya tiene una dirección configurada externamente

Tabla 5.6 Protocolo DHCP

Descripción	Protocolo	Dirección remota	Remoto		Local
			Puerto	Dirección local de entrada/salida	Puerto
DHCPDESCUBRIR; DHCPREQUEST UDP		255.255.255.255 67		Fuera 0.0.0.0	68
DHCPOFFER	UDP	0.0.0.0	67	En 255.255.255.255 68	
DHCPOFFER	UDP	SERVIDOR DHCP	67	En 255.255.255.255 68	
SOLICITUD DHCP; DHCPDECLINE UDP		SERVIDOR DHCP	67	Fuera 0.0.0.0	68
DHCPPACK; DHCPNACK	UDP	SERVIDOR DHCP	67	En ISP/MASCARA DE RED	68
DHCPPACK	UDP	SERVIDOR DHCP	67	En IPADDR	68
SOLICITUD DHCP; UDP DE LIBERACIÓN DE DHCP		SERVIDOR DHCP	67	Fuera IPADDR	68

El protocolo DHCP es mucho más complicado que este breve resumen, pero la suma describe los elementos esenciales del intercambio típico de cliente y servidor.

Las siguientes reglas de firewall permiten la comunicación entre su cliente DHCP y un servidor remoto:

```
# Inicialización o reenlace: Sin concesión o Tiempo de concesión vencido.
$IPT -A SALIDA -o $INTERNET -p udp \
    -s $BROADCAST_SRC --deporte 67:68 \
    -d $ BROADCAST_DEST --dport 67:68 -j ACEPTAR
```

DHCPOFFER entrante de los servidores DHCP disponibles

```
$IPT -A ENTRADA -i $INTERNET -p udp \
    --deporte 67:68 \
    --dport 67:68 -j ACEPTAR
```

Las reglas de nftables se ven así:

```
$NFT añadir salida de filtro de regla oif $INTERNET ip saddr $BROADCAST_SRC udp sport 67-68 ¿ip daddr
$BROADCAST_DEST udp dport 67-68 aceptar
$NFT agregar entrada de filtro de regla si $INTERNET udp sport 67-68 udp dport 67-68 aceptar
```

Tenga en cuenta que el tráfico DHCP no se puede limitar completamente a su servidor DHCP. Durante las secuencias de inicialización, cuando su cliente aún no tiene una dirección IP asignada o incluso la dirección IP del servidor, los paquetes se transmiten en lugar de enviarse punto a punto. En el nivel de Capa 2, los paquetes pueden dirigirse a la dirección de hardware de su tarjeta de red.

Acceso a servidores de hora de red remota (Puerto UDP 123)

Los servicios de tiempo de red como NTP permiten el acceso a uno o más proveedores de tiempo de Internet públicos. Esto es útil para mantener un reloj del sistema preciso, especialmente si su sistema interno

el reloj tiende a desviarse y a establecer la hora y la fecha correctas en el arranque o después de un corte de energía. Un usuario de sistema pequeño debe usar el servicio solo como un cliente de Internet. Pocos sitios pequeños tienen un enlace satelital a Greenwich, Inglaterra; un enlace de radio a un reloj atómico; o un reloj atómico propio tirado por ahí.

ntpd es el demonio del servidor. Además de brindar un servicio de tiempo a los clientes, ntpd utiliza una relación de igual a igual entre los servidores. Pocos sitios pequeños requieren la precisión adicional ntpd proporciona. ntpdate es el programa cliente y utiliza una relación cliente-servidor. El programa cliente es todo lo que necesitará un sitio pequeño. La Tabla 5.7 enumera solo el protocolo de intercambio cliente/servidor para el servicio NTP. Rara vez, si alguna, hay una razón para ejecutar ntpd porque ese es el componente del servidor. Si debe ejecutar el servidor NTP (a diferencia del cliente), hágalo en un entorno chroot .

La secuencia de comandos de inicio de ntpd que se ejecuta en el momento del arranque utiliza ntpdate para consultar una serie de proveedores de servicios de hora pública. El demonio ntpd se inicia después de la respuesta del servidor. Estos hosts se especificarían individualmente en una serie de reglas de firewall:

```
TIME_SERVER="mi.tiempo.servidor"                # Servidor de tiempo externo, si lo hay

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p udp \
        -s $IPADDR --deporte $UNPRIVPORTS \
        -d $SERVIDOR_HORA --dport 123 \
        -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p udp \
    -s $IPADDR --deporte $UNPRIVPORTS \
    -d $SERVIDOR_HORA --dport 123 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p udp \
    -s $SERVIDOR_HORA --deporte 123 \
    -d $IPADDR --dport $UNPRIVPORTS -j ACEPTAR
```

Las reglas del script nftables son:

```
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR udp sport $UNPRIVPORTS yip daddr $TIME_SERVER udp dport 123
aceptar
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $TIME_SERVER udp sport 123 ydaddr $IPADDR udp dport $UNPRIVPORTS
aceptar
```

Tabla 5.7 Protocolo NTP

Descripción	Protocolo	Remoto Habla a	Remoto Puerto	En fuera	Local Dirección Puerto local
cliente local					
consulta	UDP	SERVIDOR DE TIEMPO	123	Fuera	IPADDR 1024:65535
Servidor remoto					
respuesta	UDP	SERVIDOR DE TIEMPO	123	En	IPADDR 1024:65535

Tenga en cuenta que las reglas anteriores están escritas para una comunicación UDP cliente/servidor estándar. Dependiendo de su cliente y software de servidor en particular, es posible que uno o ambos usen el modelo de comunicación de servidor a servidor NTP, con el cliente y el servidor usando el puerto UDP 123.

Registro de paquetes entrantes descartados

Cualquier paquete que coincida con una regla se puede registrar mediante el objetivo -j LOG para iptables o la declaración de registro para nftables. Sin embargo, registrar un paquete no tiene efecto en la disposición del paquete. El paquete debe coincidir con una regla de aceptación o eliminación. Algunas de las reglas presentadas anteriormente tenían el registro habilitado, antes de hacer coincidir el paquete por segunda vez para descartarlo. Algunas de las reglas de suplantación de direcciones IP son ejemplos.

Las reglas se pueden definir con el propósito explícito de registrar ciertos tipos de paquetes. Más normalmente, los paquetes de interés son paquetes sospechosos que indican algún tipo de sondeo o exploración. Debido a que todos los paquetes se niegan de forma predeterminada, si se desea el registro para ciertos tipos de paquetes, se deben definir reglas explícitas antes de que el paquete caiga al final de la cadena y la política predeterminada entre en vigor. Esencialmente, de todos los paquetes denegados, es posible que le interese registrar algunos de ellos, utilizar el registro de velocidad limitada para algunos y descartar silenciosamente otros.

Qué paquetes se registran es un asunto individual. Algunas personas quieren registrar todos los caídos paquetes. Para otras personas, el registro de todos los paquetes perdidos pronto podría desbordar los registros de su sistema. Algunas personas, seguras de saber que los paquetes se descartan, no se preocupan por ellos y no quieren saber nada de ellos. Otras personas están interesadas en los escaneos de puertos obvios o en algún tipo de paquete en particular.

Debido al comportamiento de la primera regla que coincide, gana, puede registrar todos los paquetes entrantes descartados con una sola regla. La suposición aquí es que todas las reglas de aceptación de coincidencia de paquetes han sido probadas y el paquete está a punto de caer al final de la cadena y desecharse:

```
$IPT -A ENTRADA -i $INTERNET -j REGISTRO
```

O para nftables:

```
$NFT agregar entrada de filtro de regla si $INTERNET registro
```

Registro de paquetes salientes descartados

Es necesario registrar el tráfico saliente bloqueado por las reglas del firewall para depurar las reglas del firewall y recibir alertas sobre problemas de software local.

Todo el tráfico a punto de ser descartado por la política predeterminada podría registrarse:

```
$IPT -A SALIDA -o $INTERNET -j REGISTRO
```

Para nftables:

```
$NFT agregar salida de filtro de regla o registro de $INTERNET
```

Instalación del cortafuegos

Esta sección asume que la secuencia de comandos del cortafuegos se llama `rc.firewall`. No hay ninguna razón por la que el script no pueda llamarse simplemente `fwscript` o cualquier otra cosa. De hecho, en los sistemas Debian, el estándar está más cerca del nombre único, `fwscript`, en lugar de un nombre con el prefijo `rc.` como es el caso de Red Hat. Esta sección cubre los comandos como si el script estuviera instalado en `/etc/rc.d/` para un sistema Red Hat o SUSE o `/etc/init.d/` para un sistema Debian.

Como script de shell, la instalación inicial es simple. El script debe ser propiedad de root. En RedHat y SUSE:

```
chown root.root /etc/rc.d/rc.firewall
```

En Debian:

```
chown root.root /etc/init.d/rc.firewall
```

El script debe ser escribible y ejecutable solo por root . Idealmente, el usuario general no debe tener acceso de lectura. En Red Hat y SUSE:

```
chmod u=rwx /etc/rc.d/rc.firewall
```

En Debian:

```
chmod u=rwx /etc/init.d/rc.firewall
```

Para inicializar el cortafuegos en cualquier momento, ejecute el script desde la línea de comandos. Ahí no es necesario reiniciar:

```
Inicio /etc/rc.d/rc.firewall
```

Técnicamente, el argumento de inicio no se requiere allí, pero es un buen hábito de todos modos—de nuevo, prefiero errar por el lado de la integridad que tener ambigüedad con un firewall. El script incluye una acción de detención que vacía el firewall por completo. Por lo tanto, si desea detener el firewall, llame al mismo comando con el argumento de detención :

```
/etc/rc.d/rc.firewall parada
```

Tenga cuidado: si detiene el cortafuegos de esta manera, se está ejecutando sin protección. Los abogados me dicen que debo decirte: “¡Deja siempre el firewall habilitado!”.

En Debian, cambie la ruta del comando a `/etc/init.d`. Inicie el cortafuegos:

```
/etc/init.d/rc.firewall inicio
```

Detenga el cortafuegos en Debian:

```
/etc/init.d/rc.firewall parada
```

Sugerencias para depurar la secuencia de comandos del cortafuegos

Cuando está depurando un nuevo script de cortafuegos a través de SSH u otra conexión remota, es muy posible que se quede fuera del sistema. De acuerdo, esto no es una preocupación cuando instala el firewall desde la consola, pero como alguien que

140 Capítulo 5 Creación e instalación de un cortafuegos independiente

administra servidores Linux remotos, encuentro que el acceso a la consola rara vez es posible. Por lo tanto, es necesario un método para detener el cortafuegos automáticamente después de que se inicie, en caso de que el cortafuegos bloquee mi conexión. Cron al rescate.

Usando una entrada cron, puede detener el firewall ejecutando el script con un argumento de detención en algún intervalo predefinido. Encuentro que cada 2 minutos funciona bien durante la depuración inicial. Si desea utilizar este método, configure una entrada cron con el siguiente comando como root (en Debian):

```
crontab -e
*/2 * * * * /etc/init.d/rc.cortafuegos detener
```

En Red Hat y SUSE:

```
crontab -e
*/2 * * * * /etc/rc.d/rc.cortafuegos detener
```

Con esta entrada de cron en su lugar, puede iniciar el firewall y hacer que se detenga cada 2 minutos. Sin embargo, el uso de dicho mecanismo es algo así como una compensación, ya que debe realizar la depuración inicial antes de que el reloj alcance un minuto divisible por dos. Además, depende de usted recordar eliminar esta entrada cron cuando haya depurado el firewall. Si olvida eliminar esta entrada, el cortafuegos se detendrá y volverá a funcionar sin cortafuegos.

Iniciar el cortafuegos en el arranque con Red Hat y SUSE

En Red Hat y SUSE, la forma más sencilla de inicializar el cortafuegos es editar `/etc/rc.d/rc.local` y agregue la siguiente línea al final del archivo:

```
Inicio /etc/rc.d/rc.firewall
```

Una vez que las reglas del cortafuegos se depuran y se estabilizan, Red Hat Linux ofrece una versión más estable. Manera segura de iniciar y detener el cortafuegos. Si eligió iptables mientras usaba uno de los administradores de nivel de ejecución, el directorio de nivel de ejecución predeterminado contiene un enlace a `/etc/rc.d/init.d/iptables`. Al igual que con los otros scripts de inicio en este directorio, el sistema iniciará y detendrá el firewall automáticamente al iniciar o cambiar los niveles de ejecución.

Sin embargo, se requiere un paso adicional para usar el sistema de nivel de ejecución estándar. Debes primero instalar manualmente las reglas del firewall:

```
/etc/rc.d/rc.firewall
```

Luego ejecuta el comando

```
/etc/init.d/iptables guardar
```

Las reglas se guardarán en un archivo, `/etc/sysconfig/iptables`. Después de esto, el script de inicio encontrará este archivo y cargará las reglas guardadas automáticamente.

Se debe tener precaución al guardar y cargar las reglas del cortafuegos con este método. Las funciones de guardar y cargar de iptables no están completamente depuradas en este momento.

Si su configuración particular de firewall da como resultado un error de sintaxis al guardar o cargar

las reglas, debe continuar usando algún otro mecanismo de inicio, como ejecutar el script de firewall desde `/etc/rc.d/rc.local`.

Iniciar el cortafuegos en el arranque con Debian

Como ocurre con muchas otras cosas, configurar el script del cortafuegos para que se inicie en el arranque es más sencillo en Debian que en otras distribuciones. Puede hacer que el cortafuegos se inicie y se detenga al arrancar con el comando `update-rc.d`. Ejecute `update-rc.d` con la secuencia de comandos del cortafuegos en `/etc/init.d`, y configure su directorio actual en `/etc/init.d/` también:

```
cd /etc/init.d
actualizar-rc.d rc.firewall por defecto
```

Consulte la página de manual de `update-rc.d` para obtener más información sobre su uso más allá del que se muestra aquí.

Otros aspectos de la secuencia de comandos del cortafuegos dependen de si tiene una dirección IP estática registrada o una dirección IP dinámica asignada por DHCP. La secuencia de comandos del cortafuegos que se presenta en este capítulo está configurada para un sitio con una dirección IP permanente asignada estáticamente.

Instalación de un cortafuegos con una dirección IP dinámica

Si tiene una dirección IP asignada dinámicamente, el método de instalación de firewall estándar no funcionará sin modificaciones. Las reglas del cortafuegos se instalarían antes que la red.

se activan las interfaces de trabajo, antes de que se le asigne una dirección IP al sistema y, posiblemente, antes de que se le asigne un enrutador de puerta de enlace predeterminado o servidores de nombres.

El propio script del cortafuegos necesita que se definan los valores `IPADDR` y `NAMESERVER`. Ambos El servidor DHCP y el archivo local `/etc/resolv.conf` pueden definir hasta tres servidores de nombres. Además, cualquier sitio determinado puede conocer o no las direcciones de sus servidores de nombres, enrutador de puerta de enlace predeterminado o servidor DHCP con anticipación. Además, no es raro que su máscara de red, subred y direcciones de transmisión cambien con el tiempo a medida que el ISP vuelve a numerar su red. Algunos ISP asignan una dirección IP diferente con frecuencia, por lo que su dirección IP puede cambiar varias veces durante el transcurso de una conexión en curso.

Su sitio debe proporcionar algún medio para actualizar dinámicamente las reglas de firewall instaladas. medida que ocurren estos cambios. El Apéndice B, "Ejemplos de cortafuegos y secuencias de comandos de soporte", proporciona secuencias de comandos de muestra diseñadas para manejar estos cambios automáticamente.

La secuencia de comandos del cortafuegos podría leer estas variables de shell directamente desde el entorno o podría leerlas desde un archivo. En cualquier caso, las variables no estarían codificadas en el script del cortafuegos, como lo están en el ejemplo de este capítulo.

Resumen

Este capítulo lo guió a través de los procesos involucrados en el desarrollo de un firewall independiente utilizando iptables y nftables. Se estableció la política de denegación por defecto. Algunos vectores de ataque de uso común se corrigieron al comienzo del script, incluida la fuente

suplantación de identidad, protección de servicios que se ejecutan en puertos no privilegiados y DNS. Se mostraron ejemplos de reglas para servicios de red populares. Finalmente, se describieron los problemas relacionados con la instalación de firewall, tanto para sitios con una dirección IP estática como para sitios con una dirección IP asignada dinámicamente.

El Capítulo 6 utiliza el cortafuegos independiente como base para crear un cortafuegos optimizado. El Capítulo 7 lo usa como base para una arquitectura de firewall más complicada. En el Capítulo 7 se describe una arquitectura de subred filtrada que usa dos firewalls que separan una red perimetral DMZ. Una pequeña empresa fácilmente podría tener la necesidad y los recursos para esta configuración más elaborada. El Capítulo 8 utiliza el cortafuegos independiente como base para sus ejemplos, pero no se basa directamente en este ejemplo.

yl

problemas avanzados, **Múltiples** **cortafuegos y perímetro** **Redes**

- 6 Optimización de cortafuegos
- 7 Reenvío de paquetes
- 8 NAT: traducción de direcciones de red
- 9 Depuración de las reglas del cortafuegos
- 10 redes privadas virtuales

Esta página se dejó en blanco intencionalmente

Optimización de cortafuegos

El Capítulo 5, "Creación e instalación de un cortafuegos independiente", usó las iptables y programas de administración de cortafuegos nftables para crear un cortafuegos de diseño personalizado, simple y de un solo sistema. Este capítulo presenta la optimización del cortafuegos. La optimización se puede dividir en tres categorías principales: organización de reglas, uso del módulo de estado y cadenas definidas por el usuario. El ejemplo del capítulo anterior se mostró con y sin el uso del módulo de estado. Este capítulo se centra en la organización de reglas y las cadenas definidas por el usuario.

Organización de reglas

Se puede hacer poca optimización usando solo las cadenas INPUT, OUTPUT y FORWARD .

El recorrido de la cadena es de arriba a abajo, una regla a la vez, hasta que el paquete coincide con una regla. Las reglas de una cadena deben ordenarse jerárquicamente, de la más general a la más específica.

No existe una fórmula estricta para la organización de reglas. Los dos principales factores subyacentes son qué servicios están alojados en la máquina y el propósito principal de la máquina, destacando especialmente los servicios con mayor tráfico en la máquina. Los requisitos de un servidor de seguridad dedicado y un reenviador de paquetes son muy diferentes de los de un servidor de seguridad bastión que protege un servidor web o de correo dedicado. Del mismo modo, es probable que un administrador del sitio establezca diferentes prioridades de rendimiento en una máquina con firewall que sirve principalmente como estación de trabajo que en un firewall que sirve como puerta de enlace residencial y servidor Linux para un hogar.

El tercer factor subyacente a considerar cuando se prepara para organizar reglas para la optimización del firewall es el ancho de banda de red disponible, la velocidad de la conexión a Internet.

No es probable que la optimización compre mucho, si es que algo, para un sitio con una conexión a Internet de velocidad residencial. Incluso para un sitio web muy visitado, es poco probable que la CPU de la máquina sude. El cuello de botella es la propia conexión a Internet.

Comience con reglas que bloqueen el tráfico en puertos altos

Como demostraron los ejemplos del Capítulo 5, la mayor parte de las reglas son reglas antispoofing o reglas que bloquean el tráfico en puertos altos específicos (como NFS o X Windows). Este tipo de reglas deben preceder a las reglas que permiten el tráfico a servicios específicos. Obviamente, las reglas del canal de datos FTP deben estar cerca del final de la lista de reglas, aunque querrá que las reglas estén cerca de la parte superior de la lista porque las transferencias FTP tienden a ser grandes.

Use el módulo de estado para partidos ESTABLECIDOS y RELACIONADOS

El uso de las coincidencias ESTABLECIDAS y RELACIONADAS del módulo de estado esencialmente permite mover todas las reglas para los intercambios en curso a la cabeza de las cadenas, así como eliminar la necesidad de reglas específicas para la mitad del servidor de una conexión. De hecho, eludir la coincidencia de filtros para intercambios en curso, reconocidos y previamente aceptados es uno de los dos propósitos principales del módulo de estado.

El segundo propósito principal del módulo de estado es cumplir una función de filtrado de cortafuegos.

El seguimiento del estado de la conexión permite que el cortafuegos asocie paquetes con intercambios en curso.

Esto es particularmente útil para intercambios UDP sin conexión y sin estado.

Considere el Protocolo de Transporte

El protocolo de transporte sobre el que se ejecuta el servicio es otro factor. En un firewall estático, la sobrecarga de probar cada paquete entrante contra todas las reglas de suplantación de identidad es una gran pérdida.

Servicios TCP: eludir las reglas de suplantación de identidad

Incluso sin el módulo de estado, para los servicios basados en TCP, la regla para la mitad del servidor remoto de una conexión puede eludir las reglas de suplantación de identidad. La capa TCP eliminará los paquetes falsificados entrantes con el bit ACK establecido porque el paquete no coincidirá con ninguno de los estados de conexión establecidos de la capa TCP.

Sin embargo, la mitad del cliente remoto de un par de reglas debe seguir las reglas de suplantación de identidad, ya que la regla de cliente típica cubre tanto la solicitud de conexión inicial como el tráfico continuo del cliente. Si los indicadores SYN y ACK se prueban individualmente, las reglas que prueban el indicador ACK en los paquetes que llegan de clientes remotos pueden pasar por alto las pruebas de suplantación de identidad. Las pruebas de suplantación de identidad deben aplicarse solo a la solicitud SYN inicial.

El uso del módulo de estado también permite que la regla para la solicitud de conexión entrante del cliente remoto, el paquete SYN inicial, sea lógicamente independiente de la regla para los paquetes ACK subsiguientes del cliente. Solo la solicitud de conexión inicial, el paquete NUEVO inicial, debe probarse con las reglas de suplantación de identidad.

Servicios UDP: Coloque las reglas de paquetes entrantes después de las reglas de suplantación de identidad

Sin el módulo de estado, para los servicios basados en UDP, la regla para los paquetes entrantes siempre debe seguir las reglas de suplantación de identidad. El concepto de cliente y servidor se mantiene a nivel de aplicación, suponiendo que se mantenga en absoluto. En los niveles de firewall y UDP, sin estado de conexión, no hay indicación del iniciador y el respondedor, aparte del puerto de servicio o el puerto sin privilegios utilizado.

DNS es un ejemplo de un servicio UDP sin conexión. Sin estado de conexión, hay no es una asignación entre la dirección de destino donde el cliente envió una consulta y la dirección de origen en un paquete de respuesta entrante. Una de las razones por las que es posible el envenenamiento de la caché DNS es que las implementaciones del servidor DNS no verifican si un paquete entrante fue una respuesta legítima del servidor consultado previamente o si el paquete se envió desde alguna otra dirección. Además, algunas implementaciones ni siquiera aseguran

que un cliente hizo una solicitud. Un paquete malicioso entrante no solicitado podría usarse para actualizar el caché de DNS local incluso sin que se haya realizado una consulta inicial.

Servicios TCP frente a UDP: coloque las reglas UDP después de las reglas TCP

En general, las reglas UDP deben colocarse más adelante en las cadenas de firewall, después de cualquier regla TCP. Esto se debe a que la mayoría de los servicios de Internet se ejecutan a través de TCP, y los servicios UDP sin conexión suelen ser servicios de consulta y respuesta simples, de un solo paquete. Probar el paquete único o UDP o un puñado de ellos con las reglas anteriores para las conexiones TCP en curso no agrega un arrastre notable a una consulta y respuesta UDP. Para empezar, los protocolos de sesión de conexión múltiple no son amigables con el firewall. Dichos servicios no pueden pasar a través de un firewall o NAT sin soporte ALG específico.

Servicios ICMP: coloque sus reglas al final de la cadena de reglas

ICMP es otro protocolo cuyas reglas de firewall se pueden colocar al final de la cadena de reglas. Los paquetes ICMP son pequeños mensajes de control y estado. Como tales, se envían con relativa poca frecuencia. Los paquetes ICMP legítimos generalmente consisten en un solo paquete no fragmentado. Con la excepción de la solicitud de eco, los paquetes ICMP casi siempre se envían como un mensaje de estado o de control en respuesta a algún tipo de paquete saliente excepcional.

Coloque reglas de firewall para servicios muy utilizados lo antes posible

Por lo general, no existen reglas estrictas para la colocación de reglas de firewall en una lista. Las reglas para servicios muy utilizados, como las reglas relacionadas con HTTP para un servidor web dedicado, deben colocarse lo antes posible. Las reglas para las aplicaciones que implican un alto número de paquetes en curso también deben establecerse lo antes posible. Sin embargo, como se mencionó anteriormente, los protocolos de flujo de datos para aplicaciones como FTP requieren que las reglas se coloquen cerca del final de la cadena, después de cualquier otra regla de aplicación, a menos que se use un ayudante específico para esos protocolos.

Use el flujo de tráfico para determinar dónde colocar las reglas para múltiples interfaces de red

Si el host tiene múltiples interfaces de red, se deben establecer reglas específicas para una determinada interfaz con respecto a qué interfaces tendrán el mayor flujo de tráfico. Las reglas para esas interfaces deben preceder a las reglas para otras interfaces. Las consideraciones de interfaz probablemente sean de poco interés para un sitio residencial, pero pueden tener un gran impacto en el rendimiento de un sitio comercial.

Como ejemplo, este problema surgió hace varios años con un pequeño ISP que había construido un firewall basado en los ejemplos `ipfwadm` e `ipchains` en el sitio web del autor anterior Bob Ziegler. Como se muestra en la Figura 6.1 y la Figura 6.2, la ruta que toman los paquetes a través del sistema operativo es muy diferente entre IPFW y Netfilter. A diferencia de Netfilter e `iptables`, con `ipchains`, los paquetes que pasan entre las interfaces de red se pasan desde el

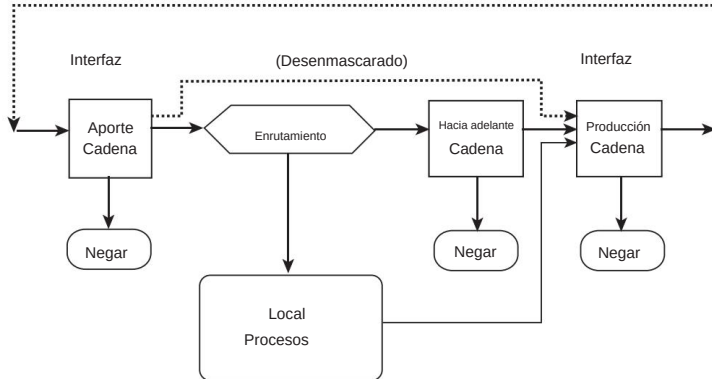


Figura 6.1 Loopback de IPFW y recorrido de paquetes enmascarados

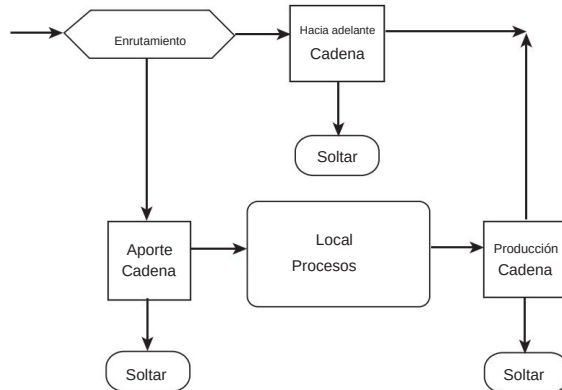


Figura 6.2 Travesía de paquetes de Netfilter

cadena de ENTRADA a la cadena ADELANTE a la cadena de SALIDA . Los ejemplos en el sitio web fueron pensados como ejemplos para las personas en el hogar. Las reglas de entrada y salida para la LAN fueron las últimas reglas en los scripts. Las reglas específicas para el host local de Linux fueron lo primero. El cortafuegos del ISP funcionaba principalmente como un enrutador o puerta de enlace. A través de la experimentación, el ISP descubrió que mover las reglas de E/S para la interfaz LAN al comienzo de las cadenas de ENTRADA y SALIDA resultó en un aumento de más de un megabit por segundo en el rendimiento.

Cadenas definidas por el usuario

Para iptables , la tabla de filtros tiene tres cadenas integradas permanentes: INPUT, OUTPUT y FORWARD. iptables le permite definir sus propias cadenas, llamadas cadenas definidas por el usuario. En nftables , todas las tablas están definidas por el usuario, pero en términos prácticos, todavía se usa la tabla de filtros .

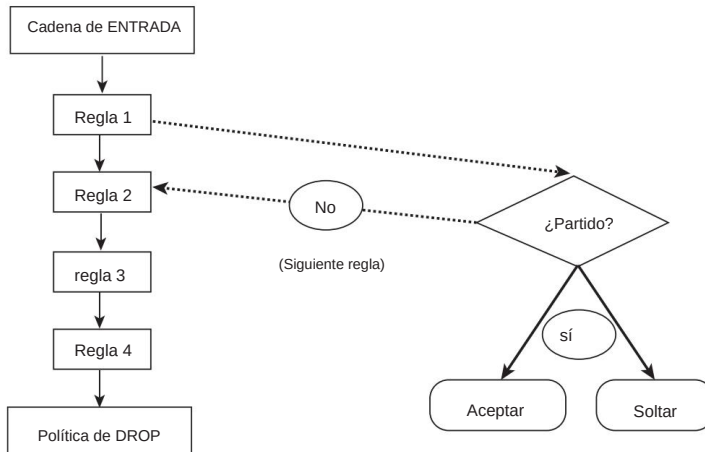


Figura 6.3 Travesía de cadena estándar

Las cadenas definidas por el usuario se tratan como objetivos de regla, es decir, según el conjunto de coincidencias especificado en una regla, el objetivo puede bifurcarse o saltar a una cadena definida por el usuario. En lugar de aceptar o descartar el paquete, el control se pasa a la cadena definida por el usuario para realizar pruebas de coincidencia más específicas en relación con los paquetes que coinciden con la regla de bifurcación. Después de atravesar la cadena definida por el usuario, el control vuelve a la cadena de llamada y la coincidencia continúa a partir de la siguiente regla en la cadena de llamada, a menos que la cadena definida por el usuario coincida y tome medidas en el paquete.

La figura 6.3 muestra el recorrido estándar de la regla de arriba hacia abajo utilizando las cadenas integradas.

Las cadenas definidas por el usuario son útiles para optimizar el conjunto de reglas y, por lo tanto, se utilizan con frecuencia. Permiten que las reglas se organicen en árboles categóricos. En lugar de confiar en el tipo de comparación directa y de arriba hacia abajo de la lista de verificación inherente al recorrido de la cadena estándar, las pruebas de comparación de paquetes se pueden reducir selectivamente en función de las características del paquete. La figura 6.4 muestra el flujo de paquetes inicial. Después de realizar las pruebas iniciales comunes a todos los paquetes entrantes, la coincidencia de paquetes se bifurca en función de la dirección de destino en el paquete.

La ramificación se basa en la dirección de destino en este ejemplo. La coincidencia de la dirección de origen se realiza más adelante en relación con aplicaciones específicas, como DNS remoto o servidores de correo. En la mayoría de los casos, la dirección remota será "cualquier lugar". La coincidencia en la dirección de destino en este punto distingue entre los paquetes de unidifusión destinados a esta máquina, los paquetes de difusión, los paquetes de multidifusión y (dependiendo de si es la cadena INPUT o FORWARD) los paquetes destinados a los hosts internos.

La Figura 6.5 detalla la cadena definida por el usuario para las reglas de protocolo para los paquetes dirigidos específicamente a este host. Como se muestra, las cadenas definidas por el usuario pueden saltar a otras cadenas definidas por el usuario que contienen pruebas aún más específicas.

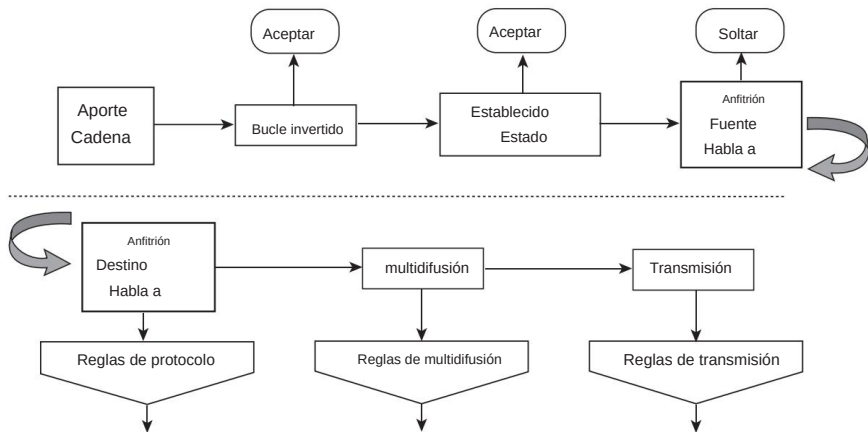


Figura 6.4 Cadenas definidas por el usuario basadas en la dirección de destino

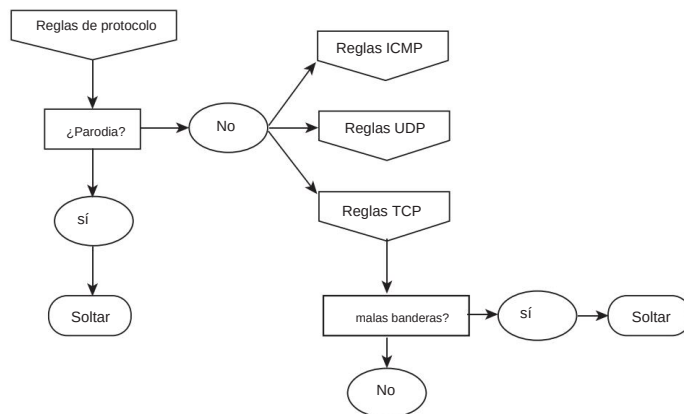


Figura 6.5 Cadenas definidas por el usuario basadas en el protocolo

Esta lista resume las características de las cadenas definidas por el usuario del Capítulo 3, "iptables: el programa de administración de firewall heredado de Linux":

- Las cadenas definidas por el usuario se crean con la operación `-N` o `--new-chain`.
- Los nombres de cadena definidos por el usuario pueden tener hasta 30 caracteres de longitud.
- Los nombres de cadena definidos por el usuario pueden contener guiones (`-`) pero no guiones bajos (`_`).
- Se accede a las cadenas definidas por el usuario como objetivos de reglas.
- Las cadenas definidas por el usuario no tienen políticas predeterminadas.

- Las cadenas definidas por el usuario pueden llamar a otras cadenas definidas por el usuario.
- Si el paquete no coincide con una regla en la cadena definida por el usuario, el control vuelve a la siguiente regla en la cadena de llamadas.
- La cadena definida por el usuario se puede salir antes y el control regresa a la siguiente regla en la cadena de llamadas, mediante el uso del destino RETURN .
- Las cadenas definidas por el usuario se eliminan con la operación -X o --delete-chain .
- Una cadena debe estar vacía antes de que se pueda eliminar.
- Una cadena no puede tener ninguna referencia a ella desde otras cadenas para ser eliminada.
- Una cadena se vacía específicamente por nombre, o todas las cadenas existentes se vacían si no se especifica ninguna cadena, con la operación -F o --flush .

La siguiente sección aprovecha las cadenas definidas por el usuario y los conceptos presentados en la sección sobre organización de reglas para optimizar el firewall de sistema único presentado en el Capítulo 5.

Ejemplos optimizados

A continuación, se muestran ejemplos optimizados de los cortafuegos creados en el Capítulo 5. El primer ejemplo es para el cortafuegos basado en iptables . Si está utilizando nftables, puede omitir esta sección de manera segura e ir al ejemplo de secuencia de comandos de nftables .

El script de iptables optimizado

Se declara una nueva variable, USER_CHAINS, que contiene los nombres de todas las cadenas definidas por el usuario utilizadas en el script. Las cadenas se enumeran aquí:

- **tcp-state-flags:** contiene las reglas para verificar si el indicador de estado de TCP no es válido. combinaciones
- **seguimiento de conexión:** contiene las reglas para verificar coincidencias relacionadas con el estado, NO VÁLIDO, ESTABLECIDO y RELACIONADO.
- **source-address-check:** contiene las reglas para comprobar si hay direcciones de origen ilegales.
- **comprobación de dirección de destino:** contiene las reglas para comprobar si hay un destino ilegal direcciones.
- **Entrada EXT:** contiene las cadenas definidas por el usuario específicas de la interfaz para la ENTRADA cadena. En este ejemplo, el host tiene una interfaz conectada a Internet.
- **Salida EXT:** contiene las cadenas definidas por el usuario específicas de la interfaz para la SALIDA cadena. En este ejemplo, el host tiene una interfaz conectada a Internet.
- **local-dns-server-query:** contiene las reglas para las consultas salientes del servidor DNS local o de los clientes locales.
- **remote-dns-server-response:** contiene las reglas para las respuestas entrantes de un servidor DNS remoto.

- **local-tcp-client-request:** contiene las reglas para las solicitudes de conexión TCP salientes y el tráfico de clientes generado localmente a servidores remotos.
- **remote-tcp-server-response:** contiene las reglas para las respuestas entrantes de servidores TCP remotos.
- **remote-tcp-client-request:** contiene las reglas para la conexión TCP entrante solicitudes y tráfico de clientes generado de forma remota a servidores locales.
- **local-tcp-server-response:** contiene las reglas para las respuestas salientes a clientes remotos.
- **local-udp-client-request:** contiene las reglas para el tráfico de cliente UDP saliente a servidores remotos.
- **remote-udp-server-response:** contiene las reglas para las respuestas entrantes de servidores UDP remotos.
- **EXT-icmp-out:** contiene las reglas para los paquetes ICMP salientes.
- **EXT-icmp-in:** contiene las reglas para los paquetes ICMP entrantes.
- **Inicio de sesión EXT:** contiene las reglas de registro para los paquetes entrantes antes de descartarlos mediante la política de ENTRADA predeterminada .
- **EXT-log-out:** contiene las reglas de registro para los paquetes salientes antes de descartarlos mediante la política de SALIDA predeterminada .
- **log-tcp-state:** contiene las reglas de registro para los paquetes TCP con combinaciones de indicadores de estado ilegales antes de descartarlos.
- **remote-dhcp-server-response:** contiene las reglas para los paquetes entrantes del servidor DHCP de este host.
- **local-dhcp-client-query:** contiene las reglas para el cliente DHCP saliente paquetes

Algunas cadenas específicas de interfaz tienen como prefacio EXT para diferenciarlas de cualquier cadena definida por el usuario que contenga reglas para cualquier interfaz LAN. Este ejemplo de firewall asume que solo hay una interfaz, la interfaz externa. El punto es sugerir que se podrían definir diferentes reglas y políticas de seguridad por interfaz.

La declaración real en el script de shell del firewall sería como se muestra aquí:

```
USER_CHAINS="EXT-entrada          Salida EXT \
tcp-state-flags seguimiento de conexión \
verificación de la dirección de origen verificación de la dirección de destino \
consulta-servidor-dns-local respuesta-servidor-dns-remoto \
solicitud-cliente-tcp-local respuesta-servidor-tcp-remoto \
solicitud-cliente-tcp-remoto respuesta-servidor-tcp-local \
solicitud-cliente-udp-local respuesta-servidor-udp-remoto \
local-dhcp-client-query remote-dhcp-server-response \
EXT-icmp-salida          Entrada EXT-icmp \
Inicio de sesión EXT      EXT-cierre de sesión \
registro-tcp-estado"
```

Inicialización del cortafuegos

La secuencia de comandos del firewall comienza de manera idéntica al ejemplo del Capítulo 5. Recuerde que se establecieron varias variables de shell, incluida una llamada \$IPT para definir la ubicación del comando de administración del firewall de iptables :

```
#!/bin/sh
```

```
IPT="/sbin/iptables"           # Ubicación de iptables en su sistema
INTERNET="eth0"                 # Interfaz conectada a Internet
LOOPBACK_INTERFACE="lo"        # Sin embargo, su sistema lo nombra
IPADDR="mi.dirección.ip"       # Su dirección IP
MY_ISP="mi.isp.dirección.rango" # Servidor ISP y rango de direcciones NOC
SUBNET_BASE="mi.subred.red"     # La dirección de red de su subred
SUBNET_BROADCAST="my.subnet.bcast" # Dirección de transmisión de su subred
LOOPBACK="127.0.0.0/8"         # Rango de direcciones de loopback reservado
CLASE_A="10.0.0.0/8"           # Redes privadas clase A
CLASE_B="172.16.0.0/12"        # Redes privadas clase B
CLASE_C="192.168.0.0/16"       # Redes privadas clase C
CLASE_D_MULTICAST="224.0.0.0/4" # Direcciones de multidifusión de clase D
CLASS_E_RESERVED_NET="240.0.0.0/5" # Direcciones reservadas Clase E
TRANSMITIR_SRC = "0.0.0.0"      # Dirección de origen de la transmisión
BROADCAST_DEST = "255.255.255.255" # Dirección de destino de transmisión
PRIVPORTS="0:1023"             # Rango de puertos privilegiado y bien conocido
UNPRIVPORTS="1024:65535"       # Rango de puertos sin privilegios
```

También se establecieron una serie de parámetros del núcleo; consulte el Capítulo 5 para obtener una explicación de estos parámetros:

```
# Habilitar la protección de eco de transmisión
eco 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Deshabilitar paquetes enrutados de origen
para f en /proc/sys/net/ipv4/conf/*accept_source_route; hacer
    eco 0 > $f
hecho

# Habilitar la protección de cookies TCP SYN
eco 1 > /proc/sys/net/ipv4/tcp_syncookies
# Deshabilitar la aceptación de redirección ICMP
para f en /proc/sys/net/ipv4/conf/*accept_redirects; hacer
    eco 0 > $f
hecho

# No envíe mensajes de redirección
para f en /proc/sys/net/ipv4/conf/*send_redirects; hacer
    eco 0 > $f
hecho

# Soltar paquetes falsificados que llegan a una interfaz que, si se responde,
# daría como resultado que la respuesta saliera de una interfaz diferente.
para f en /proc/sys/net/ipv4/conf/*rp_filter; hacer
    eco 1 > $f
hecho

# Registra paquetes con direcciones imposibles.
para f en /proc/sys/net/ipv4/conf/*log_martians; hacer
    eco 1 > $f
hecho
```

154 Capítulo 6 Optimización del cortafuegos

Las cadenas integradas y cualquier cadena definida por el usuario preexistente se vacían:

```
# Eliminar cualquier regla existente de todas las cadenas
$IPT --vaciar
$IPT -t nat --vaciar
$IPT -t mangle --vaciar
```

El siguiente paso sería eliminar las cadenas definidas por el usuario. Se pueden eliminar con los siguientes comandos:

```
$IPT-X
$IPT -t nacional -X
$IPT -t destrozar -X
```

La política predeterminada se establece primero en ACEPTAR para todas las cadenas integradas:

```
# Restablecer la política predeterminada
$IPT --política ENTRADA ACEPTADA
$IPT --POLITICA SALIDA ACEPTADA
$IPT --política ADELANTE ACEPTAR
$IPT -t nat --policy ACEPTACIÓN DE ENRUTAMIENTO PREVIO
$IPT -t nat --policy SALIDA ACEPTAR
$IPT -t nat --policy ACEPTAR POSTERIOR ENRUTAMIENTO
$IPT -t mangle --policy ACEPTACIÓN DE ENRUTAMIENTO PREVIO
$IPT -t mangle --policy SALIDA ACEPTAR
```

Aquí está el código final para el comienzo de la secuencia de comandos del cortafuegos, es decir, el código para permitir que el cortafuegos se detenga fácilmente. Con este código colocado debajo del código anterior, cuando llame a la secuencia de comandos con un argumento de detención, la secuencia de comandos vaciará, borrará y restablecerá las políticas predeterminadas, y el cortafuegos se detendrá de manera efectiva.

```
if [ "$1" = "stop" ] luego echo
";Firewall completamente
detenido! ADVERTENCIA: ESTE HOST NO TIENE FIREWALL FUNCIONANDO". salir 0 fi
```

Ahora restablezca la política predeterminada real a DROP:

```
$IPT --política ENTRADA DROP
$IPT --política CAÍDA DE SALIDA
$IPT --política FORWARD DROP
```

El tráfico a través de la interfaz loopback está habilitado:

```
# Tráfico ilimitado en la interfaz loopback
$IPT -A ENTRADA -i lo -j ACEPTAR
$IPT -A SALIDA -o lo -j ACEPTAR
```

Ahora el script comienza a diferir del ejemplo del Capítulo 5.

Ahora se pueden crear las cadenas definidas por el usuario. Sus nombres se incluyeron en la única variable de shell, USER_CHAINS, solo con este propósito:

```
# Crea las cadenas definidas por el usuario para i en
$USER_CHAINS; hacer
    $IPT -N $he
terminado
```

Instalación de las cadenas

Desafortunadamente, la naturaleza similar a una llamada de función de construir e instalar las cadenas no se presta a una explicación en serie, paso a paso, sin la capacidad de mostrar diferentes lugares en el script simultáneamente, uno al lado del otro.

La idea es colocar las reglas en las cadenas definidas por el usuario y luego instalar esas cadenas en las cadenas integradas INPUT, OUTPUT y FORWARD . Si el script contiene un error y sale mientras se construyen las cadenas definidas por el usuario, las cadenas integradas no contendrán reglas, la política DROP predeterminada estará en vigor y, presumiblemente, se habilitará el tráfico de loopback.

Por lo tanto, esta primera sección de instalación se coloca al final del script del cortafuegos. El primer paso es verificar si hay combinaciones ilegales de indicadores de estado de TCP:

```
# Si es TCP: compruebe si hay patrones comunes de estado TCP de escaneo sigiloso
$IPT -A INPUT -p tcp -j tcp-state-flags
$IPT -A SALIDA -p tcp -j tcp-state-flags
```

Tenga en cuenta que se puede hacer referencia a la misma cadena desde más de una cadena de llamada. No es necesario duplicar las reglas de las cadenas definidas por el usuario para las cadenas de ENTRADA y SALIDA . Ahora, cuando el procesamiento del paquete llegue a este punto, el procesamiento "saltará" a la cadena tcp-state-flags definida por el usuario. Cuando se complete el procesamiento dentro de esa cadena, el procesamiento volverá aquí y continuará, a menos que se encuentre una disposición final para el paquete en la cadena definida por el usuario.

Si se está utilizando el módulo de estado, el siguiente paso es omitir el firewall por completo si el paquete es parte de un intercambio en curso previamente aceptado:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    # Omita los filtros de firewall para intercambios establecidos
    $IPT -A INPUT -j seguimiento de conexión
    $IPT -A SALIDA -j seguimiento de conexión
ser
```

Si la máquina es un cliente DHCP, se debe hacer una provisión para los mensajes de difusión enviados entre el cliente y el servidor durante la inicialización. También se debe hacer una provisión para aceptar la dirección de origen de la transmisión, 0.0.0.0. Las pruebas de verificación de direcciones de origen y destino eliminarían el tráfico DHCP inicial:

```
si [ "$DHCP_CLIENT" = "1" ]; luego
    $IPT -A ENTRADA -i $INTERNET -p udp \
        --sport 67 --dport 68 -j respuesta-servidor-dhcp-remoto
    $IPT -A SALIDA -o $INTERNET -p udp \
        --sport 68 --dport 67 -j local-dhcp-cliente-consulta
ser
```

Ahora salte a la cadena definida por el usuario para eliminar los paquetes entrantes que utilizan la dirección IP de este host como su dirección de origen. Luego pruebe otras direcciones de origen y destino ilegales:

```
# Prueba de direcciones de origen y destino ilegales en los paquetes entrantes
$IPT-UNA ENTRADA! -p tcp -j verificación de dirección de origen
$IPT -A INPUT -p tcp --syn -j verificación de dirección de origen
$IPT -A INPUT -j verificación-dirección-destino
```

156 Capítulo 6 Optimización del cortafuegos

```
# Prueba de direcciones de destino ilegales en paquetes salientes
$IPT -A SALIDA -j verificación-dirección-destino
```

Los paquetes salientes generados localmente no necesitan que se verifique su dirección de origen porque las reglas del firewall requieren explícitamente la dirección IP de este host en el campo de origen. Sin embargo, la verificación de la dirección de destino se realiza en los paquetes salientes.

En este punto, los paquetes entrantes regulares dirigidos a la dirección IP de este host pueden transferirse a la sección principal del firewall. Cualquier paquete entrante que no coincida con una regla en la cadena de entrada EXT volverá aquí para ser registrado y descartado:

```
# Comience las pruebas de firewall estándar para los paquetes dirigidos a este host
$IPT -A ENTRADA -i $INTERNET -d $IPADDR -j Entrada EXT
```

Es necesario un conjunto final de pruebas sobre la dirección de destino. Paquetes de difusión y multidifusión no están dirigidos a la dirección IP de unidifusión de este host. Están dirigidos a una dirección de difusión o multidifusión.

Como se mencionó en el Capítulo 5, los paquetes de multidifusión no se recibirán a menos que se registre para recibir paquetes dirigidos a una dirección de multidifusión en particular. Si desea recibir paquetes de multidifusión, debe aceptarlos todos o agregar una regla específica para la dirección y el puerto en particular utilizados para una sesión determinada. El siguiente código le permite elegir si descartar o aceptar el tráfico:

```
# tráfico de multidifusión
$IPT -A INPUT -i $INTERNET -p udp -d $CLASS_D_MULTICAST -j [ DROP | ACCEPTAR ]
$IPT -A SALIDA -o $INTERNET -p udp -s $IPADDR -d $CLASS_D_MULTICAST \
    -j [ SOLTAR | ACCEPTAR ]
```

En este punto, los paquetes salientes regulares de este host pueden transferirse a la sección principal del firewall. Cualquier paquete saliente que no coincida con una regla en la salida EXT la cadena volverá aquí para ser registrada y descartada:

```
# Comience las pruebas de firewall estándar para los paquetes enviados desde este host.
# La suplantación de dirección de origen por parte de este host no está permitida debido a la
# prueba en la dirección de origen en esta regla.
$IPT -A SALIDA -o $INTERNET -s $IPADDR -j Salida EXT
```

Las últimas reglas de entrada y salida ignoran implícitamente cualquier mensaje de difusión. Dependiendo de la naturaleza de la red pública o externa a la que la máquina está directamente conectada, las transmisiones pueden ser muy comunes en la subred local. Probablemente no desee registrar dichos mensajes, incluso con el registro de frecuencia limitada.

Finalmente, la política predeterminada elimina los paquetes restantes. Cualquier registro se haría en este punto:

```
# Registre cualquier cosa de interés que falle,
# antes de que la política predeterminada descarte el paquete.
$IPT -A INPUT -j EXT-inicio de sesión
$IPT -A SALIDA -j EXT-cierre de sesión
```

Esto marca el final del cortafuegos y es la última referencia a la ENTRADA y SALIDA cadenas

Creación de cadenas de entrada y salida EXT definidas por el usuario

Esta sección describe la construcción de las cadenas definidas por el usuario a las que se saltó en la sección anterior.

En el nivel superior, las reglas se basan en las cadenas generales de entrada EXT y salida EXT. Estas reglas son saltos a conjuntos de coincidencias más específicos contenidos en las cadenas dedicadas definidas por el usuario que ha creado.

Tenga en cuenta que la capa de entrada EXT y salida EXT no es necesaria. Las siguientes reglas y los saltos podrían haberse asociado fácilmente con las cadenas de ENTRADA y SALIDA integradas.

Sin embargo, el uso de estas cadenas tiene una ventaja. Debido a que los saltos a estas cadenas dependen de la dirección de origen o de destino, en este punto sabe que el paquete entrante está dirigido a este host y tiene una dirección de origen que se cree que es legítima. El paquete saliente se dirige desde este host y tiene una dirección de destino que se cree que es legítima. Además, si el módulo de estado está en uso, el paquete es el primer paquete en un intercambio o un nuevo paquete ICMP no relacionado.

En resumen, las cadenas de entrada EXT y salida EXT se utilizarán para seleccionar el tráfico por protocolo y por dirección, en términos de cliente o servidor. Cada regla proporcionará el punto de ramificación a las reglas de firewall específicas para ese protocolo y las características del paquete. Las coincidencias realizadas por las reglas de entrada EXT y salida EXT son el corazón de la optimización disponible con cadenas definidas por el usuario.

Tráfico DNS

Las reglas para identificar el tráfico DNS son lo primero. Hasta que se instalen las reglas de DNS, su software de red no podrá ubicar servicios y hosts en Internet a menos que use la dirección IP.

El primer par de reglas coincide con las consultas del caché local y el servidor de nombres de reenvío, si tiene uno, y las respuestas del servidor DNS remoto. El servidor local está configurado como esclavo del servidor principal remoto, por lo que el servidor local fallará si la búsqueda no se realiza correctamente. Esta configuración es menos común para una pequeña oficina/oficina en el hogar:

```
$IPT -A EXT-salida -p udp --sport 53 --dport 53 \  
-j local-dns-servidor-consulta
```

```
$IPT -A Entrada EXT -p udp --sport 53 --dport 53 \  
-j respuesta-servidor-dns-remoto
```

El siguiente par de reglas coincide con las solicitudes de búsqueda de clientes DNS estándar a través de TCP, cuando la respuesta del servidor es demasiado grande para caber en un paquete DNS UDP. Estas reglas serían utilizadas tanto por un servidor de nombres de reenvío como por un cliente estándar:

```
$IPT -A EXT-salida -p tcp \  
--sport $UNPRIVPORTS --dport 53 \  
-j local-dns-servidor-consulta
```

```
$IPT A Entrada EXT -p tcp ! --syn \  
--deporte 53 --dport $UNPRIVPORTS\  
-j respuesta-servidor-dns-remoto
```

Lo que sigue son las cadenas definidas por el usuario que contienen las reglas ACCEPT y DROP reales.

local-dns-server-query y remote-dns-server-response Estas dos cadenas definidas por el usuario, local-dns-server-query y remote-dns server-response, realizan la determinación final sobre el paquete.

La cadena local-dns-server-query selecciona los paquetes de solicitud salientes en función de la dirección de destino del servidor remoto. Para esta cadena, debe definir los servidores de nombres que le gustaría usar:

```

NAMESERVER_1="su.nombre.servidor"
NAMESERVER_2="su.servidor.de.nombre.secundario"
NAMESERVER_3="su.servidor.de.nombre.terciario"

# Servidor de nombres de reenvío de DNS o solicitudes de clientes si [ "$
CONNECTION_TRACKING" = "1" ]; luego
    $IPT -Una consulta-servidor-dns-local \
        -d $SERVIDOR DE
        NOMBRES_1 \ -m estado --estado NUEVO -j ACCEPTAR

    $IPT -Una consulta-servidor-dns-local \
        -d $SERVIDOR DE
        NOMBRES_2 \ -m estado --estado NUEVO -j ACCEPTAR

    $IPT -Una consulta-servidor-dns-local \
        -d $SERVIDOR DE
        NOMBRES_3 \ -m estado --estado NUEVO -j ACCEPTAR
ser

$IPT -A local-dns-server-query \ -d
    $NAMESERVER_1 -j ACCEPT

$IPT -A local-dns-server-query \ -d
    $NAMESERVER_2 -j ACCEPT

$IPT -A local-dns-server-query \ -d
    $NAMESERVER_3 -j ACCEPT

```

La cadena de respuesta del servidor dns remoto selecciona los paquetes de respuesta entrantes en función de la dirección de origen del servidor remoto:

```

# Respuestas del servidor DNS a solicitudes locales
$IPT -A respuesta-servidor-dns-remoto \ -s
    $SERVIDOR_NOMBRE_1 -j ACCEPTAR

$IPT -A respuesta-servidor-dns-remoto \ -s
    $SERVIDOR_NOMBRE_2 -j ACCEPTAR

$IPT -A respuesta-servidor-dns-remoto \ -s
    $SERVIDOR_NOMBRE_3 -j ACCEPTAR

```

Tenga en cuenta que las reglas finales seleccionan solo la dirección IP del servidor remoto. Las reglas de llamada en las cadenas de entrada EXT y salida EXT ya han coincidido en los campos de encabezado UDP o TCP. No es necesario volver a realizar esas pruebas de coincidencia.

solicitud-cliente-dns-local y respuesta -servidor-dns-remoto

Estas dos cadenas definidas por el usuario, local-dns-client-request y remote-dns server-response, realizan la determinación final sobre los paquetes intercambiados entre los clientes TCP locales y los servidores remotos.

La cadena local-dns-client-request selecciona los paquetes de solicitud salientes en función de la dirección y el puerto de destino del servidor remoto. La respuesta del servidor DNS remoto chain selecciona los paquetes de respuesta entrantes en función de la dirección y el puerto de origen del servidor remoto.

Tráfico de clientes locales a través de TCP

El siguiente par de reglas coincide con el tráfico estándar de clientes locales a servidores remotos a través de TCP:

```
$IPT -A EXT-salida -p tcp \
    --deporte $UNPRIVPORTS \
    -j solicitud-cliente-tcp-local

$IPT -A Entrada EXT -p tcp ! --syn \
    --dport $UNPRIVPORTS \
    -j respuesta-servidor-tcp-remoto
```

Recuerde que estas reglas normalmente no se prueban cuando se usa el módulo de estado, con la excepción de la primera solicitud SYN saliente.

La referencia específica al protocolo TCP es necesaria en las siguientes reglas, aunque la regla de llamada coincida con el campo de protocolo, porque se especifica el puerto de origen o de destino. Este es un requisito sintáctico de iptables. También tenga en cuenta que debe definir los hosts de origen y destino dentro de estas reglas, como lo indica el <host seleccionado> y otras llamadas similares. Además, si utiliza estas reglas, asegúrese de definir variables para las que elija, como POP_SERVER, MAIL_SERVER, NEWS_SERVER, etc. El siguiente código habilita el tráfico TCP desde clientes locales:

Salida del cliente TCP local y cadenas de entrada del servidor remoto

#cliente SSH

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A solicitud-cliente-tcp-local -p tcp \
        -d <host seleccionado> --dport 22 \
        -m estado --estado NUEVO \
        -j ACEPTAR
```

ser

```
$IPT -A solicitud-cliente-tcp-local -p tcp \
    -d <host seleccionado> --dport 22 \
    -j ACEPTAR
```

```
$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \
    -s <host seleccionado> --sport 22 \
    -j ACEPTAR
```

160 Capítulo 6 Optimización de cortafuegos

```
# Reglas de cliente para solicitudes de control HTTP, HTTPS y FTP if
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A local-tcp-client-request -p tcp -m multiport --destination-
        port 80,443,21 \ --syn -m state --state NEW \ -j ACCEPT
```

```
ser
$IPT -A solicitud-cliente-tcp-local -p tcp \ -m multipuerto --puerto-
    destino 80,443,21 \ -j ACEPTAR
```

```
$IPT -A respuesta-servidor-tcp-remoto -p tcp \ -m multiport --source-
    port 80,443,21 ! --syn \ -j ACEPTAR
```

```
# cliente POP if
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A solicitud-cliente-tcp-local -p tcp \ -d $SERVIDOR_POP
        --dport 110 \ -m estado --estado NUEVO \ -j
        ACEPTAR
```

```
ser
```

```
$IPT -A solicitud-cliente-tcp-local -p tcp \ -d $SERVIDOR_POP
    --dport 110 \ -j ACEPTAR
```

```
$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \ -s $SERVIDOR_POP --
    sport 110 \ -j ACEPTAR
```

```
# Cliente de correo SMTP
if [ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A solicitud-cliente-tcp-local -p tcp \ -d
        $SERVIDOR_CORREO --dport 25 \ -m estado --
        estado NUEVO \ -j ACEPTAR
```

```
ser
```

```
$IPT -A solicitud-cliente-tcp-local -p tcp \ -d
    $SERVIDOR_CORREO --dport 25 \ -j ACEPTAR
```

```
$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \ -s
    $SERVIDOR_DE_CORREO --sport 25 \ -j ACEPTAR
```

```
# Cliente de noticias Usenet if
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A local-tcp-client-request -p tcp \ -d $NEWS_SERVER --
        dport 119 \ -m state --state NEW \ -j ACEPTAR
```

```
ser
```

```
$IPT -A local-tcp-client-request -p tcp \ -d $NEWS_SERVER --
    dport 119 \ -j ACEPTAR
```

```
$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \ -s $NOTICIAS_SERVIDOR
--sport 119 \ -j ACEPTAR
```

```
# Cliente FTP - conexión de canal de datos en modo pasivo if
[ "$CONNECTION_TRACKING" = "1" ]; luego
$IPT -A local-tcp-client-request -p tcp \ --dport $UNPRIVPORTS
\ -m state --state NEW \ -j ACEPTAR
```

ser

```
$IPT -A solicitud-cliente-tcp-local -p tcp \ --dport
$UNPRIVPORTS -j ACEPTAR
```

```
$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \ --sport $UNPRIVPORTS
-j ACEPTAR
```

Tráfico del servidor local a través de TCP

El siguiente par de reglas coincide con el tráfico de servidor local estándar a clientes remotos a través de TCP.

Estos serían aplicables solo si realmente está ofreciendo servicios a hosts remotos:

```
$IPT -A Entrada EXT -p tcp \ --sport
$UNPRIVPORTS \ -j solicitud-
cliente-tcp-remoto
```

```
$IPT -A Salida EXT -p tcp ! --syn \ --dport
$UNPRIVPORTS \ -j local-tcp-server-
response
```

El siguiente par de reglas maneja las conexiones de canales de datos entrantes desde servidores FTP remotos cuando se usa el modo de puerto:

```
# Kludge para conexiones de canales de datos FTP entrantes # desde
servidores remotos usando el modo de puerto.
# Los módulos de estado tratan esta conexión como RELACIONADA # si el
módulo ip_conntrack_ftp está cargado.
```

```
$IPT -A Entrada EXT -p tcp \ --sport
20 --dport $UNPRIVPORTS \ -j ACEPTAR
```

```
$IPT -A Salida EXT -p tcp ! --syn \ --sport
$UNPRIVPORTS --dport 20 \ -j ACEPTAR
```

solicitud-cliente-tcp-remoto y respuesta -servidor-tcp-local Estas dos cadenas definidas por el usuario, solicitud-cliente-tcp-remoto y respuesta -servidor-tcp-local, realizan la determinación final sobre los paquetes intercambiados entre clientes TCP remotos y clientes locales . servidores.

La cadena remote-tcp-client-request selecciona los paquetes de solicitud entrantes en función de la dirección y el puerto de origen del cliente remoto. La cadena de respuesta del servidor TCP local

selecciona los paquetes de respuesta salientes en función de la dirección y el puerto de destino del cliente remoto:

```
# Entrada del cliente TCP remoto y cadenas de salida del servidor local

# servidor SSH si
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A remote-tcp-client-request -p tcp \ -s <host seleccionado> --
        destination-port 22 \ -m state --state NEW \ -j ACCEPTAR

ser

$IPT -A solicitud-cliente-tcp-remota -p tcp \
    -s <host seleccionado> --destination-port 22 \ -j ACCEPTAR

$IPT -A local-tcp-server-response -p tcp ! --syn \
    --source-port 22 -d <host seleccionado> \ -j ACCEPTAR

# Servidor de identificación AUTH
$IPT -A solicitud-cliente-tcp-remoto -p tcp \ --destination-port 113 \ -j
    REJECT --reject-with tcp-reset
```

Tráfico de clientes locales a través de UDP

El siguiente par de reglas coincide con el tráfico estándar de clientes locales a servidores remotos a través de UDP:

```
# Cliente UDP local, servidor remoto
$IPT -A Salida EXT -p udp \ --sport
    $UNPRIVPORTS \ -j solicitud-cliente-
    udp-local

$IPT -A Entrada EXT -p udp \ --dport
    $UNPRIVPORTS \ -j respuesta-servidor-
    udp-remoto
```

Omitir la verificación de la dirección de origen sin utilizar el módulo de estado Si no está utilizando el módulo de estado, la mayoría de las reglas de TCP podrían colocarse antes que las reglas de suplantación de dirección de origen. TCP mantiene la información del estado de la conexión por sí mismo. Solo la primera solicitud de conexión entrante, el primer paquete SYN, requiere la verificación de la dirección de origen. Puede hacerlo reorganizando las reglas y dividiendo la regla para el tráfico de clientes entrantes en dos pruebas, una para el indicador SYN inicial y otra para todos los indicadores ACK posteriores.

Usando las reglas para un servidor web local como ejemplo, la primera regla seguiría las reglas de suplantación de identidad:

```
if [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A solicitud-cliente-tcp-remoto -p tcp \ --destination-port 80 \ -m state
        --state NEW \ -j ACCEPTAR
```

```
demás
    $IPT -A solicitud-cliente-tcp-remota -p tcp --syn \
        --destino-puerto 80 \ -j ACEPTAR
```

ser

Las siguientes dos reglas precederían a las reglas de suplantación de identidad:

```
$IPT -A ENTRADA -p tcp ! --syn \ --source-
    port $UNPRIVPORTS \ -d $IPADDR --
    destination-port 80 \ -j ACEPTAR
```

```
$IPT -A SALIDA -p tcp ! --syn \ -s $IPADDR --
    puerto-origen 80 \ --puerto-destino
    $UNPRIVPORTS \ -j ACEPTAR
```

solicitud-cliente-udp-local y respuesta -servidor-udp-remoto Estas dos cadenas definidas por el usuario, solicitud-cliente-udp-local y respuesta -servidor-udp-remoto, realizan la determinación final sobre los paquetes intercambiados entre clientes UDP locales y clientes remotos . servidores.

La cadena local-udp-client-request selecciona los paquetes de solicitud salientes en función de la dirección y el puerto de destino del servidor remoto. La cadena de respuesta del servidor UDP remoto selecciona los paquetes de respuesta entrantes en función de la dirección y el puerto de origen del servidor remoto. Asegúrese de definir la variable TIME_SERVER antes de implementar esta regla:

```
# Cliente de tiempo NTP
if [ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A solicitud-cliente-udp-local -p udp \ -d $SERVIDOR_HORA
        --dport 123 \ -m estado --estado NUEVO \ -j
        ACEPTAR
```

ser

```
$IPT -A solicitud-cliente-udp-local -p udp \ -d $SERVIDOR_HORA
    --dport 123 \ -j ACEPTAR
```

```
$IPT -A respuesta-servidor-udp-remoto -p udp \
    -s $SERVIDOR_HORA --sport 123 \ -j
    ACEPTAR
```

Tráfico ICMP

Finalmente, el último par de reglas coincide con el tráfico ICMP entrante y saliente:

```
# tráfico ICMP
$IPT -A Entrada-EXT -p icmp -j Entrada-EXT-icmp

$IPT -A Salida-EXT -p icmp -j Salida-icmp-EXT
```

EXT-icmp-in y EXT-icmp-out Estas dos

cadena definidas por el usuario, EXT-icmp-in y EXT-icmp-out, realizan la determinación final sobre los paquetes ICMP intercambiados entre el host local y las máquinas remotas.

La cadena EXT-icmp-in selecciona los paquetes ICMP entrantes según el tipo de mensaje. La cadena EXT-icmp-out selecciona los paquetes ICMP salientes según el tipo de mensaje:

```
# Registrar y soltar fragmentos ICMP iniciales
$IPT -A EXT-icmp-in --fragment -j LOG \ --log-prefix "ICMP
    entrante fragmentado:"

$IPT -A EXT-icmp-entrada --fragmento -j DROP

$IPT -A EXT-icmp-out --fragment -j LOG \ --log-prefix "ICMP
    saliente fragmentado:"

$IPT -A EXT-icmp-out --fragmento -j DROP

# Ping saliente si
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A EXT-icmp-out -p icmp \ --icmp-type
        echo-request \ -m state --state NEW \
        -j ACEPTAR

ser

$IPT -A EXT-icmp-salida -p icmp \
    --petición de eco de tipo icmp -j ACEPTAR

$IPT -A EXT-icmp-entrada -p icmp \
    --icmp-type eco-respuesta -j ACEPTAR

# Ping entrante

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A EXT-icmp-in -p icmp \ -s $MI_ISP
        \ --icmp-type echo-request \
        -m state --state NEW \ -j ACEPTAR

ser

$IPT -A EXT-icmp-in -p icmp \ --icmp-type
    echo-request \ -s $MY_ISP -j ACEPTAR

$IPT -A EXT-icmp-out -p icmp \ --icmp-type
    eco-respuesta \ -d $MI_ISP -j
    ACEPTAR

# Destino inalcanzable Tipo 3
$IPT -A EXT-icmp-out -p icmp \ --icmp-type
    fragmentación necesaria -j ACEPTAR
```



```

$IPT -A EXT-icmp-in -p icmp \ --icmp-type destino-
in alcanzable -j ACEPTAR

# Problema de parámetro
$IPT -A EXT-icmp-salida -p icmp \
--icmp-type parámetro-problema -j ACEPTAR

$IPT -A EXT-icmp-in -p icmp \ --icmp-type
parámetro-problema -j ACEPTAR

# Tiempo excedido
$IPT -A EXT-icmp-entrada -p icmp \
--icmp-type time-exceeded -j ACCEPT

# Fuente para enfriar
$IPT -A EXT-icmp-salida -p icmp \
--icmp-type source-quench -j ACEPTAR

$IPT -A EXT-icmp-entrada -p icmp \
--icmp-type source-quench -j ACEPTAR

```

tcp-state-flags La cadena tcp-

state-flags es la primera cadena definida por el usuario que adjuntará a las cadenas de ENTRADA y SALIDA integradas . Las pruebas coinciden con las combinaciones de indicadores de estado de TCP que se crean artificialmente y, a menudo, se usan en escaneos sigilosos:

```

# Todos los bits se borran
$IPT -A tcp-state-flags -p tcp --tcp-flags TODO NINGUNO -j log-tcp-state

# SYN y FIN están configurados
$IPT -A tcp-state-flags -p tcp --tcp-flags SYN,FIN SYN,FIN -j log-tcp-state

# SYN y RST están configurados
$IPT -A tcp-state-flags -p tcp --tcp-flags SYN,RST SYN,RST -j log-tcp-state

# FIN y RST están configurados
$IPT -A tcp-state-flags -p tcp --tcp-flags FIN,RST FIN,RST -j log-tcp-state

# FIN es el único bit establecido, sin el esperado ACK que lo acompaña
$IPT -A tcp-state-flags -p tcp --tcp-flags ACK,FIN FIN -j log-tcp-state

# PSH es el único conjunto de bits, sin el esperado ACK que lo acompaña
$IPT -A tcp-state-flags -p tcp --tcp-flags ACK,PSH PSH -j log-tcp-state

# URG es el único conjunto de bits, sin el ACK que se espera que lo acompañe
$IPT -A tcp-state-flags -p tcp --tcp-flags ACK,URG URG -j log-tcp-state

```

log-tcp-state La cadena

log-tcp-state se usa por dos razones. En primer lugar, el mensaje de registro tiene como prefijo un mensaje explicativo específico y, dado que se trata de un paquete diseñado, se notifican las opciones de IP o TCP. En segundo lugar, el paquete coincidente se descarta inmediatamente. Los dos generalizados

las cadenas de registro que aparecen más tarde se escriben bajo el supuesto de que la política predeterminada eliminará los paquetes registrados inmediatamente después de que regresen de las cadenas:

```
$IPT -A registro-tcp-estado -p tcp -j REGISTRO \
    --log-prefix "Estado TCP ilegal: " \
    --log-ip-opciones --log-tcp-opciones
```

```
$IPT -A log-tcp-state -j DROP
```

seguimiento de conexión

La cadena de seguimiento de conexión es la segunda cadena definida por el usuario que adjuntará a las cadenas de ENTRADA y SALIDA integradas. Los paquetes coincidentes eluden las reglas del cortafuegos y se aceptan de inmediato:

```
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    # Omita los filtros de firewall para intercambios establecidos
    $IPT -A seguimiento de conexión -m estado \
        --estado ESTABLECIDO, RELACIONADO \
        -j ACEPTAR

    $IPT -A seguimiento de conexión -m estado --estado NO VÁLIDO \
        -j LOG --log-prefix "paquete NO VÁLIDO:"
    $IPT -A seguimiento de conexión -m estado --estado NO VÁLIDO -j DROP
ser
```

consulta-cliente-dhcp-local y respuesta -servidor-dhcp-remoto

Las cadenas local-dhcp-client-query y remote-dhcp-server-response contienen las reglas requeridas de un cliente DHCP. La ubicación de estas reglas en la jerarquía de la cadena es importante en relación con cualquier suplantación de identidad o reglas de transmisión generalizadas. Además, el host no configurará su dirección IP hasta después de recibir el mensaje de compromiso DHCPACK del servidor. La dirección de destino que utiliza el servidor en el mensaje DHCPACK depende de la implementación del servidor en particular. Si desea utilizar esta regla, deberá establecer DHCP_CLIENT en 1 y también definir la variable DHCP_SERVER :

```
# El cortafuegos ignora explícitamente algunos paquetes de difusión.
# Otros son eliminados por la política predeterminada.
# Las pruebas de DHCP deben preceder a las reglas relacionadas con la transmisión, ya que DHCP se basa
# en el tráfico de transmisión inicialmente.
```

```
si [ "$DHCP_CLIENT" = "1" ]; luego
    DHCP_SERVER="mi.dhcp.servidor"

    # Inicialización o reenlace: Sin concesión o Tiempo de concesión vencido.

    $IPT -A local-dhcp-client-consulta \
        -s $ BROADCAST_SRC \
        -d $ BROADCAST_DEST -j ACEPTAR

    # DHCP OFFER entrante de los servidores DHCP disponibles
```

```
$IPT -A respuesta-del-servidor-dhcp-remoto \ -s
      $BROADCAST_SRC \ -d
      $BROADCAST_DEST -j ACCEPT

# Regresar a la inicialización # El cliente
# conoce su servidor, pero ha perdido su concesión, # o necesita volver a confirmar la
# dirección IP después de reiniciar.

$IPT -A local-dhcp-client-query \ -s
      $BROADCAST_SRC \ -d
      $DHCP_SERVER -j ACEPTAR

$IPT -A respuesta-del-servidor-dhcp-remoto \ -s
      $DHCP_SERVER \ -d
      $BROADCAST_DEST -j ACCEPT

# Como resultado de lo anterior, se supone que debemos cambiar nuestra dirección IP
# con este mensaje, que está dirigido a nuestra nueva # dirección antes de que el
# cliente dhcp haya recibido la actualización.
# Según la implementación del servidor, la dirección de destino # puede ser la nueva dirección
# IP, la dirección de subred o la # dirección de difusión limitada.

# Si la dirección de la subred de la red se usa como destino, # la siguiente regla debe
# permitir los paquetes entrantes destinados a la # dirección de la subred, y la regla debe
# preceder a cualquier regla general # que bloquee dichos paquetes de transmisión entrantes.

$IPT -A respuesta-del-servidor-dhcp-remoto \ -s
      $DHCP_SERVER -j ACCEPT

# Renovación de arrendamiento

$IPT -A local-dhcp-client-query \ -s $IPADDR \
      -d $DHCP_SERVER -j ACEPTAR
```

ser

verificación de la dirección de origen

La cadena de comprobación de dirección de origen comprueba si hay direcciones de origen ilegales identificables. La cadena está unida a la cadena INPUT sola. Las reglas del cortafuegos garantizan que los paquetes generados por este host contengan su dirección IP como dirección de origen. Tenga en cuenta que estas reglas necesitarán algunos ajustes si el host tiene más de una interfaz de red o si una LAN privada usa direcciones IP de clase privada.

Un cliente DHCP necesita manejar el tráfico de difusión relacionado con DHCP antes de realizar estas pruebas:

```
# Descartar paquetes que fingen ser originarios de la interfaz receptora
$IPT -A verificación-dirección-fuente -s $IPADDR -j DROP
```

```
# Rechazar paquetes que afirman ser de redes privadas
```

168 Capítulo 6 Optimización del cortafuegos

```

$IPT -A verificación-dirección-fuente -s $CLASS_A -j DROP
$IPT -A verificación-dirección-fuente -s $CLASS_B -j DROP
$IPT -A verificación-dirección-fuente -s $CLASS_C -j DROP
$IPT -A verificación-dirección-fuente -s $CLASS_D_MULTICAST -j DROP
$IPT -A verificación-dirección-fuente -s $CLASS_E_RESERVED_NET -j DROP
$IPT -A verificación-dirección-fuente -s $LOOPBACK -j DROP

$IPT -A verificación-dirección-fuente -s 0.0.0.0/8 -j DROP $IPT -A verificación-
dirección-fuente -s 169.254.0.0/16 -j DROP $IPT -A verificación-dirección-fuente -s 192.0
.2.0/24 -j CAÍDA

```

Comprobación de la dirección de destino

La cadena de comprobación de direcciones de destino comprueba los paquetes de difusión, las direcciones de multidifusión mal utilizadas y los puertos de servicio no privilegiados conocidos. La cadena está unida a las cadenas de ENTRADA y SALIDA. Un cliente DHCP necesita manejar el tráfico de difusión relacionado con DHCP antes de realizar estas pruebas:

```

# Bloquear transmisiones dirigidas desde Internet

$IPT -A verificación de dirección de destino $BROADCAST_DEST -j DROP $IPT -A verificación
de dirección de destino -d $SUBNET_BASE -j DROP $IPT -A verificación de dirección de destino
-d $SUBNET_BROADCAST -j DROP $IPT -A destino-dirección-comprobación! -puppet

-d $CLASS_D_MULTICAST -j DROP

# Evitar puertos sujetos a problemas de protocolo y administración del sistema

# puertos TCP sin privilegios
# Denegar solicitudes de conexión a puertos NFS, SOCKS y X Window
$IPT -A verificación-dirección-destino -p tcp -m multipuerto \ --puerto-destino

$NFS_PORT,$OPENWINDOWS_PORT,$SOCKS_PORT,$SQUID_PORT \
--syn -j CAÍDA

$IPT -A verificación de dirección de destino -p tcp --syn \
--destination-port $XWINDOW_PORTS -j DROP

# puertos UDP sin privilegios
# Denegar solicitudes de conexión a NFS y puertos bloqueados
$IPT -A verificación-dirección-destino -p udp -m multipuerto \
--destination-port $NFS_PORT,$LOCKD_PORT -j DROP

```

Registro de paquetes descartados con iptables Las cadenas EXT-

log-in y EXT-log-out contienen las reglas que registran los paquetes inmediatamente antes de que los paquetes caigan al final de sus respectivas cadenas y se descarten por la política predeterminada. Casi todos los paquetes salientes que se descartan se registran porque indican un problema en las reglas del firewall o un servicio desconocido (o no autorizado) que intenta comunicarse con el mundo exterior:

```
# reglas ICMP

$IPT -A EXT-iniciar sesión -p icmp \! --icmp-type
    echo-request -m limite -j LOG

# reglas TCP

$IPT -A EXT-iniciar sesión -p tcp \ --dport 0:19
    -j LOG

# Saltar ftp, telnet, ssh
$IPT -A EXT-iniciar sesión -p tcp \ --dport 24 -j
    LOG

# Saltar smtp
$IPT -A EXT-iniciar sesión -p tcp \ --dport 26:78
    -j LOG

# Saltar dedo, www
$IPT -A EXT-iniciar sesión -p tcp \ --dport
    81:109 -j LOG

# Saltar pop-3, sunrpc
$IPT -A EXT-iniciar sesión -p tcp \ --dport
    112:136 -j LOG

# Saltar NetBIOS
$IPT -A EXT-iniciar sesión -p tcp \ --dport
    140:142 -j LOG

# Saltar imagen
$IPT -A EXT-iniciar sesión -p tcp \ --dport
    144:442 -j LOG

# Omitir web_segura/SSL
$IPT -A EXT-iniciar sesión -p tcp \ --dport
    444:65535 -j REGISTRO

#reglas UDP

$IPT -A EXT-iniciar sesión -p udp \ --dport
    0:110 -j LOG

# Saltar sunrpc
$IPT -A EXT-iniciar sesión -p udp \ --dport
    112:160 -j LOG

# Saltar snmp
$IPT -A EXT-iniciar sesión -p udp \ --dport
    163:634 -j LOG

# Omitir NFS montado
$IPT -A EXT-iniciar sesión -p udp \ --dport
    636:5631 -j LOG
```

170 Capítulo 6 Optimización del cortafuegos

```
# Saltar pcAnywhere
$IPT -A EXT-iniciar sesión -p udp \ --
    dport 5633:31336 -j REGISTRO

# Omitir los puertos predeterminados de traceroute
$IPT -A EXT-iniciar sesión -p udp \ --
    sport $TRACEROUTE_SRC \ --dport
    $TRACEROUTE_DEST -j LOG

# Saltar el resto
$IPT -A EXT-iniciar sesión -p udp \
    --dport 33434:65535 -j REGISTRO

# paquetes salientes

# No registrar paquetes de destino inalcanzables ICMP salientes rechazados
$IPT -A EXT-cierre de sesión -p icmp \
    --icmp-type destino-inalcanzable -j DROP

$IPT -A EXT-cierre de sesión -j LOG
```

El script optimizado de nftables

El script nftables aprovecha los archivos de reglas externas adicionales en la sintaxis de nftables :

- **ñ nft-vars:** contiene variables relacionadas con el script, definidas en formato nftables en lugar de formato de shell
- **ñ setup-tables:** contiene el filtro principal y la arquitectura de la tabla nat , incluidos Cadenas de ENTRADA y SALIDA
- **ñ política de host local:** contiene reglas para el tráfico de host local .
- **ñ connectionstate-policy:** establece las políticas de estado de conexión ñ
- **invalid-policy:** establece políticas relacionadas con el tráfico no válido ñ **dns-policy:** contiene políticas relacionadas con búsquedas de DNS ñ **tcp-client-policy:** contiene reglas relacionadas con conexiones de clientes salientes
- **ñ tcp-server-policy:** contiene reglas de conexiones entrantes relacionadas si el com
la computadora actúa como un servidor
- **ñ icmp-policy:** contiene las solicitudes ICMP relacionadas con las reglas ñ **log-policy:** contiene las reglas relacionadas con el registro
- **ñ default-policy:** contiene las reglas de política predeterminadas finales para el cortafuegos

Inicialización del cortafuegos

La secuencia de comandos del cortafuegos comienza definiendo la ubicación del comando de administración del cortafuegos nftables :

```
#!/bin/sh
```

```
NFT = "/usr/local/sbin/nft"
```

```
# Ubicación de nft en su sistema
```

También se establecieron una serie de parámetros del núcleo; consulte el Capítulo 5 para obtener una explicación de estos parámetros:

```
# Habilitar protección de eco de transmisión eco 1
> /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Deshabilite los paquetes enrutados de origen
para f en /proc/sys/net/ipv4/conf/*/accept_source_route; hacer echo 0 > $f hecho

# Habilitar protección de cookies TCP SYN echo 1
> /proc/sys/net/ipv4/tcp_syncookies
# Deshabilitar la aceptación de redirección ICMP
para f en /proc/sys/net/ipv4/conf/*/accept_redirects; hacer
    echo 0 > $f
hecho

# No envíe mensajes de redirección para f
en /proc/sys/net/ipv4/conf/*/send_redirects; hacer
    echo 0 > $f done
# Soltar paquetes
falsificados que ingresan a una interfaz, los cuales, si se responden, # darían como resultado que la
respuesta saliera por una interfaz diferente. para f en /proc/sys/net/ipv4/conf/*/rp_filter; do echo 1 > $f
done # Registra paquetes con direcciones imposibles. para f en /proc/sys/net/ipv4/conf/*/log_martians; haz
    echo 1 > $f hecho
```

La primera parte del script restablece y elimina las cadenas existentes, como se muestra en el Capítulo 5:

```
para i en 'tablas de lista de $ NFT | awk '{imprimir $2}'
hacer
    echo "Enjuagar ${i}"
    $NFT vaciar la tabla ${i} para j
    en '$NFT list table ${i} | cadena grep | awk '{imprimir $2}'
    hacer
        echo "...Eliminando la cadena ${j} de la tabla ${i}"
        $NFT eliminar cadena ${i} ${j} hecho

    echo "Eliminando ${i}"
    $NFT eliminar tabla ${i} hecho
```

Aquí está el código final para el comienzo de la secuencia de comandos del cortafuegos, es decir, el código para permitir que el cortafuegos se detenga fácilmente. Con este código colocado debajo del código anterior, cuando llame a la secuencia de comandos con un argumento de "detener" , la secuencia de comandos vaciará, borrará y restablecerá las políticas predeterminadas, y el firewall se detendrá de manera efectiva.

```
if [ "$1" = "stop" ] luego echo
";Firewall completamente
detenido! ADVERTENCIA: ESTE HOST NO TIENE FIREWALL FUNCIONANDO". salir 0 fi
```

Ahora se vuelven a crear las tablas:

```
$NFT -f tablas de configuración
$NFT -f política de host local
$NFT -f política de estado de conexión
```

Creación de los archivos de reglas

Las siguientes secciones muestran los archivos que contienen las reglas de nftables para los diversos componentes del cortafuegos. Los archivos de reglas usan variables definidas por nftables que se incluyen dentro de cada archivo de reglas y se alojan en un archivo llamado nft-vars. El archivo nft-vars creará a medida que se agreguen reglas. Para empezar, el archivo nft-vars contiene lo siguiente:

```
define int_loopback = lo
define int_internet = eth0
define ip_external = <su IP externa>
define subnet_external = <su subred externa>
define net_loopback = 127.0.0.0/8
define net_class_a = 10.0.0.0/8
define net_class_b = 172.16.0.0/16
define net_class_c = 192.168.0.0/16
define net_class_d = 224.0.0.0/4
define net_class_e = 240.0.0.0/5
define broadcast_src = 0.0.0.0
define broadcast_dest = 255.255.255.255
define ports_priv = 0-1023
define ports_unpriv = 1024-65535
```

Creación de tablas

Las reglas de configuración de tablas crean las tablas de filtro y nat, crean cadenas de ENTRADA y SALIDA para cada una, y conectan esas cadenas en sus respectivos ganchos dentro de nftables para que las cadenas reciban paquetes. Las reglas de las tablas de configuración son:

```
incluir filtro de tabla "nft-
vars" { entrada de cadena {

    tipo filtro gancho entrada prioridad 0;

} cadena de salida {
    tipo filtro gancho salida prioridad 0;
}
}
```

Habilitación del tráfico de host

La habilitación de la comunicación de host local tiene lugar en el archivo de reglas de política de host local. Tenga en cuenta el uso de una variable definida por nftables (\$int_loopback) en este ejemplo:

```
incluir filtro de tabla "nft-
vars" { entrada de cadena {

    iifname $int_loopback aceptar

} cadena de salida {
    oifname $int_loopback aceptar
}
}
```


Habilitación del estado de conexión

El seguimiento del estado de conexión viene dentro del archivo de reglas de política de estado de conexión:

```
incluir filtro de tabla "nft-
vars" { entrada de cadena {

    estado de ct establecido, relacionado aceptar prefijo de
    registro no válido de estado de ct "Entrada NO VÁLIDA:" velocidad límite 3/segundo ycaída

} cadena de salida {
    estado de ct establecido, relacionado aceptar prefijo de
    registro no válido de estado de ct "salida NO VÁLIDA: ysegunda gota" " tasa límite 3/

}
}
```

Eliminación de tráfico no válido

Las reglas para eliminar el tráfico no válido están contenidas en un archivo de reglas llamado política no válida:

```
incluir filtro de tabla "nft-
vars" { entrada de cadena {

    iif $int_internet ip saddr $ip_external drop iif $int_internet ip saddr
    $net_class_a drop iif $int_internet ip saddr $net_class_b drop iif $int_internet
    ip saddr $net_class_c drop iif $int_internet ip protocol udp ip daddr
    $net_class_d accept iif $int_internet ip saddr $net_class_e drop iif
    $int_internet ip saddr $net_loopback drop iif $int_internet ip daddr $subnet_external drop

}
}
```

Habilitación del tráfico DNS

Las reglas para identificar el tráfico DNS son lo primero. Hasta que se instalen las reglas de DNS, su software de red no podrá ubicar servicios y hosts en Internet a menos que use la dirección IP.

Para estas reglas, se agregan tres nuevas variables al archivo nft-vars :

```
define nameserver_1 = <la ip de su servidor de nombres> define
nameserver_2 = <ip del segundo servidor de nombres> define nameserver_3
= <ip del tercer servidor de nombres, si es necesario>
```

El archivo dns-policy se crea para contener las siguientes reglas y luego se carga en el script principal rc.firewall con el comando nft -f dns-policy:

```
incluir filtro de tabla "nft-
vars" { entrada de cadena {

    ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } udp sport y53 udp dport 53 accept ip daddr
    { $nameserver_1,$nameserver_2,$nameserver_3 } tcp sport y53 tcp dport $ports_unpriv accept
```

```

        ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } udp sport ¡53 udp dport $ports_unpriv
        aceptar
    }
    salida en cadena {
        ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } udp sport ¡53 udp dport 53 aceptar

        ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } tcp sport ¡$ports_unpriv tcp dport 53
        aceptar
        ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } udp sport ¡$ports_unpriv udp dport 53
        aceptar
    }
}

```

La primera regla en las cadenas de ENTRADA y SALIDA coincide con las consultas del caché local y el servidor de nombres de reenvío, si tiene uno, y las respuestas del servidor DNS remoto. El servidor local está configurado como esclavo del servidor principal remoto, por lo que el servidor local fallará si la búsqueda no se realiza correctamente. Esta configuración es menos común para una pequeña oficina/oficina en el hogar.

Las siguientes reglas en cada una de las cadenas de ENTRADA y SALIDA coinciden en el cliente DNS estándar solicitudes de búsqueda a través de TCP, cuando la respuesta del servidor es demasiado grande para caber en un paquete DNS UDP. Estas reglas serían utilizadas tanto por un servidor de nombres de reenvío como por un cliente estándar.

Tráfico de clientes locales a través de TCP

La conexión desde su computadora a Internet a través de TCP se puede lograr agregando reglas de salida específicas para los servidores y servicios a los que desea conectarse. Como antes, las reglas mostradas para este propósito asumen el uso del módulo de seguimiento de estado. Si el seguimiento de estado no está habilitado, se debe agregar una regla de espejo a la cadena INPUT para habilitar el tráfico que regresa para una conexión determinada.

Las reglas requieren la adición de una variable `server_smtp` al programa `nft-vars` :

```
define server_smtp = <su servidor SMTP>
```

Las reglas van a un archivo llamado `tcp-client-policy` que luego se carga en el programa `rc.firewall` con el comando

```
nft -f política-cliente-tcp
```

Aquí están las reglas para `tcp-client-policy`:

```

incluir "nft-vars"
filtro de tabla {
    entrada en cadena {
    }
    salida en cadena {
        tcp dport {21,22,80,110,143,993,995,443} tcp sport $ports_unpriv ¡aceptar ip daddr $server_smtp
        tcp dport 25 tcp sport $ports_unpriv aceptar
    }
}

```

Tráfico del servidor local a través de TCP

Habilitar los servicios de su servidor local para permitir que los clientes se conecten con usted se logra agregando reglas específicas a la cadena de ENTRADA . Idealmente, podría limitar las conexiones a clientes conocidos y luego agregar una regla para la dirección de origen (ip saddr <la ip de su cliente>), pero en el mundo real eso normalmente no es posible.

El siguiente archivo se carga en el script rc.firewall con

```
nft -f política-servidor-tcp
```

El archivo de reglas llamado tcp-server-policy contiene lo siguiente para habilitar las conexiones al puerto 22 (SSH) en el servidor local desde cualquier cliente:

```
incluir "nft-vars"
filtro de tabla {
    entrada de cadena {
        ip daddr $ip_external tcp sport $ports_unpriv tcp dport {22} aceptar
    }
    salida en cadena {
    }
}
```

Tráfico ICMP

Finalmente, el último par de reglas coincide con el tráfico ICMP entrante y saliente. Estas reglas se cargan en un archivo llamado icmp-policy:

```
incluir "nft-vars"
filtro de tabla {
    entrada de cadena {
        tipo icmp { respuesta de eco, destino inalcanzable, parámetro- y problema, extinción de
        la fuente, tiempo excedido} aceptar
    }
    salida en cadena {
        tipo icmp { petición de eco, problema de parámetro, extinción de fuente} aceptar
    }
}
```

Luego, el archivo se agrega al script principal rc.firewall con el comando

```
nft -f icmp-política
```

Registro de paquetes descartados con nftables

Las reglas finales para los paquetes de registro del cortafuegos que no han sido manejados por otras reglas cargadas previamente. Casi todos los paquetes salientes que se descartan se registran porque indican un problema en las reglas del firewall o un servicio desconocido (o no autorizado) que intenta comunicarse con el mundo exterior.

El archivo se llama log-policy y contiene las siguientes reglas. Cargue el archivo justo antes a la política predeterminada con el comando

```
nft -f política de registro
```

Estas son las reglas:

```
incluir "nft-vars"
filtro de tabla {
    entrada de cadena {
        prefijo de registro "Paquete de ENTRADA descartado:" velocidad límite 3/segundo
    }
    salida en cadena {
        prefijo de registro "Paquete de SALIDA descartado:" velocidad límite 3/segundo
    }
}
```

¿Qué compró la optimización?

El objetivo de la optimización es hacer que el paquete pase por el procesamiento del filtro lo más rápido posible, con la menor cantidad posible de pruebas innecesarias. Idealmente, desea que los paquetes fluyan a la velocidad de la línea.

En términos del propio firewall, tres factores afectan el rendimiento: la cantidad de reglas instaladas en el kernel; la longitud del recorrido de la cadena, o el número de reglas con las que se prueba cualquier paquete antes de que coincida; y el número total de pruebas de coincidencia realizadas en el paquete. Además, cuando se trata de usar el módulo de estado, recuerde que la compensación es la velocidad frente a la memoria.

Tenga en cuenta que algunas de las variaciones en las tablas son artefactos de cómo se organizan los scripts de muestra en el Capítulo 5 y en este capítulo, así como algunas diferencias en el indicador de estado TCP y las verificaciones de dirección de origen entre los dos ejemplos.

Optimización de iptables

¡Para iptables, las versiones optimizadas tienen más reglas que sus contrapartes directas! ¡Aún más sorprendente, las versiones de seguimiento de conexión tienen más reglas que las versiones clásicas sin estado! ¿No llegamos ya a la conclusión de que el uso del módulo de coincidencia de estado reduce la cantidad de reglas al eliminar las reglas ACCEPT individuales para las respuestas del servidor a las solicitudes de los clientes? Si y no. . . . El número absoluto de reglas aumenta, porque las reglas estáticas deben permanecer presentes para dar cuenta de cualquier caso en el que una entrada de la tabla de estado haya expirado o haya sido reemplazada debido a una escasez de recursos. El número de reglas de entrada *atravesadas* puede caer drásticamente.

El uso de cadenas definidas por el usuario también da como resultado algunas reglas más. Las reglas adicionales realizan la selección y ramificación de paquetes intermedios. Hay una pequeña cantidad de gastos generales en las decisiones de bifurcación de nivel superior. La sobrecarga no es significativa, e incluso la pequeña cantidad de sobrecarga aparente es engañosa. La cantidad de reglas *atravesadas* no es una métrica de rendimiento tan crítica como la cantidad de pruebas de coincidencia de campos de encabezado individuales realizadas.

La optimización con cadenas definidas por el usuario puede reducir significativamente la cantidad de reglas con las que se debe probar un paquete de respuesta antes de llegar a su regla de coincidencia final. Lo que no es obvio aquí es que el conjunto de reglas directas incluye una cantidad determinada de gastos generales debido a las reglas contra la suplantación de identidad. Debido a que los cortafuegos de ejemplo estaban centrados en el cliente, el

Las longitudes transversales directas del servidor son mucho más largas que las longitudes del cliente como resultado de la verificación de direcciones realizada en los paquetes entrantes.

El uso del módulo de estado y, por lo tanto, eludir el cortafuegos para el tráfico establecido reduce drásticamente la longitud transversal de la lista para las conexiones establecidas. El aumento en el número de reglas atravesadas por nuevas conexiones es el resultado de las reglas duplicadas, las reglas de seguimiento de conexiones y sus contrapartes estáticas.

Incluso con el módulo de estado, el paquete inicial siempre toma la ruta estática. Por lo tanto, la ganancia que muestra la versión optimizada clásica sobre la versión directa aún se aplica al primer paquete.

Finalmente, ¡los beneficios tanto de la optimización como del seguimiento del estado de la conexión son obvios! La categorización que realizan las cadenas definidas por el usuario reduce drásticamente la cantidad real de pruebas contra las que se prueba un paquete en un firewall de filtrado de paquetes clásico. El uso de los módulos de estado reduce aún más la cantidad de pruebas al omitir las reglas por completo. Además, la conexión del canal de datos se compara inmediatamente como una conexión RELACIONADA .

Lo que quizás sea menos obvio, a menos que haya leído el código del cortafuegos del núcleo, es que el verdadero problema de rendimiento no es el número de reglas atravesadas per se, sino el número de pruebas de comparación realizadas. Cada regla atravesada no coincidente representa al menos una comparación (por ejemplo, ¿el paquete entrante es un paquete ICMP o un paquete TCP, o está llegando a la interfaz de bucle invertido o alguna otra interfaz específica?). Las cadenas definidas por el usuario permiten dividir las comparaciones al dividirse en cadenas dedicadas en puntos críticos de decisión de comparación.

Optimización de tablas nft

Para el script de nftables , gran parte del procesamiento se trasladó a archivos de reglas de nftables nativos que luego fueron cargados por el script de shell. El beneficio de esta optimización es que acerca las reglas al lugar donde se procesan sin tener que ejecutar un comando de shell para todas y cada una de las reglas individuales.

Las reglas también se dividen lógicamente en varios archivos de políticas, cada uno de los cuales contiene reglas para un tráfico similar. Esto permite un mantenimiento más sencillo y puede condensarse o expandirse según sea necesario, según su situación particular.

Resumen

El Capítulo 5 introdujo iptables y nftables siguiendo los pasos para construir un firewall simple para un sistema independiente. Este capítulo analizó las ideas detrás de la optimización del firewall y luego creó cadenas definidas por el usuario para optimizar el ejemplo del firewall del Capítulo 5. Finalmente, se examinaron los efectos de la optimización, así como los efectos del uso del módulo de estado.

Esta página se dejó en blanco intencionalmente

Reenvío de paquetes

Esta El capítulo cubre algunos de los problemas básicos subyacentes a la seguridad de LAN, el reenvío de firewalls de puerta de enlace y las redes perimetrales. Las políticas de seguridad se definen en relación con el nivel de necesidades de seguridad del sitio, la importancia o el valor de los datos que se protegen y el costo de la pérdida de datos o la privacidad. Este capítulo comienza con una revisión de los firewalls presentados en capítulos anteriores y luego analiza los problemas que el creador de políticas del sitio debe abordar al elegir la ubicación del servidor y determinar las políticas de seguridad.

Es posible que necesite traducción de direcciones de red (NAT) para acceder a Internet desde máquinas. NAT no se analiza hasta el Capítulo 8, "NAT: traducción de direcciones de red". Este capítulo se centra únicamente en el reenvío.

Para los lectores familiarizados con ipchains o ipfwadm, el reenvío y NAT se combinaron sintácticamente. Ambas funciones fueron especificadas por una sola regla directa. Estas funciones lógicamente distintas son claramente distintas en iptables y nftables. De hecho, las dos funciones son manejadas por tablas separadas con cadenas separadas. NAT se aplica por separado en un punto diferente en la ruta transversal del paquete a través del sistema. Este capítulo se centra en los servicios de iptables disponibles en la tabla de filtros y en sus extensiones y las funciones de reenvío de nftables. El Capítulo 8 analiza los servicios relacionados con NAT.

Las limitaciones de un firewall independiente

El cortafuegos de un solo sistema presentado en el Capítulo 5, "Creación e instalación de un cortafuegos independiente", es un cortafuegos bastión básico, que usa solo cadenas básicas en la tabla de filtros. Cuando el cortafuegos es un enrutador de filtrado de paquetes que tiene una interfaz de red conectada a Internet y otra conectada a su LAN (denominado *sistema de doble conexión*), el cortafuegos aplica reglas para decidir si reenvía o bloquea los paquetes que cruzan entre las dos interfaces. En este caso, el firewall de filtrado de paquetes es un enrutador estático con reglas de filtrado de tráfico que hacen cumplir las políticas locales sobre qué paquetes se permiten a través de las interfaces de red.

Como se señaló en el Capítulo 3, "iptables: el programa de administración de firewall heredado de Linux", Netfilter maneja los paquetes reenviados de manera bastante diferente al mecanismo IPFW anterior. Los paquetes reenviados son inspeccionados solo por la cadena FORWARD. Las reglas de ENTRADA y SALIDA no se aplican. El tráfico de red relacionado con el host del cortafuegos local y el tráfico de red relacionado con la LAN tienen conjuntos de reglas y cadenas de reglas completamente diferentes.

180 Capítulo 7 Reenvío de paquetes

Las reglas de la cadena FORWARD pueden especificar tanto la interfaz de entrada como la de salida. Para una configuración de host dual con una LAN, las reglas de firewall aplicadas a las interfaces de red entrantes y salientes representan un par de E/S: una regla para los paquetes que llegan y una regla inversa para los paquetes que salen. Las reglas son direccionales. Las dos interfaces se manejan como una unidad.

El tráfico no se enruta directamente entre Internet y la LAN automáticamente. Paquetes para ser reenviado no fluirán sin un par de reglas para aceptar el tráfico. Las reglas de filtrado aplicadas a las dos interfaces actúan como un cortafuegos y un enrutador estático entre las dos redes.

La configuración de firewall presentada en el Capítulo 5 suele ser adecuada para un individuo sistema doméstico individual con una sola interfaz de red.

Como un servidor de seguridad de puerta de enlace independiente que protege una LAN, si la máquina del servidor de seguridad alguna vez se ve comprometida, se acabó. Incluso si las interfaces locales del cortafuegos tienen políticas completamente diferentes a las del tráfico reenviado, si el sistema ha sido comprometido, no pasará mucho tiempo antes de que el intruso obtenga acceso a la raíz. En ese punto, si no antes, los sistemas internos también están completamente abiertos. Lo más probable es que una LAN doméstica nunca tenga que enfrentarse a esta situación si los servicios ofrecidos a Internet se eligen con cuidado y se aplica una estricta política de cortafuegos. Aún así, un firewall de puerta de enlace independiente representa un único punto de falla. Es una situación de todo o nada.

Muchas organizaciones y corporaciones más grandes confían en una única configuración de firewall y muchas otras usan una de otras dos arquitecturas: una arquitectura de host filtrado sin acceso directo enrutamiento o una arquitectura de subred filtrada con servicios de proxy, junto con una red perimetral DMZ creada entre o junto al firewall externo, separada de la LAN privada. Los servidores públicos en la red DMZ también tienen sus propios firewalls de bastión especializados. Esto significa que estos sitios tienen muchas más computadoras a su disposición— y personal para gestionarlos.

DMZ: una red perimetral con cualquier otro nombre

Una red perimetral entre dos cortafuegos se denomina zona desmilitarizada (DMZ). El propósito de una DMZ es establecer un espacio protegido desde el cual ejecutar servidores (o servicios) públicos y aislar ese espacio del resto de la LAN privada. Si un servidor en la DMZ se ve comprometido, ese servidor permanece aislado de la LAN; los cortafuegos de puerta de enlace y los cortafuegos de bastión que se ejecutan en los otros servidores DMZ ofrecen protección contra los servidores comprometidos servidor.

Además del cortafuegos independiente de un solo sistema, el cortafuegos presentado en el Capítulo 5 se puede ampliar para formar la base de un cortafuegos de puerta de enlace de doble alojamiento que protege al host, que ofrece uno o unos pocos servicios públicos. Una LAN doméstica a menudo está protegida por un único servidor de seguridad de puerta de enlace que filtra el tráfico reenviado y ofrece servicios públicos.

¿Qué opciones están disponibles para un sistema de doble alojamiento que no puede permitirse el riesgo de un firewall de puerta de enlace única o el costo de muchas computadoras y el personal para administrarlas? Afortunadamente, un cortafuegos de doble alojamiento y una LAN ofrecen una mayor seguridad cuando el sistema se configura con cuidado. La pregunta es la siguiente: ¿Vale la pena el esfuerzo extra de mantener el cortafuegos y aumentar la seguridad en un entorno de confianza?

Configuraciones básicas de firewall de puerta de enlace

Aquí se utilizan dos configuraciones básicas de cortafuegos de puerta de enlace. Como se muestra en la Figura 7.1, la puerta de enlace tiene dos interfaces de red: una conectada a Internet y otra conectada a la DMZ.

Los servicios públicos de Internet se ofrecen desde máquinas en la red DMZ. El cortafuegos de la puerta de enlace no ofrece ningún servicio. Un segundo cortafuegos, un cortafuegos de estrangulación, también está conectado a la red DMZ, separando las redes privadas internas de las máquinas de servidor casi públicas en la red perimetral. Las máquinas privadas están protegidas detrás del cortafuegos de estrangulamiento en la LAN interna. Además, cada uno de los servidores de la DMZ ejecuta su propio firewall especializado. Si falla el cortafuegos de la puerta de enlace o uno de los servidores, las máquinas del servidor público en la DMZ continúan ejecutando sus cortafuegos individuales. El firewall de estrangulamiento protege la LAN interna de una puerta de enlace comprometida o de cualquier otra máquina comprometida en la red perimetral. El tráfico entre la LAN e Internet pasa a través de ambos cortafuegos y cruza la red perimetral.

En la segunda configuración, la puerta de enlace tiene tres interfaces de red: una conectada a Internet, otra conectada a la DMZ y otra conectada a la LAN privada. Como se muestra en la Figura 7.2, el tráfico entre la LAN e Internet y el tráfico entre la DMZ e Internet no comparten nada excepto la interfaz de red externa de la puerta de enlace.

Una ventaja de esta configuración sobre la primera es que ni la LAN ni la DMZ comparten la carga de tráfico de ambas redes. Otra ventaja es que es más fácil definir reglas que se refieran específicamente a todo el tráfico LAN o DMZ, a diferencia del tráfico relacionado con la otra red. Otra ventaja es que un host de puerta de enlace única es menos costoso que dos dispositivos de firewall separados.

La desventaja de esta configuración sobre la primera es que la puerta de enlace se convierte en un único punto de falla para ambas redes. Además, las reglas de firewall en el host único incluyen

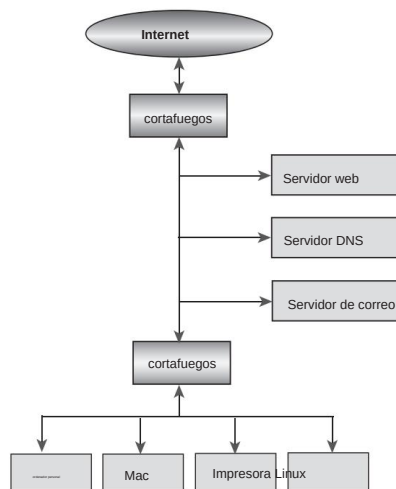


Figura 7.1 Una DMZ entre una puerta de enlace de doble alojamiento y un cortafuegos de estrangulamiento

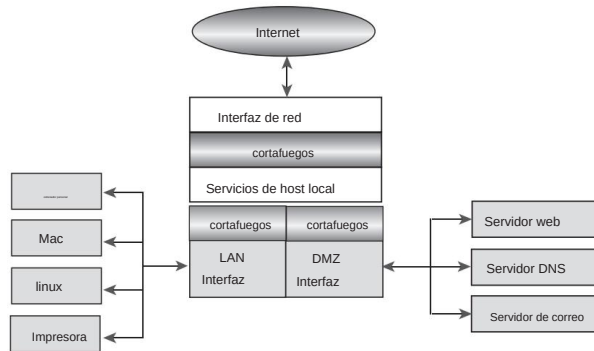


Figura 7.2 Un firewall tri-homed que separa una LAN y una DMZ

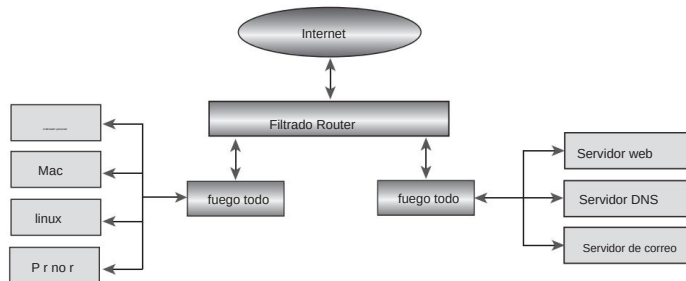


Figura 7.3 Un enrutador de filtrado frente a los cortafuegos LAN y DMZ

toda la complejidad relacionada tanto con la DMZ como con la LAN. Esta complejidad puede volverse confusa cuando desarrolla reglas de firewall a mano.

Una tercera alternativa común es agregar un enrutador de filtrado que separe el tráfico LAN y DMZ. Los servidores DMZ ejecutan sus propios firewalls de bastión. Puede haber o no un cortafuegos generalizado entre el enrutador y la DMZ. Como se muestra en la Figura 7.3, el cortafuegos de la puerta de enlace está separado del enrutador y protege la LAN. El enrutador de filtrado realiza parte del filtrado básico tanto para la LAN como para la DMZ. El cortafuegos de la puerta de enlace no necesita proporcionar este filtrado básico y funciona de forma similar al cortafuegos de estrangulación en la primera configuración.

Problemas de seguridad de LAN

Los problemas de seguridad dependen en gran medida del tamaño de la LAN, su arquitectura y para qué se utiliza. Los servicios y la arquitectura también están influenciados por el direccionamiento IP público disponible para el sitio. Quizás aún más básico que eso es el tipo de conexión a Internet que tiene el sitio: acceso telefónico, DSL, inalámbrico, cable, satélite, ISDN, línea alquilada o cualquiera de los otros.

tipos de conexiones a Internet. Las siguientes son algunas preguntas que debe tener en cuenta al crear una política de seguridad para su sitio.

¿Se asigna dinámica y temporalmente una dirección IP pública a través de DHCP o IPCP? ¿Lo hace el sitio tiene una sola dirección IP pública asignada permanentemente o un bloque de ellas?

¿Se ofrecen servicios a Internet? ¿Estos servicios están alojados en la máquina del cortafuegos o están alojados en máquinas internas? Por ejemplo, puede ofrecer un servicio de correo electrónico desde la máquina del servidor de seguridad de la puerta de enlace, pero servir un sitio web desde una máquina interna en la DMZ.

Cuando los servicios se alojan desde máquinas internas, desea colocar esas máquinas en una red perimetral y aplicarles políticas de acceso y filtrado de paquetes completamente diferentes. Si los servicios se ofrecen desde máquinas internas, ¿este hecho es visible para el exterior, o los servicios se envían mediante proxy o de forma transparente a través de NAT para que parezcan estar disponibles desde la máquina del cortafuegos?

¿Cuánta información desea poner a disposición del público sobre las máquinas en tu LAN? ¿Tiene la intención de alojar servicios DNS locales? ¿Están disponibles en Internet los contenidos de la base de datos DNS local?

¿Las personas pueden iniciar sesión en sus máquinas desde Internet? ¿Cuántas y qué máquinas locales son accesibles para ellos? ¿Todas las cuentas de usuario tienen los mismos derechos de acceso? ¿Se enviarán las conexiones entrantes para un control de acceso adicional?

¿Todas las máquinas internas son igualmente accesibles para los usuarios locales y desde todas las máquinas locales? ¿Los servicios externos son igualmente accesibles desde todas las máquinas internas? Por ejemplo, si utiliza una arquitectura de servidor de seguridad de host filtrado, los usuarios deben iniciar sesión en la máquina del servidor de seguridad directamente para tener acceso a Internet. No se realizaría ningún enrutamiento.

¿Los servicios de LAN privada se ejecutan detrás del firewall? Por ejemplo, ¿se usa NFS internamente, o Samba, o una impresora en red? ¿Necesita evitar que alguno de estos servicios filtre información o transmita tráfico a Internet, como SNMP, DHCP o ntpd? El mantenimiento de dichos servicios detrás del cortafuegos de estrangulamiento secundario garantiza el aislamiento completo de estos servicios de Internet.

En relación con los servicios diseñados para el uso de LAN, hay preguntas sobre el acceso local frente al externo a los servicios diseñados para el uso de Internet. ¿Ofrecerá FTP internamente pero no externamente, o posiblemente ofrecerá diferentes tipos de servicios FTP para ambos? ¿Ejecutará un servidor web privado o configurará diferentes partes del mismo sitio para que estén disponibles para los usuarios locales en lugar de los usuarios remotos? ¿Ejecutará un servidor de correo local para enviar correo pero utilizará un mecanismo diferente para recuperar el correo entrante de Internet (es decir, su correo se entregará directamente a las cuentas de usuario de su máquina, o recuperará explícitamente el correo de un ISP)?

Opciones de configuración para una LAN doméstica de confianza

Debe considerar dos tipos de tráfico de red interna. El primer tipo es el acceso local a

el firewall de la puerta de enlace, a través de la interfaz interna, como se muestra en la Figura 7.4. El segundo es el acceso local a Internet, a través de la interfaz externa de la máquina de puerta de enlace.

Presuntamente, la mayoría de los sistemas pequeños no tienen motivos para filtrar paquetes entre el firewall y la red local en general. Sin embargo, debido a que a la mayoría de los sitios basados en el hogar se les asigna un

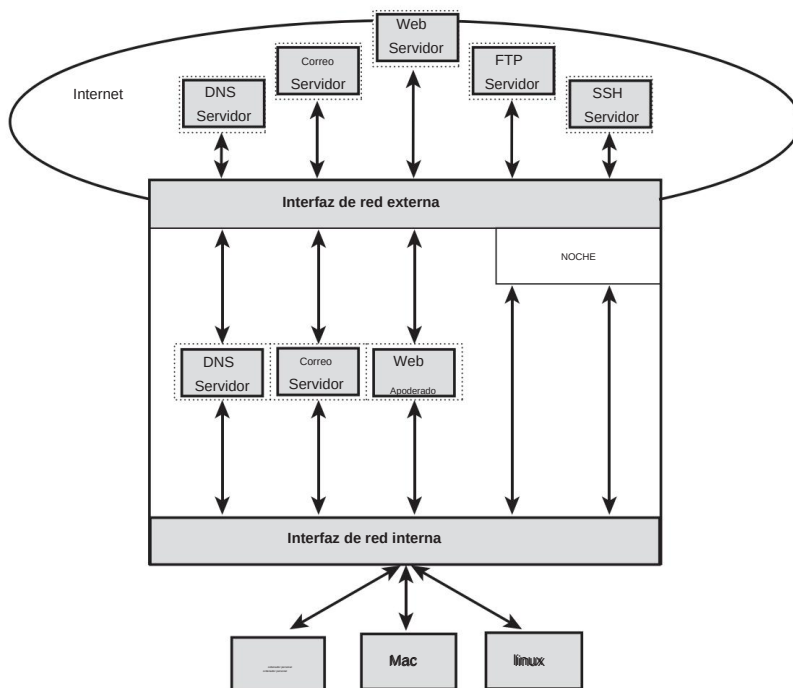


Figura 7.4 Tráfico de LAN a la máquina de firewall y a Internet

única dirección IP, surge una excepción: NAT. Presumiblemente, la única acción relacionada con el filtrado interno que debe realizar será habilitar su propia forma de suplantación de dirección de origen mediante la aplicación de paquetes NAT que se mueven entre sus máquinas internas e Internet. La mayor parte del énfasis está en el filtrado de paquetes entre el firewall e Internet.

¿Qué tan confiables son las "LAN domésticas confiables"?

Aunque a los sitios residenciales y de pequeñas empresas a menudo les gusta ver sus redes como "confiables", este no suele ser el caso. El problema no son los usuarios locales, sino la alta tasa de incidencia de compromiso entre estos sistemas.

Acceso LAN al cortafuegos de la puerta de enlace

En un entorno doméstico, es muy probable que desee habilitar el acceso sin restricciones entre las máquinas LAN y el firewall de la puerta de enlace. (Algunos padres tienen motivos para no estar de acuerdo).

La suposición en esta sección es que todos los servicios públicos están alojados en el firewall. Los hosts LAN son puramente máquinas cliente. La LAN puede iniciar conexiones con el cortafuegos, pero el cortafuegos no puede iniciar conexiones con la LAN. Habrá

excepciones a esta regla general. Es posible que desee que el servidor de seguridad tenga acceso a una impresora local en red, por ejemplo. (Un sitio comercial nunca tomaría esta decisión. El cortafuegos estaría tan protegido contra los problemas que se originan en la LAN como lo está contra los problemas que se originan en la Internet pública).

Comenzando con el firewall desarrollado en el Capítulo 5 como base, se necesitan constantes adicionales en el ejemplo del firewall para hacer referencia a la interfaz interna que se conecta a la LAN. Este ejemplo define la interfaz de red interna como eth1; la LAN se define como que incluye direcciones Clase C que van desde 192.168.1.0 a 192.168.1.255; una interfaz externa, la que da al mundo exterior, se define como eth0:

```
LAN_INTERFACE="eth1"
EXTERNAL_INTERFACE="eth0"
DIRECCIONES_LAN="192.168.1.0/24"
```

Permitir el acceso sin restricciones a través de las interfaces es una simple cuestión de permitir todos los protocolos y todos los puertos de forma predeterminada. Tenga en cuenta que la LAN puede iniciar nuevas conexiones a servidores remotos, pero no se aceptan nuevas conexiones entrantes desde sitios remotos. Estas son las reglas de iptables :

```
$IPT -A ADELANTE -i $LAN_INTERFACE -o $EXTERNAL_INTERFACE \
    -p tcp -s $LAN_ADDRESSES --sport $UNPRIVPORTS \
    -m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

```
$IPT -A ADELANTE -i $EXTERNAL_INTERFACE -o $LAN_INTERFACE \
    -m estado --estado ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

Aquí hay reglas similares de nftables :

```
$NFT agregar filtro de regla adelante si $LAN_INTERFACE o si $EXTERNAL_INTERFACE \ ip protocol tcp
    ip saddr $LAN_ADDRESSES tcp sport $UNPRIVPORTS \
    ct estado nuevo,establecido,relacionado aceptar
$NFT agregar filtro de regla adelante si $EXTERNAL_INTERFACE o si $LAN_INTERFACE ct estado yestablecido,
relacionado aceptar
```

Observe también que estas dos reglas reenvían el tráfico. No afectan al tráfico local entre la LAN y el propio cortafuegos. Para acceder a los servicios en el host del cortafuegos, también se necesitan reglas de ENTRADA y SALIDA locales:

```
$IPT -A ENTRADA -i $LAN_INTERFACE \
    -p tcp -s $LAN_ADDRESSES --sport $UNPRIVPORTS \
    -m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

```
$IPT -A SALIDA -o $LAN_INTERFACE \
    -m estado --estado ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

Las reglas de nftables son:

```
$NFT agregar entrada de filtro de regla si $LAN_INTERFACE \
    protocolo ip tcp ip saddr $LAN_ADDRESSES \
    tcp sport $UNPRIVPORTS ct estado nuevo,establecido,relacionado aceptar
$NFT agregar salida de filtro de regla o si $LAN_INTERFACE ct estado establecido, relacionado yaceptar
```

Tanto las reglas de reenvío como las de interfaz interna podrían ser tan específicas del servicio como las reglas de interfaz externa del Capítulo 5. En el mundo actual, las reglas de interfaz interna y de reenvío deberían ser así de específicas. Las reglas en esta sección simplemente sientan las bases, introduciendo las reglas de reenvío en sí mismas.

Acceso de LAN a otras LAN: reenvío de tráfico local entre varias LAN

Si las máquinas en su LAN, o en varias LAN, requieren enrutamiento entre ellas, debe permitir el acceso entre las máquinas para los puertos de servicio que requieren, a menos que tengan rutas de conexión internas alternativas. En el primer caso, cualquier enrutamiento local realizado entre LAN sería realizado por el cortafuegos.

La suposición en esta sección es que hay un cortafuegos de puerta de enlace con dos interfaces de red, una red de servidor DMZ, un cortafuegos de estrangulación interno con dos interfaces de red y la red privada LAN. Esta es la configuración que se muestra anteriormente en la Figura 7.1. El tráfico entre la LAN e Internet cruza a través de la red DMZ entre los cortafuegos de estrangulamiento y puerta de enlace. Esta configuración es común en sitios más pequeños.

Este ejemplo cambia el nombre de la interfaz de red interna en la puerta de enlace como DMZ_INTERFACE. Se necesita otra constante para el cortafuegos. La DMZ se define como la inclusión de direcciones privadas de Clase C que van desde 192.168.3.0 a 192.168.3.255:

```
DMZ_INTERFACE="eth1"
DMZ_DIRECCIONES="192.168.3.0/24"
```

Las dos primeras reglas siguientes permiten el acceso local al servidor de seguridad de la puerta de enlace desde la LAN. En la práctica, la LAN no tendría acceso a todos los puertos del cortafuegos. Las segundas dos reglas permiten que el propio cortafuegos acceda a servicios específicos ofrecidos en la DMZ servidor por servidor. Una vez más, un cortafuegos en un entorno más grande tendría pocas o ninguna razón para acceder a los servicios alojados en la DMZ. En la mayoría de los casos, el servidor de seguridad no ofrece ningún servicio a la DMZ. En sitios más grandes, es probable que el firewall tampoco ofrezca ningún servicio a la LAN:

```
$IPT -A ENTRADA -i $DMZ_INTERFACE -s $LAN_ADDRESSES -d $GATEWAY \
    -m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACCEPTAR

$IPT -A SALIDA -o $DMZ_INTERFACE -s $GATEWAY -d $LAN_ADDRESSES \
    -m estado --estado ESTABLECIDO, RELACIONADO -j ACCEPTAR

$IPT -A SALIDA -o $DMZ_INTERFACE -s $GATEWAY -d $DMZ_ADDRESSES \
    -m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACCEPTAR

$IPT -A ENTRADA -i $DMZ_INTERFACE -s $DMZ_ADDRESSES -d $GATEWAY \
    -m estado --estado ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

Las reglas de nftables correspondientes son:

```
$NFT añadir entrada de filtro de regla si $DMZ_INTERFACE ip saddr $LAN_ADDRESSES ip daddr y$GATEWAY ct
estado nuevo, establecido, relacionado aceptar
$NFT agregar salida de filtro de regla oif $DMZ_INTERFACE ip saddr $GATEWAY ip daddr
y$LAN_ADDRESSES ct estado establecido, relacionado aceptar
```

```
$NFT agregar salida de filtro de regla oif $DMZ_INTERFACE ip saddr $GATEWAY ip daddr
y$DMZ_ADDRESSES ct estado nuevo, establecido, relacionado aceptar
$NFT añadir entrada de filtro de regla si $DMZ_INTERFACE ip saddr $DMZ_ADDRESSES ip daddr y$GATEWAY ct
estado establecido, relacionado aceptar
```

Las siguientes reglas reenvían el tráfico entre las redes internas e Internet. El tráfico de DMZ y LAN se maneja por separado. El tráfico DMZ representa las solicitudes de conexión entrantes de Internet. El tráfico LAN representa las solicitudes de conexión salientes a Internet. Nuevamente, en la práctica, las reglas de DMZ serían muy específicas por dirección de servidor y servicio:

```
$IPT -A ADELANTE -i $EXTERNAL_INTERFACE -o $DMZ_INTERFACE \
-d $DIRECCIONES_DMZ \
-m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACEPTAR
```

```
$IPT -A ADELANTE -i $DMZ_INTERFACE -o $EXTERNAL_INTERFACE \
-s $DIRECCIONES_DMZ \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

```
$IPT -A ADELANTE -i $DMZ_INTERFACE -o $EXTERNAL_INTERFACE \
-s $DIRECCIONES_LAN \
-m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACEPTAR
```

```
$IPT -A ADELANTE -i $EXTERNAL_INTERFACE -o $DMZ_INTERFACE \
-d $DIRECCIONES_LAN \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

Las reglas de nftables son:

```
$NFT agregar filtro de regla adelante si $EXTERNAL_INTERFACE oif $DMZ_INTERFACE ip daddr y$DMZ_ADDRESSES
ct estado nuevo, establecido, relacionado aceptar
$NFT agregar filtro de regla adelante si $DMZ_INTERFACE o si $EXTERNAL_INTERFACE ip saddr y$DMZ_ADDRESSES
ct estado establecido, relacionado aceptar
$NFT agregar filtro de regla adelante si $DMZ_INTERFACE o si $EXTERNAL_INTERFACE ip saddr y$LAN_ADDRESSES
ct estado nuevo, establecido, relacionado aceptar
$NFT agregar filtro de regla adelante si $EXTERNAL_INTERFACE o si $DMZ_INTERFACE ip daddr y$LAN_ADDRESSES
ct estado establecido, relacionado aceptar
```

Tenga en cuenta que las reglas de reenvío anteriores para la DMZ no están completas. Los servidores en la DMZ a veces también inician conexiones salientes, como solicitudes de conexión desde un servidor proxy web o un servidor de puerta de enlace de correo.

En el cortafuegos de estrangulamiento, las siguientes reglas reenvían el tráfico entre la LAN y la DMZ redes. Tenga en cuenta que la LAN puede iniciar nuevas conexiones, pero no se aceptan nuevas conexiones entrantes desde la DMZ o desde Internet a la LAN. Nuevamente, en la práctica, la LAN tendría un acceso más controlado a la DMZ, así como al firewall de la puerta de enlace, suponiendo que la puerta de enlace proporcionara algún servicio:

```
$IPT -A ADELANTE -i $LAN_INTERFACE -o $DMZ_INTERFACE \
-s $DIRECCIONES_LAN \
-m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACEPTAR
```

```
$IPT -A ADELANTE -i $DMZ_INTERFACE -o $LAN_INTERFACE \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

188 Capítulo 7 Reenvío de paquetes

Las reglas de nftables son:

```
$NFT agregar filtro de regla hacia adelante si $LAN_INTERFACE o si $DMZ_INTERFACE ip saddr  
y$LAN_ADDRESSES ct estado nuevo, establecido, relacionado aceptar  
$NFT agregar filtro de regla adelante si $DMZ_INTERFACE o si $LAN_INTERFACE ct estado yestablecido,  
relacionado aceptar
```

Opciones de configuración para mayor o menor LAN de confianza

Una empresa u organización, y muchos sitios domésticos, utilizarían mecanismos más elaborados y específicos que las reglas de cortafuegos de reenvío simples y genéricas presentadas en las dos secciones anteriores para una LAN doméstica de confianza. En entornos menos confiables, las máquinas con firewall están tan protegidas de los usuarios internos como de los usuarios externos.

Las reglas de firewall específicas del puerto se definen tanto para las interfaces internas como para las externas. Las reglas internas pueden ser una imagen reflejada de las reglas para las interfaces externas, o las reglas pueden ser más inclusivas. Lo que se permite a través de la interfaz de red interna de la máquina del cortafuegos de estrangulación depende de los tipos de sistemas que se ejecutan en la LAN y los tipos de servicios locales que se ejecutan en la DMZ, así como también qué servicios de Internet son accesibles para la LAN de acuerdo con las políticas de seguridad locales. .

Por ejemplo, es posible que desee bloquear los mensajes de difusión local para que no lleguen al cortafuegos de la puerta de enlace. Si no todos sus usuarios son completamente confiables, es posible que desee restringir lo que pasa al cortafuegos de estrangulamiento desde las máquinas internas con tanta fuerza como lo que entra desde Internet. Además, debe mantener la cantidad de cuentas de usuario al mínimo en la máquina de firewall. Idealmente, un firewall no tiene cuentas de usuario, con la excepción de una única cuenta administrativa sin privilegios.

Una empresa en el hogar puede tener una sola dirección IP, lo que requiere la traducción de direcciones de red LAN. Sin embargo, las empresas suelen arrendar varias direcciones IP registradas públicamente o un bloque completo de direcciones de red. Las direcciones públicas generalmente se asignan a los servidores públicos de una empresa. Con direcciones IP públicas, las conexiones salientes se reenvían y las conexiones entrantes se enrutan normalmente. Se puede definir una subred local para crear una DMZ pública local.

División del espacio de direcciones para crear múltiples redes

Las direcciones IP se dividen en dos partes: una dirección de red y una dirección de host dentro de esa red. Como se indicó en el Capítulo 1, "Conceptos preliminares que subyacen a los cortafuegos de filtrado de paquetes", las direcciones de clase A, B y C son algo así como un artefacto, pero siguen siendo las direcciones más fáciles de usar como ejemplos porque sus campos de red y host se encuentran en byte. límites.

Las direcciones de red de clase A, B y C se definen por sus primeros 8, 16 y 24 bits, respectivamente. Dentro de cada clase de dirección, los bits restantes definen la parte del host de la dirección IP. Esto se muestra visualmente en la Tabla 7.1.

La división en subredes es una extensión local de la dirección de red que forma parte de las direcciones IP locales. Una máscara de red local se define como aquella que trata algunas de las direcciones de host más importantes.

Tabla 7.1 Campos de red y host en una dirección IP

	Clase A	Clase B	Clase C
Bits de red principales	0	10	110
campo de red	1 byte	2 bytes	3 bytes
campo anfitrión	3 bytes	2 bytes	1 byte
Prefijo de red	/8	/16	/24
Rango de direcciones	1–126	128–191	192–223
Máscara de red	255.0.0.0	255.255.0.0	255.255.255.0

bits como si fueran parte de la dirección de red. Estos bits de dirección de red adicionales sirven para definir múltiples redes localmente. Los sitios remotos no conocen las subredes locales. Ven el rango de direcciones como direcciones normales de clase A, B o C.

Por ejemplo, tomemos el bloque de direcciones privadas de Clase C 192.168.1.0. La dirección base, conocida como dirección de red, es 192.168.1.0 para este ejemplo. La máscara de red para este ejemplo es 255.255.255.0, que coincide exactamente con los primeros 24 bits, la dirección de red, de la red 192.168.1.0/24.

Esta red se puede dividir en dos redes locales definiendo los primeros 25 bits, en lugar de los primeros 24 bits, como la porción de red de la dirección. En el lenguaje actual, decimos que la red local tiene una longitud de prefijo de 25 en lugar de 24. El bit más significativo del campo de dirección del host ahora se trata como parte del campo de dirección de red. El campo de host ahora contiene 7 bits en lugar de 8. La máscara de red se convierte en 255.255.255.128 o /25 en notación CIDR. Se definen dos subredes: 192.168.1.0, que se dirige a los hosts del 1 al 126, y 192.168.1.128, que se dirige a los hosts del 129 al 254. Cada subred pierde dos direcciones de host porque cada subred usa la dirección de host más baja, 0 o 128, como red, y utiliza la dirección de host más alta, 127 o 255, como dirección de transmisión. La tabla 7.2 muestra esto en forma tabular.

Las subredes 192.168.1.0 y 192.168.1.128 se pueden asignar a dos tarjetas de interfaz de red interna separadas. Cada subred consta de dos redes independientes, cada una de las cuales contiene hasta 126 hosts.

Tabla 7.2 Red Clase C **192.168.1.0** dividida en dos subredes

Número de subred	Ninguna	0	1
Dirección de red	192.168.1.0	192.168.1.0	192.168.1.128
Máscara de red	255.255.255.0	255.255.255.128	255.255.255.128
Primera dirección de host	192.168.1.1	192.168.1.1	192.168.1.129
Última dirección del anfitrión	192.168.1.254	192.168.1.126	192.168.1.254
Dirección de Difusión	192.168.1.255	192.168.1.127	192.168.1.255
Anfitriones totales	254	126	126

Tabla 7.3 Red Clase C **192.168.1.0** dividida en cuatro subredes

subred número 0	1	2	3
La red dirección 192.168.1.0	192.168.1.64	192.168.1.128	192.18.1.192
La red máscara 255.255.255.192	255.255.255.192	255.255.255.192	255.255.255.192
primer anfitrión dirección 192.168.1.1	192.168.1.65	192.168.1.129	192.168.1.193
último anfitrión dirección 192.168.1.62	192.168.1.126	192.168.1.190	192.168.1.254
Transmisión dirección 192.168.1.63	192.168.1.127	192.168.1.191	192.168.1.255
Total			
Hospedadores 62	62	62	62

La división en subredes permite la creación de múltiples redes internas, cada una de las cuales contiene diferentes diferentes clases de máquinas cliente o servidor y cada una con su propio enrutamiento independiente.

A continuación, se pueden aplicar diferentes políticas de cortafuegos a las redes.

Por supuesto, este ejemplo mostró que la red se dividía en dos partes. De hecho, la red se puede dividir en muchas partes para crear varias redes más pequeñas. Es bastante común ver una red con una máscara de subred de 255.255.255.252

o /30 utilizado entre enrutadores en dos ubicaciones. La Tabla 7.3 lleva el proceso un paso más allá y muestra la misma red dividida en cuatro subredes.

Acceso interno selectivo por host, rango de direcciones o puerto

El tráfico a través de la interfaz interna de una máquina de cortafuegos se puede limitar selectivamente, al igual que el tráfico a través de la interfaz externa. Por ejemplo, en un firewall para un sitio residencial pequeño, en lugar de permitir que todo pase por la interfaz interna, el tráfico podría limitarse a DNS, SMTP, POP y HTTP. En este caso, digamos que una máquina de cortafuegos proporciona estos servicios para la LAN. Las máquinas locales no tienen permitido ningún otro acceso a servicios externos. En este caso, el reenvío no se realiza.

Punto de interés

En este ejemplo, los hosts locales están limitados a servicios específicos: DNS, SMTP, POP y HTTP. Debido a que POP es un servicio de recuperación de correo local en este caso, y debido a que DNS, SMTP y HTTP son servicios de proxy, los clientes LAN no tienen acceso directo a Internet. En cada caso, los clientes locales se conectan a servidores locales. POP es un servicio de LAN local. Los otros tres servidores establecen conexiones remotas en nombre del cliente.

Este ejemplo sería utilizado solo por un sitio pequeño, probablemente residencial. Colocar la puerta de enlace de correo y los servicios POP en el servidor de seguridad puede requerir que el servidor tenga cuentas de usuario. Sin embargo, no es necesario que estas cuentas sean cuentas de inicio de sesión.

Opciones de configuración para una LAN interna El

siguiente ejemplo considera una máquina con firewall con una interfaz interna conectada a una LAN. Las constantes para la interfaz interna se muestran aquí:

```
LAN_INTERFACE="eth1"           # Interfaz interna a la LAN
LAN_GATEWAY="192.168.1.1"      # Cortafuegos interno de la máquina
                                # dirección de interfaz
DIRECCIONES_LAN="192.168.1.0/24" # Rango de direcciones utilizadas en la LAN
```

Las máquinas LAN apuntan a la interfaz interna de la máquina cortafuegos como su servidor de nombres:

```
# Regla de respuesta de puerta de enlace genérica
$IPT -A SALIDA -o $LAN_INTERFACE -s $LAN_GATEWAY
    \ -d $LAN_ADDRESSES --dport
    $UNPRIVPORTS \ -m state --state ESTABLECIDO, RELACIONADO
    -j ACCEPTAR

# Reglas de solicitud de LAN específicas del servicio

$IPT -A INPUT -i $LAN_INTERFACE -p udp \ -s $LAN_ADDRESSES
    --sport $UNPRIVPORTS \ -d $LAN_GATEWAY --dport 53 \ -m state
    --state NUEVO, ESTABLECIDO, RELACIONADO -j ACCEPTAR

$IPT -A INPUT -i $LAN_INTERFACE -p tcp \ -s $LAN_ADDRESSES
    --sport $UNPRIVPORTS \ -d $LAN_GATEWAY --dport 53 \ -state --
    state NUEVO, ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

Las reglas de nftables coincidentes son:

```
$NFT agregar salida de filtro de regla oif $LAN_INTERFACE \
    ip saddr $LAN_GATEWAY ip daddr $LAN_ADDRESSES \ tcp dport
    $UNPRIVPORTS ct estado establecido, relacionado aceptar

$NFT agregar entrada de filtro de regla si $LAN_INTERFACE \ ip protocol udp
    ip saddr $LAN_ADDRESSES \ udp sport $UNPRIVPORTS ip daddr
    $LAN_GATEWAY \ udp dport 53 ct estado nuevo, establecido, relacionado
    aceptar

$NFT agregar entrada de filtro de regla si $LAN_INTERFACE \ ip protocol tcp
    ip saddr $LAN_ADDRESSES \ tcp sport $UNPRIVPORTS ip daddr
    $LAN_GATEWAY \ tcp dport 53 ct estado nuevo, establecido, relacionado
    aceptar
```

Las máquinas LAN también apuntan al cortafuegos como su servidor SMTP y POP:

```
# Envío de correo - SMTP

$IPT -A INPUT -i $LAN_INTERFACE -p tcp \ -s $LAN_ADDRESSES
    --sport $UNPRIVPORTS \ -d $GATEWAY --dport 25 \ -state --state
    NUEVO, ESTABLECIDO, RELACIONADO -j ACCEPTAR

# Recepción de correo - POP

$IPT -A INPUT -i $LAN_INTERFACE -p tcp \ -s $LAN_ADDRESSES
    --sport $UNPRIVPORTS \ -d $GATEWAY --dport 110 \ -state --state
    NUEVO, ESTABLECIDO, RELACIONADO -j ACCEPTAR
```

Finalmente, un servidor proxy de caché web local se está ejecutando en la máquina de firewall en el puerto 8080. Las máquinas internas apuntan al servidor web en el firewall como su proxy, y el servidor web reenvía cualquier solicitud saliente en su nombre, junto con el almacenamiento en caché de las páginas recuperadas. desde Internet. Todas las conexiones al proxy se realizan a través del puerto 8080. El servidor proxy inicia el acceso web seguro y FTP a sitios remotos:

```
$IPT -A ENTRADA -i $LAN_INTERFACE -p tcp \
-s $DIRECCIONES_LAN --sport $UNPRIVPORTS \
-d $PUERTA DE ENLACE --dport 8080 \
--estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACEPTAR
```

Aquí está la regla de nftables ; observe cómo todos los puertos de destino (para SMTP, POP y proxy) se manejan en una regla:

```
$NFT agregar entrada de filtro de regla si $LAN_INTERFACE \
tcp dport { 25,995,8080 } ip saddr $LAN_ADDRESSES \
tcp sport $UNPRIVPORTS ip daddr $GATEWAY \
ct estado nuevo,establecido,relacionado aceptar
```

Recuerde que el servidor web utilizará el protocolo de modo pasivo FTP para recuperar datos de sitios FTP remotos. La interfaz externa del cortafuegos necesitará reglas de entrada y salida para acceder a los puertos de servicio FTP, HTTP y HTTPS remotos. El host de la puerta de enlace también debe tener reglas para que la interfaz externa tenga en cuenta el correo electrónico y las consultas DNS locales a los hosts remotos.

Opciones de configuración para varias LAN

Agregar una segunda LAN interna permite que este ejemplo se desarrolle más. El siguiente ejemplo se puede asegurar mejor que el ejemplo anterior. Como se muestra en la Figura 7.5, los servicios DNS, SMTP, POP y HTTP se ofrecen desde máquinas servidor en una segunda LAN en lugar de desde la máquina cortafuegos. La segunda LAN puede servir o no como DMZ pública. Es igualmente posible que la segunda LAN represente una LAN de servicio interna y sus servicios no se ofrezcan a Internet (aunque, en ese caso, se podría requerir que el firewall sea al menos una puerta de enlace de correo, dependiendo de la configuración del firewall local). En cualquier caso, los hosts de cortafuegos normalmente no alojan servicios. En este ejemplo, el tráfico se enruta entre las dos LAN mediante las interfaces internas en la máquina del cortafuegos.

Las siguientes variables se utilizan para definir la LAN, las interfaces de red y el servidor máquinas en este ejemplo:

```
CLIENT_LAN_INTERFACE="eth1" # Interfaz interna a la LAN
SERVIDOR_LAN_INTERFACE="eth2" # Interfaz interna a la LAN
CLIENT_ADDRESSES="192.168.1.0/24" # Rango de direcciones utilizadas en la LAN del cliente
SERVER_ADDRESSES="192.168.3.0/24" # Rango de direcciones utilizadas en el servidor LAN
SERVIDOR_DNS="192.168.3.2" # Servidor DNS LAN
```

La primera regla cubre todas las respuestas del servidor a los clientes en la LAN del cliente:

```
$IPT -A ADELANTE -i $SERVER_LAN_INTERFACE -o $CLIENT_LAN_INTERFACE \
-s $DIRECCIONES_SERVIDOR -d $DIRECCIONES_CLIENTE \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

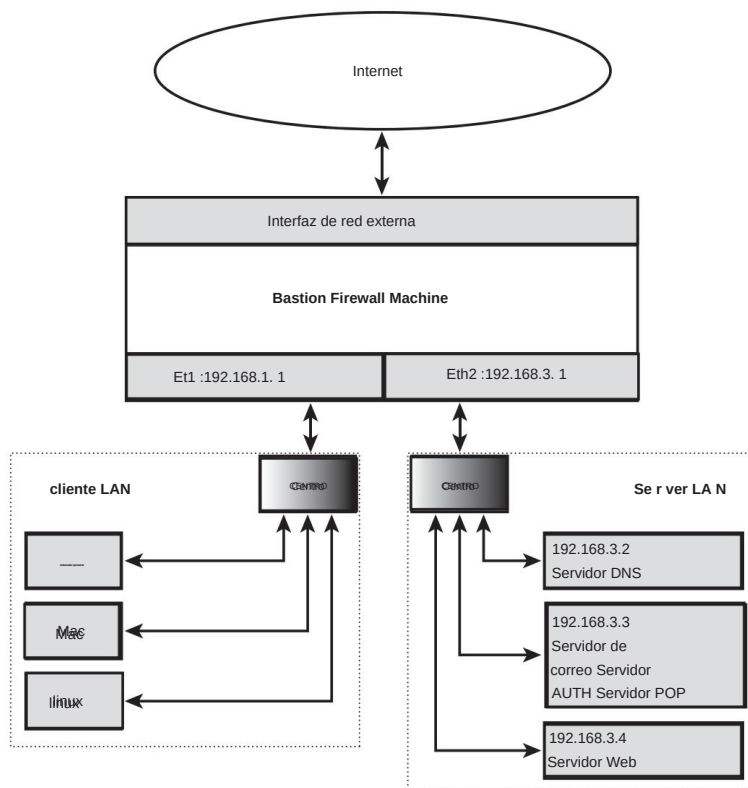


Figura 7.5 Separación de clientes y servidores en varias LAN

La segunda regla cubre todo el tráfico de conexión en curso desde los clientes LAN a los servidores locales en el servidor LAN:

```
$IPT -A ADELANTE -i $CLIENT_LAN_INTERFACE -o $SERVER_LAN_INTERFACE \
-s $DIRECCIONES_CLIENTE -d $DIRECCIONES_SERVIDOR \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

La tercera regla cubre todas las respuestas del servidor remoto a las solicitudes de los clientes desde el servidor local. servidores en la LAN del servidor:

```
$IPT -A ADELANTE -i $EXTERNAL_INTERFACE -o $SERVER_LAN_INTERFACE \
-d $DIRECCIONES_SERVIDOR \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

La cuarta regla cubre todas las respuestas del servidor local a las solicitudes del cliente desde el control remoto. anfitriones en Internet:

```
$IPT -A ADELANTE -i $SERVER_LAN_INTERFACE -o $EXTERNAL_INTERFACE \
-s $DIRECCIONES_SERVIDOR \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

194 Capítulo 7 Reenvío de paquetes

Estas cuatro reglas se expresan en la sintaxis de nftables de la siguiente manera:

```
$NFT agregar filtro de regla hacia adelante si $SERVER_LAN_INTERFACE o si $CLIENT_LAN_INTERFACE \
ip saddr $SERVER_ADDRESSES ip daddr $CLIENT_ADDRESSES \ ct estado
establecido, relacionado aceptar
```

```
$NFT agregar filtro de regla hacia adelante si $CLIENT_LAN_INTERFACE o si $SERVER_LAN_INTERFACE \
ip saddr $CLIENT_ADDRESSES ip daddr $SERVER_ADDRESSES \ ct estado
establecido, relacionado aceptar
```

```
$NFT agregar filtro de regla hacia adelante si $EXTERNAL_INTERFACE o si $SERVER_LAN_INTERFACE \
ip daddr $SERVER_ADDRESSES \ ct
estado establecido, relacionado aceptar
```

```
$NFT agregar filtro de regla hacia adelante si $SERVER_LAN_INTERFACE o si $EXTERNAL_INTERFACE \
ip saddr $SERVER_ADDRESSES \ ct
estado establecido, relacionado aceptar
```

Las máquinas locales usan el servidor DNS en SERVER_LAN como su servidor de nombres. Al igual que con las reglas entre la interfaz interna y la interfaz externa del cortafuegos, las reglas de acceso al servidor se definen para la interfaz de la LAN del cliente. Las reglas de acceso del cliente se definen para el interfaz LAN del servidor:

```
$IPT -A FORWARD -i $CLIENT_LAN_INTERFACE -o $SERVER_LAN_INTERFACE -p udp \ -s
$DIRECCIONES_CLIENTE --sport $UNPRIVPORTS \ -d $DNS_SERVER --dport 53 \ -m estado --
estado NUEVO -j ACCEPTAR
```

```
$IPT -A FORWARD -i $CLIENT_LAN_INTERFACE -o $SERVER_LAN_INTERFACE -p tcp \ -s
$DIRECCIONES_CLIENTE --sport $UNPRIVPORTS \ -d $DNS_SERVER --dport 53 \ -m estado --
estado NUEVO -j ACCEPTAR
```

Aquí están las reglas de nftables :

```
$NFT agregar filtro de regla adelante si $CLIENT_LAN_INTERFACE oif $SERVER_LAN_INTERFACE \ ip saddr
$CLIENT_ADDRESSES ip daddr $DNS_SERVER \ udp dport 53 udp sport $UNPRIVPORTS \ ct state new accept
```

```
$NFT agregar filtro de regla adelante si $CLIENT_LAN_INTERFACE oif $SERVER_LAN_INTERFACE \ ip saddr
$CLIENT_ADDRESSES ip daddr $DNS_SERVER \ tcp dport 53 tcp sport $UNPRIVPORTS \ ct state new accept
```

El servidor DNS en la segunda LAN necesita obtener su información de una fuente externa. Si el servidor local fuera un servidor de caché y reenvío a un servidor externo, reenviando búsquedas no resueltas al servidor externo, las reglas de reenvío del firewall para su interfaz LAN de servidor interno y su interfaz de Internet externa serían las siguientes:

```
$IPT -A FORWARD -i $SERVER_LAN_INTERFACE -o $EXTERNAL_INTERFACE -p udp \ -s $DNS_SERVER
--sport 53 \ -d $NAME_SERVER_1 --dport 53 \ -m state --state NEW -j ACCEPTAR
```

```
$IPT -A FORWARD -i $SERVER_LAN_INTERFACE -o $EXTERNAL_INTERFACE -p udp \-s $DNS_SERVER
--sport $UNPRIVPORTS \-d $NAME_SERVER_1 --dport 53 \-m state --state NEW -j ACCEPTAR
```

```
$IPT -A FORWARD -i $SERVER_LAN_INTERFACE -o $EXTERNAL_INTERFACE -p tcp \-s $DNS_SERVER
--sport $UNPRIVPORTS \-d $NAME_SERVER_1 --dport 53 \-m state --state NEW -j ACCEPTAR
```

Estas son las reglas de nftables , condensando las reglas UDP en una sola regla:

```
$NFT agregar filtro de regla hacia adelante si $SERVER_LAN_INTERFACE o si $EXTERNAL_INTERFACE \
udp sport { 53,$UNPRIVPORTS } udp dport 53 \ ip saddr
$DNS_SERVER ip daddr $NAME_SERVER \ ct state new accept
```

```
$NFT agregar filtro de regla adelante si $SERVER_LAN_INTERFACE oif $EXTERNAL_INTERFACE \ tcp sport
$UNPRIVPORTS tcp dport 53 \ ip daddr $NAME_SERVER ip saddr $DNS_SERVER \ ct state new accept
```

Resumen

Este capítulo cubrió algunas de las opciones de firewall disponibles cuando protege una LAN.

Las políticas de seguridad se definen en relación con el nivel de necesidades de seguridad del sitio, la importancia de proteger los datos y el costo de la pérdida de datos o la privacidad. Comenzando con el muro de fuego bastión desarrollado en el Capítulo 5 como base, las opciones de configuración de LAN y firewall se analizaron en configuraciones cada vez más complejas.

Esta página se dejó en blanco intencionalmente

NAT: dirección de red

Traducción

La traducción de direcciones de red es una tecnología para sustituir una fuente o un destino dirección en el encabezado IP con otra dirección. Tradicionalmente, es una tecnología de traducción de direcciones IP para mapear paquetes entre dos reinos de direccionamiento diferentes. El uso más común de NAT es mapear conexiones salientes entre una red local de dirección privada y la Internet de dirección pública. De hecho, eso fue lo que originalmente se propuso hacer, principalmente en conjunto con los espacios de direcciones de clases privadas recién definidos; ambos fueron intentos de aliviar la escasez de espacio de direcciones IPv4.

Este capítulo introduce el concepto de NAT y explica cuáles son los distintos tipos de NAT. se utilizan típicamente para.

El trasfondo conceptual de NAT

NAT se presentó por primera vez en 1994 en RFC 1631, "El traductor de direcciones de red IP (NAT)", que luego fue reemplazado por RFC 3022, "Traductor de direcciones de red IP tradicional (NAT tradicional)". NAT se propuso como una posible solución temporal a corto plazo (que se utilizará hasta que se implemente IPv6) para la creciente escasez de direcciones IP públicas. NAT también se vio como una posible solución a las crecientes demandas de enrutadores que manejaban bloques de direcciones no contiguos. Se pensó que NAT posiblemente podría reducir o eliminar la necesidad de CIDR, lo que, a su vez, estaba provocando reasignaciones de direcciones y cambios en el software del enrutador y las configuraciones de red. NAT también se vio como un medio para evitar el costo y la sobrecarga de la reenumeración de la red local cuando se reasignaron los espacios de direcciones o cuando un sitio cambió de proveedor de servicios y se le asignó un nuevo bloque de direcciones públicas.

NAT se vio no solo como una solución a corto plazo, sino también como una solución que posiblemente podría causar más problemas de los que resolvía. Con la excepción de FTP, se pensó que los protocolos de aplicación más problemáticos eran protocolos heredados que gradualmente caerían en desuso. Se asumió que, frente a NAT, los desarrolladores de aplicaciones de red naturalmente se volverían más conscientes de las consideraciones de extremo a extremo, tendrían cuidado de no incrustar información de direcciones en los datos de las nuevas aplicaciones y evitarían divergir del estándar cliente/ modelo de servidor

Justo lo contrario resultó ser el caso. IPv6 aún no se ha implementado, dando NAT estado permanente a largo plazo. El uso de NAT se volvió casi universal a medida que el acceso a Internet se hizo más accesible para el público en general y las direcciones IPv4 disponibles se volvieron cada vez más escasas. Los protocolos de aplicación estándar de la época y los protocolos estándar comunes que todavía se usan en la actualidad, incluidos DNS, HTTP, SMTP, POP y NNTP, funcionan bien con NAT, y todas las implementaciones de NAT brindan soporte especial para FTP.

El éxito de NAT en la traducción transparente es el resultado de las características estándar de conexión cliente/servidor de estos protocolos comunes. Sin embargo, las excepciones no resultaron ser unas pocas aplicaciones heredadas y extrañas. Las aplicaciones de Internet se han vuelto cada vez más interactivas. Las aplicaciones más nuevas a veces no tienen una relación cliente/servidor clara. A veces, un solo servidor coordina la comunicación entre varios usuarios, que también pueden iniciar la comunicación entre ellos, independientemente del servidor. Varios servidores pueden funcionar junto con servicios distribuidos que se ejecutan en varios dominios de direcciones NAT, o con servicios proporcionados por diferentes tipos de servidores que funcionan de forma cooperativa. Ciertas sesiones multimedia heredadas y otras multitransmisión y bidireccionales, multiconexión pueden iniciar conexiones en ambas direcciones, pueden tener muchas conexiones simultáneas por sesión y pueden depender tanto de TCP como de UDP simultáneamente. El cliente no siempre es una entidad estacionaria y permanentemente direccionable, como ocurre con la ubicación dinámica del cliente en términos de dispositivos móviles y empleados que trabajan a distancia. Algunos servicios se basan en la integridad de datos y paquetes de extremo a extremo, al igual que los protocolos de autenticación y encriptación IPsec.

Estas aplicaciones de red no funcionan con NAT de forma transparente. Se debe proporcionar soporte específico de puerta de enlace de capa de aplicación (ALG) para cada aplicación para que NAT sea capaz de traducir estos paquetes correctamente. En el caso del cifrado, los protocolos de seguridad de la capa de transporte de extremo a extremo que utilizan métodos de cifrado y autenticación no funcionan, punto.

Independientemente de las dificultades asociadas con NAT, su utilidad asegura que esté aquí permanecer durante la duración de IPv4. Mientras tanto, la gente de firewall está buscando formas alternativas de usar firewall para resolver los problemas que presentan los protocolos más nuevos, tanto en términos de NAT como en términos de filtrado de paquetes en sí.

Necesitamos métodos de firewall alternativos porque el firewall en sí tiene problemas cuando se implementa con la tecnología actual. NAT no es el único problema. La multimedia y el costo y la sobrecarga de las puertas de enlace de nivel de aplicación están forzando gradualmente el problema. Algunos de estos protocolos simplemente no se pueden filtrar con los enfoques de firewall (y NAT) actuales.

Existen tres categorías generales de NAT, como se describe en RFC 2663, "IP Network Terminology and Considerations of the Translator of Addresses (NAT)":

• La NAT tradicional, saliente y unidireccional se usa para redes con espacio de direcciones privado. Las sesiones salientes se pueden iniciar desde la LAN privada a hosts remotos de Internet. Las sesiones entrantes no se pueden iniciar desde hosts remotos a hosts locales en la LAN de dirección privada.

La NAT tradicional se divide en dos subtipos generales, aunque los dos subtipos pueden superponerse en la práctica:

- La NAT básica solo realiza la traducción de direcciones. Por lo general, se usa para asignar direcciones de fuentes privadas locales a una de un grupo de direcciones públicas. Durante la duración de todas las sesiones iniciadas por un host local en particular, existe una asignación uno a uno entre un par de direcciones públicas y privadas en particular.
 - Traducción de direcciones y puertos de red (NAPT) realiza la traducción de direcciones pero también reemplaza el puerto de origen del host LAN local con un puerto de origen en el dispositivo NAT. Por lo general, se usa para asignar direcciones de origen privadas locales a una única dirección pública (como en el enmascaramiento de Linux). Debido a que el dispositivo NAT tiene una única dirección IP para asignar todas las conexiones LAN privadas salientes, el par de puertos de origen público y privado se utiliza para asociar una conexión particular con una dirección de host privada particular y una conexión particular de ese host.
 - NAT bidireccional realiza una traducción de direcciones bidireccional, lo que permite conexiones entrantes y salientes. Hay un mapeo uno a uno entre una dirección pública y una dirección privada. Efectivamente, la dirección pública es un alias público de la dirección privada del host local. Esto permite que los hosts remotos se dirijan al host privado mediante la dirección pública asociada a él. El dispositivo NAT traduce la dirección de destino pública en el paquete entrante a la dirección privada que realmente se le asigna al host local.
- Un uso de esto es el mapeo de direcciones bidireccional entre espacios de direcciones IPv4 e IPv6. Aunque ambas direcciones se pueden enrutar dentro de sus propios espacios de direcciones, las direcciones IPv6 no se pueden enrutar dentro del espacio de direcciones IPv4. Un host en el espacio de direcciones IPv4 no puede hacer referencia directamente a un host en el espacio de direcciones IPv6. Asimismo, un host en el espacio IPv6 no puede hacer referencia directamente a un host en el espacio IPv4. Es la dirección del host IPv6 la que se traduce de un lado a otro entre los dos dominios de direccionamiento.
- Otro uso de NAT bidireccional que es más relevante para los usuarios de Linux es reenviar conexiones entre Internet y servidores locales con direcciones privadas cuando el sitio ofrece servicios públicos desde una LAN pero tiene una única dirección IP pública.
- Twice NAT realiza una traducción bidireccional de direcciones de origen y destino, pero tanto las direcciones de origen como las de destino se traducen en ambas direcciones. Dos veces NAT es se utiliza cuando los espacios de direcciones de las redes de origen y destino chocan. Esto podría deberse a que un sitio usó por error direcciones públicas asignadas a otra persona. Se puede usar Twice NAT como una conveniencia cuando un sitio se vuelve a numerar o se asigna a un nuevo bloque de direcciones públicas pero el propietario no desea administrar las nuevas asignaciones de direcciones de inmediato.

Las ventajas de NAT incluyen estas:

- Los paquetes que contienen datos de protocolo de aplicación estándar se traducen de forma transparente entre redes.
- Los servicios estándar de cliente/servidor "simplemente funcionan" con NAT.

200 Capítulo 8 NAT: traducción de direcciones de red

- NAT alivia los problemas causados por la creciente escasez de IP disponible direcciones compartiendo una dirección pública o un pequeño bloque de direcciones públicas entre una red local completa.
 - NAT reduce la necesidad de volver a numerar las direcciones IP públicas y locales.
 - NAT reduce la necesidad de implementar y administrar esquemas de enrutamiento más complicados dentro de redes locales más grandes.
 - En la forma más común de NAT junto con direcciones IP privadas, el tráfico entrante no deseado no pasa porque las máquinas locales no son direccionables.
 - En una de las otras formas de NAT, se usa para permitir servidores virtuales, en los que un servidor farm parece ser un único servidor direccionable para el equilibrio de carga.
- Las desventajas de NAT incluyen las siguientes:
- NAT introduce puntos únicos de falla dentro de la red al mantener estado dentro de la propia red.
 - Mantener el estado crítico en el dispositivo NAT rompe el paradigma de Internet en el sentido de que los paquetes ya no se pueden redirigir automáticamente alrededor de los enrutadores NAT fallidos.
 - NAT rompe el paradigma de Internet de transparencia de extremo a extremo al modificar paquetes feliz en el camino.
 - Como resultado de la modificación de la información de direccionamiento, la compatibilidad con NAT específica de la aplicación es necesario para cualquier aplicación que incruste direcciones o puertos locales en la carga útil de la aplicación.
 - Como resultado de la modificación de la información de direccionamiento para aplicaciones que incorporan direcciones locales o puertos en la carga útil de la aplicación, los paquetes entrantes destinados a un host NAT deben desfragmentarse antes de reenviarse.
 - NAT aumenta los requisitos de recursos y rendimiento para los dispositivos NAT, lo que de lo contrario, estaría dedicado al reenvío rápido de datagramas. NAT representa no solo la sobrecarga de desfragmentación, inspección de paquetes y modificación de paquetes, sino también la sobrecarga de mantenimiento de estado, tiempos de espera de estado y recolección de basura de estado.
 - Debido al mantenimiento del estado dentro de la red y el recurso asociado requisitos, los dispositivos NAT no son infinitamente escalables. Además, sin técnicas de uso compartido complicadas, los hosts no pueden usar varios dispositivos NAT de pares, un aspecto del punto único de falla.
 - Los protocolos multisequencia bidireccionales requieren compatibilidad con NAT específica de la aplicación para reenviar las secuencias secundarias entrantes al host local adecuado. (Tenga en cuenta que estos protocolos generalmente también requieren compatibilidad con ALG para el firewall).
 - NAT puede interrumpir la capacidad de ejecutar varias instancias de la misma aplicación de cliente de red local en conexión con el mismo servidor remoto. Este problema tiende a ocurrir con juegos en red e IRC, donde la sesión tiene entradas asociadas.
- arroyos

¿ NAT no se puede usar con IPsec en modo de transporte para la seguridad de un extremo a otro durante algunos razones:

- ¿ Las técnicas de seguridad de la capa de transporte de extremo a extremo no son posibles porque las técnicas se basan en la integridad de extremo a extremo del contenido del encabezado del paquete para la autenticación.
- ¿ Las técnicas de seguridad de la capa de transporte de extremo a extremo no son posibles porque las técnicas dependen de la integridad de extremo a extremo de la carga de datos del paquete, que también depende de la integridad del encabezado del paquete.
- ¿ Las técnicas de seguridad de la capa de transporte de extremo a extremo no son posibles porque los datos el cifrado hace que el contenido del paquete no esté disponible para su inspección. Las modificaciones de NAT no son posibles para cambiar la dirección incrustada y la información del puerto.
- ¿ Las relaciones de confianza de seguridad deben extenderse a la red desde los hosts de punto final, posiblemente hasta un punto completamente fuera del sitio local. IPsec y la mayoría de las tecnologías VPN deben extenderse al dispositivo NAT (en otras palabras, modo túnel IPsec). Nuevamente, el dispositivo NAT se convierte en un único punto de falla porque el dispositivo NAT debe terminar la VPN y establecer un nuevo enlace como proxy al destino.

Semántica NAT con iptables y nftables

Tanto iptables como nftables brindan una funcionalidad NAT completa, incluida la asignación de direcciones de origen (SNAT) y destino (DNAT). El término *NAT completo* no es un término formal; Me refiero a la capacidad de realizar NAT tanto de origen como de destino, para especificar una o un rango de direcciones de traducción, para realizar la traducción de puertos y para realizar la reasignación de puertos.

En versiones anteriores de Linux, se proporcionó una implementación parcial de NAT, conocida como "enmascaramiento" entre los usuarios de Linux. Se utilizó para asignar todas las direcciones privadas locales a la dirección IP pública única de la interfaz de red pública única del sitio.

A menudo se hablaba de NAT y de reenvío como dos componentes de lo mismo porque el enmascaramiento se especificaba como parte de la semántica de la regla FORWARD . Borrar los conceptos era funcionalmente irrelevante. Ahora es muy importante tener en cuenta la distinción. El reenvío y NAT son dos funciones y tecnologías distintas.

El reenvío es el enrutamiento del tráfico entre redes. El reenvío enruta el tráfico entre la red interfaces de trabajo *tal cual*. Las conexiones se pueden reenviar en cualquier dirección.

El enmascaramiento se encuentra encima del reenvío como un servicio de kernel separado. El tráfico es masqueraded en ambas direcciones, pero no simétricamente. El enmascaramiento es unidireccional. Solo se pueden iniciar conexiones salientes. A medida que el tráfico de las máquinas locales pasa a través del firewall a una ubicación remota, la dirección IP y el puerto de origen de la máquina interna se reemplazan con la dirección de la interfaz de red externa de la máquina del firewall y un puerto de origen libre en la interfaz. El proceso se invierte para las respuestas entrantes. Antes de que el paquete se reenvíe a la máquina interna, la dirección IP y el puerto de destino del cortafuegos

se reemplazan con la dirección IP real y el puerto de la máquina interna que participa en la conexión. El puerto de la máquina cortafuegos determina si el tráfico entrante, todo el cual se dirige a la máquina cortafuegos, está destinado a la máquina cortafuegos misma o a un host local en particular.

La semántica de reenvío y NAT está separada en iptables. La función de reenviar el paquete se realiza en la tabla de filtros utilizando la cadena FORWARD. La función de aplicar NAT al paquete se realiza en la tabla nat, utilizando una de las cadenas POSTROUTING, PREROUTING o OUTPUT de la tabla nat:

- El reenvío es una función de enrutamiento. La cadena FORWARD es parte de la tabla de filtros.
- NAT es una función de traducción que se especifica en la tabla nat. NAT toma colocar antes o después de la función de enrutamiento. Las cadenas POSTROUTING, PREROUTING y OUTPUT de la tabla nat son parte de la tabla nat. El NAT de origen se aplica en la cadena POSTROUTING después de que un paquete haya pasado por la función de enrutamiento. Source NAT también se aplica en la cadena OUTPUT para paquetes salientes generados localmente. (La cadena de SALIDA de la tabla de filtro y la cadena de SALIDA de la tabla nat son dos cadenas separadas y no relacionadas). El NAT de destino se aplica en la cadena PREROUTING antes de pasar el paquete a la función de enrutamiento.

¿Qué dirección de destino se ve dónde?

El NAT de destino se aplica en la cadena PREROUTING de la tabla nat, antes de que se tome la decisión de enrutamiento. Las reglas de la cadena PREROUTING deben coincidir con la dirección de destino original en el encabezado IP del paquete. Las reglas en la cadena INPUT o FORWARD de la tabla de filtros deben coincidir en la dirección NAT modificada en el mismo encabezado del paquete. Del mismo modo, si a ese mismo paquete también se le aplicara NAT de origen después de tomar la decisión de enrutamiento, y si la dirección de destino fuera importante para coincidir, la regla en la cadena POSTROUTING de la tabla nat coincidiría en la dirección de destino modificada.

El NAT de origen se aplica en la cadena POSTROUTING de la tabla nat, después de tomar la decisión de enrutamiento. Reglas sobre cualquier coincidencia de cadena en la dirección de origen original. La dirección de origen se modifica inmediatamente antes de enviar el paquete al próximo salto o al host de destino. La dirección de origen modificada no se ve en el host que aplica el NAT de origen.

Ninguna de estas distinciones entre reenvío y NAT fue un problema aparente con ipfwadm e ipchains. El par de reglas de reenvío no era necesario cuando se enmascaraba. El reenvío bidireccional y NAT estaban implícitos en una sola regla. La interfaz local entrante estaba implícita en la dirección de origen. La dirección de origen traducida se tomó de la especificación de la interfaz pública saliente. La traducción inversa de los paquetes de respuesta estaba implícita sin una regla explícita.

No se exceda con el uso de la sintaxis NAT

El resto de este capítulo presenta la sintaxis de la tabla nat. Se requiere otra palabra de precaución al mirar la sintaxis NAT completa. Las siguientes secciones describen la sintaxis más simple y general que se usa con NAT, que es la que se usará con más frecuencia. El sitio promedio no tendrá un uso para las funciones especializadas disponibles en la tabla nat.

Tanto las reglas SNAT como DNAT pueden especificar el protocolo, las direcciones de origen y destino, los puertos de origen y destino y los indicadores de estado, además de la dirección y los puertos traducidos. Cuando se hace esto, las reglas de la tabla nat se parecen mucho a las reglas de la tabla de filtros. Es muy fácil confundir las reglas NAT con las reglas del firewall, especialmente para las personas que están acostumbradas a la sintaxis de ipchains. El filtrado real se realiza en la cadena FORWARD.

Podría reflejar los campos de coincidencia entre las reglas FORWARD y las reglas NAT; los dos conjuntos de reglas podrían parecer casi idénticos. Para grandes conjuntos de reglas, se convertiría rápidamente en una pesadilla administrativa propensa a errores y lograría muy poco.

Recuerde que el reenvío de iptables y NAT son dos funciones completamente separadas. El filtrado del cortafuegos real se realiza mediante las reglas de la tabla de filtros. Para la mayoría de las personas, es mejor mantener las reglas de la mesa nat simples.

NAT de origen

Existen dos formas de NAT de origen en la tabla nat de iptables, especificadas como dos objetivos distintos, SNAT y MASQUERADE. SNAT es una traducción de dirección de origen estándar.

MASQUERADE es una forma especializada de NAT de origen para su uso en entornos en los que se asigna una dirección IP arbitraria y dinámicamente de forma temporal, conexión por conexión. En el momento de escribir este artículo, hay soporte enmascarado en nftables.

Ambos objetivos se utilizan en la cadena POSTROUTING de la tabla iptables nat. Dirección de la fuente las modificaciones se aplican después de que se haya tomado la decisión de enrutamiento para elegir la interfaz de salida adecuada. Por lo tanto, las reglas SNAT están asociadas con una interfaz saliente, no con una interfaz entrante.

Debido a que nftables no incluye ninguna tabla predeterminada, la tabla nat debe configurarse antes de que funcione cualquiera de los ejemplos de este capítulo. Para hacerlo, puede agregar una tabla nat al archivo de reglas setup-tables que se desarrolló en capítulos anteriores. El resultado final se ve así:

```
filtro de tabla {
    entrada de cadena {
        tipo filtro gancho entrada prioridad 0;
    }
    salida en cadena {
        tipo filtro gancho salida prioridad 0;
    }
    cadena adelante {
        escriba el enlace de filtro con prioridad de reenvío 0;
    }
}
```

```

tabla nat {
    enrutamiento previo de la cadena {
        escriba nat hook prerouting prioridad 0;
    }
    cadena posterior al enrutamiento {
        escriba nat hook postrouting prioridad 0;
    }
    salida en cadena {
        tipo nat gancho salida prioridad 0;
    }
}

```

Este archivo, guardado como tablas de configuración, se puede cargar con el comando

```
nft -f tablas de configuración
```

SNAT estándar

Esta es la sintaxis general para SNAT:

```

iptables -t nat -A POSTROUTING -o <interfaz de salida> ... \
    -j SNAT --to-source <dirección> [- <dirección>] [: puerto-puerto]

```

Para nftables la sintaxis general es

```

agregue la regla nat postrouting oif <interfaz saliente> \
    snat <dirección> [: puerto-puerto]

```

La dirección es la dirección de origen para sustituir la dirección de origen original en el paquete, presumiblemente la dirección de la interfaz de salida. NAT de origen es para lo que se usa tradicionalmente NAT, para permitir conexiones salientes. Especificar una sola dirección de traducción realiza NAPT, lo que permite que todos los hosts locales con direcciones privadas compartan la dirección IP pública única de su sitio.

Opcionalmente, se puede especificar un rango de direcciones de origen. Los sitios que tienen un bloque de direcciones públicas usarían este rango. A las conexiones salientes de los hosts locales se les asignaría una de las direcciones disponibles, y la dirección pública se asociaría con la dirección IP de un host local en particular.

MASCARADA Fuente NAT

La sintaxis general de MASQUERADE para iptables es la siguiente:

```

iptables -t nat -A POSTROUTING -o <interfaz de salida> ... \
    -j MASQUERADE [--to-ports <puerto>[-puerto]]

```

MASQUERADE no tiene una opción para especificar una dirección de origen en particular para usar en el dispositivo NAT. La dirección de origen utilizada es la dirección de la interfaz de salida.

La especificación de puerto opcional es un puerto de origen o un rango de puertos de origen para elegir en la interfaz de salida del dispositivo NAT.

Al igual que con SNAT, los puntos suspensivos representan cualquier otro selector de paquetes que se especifique. Por ejemplo, MASQUERADE podría aplicarse solo a un host local seleccionado.

NAT de destino

Existen dos formas de NAT de destino en la tabla nat de iptables , especificadas como dos objetivos distintos: DNAT y REDIRECT. DNAT es la traducción estándar de direcciones de destino. REDIRECT es una forma especializada de NAT de destino que redirige los paquetes a la interfaz de bucle invertido o de entrada del dispositivo NAT. No hay destino de redirección para nftables.

Los dos objetivos de iptables se pueden usar en cualquiera de las cadenas PREROUTING o OUTPUT de la tabla iptables nat . Las modificaciones de la dirección de destino se aplican antes de que se tome la decisión de enrutamiento para elegir la interfaz adecuada. Por lo tanto, en la cadena PREROUTING , las reglas DNAT y REDIRECT están asociadas con una interfaz de entrada para los paquetes que deben reenviarse a través del dispositivo o que se dirigen a la interfaz de entrada de este host. En la cadena OUTPUT , las reglas DNAT y REDIRECT se refieren a paquetes salientes generados localmente desde el propio host NAT.

ADNT estándar

La sintaxis general para DNAT es como se muestra aquí:

```
iptables -t nat -A PREROUTING -i <interfaz entrante> ... \
    -j DNAT --to-destination <dirección>[<dirección>][:puerto-puerto]
```

y

```
iptables -t nat -A SALIDA -o <interfaz de salida> ... \
    -j DNAT --to-destination <dirección>[<dirección>][:puerto-puerto]
```

La sintaxis de nftables es

```
nft add rule nat prerouting iif <interfaz de entrada> \ dnat <dirección de
destino>[:puerto-puerto]
```

y

```
nft agregar regla nat salida oif <interfaz saliente> \
    dnat <dirección de destino>[:puerto-puerto]
```

La dirección es la dirección de destino para sustituir la dirección de destino original en el paquete, presumiblemente la dirección de un servidor local.

Opcionalmente, se puede especificar un rango de direcciones de destino. Los sitios que tienen un grupo de servidores pares públicos utilizarían este rango. Las conexiones entrantes desde sitios remotos se asignarían a uno de los servidores. Estas direcciones podrían ser direcciones públicas asignadas a las máquinas internas. Por ejemplo, un grupo de servidores pares parece ser un único servidor para hosts remotos. Alternativamente, las direcciones podrían ser direcciones privadas y los servidores no serían directamente visibles o direccionables desde Internet. En este último caso, es probable que el sitio no tenga direcciones públicas para asignar a los servidores. Los hosts remotos intentan conectarse al servicio en el host NAT. El host NAT reenvía las conexiones de forma transparente al servidor interno de dirección privada.

La especificación del puerto final es otra opción. El puerto especifica el puerto de destino, o intervalo de puertos, en la interfaz de entrada del host de destino a la que se debe enviar el paquete.

Los puntos suspensivos representan cualquier otro selector de paquetes que se especifique. Por ejemplo, DNAT podría aplicarse para redirigir las conexiones entrantes desde un host remoto específico a un host interno. Otro uso podría ser redirigir las conexiones entrantes a un puerto de servicio particular al servidor real que se ejecuta en la red local.

REDIRECT Destino NAT

La sintaxis general para REDIRECT se muestra aquí:

```
iptables -t nat -A PREROUTING -i <interfaz entrante> ... \
-j REDIRECT [--to-ports <puerto[-puerto]>]
```

y

```
iptables -t nat -A SALIDA -o <interfaz de salida> ... \
-j REDIRECT [--to-ports <puerto[-puerto]>]
```

Recuerde, REDIRECT redirige el paquete a este host, que es el host que realiza la redirección.

Los paquetes que llegan a la interfaz entrante presumiblemente están dirigidos a algún otro host local. Otra alternativa podría ser que el paquete se dirija a un puerto de servicio local en particular, y el paquete se redirija a un puerto diferente en el host de forma transparente.

Los paquetes generados localmente, destinados a otro lugar, se redireccionan a la interfaz de bucle invertido en este host. Nuevamente, un paquete destinado a un servicio remoto específico podría redirigirse a la máquina local, tal vez a un proxy de almacenamiento en caché, por ejemplo.

Opcionalmente, se puede especificar un puerto de destino diferente o un rango de puertos. Si no se especifica ningún puerto, el paquete se entrega al puerto de destino que el remitente definió en el paquete.

Los puntos suspensivos representan cualquier otro selector de paquetes que se especifique. Por ejemplo, REDIRECT podría aplicarse para redirigir las conexiones entrantes a un servicio específico a un servidor, registrador, autenticador o algún tipo de software de inspección en el host local. El paquete podría enviarse desde este host después de realizar algún tipo de función de inspección. Otro uso podría ser redirigir las conexiones salientes a un servicio en particular a un servidor o un servicio intermedio en este host.

Ejemplos de SNAT y LAN privadas

Source NAT es, con mucho, la forma más común de NAT. El propósito original de NAT era usar NAT para brindar acceso saliente a Internet a hosts locales con direcciones privadas. Las siguientes secciones proporcionan algunos ejemplos simples del mundo real del uso de los objetivos MASQUERADE y SNAT de la tabla nat .

Enmascaramiento del tráfico LAN a Internet

La versión MASQUERADE de fuente NAT está destinada a personas con cuentas de acceso telefónico a las que se les asigna una dirección IP diferente en cada conexión. También lo utilizan personas que *siempre* tienen conexiones pero cuyo ISP les asigna una dirección IP diferente de forma regular. Esta versión se aplica solo a iptables.

El ejemplo más simple es una conexión PPP. Estos sitios a menudo usan una sola regla para enmascarar todas las conexiones salientes de la LAN:

```
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

El enmascaramiento, y NAT en general, se configura con el primer paquete. Con el enmascaramiento, una sola regla nacional puede ser suficiente. El NAT y el seguimiento del estado de la conexión se encargan de los paquetes entrantes. Sin embargo, el par de reglas FORWARD es necesario, como en este ejemplo:

```
iptables -A ADELANTE -o ppp0 \
    -m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACEPTAR
```

```
iptables -A FORWARD -o <interfaz LAN> \
    -m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

En este tipo simple de configuración, no es necesario especificar la interfaz de entrada.

Las reglas FORWARD se refieren al cruce de tráfico entre interfaces. Si el host tiene una sola interfaz de red y una sola interfaz ppp, todo lo que se reenvía desde una interfaz debe originarse necesariamente en la otra interfaz. Cualquier cosa aceptada por las reglas FORWARD en el filtro la tabla durante el enrutamiento será enmascarada por la regla POSTROUTING en la tabla nat.

Incluso con conexiones telefónicas a corto plazo, la única regla FORWARD que permite NUEVAS conexiones salientes debe dividirse en reglas para servicios específicos. Dependiendo de los dispositivos en red en la LAN y de cómo operen, lo más probable es que desee limitar el tráfico de LAN que se reenvía.

Este es un ejemplo de un único par de reglas FORWARD :

```
iptables -A FORWARD -i <interfaz LAN> -o ppp0 \
    -m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACEPTAR
```

```
iptables -A FORWARD -i ppp0 -o <interfaz LAN> \
    -m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

En este ejemplo, el único par de reglas FORWARD se divide en varios más específicos reglas que permiten solo consultas DNS y acceso web estándar. No se reenvía otro tráfico LAN, como se muestra en estos comandos:

```
iptables -A FORWARD -i ppp0 -o <interfaz LAN> \
    -m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

```
iptables -A ADELANTE -o ppp0 \
    -m estado --estado RELACIONADO, ESTABLECIDO -j ACEPTAR
```

```
iptables -A ADELANTE -o ppp0 -p udp \
    --sport 1024:65535 -d <servidor de nombres> --dport 53 \
    -m estado --estado NUEVO -j ACEPTAR
```

```
iptables -A ADELANTE -o ppp0 -p tcp \
    --sport 1024:65535 -d <servidor de nombres> --dport 53 \
    -m estado --estado NUEVO -j ACEPTAR
```

```
iptables -A ADELANTE -o ppp0 -p tcp \
    -s <host local> --sport 1024:65535 --dport 80 \
    -m estado --estado NUEVO -j ACEPTAR
```

208 Capítulo 8 NAT: traducción de direcciones de red

La única regla MASQUERADE en la cadena POSTROUTING de la tabla nat permanece sin cambios. Todo el tráfico reenviado está enmascarado. (El tráfico generado localmente que sale de la interfaz ppp0 no está enmascarado porque el tráfico se identifica con la dirección IP de la interfaz, por definición). Las reglas FORWARD en la tabla de filtros limitan qué tráfico se reenvía y, por lo tanto, qué tráfico se ve en Cadena POSTROUTING .

Aplicación de NAT estándar al tráfico de LAN a Internet

Suponiendo que ese mismo sitio tuviera una dirección IP asignada dinámicamente pero semipermanente o que tuviera una dirección IP asignada de forma permanente, se usaría la versión SNAT más general de NAT de origen. Al igual que en el ejemplo de enmascaramiento, los sitios residenciales pequeños a menudo reenvían y NAT todo el tráfico LAN saliente:

```
iptables -t nat -A POSTROUTING -o <interfaz externa> \
-j SNAT
--to-source <dirección externa>
```

Para nftables, una sola regla es todo lo que se necesita:

```
nft agregar regla nat postrouting ip saddr <direcciones de origen> \
oif <interfaz externa> snat <dirección externa>
```

Al igual que con el enmascaramiento, una sola regla SNAT puede ser suficiente. El NAT y el seguimiento del estado de la conexión se encargan de los paquetes entrantes. Sin embargo, para iptables, el par de reglas FORWARD es necesario, como en el siguiente ejemplo:

```
iptables -A FORWARD -o <interfaz externa> \
-m estado --estado NUEVO, ESTABLECIDO, RELACIONADO -j ACEPTAR
```

```
iptables -A FORWARD -o <interfaz LAN> \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

En el caso de sitios pequeños con conexiones 24x7, es especialmente importante ser selectivo sobre qué tráfico se reenvía. La única regla FORWARD que permite *nuevas* conexiones salientes no es suficiente. Los troyanos y los virus son comunes. Los dispositivos en red más nuevos tienden a ser algo promiscuos en lo que hacen a través de la red. Es muy probable que las máquinas y los dispositivos con Microsoft Windows, como las impresoras en red, generen mucho más tráfico del que cree. Además, gran parte de ese tráfico local es de difusión. Es una buena idea evitar el riesgo de reenviar el tráfico de difusión. Ya no se supone que los enrutadores reenvíen el tráfico de transmisión dirigido de forma predeterminada, pero muchos todavía lo hacen. (Las transmisiones limitadas no cruzan los límites de la red sin un agente de retransmisión para duplicar el paquete y transmitirlo. La mayoría de los dispositivos usan transmisiones limitadas). Una razón final es el caso de conectar computadoras portátiles de trabajo a la red doméstica. Muchos empleadores no quieren que las computadoras portátiles externas tengan acceso a Internet sin VPN o la protección de sus firewalls corporativos y software antivirus.

Ejemplos de DNAT, LAN y Proxies

Para el sitio residencial y de pequeñas empresas, NAT de destino fue probablemente la adición más bienvenida a NAT de Linux cuando se agregó a iptables.

Reenvío de host

DNAT proporciona la capacidad de reenvío de host que, hasta ahora, solo estaba disponible a través de soluciones de terceros. Para sitios pequeños con una única dirección IP pública, DNAT permite que las conexiones entrantes a los servicios locales se reenvíen de forma transparente a un servidor que se ejecuta en una DMZ. No es necesario que los servicios públicos se ejecuten en la máquina del cortafuegos.

Con una sola dirección IP, los sitios remotos envían solicitudes de clientes a la máquina del cortafuegos. El cortafuegos es el único host local visible en Internet. El propio servicio (por ejemplo, un servidor web o de correo) está alojado internamente en una red privada. Para los paquetes que llegan al puerto de ese servicio, el cortafuegos cambia la dirección de destino a la de la interfaz de red del servidor local y reenvía el paquete a la máquina privada. Se hace lo contrario para las respuestas del servidor. Para los paquetes del servidor, el cortafuegos cambia la dirección de origen por la de su propia interfaz externa y reenvía el paquete al cliente remoto.

El ejemplo más común es el reenvío de conexiones HTTP entrantes a una web local.

servidor:

```
iptables -t nat -A PREROUTING -i <interfaz pública> -p tcp \
--sport 1024:65535 -d <dirección pública> --dport 80 \
-j DNAT --to-destination <servidor web local>
```

La regla de nftables se ve así:

```
nft agregar regla nat preenrutamiento iif <interfaz pública> \
tcp sport 1024-65535 ip daddr <dirección pública> \
tcp dport 80 dnat <servidor web local>
```

La parte complicada es la cuestión de qué dirección se ve en cada cadena. NAT de destino se aplicó antes de que el paquete alcanzara la cadena FORWARD. Así que la regla en el ADELANTE. La cadena debe hacer referencia a la dirección IP privada del servidor interno en lugar de a la dirección pública del cortafuegos:

```
iptables -A FORWARD -i <interfaz pública> -o <interfaz DMZ> -p tcp \
--sport 1024:65535 -d <servidor web local> --dport 80 \
-m estado --estado NUEVO -j ACEPTAR
```

Para iptables, la regla se ve así:

```
nft agregar filtro de regla adelante iif <interfaz pública> \
oif <interfaz DMZ> tcp sport 1024-65535 \
tcp dport 80 ip daddr <servidor web local> \
ct estado nuevo aceptar
```

El seguimiento de conexiones y NAT invierten automáticamente la traducción de los paquetes que regresan del servidor. Debido a que se aceptó la solicitud de conexión inicial, una regla FORWARD genérica es suficiente para reenviar el tráfico de retorno desde el servidor local a Internet:

```
iptables -A FORWARD -i <interfaz DMZ> -o <interfaz pública> \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

La versión de nftables se ve así:

```
nft agregar filtro de regla adelante iif <interfaz DMZ> \
oif <interfaz pública> \
ct estado establecido, relacionado aceptar
```

Por supuesto, no olvide que el tráfico en curso del cliente también debe reenviarse porque la convención utilizada en este libro ha sido separar las reglas de servicio individuales que especifican el estado NUEVO de una sola regla para todo el tráfico ESTABLECIDO o RELACIONADO :

```
iptables -A FORWARD -i <interfaz pública> -o <interfaz DMZ> \
-m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
```

Para nftables:

```
nft agregar filtro de regla adelante iif <interfaz pública> \
oif <interfaz DMZ> \
ct estado establecido, relacionado aceptar
```

Resumen

Este capítulo cubrió la traducción de direcciones de red. Inicialmente, se describieron tres tipos básicos de NAT. También se discutieron el propósito original de NAT, para qué se usa hoy en día y sus ventajas y desventajas.

En iptables, se accede a las características de NAT a través de la tabla nat y las cadenas de esa tabla en lugar de a través de la tabla de filtros y la cadena FORWARD . Se discutieron las implicaciones del flujo de paquetes a través del sistema operativo y las diferencias entre las reglas de dirección que coinciden en la cadena FORWARD y en las cadenas nat .

iptables implementa NAT de origen y NAT de destino. La fuente NAT está dividida en dos subcategorías, SNAT y MASQUERADE. SNAT es una traducción regular de direcciones de origen. MASQUERADE es una implementación especializada de fuente NAT. Elimina cualquier estado de la tabla NAT tan pronto como se interrumpe la conexión.

La NAT de destino también se divide en dos subcategorías, DNAT y REDIRECT. DNAT es una traducción regular de direcciones de destino. REDIRECT es un caso especial de traducción de direcciones de destino. Es un alias para redirigir paquetes al host local, independientemente del destino original del paquete.

Finalmente, se presentó una serie de ejemplos prácticos de NAT del mundo real.

Depuración de las reglas del cortafuegos

Así que ahora el firewall está configurado, instalado y activado. ¡Pero nada funciona! estás bloqueado fuera. ¿Quién sabe lo que está pasando? ¿Ahora que? ¿Por dónde empiezas?

Las reglas de firewall son notoriamente difíciles de acertar. Si está desarrollando a mano, invariablemente aparecerán errores. Incluso si produce una secuencia de comandos de firewall con una herramienta de generación automática de firewall, su secuencia de comandos sin duda requerirá ajustes personalizados eventualmente.

Este capítulo presenta funciones de informes adicionales de iptables y nftables herramientas y otras herramientas del sistema. La información es invaluable al depurar las reglas de su firewall. Este capítulo explica lo que la información puede decirle acerca de su cortafuegos.

Sugerencias generales para el desarrollo de firewalls

Rastrear un problema en el firewall es detallado y laborioso. No hay atajos para depurar las reglas cuando algo sale mal. En general, los siguientes consejos pueden facilitar un poco el proceso:

- Ejecute siempre las reglas desde un script de prueba completo, como el que he mostrado cómo construir a lo largo de este libro. Asegúrese de que la secuencia de comandos vacíe todas las reglas existentes, elimine las cadenas definidas por el usuario existentes y restablezca primero las políticas predeterminadas. De lo contrario, no puede estar seguro de qué reglas están vigentes o en qué orden.
- No ejecute reglas nuevas desde la línea de comandos. Especialmente no ejecute las reglas de política predeterminadas desde la línea de comandos. Se le interrumpirá inmediatamente si ha iniciado sesión con X Windows o de forma remota desde otro sistema, incluido un sistema en la LAN.
- Ejecute el script de prueba desde la consola si puede. Trabajando en X Windows en la consola puede ser más conveniente, pero persiste el peligro de perder el acceso a X Windows localmente. Está preparado para la posibilidad de tener que cambiar a una consola virtual para recuperar el control. Si debe utilizar una máquina remota para probar el script del cortafuegos, utilice cron para detener automáticamente el cortafuegos en caso de que quede bloqueado. Sin embargo, asegúrese de eliminar el trabajo cron antes de activar el firewall.
- Recuerde que vaciar las cadenas de reglas no afecta la política predeterminada actual en efecto.
- Con una política de denegación predeterminada, habilite siempre la interfaz de bucle invertido inmediatamente.

- Cuando sea factible, trabaje en un servicio a la vez. Agregar reglas de una en una o como entrada y pares de reglas de salida si no está utilizando el módulo de estado. Prueba sobre la marcha. Esto hace que sea mucho más fácil aislar las áreas problemáticas en las reglas de inmediato. El uso liberal del comando `echo` dentro de la secuencia de comandos del firewall puede ayudar a reducir la ubicación de las reglas que son problemáticas dentro de la secuencia de comandos.
- La primera regla que coincida gana. El orden es importante. Utilice los comandos de lista sobre la marcha para tener una idea de cómo se ordenan las reglas. Traza un paquete imaginario a través de la lista.
- Si la secuencia de comandos parece bloquearse, es probable que una regla esté haciendo referencia a un nombre de host simbólico en lugar de una dirección IP antes de que se hayan habilitado las reglas de DNS. Cualquier regla que use un nombre de host en lugar de una dirección debe ir después de las reglas de DNS, a menos que el host tenga una entrada en el archivo `/etc/hosts`.
- La tabla de filtros está implícita de forma predeterminada para `iptables` pero no para `nftables`.
- La mayoría de los módulos de coincidencia requieren que haga referencia al módulo por su nombre con la opción `-m` antes de especificar la sintaxis de características del módulo, como en el estado `-m --estado NUEVO`.
- Cuando un servicio no funciona, registre todos los paquetes perdidos que van en ambas direcciones, como así como todos los paquetes aceptados pertinentes. ¿Las entradas de registro en `/var/log/messages` o `/var/log/kern.log` muestran algo que se descarta cuando prueba el servicio? Si lo hacen, puede ajustar las reglas de su firewall para permitir los paquetes. Si no, el problema debe estar en otra parte.
- Si tiene acceso a Internet desde la máquina cortafuegos pero no desde la LAN,
 - Vuelva a verificar que el reenvío de IP esté habilitado ejecutando `cat /proc/sys/net/ipv4/ip_forward`. Se debe informar el valor 1. El reenvío de IP se puede configurar manualmente de forma permanente en `/etc/sysctl.conf` o en el propio script del cortafuegos. El primer método de configuración surte efecto cuando se reinicia la red.
 - Si el reenvío de IP no estaba habilitado, puede habilitarlo inmediatamente escribiendo la siguiente línea como raíz o incluyéndola en la secuencia de comandos del firewall y volviendo a ejecutar la secuencia de comandos:


```
echo "1" > /proc/sys/net/ipv4/ip_forward
```
- Si un servicio funciona en la LAN pero no externamente, active el registro de paquetes aceptados en la interfaz interna. Use el servicio `muy` brevemente para ver qué puertos, direcciones, banderas, etc., están en uso en ambas direcciones. No querrá registrar los paquetes aceptados durante mucho tiempo, o tendrá cientos o miles de entradas de registro en `/var/log/messages`.
- Si un servicio no funciona en absoluto, inserte temporalmente reglas de entrada y salida en el principio de la secuencia de comandos del cortafuegos para aceptar todo en ambas direcciones y registrar todo el tráfico. ¿El servicio está disponible ahora? Si es así, verifique las entradas de registro en `/var/log/messages` para ver qué puertos están en uso.

Listado de las reglas del cortafuegos

Es una buena idea enumerar las reglas que ha definido, para verificar que estén instaladas y en el orden esperado. El comando `-L` enumera las reglas reales para una cadena determinada tal como existen en la tabla interna del núcleo. Las reglas se enumeran en el orden en que se comparan con un paquete.

El formato básico del comando `iptables list` es el siguiente (ejecutar como root):

```
iptables [-v -n] -L [cadena]
```

o

```
iptables [-t <tabla>] [-v -n] -L [cadena]
```

El primer formato se refiere a la tabla de filtros **predeterminada**. Si no se especifica una cadena específica, el comando enumera todas las reglas en las tres cadenas de tablas de filtro **integradas**, además de las cadenas definidas por el usuario.

El segundo formato es necesario para enumerar las reglas en la **tabla nat** o **mangle**.

Agregar la opción `-v` es útil para ver la interfaz a la que se aplica la regla. Agregar la opción `-n` es útil si las reglas del firewall se refieren a direcciones remotas o ilegales, para evitar el largo tiempo de resolución de nombres para esas direcciones. Recuerde que si se especifica una cadena, debe seguir el comando `-L`. También tenga en cuenta que `-L` es un comando y `-v` y `-n` son opciones.

No se pueden combinar como en `-Lvn`.

A diferencia de usar `iptables` para definir reglas reales, usar `iptables` para listar reglas existentes puede hacerse desde la línea de comando. La salida va a su terminal o se puede redirigir a un archivo.

La sintaxis para `nftables` se ve así:

```
lista nft [cadena|tabla] <nombre de la tabla> [cadena] <nombre> [-a -n -n]
```

El nombre de la tabla siempre se requiere con cualquier comando de lista, pero al solicitar una lista en cadena no necesita usar la palabra clave de la tabla. El lector astuto reconocerá que no hay nombres de tabla predeterminados para `nftables`. El comando `list tables` muestra todas las tablas:

tablas de lista nft

La opción `-a` agrega el número de identificador a la lista, lo que puede ser útil para insertar o eliminar reglas individuales. La primera opción `-n` impide la traducción de nombres de IP a DNS. Agregar un segundo `-n` también evita la búsqueda de nombre de puerto a servicio.

Ejemplo de listado de tabla de iptables

El formato básico del comando `iptables table list` para enumerar todas las reglas en todas las cadenas de tablas de filtro es este:

```
iptables -vn -L ENTRADA
```

```
iptables -vn -L SALIDA
```

```
iptables -vn -L ADELANTE
```

o

```
iptables-vn-L
```

Las siguientes tres secciones usan siete reglas de muestra en la cadena de ENTRADA para ilustrar el diferencias entre las diversas opciones de formato de lista disponibles con la tabla de filtros y para explicar qué significan los campos de salida. Usando las diferentes opciones de formato de lista, las mismas siete reglas de muestra se enumeran con diversos grados de detalle y capacidad de lectura. Las opciones de formato de listado y los campos son los mismos para las cadenas INPUT, OUTPUT y FORWARD .

Aquí hay una lista abreviada de siete reglas de una cadena de ENTRADA usando las opciones de listado predeterminadas:

```

1 ENTRADA (política DROP)
2 prot opt fuente total cualquier lugar cualquier lugar cualquier lugar ESTABLECIDO RECHAZO sin severencia
3 lugar DROP tcp flags FIN, RST, RES, IS, PS, ACK, MP, RST, SYN, EST, URG, 'tcp,puerto cualquier lugar
4 mi.host.dominio \ ACCEPTAR estado RELACIONADO, ESTABLECIDO 192.168.1.0/25
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021

```

Los números de línea en los listados a lo largo de este capítulo no son parte del resultado; son simplemente marcadores de referencia. Los números se pueden generar agregando `--line-numbers` opción al comando. Los “números de línea” generados son las posiciones de las reglas dentro de la cadena.

¿ opt significa *opciones de fragmentación*, que se habrían configurado con -f
o el ! opción -f . ; Un ! en el primer espacio indica el ! opción -f , lo que significa

para hacer coincidir paquetes no fragmentados o el primer fragmento de una serie. Una f en el segundo espacio indica la opción -f , lo que significa hacer coincidir el segundo fragmento y los subsiguientes.

ÿ source es la dirección de origen en el encabezado del paquete IP.

ÿ destino es la dirección de destino en el encabezado del paquete IP.

La línea 3 ilustra cómo el comando de lista simple -L , sin argumentos de calificación, carece algún detalle importante. La regla parece aceptar todos los paquetes entrantes (tcp, udp e icmp) desde cualquier lugar. El detalle que falta, en este caso, es la interfaz, lo. Esta es la regla que acepta todas las entradas en la interfaz de loopback.

La línea 4 es una regla para registrar cualquier paquete ICMP fragmentado (segundo y posterior). El nivel de registro predeterminado para syslog es advertir. La regla LOG tiene un prefijo --log asociado cadena definida para ello.

La línea 5 es una regla que elimina los paquetes TCP sin establecer ningún indicador de estado.

La línea 6 es una regla que acepta cualquier paquete entrante que sea parte de una conexión ESTABLECIDA , o un paquete RELACIONADO con dicha conexión (es decir, un error ICMP asociado o una conexión de datos FTP).

La línea 7 es una regla que acepta solicitudes DNS UDP entrantes de hosts en la red local, 192.168.1.0/25. Tenga en cuenta que la red está dividida en dos subredes, por lo que los hosts pueden oscilar entre 192.168.1.1 y 192.168.1.126.

La línea 8 es una regla que rechaza las solicitudes o consultas de autenticación de TCP entrantes al local . servidor identificado . El mensaje de error ICMP tipo 3 devuelto contiene el código de puerto inalcanzable predeterminado. No es evidente en la lista que la máquina tenga dos interfaces de red. Se rechazan las solicitudes de la red "externa", dominio2.

La línea 9 acepta solicitudes de conexión TCP entrantes desde la LAN local para conexiones web HTTP estándar y conexiones web HTTPS. Se definió una lista de puertos de destino con la opción de coincidencia multipuerto .

iptables -n -L ENTRADA

La opción -n informa todos los campos como valores numéricos en lugar de nombres simbólicos. Esta opción puede ahorrar tiempo si sus reglas usan muchas direcciones IP específicas que, de otro modo, requerirían búsquedas de DNS antes de aparecer en la lista. Además, un intervalo de puertos es más informativo si aparece como 23:79 en lugar de como telnet:finger.

Usando las mismas siete reglas de muestra de la cadena INPUT , lo siguiente muestra lo que la salida de la lista parece usar la opción numérica -n :

> iptables -n -L ENTRADA

```

1  ENTRADA (política DROP)
2  prot opt fuente todos -- 0.0.0.0/0 0.0.0.0/0 destino objetivo
3  ACCEPT
4  fragmentado -f 0.0.0.0/0 0.0.0.0/0 LOG LOG flags 0 nivel 4 prefijo 'ICMP' \
5  tcp -- 0.0.0.0/0 0.0.0.0/0 DROP banderas tcp: 0x023F/0x020 \
```

216 Capítulo 9 Depuración de las reglas del cortafuegos

```

6   ACEPTAR      todo -- 0.0.0.0/0                                0.0.0.0/0          \
    estado RELACIONADO, ESTABLECIDO
7   ACEPTAR udp spts:1024:65535 dpts:1024:65535 192.168.1.2 \
    estado NUEVO
8   RECHAZAR tcp -- 0.0.0.0/0 192.168.1.254 \
    tcp dpt: 113 rechazo con puerto icmp inalcanzable
9   ACEPTAR tcp -- 192.168.1.0/25 dports multipuerto 192.168.1.2 \
    80,443 tcp spts:1024:65535 flags:0x0216/0x022 estado NUEVO

```

iptables -v -L ENTRADA

La opción -v produce una salida más detallada, incluido el nombre de la interfaz. Informar el nombre de la interfaz es especialmente útil cuando la máquina tiene más de una interfaz de red.

Usando las mismas siete reglas de muestra de la cadena INPUT , lo siguiente muestra lo que la salida de la lista parece usar la opción -v detallada:

```
> iptables -v -L ENTRADA
```

```

1   ENTRADA (política DROP 0 paquetes, 0 bytes)
2   bytes target out destino          prot opt in pkts          fuente          \
3   32 3416 ACEPTAR                  todo - hola      ninguna         en cualquier sitio \
    en cualquier sitio
4   0 0 REGISTRO                       icmp -f cualquier lugar prefijo de advertencia de nivel \
    en cualquier lugar de REGISTRO 'ICMP fragmentado: ' tcp -- cualquier lugar cualquier
5   0 0 GOTA                           indicador de tcp: FIN, SYN, RST, PSH, ACK, URG/NONE \
    en cualquier sitio
6   94 6586 ACEPTAR                  all -- cualquier lugar cualquier lugar \
    ESTABLECIDO 1 65 ACEPTAR spts:1024:65535 dpts:1024:65535 estado NUEVO \
    spts:1024:65535 dpts:1024:65535 estado NUEVO
7   192.168.1.0/25 \
8   0 0 REJECT tcp -- eth1 cualquiera en cualquier lugar my.host.domain2 tcp dpt:auth \
    rechazo-con icmp-port-unreach 192.168.1.0/25 my host de cualquier puerto dports http,https tcp
9   spts:1024:65535 \
    banderas: SYN, RST, estado ACK/SYN NUEVO

```

Ejemplo de listado de tabla nftables

El formato básico que normalmente uso cuando enumero las reglas de nftables para la tabla de filtros es

```
filtro de tabla de lista nft -a
```

Esto supone que se ha definido la tabla de filtros . Puedes usar las tablas de lista comando para buscar todas las tablas si la tabla de filtro no es el nombre de la tabla para la que necesita la lista. Aquí hay una lista abreviada de reglas para la cadena de ENTRADA del filtro tabla, como ejemplo:

```

1 filtro de ip de tabla {
2 entrada de cadena {
3     tipo filtro gancho entrada prioridad 0;
4     ifname "lo" aceptar # manejar 5
5     ct estado establecido, relacionado aceptar # manejar 8

```

```

6          Prefijo de registro no válido de estado de ct "Entrada NO VÁLIDA:" tasa de límite 3/segundo ydrop # handle 11

          iif eth0 ip saddr 255.255.255.255 soltar # manejar 18
          registro # manejar 19
          gota # manejar 20

7          }
8 9 10 11 }
```

La línea 1 muestra el nombre de la tabla.

La línea 2 muestra el nombre de la cadena.

La línea 3 muestra el tipo de tabla, el gancho y su prioridad.

La línea 4 muestra una regla que utiliza la interfaz localhost (lo) con un veredicto o disposición de aceptación. Las partes que siguen al # indican el número de identificador (5 en este caso) para acceder a esta regla directamente.

La línea 5 muestra una regla de estado de conexión que empareja paquetes establecidos o relacionados con un disposición de aceptar y su mango, 8.

La línea 6 muestra una regla para el estado de conexión no válido que se registrará con un prefijo de "entrada NO VÁLIDA:". El registro se limitará a una velocidad de 3 por segundo y, en última instancia, el paquete se eliminará. El identificador de esta regla es 11.

La línea 7 expresa una regla para los paquetes que llegan a eth0 con una dirección de origen de 255.255.255.255 que se descartarán. El mango es 18.

La línea 8 muestra una regla para registrar todos los paquetes que aún no han sido filtrados o procesados por un regla precedente. El identificador de esta regla es 19.

La línea 9 muestra la disposición final de los paquetes que llegan hasta aquí sin encontrarlos. una regla de concordancia. Su destino es ser abandonado. El mango de la regla es 20.

Las líneas 10 y 11 muestran las llaves de cierre de la definición de la regla.

Como puede ver en la salida de la regla, puede copiar esta salida o redirigir la salida a un archivo e inmediatamente volver a crear las reglas existentes.

Interpretación de los registros del sistema

syslogd y su hermano rsyslogd son los demonios de servicio que registran los eventos del sistema. En un sistema típico, el archivo de registro del sistema principal es /var/log/messages. Muchos programas utilizan los servicios de registro estándar de syslog. Otros programas, como el servidor web Apache, mantienen sus propios archivos de registro separados.

Configuración de registro del sistema

No todos los mensajes de registro son igualmente importantes, o incluso interesantes. Aquí es donde entra /etc/syslog.conf. El archivo de configuración /etc/syslog.conf le permite personalizar la salida del registro para satisfacer sus propias necesidades.

Los mensajes se clasifican por el subsistema que los produce. En las páginas man, estas categorías se denominan *instalaciones* (ver Tabla 9.1).

Tabla 9.1 Categorías de recursos de registro de **syslog**

Instalaciones	Categoría de mensaje
autenticación o seguridad	Seguridad/autorización
autorizaciónprivada	Seguridad privada/autorización
cron	mensajes del demonio cron
demonio	Mensajes generados por el demonio del sistema
ftp	Mensajes del servidor FTP
centro	Mensajes del núcleo
lpr	Subsistema de impresora
correo	subsistema de correo
noticias	Subsistema de noticias de la red
registro del sistema	mensajes generados por syslogd
usuario	Mensajes generados por el programa de usuario
uucp	subsistema UUCP

Dentro de cualquier categoría de instalación determinada, los mensajes de registro se dividen en tipos de *prioridad* . los Las prioridades, en orden creciente de importancia, se enumeran en la Tabla 9.2.

Una entrada en syslog.conf especifica una instalación de registro, su prioridad y dónde escribir los mensajes. No es obvio que la prioridad sea inclusiva. Se entiende que significa todos los mensajes con esa prioridad y superior. Si especifica mensajes con prioridad de error , por ejemplo, se incluyen todos los mensajes con prioridad de error y superior: crítico, alerta y emergencia.

Los registros se pueden escribir en dispositivos, como la consola, así como en archivos y máquinas remotas.

Consejos sobre los archivos de registro en **/var/log**
 syslogd no crea archivos. Solo escribe en archivos existentes. Si no existe un archivo de registro, puede crearlo con el comando táctil y luego asegurarse de que sea propiedad de root. Por razones de seguridad, los usuarios generales a menudo no pueden leer los archivos de registro. El archivo de registro de seguridad, /var/log/secure, en particular, solo puede ser leído por root.

Tabla 9.2 Prioridades de mensajes de registro de **syslog**

Prioridad	Tipo de mensaje
depurar	Mensajes de depuración
información	Mensajes informativos de estado
aviso	Condiciones normales pero importantes
advertir o advertir	Mensajes de advertencia
error o error	Error de mensajes
crítico	Condiciones críticas
alerta	Se requiere atención inmediata
emergencia o pánico	El sistema es inutilizable

Estas dos entradas escriben todos los mensajes del kernel tanto en la consola como en `/var/log/messages`. Los mensajes se pueden duplicar a múltiples destinos:

núcleo.*	/dev/console
núcleo.*	/var/log/mensajes

Esta entrada escribe mensajes de pánico en todas las ubicaciones predeterminadas, incluidas `/var/log/messages`, la consola y todas las sesiones de terminal de usuario:

*.emerg	*

Las siguientes dos entradas escriben información de autenticación relacionada con el privilegio de root y conexiones a `/var/log/secure`, e información de autorización de usuario a `/var/log/auth`. Con la prioridad definida en el nivel de información, los mensajes de depuración no se escribirán:

authpriv.info	/var/log/seguro
auth.info	/var/log/auth

Las siguientes dos entradas escriben información general del demonio en `/var/log/daemon` y envían información de tráfico a `/var/log/maillog`:

daemon.notice	/var/log/daemon
mail.info	/var/log/maillog

Mensajes de daemon en la depuración y prioridades de información y mensajes de correo en la depuración prioridad no se registran (preferencia del autor). Las comprobaciones de correo nombradas, crond y sistemáticas producen mensajes informativos poco interesantes de forma regular.

La entrada final registra todas las categorías de mensajes de información prioritaria o superior en `/var/log/messages`, con la excepción de auth, authpriv, daemon y mail. En este caso, las últimas cuatro funciones de mensajes se establecen en none porque sus mensajes se dirigen a sus propios archivos de registro dedicados:

*.info;auth,authpriv,daemon,mail.none	/var/log/mensajes

Más información sobre la configuración de **syslog**

Para obtener una descripción más completa de las opciones de configuración de syslog y configuraciones de muestra, consulte las páginas man de `syslog.conf(5)` y `syslogd(8)`.

syslogd se puede configurar para escribir los registros del sistema en una máquina remota. Un sitio que utiliza una configuración de servidor en red similar al ejemplo del Capítulo 7, "Reenvío de paquetes", con servicios ofrecidos desde máquinas internas en la DMZ, podría querer mantener una copia remota de los registros del sistema. Mantener una copia remota ofrece dos ventajas: en primer lugar, los archivos de registro se consolidan en una sola máquina, lo que facilita que un administrador del sistema controle los registros. En segundo lugar, la información está protegida si una de las máquinas del servidor se ve comprometida alguna vez.

El Capítulo 11, "Detección y respuesta a intrusiones", analiza la importancia de los registros del sistema durante la recuperación si un sistema se ve comprometido alguna vez. Una de las primeras cosas que hace un atacante después de obtener acceso de root a una máquina comprometida es borrar

220 Capítulo 9 Depuración de las reglas del cortafuegos

el sistema registra o instala programas troyanos que no registran sus actividades. Los archivos de registro del sistema se han ido o no son confiables exactamente en el momento en que más los necesita. Mantener una copia remota de los registros ayuda a proteger esta información, al menos hasta que el pirata informático reemplace los demonios que escriben la información del archivo de registro.

Para registrar la información del sistema de forma remota, tanto la configuración de registro local como la configuración de registro remoto requieren ligeras modificaciones.

En la máquina remota que recopila los registros del sistema, agregue la opción `-r` al `syslogd` invocación. La opción `-r` le dice a `syslogd` que escuche en el puerto UDP 514 la información de registro entrante de los sistemas remotos.

En la máquina local que produce los registros del sistema, edite el archivo de configuración de `syslogd`, `/etc/syslog.conf`, y agregue líneas que especifiquen qué funciones de registro y prioridades desea escribir en un host remoto. Por ejemplo, lo siguiente copia toda la información de registro en nombre de host:

```
*. * @nombre de host
```

La salida de `syslogd` se envía a través de UDP. Tanto el puerto de origen como el de destino son 514. La regla de firewall del cliente sería la siguiente:

```
iptables -A SALIDA -o <interfaz de salida> -p udp \
-s <este anfitrión> --sport 514 \
-d <host de registro> --dport 514 -j ACEPTAR
```

Mensajes de registro del cortafuegos: ¿qué significan?

Para generar registros de firewall, el kernel debe compilarse con el registro de firewall habilitado. De forma predeterminada, los paquetes coincidentes individualmente se registran como mensajes `kern.warn` (prioridad 4). La mayoría de los campos de encabezado del paquete IP se informan cuando un paquete coincide con una regla con el REGISTRO objetivo. Los mensajes de registro del cortafuegos se escriben en `/var/log/messages` de forma predeterminada. El siguiente análisis se aplica tanto a `nftables` como a `iptables`.

Puede duplicar los mensajes de registro del firewall en un archivo diferente creando un nuevo archivo de registro y agregando una línea a `/etc/syslog.conf`:

```
core.warn /var/log/fwlog
```

Como ejemplo de TCP, esta regla niega el acceso al `portmap/sunrpc` puerto TCP 111 produciría el siguiente mensaje en `/var/log/messages`:

```
iptables -A ENTRADA -i $EXTERNAL_INTERFACE -p tcp \
--dport 111 -j REGISTRO --log-prefix "DROP portmap: "
```

```
iptables -A ENTRADA -i $EXTERNAL_INTERFACE -p tcp \
--dport 111 -j GOTA
```

nft agregar entrada de filtro de regla iif \$EXTERNAL_INTERFACE tcp dport 111 prefijo de registro "DROP yportmap:" soltar

(1)	(2)	(3)	(4) (7)	(5)	(6)
19 de junio 15:24:16 kernel de cortafuegos: DROP portmap: IN=eth0 OUT=					


```

                (8)
MAC=00:a0:cc:40:9b:a8:00:a0:cc:d4:a7:81:08:00

                (9)                (10)                (11)
SRC = 192.168.1.4 DST = 192.168.1.2 LEN = 60

                (12)                (13)                (14)                (15) (16)
TOS=0x00 PREC=0x00 TTL=64 ID=57743 DF

                (17)                (18)                (19)                (20)
PROTO = TCP SPT = 33926 DPT = 111 VENTANA = 5840

                (21) (22) (23)
RES=0x00 SYN URGP=0

```

Los campos de mensajes de registro están numerados para fines de discusión:

- El campo 1 es la fecha, 19 de junio.
- El campo 2 es la hora en que se escribió el registro, 15:24:16.
- El campo 3 es el nombre de host de la computadora, firewall.
- El campo 4 es la función de registro que genera el mensaje, kernel.
- El campo 5 es la cadena de prefijo de registro definida en la regla LOG .
- El campo 6 es la interfaz de red entrante a la que se adjunta la regla de entrada, eth0.
- El campo 7 es la interfaz de salida, que no tiene valor en una regla en la cadena de ENTRADA .
- El campo 8 es la dirección MAC de la interfaz a la que llega el paquete, seguido por ocho pares de dígitos hexadecimales basura.
- El campo 9 es la dirección de origen del paquete, 192.168.1.4.
- El campo 10 es la dirección de destino del paquete, 192.168.1.2.
- El campo 11 es la longitud total del paquete IP en bytes, LEN=60, incluidos el encabezado del paquete y sus datos.
- El campo 12 son los 3 bits de servicio del campo de tipo de servicio (TOS) , más un bit final reservado, TOS=0x00.
- El campo 13 son los 3 bits de precedencia superiores del campo TOS , PREC=0x00.
- El campo 14 es el campo de tiempo de vida (TTL) del paquete , TTL=64. El tiempo de vivir es el máximo número mínimo de saltos (es decir, enrutadores visitados) restantes antes de que caduque el paquete.
- El campo 15 es el ID del datagrama del paquete, ID=57743. El ID del datagrama es el ID de paquete o el segmento al que pertenece este fragmento TCP.
- El campo 16 es el campo de indicadores de fragmentos, lo que indica que el bit Don't Fragment (DF) está establecido.
- El campo 17 es el tipo de protocolo de mensaje contenido en el paquete, PROTO=TCP. Los valores de campo incluyen 6 (TCP), 17 (UDP), 1 (ICMP/<código>) y PROTO=<número> para otros tipos de protocolo.
- El campo 18 es el puerto de origen del paquete, 33926.

- El campo 19 es el puerto de destino del paquete, 111.
- El campo 20 es el tamaño de la ventana del remitente, VENTANA=5840, que indica la cantidad de datos que está dispuesto a aceptar y almacenar en búfer de este host en este momento.
- El campo 21 informa el campo reservado en el encabezado TCP. Los 4 bits deben ser 0.
- El campo 22 es el campo de estado de TCP. En este caso, se establece el indicador SYN.
- El campo 23 es el puntero urgente, que indica la cantidad de datos que se considera urgente. El campo es 0 porque el indicador URG no está establecido.

A la hora de interpretar el mensaje de log, los campos más interesantes son estos:

```
19 de junio 15:24:16 DROP portmap: IN=eth0 SRC=192.168.1.4 DST=192.168.1.2 PROTO=TCP SPT=33926
DPT=111 SYN
```

Esto dice que el paquete descartado es un paquete TCP que ingresa a la interfaz eth0 desde un puerto sin privilegios en 192.168.1.4. Era una solicitud de conexión TCP dirigida al puerto 111 de esta máquina (192.168.1.2), el puerto sunrpc/portmap. (Este puede ser un mensaje común porque Portmap históricamente es uno de los servicios más comúnmente atacados).

Como ejemplo de UDP, esta regla niega el acceso al puerto 111 de portmap/sunrpc UDP produciría el siguiente mensaje en /var/log/messages:

```
iptables -A ENTRADA -i $EXTERNAL_INTERFACE -p udp \
--dport 111 -j REGISTRO --log-prefix "DROP portmap:"
```

```
iptables -A ENTRADA -i $EXTERNAL_INTERFACE -p udp \
--dport 111 -j GOTA
```

nft agregar entrada de filtro de regla iif \$EXTERNAL_INTERFACE udp dport 111 prefijo de registro "DROP yportmap:" soltar

```
(1)      (2)      (3)      (4) (7)      (5)      (6)
19 de junio 15:24:16 kernel de cortafuegos: DROP portmap: IN=eth0 OUT=
```

```
(8)
MAC=00:a0:cc:40:9b:a8:00:a0:cc:d4:a7:81:08:00

(9)      (10)      (11)
SRC = 192.168.1.4 DST = 192.168.1.2 LEN = 28

(12)      (13)      (14)      (15)
TOS=0x00 PREC=0x00 TTL=40 ID=50655

(decadido) (17) (18) (19)
PROTO = UDP SPT = 33926 DPT = 111 LEN = 8
```

Los campos de mensajes de registro están numerados para fines de discusión:

- El campo 1 es la fecha, 19 de junio.
- El campo 2 es la hora en que se escribió el registro, 15:24:16.
- El campo 3 es el nombre de host de la computadora, firewall.

- El campo 4 es la función de registro que genera el mensaje, kernel.
- El campo 5 es la cadena de prefijo de registro definida en la regla LOG .
- El campo 6 es la interfaz de red entrante a la que se adjunta la regla de entrada, eth0.
- El campo 7 es la interfaz de salida, que no tiene valor en una regla en la cadena de ENTRADA .
- El campo 8 es la dirección MAC de la interfaz a la que llega el paquete, seguido por ocho pares de dígitos hexadecimales basura.
- El campo 9 es la dirección de origen del paquete, 192.168.1.4.
- El campo 10 es la dirección de destino del paquete, 192.168.1.2.
- El campo 11 es la longitud total del paquete IP en bytes, LEN=28, incluidos el encabezado del paquete y sus datos.
- El campo 12 son los 3 bits de servicio del campo TOS , más un bit final reservado, TOS=0x00.
- El campo 13 son los 3 bits de precedencia superiores del campo TOS , PREC=0x00.
- El campo 14 es el campo TTL del paquete , TTL=40. El tiempo de vida es el número máximo de saltos (es decir, enrutadores visitados) restantes antes de que caduque el paquete.
- El campo 15 es el ID del datagrama del paquete, ID=50655.
- El campo 16 es el tipo de protocolo de mensaje contenido en el paquete, PROTO=UDP. Los valores de campo incluyen 6 (TCP), 17 (UDP), 1 (ICMP/<código>) y PROTO=<número> para otros tipos de protocolo.
- El campo 17 es el puerto de origen del paquete, 33926.
- El campo 18 es el puerto de destino del paquete, 111.
- El campo 19 es la longitud del paquete UDP, incluidos el encabezado y los datos, LEN=8.

A la hora de interpretar el mensaje de log, los campos más interesantes son estos:

```
19 de junio 15:24:16 DROP portmap: IN=eth0 SRC=192.168.1.4 DST=192.168.1.2 PROTO=UDP
SPT=33926 DPT=111
```

Esto dice que el paquete descartado es un paquete UDP que ingresa a la interfaz eth0 desde un puerto sin privilegios en 192.168.1.4. Era un intercambio UDP dirigido al puerto 111 de esta máquina (192.168.1.2) , el puerto sunrpc/portmap . (Este puede ser un mensaje común porque Portmap históricamente es uno de los servicios más comúnmente atacados).

Comprobación de puertos abiertos

Enumerar las reglas de su cortafuegos con iptables -L es la principal herramienta disponible para verificar los puertos abiertos. Los puertos abiertos están definidos para estar abiertos por sus reglas ACCEPT . Más allá del comando iptables -L , otras herramientas como netstat son útiles para averiguar qué puertos están escuchando en el firewall.

netstat tiene varios usos. En la siguiente sección, lo usaremos para verificar los puertos activos para que podamos verificar que los puertos TCP y UDP en uso son los puertos que las reglas del firewall están tomando en cuenta.

El hecho de que netstat informe que el puerto está escuchando o abierto no significa que tenga acceso. posible a través de las reglas del cortafuegos. Después de esto, se presenta una herramienta de escaneo de puertos de terceros, Nmap. Estas herramientas deben usarse desde una ubicación externa para probar exactamente qué puertos están escuchando en el firewall. netstat es un buen indicador de los servicios que se ejecutan en la máquina. Recuerde, si el servicio no es absolutamente necesario, debe deshabilitarlo y considerar eliminarlo por completo, especialmente de un firewall. Deje que los cortafuegos sean cortafuegos: no deberían ejecutar servicios adicionales.

netstat -a [-n -p -A inet]

netstat reporta información diversa sobre el estado de la red. Se documentan bastantes opciones de línea de comandos para seleccionar qué información reporta netstat. Las siguientes opciones son útiles para identificar puertos abiertos, informar si están en uso activo y por quién, e informar qué programas y qué procesos específicos están escuchando en los puertos:

- `-a` enumera todos los puertos que están en uso activo o que los servidores locales están escuchando.
- `-n` muestra los nombres de host y los identificadores de puerto en formato numérico. sin la `-n` opción, los nombres de host y los identificadores de puerto se muestran como nombres simbólicos, tanto como quepan en 80 columnas. El uso de `-n` evita una espera potencialmente larga mientras se buscan los nombres de los hosts remotos. No usar `-n` produce una lista más legible.
- `-p` enumera el nombre del programa que escucha en el socket. Debe iniciar sesión como raíz para usar la opción `-p`.
- `-A` inet especifica la familia de direcciones informada. La lista incluye los puertos en uso, ya que están asociados con sus tarjetas de interfaz de red. Las conexiones de socket de la familia de direcciones locales no se informan, incluidas las conexiones basadas en la red local que utilizan los programas (como cualquier programa X Window que pueda estar ejecutando).

Tipos de sockets: TCP/IP y Linux

Los sockets se introdujeron en BSD 4.3 UNIX en 1986 y los conceptos han sido adoptados en gran medida por Linux. Dos tipos principales de sockets eran el dominio de Internet, `AF_INET`, y los sockets de dominio UNIX, `AF_UNIX`. `AF_INET` es el socket TCP/IP utilizado en una red. `AF_UNIX` es un tipo de socket local para el kernel. El tipo de socket de dominio UNIX se utiliza para la comunicación entre procesos en la misma computadora; es más eficiente que usar TCP/IP para sockets locales. No sale nada en la red.

La siguiente salida de netstat está limitada a los sockets de dominio INET. La lista informa todos los puertos que están siendo escuchados por los servicios de red, incluido el nombre del programa y el ID de proceso específico del programa de escucha:

```
> netstat -a -p -A inet
```

1 Conexiones a Internet activas (servidores y establecidas)

2 Proto Recv-Q Enviar-Q Dirección local

Nombre del programa

Dirección Extranjera Estado PID/

```

3 tcp      0      143 interno: ssh          servidor de red: 62360 ESTABLECIDO
15392/sshd
4 tcp 0 *:smtp 3674/sendmail: acceso          *: *          ESCUCHA

interno:dominio 6 tcp *: * 0 localhost: www: 80 tcp 0      0          ESCUCHA 638/httpd
interno:dominio 10 udp *: * 0 localhost: www: 53 udp 0      0          ESCUCHAR 588/nombrado
0      0          ESCUCHAR 588/nombrado
0      0          ESCUCHA 574/xinetd
0      0          588/nombrado
0      0          588/nombrado
0      0          588/nombrado

```

La línea 1 identifica que la lista incluye servidores locales y conexiones a Internet activas.

Esta selección se indicó con la opción -A inet a netstat.

La línea 2 contiene estos encabezados de columna:

• Proto hace referencia al protocolo de transporte sobre el que se ejecuta el servicio, TCP o UDP.

• Recv-Q es el número de bytes recibidos del host remoto pero aún no entregados al programa local.

• Send-Q es el número de bytes enviados desde el programa local que aún no han sido reconocidos por el host remoto.

• La dirección local es el par de socket local, interfaz de red y puerto de servicio.

• La dirección externa es el socket remoto, la interfaz de red remota y el servicio par de puertos.

• State es el estado de conexión del socket local para sockets que usan el protocolo TCP, ya sea conexión ESTABLECIDA o LISTENing para una solicitud de conexión, así como una serie de estados intermedios de establecimiento y cierre de conexión.

• PID/Nombre del programa es el ID del proceso (PID) y el nombre del programa que posee el enchufe.

La línea 3 muestra que se establece una conexión SSH a través de la interfaz de red LAN interna desde una máquina conocida como servidor de red. El comando netstat se escribió desde esta conexión.

La línea 4 es un sendmail que escucha el correo entrante en el puerto SMTP asociado con todas las interfaces de red, incluida la interfaz externa conectada a Internet, la interfaz LAN interna y la interfaz localhost de loopback.

La línea 5 muestra que un servidor web local está escuchando conexiones en la interfaz externa. cara a Internet.

La línea 6 muestra que el servidor de nombres está escuchando en la interfaz LAN interna las solicitudes de conexión de búsqueda de DNS de las máquinas locales a través de TCP.

La línea 7 muestra que el servidor de nombres está escuchando en la interfaz de bucle invertido para DNS buscar solicitudes de conexión de clientes en esta máquina a través de TCP.

La línea 8 muestra que xinetd está escuchando conexiones en el puerto POP asociado con todas las interfaces en nombre de popd. (xinetd está escuchando en todas las interfaces las conexiones POP entrantes. Si llega una solicitud de conexión, xinetd inicia un servidor popd para dar servicio a la

solicitud). El firewall y los mecanismos de seguridad de nivel superior en el nivel tcp_wrappers y el nivel de configuración popd limitan las conexiones entrantes a las máquinas LAN.

La línea 9 muestra que el servidor de nombres está escuchando en todas las interfaces para el servidor DNS comunicaciones del servidor y está aceptando solicitudes de búsqueda locales a través de UDP.

La línea 10 muestra que el servidor de nombres está escuchando en la interfaz de red LAN interna para comunicaciones de servidor a servidor DNS y solicitudes de búsqueda a través de UDP.

La línea 11 muestra que el servidor de nombres está escuchando en la interfaz de bucle invertido para DNS solicitudes de búsqueda de clientes locales en esta máquina a través de UDP.

Convenciones de generación de informes de salida de netstat

En la salida de netstat, las direcciones locales y externas (es decir, remotas) se enumeran como <dirección:puerto>. En la columna Dirección local, la dirección es el nombre o la dirección IP de una de sus tarjetas de interfaz de red. Cuando la dirección aparece como *, significa que el servidor está escuchando en todas las interfaces de red en lugar de en una sola interfaz. El puerto es el identificador de puerto de servicio numérico o simbólico que utiliza el servidor. En la columna Dirección externa, la dirección es el nombre o la dirección IP del cliente remoto que participa actualmente en una conexión. El *.* se imprime cuando el puerto está inactivo o para el demonio predeterminado. El puerto es el puerto del cliente remoto en su extremo.

Se informa que los servidores inactivos que escuchan el protocolo TCP escuchan una solicitud de conexión. Los servidores inactivos que escuchan el protocolo UDP se informan como en blanco. UDP no tiene estado: la salida de netstat simplemente hace una distinción entre TCP con estado y UDP sin estado.

Comprobación de un proceso vinculado a un puerto particular con fuser

El comando fuser identifica qué procesos están utilizando un archivo, sistema de archivos o puerto de red en particular. netstat -a -A inet informará un número de puerto en lugar de un nombre de servicio si el puerto no tiene una entrada en /etc/services. fuser puede ser útil para determinar qué programa está vinculado a ese puerto.

El formato de comando general del fuser para identificar qué programa está vinculado a un puerto determinado es el siguiente:

```
fuser -n tcp|udp -v <número de puerto>[,<dirección remota>[,<puerto remoto>]
```

Por ejemplo,

```
> fuser -n tcp -v 515
```

produce la siguiente salida:

	USUARIO raíz	COMANDO DE ACCESO PID
515/tcp	718 f.... lpd	

La opción -v genera los campos USUARIO, ACCESO y COMANDO . sin la -v opción, se informará el puerto/protocolo y el PID . Necesitaría usar ps para identificar el programa al que se le asignó ese ID de proceso.

Los códigos de campo de acceso se refieren al tipo de acceso al que el proceso accede al archivo o sistema de archivos. La f indica que el objeto está abierto.

La siguiente sección describe Nmap.

Nmap

Nmap es una herramienta de auditoría de seguridad de red mucho más poderosa que incluye muchas de las nuevas técnicas de escaneo sigiloso que se usan en la actualidad. Debe verificar la seguridad de su sistema con Nmap; es un hecho que otras personas lo harán. Nmap está disponible en <http://www.insecure.org/nmap/>. Debe usar Nmap desde un host fuera de su firewall para verificar que el firewall no esté escuchando en puertos inesperados.

El siguiente ejemplo de salida de Nmap informa el estado de todos los puertos TCP y UDP. Debido a que no se usa la opción detallada, Nmap informa solo los puertos que están abiertos y que tienen servidores escuchando en ellos. La salida de Nmap incluye el nombre de host escaneado, la dirección IP, el puerto, el estado abierto o cerrado, el protocolo de transporte en uso en ese puerto y el nombre de puerto de servicio simbólico de `/etc/services`. Debido a que choke es un host interno, ssh adicional y los puertos ftp están abiertos para acceso LAN interno:

```
> enrutador nmap-sT
```

```
Iniciando nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
```

```
Puertos interesantes en choke.private.lan (192.168.1.2):
```

```
(Los puertos 3100 escaneados pero que no se muestran a continuación están en estado: filtrado)
```

Puerto	Estado	Servicio
21/tcp	abierto 22/	ftp
tcp	abierto 53/tcp	ssh
abierto 80/tcp		dominio
abierto 443/tcp		http
abierto		https

```
Ejecución de Nmap completada: 1 dirección IP (1 host arriba) escaneada en 236 segundos
```

Resumen

Este capítulo presentó el mecanismo de lista de reglas de iptables, el puerto de Linux y la información del demonio de red disponible a través de netstat, y una herramienta de terceros disponible para verificar que las reglas del firewall estén instaladas y funcionen como usted espera.

Este capítulo hizo hincapié en las reglas del cortafuegos y los puertos que protegen. El Capítulo 10, "Redes privadas virtuales", cambia el enfoque de los cortafuegos al tema más amplio de la seguridad de redes y sistemas.

Esta página se dejó en blanco intencionalmente

10

Redes Privadas Virtuales

Contribución de Carl B. Constantine

El uso de redes privadas virtuales, o VPN, se está convirtiendo rápidamente en el método preferido para acceder a redes remotas y privadas por parte de usuarios domésticos y empresariales por igual. En este capítulo se analizan las VPN y se brindan algunos antecedentes sobre las VPN en sí mismas y una idea de cómo podría implementar una VPN con Linux.

Descripción general de las redes privadas virtuales

Los sistemas VPN están diseñados para conectar dos o más dispositivos o redes de forma segura a través de una red pública como Internet. Una VPN se llama así porque es virtual y utiliza una infraestructura ya existente; es privado, teniendo los datos encapsulados a través de un protocolo seguro; y es una red, porque conecta dos o más dispositivos o redes entre sí. Las VPN son populares hoy en día porque brindan una mejor propuesta de valor que configurar conexiones arrendadas individuales entre ubicaciones, especialmente para guerreros de la carretera u otras conexiones de corta duración. Las VPN también pueden proporcionar un funcionamiento sin problemas. Una vez realizada la configuración inicial, las redes conectadas con una VPN pueden funcionar como si fueran una sola red.

Protocolos VPN

La mayoría de los sistemas VPN utilizan uno de los tres protocolos principales: Protocolo de tunelización punto a punto (PPTP), Protocolo de tunelización de capa 2 (L2TP) o Seguridad IP (IPsec). Esta sección analiza los tres.

PPTP y L2TP

El protocolo de tunelización punto a punto fue diseñado y desarrollado originalmente por un consorcio de empresas para encapsular protocolos que no son TCP/IP, como IPX, a través de Internet mediante la encapsulación de enrutamiento genérico (GRE). La seguridad en el protocolo se agregó más tarde.

El protocolo de tunelización de capa 2 (L2TP) es ampliamente considerado como el sucesor de PPTP para la mayoría de los entornos.

Encapsulación de enrutamiento genérico

Actualmente hay muchos protocolos disponibles que están diseñados para encapsular u ocultar un protocolo en otro, normalmente IP. GRE está diseñado para ser más genérico (de ahí el nombre) que estos otros protocolos. Sin embargo, como tal, puede que no se ajuste a la necesidad de encapsular específicamente el protocolo X sobre el protocolo Y; en cambio, está diseñado para ser un protocolo de encapsulación simple y de propósito general que reduce la sobrecarga de proporcionar encapsulación. RFC 2784, "Encapsulación de enrutamiento genérico (GRE)", describe GRE en detalle.

PPTP y L2TP son muy populares en muchos entornos corporativos, particularmente aquellos que están centrados en Windows. Hay clientes PPTP y L2TP para Windows, Linux, OS X y también para las principales plataformas móviles.

IPsec

IPsec se diseñó teniendo en cuenta la seguridad y se considera el estándar de facto para la comunicación privada segura en redes públicas como Internet. Como se mencionó anteriormente, IPsec se ha incluido en IPv6 y también se puede utilizar en el estándar IPv4 actual.

IPsec proporciona integridad de datos, autenticación y confidencialidad. Todos los servicios IPsec son en la capa IP y proporciona protección para IP y protocolos de capa superior. Estos servicios son proporcionados por dos protocolos de seguridad de tráfico, el encabezado de autenticación (AH) y la carga útil de seguridad de encapsulación (ESP). IPsec utiliza un sistema de administración de claves criptográficas a través del protocolo de Intercambio de claves de Internet (IKE) y un sistema de conexión de la Asociación de seguridad (SA) administrado.

IPsec ofrece muchas ventajas en comparación con otros métodos seguros de acceso a la red. Una de las mayores ventajas es que IPsec puede funcionar en segundo plano sin que el usuario sepa lo que está sucediendo.

Encabezado de autenticación

Los paquetes IP normales constan de un encabezado y una carga útil. El encabezado contiene las direcciones IP de origen y destino que se requieren para el enrutamiento. La carga útil consiste en información que puede ser confidencial. Los encabezados pueden falsificarse o modificarse mediante un tipo de ataque man-in-the-middle. El AH en realidad firma digitalmente el paquete de salida, verificando la identidad de las direcciones de origen y destino y la integridad de los datos de carga útil.

AH solo proporciona autenticación, no cifrado, y se puede configurar de dos maneras: en modo de transporte o en modo de túnel. El modo de transporte realmente se aplica solo a la implementación del host y brinda protección para los protocolos de nivel superior, así como para los campos de encabezado IP seleccionados. Usando el modo de transporte, el AH se inserta después del encabezado IP y antes del protocolo de capa superior (TCP, UDP, ICMP, etc.) o antes de otros encabezados IPsec que ya se hayan insertado.

El AH en modo túnel protege todo el paquete IP, incluido el encabezado IP interno completo. Como en el modo de transporte, el AH se inserta después del encabezado IP externo del paquete.

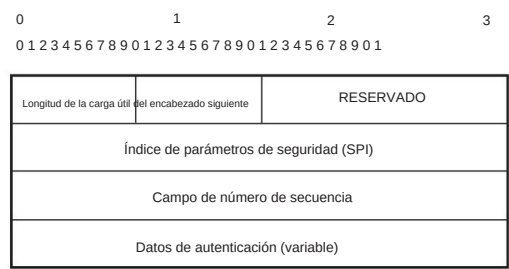


Figura 10.1 El formato del encabezado AH

El AH se inserta después del encabezado IP. En las implementaciones de IPv4, el encabezado IP con
 Contiene el número de protocolo 51 (AH). El AH se muestra en la Figura 10.1.

Todos los campos en formato AH deben estar siempre presentes y se incluyen en el cálculo del valor de
 verificación de integridad (ICV).

Carga útil de seguridad encapsulada

El uso de ESP garantiza la integridad y confidencialidad de los datos en el mensaje original por medio de un
 cifrado seguro de la carga útil original por sí misma o la combinación de los encabezados y la carga útil del
 paquete original.

ESP se puede utilizar en modo de transporte o modo de túnel, como AH, para proporcionar cifrado y
 autenticación. El modo de transporte solo se aplica a las implementaciones de host. Proporciona protección para
 los protocolos de capa superior, pero no para el encabezado IP. Para el modo de túnel, el ESP se inserta después
 del encabezado IP y antes de cualquier protocolo de capa superior, como TCP y UDP, o antes de cualquier otro
 encabezado IPsec que ya se haya insertado. En la implementación IPv4 actual de TCP/IP, el ESP se coloca
 después del encabezado IP pero antes del protocolo de la capa superior. Esto hace que ESP sea compatible con
 hardware no compatible con IPsec.

El modo de túnel de ESP se puede usar en hosts o puertas de enlace de seguridad. Debe usar ESP en modo
 túnel si implementa una puerta de enlace de seguridad. En el modo de túnel, el encabezado de IP interno
 contiene las direcciones de origen y de destino adecuadas, mientras que el encabezado de IP externo puede
 contener distintas direcciones de IP, como direcciones de puertas de enlace de seguridad. ESP protege todo el
 paquete en modo túnel, incluido el encabezado IP interno. La posición del paquete ESP es similar a la del modo
 de transporte.

ESP puede utilizar una amplia variedad de algoritmos de cifrado para servicios de seguridad.

Modos de transporte y túnel

En el modo de transporte, la puerta de enlace IPsec es el destino de los paquetes protegidos: una
 máquina actúa como su propia puerta de enlace. En el modo de túnel, una puerta de enlace IPsec brinda
 protección para los paquetes hacia y desde otros sistemas.

El ESP se inserta después del encabezado IP. En las implementaciones de IPv4, el encabezado IP con
 Contiene el número de protocolo 50 (ESP). La Figura 10.2 muestra un ejemplo de un ESP.

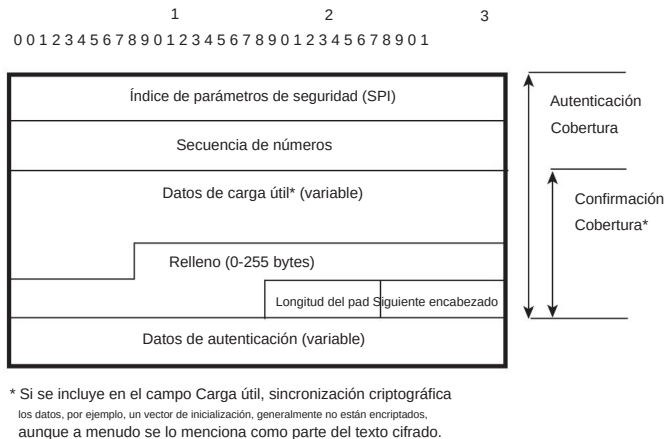


Figura 10.2 El formato ESP

Intercambio de claves de Internet

El Intercambio de claves de Internet, o IKE, es una parte importante de una VPN IPsec. IKE en sí mismo es un protocolo híbrido y permite la negociación y autenticación de material con clave para asociaciones de seguridad de manera protegida.

Asociaciones de seguridad

Para tener un tráfico seguro, debe haber dos asociaciones de seguridad, una para cada dirección.

La asociación de seguridad es esencialmente un canal unidireccional negociado por el sistema IPsec de nivel superior y utilizado por el nivel inferior.

Una asociación de seguridad se define por tres cosas:

- La dirección IP de destino
- El protocolo (AH o ESP)
- El índice de parámetros de seguridad (SPI)

Una SA se puede utilizar en modo de transporte o en modo de túnel. Una SA en modo de transporte es una asociación de seguridad entre dos hosts. Una SA en modo túnel es una asociación de seguridad aplicada a un túnel IP. Si cualquiera de los extremos de una SA es una puerta de enlace de seguridad, la SA es una asociación de seguridad en modo túnel. La asociación de seguridad entre dos puertas de enlace de seguridad es siempre la SA en modo túnel, al igual que una SA entre un host y una puerta de enlace de seguridad.

Productos Linux y VPN

Linux tiene una serie de soluciones sólidas de VPN, con compatibilidad con IPsec disponible a partir de la serie del kernel de Linux 2.6 y posteriores. Esta sección analiza algunos de los software VPN de Linux.

Openswan/Libreswan

Openswan y su bifurcación Libreswan son implementaciones de código abierto de software VPN que funcionan muy bien con Linux. Openswan y/o Libreswan se incluyen con muchas distribuciones de Linux, incluidas Fedora, Debian, Ubuntu y Red Hat. Openswan/Libreswan se encuentran entre las implementaciones de Linux de software VPN más fáciles de configurar. Se puede encontrar más información en <http://www.openswan.org/> y <https://libreswan.org>.

OpenVPN

OpenVPN es una implementación popular de VPN para Linux, con software también disponible para Windows y OS X. OpenVPN usa clave estática y autenticación TLS y también tiene una variedad de opciones para ejecutar un servidor y escenarios de VPN de cliente a cliente.

PPTP

El demonio PPTP, pptpd, normalmente proporciona soporte para PPTP. La configuración de pptpd es bastante sencilla, pero adolece de una falta de potencia general en sí misma en comparación con otras soluciones de servidor VPN de Linux como OpenVPN.

VPN y cortafuegos

Una VPN puede colocarse frente a un firewall, detrás de un firewall o ser parte de una implementación de firewall. No es muy común colocar la VPN frente a un firewall. Es más común usar un combo de firewall/VPN o poner la VPN detrás del propio firewall.

Combinar un sistema VPN y un firewall es una de las soluciones más flexibles. Requiere menos hardware pero también crea un único punto de falla. Una solución más robusta es tener una VPN detrás del firewall o como parte de una configuración de DMZ.

Si su firewall también realiza NAT, es posible que tenga algunos problemas con algunas configuraciones de VPN. En particular, su firewall debe configurarse para enrutar paquetes según el protocolo (GRE, AH, ESP) en lugar de solo en el puerto.

Un NAT/cortafuegos suele ser incompatible con el protocolo AH independientemente del modo (transporte o túnel). Las VPN IPsec que usan AH firman digitalmente el paquete saliente, tanto la carga de datos como los encabezados, con un valor hash adjunto al paquete. AH no cifra el contenido del paquete (carga útil de datos). Si un NAT/cortafuegos se encuentra entre los extremos de IPsec, vuelve a escribir la dirección de origen o la dirección de destino con una propia (dependiendo de la configuración de NAT). La VPN en el extremo receptor intenta verificar la integridad del paquete entrante calculando su propio valor hash y se queja de que el valor hash agregado al paquete no coincide. La VPN, sin darse cuenta del NAT/cortafuegos en el medio, piensa que el paquete ha sido alterado.

Puede usar IPsec con ESP en modo túnel con autenticación. ESP en modo túnel encapsula todo el paquete original (incluidos los encabezados) en un nuevo paquete IP. La dirección de origen del nuevo paquete es la dirección de salida de la puerta de enlace VPN de envío,

y su dirección de destino es la dirección de entrada de la VPN en el extremo receptor. Cuando se utiliza ESP en modo túnel con autenticación, el contenido del paquete se cifra. Los contenidos cifrados (el paquete original), no los nuevos encabezados, se firman con un valor hash adjunto al paquete.

Las comprobaciones de integridad se realizan sobre la combinación del encabezado original más la carga útil original. Si está utilizando ESP en modo túnel con autenticación, el NAT/firewall no los cambia.

Un NAT/firewall puede interferir con IPsec (tanto AH como ESP) si impide que las dos puertas de enlace VPN negocien con éxito asociaciones de seguridad mediante ISAKMP/IKE con certificados X.509. Si las dos puertas de enlace VPN intercambian certificados firmados que vinculan la identidad de cada puerta de enlace con su dirección IP, la reescritura de la dirección NAT hará que la negociación IKE falle.

Es por esta razón que las configuraciones combinadas de VPN y firewall se están volviendo tan popular. Las reglas para gestionar esta situación se pueden configurar y mantener fácilmente.

Resumen

Las redes privadas virtuales son populares porque aprovechan la infraestructura existente para brindar una experiencia de red perfecta a los usuarios finales. Están disponibles numerosas implementaciones de VPN, aprovechando los diferentes protocolos disponibles para crear VPN. Como era de esperar, Linux tiene varias opciones disponibles, Openswan y OpenVPN, entre otras.

Existen algunos problemas al conectar VPN a través de firewalls habilitados para NAT. Esto se debe a que IPsec crea una firma digital basada en el encabezado de IP, que se modifica durante el proceso de NAT.

Más allá de iptables y nftables

11 Detección y respuesta a intrusiones

12 herramientas de detección de intrusos

13 Supervisión de red y detección de ataques

14 Integridad del sistema de archivos

Esta página se dejó en blanco intencionalmente

Detección y respuesta a intrusiones

Ahora ha creado un firewall con Linux usando iptables o nftables. La seguridad en capas. El enfoque de seguridad incluye seguridad basada tanto en la red como en el host. Cuando el cortafuegos proporciona seguridad tanto para la red como para los hosts, también hay pasos que deben tomarse en la propia máquina del cortafuegos, así como en los hosts dentro de la red. Ya sea que tome la forma de verificación de integridad del sistema de archivos, análisis de virus/rootkit o monitoreo de la red en busca de actividad sospechosa, estos procesos ayudan a garantizar que sus datos permanezcan seguros.

Este capítulo trata sobre la seguridad del host y la red y la detección de intrusos. El objetivo de este capítulo es proporcionar una descripción general de alto nivel de algunos de los conceptos para que pueda realizar más investigaciones sobre las áreas específicas de interés. El capítulo amplía el alcance más allá de la máquina cortafuegos para incluir la seguridad de la red, así como sugerencias para computadoras individuales dentro de la red.

Detección de intrusiones

¿Cómo sabes cuando has sido atacado con éxito? Los administradores y los analistas de intrusiones se han planteado esa pregunta durante mucho tiempo. Los métodos utilizados para detectar ataques exitosos solían ser más arte que ciencia. Afortunadamente, ahora hay varias herramientas disponibles para hacer que la detección de intrusos sea mucho más una ciencia que un arte.

Dicho esto, la principal herramienta para la detección de intrusos sigue siendo un ser humano que puede recopile datos de varias fuentes y tome una decisión inteligente y educada sobre el significado de los datos. Las herramientas actuales son sofisticadas y pueden realizar parte de esta correlación por sí mismas, pero el verdadero valor de un analista de intrusiones se demuestra en la capacidad de evaluar la situación y presentar las posibles causas y efectos.

Muchas veces, se detecta un ataque cuando se informa una interrupción del servicio. De esta manera, es importante monitorear activamente sus servicios usando un paquete como Nagios. Al monitorear activamente tantos servicios como sea posible, puede detectar rápidamente una anomalía que amerita una mayor investigación.

Si ejecuta un servidor web, en lugar de monitorear simplemente si el servidor está escuchando (generalmente en el puerto TCP 80), debe monitorear texto específico en una o más páginas web. Si usted

monitoree solo el estado del servidor y si está escuchando, no detectará una desfiguración del sitio web. En esencia, debe monitorear el comportamiento de los servicios específicos para asegurarse de que se ejecuten como se esperaba en lugar de asegurarse de que simplemente se estén ejecutando.

También es importante monitorear recursos como el espacio en disco, el uso de memoria y el promedio de carga. Supervisar estos recursos puede indicar si un proceso se ha escapado y está consumiendo demasiados recursos (como podría ser el caso de un exploit mal escrito). Además, monitorear el espacio en disco es otra tarea útil. Si normalmente consume el 25 % del disco y, de repente, el uso del disco salta al 85 %, querrá investigar para ver si un atacante está usando el servidor como punto de entrega de archivos.

El monitoreo básico del servicio, realizado tanto como pueda, con la mayor frecuencia posible, ayudará proporcionar una alerta temprana de anomalías. Los servicios de monitoreo también ayudarán a mejorar la confiabilidad de los servicios, dejando de lado todas las consideraciones de seguridad. Sin embargo, el monitoreo no debe reemplazar las herramientas de detección de intrusos como Snort o Suricata, ni debe reemplazar una buena política de seguridad implementada a través de una estrategia profunda.

Una vez detectada una anomalía, ya sea a través de la supervisión normal del servicio o por otros medios, depende de usted investigar la anomalía. Su investigación debe ajustarse a la política de seguridad que tiene implementada. Una de las primeras respuestas probablemente sería determinar si realmente se ha producido una intrusión. Puede haber muchas razones por las que el promedio de carga se disparó o por las que aumentó el uso del disco, por lo que no debe suponer que se ha producido un ataque simplemente debido a una alerta de interrupción.

Determinar la causa raíz de una interrupción del servicio es una tarea difícil que generalmente termina con el reinicio de un servicio o la realización de algún procedimiento de rutina similar para solucionar la interrupción. Sin embargo, es importante buscar las causas subyacentes de dichas interrupciones para garantizar que no se esté produciendo un ataque o que no se haya producido ya. Es en esta área, la correlación de eventos, donde un ser humano es más necesario. Por ejemplo, ¿la partición del disco se quedó sin espacio porque un atacante está usando el espacio o porque los archivos de registro llenaron la partición?

Síntomas que sugieren que el sistema podría estar comprometido

A menudo, los atacantes exitosos intentarán ocultar sus huellas y, por lo tanto, la simple supervisión del servicio no será de ayuda. Los atacantes pueden ser mucho más hábiles para ocultar sus huellas que usted para rastrear estados anómalos del sistema.

Los sistemas Linux son demasiado diversos, personalizables y complicados para definir una lista completa y completa de síntomas definitivos que demuestren que el sistema está comprometido.

Al igual que con cualquier tipo de trabajo de detective o diagnóstico, debe buscar pistas donde pueda, de la manera más sistemática posible. RFC 2196, "Manual de seguridad del sitio", proporciona una lista de señales para verificar. Aunque no se mantiene, los "Pasos para recuperarse de un compromiso del sistema UNIX o NT", disponibles en CERT en http://www.cert.org/historical/tech_tips/win-unix-system_compromise.cfm ?, proporcionan otra lista de anomalías para comprobar.

Las siguientes secciones incorporan y amplían las ideas que se encuentran en ambas listas, incluyendo todos o la mayoría de sus puntos de una forma u otra. Las anomalías se han clasificado aproximadamente en las siguientes: indicaciones relacionadas con los registros del sistema; cambios en la configuración del sistema; cambios relacionados con el sistema de archivos, el contenido de los archivos, los permisos de acceso a los archivos y el tamaño de los archivos; cambios en cuentas de usuario, contraseñas y acceso de usuario; problemas indicados en los informes de auditoría de seguridad; y degradación inesperada del rendimiento. Las indicaciones anómalas a menudo cruzan los límites de categoría.

Indicaciones de registro del sistema

Las indicaciones del registro del sistema incluyen errores inusuales y mensajes de estado en los registros, archivos de registro truncados, archivos de registro eliminados e informes de estado enviados por correo electrónico:

- **Archivos de registro** del sistema: las entradas sin explicación en los archivos de registro del sistema, los archivos de registro reducidos y los archivos de registro faltantes sugieren que algo anda mal. Por ejemplo, `/var/registro/` Los mensajes contienen la mayor parte de la información de registro del sistema en la mayoría de los sistemas Linux. Si ese archivo de registro tiene un tamaño cero o le faltan porciones grandes, se justifica una investigación adicional.
- **Informes de estado del demonio del sistema** : en lugar de (o además de) escribir en los archivos de registro, algunos demonios, como `crond` , envían informes de estado por correo electrónico. Tener informes inusuales o faltantes sugiere que algo no está bien.
- **Mensajes anómalos de consola y terminal** : mensajes sin explicación, posiblemente destinadas a anunciar la presencia del hacker, durante una sesión de inicio de sesión son obviamente sospechosas.
- **Intentos de acceso repetidos** : los intentos continuos de inicio de sesión o los intentos de acceso ilegal a archivos a través de FTP o un servidor web, en particular los intentos de subvertir una aplicación web, son sospechosos cuando los intentos son persistentes, incluso si parecen terminar en fallas repetidas.

Indicaciones de configuración del sistema

Las indicaciones de configuración del sistema incluyen archivos de configuración modificados y secuencias de comandos del sistema, procesos no deseados que se ejecutan de manera inexplicable, uso y asignaciones inesperados del puerto de servicio y cambios en el estado operativo del dispositivo de red:

- **Trabajos cron** : verifique los scripts de configuración cron y los ejecutables para modificarlos.
- **Archivos de configuración del sistema alterados**: una verificación de integridad del sistema de archivos, manual o con una herramienta como se describe en el Capítulo 14, "Integridad del sistema de archivos", indicaría archivos de configuración modificados en `/etc`. Estos archivos son fundamentales para el correcto funcionamiento del sistema. Es importante verificar cualquier cambio en un archivo (como en `/etc/`, como `/etc/passwd`, `/etc/group` y archivos similares).

240 Capítulo 11 Detección y respuesta a intrusiones

• **Servicios y procesos sin explicación, como se muestra en ps:** los programas que se ejecutan inesperadamente son una mala señal. Tenga en cuenta que, como parte del ataque, es posible que se haya reemplazado el comando ps. Más sobre esto más adelante.

• **Conexión inesperada y uso de puerto inesperado, como lo muestra netstat**

- tcpdump: el tráfico de red inesperado es una mala señal.

• **Fallos del sistema y procesos faltantes:** los fallos del sistema, así como los fallos inesperados del servidor, pueden ser sospechosos. Un bloqueo del sistema también puede sugerir un reinicio del sistema iniciado por un atacante, lo que podría ser necesario para reiniciar ciertos procesos críticos del sistema después del reemplazo con una versión troyana.

• **Cambios en la configuración del dispositivo:** reconfiguración de una interfaz de red para estar en

El modo promiscuo o de depuración es una señal de que se ha instalado un rastreador de paquetes.

Indicaciones del sistema de archivos

Las indicaciones del sistema de archivos incluyen archivos y directorios nuevos, archivos y directorios que faltan, contenidos de archivos alterados, discrepancias de md5sum o sha1sum, nuevos programas setuid y sistemas de archivos que crecen rápidamente o se desbordan:

• **Nuevos archivos y directorios:** además de archivos con firmas digitales repentinamente malas, es posible que descubra nuevos archivos y directorios. Especialmente sospechosos son los nombres de archivo que comienzan con uno o más puntos y los nombres de archivo que suenan legítimos que aparecen en lugares poco probables.

• **Programas setuid y setgid:** los archivos setuid nuevos y los archivos setgid recién configurados son un buen lugar para comenzar a buscar problemas ocultos.

• **Archivos que faltan:** los archivos que faltan, en particular los archivos de registro, indican algún tipo de problema.

• **Tamaños de sistemas de archivos que cambian rápidamente, como se muestra en df:** si la máquina está comprometida, los sistemas de archivos que crecen rápidamente pueden ser una señal de que el programa de monitoreo de un pirata informático está generando archivos de registro de gran tamaño.

• **Archivos de archivos públicos modificados:** verifique el contenido de sus áreas web y FTP en busca de archivos nuevos o modificados.

Indicaciones de cuenta de usuario

Las indicaciones de cuentas de usuario incluyen nuevas cuentas de usuario, cambios en el archivo passwd, actividad inusual en los informes de contabilidad de procesos del usuario o informes de contabilidad de procesos faltantes, cambios en los archivos de usuario, especialmente archivos ambientales, y pérdida de acceso a la cuenta:

• **Cuentas de usuario nuevas y modificadas:** cuentas nuevas en /etc/passwd y pro

Los procesos que se ejecutan con ID de usuario nuevos o inesperados, como se muestra en ps, son indicaciones de cuentas nuevas. Las cuentas a las que les faltan contraseñas repentinamente indican una cuenta abierta.

• **Registros contables** de usuarios: los informes contables de usuarios inusuales, los inicios de sesión inexplicables, los archivos de registro faltantes o editados (como /var/log/lastlog, /var/log/pacct o /var/log/usracct) y la actividad irregular de los usuarios son signos de problemas

Qué hacer si su sistema está comprometido 241

• **Cambios en las cuentas raíz o de usuario:** un signo grave es el entorno de inicio de sesión de un usuario. que ha sido modificado o dañado hasta el punto de que la cuenta es inaccesible. De particular preocupación son los cambios en la variable de entorno PATH de los usuarios o sshauthorized_keys.

• **Pérdida de acceso a la cuenta:** similar a los cambios en el entorno de inicio de sesión de un usuario es denegación de acceso intencional, ya sea cambiando la contraseña de la cuenta, eliminando la cuenta o, para usuarios regulares, cambiando el nivel de ejecución al modo de usuario único.

Indicaciones de la herramienta de auditoría de seguridad

Las indicaciones de la herramienta de auditoría de seguridad incluyen discrepancias en la integridad del sistema de archivos, cambios en el tamaño de los archivos, cambios en los bits del modo de permisos de archivos, nuevos programas setuid y setgid, alertas de herramientas de detección de intrusos como Snort y datos de monitoreo de servicios.

Los archivos con firmas hash que no coinciden pueden ser archivos nuevos, archivos cuyas longitudes o han cambiado las fechas de creación o modificación, y los archivos cuyos modos de acceso están alterados.

De particular preocupación son los programas troyanos recién instalados. Los objetivos frecuentes para el reemplazo son programas administrados por inetd o xinetd, inetd o xinetd mismo, ls, ps, netstat, ifconfig, telnet, login, su, ftp, syslogd, du, df, sync y libc

Biblioteca.

Indicaciones de rendimiento del sistema

Las indicaciones de rendimiento del sistema incluyen promedios de carga inusualmente altos y disco pesado acceso.

El rendimiento deficiente del sistema sin explicación podría deberse a una actividad de proceso inusual, promedios de carga inusualmente altos, tráfico de red excesivo o acceso pesado al sistema de archivos.

Si su sistema muestra signos de un compromiso exitoso, no entre en pánico. No reinicie el sistema; podría perderse información importante. Simplemente desconecte físicamente el sistema de Internet.

Qué hacer si su sistema está comprometido

Independientemente de cómo se investigue una anomalía, el analista de intrusiones debe tener cuidado al realizar la investigación.

Si los atacantes notan que hay un investigador investigando el mismo sistema, es mucho más probable que corten y quemen para salir del sistema. Si un atacante piensa que lo están siguiendo o monitoreando, el atacante podría comenzar a eliminar todo lo que se interponga en el camino, causando un daño real a los sistemas en cuestión. Un ataque que podría haber resultado solo en la desfiguración de un sitio web podría convertirse repentinamente en la eliminación de particiones enteras si el atacante se da cuenta de la investigación.

Después de que se haya determinado que ha habido un ataque exitoso o que un ataque está actualmente en curso, con frecuencia se producen una serie de respuestas. Estos, por supuesto, dependerán de su política de seguridad.

Si hay espacio de almacenamiento disponible, tome una instantánea de todo el sistema en su estado actual para análisis posterior. Si esa no es una opción para usted, al menos tome una instantánea de los registros del sistema en `/var/log` y los archivos de configuración del sistema en el directorio `/etc` .

Mantenga un registro. Anota todo. Documentar lo que haces y lo que no encuentras solo es bueno para informar el incidente a un equipo de respuesta, su ISP o un abogado, pero también lo ayuda a realizar un seguimiento de lo que ha examinado y lo que queda por hacer.

Si se ha producido un ataque o está en curso, una de las primeras prioridades suele ser detener el ataque y evitar que se produzcan más daños. Teniendo en cuenta que un atacante que detecta a un investigador en el mismo sistema tiene más probabilidades de causar daños colaterales, desconectar el sistema de la red es una recomendación común. Con el cable de red desenchufado, el atacante simplemente no puede causar daños adicionales. Existe, por supuesto, la posibilidad de que el atacante utilice una herramienta para monitorear la interfaz de red y cubra automáticamente sus huellas en caso de que cambie el estado de esa interfaz.

Buscar cambios sutiles en el sistema es parte de esta fase. Los atacantes pueden haber configurado un trabajo cron para reiniciar sus demonios si se detienen. Además, es bastante común que los atacantes reemplacen las utilidades comunes de Linux, como `ls` y `ps` , con sus propias versiones para ocultar sus procesos y archivos. En este sentido, un programa como `Chkrootkit` puede ser útil para detectar intrusiones basadas en host.

Las herramientas que utilice para mitigar el daño estarán determinadas por el tipo de ataque. Por ejemplo, un ataque de denegación de servicio contra un enrutador requerirá necesariamente diferentes pasos para mitigar el ataque. Los pasos que puede seguir para determinar si el sistema está comprometido son los mismos pasos que debe seguir para analizar el sistema comprometido:

1. Verifique los registros del sistema y use `netstat` y `lsof` para ver qué procesos se están ejecutando y qué puertos están vinculados. Compruebe el contenido de los archivos de configuración del sistema. Verifique el contenido y los modos de acceso de todos sus archivos y directorios comprobando sus firmas digitales. Compruebe si hay nuevos programas `setuid` . Compare los archivos de configuración y de usuario con copias de seguridad limpias. Es muy probable que el atacante haya instalado programas troyanos en lugar de las mismas herramientas del sistema que está utilizando para analizar el sistema.
 2. Haga un balance de cualquier información volátil, como qué procesos se están ejecutando y qué puertos están en uso.
 3. Arranque desde una unidad externa o una copia de seguridad del sistema. Examine el sistema utilizando las herramientas limpias del sistema no afectado. Como alternativa, instale las unidades de disco como unidades secundarias en un sistema no comprometido y examine los discos como datos.
 4. Determinar cómo logró el atacante entrar y determinar qué fue hecho a su sistema.
 5. Si es posible, reinstale completamente el sistema desde la distribución original de Linux
- medios de comunicación.
6. Corregir la vulnerabilidad de seguridad, ya sea haciendo una selección más cuidadosa de ejecutar los servicios, reconfigurando los servidores de manera más segura, definiendo listas de acceso en el

xinetd o tcp_wrappers y a nivel de servidor individual, instalando un cortafuegos de filtrado de paquetes o instalando servidores proxy de aplicaciones.

7. Instale y configure cualquier paquete de integridad del sistema.
8. Habilite todos los registros.
9. Restaure los archivos de configuración especiales y de usuario que se sabe que no están contaminados.
10. Cree sumas de verificación MD5 o SHA para los archivos binarios y de configuración estática del sistema.
11. Vuelva a conectar el sistema a la red e instale cualquier nueva actualización de seguridad desde su proveedor de Linux.
12. Cree sumas de verificación MD5 o SHA para los binarios recién instalados y almacene la base de datos de suma de verificación en una unidad USB, CD/DVD o algún otro sistema.
13. Supervise el sistema para detectar intentos de acceso ilegales recurrentes.

Informe de incidentes

Un *incidente* puede ser varias cosas; tienes que definirlo por ti mismo. Por ejemplo, un incidente puede definirse como un intento anómalo de obtener o escalar privilegios o comprometer la confidencialidad, integridad o disponibilidad de uno o más sistemas.

Es una buena práctica monitorear los archivos de registro del sistema, los informes de integridad del sistema y el sistema. informes contables como una cuestión de hábito. Incluso con el registro mínimo habilitado, tarde o temprano verá algo que su política de seguridad dicta que es lo suficientemente importante como para informar.

Con el registro completo habilitado, tendrá muchas entradas de registro para reflexionar las 24 horas del día.

Algunos intentos de acceso son más serios que otros. Algunos te molestarán personalmente más que otros. Las siguientes secciones comienzan discutiendo las razones por las que podría querer informar un incidente y cubren las consideraciones sobre qué tipos de incidentes puede informar. Estas son decisiones individuales. Las secciones restantes se enfocan en los diversos grupos de informes disponibles y el tipo de información que necesita proporcionar si decide informar algo.

¿Por qué reportar un incidente?

Es posible que desee informar un incidente incluso si el intento de ataque no tuvo éxito. Estas son algunas de las razones:

- **Para finalizar los sondeos:** su firewall garantiza que la mayoría de los sondeos permanezcan inofensivos. Pero incluso las sondas inofensivas son molestas si ocurren repetidamente. Los escaneos persistentes, repetidos y continuos llenan sus archivos de registro. Dependiendo de cómo haya definido los activadores de notificación en cualquier software de monitoreo de registros que ejecute, las sondas repetidas pueden molestarlo con notificaciones continuas por correo electrónico. Sin embargo, en la era actual de sondeos aparentemente interminables de bots propiedad de usuarios de banda ancha inconscientes, especialmente aquellos en los Estados Unidos, simplemente llevaría demasiado tiempo para la mayoría de las personas informar cada sondeo.

- **Para ayudar a proteger otros sitios:** las sondas y exploraciones automatizadas generalmente crean una base de datos de todos los hosts vulnerables en un gran bloque de direcciones IP. Cuando se identifican como potencialmente vulnerables a explotaciones específicas, estos hosts son objeto de ataques selectivos. Las sofisticadas herramientas de análisis y craqueo de hoy en día pueden comprometer un sistema vulnerable y ocultar sus rastros en segundos. Informar de un incidente puede detener los escaneos antes de que alguien en otro lugar resulte herido.
- **Para informar al administrador del sistema o de la red:** los sitios atacantes a menudo son sistemas comprometidos, alojan una cuenta de usuario comprometida, tienen software mal configurado, están siendo falsificados o tienen un alborotador individual. Los administradores del sistema generalmente responden a un informe de incidente. Los ISP tienden a detener sus problemas para crear clientes antes de que otros clientes comiencen a quejarse de que los sitios remotos han bloqueado el acceso desde su bloque de direcciones y que no pueden intercambiar correos electrónicos con un amigo o familiar en un sitio remoto.
- **Para recibir la confirmación del ataque:** en ocasiones, es posible que simplemente desee confirmación de que lo que está viendo en los registros es un problema. A veces, es posible que desee la confirmación de que un sitio remoto estaba filtrando paquetes de forma involuntaria debido a una configuración defectuosa. El sitio remoto también suele alegrarse por el aviso de que su red no se está comportando como se esperaba.
- **Para aumentar la conciencia y el control de todas las partes involucradas:** si informar el incidente al sitio atacante, idealmente el sitio monitoreará sus configuraciones y las actividades de los usuarios más cuidadosamente. Si informa el incidente a un centro de abuso, el personal de abuso puede comunicarse con el sitio remoto con más influencia que un automóvil individual ries, esté atento a la actividad continua y ayude mejor a los clientes que se han visto comprometidos. Si informa el incidente a un grupo de noticias de seguridad, otras personas pueden tener una mejor idea de lo que deben tener en cuenta.

¿Qué tipo de incidentes podría informar?

Los incidentes que informe dependen completamente de su tolerancia, qué tan serios considere que son las diferentes investigaciones y cuánto tiempo le importa dedicar contra lo que es una infestación global que crece exponencialmente. Todo se reduce a cómo se define el término *incidente*. En la mente de diferentes personas, los incidentes pueden variar desde simples escaneos de puertos hasta intentos de acceder a sus archivos privados o recursos del sistema, ataques de denegación de servicio, fallas en sus servidores o todo su sistema, para obtener acceso de inicio de sesión raíz a su sistema:

- **Ataques de denegación de servicio:** cualquier tipo de ataque de denegación de servicio es descaradamente loseta. Es difícil no tomar ese ataque como algo personal. Estos ataques son la forma electrónica de vandalismo, obstrucción, acoso y robo de servicio. Debido a que algunas formas de ataques de denegación de servicio son posibles debido a la naturaleza inherente de los dispositivos en red, puede hacer poco o nada sobre algunas formas de ataque además de informar los incidentes y bloquear todo el bloque de direcciones del atacante.

• **Intentos de reconfigurar su sistema:** un atacante no puede reconfigurar sus servidores sin una cuenta de inicio de sesión raíz en su máquina, pero posiblemente podría modificar las tablas relacionadas con la red en la memoria de su sistema, o al menos intentarlo. Los exploits a considerar incluyen estos:

• Transferencias de zona DNS no autorizadas hacia o desde su máquina a través de TCP

• Cambios en sus tablas de enrutamiento en memoria

• Intentos de reconfigurar sus interfaces de red o tablas de enrutamiento a través de sondas para Puerto UDP 161 para snmpd

• **Intentos de obtener acceso a la cuenta** de inicio de sesión : las sondas al puerto 22 TCP ssh son obvias. Menos obvios son los sondeos a puertos asociados con servidores que se sabe que pueden explotarse, ya sea históricamente o actualmente. Los exploits de desbordamiento de búfer generalmente están destinados en última instancia a ejecutar comandos y obtener acceso al shell. El exploit mountd es un ejemplo de esto.

• **Intentos de acceder a** archivos no públicos: los intentos de acceder a archivos privados, como el archivo `/etc/passwd`, los archivos de configuración o los archivos propietarios, aparecen en su registro de FTP (`/var/log/xferlog` o `/var/log/messages`) y en el registro de acceso de su servidor web (`/var/log/httpd/error_log`).

• **Intentos de utilizar servicios privados:** por definición, cualquier servicio que no haya puesto a disposición de Internet es privado. Estos son los servicios privados potencialmente disponibles a través de sus servidores públicos, como los intentos de retransmitir correo a través de su servidor de correo.

Lo más probable es que las personas no hagan nada bueno si intentan usar su máquina en lugar de la suya o la de su ISP. Los intentos de retransmisión aparecen en su archivo de registro de correo (`/var/log/maillog`).

• **Intentos de almacenar archivos en su disco:** si aloja un sitio FTP anónimo mal configurado, es posible que alguien configure un depósito de software o medios robados en su máquina. Los intentos de cargar archivos se registran en su registro de FTP (`/var/log/xferlog`) si `ftpd` está configurado para registrar cargas de archivos.

• **Intentos de bloquear su sistema o** servidores individuales: desbordamiento de búfer
los intentos contra los programas disponibles a través de su sitio web son posiblemente los más fáciles de identificar por los mensajes de error escritos en los archivos de registro del servidor web. Otros informes de datos erróneos aparecerán en su archivo `syslog` general (`/var/log/messages`), su registro de `daemon` general (`/var/log/daemon`), su registro de correo (`/var/log/maillog`), su registro de FTP (`/var/log/xferlog`), o su registro de acceso seguro (`/var/log/secure`).

• **Intentos de explotar vulnerabilidades específicas, conocidas y actualmente explotables—**

Los atacantes encuentran nuevas vulnerabilidades con cada nueva versión de software (y también con las antiguas).

¿A quién informa un incidente?

Tiene una serie de opciones en términos de a quién informar un incidente:

- **root, postmaster o abuso en el sitio infractor** : el lugar obvio para presentar una queja es el administrador del sitio infractor. Informar al administrador del sistema suele ser todo lo que se necesita para solucionar un problema. Sin embargo, esto no siempre es posible porque muchas sondas se originan en direcciones IP falsificadas e inexistentes.
- **Coordinador de red**: si la dirección IP no tiene una entrada de DNS, comuníquese el coordinador del bloque de direcciones de red suele ser útil. El coordinador puede comunicarse con el administrador en el sitio infractor o ponerlo en contacto directo. Si la dirección IP no se resuelve a través de los comandos `host` o `dig`, casi siempre puede encontrar al coordinador de la red proporcionando la dirección a las bases de datos `whois`. El comando `whois` está integrado en la base de datos de ARIN. Tres bases de datos principales están disponibles a través de la web:
 - **ARIN**: el Registro Estadounidense de Números de Internet mantiene la base de datos de direcciones IP para el Hemisferio Occidental, las Américas. ARIN se encuentra en <http://whois.arin.net/ui>.
 - **APNIC**: el Centro de información de la red de Asia Pacífico mantiene la base de datos de direcciones IP para Asia. APNIC se encuentra en <http://www.apnic.net/apnic-bin/whois.pl>.
 - **RIPE**: Réseaux IP Européens mantiene la base de datos de direcciones IP para Europa. RIPE se encuentra en <https://apps.db.ripe.net/search/query.html>.
- **Su centro de abuso de ISP**: si los escaneos se originan dentro del espacio de direcciones de su ISP, su centro de abuso es el lugar para contactar. Su ISP también puede ser útil con los escaneos que se originan en otros lugares, poniéndose en contacto con el sitio infractor en su nombre. Es muy probable que su máquina no sea la única máquina que se prueba en la red del ISP.
- **Su proveedor de Linux**: si su sistema se ve comprometido debido a una vulnerabilidad de software en su distribución, su proveedor querrá saberlo para poder desarrollar y lanzar una actualización de seguridad.

¿Qué información proporciona?

Un informe de incidentes debe contener suficiente información para ayudar al equipo de respuesta a incidentes a localizar el problema. Cuando se comunique con el sitio desde el que se originó el ataque, recuerde que su persona de contacto podría ser la persona que lanzó el ataque intencionalmente. Lo que incluya de la siguiente lista depende de con quién se comunique y de qué tan cómodo se sienta al incluir la información, así como de las políticas de privacidad y de otro tipo que puedan estar vigentes:

• Su dirección de correo electrónico

• Su número de teléfono, si corresponde

- Su dirección IP, nombre de host y nombre de dominio
- Las direcciones IP y los nombres de host, si están disponibles, involucrados en el ataque
- La fecha y la hora del incidente (incluida su zona horaria en relación con GMT)
- Una descripción del ataque
 - Cómo detectó el ataque
 - Entradas representativas del archivo de registro que muestran el incidente
 - Una descripción del formato del archivo de registro
 - Referencias a avisos y avisos de seguridad que describen la naturaleza y significado cancelación del ataque, si procede
- Lo que quiere que haga la persona (arréglo, confírmelo, explíquelo, controlelo o sea informado de ello)

Resumen

Este capítulo se centró en la supervisión de la integridad del sistema y la detección de intrusos. Si sospecha que un sistema podría estar comprometido, puede consultar la lista de posibles indicaciones de problemas de este capítulo. Si ve algunas de estas indicaciones y concluye que el sistema está comprometido, puede utilizar la lista de pasos de recuperación discutidos. Finalmente, se discutieron las consideraciones sobre el informe de incidentes y se dieron indicaciones sobre a quién puede informar un incidente.

El Capítulo 12, "Herramientas de detección de intrusos", analiza la implementación de algunas de las cosas que aprendió en este capítulo al observar las herramientas específicas involucradas en la detección de intrusos y las pruebas del sistema.

Esta página se dejó en blanco intencionalmente

Herramientas de detección de intrusos

En el capítulo anterior, aprendió los conceptos de detección de intrusiones y respuesta a intrusiones. Rara vez dos ataques son exactamente iguales, aunque las técnicas utilizadas con frecuencia se basan en un conjunto común de métodos y dan como resultado muchos de los mismos síntomas, como se describe en el capítulo anterior. Es a través de estos métodos y síntomas comunes que las herramientas de detección de intrusiones pueden ayudar al analista de intrusiones con su trabajo.

El analista de intrusiones tiene mucho para elegir cuando busca herramientas de software para ayudar en la correlación, el diagnóstico y la resolución de problemas. Este capítulo se centra en las herramientas de software utilizadas en la detección de intrusos y en las herramientas que pueden ayudar en el conjunto de herramientas de cualquier administrador. El capítulo comienza con una mirada a los sniffers de red y continúa a través de las herramientas para buscar rootkits y verificadores de sistemas de archivos y monitoreo de archivos de registro.

Kit de herramientas de detección de intrusos: herramientas de red

Algunas de las herramientas principales de los administradores de seguridad y de red son las herramientas de análisis de red. Estos incluyen rastreadores de red, software de detección de intrusos y analizadores de red.

Un rastreador de red es un software que escucha pasivamente el tráfico recibido y enviado por una interfaz de red. TCPDump es lo suficientemente simple como para que los principiantes puedan aprenderlo rápidamente pero lo suficientemente potente como para proporcionar la funcionalidad necesaria para múltiples protocolos en múltiples situaciones. Usando TCPDump, es posible ver el tráfico en numerosos formatos, incluido ASCII, y usar expresiones para ajustar el tráfico exacto que se verá a través de la herramienta.

TCPDump es un software de detección de intrusos manual y primitivo. Si sabe lo que está buscando, TCPDump puede ayudarlo a detectar el tráfico anómalo a medida que pasa por la red. TCPDump en sí mismo no sabrá que un ataque acaba de pasar por debajo de sus narices; ese es el trabajo del analista de intrusiones (así como otro software). Sin embargo, TCP Dump casi siempre se convierte en una herramienta integral para investigar ataques activos porque permite al analista observar el ataque en tiempo real.

TCPDump se cubre en profundidad en el Capítulo 13, "Monitoreo de red y detección de ataques". Allí encontrará cobertura de la actividad normal del protocolo, así como una mirada a algunas hazañas a través de los ojos, o la nariz, por así decirlo, de TCPDump.

Cuando se trata de herramientas que escuchan la red y realizan cierto nivel de análisis del tráfico, Snort es una excelente opción. Snort es una intrusión de clase empresarial y proveedor

software de detección ampliamente implementado y maduro. Snort funciona utilizando el concepto de firmas de intrusión. La teoría es que muchos ataques siguen el mismo patrón o se ven iguales o muy similares a nivel de red.

Considere este ejemplo: suponga que un paquete se recibe en un determinado puerto con sus banderas de encabezado configuradas de cierta manera. Cuando esto ocurre, siempre es un precursor de un ataque o un intento de explotar una determinada vulnerabilidad. Se puede decir que este ataque en particular tiene, por lo tanto, una firma que lo identifica como tráfico malicioso. Esta firma, exclusiva de la explotación de esta vulnerabilidad, puede ser utilizada por software como Snort para detectar que hubo un intento de explotación de la vulnerabilidad. Snort puede entonces realizar una acción basada en esta detección (o no puede realizar ninguna acción).

ntop es un software de análisis de red, a diferencia del sniffer que produce informes de uso basados en protocolo, flujo, host y otros parámetros. Se recomienda usar ntop en puntos estratégicos de la red para establecer una línea de base de los flujos de tráfico normales en la red.

Elegí presentar ntop aquí porque es fácil comenzar a trabajar rápidamente. Sin embargo, yo también recomendar otro software de análisis para el tráfico de red. Entre otros software de análisis, MRTG es otra excelente opción para el análisis de tráfico, al igual que RRDtool y Scrutinizer.

Crear informes de tráfico de referencia y mantenerlos actualizados ayuda no solo a detectar anomalías, incluidos aumentos y disminuciones inesperados en el tráfico, sino también a rastrear cuándo podría ser necesario un nuevo ancho de banda. Es este uso dual (anomalías de seguridad y monitoreo del uso del ancho de banda) lo que hace que el análisis de tráfico sea invaluable.

Para establecer líneas de base de tráfico y monitorear efectivamente la red en busca de intrusiones usando Snort y TCPDump en redes grandes, es importante colocar las herramientas en ubicaciones estratégicas dentro de la red. La mayoría de las redes grandes (incluso las medianas y pequeñas) usan conmutadores para pasar el tráfico. Comprender la diferencia entre conmutadores y concentradores es importante al considerar dónde colocar las herramientas de red.

Conmutadores y concentradores y por qué le importa

En una red conmutada, cualquier interfaz de red determinada recibiría solo el tráfico destinado a ella, así como el tráfico de difusión. En un entorno de red concentrador, la interfaz de red recibe todo el tráfico, ya sea que ese tráfico esté destinado a ella o a otro dispositivo. Esta es la razón por la que las redes conmutadas son más rápidas que las redes concentradas: el tráfico innecesario no se envía a todos los puertos del conmutador.

Hay situaciones en las que una interfaz de red puede recibir todo el tráfico o un subconjunto mayor que el suyo propio en una red conmutada, como cuando un conmutador está configurado para reflejar el tráfico en un puerto específico. En la práctica, esto se puede hacer, pero puede generar problemas de rendimiento para el conmutador porque ahora tiene que copiar todo el tráfico a dos puertos en lugar de a uno. Consulte la documentación de su conmutador para obtener más información.

Independientemente de dónde se origine el tráfico, si entra en la interfaz donde se ejecuta el sniffer, se puede capturar el tráfico. La clave, a nivel de red, es colocar rastreadores y el software de detección de intrusos relacionado en las ubicaciones correctas. Para el rastreo de tráfico basado en host, la ubicación del rastreador es obvia, en el propio host.

Reloj ARP

Otro elemento que se discutirá en el Capítulo 13 es ARPWatch. ARPWatch es un software que busca nuevos dispositivos en la red. ARPWatch puede ser útil para auditar los dispositivos de la red, especialmente las redes inalámbricas.

Comprobadores de rootkits

Un *rootkit* es una pieza de software o una agrupación de software que intenta explotar una o más vulnerabilidades con el objetivo de permitir que un atacante obtenga privilegios elevados o realice cualquier otro tipo de ataque contra el objetivo. Con frecuencia, los atacantes menos hábiles utilizan los rootkits que utilizan el software creado por otro atacante pero que no entienden realmente el exploit subyacente; simplemente están interesados en los resultados.

Muchos rootkits no solo ejecutan el exploit inicial para otorgar al atacante privilegios de raíz, sino que también intentan enmascarar u ocultar el hecho de que se ha lanzado un ataque. Lo hacen eliminando archivos de registro o ciertas entradas de los archivos de registro, instalando versiones de programas troyanos y empleando otros medios. Tampoco hay nada que impida que un atacante encadene rootkits para múltiples niveles de engaño y posible explotación.

Al igual que los ataques centrados en la red, los rootkits suelen tener firmas o dejar otros rastros que los identifican. Estos rastros y firmas pueden ser la eliminación de archivos de registro antes mencionada, la presencia de uno o más procesos u otros cambios en el sistema que son específicos del software rootkit o el exploit.

Además, al igual que con los ataques centrados en la red, también existe un software para buscar firmas y rastros de rootkits. Una de esas aplicaciones es Chkrootkit.

Ejecutar Chkrootkit

Antes de poder ejecutar Chkrootkit, debe obtenerlo. Chkrootkit se puede descargar desde <http://www.chkrootkit.org/> y también se incluye como un paquete con muchas versiones de Linux. Después de descargarlo, Chkrootkit debe desarchivarse y compilarse:

```
tar -zxvf chkrootkit.tar.gz
cd chkrootkit- <NNNN>
tener sentido
```

Sí, eso dice que tiene sentido en el ejemplo del código. Aunque Chkrootkit es un script de shell, se obtiene alguna funcionalidad adicional al compilar el código. No se requiere compilar, pero como es rápido y agrega algunos niveles adicionales de verificación, recomiendo hacerlo. Específicamente, compilar Chkrootkit habilitará estas comprobaciones adicionales, aunque los paquetes estándar de un sistema operativo determinado también pueden tenerlas:

```
• si promisc
• chklastlog
• chkwtmp
• comprobar_wtmpx
```

y chkproc

y chkdirs

y cuerdas

De todas las herramientas utilizadas en este libro, Chkrootkit es probablemente la más fácil de usar. Correr Chkrootkit, desde el directorio fuente de chkrootkit, simplemente escriba esto:

```
./chkrootkit | menos
```

No es necesario canalizar la salida a menos, pero hay una gran cantidad de producción. Entonces, si realmente desea leer la salida, probablemente deba canalizarla en algún lugar, a menos, por supuesto, que tenga un gran búfer de desplazamiento hacia atrás.

Debido a que ejecutar Chkrootkit produce una gran cantidad de salida, es aconsejable canalizar la salida a más o menos, según sus preferencias. Alternativamente, puede redirigir la salida a un archivo:

```
./chkrootkit > salida.txt
```

Chkrootkit generará una serie de líneas que le informan lo que está comprobando actualmente junto con el estado final del cheque. La salida se verá similar a esto:

```
Comprobando 'amd'... no encontrado
Comprobando 'nombre base'... no infectado
Comprobando 'biff'... no infectado
Comprobando 'chfn'... no infectado
Comprobando 'chsh'... no infectado
Buscando el gusano ShitC... no se encontró nada
Buscando el Gusano Omega... no se encontró nada
Buscando Sadmind/IIS Worm... no se encontró nada
Buscando MonKit... no se encontró nada
Buscando a Showtee... no se encontró nada
```

Como puede ver en la muestra de salida, no parece que se hayan detectado archivos troyanos o rootkits. Un archivo infectado o la detección de un rootkit tendrá un aspecto similar al siguiente:

```
Comprobando 'bindshell'... INFECTADO (PUERTOS: 1524 31337)
```

Aunque la salida de Chkrootkit parece indicar que la computadora está infectada con bindshell, Chkrootkit a veces produce falsos positivos. Sin embargo, si ve la salida INFECTADA de Chkrootkit, le conviene asumir que Chkrootkit informó correctamente y tomar medidas para mitigar el daño.

Un falso positivo ocurre cuando una herramienta detecta y reporta un problema cuando en realidad no hay problema. La causa subyacente de los falsos positivos varía según la naturaleza del software que informa la ocurrencia. Los falsos positivos no son tan malos como los falsos negativos. Un falso negativo ocurre cuando realmente hay un problema, pero las herramientas que deberían encontrarlo no lo informan.

Los falsos positivos y negativos no se limitan a la informática. Imagine el caso en el que una persona va al médico y se hace una ecografía. Basándose en los resultados de la exploración, el médico informa que la persona tiene cáncer. Sin embargo, en un examen más detallado parece que la

el informe inicial era incorrecto. Este es un ejemplo de un falso positivo. Aunque se realizaron innecesariamente pruebas adicionales basadas en el falso positivo, todavía es mucho mejor que tener un falso negativo, con el cáncer pasando desapercibido y sin tratamiento.

Debido a que Chkrootkit informa que usa herramientas en la computadora, puede informar un falso negativo. Hay formas de solucionar este problema, como se describe más adelante en esta sección.

¿Qué pasa si Chkrootkit dice que la computadora está infectada?

Si Chkrootkit dice que su computadora está infectada, lo primero que debe hacer es mantener la calma. Aunque no debe asumirlo, existe la posibilidad de que Chkrootkit informe un falso positivo. Si Chkrootkit informa una infección, debe tomar medidas de inmediato para mitigar cualquier daño adicional.

El capítulo anterior del libro analizó la respuesta a incidentes. Por lo tanto, sería redundante cubrir ese mismo material en este capítulo. Sin embargo, como con todas las herramientas de esta naturaleza, los falsos positivos vienen con el territorio. Lo mejor para usted es tomar el aviso en serio, pero también puede ser conveniente intentar determinar si Chkrootkit ha informado un falso positivo.

Chkrootkit utiliza varios medios para encontrar rootkits. Muchas veces, Chkrootkit busca una determinada firma en un archivo basándose en una versión troyana conocida del archivo. Otras veces, Chkrootkit busca puertos abiertos que se sabe que son el resultado de un rootkit u otro ataque. Este fue el caso del informe de infección destacado anteriormente en esta sección. Chkrootkit informó que creía que la computadora estaba infectada con el rootkit bindshell. Basó este hallazgo en dos puertos que encontró abiertos, 1524 y 31337.

En realidad, estos puertos estaban abiertos debido a otra herramienta de seguridad, PortSentry, que escucha en esos puertos con la esperanza de detectar otros hosts infectados. Usé el programa Isof con la opción -i para determinar el programa exacto que estaba escuchando en esos puertos.

Con docenas de rootkits informados por Chkrootkit, probablemente no sepa las ramificaciones exactas de ser infectado por un rootkit determinado. Además, existe una buena posibilidad de que si se ha ejecutado un rootkit, se hayan ejecutado varios rootkits, lo que dificulta mucho más la limpieza. Para comenzar el proceso de control de daños, puede buscar en la web cada rootkit individual para determinar qué acciones toma cuando se ejecuta. Sin embargo, tenga en cuenta que, por definición, después de que un rootkit se haya ejecutado con éxito, el atacante tiene privilegios de root en la computadora y, por lo tanto, puede haber causado un daño mucho mayor al sistema o puede estar en proceso de hacerlo ahora.

Siempre que Chkrootkit informe de una infección, debe tomárselo en serio y asumir siempre lo peor. Prudence sugiere que debe desconectar inmediatamente la computadora de la red y tomar medidas para limpiar el rootkit. En realidad, rara vez es tan fácil o sencillo.

Limitaciones de Chkrootkit y herramientas similares

Chkrootkit es una herramienta poderosa e increíblemente útil, pero no está exenta de limitaciones.

Estas limitaciones no son realmente específicas de Chkrootkit, sino que son una limitación de cualquier herramienta que intente realizar comprobaciones complejas como esta. Una de esas limitaciones, los falsos positivos,

ya ha sido discutido. Otra limitación de Chkrootkit y otras herramientas similares es que se basan, por defecto, en programas incluidos con la propia computadora Linux, programas que pueden haber sido comprometidos o alterados para evitar ser detectados por miradas indiscretas como los de Chkrootkit y utilidades relacionadas.

Aquí hay una lista parcial de programas que usa Chkrootkit; tenga en cuenta que estos programas a su vez pueden depender de bibliotecas u otras cosas en la computadora Linux que también pueden verse comprometidas:

```
ŷ awk
ŷ cortar
ŷ eco
ŷ egrep
ŷ encontrar
ŷ cabeza
ŷ identificación
ŷ ls
ŷ netstat
ŷ PD
ŷ sed
ŷ cuerdas
ŷ nombre
```

Otra limitación de herramientas como Chkrootkit, que comparten herramientas similares, es que solo puede detectar rootkits que se hayan informado y para los que se haya configurado. Algún desafortunado tiene que ser el primero en ejecutar el rootkit en su computadora. Si resulta que eres esa persona, Chkrootkit no te ayudará. Tenga en cuenta, sin embargo, que existe una buena posibilidad de que se ejecuten múltiples rootkits en la computadora, lo que facilitará la detección. Me doy cuenta de que esto es un pequeño consuelo.

Usar Chkrootkit de forma segura

Es una buena idea usar conjuntos conocidos de binarios del sistema cuando se usa una herramienta como Chkrootkit. Muchos rootkits reemplazan binarios vitales del sistema, como `/bin/ps`, con versiones propias. Por lo tanto, si intenta usar `ps` para encontrar procesos desconocidos, es posible que no pueda verlos porque la versión troyana de `ps` los oculta.

Chkrootkit ofrece dos métodos para solucionar este problema. El primer método involucra el uso de un conjunto de binarios en buen estado, probablemente montado desde un CD-ROM. El segundo método implica montar físicamente el disco duro posiblemente comprometido en una computadora diferente y luego ejecutar la verificación desde allí. Este segundo método es más apropiado para el análisis forense después de un ataque exitoso que para investigar un posible ataque.

Montar un CD-ROM con versiones conocidas de archivos binarios es una forma fácil y segura. método para realizar un examen completo usando Chkrootkit. Este método supone que ya tiene un CD-ROM con los archivos binarios correctos en el disco. Para ejecutar Chkrootkit con una copia en CD-ROM de los binarios, primero monte el CD. Esto generalmente se logra usando el comando de montaje , aunque a veces se monta automáticamente.

Aquí se muestra un método común para montar la unidad de CD-ROM en la mayoría de las distribuciones modernas de Linux:

```
montaje -t iso9660 /dev/cdrom /mnt/cdrom
```

Chkrootkit usa la opción -p para definir la ubicación de los archivos binarios que debe usar.

Por lo tanto, si el CD-ROM está montado en /mnt/cdrom, ejecutaría Chkrootkit así:

```
./chkrootkit -p /mnt/cdrom
```

El otro método para ejecutar Chkrootkit es montar físicamente el disco duro posiblemente comprometido en otra computadora y ejecutar Chkrootkit contra el contenido de esa unidad. Esto se logra especificando un directorio "raíz" alternativo para Chkrootkit.

Suponga que la segunda unidad está montada en /mnt/drive2:

```
./chkrootkit -r /mnt/unidad2
```

¿Cuándo se debe ejecutar Chkrootkit?

Chkrootkit debe ejecutarse cuando lo desee. No hay un programa recomendado para Chkrootkit. Personalmente, lo ejecuto a intervalos irregulares por diversión, pero, de nuevo, soy ese tipo de persona. Definitivamente, debe ejecutar Chkrootkit cada vez que observe alguna actividad sospechosa en la computadora o en otras computadoras que puedan interactuar o residir en el mismo bloque de red que la computadora en cuestión. Siempre que ejecute Chkrootkit, siempre debe visitar el sitio web, <http://www.chkrootkit.org/>, para buscar una nueva versión de la herramienta. Es posible que se haya agregado una nueva firma para un rootkit o una funcionalidad adicional desde la versión que está utilizando.

También puede ejecutar Chkrootkit todas las noches desde cron. Sin embargo, no confiaría en que dicho informe sea completamente preciso, pero podría proporcionar una advertencia temprana de una anomalía que requiere su atención. Ejecutar Chkrootkit desde cron podría verse así:

```
0 4 * * * /ruta/a/chkrootkit
```

La entrada cron que se muestra ejecutará Chkrootkit todas las mañanas a las 4:00 a. m. y root (el destinatario de la salida del trabajo cron) recibirá un informe todas las mañanas que detalla la ejecución de Chkrootkit.

Integridad del sistema de archivos

De la mano de un comprobador de rootkits como Chkrootkit va el software de integridad del sistema de archivos. El software de integridad del sistema de archivos supervisa los archivos importantes en la computadora y genera informes basados en los cambios en esos archivos. El administrador puede entonces estar atento a eventos inesperados.

cambios en los archivos en cuestión. Por ejemplo, si archivos como `/etc/resolv.conf` o incluso `/etc/shadow` cambian sin motivo aparente, el administrador puede tomar medidas.

Algunas herramientas populares de integridad del sistema de archivos son OSSEC, Samhain y AIDE. AIDE se cubre en detalle, y se proporciona una descripción más completa de cómo funciona la integridad del sistema de archivos en el Capítulo 14, "Integridad del sistema de archivos".

Supervisión de registros

Los archivos de registro se supervisan para detectar anomalías que puedan indicar un ataque. Aunque este método se usa con éxito, puede generar grandes cantidades de datos y volverse engorroso en redes grandes.

Cuando se combina con otras herramientas, se puede hacer que la supervisión de registros funcione. Por ejemplo, el uso de la supervisión de registros en algunos sistemas clave puede reducir la cantidad de datos que se reciben. Sin embargo, esta y otras medidas similares son realmente medidas provisionales porque hacen poco para garantizar la seguridad de los sistemas que no están monitoreados.

Numerosos paquetes están disponibles para monitorear archivos de registro. Uno de esos paquetes es Swatch. Cubriré brevemente Swatch aquí solo para darle una idea de algunas de las capacidades de este tipo de herramientas.

Muestra de tela

Swatch está disponible con muchas distribuciones de Linux como un paquete complementario o se puede descargar desde <http://swatch.sourceforge.net/>. Swatch es altamente configurable y puede realizar una serie de acciones basadas en una coincidencia.

Swatch funciona en varios modos, incluido un modo llamado paso único, en el que el programa analiza un archivo de registro una vez, busca coincidencias y toma medidas en función de esas coincidencias. En otro modo, Swatch realiza una cola continua (`tail -f`) de un archivo de registro en busca de coincidencias. De manera predeterminada, Swatch monitorea `/var/log/messages`, pero puede configurarse para monitorear cualquier archivo o incluso un socket.

Debido a que Swatch es tan poderoso, no siento que pueda hacerle justicia en un libro sobre Cortafuegos de Linux. Te invito a leer más sobre Swatch. Por ahora, daré una receta para monitorear un archivo de registro con Swatch. Otra receta similar aparece en el Capítulo 13, en la sección sobre Snort titulada "Monitoreo de intrusión automatizado con Snort".

Uso de Swatch para monitorear fallas de inicio de sesión SSH

Hay constantes intentos de inicio de sesión de fuerza bruta contra SSH. Estos por lo general no dan como resultado mucho más que molestia. Sin embargo, generalmente es útil monitorear los archivos de registro para estos y otros intentos de ataque de fuerza bruta a un servidor. Swatch se puede configurar para enviar un correo electrónico (o realizar cualquier otra acción) cuando se registra dicho intento. Esta sección muestra cómo enviar una alerta por correo electrónico cuando se registra una falla de autenticación.

El sistema registra una línea similar a la siguiente cuando se intenta iniciar sesión y falla:

```
7 de junio 17:09:10 ord sshd [3434]: error: \
PAM: Fallo de autenticación para root desde 192.168.1.10
```

Hay una serie de artículos únicos en esta línea, pero optaré por buscar las palabras "Error de autenticación" porque ese es el tipo de cosas sobre las que quiero recibir alertas. La sintaxis de Swatch es dolorosamente fácil pero puede ser increíblemente poderosa. Esto se debe a que Swatch utiliza la sintaxis de expresiones regulares para la coincidencia. El partido en este caso es bastante trivial. Simplemente decirle a Swatch lo que debe vigilar con la directiva de configuración `watchfor` debidamente titulada y luego darle una o más acciones para realizar cuando se observa una coincidencia es todo lo que se requiere para la configuración de Swatch. Por ejemplo, para buscar las palabras "Error de autenticación" y enviar un correo electrónico, la configuración de Swatch consta de lo siguiente:

```
watchfor /Error de autenticación/  
correo
```

Estas dos líneas se guardan en `~/swatchrc`. En este caso, lo hago como `root` porque Swatch necesitará acceso de lectura al archivo de registro en cuestión.

A continuación, inicie Swatch y dígame qué archivo monitorear. Nuevamente, el valor predeterminado es `/var/log/mensajes`. Sin embargo, estoy creando este ejemplo en un sistema Debian, por lo que los errores de autenticación se registran en `/var/log/auth.log` de forma predeterminada. Por lo tanto, apunto Swatch al archivo de configuración correcto y lo inicio:

```
muestra —tail-file=/var/log/auth.log
```

Swatch ahora controlará el archivo de registro en busca de las palabras "Error de autenticación" y envíe un correo electrónico a la raíz si y cuando se encuentran las palabras.

Como se indicó anteriormente, hay varias opciones para las alertas, incluida la ejecución de otros programas. Estos programas pueden ser scripts de shell o cualquier cosa, por lo que las posibilidades son prácticamente ilimitadas.

Cómo no comprometerse

Prácticamente no se puede hacer nada para detener a un atacante con recursos ilimitados y tiempo ilimitado. Desde ataques DoS hasta rootkits y ataques físicos, si alguien quiere sus datos lo suficiente, lo más probable es que pueda acceder a ellos, sin otras limitaciones. Dicho esto, hay muchas cosas que puede hacer para limitar su exposición a la mayoría de los riesgos.

Ni este capítulo ni este libro tratan sobre ataques físicos en ningún nivel. Si un atacante está en el sitio y simplemente puede irse con la computadora o el disco duro que contiene los datos, no hay firewall que ayude. Si el atacante tiene acceso físico a la computadora o dispositivo que contiene los datos, el atacante puede robar los datos por sí mismo o posiblemente plantar su propio software troyano malicioso.

Esta sección brinda algunas sugerencias generales que se prueban en el campo para mantener los sistemas seguros. Estas sugerencias de ninguna manera lo abarcan todo; más bien son simplemente cosas que sugiero para ayudar a garantizar la integridad del sistema.

Asegure a menudo

Proteger el entorno informático es un proceso continuo en lugar de un punto final. A medida que trabaja para proteger los sistemas y las redes, se descubren nuevas vulnerabilidades y

se está desarrollando un nuevo software. Simplemente no existe una varita mágica que le permita estar listo y completo cuando se trata de proteger un entorno informático. Este libro se ha dedicado a proteger una red y sus sistemas mediante el uso de un firewall integrado en Linux. Este capítulo ha presentado algunos de los otros aspectos de un proceso de seguridad en profundidad.

Usando las herramientas disponibles para usted, como las que ya se presentaron en este capítulo, usted puede proteger una computadora y la red en la que reside. Por supuesto, hay pasos adicionales que puede tomar para mejorar aún más la seguridad del medio ambiente.

bastilla linux

Bastille Linux es un programa que ayuda a automatizar el proceso de seguridad del sistema, así como a informar sobre la seguridad del sistema. Bastille Linux implementa muchas de las mejores prácticas de seguridad que podría encontrar al leer volúmenes de material e innumerables sitios web. Todas esas mejores prácticas se implementan a través de una interfaz similar a un asistente (línea de comando o GUI) que contiene mucha información no solo sobre lo que se le pregunta, sino también por qué es importante.

Bastille Linux va tan lejos como para dar recomendaciones para ciertas funciones. A diferencia de muchos herramientas que intentan dar recomendaciones, Bastille lo hace bien al explicar el razonamiento detrás del cambio propuesto, así como las implicaciones que podría tener si elige usar el paso.

Finalmente, Bastille también incluye un proceso de deshacer para que pueda deshacer rápidamente cualquier cambio. eso podría estar causando problemas. Bastille es bienvenido tanto por los administradores de Linux experimentados como por los nuevos en Linux. Algunas distribuciones de Linux incluyen Bastille como paquete. Puede encontrar más información sobre Bastille Linux en <http://bastille-linux.sourceforge.net>.

Actualizar a menudo

Aunque, con mucho, es la tarea más sencilla de todas las de este libro, mantener un sistema informático actualizado es un aspecto de la seguridad del sistema que a menudo se pasa por alto. La mejor manera de asegurarse de que una computadora sea asaltada es dejarla funcionando sin actualizarla.

Una de las mayores fortalezas de Linux y el software de código abierto es la seguridad. Algunas personas intentan argumentar que esta seguridad se logra porque el software de código abierto es menos popular. Por supuesto, esto ignora por completo las estadísticas de participación de mercado, como la encuesta de servidores web de Netcraft que muestra que Apache posee casi el 40% del mercado de servidores web e incluso más si se excluyen los sitios de marcadores de posición que ejecutan Microsoft IIS.

Parte de esta solidez de la seguridad proviene de la capacidad de la comunidad de código abierto para proporcionar soluciones a las pocas horas de la revelación de la vulnerabilidad. Es bastante común que las correcciones estén disponibles el mismo día de la divulgación, incluso para problemas de seguridad que no se divulgaron públicamente anteriormente. Para eventos en los que una solución puede llevar un poco de tiempo, históricamente la comunidad ha sido excelente en proporcionar soluciones alternativas para mitigar y, a veces, eliminar las vulnerabilidades por completo.

Ambas características, arreglos rápidos y soluciones alternativas rápidas, funcionan a su favor. etapa en el mantenimiento de la seguridad del sistema. Sin embargo, para que cualquiera de los dos sea útil, debe

realizar un seguimiento de su disponibilidad mediante el control de listas de correo y sitios web de seguridad. La mayoría de los principales proveedores de Linux ofrecen listas de correo de seguridad solo para anuncios en las que los suscriptores reciben un correo electrónico cada vez que se revela una vulnerabilidad.

Mantener el software actualizado es un aspecto importante de la seguridad del sistema. Recomiendo actualizar con la mayor frecuencia posible, obviamente prestando atención al software que se está actualizando para garantizar que ninguna de las actualizaciones rompa los sistemas en vivo.

Prueba a menudo

No es suficiente asegurar y actualizar con frecuencia, aunque esos dos elementos ciertamente contribuyen en gran medida a garantizar un entorno seguro. Otro punto básico de la seguridad en profundidad es probar a menudo. Las pruebas garantizan que las políticas de seguridad se están aplicando y que la implementación de esas políticas de seguridad es exitosa.

Las pruebas de penetración son otro aspecto importante de la seguridad del sistema. Pruebas de penetración, o pen-testing, es un proceso mediante el cual se prueba la seguridad de un sistema probando una serie de vectores de ataque para conseguir que el sistema se comporte de forma inesperada. La definición de pruebas de penetración es deliberadamente vaga para que no se limite a ataques de una determinada clase o tipo.

Las pruebas de penetración pueden ser tanto informales como formales. Las pruebas de penetración informales son típicas. Los administradores de seguridad o incluso los desarrolladores lo ejecutan de manera independiente utilizando cualquier cosa, desde intentos manuales para ingresar a una aplicación hasta ataques automatizados que utilizan una serie de herramientas. Un tercero realizaría una prueba de penetración formal y probablemente usaría una combinación de ataques manuales y automáticos para probar el sistema. El tipo y la frecuencia de las pruebas de penetración es un asunto de su política de seguridad.

Por supuesto, cuando haces pruebas, es importante probar ambas como si fueras un atacante normal. y como si fueras un iniciado. Probar como un atacante normal significa probar la aplicación o el sistema sin ningún otro conocimiento que no sea el que se puede obtener desde fuera del sistema. En otras palabras, si está probando una aplicación web, vea el código fuente de la página web para ver qué parámetros se están utilizando. Muchas veces, probar como un atacante normal también significa que tendrá que probar desde una ubicación externa a la red local. Esto es especialmente importante cuando se prueba un conjunto de reglas de firewall.

Esta sección examina algunas de las herramientas que puede utilizar para probar una red y un sistema informático. Al igual que con otras listas presentadas en este capítulo, no pretende ser exhaustiva ni exhaustiva. Más bien, las herramientas examinadas aquí brindan un buen punto de partida sobre el cual puede desarrollar su conocimiento de los conceptos e instalaciones de seguridad y pruebas de penetración.

Nmap

Nmap, Network Mapper, es un programa utilizado para identificar puertos abiertos y dispositivos disponibles en una red. El analista de intrusiones utiliza con frecuencia Nmap para determinar qué puertos están abiertos y escuchando en un host determinado. En el contexto de un firewall, Nmap se puede usar desde una ubicación externa para probar las reglas del firewall y garantizar que no haya puertos abiertos y disponibles inesperados.

Nmap está disponible como paquete en muchas distribuciones populares de Linux. Si Nmap no es disponible en su distribución, se puede descargar desde <http://www.nmap.org/>.

Nmap incluye muchas opciones para sondear hosts y redes completas. Estas opciones son demasiado numerosos para cubrirlos en profundidad aquí. En la práctica, he encontrado que la siguiente sintaxis es más útil para realizar el escaneo de puertos antes mencionado, este busca puertos TCP:

```
nmap -sS -v <host>
```

Por ejemplo, para escanear el host 192.168.1.10 en busca de puertos TCP abiertos, se usaría la siguiente sintaxis:

```
nmap -sS -v 192.168.1.10
```

Tenga en cuenta que el uso de la opción `-v` permite verbosidad adicional. Aunque esta opción no es obligatoria, se recomienda, e incluso puede agregar instancias adicionales de `-v` para aumentar la verbosidad.

Varios tipos de escaneos TCP están disponibles con Nmap. Elegí un escaneo SYN porque descubrí que, en general, es el más confiable para este tipo de prueba.

Cuando Nmap comienza un escaneo, envía un ping inicial o una solicitud de eco ICMP al host de destino. A veces, el objetivo no responde a la solicitud de eco ICMP. En estos casos, puede deshabilitar la solicitud de eco ICMP inicial enviada por Nmap utilizando la opción `-P0`.

Como se indicó anteriormente, hay varias opciones disponibles con Nmap. Al escribir simplemente `nmap` en la línea de comando, se imprimirá un conjunto relativamente detallado de instrucciones de uso que contienen muchas de estas opciones.

hping3

hping3 es otra utilidad de red que se puede usar para probar puertos abiertos y también para probar el comportamiento de las aplicaciones y dispositivos de red. hping3 permite al usuario establecer numerosos atributos de un paquete de red, o crear el paquete como a veces se le llama. Cuando se elaboran paquetes, se puede observar el comportamiento de la aplicación o dispositivo de red.

hping3 se usa en el Capítulo 13 para mostrar cómo se verían algunos ataques cuando se ven con TCPDump.

Nadie

Nikto es un programa para probar un servidor web en busca de vulnerabilidades conocidas y también para proporcionar información sobre ese servidor web. Nikto se puede descargar desde <http://www.cirt.net/Nikto2>.

Debido a que Nikto es específico del servidor web, su cobertura será limitada aquí. Sin embargo, si usted está ejecutando un servidor web, recomiendo encarecidamente a Nikto para probar el servidor en busca de una serie de vulnerabilidades.

Resumen

Este capítulo proporcionó una mirada a las herramientas de detección de intrusos y algunos principios básicos de seguridad. Desde cosas como TCPDump, hasta la ubicación del sniffer y la integridad del sistema de archivos, el capítulo le mostró el mundo de la detección de intrusos.

Estas herramientas de detección de intrusos son mejores cuando se combinan con prácticas de seguridad como actualizaciones periódicas, medidas de seguridad mejoradas y pruebas de penetración para garantizar que la seguridad del sistema sea la esperada.

El próximo capítulo del libro profundiza en la seguridad de la red al examinar

TCPDump, una herramienta clave en la caja de herramientas de cualquier administrador.

Esta página se dejó en blanco intencionalmente

Supervisión de red y detección de ataques

Esta El capítulo usa el conocimiento que ha adquirido a lo largo del libro y en los dos capítulos anteriores específicamente para mostrar cómo puede usar algunas de las herramientas para el monitoreo diario y también para la investigación.

El capítulo comienza con una descripción general de la supervisión de redes o rastreo. La información al comienzo de este capítulo se basa en lo que ya ha visto en los dos primeros capítulos del libro. Luego, este capítulo continúa con una mirada a TCPDump, una herramienta clave en el conjunto de herramientas del analista de seguridad de red. Finalmente, el capítulo también analiza dos útiles paquetes de software de seguridad: Snort y ARPWatch.

Escuchando el éter

Armado con el conocimiento básico de algunos de los protocolos centrales de los primeros dos capítulos, está listo para comenzar a escuchar la red. Exactamente lo que puede ver cuando comience a monitorear su red dependerá de varios factores, uno de los cuales es la topología de la red.

Una red Ethernet moderna es una colección de dispositivos de punto final, como computadoras con interfaces de red, interconectados mediante un concentrador o conmutador. La diferencia entre un concentrador y un conmutador es importante tanto para el rendimiento como para la seguridad de la red. En un entorno de concentrador, cada trama de Ethernet se copia en cada puerto del concentrador y, por lo tanto, todos los dispositivos están conectados al concentrador. Compare un entorno concentrador con un entorno conmutado. En un entorno conmutado, el conmutador envía tramas al puerto específico al que está conectado un dispositivo determinado. En otras palabras, con un conmutador, el tráfico se dirige solo a los dispositivos que deberían recibirlo. Si un intruso puede monitorear la red en un entorno concentrador, el intruso verá todos los marcos destinados a todos los dispositivos conectados a ese concentrador. En un entorno de conmutador, el intruso solo verá el tráfico destinado a ese host o el tráfico de difusión que se copia en todos los puertos.

La mayoría de los conmutadores administrados permiten al administrador configurar un determinado puerto para recibir todo el tráfico. Cisco lo llama puerto "span", mientras que otros lo llaman puerto "espejo". En efecto, al copiar todo el tráfico a un puerto en el switch, el administrador puede monitorear todo el

tráfico para que ese conmutador busque posibles intrusiones u otras anomalías. Por supuesto, esto también puede ser peligroso. Si un atacante obtiene el control del dispositivo al final de ese puerto, ¡el atacante también puede escuchar todo! Además, en entornos de mucho tráfico, es probable que se produzca una degradación del rendimiento si intenta supervisar todos los puertos. Por lo tanto, es importante elegir dónde monitorear su red.

Si no tiene un conmutador administrado o un conmutador que le permita copiar todo el tráfico a un puerto, deberá encontrar otro medio para escuchar el tráfico. No recomiendo quitar el interruptor a favor de un concentrador. Sin embargo, un método sería conectar un concentrador al firewall y luego conectar su computadora de monitoreo o detección de intrusos a ese concentrador también, y finalmente conectar el concentrador al interruptor principal. De esta forma, puede monitorear el tráfico interno del firewall sin (mucho) degradación del rendimiento y sin comprometer gran parte de la seguridad que brinda un conmutador.

Mientras escribía la oración sobre la seguridad de un conmutador, recordé algunos tipos de ataques que permiten a un atacante escuchar otro tráfico en un conmutador, incluso si no estaba destinado al puerto donde reside el atacante. Estos ataques, principalmente la suplantación de ARP, implican interferir con el funcionamiento normal de ARP. Se puede encontrar una buena introducción a la suplantación de ARP en el artículo "An Introduction to Arp Spoofing", disponible en línea en http://packagestormsecurity.org/papers/protocols/intro_to_arp_spoofing.pdf.

Elegir puntos de monitoreo dentro de una red es más un arte que una ciencia y es inevitablemente discutible. Hay quienes dicen que solo es importante monitorear el interior de la red porque el firewall evitará que el tráfico externo sea importante de todos modos. Hay otros que sostienen que los puntos externos deben monitorearse para que pueda ver lo que se intenta en la red. Y hay quienes, como yo, creen que tanto los puntos internos como los externos deben ser monitoreados. El monitoreo de la red interna es importante por (espero) razones obvias. Puede buscar tráfico anómalo y también monitorear condiciones y rendimiento inesperados. Sin embargo, creo que monitorear la red externa también es importante. Forjé mis dientes de seguridad informática en un proveedor de Internet donde todo lo importante estaba en la red externa por naturaleza.

Por lo tanto, pude ver lo valioso que es saber lo que sucede en el exterior como un medio para evitar que los ataques tengan éxito.

Tienes que tomar decisiones que funcionen en tu entorno. Puede que no tenga sentido implementar una computadora fuera de su firewall solo para la detección de intrusos. Toda la seguridad es una compensación entre los activos que intenta proteger y los recursos limitados disponibles para protegerlos.

Tres herramientas valiosas

Existe un número cada vez mayor de herramientas y software para monitorear el tráfico de red. Algunas de estas herramientas son gratuitas (como el precio y la voz) y otras cuestan bastante dinero. He usado tanto las herramientas costosas como las gratuitas, y confío en decir que las gratuitas son mejores. Las herramientas costosas son débiles en funcionalidad pero fuertes en belleza. Las interfaces de muchos de los productos brindan una apariencia agradable (aunque muchas de ellas parecen algo inestables). En general, las herramientas de código abierto son un poco más complicadas

para configurar y usar, pero brindan una mejor funcionalidad y con un poco de trabajo pueden producir algunos de los gráficos más atractivos y otras imágenes que brindan las costosas herramientas. Por mi dinero, prefiero tener herramientas de detección de intrusos que pueda usar rápida y fácilmente cuando investigo un posible ataque. Tratar con GUI engorrosas y no intuitivas solo interfiere en el negocio de la detección de intrusos.

Esta sección analiza algunas herramientas de monitoreo con especial énfasis en las herramientas que son cubierto más adelante en el libro.

volcado TCP

Una de las principales herramientas en el conjunto de herramientas de un analista de detección de intrusos debería ser TCPDump. TCPDump coloca una interfaz de red en modo promiscuo para que capture todos los paquetes que llegan. Por supuesto, esto significa que TCPDump debe ejecutarse desde la computadora que experimenta la posible intrusión o debe ejecutarse desde una computadora que es el destinatario de un puerto "distribuido" en un entorno de conmutador. TCPDump se examina con mayor detalle en la siguiente sección.

Bufido

Snort es uno de los mejores sistemas de detección de intrusos disponibles, gratis o no. Snort captura el tráfico de red de la misma manera que lo hace TCPDump. Sin embargo, Snort utiliza una base de datos de firmas de ataques conocidas para proporcionar también un nivel de detección. Mientras que TCPDump es más un monitor manual, Snort está más automatizado en la medida en que el analista no necesita examinar manualmente cada paquete. Puede obtener más información sobre Snort en <http://www.snort.org/>.

Reloj ARP

ARPWatch es una herramienta utilizada para monitorear el tráfico ARP en una red. El objetivo sería que un administrador detecte posibles intentos de suplantación de ARP, así como dispositivos desconocidos que hayan ingresado a la red. ARPWatch se puede descargar desde <http://ee.lbl.gov/>. Al igual que otras herramientas, ARPWatch debe compilarse antes de su uso si su sistema no lo tiene disponible como paquete. ARPWatch se examina más adelante en este capítulo, en la sección "Monitoreo con ARPWatch".

TCPDump: una descripción general simple

Recuerda lo que has leído en capítulos anteriores. Aprendió sobre el direccionamiento IP, las subredes y los encabezados de algunos de esos protocolos centrales. En este capítulo se examinará la herramienta TCPDump y verá algunos de esos protocolos de cerca y personalmente. Armado con una comprensión de cómo monitorear su red a este nivel, puede estar seguro de que podrá solucionar una amplia gama de problemas, no solo aquellos relacionados con la seguridad informática.

Una herramienta importante en el conjunto de herramientas del analista de intrusiones es TCPDump. En un nivel básico, TCPDump es un software de análisis y captura de paquetes en tiempo real. Esto significa que TCPDump se puede utilizar para espiar la comunicación de la red a medida que viaja a través de la red.

Sin embargo, como ya se ha mencionado, la topología de la red dicta la cantidad de tráfico que se puede espiar. Si la computadora desde la que se ejecuta TCPDump está conectada a una red conmutada, TCPDump solo verá el tráfico destinado a ese host o tráfico de difusión/multidifusión. Un buen enfoque en una red conmutada sería utilizar un puerto "span" en el que el propio conmutador copiará todo el tráfico de la red. Por supuesto, nada de esto es motivo de preocupación en una red basada en concentrador porque todo el tráfico se copia en todos los puertos del concentrador.

TCPDump coloca la interfaz de red en modo promiscuo. Antes de que se emocione demasiado, tenga en cuenta que en las interfaces ocupadas esto significa que una gran cantidad de tráfico pasará volando por la pantalla, lo que tiene el potencial de ralentizar el tráfico ligeramente. En cualquier caso, una gran cantidad de tráfico será demasiado para que un ser humano lo comprenda, por lo que querrá capturar la salida en un archivo, canalizar la salida a un buscpersonas o filtrar el tráfico para buscar algo específico. Filtrar a través de una expresión TCPDump es, con mucho, la mejor opción, pero las opciones no son excluyentes entre sí. Usualmente uso un filtro y un buscpersonas como menos, en caso de que algo interesante pase volando por mi pantalla demasiado rápido.

TCPDump puede filtrar el tráfico por prácticamente cualquier criterio que pueda imaginar. Lo más común para el analista de intrusiones es observar el tráfico por protocolo, host, número de puerto o una combinación de los mismos. Antes de continuar, sería negligente si no recomendara leer o al menos consultar la página del manual TCPDump(1) (escriba `man tcpdump` para leerlo). La página de manual es un documento completo que proporciona no solo sintaxis sino también ejemplos de uso, así como algunos diagramas de protocolo. Si se queda atascado tratando de usar TCPDump y no tiene una copia de este libro a mano, tal vez debería comprar dos copias del libro. También puede utilizar la página de manual de TCPDump como referencia.

Obtención e instalación de TCPDump

TCPDump se puede descargar desde <http://www.tcpdump.org/>. TCPDump requiere la biblioteca libpcap de PCap, por lo que mientras descarga TCPDump, también debe descargar libpcap. Las distribuciones de Linux más populares también incluyen TCPDump como un paquete disponible. Por ejemplo, si está usando Debian, simplemente puede escribir esto:

```
apt-get install tcpdump
```

El sistema de mantenimiento de paquetes instalará TCPDump y también cualquier requisito previo. Para todos los demás, probablemente pueda buscar un paquete en el repositorio de su distribución o simplemente descargar el código fuente y compilarlo, lo cual recomiendo. Si intenta compilar TCPDump sin tener instalado libpcap, verá un error similar al siguiente mientras ejecuta el script de configuración para TCPDump:

```
buscando main en -lpcap... no
```

```
configure: error: consulte el documento INSTALL para obtener más información
```

La instalación de libpcap y TCPDump es bastante sencilla en lo que respecta a la compilación de software. Desarchive cada fragmento de código fuente, ejecute el script de configuración, compile e instale.

En esencia:

```
tar -zxvf libpcap-<versión>.tar.gz
cd libpcap-<versión>
./configurar
hacer
hacer instalar
```

Haga lo mismo para TCPDump:

```
tar -zxvf tcpdump-<versión>.tar.gz
cd tcpdump-<versión>
./configurar
hacer
hacer instalar
```

Opciones de volcado TCP

TCPDump acepta una amplia gama de opciones de línea de comandos que modifican su comportamiento, la cantidad de datos capturados y la forma en que se capturan los datos. Una gama tan amplia de opciones significa que tiene el poder de cambiar significativamente la forma en que opera el programa.

Para TCPDump, encontrará que con frecuencia usa un conjunto común de opciones para la mayoría de las actividades de captura de datos, y es posible que no use otras en absoluto.

Algunas de las opciones más utilizadas incluyen las que se enumeran en la Tabla 13.1.

Examinar cada una de estas opciones a su vez revela los pasos necesarios para realizar la captura y el análisis de paquetes básicos. No todas estas opciones son necesariamente necesarias para capturar tráfico con TCPDump (de hecho, ninguna de ellas es necesaria). Es perfectamente válido simplemente escribir el comando `tcpdump` en la línea de comando para comenzar a capturar el tráfico. Sin embargo, en la práctica, muchas de estas opciones son necesarias para obtener el nivel de detalle necesario para analizar correctamente el tráfico.

Tabla 13.1 Algunas opciones comunes para TCPDump

Opción	Descripción
-i <interfaz>	Especifica la interfaz a usar
-v	Produce salida en modo detallado
-vv	Produce salida en modo muy detallado
-x	Hace que TCPDump imprima el paquete en formato hexadecimal
-X	Hace que TCPDump también imprima la salida en ASCII
-norte	Le dice a TCPDump que no realice búsquedas de DNS para las direcciones IP vistas durante la captura
-F <archivo>	Lee la expresión de <archivo>
-D	Imprime las interfaces disponibles
-s <longitud>	Establece la longitud de cada paquete de la captura en <longitud>

268 Capítulo 13 Supervisión de red y detección de ataques

La opción `-i <interfaz>` cambia la interfaz predeterminada en la que TCPDump escuchará los paquetes para capturar. De forma predeterminada, TCPDump escuchará en la primera interfaz, `eth0`.

Sin embargo, para máquinas multitarjeta puede ser necesario usar esta opción para que se capture el tráfico correcto. Por ejemplo, en un firewall, la interfaz `eth0` podría estar conectada a la red interna mientras que la interfaz `eth1` está conectada a Internet. Puede que le interese ver el tráfico que llega a su interfaz externa (`eth1`); por lo tanto, usaría la opción `-i <interfaz>` en TCPDump.

Las opciones de modo detallado, `-v`, `-vv` y `-vvv` (no incluidas en la Tabla 13.1), hacen que TCPDump imprima más (y más, y más) información sobre cada paquete recibido.

Con `-v` esta información incluye cosas tan importantes como el TTL, la ID del paquete, la longitud y las opciones. Por lo general, es necesario experimentar durante una captura de paquetes para determinar cuál de estas opciones se adaptará a sus necesidades. Es posible que los diferentes protocolos no tengan mucha (o ninguna) información adicional para imprimir, por lo que agregar verbosidad con estos conmutadores no servirá de nada.

La opción `-x` hace que TCPDump también imprima volcados hexadecimales de cada paquete. para mis ojos, esta opción no es particularmente útil porque no leo muy bien el hexágono. Sin embargo, se requiere el uso de la `-x` minúscula para aprovechar el volcado ASCII de los paquetes que se pueden obtener mediante el uso de la `-X` mayúscula. Por lo tanto, rara vez usaré solo `-x` y en su lugar usaré tanto `-x` como `-X`. Aunque algunas partes del paquete pueden imprimirse usando solo `-X`, usar ambos puede ser útil.

Una opción a veces útil de las opciones más comunes es `-s <longitud>` opción. El uso de esta opción es útil para imprimir el contenido de los paquetes en lugar de solo los 68 bytes predeterminados. Si solo está interesado en los encabezados de los paquetes, esta opción no será de mucho uso, si es que lo tiene. Sin embargo, si desea echar un vistazo dentro del paquete, esta opción le ayudará a garantizar que la captura del paquete no se trunque.

Una opción que se vuelve más útil cuanto más usa TCPDump es `-F <archivo>` opción. Esta opción le dice a TCPDump que lea el contenido de `<archivo>` para la expresión de filtro en lugar de leer la línea de comando. Esta opción es muy útil para expresiones más largas o expresiones que se usan con frecuencia (o incluso con poca frecuencia). Después de usar TCPDump por un tiempo, es posible que se canse de escribir la misma expresión de filtro anterior para capturar los mismos paquetes semana tras semana. Almacenar esa expresión en un archivo y luego leer la expresión del archivo cuando se usa TCPDump es una excelente manera de ahorrar tiempo.

Cuando recién comienza con TCPDump, una opción que puede resultarle útil es `-D` opción. La opción `-D` le informa a TCPDump que imprima una lista de interfaces en las que puede realizar la captura de paquetes. Debido a que las capturas de paquetes dependen de la interfaz, saber qué interfaz usar es lo más importante que tendrá que elegir. En Linux, es un poco más fácil elegir la interfaz correcta porque los nombres de las interfaces suelen ser simples, como `eth0` para la primera tarjeta Ethernet. Sin embargo, en Windows, `-D` es mucho más importante porque los nombres de las interfaces pueden ser bastante difíciles de recordar.

Una opción final que vale la pena mencionar es la opción `-n`. El uso de `-n` le dice a TCPDump que no realice búsquedas DNS inversas en los hosts tal como los ve durante la captura. Hacer búsquedas inversas con frecuencia ralentiza la captura de paquetes y, naturalmente, también aumenta la cantidad de tráfico. Por lo tanto, agregar `-n` es útil para acelerar la captura y reducir el

relación señal-ruido. Cuando me olvido de configurar la opción `-n`, a veces me pregunto: "¿Por qué esta máquina realiza búsquedas de DNS?" solo para darme cuenta de que las búsquedas son el resultado de mi actividad de captura de paquetes.

Expresiones de volcado TCP

Ahora comienza la diversión. De forma predeterminada, TCPDump capturará y generará todos los paquetes que lleguen a la interfaz. A veces, esto es útil para escuchar rápidamente algo de tráfico en una interfaz silenciosa. Sin embargo, la mayoría de las capturas harán uso de expresiones en TCPDump. Una expresión TCPDump es una colección de criterios para el tráfico de red que le gustaría ver con TCPDump. Las expresiones consisten en uno o más calificadores y posiblemente una primitiva, los cuales se analizan en las siguientes subsecciones. Se puede usar una expresión para capturar solo el tráfico que se origina en un determinado host o que está destinado a un determinado host. Las posibilidades con expresiones y combinaciones de expresiones le brindan la capacidad de concentrarse exactamente en los paquetes que necesita ver para evaluar una situación de red determinada.

Una de las características más poderosas de las expresiones es la capacidad de negar. Por ejemplo, si desea escuchar todo el tráfico excepto el tráfico de red en el puerto 80 (generalmente tráfico HTTP), puede hacer que TCPDump capture todo el tráfico excepto el que se transmite o recibe en el puerto 80. TCPDump también puede usar otros términos lógicos, como AND, OR y la palabra clave de negación NOT ya mencionada.

Las expresiones de TCPDump se encierran entre comillas simples (') y se pueden agrupar encerrando entre paréntesis las diversas partes de una expresión dada. Esto significa que puede combinar varias expresiones para capturar solo el tráfico que le interese. La clave para agrupar expresiones es el uso de los términos lógicos AND, OR y NOT. TCPDump tiene tres calificadores, cada uno de los cuales se presenta a su vez en la discusión que sigue. El primer tipo de calificador es el calificador de tipo.

Calificador de tipo de TCPDump

Así como TCPDump tiene tres tipos de calificadores, el calificador de tipo en sí contiene algunas variaciones, incluidos `host`, `port`, `portrange` y `net`. El calificador de `host` se utiliza para especificar el host o el destino del tráfico interesante. El calificador de tipo de puerto no se usa sorprendentemente para especificar el puerto en el que capturar paquetes, y `portrange` especifica una serie de puertos, como 5060-5080. El tipo de `red` se usa para especificar la subred para el tráfico interesante. Podría usar el calificador de red en una expresión para escuchar el tráfico en un rango completo de direcciones.

Por supuesto, hay momentos en los que no desea escuchar un rango completo de direcciones. TCPDump también acepta la máscara modificadora con el calificador de red para especificar la máscara de subred. También puede usar la notación CIDR para especificar los bits de máscara.

Antes de continuar, aquí hay un ejemplo de una expresión TCPDump para capturar el tráfico en puerto 80:

```
tcpdump 'puerto 80'
```

Debido a que esta expresión usa solo un único criterio (puerto 80), no es necesario encerrarla entre paréntesis. Sin embargo, si el objetivo era capturar el tráfico en el puerto 80 con una fuente

270 Capítulo 13 Supervisión de red y detección de ataques

o el destino de uno o más hosts específicos, digamos 192.168.1.10 y 192.168.1.11, entonces se requerirán paréntesis, como en este ejemplo:

```
tcpdump 'puerto 80 y (host 192.168.1.10 o host 192.168.1.11)'
```

Los paréntesis son necesarios solo para la agrupación lógica. En la práctica, no sufrirá ninguna penalización por usarlos, y la verdad es que normalmente los uso por costumbre. Al escribir el ejemplo anterior del puerto 80 simple, incluí los paréntesis al principio, solo para regresar y eliminarlos después de pensar en lo que estaba haciendo. Los viejos hábitos tardan en morir. Hablando de términos innecesarios, tampoco se requiere el término host en el ejemplo, más sobre eso más adelante.

Aquí hay algunos ejemplos que usan el calificador de tipo de red para escuchar el tráfico en ambas direcciones para la red listada:

```
tcpdump 'red 192.168.1'
```

Aquí está el mismo ejemplo usando la notación CIDR:

```
tcpdump 'red 192.168.1.0/24'
```

Y finalmente, aquí está el mismo ejemplo usando el modificador de máscara :

```
tcpdump 'net 192.168.1.0 máscara 255.255.255.0'
```

Si no especifica un calificador de tipo (host, net, port, portrange) dentro de una expresión TCPCDump, se asume el tipo de host. Por lo tanto, no se sorprenda cuando reciba un "error de análisis" al intentar algo como esto:

```
volcado tcp '80'
```

Realmente, probablemente quería que TCPCDump escuchara el tráfico en el puerto 80:

```
tcpdump 'puerto 80'
```

Calificador de dirección de TCPCDump

Otro tipo de calificador dentro de una expresión TCPCDump es el calificador de dirección. Los ejemplos anteriores buscarán tráfico que fluya en cualquier dirección, yendo o viniendo, en el puerto 80, por ejemplo. Por ejemplo, esto podría significar que se capturará el tráfico destinado a un servidor web que se ejecuta en 192.168.1.10, pero también el tráfico que sale de la computadora en 192.168.1.10 y se destina a otro servidor en el puerto 80. También puede especificar la dirección con que capturar el tráfico utilizando un calificador de dirección. Los términos src para el origen y dst para el destino son los dos calificadores de dirección utilizados por TCPCDump.

Agregar el término de destino a uno de los ejemplos anteriores produce una expresión que buscará el tráfico del puerto 80 que va a 192.168.1.10 o 192.168.1.11:

```
tcpdump 'puerto 80 y (src 192.168.1.10 o src 192.168.1.11)'
```

El calificador de dirección no se limita a buscar tráfico en determinadas direcciones. Es perfectamente válido buscar tráfico con origen o destino de un puerto específico, como en este ejemplo que busca tráfico con destino del puerto 25 (generalmente SMTP):

```
tcpdump 'dst puerto 25'
```

Hay calificadores de dirección que son específicos para el tráfico de capa de enlace inalámbrico 802.11. Estas incluyen ra, ta, addr1, addr2, addr3 y addr4. Además, algunos protocolos utilizan los términos entrante y saliente para especificar la dirección. Consulte la página del manual de TCPDump para obtener más detalles sobre estos calificadores.

Calificador de protocolo de TCPDump

Un último tipo de calificador para usar en una expresión TCPDump es el calificador de protocolo. No es sorprendente que los calificadores de protocolo le permitan elegir qué protocolos se deben capturar con TCPDump. Los protocolos que se pueden capturar con TCPDump incluyen, entre otros, Ethernet (éter abreviado para la sintaxis de TCPDump), WLAN, TCP, UDP, ICMP, IP, IPv6 (ip6 abreviado para la sintaxis de TCPDump), ARP, ARP inverso (abreviado como rarp) y más.

Primitivos

Además de los tres calificadores principales (tipo, dirección y protocolo), también existen lo que se conoce como primitivos para usar en expresiones TCPDump. Las primitivas son palabras clave que ayudan a especificar parámetros adicionales para la captura de paquetes. Algunos aspectos destacados de las primitivas de uso común incluyen

- Operadores aritméticos
- transmisión
- puerta de enlace
- mayor
- menos

Los operadores aritméticos incluyen +, -, *, /, >, <, >=, <=, != y algunos otros.

TCPDump puede usar operadores aritméticos bastante complejos y compensaciones de paquetes para buscar paquetes. Prefiero dejarlo como ejercicio para que el lector se sumerja en estas áreas en caso de que lo considere necesario.

La primitiva de transmisión , cuando se le antepone ip o ether, buscará paquetes que sean transmisiones de IP o Ethernet respectivamente, aunque ether es el tipo predeterminado que se debe buscar. Por ejemplo, una expresión de TCPDump que busca transmisiones IP buscará transmisiones en una red IP. Sin embargo, si la tarjeta de interfaz en la que está escuchando TCPDump no tiene máscara de subred o si se está utilizando cualquier interfaz, esta primitiva de difusión no funcionará.

Las primitivas mayor y menor se utilizan para buscar paquetes con una longitud mayor o igual o menor o igual a la longitud dada. Estas primitivas son funcionalmente equivalentes a utilizar los operadores aritméticos para las mismas. Así, por ejemplo, la sintaxis

```
largo>=1500
```

es equivalente a esto:

```
mayor 1500
```

Más allá de lo básico con TCPDump

Ahora debería tener una idea de la sintaxis básica de TCPDump, incluidas algunas de las opciones, la sintaxis y las expresiones de TCPDump. La cantidad de solución de problemas y diagnóstico que se puede lograr incluso con un conocimiento básico de la sintaxis de TCPDump lo convierte en una herramienta esencial para cualquiera que administre computadoras en red. Sin embargo, para examinar problemas más difíciles, es posible que necesite ir más allá de los conceptos básicos de TCPDump.

Ir más allá de los conceptos básicos de TCPDump requiere una comprensión más profunda de los propios protocolos. Conocer las banderas de TCP o los tipos de ICMP puede ayudar a reducir el enfoque a solo los paquetes de interés. Aunque esta información y saber cómo usarla con TCPDump no es obligatoria, tener la capacidad de consultar la información en cualquier momento es valioso, por decir lo menos. Tómese el tiempo para familiarizarse con la sintaxis más compleja de TCPDump. Solo cuesta tiempo probar una expresión de filtrado de paquetes para ver cómo funciona en diversas condiciones de red.

Uso de TCPDump para capturar protocolos específicos

En esta sección, daré algunos ejemplos que le mostrarán cómo capturar varias formas de tráfico de red con fines de monitoreo. Incluido entre los ejemplos, verá cómo se ve una consulta de DNS a través de TCPDump, algunos ejemplos de ICMP (ping) y varios protocolos basados en TCP y UDP. Después de que vea cómo se ve el tráfico normal, le mostraré algunas de las cosas divertidas. Específicamente, mostraré cómo se ven algunos tipos de ataques a través de TCPDump para que pueda detectarlos rápidamente cuando ingresan (o salen) de su red.

A lo largo de esta sección, usaré algunos programas diferentes para generar tráfico para que lo capture TCPDump. Mi herramienta principal para las capturas relacionadas con TCP será telnet. Usaré telnet para generar tráfico y reflejar lo que hace el protocolo real (o algo parecido) en el mundo real. La generación de consultas de DNS se realizará mediante el comando `dig` y el comando `host`. Se utilizarán los comandos `ping` y `traceroute`. Finalmente, se utilizará el comando `hping3` para generar tráfico ICMP y otros paquetes interesantes, especialmente en la sección de ataque. Con la excepción de `hping3`, todos estos programas están instalados en la mayoría de las principales distribuciones de Linux.

Uso de TCPDump en el mundo real

Hasta ahora, en este capítulo, ha visto varios ejemplos del uso de TCPDump para capturar varios tipos de tráfico. Estos ejemplos se dieron para mostrar el uso de TCPDump en relación con expresiones y otras opciones. Ahora es el momento de brindarle ejemplos de la vida real del uso de TCPDump para capturar tipos específicos de tráfico. Las situaciones en las que puede usar estos ejemplos variarán, pero intentaré dar alguna pista de por qué podría usar un ejemplo dado, donde pueda. Puede ser útil ver cómo se crea una expresión de filtro cuando se intenta capturar en el mundo real. Toqué brevemente este tema antes. Sin embargo, antes de brindar soluciones tipo receta, le mostraré cómo crear un filtro con el objetivo específico de capturar una conversación HTTP.

Creación de un filtro para capturar una conversación HTTP

HTTP es el lenguaje de la Web. Por lo general, HTTP se desplaza sobre TCP, que a su vez se desplaza sobre IP. Elijo HTTP como la primera captura del mundo real solo porque las personas generalmente están familiarizadas con la navegación de una página web, aunque es posible que no estén familiarizadas con el protocolo subyacente.

Recuerde que IP es un protocolo sin conexión, mientras que TCP es un protocolo orientado a la conexión. TCP utiliza un protocolo de enlace de tres vías para iniciar una conversación. HTTP aprovecha la naturaleza orientada a la conexión de TCP y, de hecho, no conoce los protocolos de capas inferiores (recuerde el modelo OSI). En lo que respecta a HTTP, pasa sus datos a la siguiente capa inferior y listo. Con ese fin, cuando se inicia una conversación HTTP, lo primero que debe ver a través de TCPDump es el protocolo de enlace de tres vías de TCP seguido de datos específicos del protocolo.

En su mayor parte, el tráfico HTTP fluye hacia un destino del puerto 80.

Nota

Los puertos en los que normalmente operan varios servicios se pueden encontrar examinando el archivo / etc/services. Con eso en mente, la verdadera fuente para las asignaciones de números de puerto es IANA. Puede ver la lista más actualizada y completa de asignaciones de números de puertos oficiales en la URL <http://www.iana.org/assignments/port-numbers>. Sin embargo, recuerde que no hay nada que impida que alguien ejecute un servicio en un puerto que no sea el número de puerto oficial.

Debido a que HTTP generalmente se encuentra en el puerto 80, sería una buena idea comenzar con un expresión TCPDump básica que busca solo el tráfico del puerto 80 , como esta:

```
tcpdump 'puerto 80'
```

Ejecutar ese comando y luego generar algo de tráfico navegando a una página web produce estos resultados:

```
tcpdump: salida detallada suprimida, use -v o -vv para la decodificación completa del protocolo  
escuchando en eth0, tipo de enlace EN10MB (Ethernet), tamaño de captura 96 bytes
```

```
17:15:38.934337 Cliente IP.braingia.org.4485 > prueba.ejemplo.com.www: \  
S 523004834:523004834(0) ganar 5840 \  
<mss 1460, saco OK, marca de tiempo 249916003 0, nop, wscale 0>  
  
17:15:38.984650 IP prueba.ejemplo.com.www > cliente.braingia.org.4485: S \  
2810959978:2810959978(0) reconocimiento 523004835 victoria 5792 \  
<mss 1460, saco OK, marca de tiempo 1320060704 249916003, nop, wscale 0>  
  
17:15:38.984684 IP cliente.braingia.org.4485 > prueba.ejemplo.com.www: \  
. acuse de recibo 1 victoria 5840 <no, no, marca de tiempo 249916008 1320060704>  
  
17:15:38.985326 IP cliente.braingia.org.4485 > prueba.ejemplo.com.www: \  
P 1: 462 (461) acuse de recibo 1 victoria 5840 <no, no, marca de tiempo 249916008 1320060704>  
  
17:15:39.038067 IP test.example.com.www > client.braingia.org.4485: . \  
acuse de recibo 462 victoria 6432 <no, no, marca de tiempo 1320060710 249916008>
```

274 Capítulo 13 Supervisión de red y detección de ataques

```
17:15:39.065141 IP test.example.com.www > client.braingia.org.4485: . \
1:1449(1448) accuse de recibo 462 victoria 6432 <no, no, marca de tiempo 1320060712 249916008>

17:15:39.065183 IP client.braingia.org.4485 > test.example.com.www: . \
accuse de recibo 1449 victoria 8688 <no, no, marca de tiempo 249916016 1320060712>
```

Nota

He separado los resultados para una mayor legibilidad y continuaré haciéndolo durante el resto de este capítulo.

Observe las dos primeras líneas de la salida TCPDump. En este caso, la primera línea me dice que realmente debería tener una salida más detallada habilitada para ver algo interesante dentro del paquete, y la segunda línea da un estado de la interfaz en la que escucha TCPDump, así como el tamaño de la captura.

La siguiente línea es la primera línea de la captura y, coincidentemente, también es el primer paquete (SYN) enviado en el protocolo de enlace de tres vías de TCP. Lo primero que notará en la línea es una marca de tiempo, seguida del protocolo (IP). El siguiente es el nombre de host de la computadora que inició el paquete (client.braingia.org) junto con el puerto de origen del tráfico (4485). Esta combinación, equipo de origen y puerto de origen, se conoce como Source. El signo mayor que (>) muestra la dirección del flujo en relación con el destino de este tráfico, que, como puede ver en esa línea, es test.example.com.www. El www significa el puerto de destino al que se dirige el tráfico en la computadora de destino.

El siguiente elemento de interés en la línea de salida de TCPDump es la sección Banderas, en este caso indicada por una S mayúscula. Recuerde del Capítulo 1, "Conceptos preliminares subyacentes a los cortafuegos de filtrado de paquetes", que el encabezado TCP puede contener varias banderas para indicar ciertas condiciones para el paquete. Si adivinó que S indica un paquete con el indicador SYN establecido, es posible que haya ganado un premio valioso. Después de la sección Banderas está el espacio del Número de secuencia para el paquete, que indica los números de secuencia que se cubrirán dentro de este paquete. En el caso del ejemplo, el espacio del número de secuencia (523004834:523004834(0)) es de longitud cero. El siguiente elemento en esta línea es el tamaño de la ventana, como lo indica el win 5840 en la salida. Finalmente, entre corchetes están las opciones contenidas en el paquete. Aunque estas opciones pueden ser de interés en ciertos momentos, rara vez necesitarás preocuparte mucho por ellas en el mundo real.

Ahora ha visto un solo paquete de un protocolo de enlace de tres vías TCP a través de TCPDump. No te preocupes, se vuelve más emocionante que esto, de verdad. La siguiente línea de la captura contiene el paquete de respuesta que regresa de test.example.com. Observe que la marca de tiempo ha aumentado y el protocolo sigue siendo IP. Sin embargo, ahora la computadora de origen es test.example.com en el puerto 80 y el destino es client.braingia.org.4485. Observe también que el indicador SYN se establece como lo demuestra la S que sigue al área de origen > destino. Sin embargo, el espacio del número de secuencia ahora es diferente, 2810959978:2810959978(0). Esto se debe a que test.example.com eligió su propio número de secuencia como parte del proceso. La primera diferencia de verdadero interés porque no la hemos visto antes es el ack 523004835. Esta es la segunda parte del protocolo de enlace de tres vías TCP, es decir, lo que comúnmente se conoce como el paquete SYN-ACK. En este paquete, la computadora de destino original responde o reconoce la llamada a

iniciar una conexión TCP en el puerto especificado. Observe que el número que sigue al acuse de recibo es igual al número de secuencia original (523004834) más uno. Este es el protocolo mismo en acción.

El tercer paquete de la captura, representado nuevamente aquí como referencia, es el paquete final en la configuración de la conexión para TCP:

```
17:15:38.984684 IP client.braingia.org.4485 > test.example.com.www: . \
    acuse de recibo 1 victoria 5840 <no, no, marca de tiempo 249916008 1320060704>
```

En este paquete, la fuente original reconoce el establecimiento de la conexión. Observe que hay un solo punto (.) donde normalmente se mostrarían las banderas. Esto generalmente significa que no hay banderas configuradas; sin embargo, algunas banderas como ACK aparecen en un lugar diferente en la línea de salida. El lado de la fuente establece el indicador ACK y también establece un número de secuencia inicial para esta conexión. En este punto, se dice que la conexión TCP está establecida. Seguro que parece mucho trabajo, pero ¿no te prometí que esta sección era sobre HTTP? Bastante seguro. Los siguientes paquetes con salida de captura indican que una conexión HTTP está progresando:

```
17:15:38.985326 IP cliente.braingia.org.4485 > prueba.ejemplo.com.www: \
    P 1: 462 (461) acuse de recibo 1 victoria 5840 <no, no, marca de tiempo 249916008 1320060704>
```

```
17:15:39.038067 IP test.example.com.www > client.braingia.org.4485: . acuse de recibo \
    462 ganar 6432 <no, no, marca de tiempo 1320060710 249916008>
```

```
17:15:39.065141 IP test.example.com.www > client.braingia.org.4485: . \
    1:1449(1448) acuse de recibo 462 victoria 6432 <no, no, marca de tiempo 1320060712 249916008>
```

```
17:15:39.065183 IP client.braingia.org.4485 > test.example.com.www: . \
    acuse de recibo 1449 victoria 8688 <no, no, marca de tiempo 249916016 1320060712>
```

La fuente envía el comienzo de los datos para esta comunicación. Observe que el indicador PUSH se estableció en el paquete inicial y que los números de secuencia aumentan. Los dos lados reconocen los números de secuencia y se transfieren los datos. Pero con el comando TCPDump que ejecuté (tcpdump 'port 80'), no hay mucho más que ver. Por lo tanto, mejoraré ese comando para incluir las opciones que normalmente incluyo para mirar dentro de los paquetes. Dejaré que el lector vea qué hace realmente cada una de las opciones, como se explicó anteriormente en este capítulo. Aquí está el comando mejorado para TCPDump:

```
tcpdump -vv -X -s 1500 'puerto 80'
```

Con este comando en ejecución, puedo generar tráfico web adicional. Aquí hay dos y un medio paquete del resultado, retomando justo después del apretón de manos de tres vías:

```
18:18:51.986230 IP (a 0x0, ttl 64, id 10907, desplazamiento 0, banderas [DF], \
    longitud: 513) cliente.braingia.org.4564 > prueba.ejemplo.com.www: \
    P [tcp sum ok] 1:462(461) ack 1 win 5840 <nop,nop,timestamp
    250295308 1320440053>
    0x0000: 0090 2741 78f0 00e0 1833 2ee8 0800 4500 .. 'Ax....3....E.
    0x0010: 0201 2a9b 4000 4006 044b c0a8 010a 455d .. *. @. @ .. K .... E]
    0x0020: 0302 11d4 0050 0c9b 1ea0 9627 33f8 8018 .....P..... '3...
    0x0030: 16d0 4915 0000 0101 080a 0eeb 340c 4eb4 ..I.....4.N.
    0x0040: 50f5 4745 5420 2f20 4854 5450 2f31 2e30 P.GET./HTTP/1.0
    0x0050: 0d0a 486f 7374 3a20 7777 772e 6272 6169 ..Host:.texto.examen
```

276 Capítulo 13 Supervisión de red y detección de ataques

```
0x0060: 6E67 6961 2E6F 7267 0D0A 4163 6365 7074 PLE.com..Cept 0x0070: 3a20 7465 7874
2f6d 746d 6c2c 2074 6578: .tex / html, .tex 0x0080: 742F 746C 6169 6E2C 2061 7070 6c69
6361 T / Plain, .plica 0x0090: 7469 6F6E 2F6D 7377 6F72 642C 2061 7070 CION /
MSWORD, .App 0x00A0: 6C69 6361 7469 6F6E 2F70 6466 2C20 6170 Licencia / PDF, .Ap
0x00B0: 706C 6963 6174 696F 6E2F 6F63 7465 742D Plicating / Octet- 0x00C0: 7374 7265
616d 2c20 6170 706c 6963 6174 Stream, .applicat 0x00d0: 696F 6E2F 782D 7472 6F66 662D
6D61 6E2C ION / X-TROFF-HOMBRE, 0x00E0: 2061 7469 6C69 6361 7469 6F6E 2F78
2D74 .Application / XT 0x00f0: 6172 2c20 6170 7172 6963 6174 696F 6E2F AR, .Application /
0x0100: 782D 6774 6172 2c20 6170 717c 6963 6174 X-Gtar, .applicat 0x0110: 696F 6E2F 7274
662C 2061 7070 6c69 6361 Ion / RTF, .mplicka 0x0120: 7469 6F6E 2F70 6F73 7473 6372 6970
742C /posdata, 0x0130: 2061 7070 6c69 6361 7469 6f6e 2f67 686f .application/gho 0x0140:
7374 7669 6577 2c20 7465 7874 2f2a 0d0a stview,.text/*.. 0x01650 36 3 70 6 6170 706C 6963
6174 Acceptar: .applicat 0x0160: 696F 6E2F 782D 6465 6269 616E 2D70 6163 ION / X-Debian-
Pac 0x0170: 6B61 6765 2C20 6175 6469 6F2F 6175 6469 6F2F 6261 7369 Kage, .Audio / Basi
0x0180: 632C 202A 2F2A 3B71 3D30 2E30 310D 0a41 c.,*/*;q=0.01..A 0x0190: 6363 6570 742d
456e 636f 6469 6e67 3a20 ccept-Codificación:.. 0x01A0: 677A 6970 2C20 636F 6D70 7265 7373
0D0A GZIP, .Compress .. 0x01b0: 4163 6365 7074 2D4c 616E 6775 6167 653A Idioma: 0x01C0:
2065 6E0D 0A55 7365 722D 4167 656E 743A.en..user-agente: 0x01d0: 204c 796e 782f 322e
382e 3472 656c 2e31 .Lynx/2.8.4rel.1 0x01e0: 206c 6962 7777 772d 464d 2f32 2e31 3420 .libwww-
FM/2.14. 0x01f0: 5353 4c2d 4d4d 2f31 2e34 2e31 204f 7065 SSL-MM/1.4.1.Ope 0x0200: 6e53
534c 2f30 2e39 2e36 630d 0a0d 0a nSSL/0.9.6c....
```

18:18:52.039595 IP (hasta 0x0, ttl 48, id 25346, compensación 0, indicadores [DF], \ longitud: 52)

test.example.com.www > client.braingia.org.4564: .

[tcp sum ok] 1:1(0) ack 462 win 6432 <nop,nop,timestamp

1320440059 250295308>

0x0000: 00e0 1833 2ee8 0090 2741 78f0 0800 4500 ...3....'Ax...E. 0x0010: 0034 6302 4000 3006

ddb0 455d 0302 c0a8 .4c.@.0...E]... 0x0020: 010a 0050 11d4 9627 33f8 0c9b 206d

8010 ...P...3....m. 0x0030: 1920 6799 0000 0101 080a 4eb4 50fb 0eeb .g.....NP.. 0x0040: 340c

4.

18:18:52.047021 IP (hasta 0x0, ttl 48, id 25347, desplazamiento 0, indicadores [DF], \ longitud: 1500)

test.example.com.www > client.braingia.org.4564:\ . 1:1449(1448) acuse de recibo 462 victoria

6432 <nop,nop,timestamp \

1320440059 250295308>

0x0000: 00e0 1833 2ee8 0090 2741 78f0 0800 4500 ...3....'Ax...E. 0x0010: 05dc 6303 4000 3006

d807 455d 0302 c0a8 ..c.@.0...E]... 0x0020: 010a 0050 11d4 9627 33f8 0c9b 206d

8010 ...P...3....m .. 0x0030: 1920 b9f7 0000 0101 080a 4eb4 50fb 0eeb NP.. 0x0040: 340c

4854 5450 2f31 2e31 2032 3030 204f 4.HTTP/1.1.200.O 4 6a4 630 0x0050: 4 2054 7565 2c20

3237 K..date: .tue, -27 0x0060: 204a 756c 2032 3030 3420 3233 3a31 393a .jul.2004.23: 19:

0x0070: 3030 2047 4d54 0d0a 5365 7276 6572 3a20 00.gmt..server .. 0x0080: 4170 6163 6865

2F31 2E33 2E32 3620 2855 Apache / 1.3.26. (U 0x0090: 6E69 7829 2044 6562 6961 6E20 474E

552F nix) .debian.gnu / 0x00a0: 4c69 6e75 7820 6d6f 6E6F 2F30 Linux.mod_mo / 0 0x00b0:

2e31 3120 6d6f 645F 7065 726C 2F31 2E32 .11.MOD_PERL / 1.2 0x00C0: 360D 0A43 6F6E

6E65 6374 696F 6E3A 2063 6..ConDetección: .c 0x00d0: 6cf6 7365 0d0a 436F 6E74 656E 742D

5479 Pierde..content-ty 0x00e0 : 7065 3a20 7465 7874 2f68 746d 6c3b 2063 pe:.text/html;.c

0x00f0: 6861 7273 6574 3d69 736f 2d38 3835 392d harset=iso-8859-

<salida truncada>

Tenga en cuenta que esta salida contiene una solicitud real (vea el primer paquete, cerca de GET./..HTTP/1.0) y también contiene una parte de la respuesta del servidor web. Todo este tráfico está en texto sin formato porque HTTP no está encriptado. Esta salida contiene hexadecimal y ASCII. Para obtener una salida solo con ASCII, elimine -x y -X del comando y reemplácelos con una sola -A. Personalmente, encuentro que la salida dual hexadecimal y ASCII es útil a veces.

Eso es todo lo que hay que hacer para capturar el tráfico HTTP con TCPDump. Las mejoras obvias para el comando serían expandir la expresión para buscar una fuente o destino específico. Es importante comprender que solo se encontrará tráfico en el puerto 80 con el comando dado. Si está ejecutando tráfico HTTP en otro puerto, sustituya ese puerto en lugar de (o además de) el puerto que se encuentra en el comando de muestra.

Capturar una conversación SMTP

Capturar una conversación SMTP es similar a capturar una sesión HTTP. Comience con las opciones básicas de TCPDump que le gustaría usar y luego cree una expresión para obtener el tipo de datos adecuado, incluido el protocolo, el puerto y los hosts de origen o destino.

Por ejemplo, aquí hay una captura simple del tráfico del puerto 25 junto con mi elección normal de opciones de TCPDump:

```
tcpdump -vv -x -X -s 1500 'puerto 25'
```

El protocolo de enlace de tres vías de TCP vuelve a estar presente, como era de esperar:

```
20:40:08.638690 murphy.debian.org.45772 > prueba.ejemplo.com.smtp: \
  S [tcp suma bien] 1485971964:1485971964(0) ganar 5840 <mss 1460,
    sackOK, marca de tiempo 795074473 0, nop, ws cale 0> (DF) \
    (ttl 57, id 65109, solo 60)
0x0000 4500 003c FE55 4000 3906 DEAE 9252 8A06 0x0010 455D      E..<.U@.9....R..
0302 B2CC 0019 5892 21FC 0000 0000 0x0020 A002 16D0 8FE 0000    ANTERIOR.!.....
0204 05B4 0402 080A 0x0030 2F63 DFA9 0000 0000 0103 0300 20:    .....
40: 08.638769 test.example.com.smtp> murphy.debian .org.45772: S \ /C.....

      [tcp suma bien] 2853594323:2853594323(0) acuse de recibo 1485971965 ganar 5792 \
        <mss 1460,sackOK,timestamp 132 1286843 795074473,nop,wscale 0> \
        (DF) (ttl 64, id 0, longitud 60)
0x0000 4500 003C 0000 4000 4006 D604 455D 0302 0x0010 9252      Y.. <.. @. ... E] ..
8A06 0019 B2CC AA16 64D3 5892 21FD 0x0020 A012 16A0 F5B6        .R.....dX!.
0000 0204 05B4 0402 080A 0x0030 4EC1 3CBB 2F63 DFA9 0103 0300    .....
N. <./ C .....

20:40:08.640600 murphy.debian.org.45772 > prueba.ejemplo.com.smtp: . \
  [tcp sum ok] 1:1(0) ack 1 win 5840 <nop,nop,timestamp \
    795074473 1321286843 > (DF) (ttl 57, id 65110, solo 52)
0x0000 4500 0034 fe56 4000 3906 deb5 9252 8a06 0x0010 455d 0302    E..4.V @ .9 .... R ..
b2cc 0019 5892 21fd aa16 64d4 0x0020 8010 16d0 244c 0000 04x    E] ..... X.!... d.
0101 04x 0fa 9          ...$.L...../c..

N.<.
```

No hay nada realmente nuevo de interés durante el proceso de apretón de manos de tres vías. Aviso, sin embargo, que la salida ASCII no es de mucha utilidad durante el apretón de manos de tres vías.

278 Capítulo 13 Supervisión de red y detección de ataques

Al igual que con HTTP, después de que se realiza el protocolo de enlace TCP inicial, la conversación SMTP se pone en marcha:

```
20:40:08.683352 prueba.ejemplo.com.smtp > murphy.debian.org.45772: P \
[tcip sum ok] 1:51(50) ack 1 win 5792 <nop,nop,timestamp \
1321286848 795074473 > (DF) (ttl 64, id 22639, solo 102)
0x0000 4500 0066 586f 4000 4006 7d6b 455d 0302 0x0010 9252 8a06 E..fXo@.@.}KE] ..
0019 aa16 b2cc 64D4 5892 21fd 0x0020 8018 16a0 bd07 0000 0101 080A .R.....dX!..
4ec1 3cc0 0x0030 2f63 dfa9 3232 3020 6466 7730 2e69 6367 0x0040 .....N.<.
6d65 6469 612e 636f 6d20 4553 4d54 5020 0x0050 506f 7374 6669 7820 /c..220.prueba.exa
2844 6562 6961 6e2f 0x0060 474e 5529 0d0a GNU).. mple.com.ESMTP.
Postfijo.(Debian/

20:40:08.684581 murphy.debian.org.45772 > prueba.ejemplo.com.smtp: . [suma tcp bien]
1:1(0) ack 51 win 5840 <nop,nop,timestamp 795074478 1321286848> (DF) (ttl 57, i
d 65111, solo 52)
0x0000 4500 0034 FE57 4000 3906 0x0010 455d 0302 B2CC 0019 E..4.W@.9...R..
5892 21FD AA16 6506 0x0020 8010 16D0 2410 0000 0101 080A Exe.
2F63 DFAE 0x0030 4EC1 3CC0 20: 40: 08.685428 .....$.c..
Murphy.debian.org.45772> test.example.com.expense.: P [tcp
suma bien]
1:25(24) ack 51 win 5840 <nop,nop,timestamp 795074478 1321286848> (DF) (ttl 57,
identificación 65112, solo 76)
0x0000 4500 004C FE58 4000 3906 DE9B 9252 8A06 0x0010 E..LX@.9...R..
455d 0302 B2CC 0019 5892 21FD AA16 6506 0x0020 8018 16D0 Exe.
3CC4 0000 0101 080A 2CH63 DFAE 0x0030 4EC1 3CC0 4548 ....<...../c..
4C4F 206D 7572 7068 792e 0x0040 6465 6269 616e 2E6F 7267 N.<.EHLO.murphy.
0D0A debian.org..
```

Capturar una conversación SSH

Aunque en realidad no es posible capturar una conversación SSH, puede ver algunas de las partes de configuración de la conexión del protocolo. Sin embargo, debido a que SSH está encriptado, ninguna de las credenciales u otros datos durante la sesión real están disponibles para que los vea. Sin embargo, debe tenerse en cuenta que si puede obtener acceso a la clave privada del servidor, teóricamente podría descifrar el contenido de la conexión SSH. Hacer esto está mucho más allá del alcance de este texto.

Dejaré como ejercicio para el lector capturar una conexión SSH, incluida la configuración, si desea ver cómo se ve una conexión normal para SSH.

Captura de otros protocolos basados en TCP

La captura de otros protocolos basados en TCP sigue un proceso muy similar al de los ejemplos que se muestran. Por ejemplo, se pueden capturar conexiones POP3 y se puede capturar todo el flujo porque POP3, como SMTP, no se cifra durante el tránsito.

Un protocolo es de particular interés porque ha confundido a los administradores de red durante mucho tiempo. Ese protocolo es FTP.

FTP utiliza dos puertos TCP, 20 y 21. El puerto 21 normalmente se usa para comandos y se refiere a veces como el *canal de control*. El puerto 20 en FTP se usa para datos y es

a veces titulado acertadamente el *canal de datos*. Por lo tanto, si desea capturar el tráfico FTP con TCPDump, debe capturar los puertos 20 y 21 para ver todo.

Una tendencia en los últimos años han sido los protocolos que utilizan puertos no estándar para eludir los cortafuegos y la captura y el filtrado de paquetes. Dichos programas, que incluyen gran parte del software peer-to-peer, pueden ser algo difíciles de encontrar durante las capturas de paquetes porque la mayoría de los datos durante la conversación son binarios y, por lo tanto, no son legibles por humanos.

Captura de una consulta de DNS

TCPDump maneja las consultas de DNS de forma un poco diferente a un paquete que es simplemente TCP. Se puede obtener más información solo de la línea de resultado del paquete inicial en lugar de hacer que sea necesario aumentar el snaplen con la opción `-s`. Por ejemplo, considere el siguiente seguimiento de una consulta de DNS simple que buscaba la dirección IP de un host llamado `www.braingia.org`:

```
21:18:39.289121 192.168.1.10.1514 > 192.168.1.1.53: 60792+ A? www.braingia.org. y(34) (DF)
```

```
21:18:39.289568 192.168.1.1.53 > 192.168.1.10.1514: 60792*- 1/2/2 A 192.168.1.50 y(118) (DF)
```

En el seguimiento del paquete, vemos el host `192.168.1.10` en un puerto efímero que se comunica con un destino de `192.168.1.1` en el puerto `53`. Se proporciona un número de ID de consulta; en este caso es `60792`. Verá que el número de ID de la consulta va seguido de un `+`. Este símbolo indica que el interrogador solicitó recursividad en esta consulta. ¿La `A?` en la línea de seguimiento indica que se trataba de una consulta de dirección. La consulta fue para `www.braingia.org`, como se muestra, y el tamaño de la consulta es de `34` bytes, que no incluye la sobrecarga de IP o UDP.

La respuesta llega rápidamente, como vemos en la siguiente línea del seguimiento, que muestra que la fuente ahora es `192.168.1.1` hablando con el destino de `192.168.1.10`. Como puede ver, la respuesta estaba contenida en el mismo ID de consulta, `60792`; sin embargo, esta vez hay dos caracteres adicionales, el `*` y el `-`. El `*` en una respuesta indica que se trata de una respuesta autorizada y el `-` indica que la recursividad está disponible y no configurada. La siguiente parte de la respuesta, `1/2/2`, indica la cantidad de registros de respuesta (`1`), la cantidad de registros del servidor de nombres (`2`) y la cantidad de registros adicionales (`2`). La primera respuesta dada en este caso es de tipo `A` y es `192.168.1.50`. Finalmente, el tamaño de la respuesta es de `118` bytes.

Captura de pings

Aunque pueda parecer bastante inocente, ICMP (el protocolo detrás de ping) se ha utilizado con bastante frecuencia como un medio para atacar hosts y causar estragos. Por lo tanto, como analista de seguridad, administrador o espectador curioso, debe saber que le conviene ver parte de la actividad normal de ICMP a través de TCPDump para que pueda detectar una anomalía más adelante.

Me arriesgaré y diré que ICMP se usa con mayor frecuencia para la solicitud de eco simple y la respuesta de eco proporcionada por ping. Sin embargo, ICMP puede usarse y se usa para mucho más que eso, incluso informar a un remitente rápido cuándo reducir la velocidad (Source Quench), redirigir a otros hosts (Redirect) y muchas otras áreas. Consulte el Capítulo 1 para obtener más información sobre ICMP o, como siempre, consulte el RFC original sobre ICMP para obtener información autorizada sobre el protocolo.

Ataques a través de los ojos de TCPDump

Ha visto cómo se ven los seguimientos de paquetes TCP y UDP normales a través de TCPDump, pero ¿cómo sabrá si alguien o algo está actuando de manera anormal? Desafortunadamente, encontrar actividad nefasta no es tan fácil. Enterrados en los rastros de paquetes normales pueden haber señales de que alguien está intentando un ataque a su servidor. Obviamente, un atacante intentará ocultar su actividad, lo que dificulta aún más la detección. No solo tiene que atravesar todo el tráfico normal dentro del seguimiento de un paquete, sino que también tiene que buscar la proverbial aguja en un pajar para encontrar lo que puede ser un intento de ataque o incluso uno en progreso.

Recuerde del Capítulo 11, "Detección y respuesta a intrusiones", que no todo lo que las caídas fuera de la actividad normal se pueden denominar un ataque. Parte de esto es el resultado de mal funcionamiento o equipo mal configurado. La mayoría de las veces, la actividad anormal detectada en el seguimiento de un paquete se debe al reconocimiento de una forma u otra. Y la gran mayoría del trabajo de reconocimiento se realiza a través de la automatización. En lugar de pasar muchas horas sin frutos buscando un host vulnerable, los atacantes automatizarán el proceso y harán que el programa les avise cuando encuentre un host interesante.

Naturalmente, hay excepciones a la regla de la automatización. Los ataques pueden ser el resultado de actividad dirigida contra su servidor o red. Antes del ataque suele haber algún reconocimiento manual que se lleva a cabo. Esto puede incluir que el posible atacante elabore o cree manualmente paquetes para intentar explotar posibles agujeros en su servidor o red. Sin embargo, la mayoría de las veces, el atacante habrá tenido algunos datos de reconocimiento automatizados que dirigieron sus esfuerzos en su dirección. Si un análisis automatizado alertó al atacante de que uno de sus servidores puede ser vulnerable a un tipo particular de ataque, es posible que encuentre los hosts o toda su subred a la vista del atacante.

Dejar un host vulnerable o hacer que un host parezca vulnerable y luego observar los ataques es la premisa detrás de un honeypot. Un *honeypot* es un host o dispositivo que se muestra vulnerable a un atacante y, por lo tanto, parece un objetivo para el ataque. La idea es que al observar los métodos que utilizan los atacantes para explotar un agujero o al observar lo que hacen cuando comprometen al host, el observador puede aprender de ello y defenderse de tal actividad.

Como si todas las posibles razones ya dadas para ver actividad anormal no fueran suficientes, aquí hay uno más: también encontrará conexiones accidentales que pueden parecer ataques. En otras palabras, a veces alguien simplemente escribe mal una dirección IP cuando intenta conectarse a su servidor. Cualquiera que haya contestado el teléfono solo para descubrir que la persona que llamó marcó incorrectamente puede identificarse con esta situación.

En resumen, hay algunas categorías básicas dentro de las cuales podría caer la actividad anormal:

- ÿ Escaneo de reconocimiento automatizado o semiautomatizado
- ÿ Ataque dirigido
- ÿ Equipo mal configurado

ÿ Número incorrecto

ÿ Equipo averiado

Con esas categorías en mente, esta sección examina algunos rastros de paquetes anormales o rastros que no debería ver en condiciones normales. De ninguna manera esta sección incluye todos los posibles paquetes manipulados y anormales. La esperanza es brindarle una comprensión de algunos de los tipos de cosas que debe buscar al realizar una investigación.

Exploración normal (Nmap)

A veces, un atacante escaneará su subred o dirección IP individual en busca de puertos abiertos. Este análisis puede ser cualquier cosa, desde un intento inocente de buscar un servicio hasta el reconocimiento de un ataque. Muchas veces, estos escaneos están completamente automatizados, con un atacante configurando uno o más robots (bots) para escanear automáticamente en busca de versiones vulnerables de software para explotar.

Esta simulación fue creada con el programa Nmap con la siguiente línea de comando:

```
nmap-sT 192.168.1.2
```

La captura de TCPDump del escaneo de puertos se muestra en el siguiente texto; tenga en cuenta que he truncado la salida porque Nmap buscó más de 1650 puertos. He dividido la captura para facilitar la explicación también.

Los escaneos de Nmap comienzan con una solicitud de eco ICMP al host de destino, como se muestra aquí. Tenga en cuenta, sin embargo, que este intercambio ICMP puede ser deshabilitado por la persona que ejecuta el escaneo Nmap, por lo que es posible que no siempre aparezca:

```
12:31:21.834284 IP 192.168.1.10 > 192.168.1.2: icmp 8: solicitud de eco secuencia 27074
12:31:21.834508 IP 192.168.1.2 > 192.168.1.10: icmp 8: respuesta de eco sec 27074
```

A continuación, Nmap busca el puerto 80, el puerto HTTP predeterminado. Observe que el escaneo proviene de un puerto efímero en el costado del escáner dirigido al puerto 80 en el destinatario. En este caso, el host del destinatario 192.168.1.2 está escuchando en el puerto 80 y se envía una respuesta de TCP al host de escaneo con el indicador TCP RST establecido y el número de secuencia establecido:

```
12:31:21.834318 IP 192.168.1.10.60034 > 192.168.1.2.80: . reconocer 2624625246 ganar 4096
12:31:21.834363 IP 192.168.1.2.80 > 192.168.1.10.60034: R\
2624625246:2624625246(0) ganar 0
```

A continuación, Nmap busca el puerto telnet (tcp/23). Note la diferencia entre esto y el escaneo anterior. Además de que los puertos son diferentes, el paquete de respuesta también es diferente. En este caso, el host receptor no está escuchando en el puerto TCP 23, por lo que responde con un paquete con el indicador TCP RST establecido pero con el número de secuencia TCP establecido en 0:

```
12:31:21.935005 IP 192.168.1.10.3171 > 192.168.1.2.23: S 752173650:752173650(0) \
ganar 5840 <mss 1460, sackOK, marca de tiempo 1421906912 0, nop, wscale 0>
12: 31: 21.935046 IP 192.168.1.2.23> 192.168.1.10.3171: R 0: 0 (0) ack 752173651 0win 0
```

Los siguientes paquetes son esencialmente los mismos que el escaneo de puerto telnet anterior en la medida en que el host del destinatario no está escuchando en los puertos que se escanean:

```
12:31:21.935129 IP 192.168.1.10.3172 > 192.168.1.2.554: S 758180552:758180552(0) y\
    ganar 5840 <mss 1460, sackOK, marca de tiempo 1421906912 0, nop, wscale 0>
12: 31: 21.935186 IP 192.168.1.2.554> 192.168.1.10.3172: R 0: 0 (0) ack 758180553 0win 0

12:31:21.935149 IP 192.168.1.10.3174 > 192.168.1.2.21: S 751983738:751983738(0) \
    ganar 5840 <mss 1460, sackOK, marca de tiempo 1421906912 0, nop, wscale 0>
12:31:21.935289 IP 192.168.1.2.21 > 192.168.1.10.3174: R 0:0(0) ack 751983739 victoria y\

12:31:21.935255 IP 192.168.1.10.3175 > 192.168.1.2.1723: S 757954867:757954867(0) y\

    ganar 5840 <mss 1460, sackOK, marca de tiempo 1421906912 0, nop, wscale 0>
12:31:21.935320 IP 192.168.1.2.1723 > 192.168.1.10.3175: R 0:0(0) \
    ack 757954868 ganar 0
```

Finalmente, se encuentra otro puerto abierto. Esta vez el puerto abierto es tcp/25, el conocido puerto SMTP:

```
12:31:21.935381 IP 192.168.1.10.3176 > 192.168.1.2.25: S 762467904:762467904(0) \
    ganar 5840 <mss 1460, sackOK, marca de tiempo 1421906912 0, nop, wscale 0>
12:31:21.935448 IP 192.168.1.2.25 > 192.168.1.10.3176: S 2645882457:2645882457(0) y\

    ack 762467905 win 5792 <mss 1460,sackOK,timestamp \
    921140115 1421906912, espárrago, wscale 7>
```

Como se indicó anteriormente, el escaneo de Nmap continúa para otros 1650 o más puertos. Elegí guardar un árbol al no mostrar el resto de los escaneos de puertos aquí. Tenga la seguridad de que se ven en gran medida iguales a los que ya se muestran.

La respuesta que tome cuando se escanee el puerto depende de su política de seguridad. Si observo un escaneo de puerto amplio, uno en el que se escanea una gran cantidad de puertos, generalmente tomo medidas para bloquear el host. Sin embargo, debido a que no estoy en la computadora las 24 horas del día (casi, pero no del todo), uso una herramienta llamada PortSentry para monitorear este tipo de actividad. Sin embargo, existe otro software de escaneo antipuerto, incluido un complemento para Snort.

Ataque de pitufo

Un ataque Smurf es un ataque DoS mediante el cual el atacante envía solicitudes de eco ICMP con una dirección de origen falsificada a una o más direcciones de transmisión. La dirección de origen falsificada es el destinatario del ataque y se verá inundada con respuestas de eco de las direcciones de transmisión de otras redes. Imagine respuestas de eco provenientes de 254 hosts dirigidos a una máquina con una conexión a Internet pequeña o lenta. Ahora imagine esas respuestas provenientes de 100 redes de 254 hosts cada una. No pasa mucho tiempo antes de que toda una red se atasque al recibir respuestas ICMP.

El seguimiento siguiente se creó con el comando hping3 :

```
hping3 -1 -a 192.168.1.2 192.168.1.255
```

En la red bajo ataque, solo un host respondió al ping transmitido; cómo nunca, no hay forma de garantizar que otras redes no tengan una gran cantidad de hosts que respondan:

```
12:57:06.871156 IP 192.168.1.2 > 192.168.1.255: icmp 8: solicitud de eco secuencia 0
12:57:06.871637 IP 192.168.1.8 > 192.168.1.2: icmp 8: secuencia de respuesta de eco 0
12:57:07.870259 IP 192.168.1.2 > 192.168.1.255: icmp 8: solicitud de eco secuencia 256
12:57:07.871008 IP 192.168.1.8 > 192.168.1.2: icmp 8: respuesta de eco secuencia 256
12:57:08.870132 IP 192.168.1.2 > 192.168.1.255: icmp 8: solicitud de eco secuencia 512
12:57:08.870880 IP 192.168.1.8 > 192.168.1.2: icmp 8: respuesta de eco secuencia 512
```

No existe una buena defensa basada en el host para un ataque Smurf. Incluso si las respuestas ICMP están deshabilitadas en el host individual, el ancho de banda sigue siendo consumido por todas las respuestas que llegan a la red.

Para contrarrestar eficazmente un ataque Smurf, las solicitudes de eco ICMP dirigidas a direcciones de transmisión no deben cruzar los límites del enrutador. Esto significa que estás confiando en que otros sean buenos internautas. Además, las respuestas de eco ICMP deben filtrarse lo más arriba posible de su ubicación. Sin embargo, no soy partidario de filtrar las respuestas de eco ICMP. Una mejor solución es limitar la velocidad de las respuestas de eco lo más arriba posible y al mismo tiempo permitir las respuestas por todo el bien que hacen en el diagnóstico de problemas.

Árbol de Navidad y banderas de encabezado TCP

El ataque del árbol de Navidad se llama así porque todos los indicadores de bits están activados dentro del encabezado TCP. La idea es hacer que el host receptor responda, provocando así un DoS. Recuerde los bits de marca TCP SYN, RST, ACK, URG y otros del Capítulo 1. Estos bits nunca deben aparecer todos al mismo tiempo, y cuando lo hacen es una indicación de un paquete manipulado.

Los ataques del árbol de Navidad son bastante poco comunes. Sin embargo, es importante considerar el TCP banderas al examinar los paquetes. Establecer estos indicadores con combinaciones no válidas es casi siempre una indicación de un paquete manipulado (aunque también en algunos casos podría indicar software roto o mal configurado). El objetivo del paquete diseñado puede ser cualquier cosa, desde el reconocimiento hasta un ataque activo, como uno para atravesar un firewall.

La siguiente captura establece los indicadores TCP SYN, FIN, RST y PUSH, que nunca deberían aparecer en un paquete real. Fue creado con el comando hping3 :

```
hping3 -SFRP 192.168.1.2
```

Hay tres paquetes en esta captura. Observe que el puerto de origen aumenta y que el puerto de destino es 0. También se muestran las banderas TCP, SFRP en este caso. Ver esto en la naturaleza debería hacer que el analista de intrusiones comience inmediatamente a investigar los paquetes de acuerdo con la política de seguridad.

```
13:20:03.989780 IP (to 0x0, ttl 64, id 2270, desplazamiento 0, indicadores [ninguno], longitud: \
40) 192.168.1.10.2687 > 192.168.1.2.0: SFRP [tcp suma bien] \
925164686:925164686(0) ganar 512
13:20:04.989734 IP (hasta 0x0, ttl 64, id 9285, desplazamiento 0, indicadores [ninguno], \
longitud: 40) 192.168.1.10.2688 > 192.168.1.2.0: SFRP [tcp sum ok] \
1113258177:1113258177(0) ganar 512
13:20:05.989731 IP (hasta 0x0, ttl 64, id 26951, desplazamiento 0, indicadores [ninguno], \
longitud: 40) 192.168.1.10.2689 > 192.168.1.2.0: SFRP [tcp sum ok] \
2097818687:2097818687(0) ganar 512
```

284 Capítulo 13 Supervisión de red y detección de ataques

Ataque terrestre

El ataque LAND es un ataque DoS contra computadoras que ejecutan Microsoft Windows. Se informó originalmente que el ataque afectó a Windows 95 y Windows NT en 1997.

Microsoft finalmente corrigió la vulnerabilidad de los sistemas operativos. Sin embargo, la vulnerabilidad resurgió en los sistemas operativos más nuevos de Microsoft, incluidos Windows XP Service Pack 2 e incluso Windows Server 2003.

El ataque LAND es bastante trivial y ocurre cuando las direcciones de origen y destino y los puertos se establecen en el host del destinatario y se establece el indicador SYN.

hping3 nuevamente proporciona una manera fácil de recrear esto para probar:

```
hping3 -k -S -s 25 -p 25 -a 192.168.1.2 192.168.1.2
```

A continuación se muestra la captura a través de TCPDump. Observe que las direcciones y los puertos de origen y destino son los mismos y que el puerto de origen no se incrementa y que también está debajo de los puertos efímeros:

```
13:42:28.079339 IP 192.168.1.2.25 > 192.168.1.2.25: S 764505725:764505725(0) victoria y512
```

```
13:42:29.079462 IP 192.168.1.2.25 > 192.168.1.2.25: S 2081780101:2081780101(0) \  
ganar 512
```

```
13:42:30.079461 IP 192.168.1.2.25 > 192.168.1.2.25: S 390202112:390202112(0) victoria y512
```

Grabación de tráfico con TCPDump

Mientras consultaba a un pequeño proveedor de Internet, noté que había un pico rutinario y significativo en el tráfico de la red alrededor de las 3:00 am todas las mañanas, que duraba entre 15 minutos y una hora. Mi objetivo era determinar la causa de este pico de tráfico. Debido a que el tráfico era rutinario ya una hora extraña, mi pensamiento inicial fue que era el resultado de un proceso de actualización automática de los servidores de la red.

La mayoría de los servidores de la red ejecutaban Debian Linux y usaban apt-proxy.

Esto significaba que solo un servidor local se comunicaría con los servidores de actualización de Debian fuera del sitio y obtendría las actualizaciones necesarias. Todos los demás servidores de la red local contactarían entonces con ese servidor maestro local. Esta configuración redujo enormemente la utilización de Internet.

Aunque el servidor maestro sin duda podría ser un factor contribuyente, no sentí que habría suficiente tráfico de actualización cada noche para justificar un aumento tan significativo en el tráfico. Mi suposición se confirmó cuando miré el programa de actualización en el servidor maestro y descubrí que en realidad estaba buscando actualizaciones en un momento diferente de todos modos y, por lo tanto, no estaba contribuyendo en absoluto al pico de las 3:00 am.

Con esa causa eliminada, necesitaba mirar el tráfico mismo a las 3:00 a.m. Sin embargo, Realmente no tenía muchas ganas de permanecer despierto hasta esa hora, y si estaba despierto, podría no estar en forma para leer el rastro de un paquete. Ingrese cron. Al usar cron para disparar TCPDump, pude capturar el seguimiento del paquete en un archivo para su posterior análisis. No fue una gran sorpresa aquí y no estaba abriendo nuevos caminos, pero parecía una solución justa al problema. Configuré el interruptor para copiar paquetes al puerto en el que estaba conectada la máquina de monitoreo y me puse a trabajar en la parte de TCPDump.

TCPDump ofrece un par de funciones que son útiles para este tipo de seguimiento. La primera característica es la capacidad de escribir la salida en un archivo (grabación, por así decirlo) y luego leer ese archivo más tarde (reproducción). La segunda característica útil es la capacidad de salir después de capturar una cierta cantidad de paquetes. Por supuesto, podría haber usado otros medios para detener el seguimiento del paquete, como otro trabajo cron para eliminar la captura de TCPDump, pero pensé que usar la capacidad nativa de TCPDump era la solución más rápida y fácil.

Todos los comandos TCPDump que he mostrado hasta ahora en este capítulo han usado expresiones como puerto 80 o host <n>.<n>.<n>.<n>. Las expresiones son útiles cuando busca tráfico específico y conocido. Sin embargo, las expresiones no son de mucha ayuda cuando no está seguro de lo que está buscando exactamente, como fue el caso aquí. La mejor opción era capturar todo y luego trabajar en un filtro durante la reproducción.

Las dos opciones de interés de TCPDump que aún no se han tratado en este capítulo son -w y -c. La opción -w hace que TCPDump coloque la salida sin procesar en el archivo especificado para que luego pueda ser retroalimentada a través de TCPDump. El archivo resultante está en el formato nativo de TCPDump y, por lo tanto, no es legible por un buscador de texto sin formato como cat, less o more. La opción -c informa a TCPDump que debe salir después de capturar <N> paquetes. Obtener ting <N> correcto parecía ser la parte más difícil de la captura.

Los resultados de la captura necesitaban mostrarme solo información básica sobre el paquete, incluidos el origen y el destino, así como algunos bits del paquete. Con ese fin, el comando TCPDump fue bastante fácil de crear:

```
/usr/sbin/tcpdump -c 25000 -w archivo de volcado -n
```

En este comando tengo TCPDump saliendo después de capturar 25,000 paquetes, escribiendo el capturador en un archivo llamado dumpfile y no realizar consultas de DNS para el origen y el destino. Llegar a este comando requirió cierto nivel de prueba para ver cuánto tiempo tomó capturar 25,000 paquetes y qué información se incluyó en la captura. Después de probarlo, ingresé el comando en cron con este programa:

```
5 3 * * * /usr/sbin/tcpdump -c 25000 -w archivo de volcado -n
```

Es decir, todas las mañanas a las 3:05 am se realizaría la captura. Luego, a una mejor hora, como a las 11:00 am cuando me levanto de la cama, miraría para ver si realmente hubo tráfico la noche anterior que me obligó a mirar el archivo de volcado. Si hubiera tráfico, iniciaría sesión en el servidor y ejecutaría un comando para leer el archivo de volcado:

```
tcpdump -r archivo de volcado -X -vv
```

Ejecutar este comando me dio una idea de qué tráfico había. Entremezclada con el tráfico normal había una conversación FTP entre uno de los clientes más importantes del ISP y otro host en Internet. El tráfico FTP fue fácilmente el paquete más frecuente que estaba viendo dentro del seguimiento. El rastro del paquete de otra noche lo confirmó. Hice que el ISP se pusiera en contacto con su cliente para averiguar si se trataba de una actividad conocida y, de ser así, para que el cliente supiera que podría estar sobrepasando su asignación de ancho de banda para el mes.

Este ejemplo es algo típico del trabajo de un analista de seguridad. Detecte una anomalía, invéstiguela mientras descarta posibilidades y tome medidas en función de la investigación.

Aunque la causa última de esta anomalía en particular resultó no ser ningún tipo de ataque no autorizado, el resultado de la investigación fue un cliente más feliz porque ese cliente podía tomar medidas correctivas antes de exceder su ancho de banda para el mes.

Supervisión de intrusiones automatizada con Snort

Snort es un excelente paquete de software de detección de intrusos que combina la mejor tecnología de su clase con la capacidad de configuración de código abierto. Snort en realidad tiene algunos modos de operación diferentes, incluido un modo sniffer, un modo de registro de paquetes, un modo de detección de intrusos y lo que se llama modo en línea. Es el modo de detección de intrusos lo que interesa en esta sección. Sin embargo, el modo en línea también es notable porque proporciona una forma de configurar Snort e iptables para trabajar juntos para aceptar o descartar paquetes de forma dinámica según las reglas de Snort. A los efectos de este capítulo, cuando me refiero a Snort me refiero específicamente al modo de detección de intrusos.

Cuando está en modo de detección de intrusos, Snort funciona mediante el uso de una serie de reglas que definen el tráfico anómalo. Muchas de estas reglas vienen predefinidas por Sourcefire, los creadores de Snort. Muchas otras reglas están disponibles en la comunidad y, por supuesto, también puede escribir sus propias reglas según sea necesario.

Además de las reglas, Snort tiene varios preprocesadores que permiten que los módulos vean y altere los paquetes antes de que sean manejados por el motor de detección de intrusos del software. Los preprocesadores se pueden desarrollar para satisfacer sus necesidades, aunque los preprocesadores ya disponibles son útiles. Los preprocesadores preexistentes incluyen dos tipos de detectores de escaneo de puertos para ayudar a detectar y tomar medidas cuando se detecta un escaneo de puertos. También hay preprocesadores para volver a ensamblar flujos de TCP para proporcionar análisis de estado y preprocesadores para decodificar el tráfico RPC e inspeccionar el tráfico HTTP. Otros preprocesadores se describen en detalle en la documentación de Snort disponible con Snort o en línea en <https://www.snort.org/documents>.

Snort funciona detectando e informando sobre eventos. El proceso real de informar sobre eventos se puede configurar a través del manejo de eventos dentro de Snort. Gestión de llamadas de eventos para configuración basada en umbrales. Este aspecto altamente configurable de Snort ayuda a evitar que se inunde con entradas de registro y alertas.

Normalmente le gustaría ser notificado de alguna manera cuando se activen ciertas reglas de Snort. Snort utiliza módulos de salida que se pueden configurar para enviar la salida a varias ubicaciones. Un módulo de salida de uso común es el módulo `alert_syslog`, que envía alertas a la instalación de syslog local. Existen otros módulos de salida, incluidos `alert_fast` y `alert_full`. El primero pone una entrada rápida en el archivo especificado y el segundo envía el encabezado del paquete completo junto con el mensaje del evento. Existen otros módulos de salida, y se puede encontrar más información sobre ellos en la documentación de Snort.

Un módulo de salida interesante es el módulo de salida de la base de datos. El módulo de base de datos permite enviar alertas de Snort a una base de datos SQL. El uso de este módulo de salida le permite aprovechar el software que puede generar informes sobre las alertas y eventos en Snort.

Snort tiene numerosas funciones y matices adicionales que lo convierten en uno de los mejores, si no el mejor, software de detección de intrusos disponible.

Obtención e instalación de Snort

Snort está disponible con muchas distribuciones de Linux como un paquete adicional, y la mayoría de las distribuciones también incluyen las reglas de Snort, incluidas con el paquete Snort o como un paquete adicional. También puede descargar Snort desde <http://www.snort.org/>.

La instalación de Snort y las reglas predeterminadas casi siempre se deben realizar instalando el paquete disponible con su distribución. Si esto no es posible o si el paquete disponible no incluye las opciones que desea, deberá compilar desde la fuente.

El paquete Snort viene en un archivo gzip y, por lo tanto, debe descomprimirse y desarchivado antes de ser compilado:

```
tar -zxvf snort-<versión>.tar.gz
```

Después de descomprimirlo y desarchivarlo, puede hacer un cd en el directorio fuente de Snort y ejecutar el guión de configuración:

```
cd snort-<versión>
./configurar
```

Es en el momento de ejecutar el script de configuración donde se pueden configurar muchas opciones de tiempo de compilación. Para obtener una lista de algunas de estas opciones, en particular las opciones para habilitar la compatibilidad con ciertas bases de datos o habilitar otras características, escriba lo siguiente:

```
./configurar --ayuda
```

Además, el documento INSTALL y otra documentación dentro del directorio <snort-source>/doc explican estas y otras opciones disponibles al compilar Snort desde la fuente.

Ejecutar el script de configuración, junto con cualquier opción, hará que Snort busque varios requisitos previos. Por ejemplo, al compilar Snort desde la fuente, puede recibir un error que indica que no se pueden encontrar uno o más requisitos previos, como este error:

```
buscando pcre.h... no
¡ERROR! Encabezado libpcre no encontrado, ve a buscarlo
http://www.pcre.org
```

Con ese error en mente, pude instalar los archivos de desarrollo pcre , volver a ejecutar el script de configuración y continuar.

Después de que el script de configuración se haya ejecutado con éxito, compile el software escribiendo esto: hacer

El software ahora se compilará. Si obtiene algún error durante esta fase, consulte la documentación de Snort y los archivos de la lista de correo para ver si ha recibido un error conocido para su arquitectura.

Finalmente, después de compilar el software, instálelo escribiendo lo siguiente: hacer instalar

288 Capítulo 13 Supervisión de red y detección de ataques

El software ahora se instalará y debería estar listo para usar. Por defecto, el programa se instala en `/usr/local/bin`. Puede probar el comando básico Snort escribiendo esto:

```
/usr/local/bin/snort -?
```

Debería ver una salida con opciones de ayuda, similar a esta:

```
__      -> Bufa! <*-
o")~ Versión 2.3.3 (Compilación 14)
"" Por Martin Roesch y el equipo de Snort: http://www.snort.org/team.html
    (C) Copyright 1998-2004 Sourcefire Inc., et al.

USO: ./snort [-opciones] <opciones de filtro>
Opciones:
    -A      Establezca el modo de alerta: rápido, completo, consola o ninguno (solo alertas de archivos de alerta)
            "unsock" habilita el registro de socket UNIX (experimental).
...
<salida truncada>
```

Configuración de Snort

El código fuente de Snort incluye un archivo de configuración de muestra. Si ha instalado desde el paquete de su distribución, eso también debería incluir un archivo de configuración de Snort de muestra. Por lo general, este archivo se llama `snort.conf`. En las distribuciones más populares, incluida Debian, este archivo (junto con varias reglas de Snort) se coloca en `/etc/snort/`.

Si está trabajando con una instalación fuente, el archivo de configuración de muestra `snort.conf` se encuentra en `<snort-source>/etc/` y las reglas de muestra se encuentran en `<snort-source>/normas/`. Para aquellos que trabajan con una versión de código fuente, recomiendo crear un directorio en `/etc/` o `/usr/local/etc/` llamado `snort` y colocar el archivo `snort.conf` archivo de configuración y las reglas de Snort en ese directorio. Además, el archivo de configuración predeterminado de Snort también llama a varios mapas y archivos de configuración adicionales. Estos archivos también se pueden encontrar en el directorio `<snort-source>/etc/`. Crear el directorio y copiar todos los archivos en él se vería así (nuevamente, esto solo se aplica a aquellos que compilaron Snort desde la fuente):

```
mkdir /etc/snort
cp <fuente-snort>/etc/snort.conf /etc/snort/
cp <fuente-snort>/etc/*.map /etc/snort/
cp <fuente-snort>/etc/*.config /etc/snort/
cp <fuente-snort>/rules/*.rules /etc/snort/
```

Se realiza un cambio adicional e importante en el archivo de configuración `snort.conf`. Después de copiarlo en el directorio `/etc/snort`, edite el archivo y cambie la variable `RULE_PATH` de su valor predeterminado de `../rules` a `/etc/snort`. La línea debería verse así cuando hayas terminado:

```
var RULE_PATH /etc/snort
```

Finalmente, crea el directorio de registro de Snort con el siguiente comando:

```
mkdir /var/log/snort
```

```
[**] tráfico del puerto 0 tcp de BAD-TRAFFIC [**]  
07/06-16:19:00.712543 192.168.1.10:1984 -> 192.168.1.2:0  
TCP TTL:64 TOS:0x0 ID:48557 IpLen:20 DgmLen:40  
*2***** Secuencia: 0xED1609B Confirmación: 0x13E893C5 Ganancia: 0x200 TcpLen: 20
```

```

[**] tráfico del puerto 0 tcp de BAD-TRAFFIC [**]
07/06-16:19:00.712610 192.168.1.20 -> 192.168.1.10:1984
TCP TTL:64 TOS:0x0 ID:10034 IpLen:20 DgmLen:40 DF
***A*R** Secuencia: 0x0 Confirmación: 0xED1609B Ganancia: 0x0 TcpLen: 20
=====

```

Como puede ver en la salida, Snort ha capturado lo que cree que es (correctamente) paquetes TCP defectuosos. El archivo de registro de alertas `/var/log/snort/alert` también contiene información que es especialmente útil para ordenar las alertas. Estas son las entradas de registro del archivo de registro de alertas que corresponden a las entradas mostradas anteriormente del archivo de host específico:

```
[**] [1:524:8] tráfico del puerto 0 tcp de BAD-TRAFFIC [**]  
[Clasificación: Actividad miscelánea] [Prioridad: 3]  
07/06-16:19:00.712543 192.168.1.10:1984 -> 192.168.1.2:0  
TCP TTL:64 TOS:0x0 ID:48557 IpLen:20 DgmLen:40  
***** Secuencia: 0xED1609B Confirmación: 0x13E893C5 Ganancia: 0x200 TcpLen: 20
```

```

[**] [1:524:8] tráfico del puerto 0 tcp de BAD-TRAFFIC [**]
[Clasificación: Actividad miscelánea] [Prioridad: 3]
07/06-16:19:00.712610 192.168.1.2:0 -> 192.168.1.10:1984
TCP TTL:64 TOS:0x0 ID:10034 IPLen:20 DgmLen:40 DF
***A*1R** Secuencia: 0x0 Confirmación: 0xED1609B Ganancia: 0x0 TcpLen: 20

```

Como puede ver en estas entradas, hay algunos elementos adicionales, como Clasificación y Prioridad , que pueden ayudar a clasificar y priorizar la alerta. Tanto la clasificación como la prioridad se pueden configurar dentro del archivo de registro de alertas.

Recibir alertas

Recomiendo trabajar con Snort para ganar experiencia con las reglas y opciones de configuración antes de configurarlo para enviar alertas por correo electrónico o por otro medio. Es posible que se sienta abrumado por las alertas con las reglas de Snort predeterminadas, según el diseño de su red.

Recuerde del Capítulo 11 que se introdujo el software de monitoreo de archivos de registro y se dio una receta para usar Swatch para monitorear archivos de registro en busca de ciertos eventos, en cuyo momento enviaría un correo electrónico basado en la alerta. Si ves a dónde voy con esto. ¡felicidades!

Uso de Swatch para monitorear alertas de resoplido

Con su configuración predeterminada, Snort inicia sesión en `/var/log/snort/alert`. Por lo tanto, crear una configuración de Swatch para monitorear este archivo es bastante fácil. Nuevamente, sería fácil abrumarse a sí mismo o al sistema con alertas y correos electrónicos de Swatch, por lo que debe tener cuidado al configurar cualquier acción basada en alertas de Snort hasta que haya tenido la oportunidad de configurar Snort más.

Recuerde que Snort registra algunos datos de priorización dentro de `/var/log/snort/alert`. Ahí Por lo tanto, puede configurar una regla de Swatch para buscar cualquier cosa con cierta prioridad, digamos 3, por ejemplo, y enviar un correo electrónico cuando se vea. Esto se colocaría dentro de su archivo de configuración de Swatch que, de forma predeterminada, es `~/swatchrc`. Aquí está la entrada de configuración:

vigilar /Prioridad: 3/
correo

Iniciar Swatch y apuntarlo hacia el archivo de alerta de Snort, `/var/log/snort/alert`, se ve así:

```
muestra --tail-file=/var/log/snort/alert
```

Ahora, cuando se registre una alerta con Prioridad: 3 , Swatch enviará un correo electrónico.

Reflexiones finales sobre Snort

Snort es muy recomendable como medio para automatizar la tarea de detección de intrusos. Solo he podido tocar los conceptos básicos de Snort aquí con la esperanza de brindarle un punto de partida para trabajar con él. Desde aquí, puede combinar Snort con MySQL y ACID para crear un sistema de detección de intrusos de clase empresarial. Snort puede configurarse tal como lo necesita y ampliarse para adaptarse a cualquier tamaño de organización.

Monitoreo con ARPWatch

ARPWatch es un demonio que busca nuevas interfaces Ethernet en una red. Si se ve una nueva entrada ARP, podría ser indicativo de una computadora no autorizada en algún lugar dentro de la red.

ARPWatch utiliza la biblioteca PCap, que puede que (todavía) no esté en su sistema. Si no es así, lo descubrirá durante el proceso de configuración de ARPWatch. La biblioteca PCap, comúnmente conocida como libpcap, se puede descargar desde <http://www.tcpdump.org/>. La biblioteca PCap se utiliza para otros programas relacionados con la red y la seguridad, como TCPDump. Debido a que TCPDump ya estaba cubierto, me abstendré de repetir las instrucciones para instalar libpcap en este capítulo y, en su lugar, lo referiré a la sección "TCPDump: una vista general simple" para obtener esas instrucciones.

La instalación de ARPWatch implica descomprimir el archivo ARPWatch que descargó, generalmente algo como `tar -zxvf arpmatch.tar.Z`. Desde allí, cambie el directorio al directorio ARPWatch y ejecute el script de configuración:

```
./configurar
```

Verá una (espero que un poco) serie familiar de declaraciones de salida, algo como esto:

```
creando caché ./config.cache
comprobando el tipo de sistema host... i686-pc-linux-gnu
comprobando el tipo de sistema de destino... i686-pc-linux-gnu
comprobando el tipo de sistema de compilación... i686-pc-linux-gnu
buscando gcc... gcc
comprobando si el compilador de C (gcc) funciona... sí
... (salida truncada) ...
```

Si ve un error en el sentido de lo siguiente, deberá instalar libpcap:

```
buscando main en -lpcap... no
configure: error: consulte el documento INSTALL para obtener más información
```

Consulte la sección sobre TCPDump anteriormente en este capítulo para obtener información sobre la instalación de la biblioteca PCap.

Para el resto de ustedes, y si se unen a nosotros nuevamente después de instalar PCap, el siguiente paso para compilar ARPWatch es hacerlo . Desde la línea de comando dentro del directorio de código fuente de ARPWatch, escriba esto:

hacer

ARPWatch ahora compilará y verá mensajes que indican el progreso, así como posiblemente una advertencia o dos:

```
report.o(.text+0x409): el uso de 'mktemp' es peligroso, mejor usa 'mkstemp'
gcc -O2 -DDEBUG -DHAVE_FCNTL_H=1 -DHAVE_MEMORY_H=1 -DTIME_WITH_SYS_TIME=1 \
-DHAVE_BCOPY = 1 -DHAVE_STRERROR = 1 -DRETSIGTYPE = vacío -DRETSIGVAL =
-DHAVE_SIGSET=1 -DDECLWAITSTATUS=int -DSTD_HEADERS=1 \
-DARPDIR="/usr/local/arpwatch/" -DPATH \
_SENDMAIL="/usr/sbin/sendmail" -l\
-l\linux-incluye -c ./arpsnmp.c
gcc -O2 -DDEBUG -DHAVE_FCNTL_H=1 -DHAVE_MEMORY_H=1 -DTIME_WITH_SYS_TIME=1 \
-DHAVE_BCOPY = 1 -DHAVE_STRERROR = 1 -DRETSIGTYPE = vacío -DRETSIGVAL =
-DHAVE_SIGSET=1 -DDECLWAITSTATUS=int -DSTD_HEADERS=1 \
-DARPDIR="/usr/local/arpwatch/" \
-DPATH_SENDMAIL="/usr/sbin/sendmail" \
-l\linux-include -o arpsnmp \
arpsnmp.o db.o dns.o \
ec.o archivo.o intoa.o \
```

machdep.o util.o informe.o setsignal.o versión.o

report.o: En la función 'informe':

report.o(.text+0x409): el uso de 'mktemp' es peligroso, mejor usa 'mkstemp'

Una vez compilado, instala ARPWatch con el siguiente comando:

hacer instalar

ARPWatch se instalará (por defecto) en /usr/local/sbin. Este directorio es usualmente en la ruta de la raíz , pero si escribe arpwatch y recibe un error de comando no encontrado , probablemente necesite anteponer el comando con su directorio, así:

/usr/local/bin/arpwatch

A medida que se ejecuta ARPWatch, informará al demonio SYSLOG sobre las nuevas direcciones MAC encontradas en la red. Esto significa que ARPWatch generalmente generará una salida a /var/log/messages, por lo que puede ejecutar un comando grep para conocer los nuevos hosts a medida que ARPWatch los encuentra:

grep arpwatch/var/log/mensajes

ARPWatch también enviará un correo electrónico a la cuenta raíz en un sistema que detalla los nuevos hosts. El correo electrónico contiene detalles como la fecha, la dirección IP y la dirección MAC:

```
nombre de host: cliente.ejemplo.com
dirección IP: 192.168.1.10
dirección ethernet: 0:e1:18:34:2f:e8
proveedor de ethernet: <desconocido>
marca de tiempo: sábado, 22 de mayo de 2004 11:25:59 -0500
```


En ambas formas, es posible saber virtualmente al instante cuando aparece un nuevo host en la red. Dicha información sería útil para el administrador de seguridad al monitorear el posible uso no autorizado de una red.

ARPWatch se ejecutará en segundo plano como un demonio, en silencio (o idealmente en silencio) yendo sobre su negocio y le informará según sea necesario. Si, por alguna razón, ARPWatch se apaga, tal vez porque la máquina se reinició, las entradas existentes se escribirán en un archivo llamado arp.dat (la ubicación de este archivo varía mucho; si necesita encontrarlo, ejecute `find / -name "arp.dat"`). Si necesita restablecer la base de datos de monitoreo de ARPWatch para que "recupere" todos los hosts en la red nuevamente, ejecute estos comandos desde el directorio en el que ubica ARPWatch:

```
rm arp.dat  
toque arp.dat
```

Un consejo sobre el uso de ARPWatch: asegúrese de que el archivo de datos de ARPWatch, arp.dat, se controle en busca de cambios no autorizados. Si un atacante puede alterar este archivo y agregar su propia entrada manualmente, ARPWatch no lo alertará sobre la presencia del nuevo host. Asegúrese de que el archivo arp.dat sea monitoreado por AIDE (tratado en el Capítulo 14, "Integridad del sistema de archivos") o por otros medios similares.

Resumen

Este capítulo le mostró algunas de las herramientas utilizadas en la detección de intrusos. El objetivo era brindarle algo de experiencia práctica basada en los conceptos presentados en los capítulos anteriores. Aprendió sobre rastreadores de red en este capítulo y se centró específicamente en TCPDump. Algunos paquetes y tipos de ataques también se vieron a través de los ojos de TCPDump.

Otras herramientas presentadas y discutidas en este capítulo incluyeron Snort, que proporciona un excelente sistema de detección de intrusos. Finalmente, también se discutió el uso de ARPWatch para monitorear entradas ARP nuevas e inesperadas en la red.

El siguiente capítulo analiza la integridad del sistema de archivos a través de los ojos de AIDE, un verificador de integridad del sistema de archivos.

Esta página se dejó en blanco intencionalmente

Integridad del sistema de archivos

La integridad es uno de los tres principios de seguridad informática comúnmente utilizados; confidencialidad y disponibilidad son los otros dos. En el sentido más puro de los tres principios, la integridad simplemente se refiere a los medios por los cuales usted se asegura de que los datos sean auténticos y no hayan sido alterados o alterados de ninguna manera. Un aspecto de garantizar la integridad de los datos es garantizar la integridad del sistema en el que se alojan los datos.

Este capítulo analiza algunos medios muy específicos que tiene a su disposición cuando ejecuta Linux para garantizar la integridad de los datos. Estos incluyen examinar los archivos en un sistema Linux para asegurarse de que no hayan sido alterados sin su conocimiento y buscar anomalías que puedan indicar la presencia de un intruso en el sistema.

Integridad del sistema de archivos definida

Mantener la integridad del sistema es otra capa más de seguridad destinada a brindarle a usted, el administrador de seguridad, una sensación cálida y confusa. A los efectos de este capítulo, el término *integridad* del sistema de archivos se refiere al conocimiento verificable de que el sistema informático y los objetos que contiene se encuentran en buen estado. Aunque esa es una definición amplia, la integridad del sistema de archivos en este capítulo simplemente implicará la verificación de que los archivos ubicados en la computadora no hayan sido manipulados o alterados. Como tal, este capítulo se concentra en las herramientas para ayudarlo a verificar los archivos.

Integridad práctica del sistema de archivos

Hay varias herramientas disponibles para verificar la integridad de los archivos en el sistema. En este capítulo, mostraré AIDE, el entorno de detección de intrusos avanzado. AIDE es una herramienta de verificación de integridad del sistema de archivos de código abierto.

Una verificación de integridad básica de un archivo generalmente implica obtener valores de suma de verificación de los archivos en la computadora y comparar esas sumas de verificación con valores buenos conocidos. Las sumas de control a veces también se denominan valores hash o firmas. Las comprobaciones más complejas se realizan con herramientas como AIDE, como verá más adelante en este capítulo.

Las sumas de verificación se utilizan con frecuencia para verificar la integridad de un archivo descargado. Por ejemplo, muchos repositorios FTP de Linux contienen un archivo llamado sha1sums. Dentro de ese sha1sums file son las sumas de comprobación de los archivos tal como residen en el servidor FTP. cuando descargas

el archivo, luego puede verificar la suma de verificación contra el archivo descargado. Si el valor de su suma de verificación coincide con la suma de verificación en el servidor, sabrá que tiene un buen archivo. Si los valores no coinciden, algo salió mal con la descarga y puede ahorrar algo de tiempo en lugar de intentar trabajar con un archivo corrupto o desperdiciar un CD-R.

Un ejemplo práctico sería útil. Salta a una consola y escribe lo siguiente:

```
sha1sum /etc/contraseña
```

Verá un valor como este:

```
dbf758aecfc31b789336d019f650d404fc280d64 /etc/contraseña
```

Tenga en cuenta que su valor será sustancialmente diferente al mío, a menos que esté ejecutando el comando contra mi archivo de contraseña, en cuyo caso tengo otros problemas que requieren atención.

Si agrega un usuario, elimina un usuario o realiza cualquier cambio que afecte el archivo de contraseña, ese valor de sha1sum cambiará. Por ejemplo, si realiza un cambio en el nombre de alguien dentro del archivo passwd, el sha1sum del archivo passwd cambiará porque el contenido del archivo ahora es diferente. Continuando con el ejemplo anterior, puede cambiar el nombre del usuario root ejecutando esto (como root):

```
raíz chfn
```

Se le presentarán varias opciones para cambiar la información de la cuenta del usuario, comenzando con el nombre completo. Cambie el valor de Nombre completo a lo que desee y continúe con los cambios a otros valores si lo desea. Ahora, ejecutar un sha1sum contra /etc/passwd mostrará una suma de verificación diferente para el archivo:

```
sha1sum /etc/contraseña  
a22e91a7bb7a21ca6c2b9d4f32e03f4ed3eec37 /etc/contraseña
```

Instalando AIDE

AIDE es una herramienta de verificación de la integridad del sistema de archivos que ofrece muchas de las funciones que esperaría de un programa de este tipo. Más información sobre AIDE, incluidos enlaces para descargar, está disponible en <http://aide.sourceforge.net>.

Al igual que con otras herramientas presentadas en el libro, AIDE está disponible como paquete para las distribuciones de Linux más populares, o se puede descargar y compilar. Sin embargo, probablemente también necesitará algunos requisitos previos antes de intentar compilar AIDE. Si este es el caso, el script de configuración para AIDE le informará sobre estos requisitos previos, que luego deberá descargar y compilar (o instalar desde los paquetes del sistema operativo) antes de continuar con la compilación de AIDE. La compilación de AIDE sigue el mismo patrón que la compilación de otro software, como esta en Linux:

```
tar -zxvf ayudante- <NNNN> .tar.gz  
ayudante de cd- <NNNN>  
./configurar  
hacer  
hacer instalar
```

El resto de este capítulo muestra el uso de AIDE compilado en lugar de una versión envejecida preempaquetada; las rutas y los comandos específicos pueden ser ligeramente diferentes si ejecuta una versión empaquetada. Los conceptos subyacentes siguen siendo los mismos.

Configuración de AIDE

AIDE, como muchas otras aplicaciones de Linux, opera usando un archivo de configuración. El archivo de configuración está basado en texto y contiene información que el programa usa para determinar las características que usará cuando se ejecute. La primera vez que ejecute AIDE, creará e inicializará la base de datos que se utilizará para futuras comprobaciones de la integridad del sistema de archivos. Luego, esa base de datos se verifica manualmente para verificar su cordura, y ejecutará un proceso de actualización que se utilizará a partir de ese momento para buscar cambios que ocurran en el sistema de archivos.

Creación de un archivo de configuración AIDE

Después de instalar AIDE, lo primero que querrá hacer es crear un archivo de configuración. El archivo de configuración de AIDE normalmente se llama `aide.conf` y se encuentra en `/etc/` o `/etc/aide/` para una versión empaquetada de AIDE. Los comentarios dentro del archivo de configuración de AIDE comienzan con un signo de almohadilla (`#`). Hay tres categorías de líneas dentro del archivo de configuración de AIDE: líneas de configuración, líneas de macro y líneas de selección. En Debian, la configuración de AIDE se encuentra en `/etc/aide` y se divide en varios directorios, incluido un archivo genérico `aide.conf`, así como un directorio que contiene reglas específicas y otro directorio que contiene configuraciones.

El corazón del archivo de configuración de AIDE son las líneas de selección que utiliza para disuadir minar qué objetos en el sistema de archivos serán monitoreados. Las líneas de configuración también son importantes para determinar cómo funcionará AIDE, y las líneas macro son importantes para crear configuraciones avanzadas. AIDE usa una serie de directivas `parámetro=valor` para indicar el tipo de verificación a realizar en un objeto dado. La tabla 14.1 enumera algunas de esas directivas.

AIDE también permite al administrador crear grupos personalizados que contengan los grupos predeterminados. Hacerlo puede ahorrarle tiempo y mejorar la legibilidad del archivo de configuración. Puede usar un grupo personalizado para combinar otros grupos de cheques de uso común. Por ejemplo, crear un grupo llamado `MiGrupo` con tipos de cheques de uso común es tan simple como esto:

```
MiGrupo p + i + n + m + md5
```

Estas agrupaciones, ya sean predeterminadas o personalizadas, se utilizan para determinar el tipo de verificación que se realizará en una selección determinada. También configura los archivos y directorios que se comprobarán utilizando una línea de selección en el archivo de configuración. Las líneas de selección consisten en el objeto a verificar junto con el tipo de verificación a realizar. El objeto puede ser un archivo, un directorio, una expresión regular o, más comúnmente, una combinación de un archivo junto con alguna sintaxis de expresión regular. Echaré un vistazo a las expresiones regulares en una sección posterior, pero por ahora mostraré ejemplos simples del proceso de selección.

Tabla 14.1 Directrices de configuración de AYUDA

Directiva	Descripción
<code>perm</code>	permisos
<code>l</code>	inodo
<code>encl</code>	Número de enlaces
<code>tu</code>	Usuario
<code>gr</code>	Grupo
<code>s</code>	Tamaño
<code>B</code>	Recuento de bloques
<code>mtime</code>	mtime
<code>a</code>	un momento
<code>c</code>	ctime
tipo de letra	Tipo de archivo
<code>S</code>	Verifique el tamaño creciente
<code>sha1</code>	suma de comprobación sha1
<code>sha256</code>	suma de comprobación sha256
<code>sha512</code>	suma de comprobación sha512
<code>rmd160</code>	suma de comprobación rmd160
<code>Tigre</code>	suma de control de tigre
<code>R</code>	<code>p+i+n+u+g+s+m+c+md5</code>
<code>l</code>	<code>p+i+n+u+g</code>
<code>Y</code>	grupo vacío
<code>></code>	Archivo de registro creciente <code>p+u+g+i+n+S</code>
<code>haval</code>	suma de comprobación haval
<code>huésped</code>	suma de verificación de invitado
<code>crc32</code>	suma de comprobación crc32

La siguiente línea de selección examinaría todo en el directorio `/etc`, específicamente observando la cantidad de enlaces, el usuario propietario de un archivo determinado, el grupo propietario de un archivo determinado y el tamaño del archivo:

```
/etc n+u+g+s
```

Un cambio en uno de esos atributos que ocurre inesperadamente podría indicar una manipulación. En g. El siguiente ejemplo utiliza un grupo personalizado llamado `MyGroup` como verificación de los archivos dentro del directorio `/bin`:

```
/bin Mi Grupo
```

Los objetos pueden ignorarse u omitirse usando un signo de exclamación (!), como en el siguiente ejemplo, lo que hace que AIDE ignore todo en `/var/log`:

```
!/var/registro/*
```

Ignorar objetos que cambian con frecuencia puede reducir drásticamente el número de elementos irrelevantes. líneas que aparecen en el informe AIDE. Sin embargo, debe tener cuidado de no ignorar demasiado; de lo contrario, es posible que se pierda cambios importantes en el sistema de archivos.

Las líneas de regla en el archivo de configuración usan expresiones regulares para habilitar una coincidencia poderosa capacidades. No se preocupe si no está familiarizado con la magia negra involucrada en las expresiones regulares; Seré fácil contigo aquí.

Una preocupación principal con los archivos coincidentes en AIDE es que no deja espacio para que un atacante eluda el verificador de integridad de archivos. Esto podría ocurrir si especificó un nombre de archivo sin calificar completamente el archivo. Por ejemplo, si desea omitir la verificación de un archivo en el directorio `/var/log/` porque cambia, puede usar esta sintaxis (aparentemente correcta):

```
!/var/log/maillog
```

Sin embargo, debido a la coincidencia de expresiones regulares que se produce, un atacante podría crear un archivo llamado así:

```
/var/log/maillog.crack
```

Debido a que ya excluyó `/var/log/maillog`, AIDE no verificará nada que comience con `/var/log/maillog`. Para resolver este problema, agregue un signo de dólar (\$) al final del archivo. En la sintaxis de expresiones regulares, un \$ indica el final de la línea. Por lo tanto, al cambiar la sintaxis del archivo que desea excluir y agregar \$, utiliza la coincidencia más específica para esa verificación del sistema de archivos:

```
!/var/log/maillog$
```

Por defecto, AIDE creará una base de datos basada en archivos llamada `aide.db.new`. Luego, este archivo se mueve (manualmente) para futuras comprobaciones. Por lo tanto, realmente no hay necesidad de alterar este comportamiento dentro del contexto del archivo de configuración; sin embargo, ciertamente puede cambiar la ruta y el nombre de este archivo usando las opciones de configuración:

```
base de datos=archivo:<nombre de archivo>
database_out=archivo:<nombre de archivo>
```

AIDE también puede usar un servidor de base de datos SQL como PostgreSQL para almacenar el contenido de la base de datos, aunque esa configuración está más allá del alcance de este libro.

Un archivo de configuración de muestra de AIDE

Como mínimo, debe decirle a AIDE qué partes del sistema de archivos debe verificar y qué reglas usar para esas comprobaciones. También puede agregar muchos otros bits a la configuración para modificar el rendimiento de AIDE. Para los propósitos de esta sección, mostraré un archivo de configuración muy básico con la advertencia de que debe agregarlo si lo considera necesario para su instalación de Linux.

Si ha compilado AIDE, abra el archivo `/usr/local/etc/aide.conf`. Si el archivo no existe, créelo. Coloque las siguientes líneas dentro del archivo:

```
/bin R
/sbin R
/etc R+a
/lib R
/usr/libR
```

Si está utilizando una versión empaquetada de una distribución como Debian, la configuración ya se proporcionan varios scripts de contenedor, por lo que puede omitir este paso con seguridad.

Inicializando la base de datos AIDE

Con un archivo de configuración rápido y básico en la mano, es hora de inicializar la base de datos de AIDE. Este proceso puede demorar un tiempo variable según la cantidad de archivos que esté revisando y la cantidad de recursos que tenga disponible la computadora. Inicializar la base de datos AIDE es tan simple como ejecutar lo siguiente para la versión compilada:

```
/usr/local/bin/ayudante --init
```

Alternativamente, en Debian con una versión empaquetada, debe ejecutar

```
ayudante
```

AIDE ahora inicializará la base de datos según los criterios que eligió en el archivo de configuración. Cuando esté completo, verá un mensaje similar a este:

```
AYUDA, versión 0.15.1
```

```
### Base de datos AIDE inicializada.
```

El siguiente paso es cambiar el nombre (mover) la base de datos recién creada a `aide.db` para que se convierta en la base de datos maestra o predeterminada:

```
mv /usr/local/etc/aide.db.new /usr/local/etc/aide.db
```

Ahora debería poder ejecutar una verificación de la base de datos para verificar que todo funciona bien:

```
/usr/local/bin/ayudante --verificar
```

Si todo va bien, verá un resultado similar al siguiente:

```
AYUDA, versión 0.15.1
```

```
### Todos los archivos coinciden con la base de datos AIDE. ¡Se ve bien!
```

Con la base de datos AIDE inicializada, debe copiar inmediatamente la base de datos en un disco, preferiblemente en un medio de solo lectura, como un CD-R, o debe copiarla de manera segura en otra computadora. Si deja la base de datos AIDE en la computadora, los atacantes pueden simplemente alterar la base de datos AIDE para cubrir sus huellas después de reemplazar los archivos del sistema con los suyos. Cada vez que actualice la base de datos AIDE desde este punto en adelante, siempre debe copiar el archivo de la base de datos resultante en un medio seguro.

Programación de AIDE para que se ejecute automáticamente

AIDE se ejecuta mejor usando un trabajo cron (tarea programada). Por lo tanto, debe programar AIDE para que se ejecute automáticamente sin su intervención. Por lo general, AIDE se ejecuta una vez al día, pero debe programarlo para que se ejecute de acuerdo con su política de seguridad. El método más fácil y rápido para que AIDE se ejecute diariamente es crear una entrada crontab .

Crear una entrada crontab es cuestión de ejecutar esto (como root):

crontab -e

Para ejecutar AIDE todas las noches a las 2:00 am, ingrese la siguiente línea en crontab:

```
0 2 * * * /usr/local/bin/ayudante --verificar
```

Para obtener más información sobre el formato de las entradas crontab , consulte la documentación de su distribución.

Monitoreo AIDE para cosas malas

Bien, ya tiene esta nueva y brillante herramienta de verificación de integridad del sistema de archivos configurada y funcionando. Pero ahora que? Ahora siéntate y espera a que suceda algo. Por lo general, nada sucede, e incluso cuando parece que algo malo podría haber sucedido, muchas veces no es así.

AIDE continuará monitoreando el sistema de archivos de acuerdo con las reglas que configuró.

Gracias al trabajo cron, recibirá informes todas las noches que contienen los archivos y los atributos de aquellos archivos que han cambiado desde que se inicializó la base de datos o se actualizó por última vez.

Muchas veces estos cambios serán completamente benignos. Recuerde el ejemplo del principio del capítulo. Si agrega un usuario, los archivos como /etc/passwd y /etc/shadow cambiarán. AIDE lo notará e informará en consecuencia, asumiendo que está revisando /etc.

Sin embargo, si no agregó un usuario ni realizó otros cambios en /etc/passwd o /etc/shadow, puede examinar más de cerca para asegurarse de que un atacante no haya alterado ninguno de esos archivos importantes.

Por supuesto, hay otros archivos sobre los que AIDE estará informando. Debe monitorear de cerca el informe AIDE en busca de archivos que se modificaron inesperadamente. Por ejemplo, los archivos /bin/su o /usr/bin/passwd rara vez deben modificarse y solo mediante ciertas actualizaciones de software conocidas. Por lo tanto, si un archivo como /bin/su aparece en un informe de AIDE, debe examinarlo de inmediato para ver cuándo y por qué cambió ese archivo. Llevando este ejemplo un paso más allá, suponga por un momento que AIDE ejecutó su proceso normal de la noche a la mañana. Por la mañana, te despiertas y encuentras un correo electrónico que contiene algunas de las siguientes líneas:

¡AIDE 0.15.1 encontró diferencias entre la base de datos y el sistema de archivos!
 Marca de tiempo de inicio: 2014-07-31 23:50:17

Resumen:

Número total de archivos:	16112
Archivos agregados:	0
Archivos Eliminados:	0
Archivos modificados:	1

Archivos modificados:

cambiado: /etc/adjtime

Información detallada sobre los cambios:

Archivo: /etc/adjtime

Un momento : 2014-07-11 05:43:38 , 2014-07-31 23:36:06

El AIDE chequeo que usted programó encontró algo. Usted puede saber rápidamente de la líneas de resumen de lo encontrado:

Resumen:

Número total de archivos:	16112
Archivos agregados:	0
Archivos Eliminados:	0
Archivos modificados:	1

A continuación, verá un resumen un poco más detallado de los archivos que se agregaron, cambiaron o eliminaron desde que se inicializó o actualizó la base de datos por última vez. En este caso, solo se ha modificado un archivo:

Archivo: /etc/adjtime

Un momento : 2014-07-11 05:43:38 , 2014-07-31 23:36:06

Según este informe, es fácil ver que algo cambió con /etc/adjtime.

Limpieza de la base de datos de AIDE

Con el tiempo, notará que los informes de verificación de AIDE se vuelven más y más largos. Esto suele ser el resultado de la actividad normal en el servidor, como agregar y eliminar usuarios, actualizar el software y cambiar la configuración en los archivos de configuración. Debe actualizar regularmente la base de datos de AIDE no solo para acortar los informes, sino también para realizar un mejor seguimiento cuando se produzcan cambios inesperados. Si no actualiza regularmente la base de datos de AIDE, es posible que se pierda un cambio que resultó de un ataque.

Usted puede estar preguntando, "¿Con qué frecuencia debo actualizar la base de datos de AIDE?" La respuesta depende en gran medida de sus necesidades y de su política de seguridad. Cuando comience a usar AIDE por primera vez, espero que deba actualizar la base de datos al menos durante las primeras ejecuciones (nuevamente, dependiendo de su política de seguridad) y, lo que es más importante, refinar el archivo de configuración. Descubrirá que ciertos archivos cambian con tanta frecuencia que necesita excluirlos por completo o cambiar los tipos de comprobaciones que se realizan en esos archivos.

Es mucho mejor cambiar los tipos de cheques que simplemente omitir los archivos por completo. Algunos atributos de archivo que AIDE puede verificar no cambiarán con frecuencia o en absoluto para el mismo archivo. Los atributos como inode y ctime no deberían cambiar. Por lo tanto, si observa que

Ciertos archivos siguen apareciendo en el informe de AIDE y ha descartado actividad nefasta, debe cambiar el tipo de verificación que se realiza en ese archivo dentro del archivo de configuración de AIDE.

Un archivo que cambia regularmente en algunos sistemas es el archivo de contraseñas de Samba, `/etc/samba/smbpasswd`. En tales sistemas, el archivo aparece regularmente en el informe donde todo en el directorio `/etc/` se examina utilizando la verificación R (consulte la Tabla 14.2). Un tipo de verificación más apropiado para este archivo podría incluir elementos que no cambian con frecuencia, como inodo y tiempo. Tal verificación parecería así en el archivo de configuración de AIDE:

```
/etc/samba/smbpasswd$ c+i
```

Tenga en cuenta el uso de `$` al final del nombre de archivo en el ejemplo para indicar el final de la línea.

A medida que se ejecuta el informe AIDE, podrá usar más granularidad para refinar los archivos que se comprueban y los propios controles. Después de actualizar el archivo de configuración de AIDE, deberá actualizar la base de datos para que los cambios surtan efecto. Este proceso se logra ejecutando este comando:

```
/usr/local/bin/aide --actualizar
```

Una vez completada la actualización, tendrá un nuevo archivo de base de datos, `/usr/local/etc/aide.db.nuevo`, por defecto. Este archivo debe moverse para sobrescribir la base de datos existente:

```
mv /usr/local/etc/aide.db.new /usr/local/etc/aide.db
```

Ahora ejecutar `aide --check` dará un resultado limpio:

```
AYUDA, versión 0.15.1
```

```
### Todos los archivos coinciden con la base de datos AIDE. ¡Se ve bien!
```

Después de actualizar la base de datos, debe copiar el archivo en un medio seguro o en otra computadora para garantizar la integridad de la base de datos.

Con el archivo de configuración y la base de datos de AIDE actualizados y AIDE programado para ejecutarse todas las noches, ahora tiene una infraestructura para verificar la integridad de su sistema de archivos. Desde aquí, puede seguir leyendo para obtener información sobre configuraciones más avanzadas para AIDE, o puede ir al Capítulo 12 para obtener información sobre la herramienta de verificación de rootkit llamada Chkrootkit.

Cambiar la salida del informe AIDE

Es posible que desee un poco más de flexibilidad en la ubicación del informe AIDE. Por ejemplo, es posible que no desee recibir correos electrónicos si todo está bien con el informe AIDE, o puede desear que el informe AIDE se incluya en un archivo en lugar de proporcionar una salida estándar. AIDE tiene cuatro opciones básicas para configurar la salida que se pueden configurar a través del archivo de configuración de AIDE.

Flujos de salida de Linux

Linux tiene tres flujos genéricos de salida que se crean cuando se ejecuta un programa. Estos flujos se denominan STDIN, STDOUT y STDERR, que son abreviaturas de Entrada estándar, Salida estándar y Error estándar, respectivamente. Cuando ve una referencia a STDOUT, se refiere al método normal de salida a la pantalla y STDERR indica salida como resultado de una condición de error. Como era de esperar, STDIN se refiere al método de entrada cuando se lee desde el descriptor del archivo de entrada.

La opción de configuración general de AIDE llamada `report_url` configura cómo se muestra la salida. De forma predeterminada, la salida se muestra en STDOUT. La salida se puede mostrar para cualquiera o todos los siguientes:

• `Y SALIDAESTÁNDAR` (predeterminado)

• `Y STDERR`

• `Y archivo de texto`

• `Y Descriptor de archivo`

De estas cuatro posibilidades, STDOUT, STDERR y archivo de texto son de interés. Las versiones futuras de AIDE pueden incluir configuraciones de salida para correo electrónico automatizado y salida automatizada a la instalación de syslog.

De particular interés es el tipo de salida de archivo de texto para AIDE. Este tipo de salida se especifica utilizando esta línea de configuración:

```
report_url=archivo:/<ruta>/<nombre de archivo>
```

Por ejemplo, para configurar los informes de AIDE para ir a un archivo llamado `aidereport.txt` en el directorio `/var/log/aide` que usted crea, usaría esta opción de configuración en el archivo de configuración de AIDE:

```
informe_url=archivo:/var/log/help/helpreport.txt
```

Sin embargo, la opción de configuración `report_url` es solo un medio para salir en un archivo. Debido a que está ejecutando el informe AIDE desde cron, también podría simplemente redirigir la salida a un archivo. Por ejemplo, recuerde la entrada crontab que se mostró anteriormente en este capítulo:

```
0 2 * * * /usr/local/bin/ayudante --verificar
```

Puede modificar esa entrada cron para redirigir la salida a un archivo. Si lo hace, toda la salida iría a ese archivo y también habilitaría funciones adicionales, como la asignación de nombres basada en la fecha. Esto se puede hacer con un pequeño truco de shell usando comillas de ejecución (a veces llamado backtick, generalmente se encuentra con la tilde `[~]` en el teclado). Aquí está la nueva entrada de cron:

```
0 2 * * * /usr/local/bin/help --check >/var/log/help/helpreport-'date +%m%d%H' Y.txt
```

Ahora el informe AIDE se ejecutará y redirigirá STDOUT a un archivo llamado

```
/var/log/aide/aidereport-<fecha>.txt
```

Por ejemplo, para un informe ejecutado el 12 de marzo de 2014, el archivo se llamaría

```
/var/log/aide/aidereport-03122014.txt
```

Con una configuración redirigida como la que se muestra, ya no recibirá correos electrónicos cuando AIDE ejecute su trabajo cron normal. Más bien, recibirá correos electrónicos solo cuando ocurra un error con el trabajo cron de AIDE. Debido a que ya no recibirá los correos electrónicos, es posible que se sienta tentado a ignorar sus deberes de monitoreo y simplemente dejar que se acumulen todos los informes de AIDE. Sin embargo, aún debe monitorear los informes de AIDE mirándolos en busca de anomalías y limpiándolos según corresponda.

Obtener una salida más detallada

Los informes de AIDE se pueden configurar con verbosidad adicional. Agregar verbosidad a AIDE es valioso cuando está solucionando problemas de coincidencia de reglas. Por ejemplo, cuando establece la opción de configuración detallada, podrá ver cómo AIDE construye la lista de archivos para verificar.

Si ve resultados inesperados o si los archivos se incluyen o excluyen por motivos misteriosos, será útil agregar esta opción a la configuración o agregarla como una opción de línea de comandos.

La opción de configuración para agregar verbosidad es la siguiente:

```
detallado=<N>
```

En este caso, <N> es un número entero positivo con un valor máximo de 255. En la práctica, solo los números superiores a 200 brindan una salida de depuración adicional para la mayoría de las comprobaciones. Por lo tanto, para agregar el nivel máximo de verbosidad, usaría esta opción de configuración:

```
detallado=255
```

Con esta configuración establecida, verá muchos resultados adicionales durante una ejecución de AIDE:

```
Manejo de / con s "/bin" con nodo "/"
Manejo de / con s "/sbin" con nodo "/"
Manejo de / con s "/etc" con nodo "/"
Manejo de / con s "/lib" con nodo "/"
Manejo de /usr con s "/usr/lib" con el nodo "/usr"
árbol: "/"
2 ^/caja
3 ^/sbin
4 ^/etc.
5 ^/lib
árbol: "/usr"
6 ^/usr/lib
```

AYUDA, versión 0.15.1

```
### Todos los archivos coinciden con la base de datos AIDE. ¡Se ve bien!
```

La salida es mucho más detallada (como era de esperar) e incluye las funciones que se llaman dentro del propio programa AIDE, así como detalles sobre los archivos que AIDE está revisando mientras los revisa. El uso de esta salida puede ser invaluable cuando intenta solucionar un problema con su configuración de AIDE.

Definición de macros en AIDE

Las macros se utilizan en AIDE para definir objetos de uso común y objetos que se utilizarán como variables en un archivo de configuración de AIDE. Puede usar una macro AIDE para definir el directorio de nivel superior que se usará dentro de la configuración. Luego, usaría esta macro dentro de las líneas de selección y se sustituiría como una variable en esa línea de selección. También puede usar una macro para establecer una variable según ciertos criterios. Luego, las macros se pueden usar dentro de la configuración de AIDE dentro de las estructuras de control (bloques de código de decisión) para alterar la configuración de AIDE en función del resultado de la estructura de control.

Las macros se definen utilizando la siguiente sintaxis:

```
@@define <macro> <definición>
```

Las macros también pueden estar indefinidas con esta sintaxis:

```
@@undef <macro>
```

Las macros se utilizan dentro de la configuración con la siguiente sintaxis:

```
@@{<macro>}
```

Un ejemplo del uso de una macro de forma sencilla es crear una macro para una jerarquía de directorios compleja para que no tenga que escribirla en el archivo de configuración varias veces.

Suponga que tiene una cierta estructura de directorios en la computadora que desea definir usando una macro:

```
@@define DIRBASE /usr/src/linux
```

Esta macro podría luego usarse más adelante dentro de la configuración de selecciones para AIDE:

```
@@{DIRBASE}/.config R  
!@@{DIRBASE}/doc
```

Un uso más potente de las macros consiste en cambiar la configuración en función de algunos criterios. Por ejemplo, las macros se pueden utilizar dentro de dos tipos de estructuras de control, una basada en si la macro se ha definido y la otra basada en el host desde el que se ejecuta la macro.

Se está ejecutando el programa AIDE.

Estas estructuras de control son básicamente sentencias `if/then/else`, y tienen asociaciones negaciones también. La sintaxis para determinar si se ha definido una macro es la siguiente:

```
@@ifdef <macro>
```

La negación de la evaluación de `@@ifdef` es como se muestra aquí:

```
@@ifndef <macro>
```

Para determinar el host actual, esta es la sintaxis:

```
@@ifhost <nombre de host>
```

La negación de la evaluación de @@ifhost es, como puede suponer, la siguiente:

```
@@ifnhost <nombre de host>
```

Independientemente de qué estructura de control se utilice, debe cerrarse con esta declaración:

```
@@terminara si
```

Se pueden agrupar varias estructuras de control usando un tipo de estructura else que usted esperaría para una declaración if asociada. La sintaxis es simplemente como sigue:

```
@@demás
```

Aquí hay un ejemplo de cómo se puede usar cada uno de estos. El primer ejemplo comprueba para ver si se ha definido la macro llamada SOURCE y, en caso contrario, la define:

```
@@ifndef FUENTE
@@define FUENTE /usr/src
@@terminara si
```

El segundo ejemplo analiza el nombre de host de la computadora desde la que se ejecuta AIDE y establece una macro basada en los resultados de esa verificación. Esto podría ser útil si tiene estructuras de directorios que difieren entre varios hosts y le gustaría usar solo un archivo de configuración AIDE común entre ellos. Aquí está el segundo ejemplo:

```
@@ifhost cwa
@@define LOCALBINDIR /usr/local/sbin
@@terminara si
```

Finalmente, aquí hay un ejemplo usando una declaración else :

```
@@ifhost cwa
@@define LOCALBINDIR /usr/local/sbin
@@else@@define LOCALBINDIR /usr/local/bin
@@terminara si
```

Para todos los ejemplos, recuerde que usaría estas macros más adelante dentro de la configuración usando la siguiente sintaxis:

```
@@{<macro>}
```

Los tipos de cheques AIDE

Quizás se esté preguntando acerca de los diferentes tipos de controles que puede realizar AIDE. Algunas de las comprobaciones se describen de nuevo en la Tabla 14.2. Tenga en cuenta que esta no es una lista exhaustiva y es probable que se agreguen nuevas opciones con el tiempo.

Probablemente sea útil dividir los tipos de cheques AIDE en categorías. Hay tres categorías básicas de controles AIDE: lo que denominaré controles estándar, controles agrupados y sumas de control. El tipo estándar de verificación AIDE busca información que se puede recopilar del archivo o del descriptor del archivo. Estos controles se enumeran en la Tabla 14.3.

Tabla 14.2 Tipos de cheques AIDE

Directiva	Descripción
<code>perm</code>	permisos
<code>tipo de letra</code>	Tipo de archivo
<code>I</code>	inodo
<code>enlace</code>	Número de enlaces
<code>l</code>	Nombre del enlace
<code>tu</code>	Usuario
<code>grupo</code>	Grupo
<code>S</code>	Tamaño
<code>B</code>	Recuento de bloques
<code>mtime</code>	<code>mtime</code>
<code>a</code>	un momento
<code>c</code>	<code>ctime</code>
<code>S</code>	Verifique el tamaño creciente
<code>md5</code>	suma de comprobación md5
<code>sha1</code>	suma de comprobación sha1
<code>sha256</code>	suma de comprobación sha256
<code>sha512</code>	suma de comprobación sha512
<code>rmd160</code>	suma de comprobación rmd160
<code>Tigre</code>	suma de control de tigre
<code>R</code>	<code>p+i+n+u+g+s+m+c+md5</code>
<code>I</code>	<code>p+i+n+u+g</code>
<code>Y</code>	grupo vacío
<code>></code>	Archivo de registro creciente <code>p+u+g+i+n+S</code>
<code>haval</code>	suma de comprobación haval
<code>huésped</code>	suma de verificación de invitado
<code>crc32</code>	suma de comprobación crc32

Todas estas comprobaciones estándar utilizan funciones del sistema de archivos integradas o nativas en Linux. y se puede encontrar en la entrada de inodo del archivo. Como tal, ejecutar una verificación estándar determinada requiere menos recursos que una verificación de suma de verificación. Algunas de estas comprobaciones se prestan a ciertos archivos, mientras que otras harán que el archivo aparezca en un informe casi cada vez que se ejecuta la comprobación. Por ejemplo, el `ctime` de un archivo determinado no debe cambiar a menos que el archivo se elimine o se reemplace por otro.

Puede que no sea evidente lo que realmente hacen algunas de las comprobaciones estándar. Tabla 14.4 describe cuáles pueden ser los controles más oscuros dentro de este grupo.

Por otro lado, los cheques agrupados combinan algunos de los estándares más utilizados. cheques `dard`, como se describe en la Tabla 14.5.

Tabla 14.3 Controles Estándar en AIDE

Directiva	Descripción
perm	permisos
I	inodo
links	Número de enlaces
tu	Usuario
group	Grupo
S	Tamaño
B	Recuento de bloques
mtime	mtime
a	un momento
c	ctime
S	Verifique el tamaño creciente

Tabla 14.4 Explicación de algunas verificaciones estándar

Comprobar nombre	Explicación
inodo	El inodo es una estructura de datos que contiene información sobre un archivo dado en Linux. El inodo contiene información como la ubicación del archivo, los permisos, la información del propietario y del grupo, y muchos otros bits útiles.
Número de enlaces	Los enlaces son similares a los accesos directos en el mundo de Windows. Este tipo de verificación busca ver cuántos enlaces existen para el archivo dado.
mtime	El mtime de un archivo es el momento en que se modificó por última vez.
un momento	La hora de un archivo es la hora en que se accedió por última vez al archivo.
ctime	El ctime de un archivo es el momento en que se creó el archivo.

Tabla 14.5 Cheques Agrupados en AIDE

Directiva	Definición
R	p+ftype+i+l+n+u+g+s+m+c+md5
I	p+ftipo+i+l+n+u+g
Y	grupo vacío
>	Archivo de registro creciente p+u+g+i+n+S

Tabla 14.6 Comprobaciones de Checksum en AIDE

Directiva	Definición
md5	suma de comprobación md5
sha1	suma de comprobación sha1
sha256	suma de comprobación sha256
sha512	suma de comprobación sha512
rmd160	suma de comprobación rmd160
Tigre	suma de control de tigre
haval	suma de comprobación haval
huésped	suma de verificación de invitado
crc32	suma de comprobación crc32

Finalmente, las sumas de verificación utilizan sumas de verificación criptográficas de los archivos, como se explicó anteriormente en el capítulo y se define en la Tabla 14.6.

Las diferencias en los distintos tipos de verificación de suma de verificación pueden explicarse simplemente como las diferencias en los algoritmos criptográficos utilizados para crear las sumas de verificación. Te dejo a ti que investigues más a fondo los tipos de algoritmos criptográficos que utiliza AIDE. Recomiendo *Applied Cryptography*, de Bruce Schneier, como una gran referencia para este propósito.

Resumen

Este capítulo analizó la integridad del sistema de archivos y cómo puede ayudar a la seguridad de su sistema al garantizar que los archivos no hayan sido alterados inesperadamente. El capítulo comenzó con una mirada a cómo se usan las sumas de verificación para verificar archivos. El capítulo continuó con una mirada en profundidad a un paquete de software de integridad del sistema de archivos, AIDE. Viste cómo instalar, configurar y usar AIDE en tu sistema.

IV

Apéndices

Recursos de seguridad

B Ejemplos de cortafuegos y secuencias de comandos de soporte

C Glosario

D Licencia de documentación libre GNU

Esta página se dejó en blanco intencionalmente

A

Recursos de seguridad

Esta apéndice enumera algunas fuentes comunes de avisos relacionados con la seguridad, información, herramientas, actualizaciones y parches actualmente en Internet. Existen muchos más sitios y todos los días aparecen sitios nuevos. Considere esta lista como un punto de partida, no como una lista completa. El apéndice también sirve como una sección de referencia general para este libro.

Fuentes de información de seguridad

Información de seguridad de todo tipo: avisos y alertas, libros blancos, tutoriales, etc. se puede encontrar en las siguientes fuentes:

BugTraq:

<http://www.securityfocus.com/archive/1>

Centro de Coordinación CERT:

<https://www.us-cert.gov>

Grupo de Trabajo de Ingeniería de Internet (IETF):

<http://www.ietf.org/>

Tormenta de

paquetes: <http://packetstormsecurity.org/>

Editor de RFC:

<http://www.rfc-editor.org/>

Instituto SANS:

<http://www.sans.org/>

Enfoque de

seguridad: <http://www.securityfocus.com/>

Documentos de referencia y preguntas frecuentes

Algunos documentos de referencia útiles, algunos de los cuales se citaron en el libro, se pueden encontrar en estos sitios:

“Ayuda a vencer los ataques de denegación de servicio: paso a paso”: <http://www.sans.org/dosstep/>

“Cortafuegos de Internet: Preguntas frecuentes”: <http://www.interhack.net/pubs/fwfaq/>

“Registro de número de puerto de protocolo de transporte y nombre de servicio” (IANA): <http://www.iana.org/assignments/port-numbers>

Ejemplos de cortafuegos y secuencias de comandos de soporte

A cortafuegos para un sistema independiente se describe en el Capítulo 5, "Construir e instalar un cortafuegos independiente". El ejemplo independiente está optimizado en el Capítulo 6, "Optimización del cortafuegos". El mismo ejemplo se amplía en el Capítulo 7, "Reenvío de paquetes", para que funcione como puerta de enlace o cortafuegos de estrangulamiento, con un conjunto completo de reglas de cortafuegos aplicadas tanto a la interfaz pública externa como a la interfaz de red local interna. La puerta de enlace sirve como enlace entre Internet y una red DMZ que contiene servidores públicos. El estrangulador sirve como enlace entre una LAN privada y la DMZ.

Los firewalls de muestra se presentan por partes en los Capítulos 5, 6 y 7. Este apéndice presenta los mismos ejemplos de cortafuegos que aparecerían en un script de cortafuegos.

Firewall de iptables para un sistema independiente del Capítulo 5

El Capítulo 5 cubre los protocolos de aplicación y las reglas de firewall para los tipos de servicios que es más probable que se utilicen en una caja de Linux individual e independiente. Además, se presentan reglas tanto de cliente como de servidor para servicios que no todos usarán. Esta sección presenta primero el script de iptables y luego el script de nftables .

La secuencia de comandos completa del cortafuegos de iptables , tal como aparecería en /etc/rc.d/rc.firewall o /etc/init.d/firewall, sigue:

```
#!/bin/sh

/sbin/modprobe ip_conntrack_ftp

CONEXIÓN_SEGUIMIENTO="1"
ACCEPT_AUTH="0"
SSH_SERVER="0"
SERVIDOR_FTP="0"
SERVIDOR_WEB="0"
SERVIDOR_SSL="0"
DHCP_CLIENTE="1"
```

```

IPT="/sbin/iptables"                                # Ubicación de iptables en su sistema
INTERNET="eth0"                                       # Interfaz conectada a Internet
LOOPBACK_INTERFACE="lo"                             # Sin embargo, su sistema lo nombra
IPADDR="mi.dirección.ip"                             # Su dirección IP
SUBNET_BASE="mi.subred.base"                         # Dirección base del segmento de red ISP
SUBNET_BROADCAST="my.subnet.bcast" # Dirección de difusión del segmento de red
MY_ISP="mi.isp.dirección.rango"                     # Servidor ISP y rango de direcciones NOC

NAMESERVER="isp.nombre.servidor.1"                 # Dirección de un servidor de nombres remoto
POP_SERVER="isp.pop.servidor"                      # Dirección de un servidor pop remoto
MAIL_SERVER="isp.correo.servidor"                  # Dirección de una puerta de enlace de correo remoto
NEWS_SERVER="isp.noticias.servidor"                # Dirección de un servidor de noticias remoto
TIME_SERVER="algún.tiempo.servidor"                # Dirección de un servidor de tiempo remoto
DHCP_SERVER="isp.dhcp.servidor"                    # Dirección del servidor dhcp de su ISP

LOOPBACK="127.0.0.0/8"                              # Rango de direcciones de loopback reservado
CLASE_A="10.0.0.0/8"                                # Redes privadas clase A
CLASE_B="172.16.0.0/12"                             # Redes privadas clase B
CLASE_C="192.168.0.0/16"                             # Redes privadas clase C
CLASE_D_MULTICAST="224.0.0.0/4"                     # Direcciones de multidifusión de clase D
CLASS_E_RESERVED_NET="240.0.0.0/5" # Direcciones reservadas Clase E
TRANSMITIR_SRC = "0.0.0.0"                          # Dirección de origen de la transmisión
BROADCAST_DEST = "255.255.255.255"                  # Dirección de destino de transmisión

PRIVPORTS="0:1023"                                  # Rango de puertos privilegiado y bien conocido
UNPRIVPORTS="1024:65535"                            # Rango de puertos sin privilegios

SSH_PORTS="1024:65535"

#####

# Habilitar la protección de eco de transmisión
eco 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Deshabilitar paquetes enrutados de origen
para f en /proc/sys/net/ipv4/conf/*/*accept_source_route; hacer
    eco 0 > $f
hecho

# Habilitar la protección de cookies TCP SYN
eco 1 > /proc/sys/net/ipv4/tcp_syncookies

# Deshabilitar la aceptación de redirección ICMP
para f en /proc/sys/net/ipv4/conf/*/*accept_redirects; hacer
    eco 0 > $f
hecho

# No envíe mensajes de redirección
para f en /proc/sys/net/ipv4/conf/*/*send_redirects; hacer
    eco 0 > $f
hecho

# Soltar paquetes falsificados que llegan a una interfaz que, si se responde,
# daría como resultado que la respuesta saliera de una interfaz diferente.
para f en /proc/sys/net/ipv4/conf/*/*rp_filter; hacer
    eco 1 > $f
hecho

```



```
# Registra paquetes con direcciones imposibles. para f en /proc/sys/
net/ipv4/conf/*/log_martians; haz echo 1 > $f hecho
```

```
#####
```

```
# Elimina cualquier regla existente de todas las cadenas $IPT --flush $IPT
-t nat --flush $IPT -t mangle --flush $IPT -X $IPT -t nat -X $IPT -t mangle -X
$IPT - -política ENTRADA ACEPTAR $IPT --política SALIDA ACEPTAR
$IPT --política ADELANTE ACEPTAR $IPT -t nat --política PRE-
ENRUTAMIENTO ACEPTAR $IPT -t nat --política SALIDA ACEPTAR $IPT
-t nat --política POST-ENRUTAMIENTO ACEPTAR $IPT -t mangle --policy
PREROUTING ACCEPT $IPT -t mangle --policy OUTPUT ACCEPT if [ "$1"
= "stop" ] luego echo "¡Firewall completamente detenido! ADVERTENCIA:
ESTE HOST NO TIENE FIREWALL FUNCIONANDO". exit 0 fi # Tráfico
ilimitado en la interfaz loopback $IPT -A INPUT -i lo -j ACCEPT $IPT -A
OUTPUT -o lo -j ACCEPT
```

```
# Establecer la política predeterminada para descartar
```

```
$IPT --política ENTRADA DROP
$IPT --política CAÍDA DE SALIDA
$IPT --política FORWARD DROP
```

```
#####
```

```
# Exploraciones sigilosas y banderas de estado de TCP
# Todos los bits se borran
```

```
$IPT -A INPUT -p tcp --tcp-flags TODO NINGUNO -j DROP
# SYN y FIN están configurados
```

```
$IPT -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
# SYN y RST están configurados
```

```
$IPT -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
# FIN y RST están configurados
```

```
$IPT -A INPUT -p tcp --tcp-flags FIN,RST FIN,RST -j DROP
# FIN es el único bit establecido, sin el esperado ACK que lo acompaña
```

```
$IPT -A INPUT -p tcp --tcp-flags ACK,FIN FIN -j DROP
```

```
# PSH es el único conjunto de bits, sin el esperado ACK que lo acompaña
```

```
$IPT -A ENTRADA -p tcp --tcp-flags ACK,PSH PSH -j DROP
```

```
# URG es el único conjunto de bits, sin el ACK que se espera que lo acompañe
```

```
$IPT -A INPUT -p tcp --tcp-flags ACK,URG URG -j DROP
```

```
#####
```

```
# Usar el estado de conexión para omitir la verificación de reglas
```

318 Apéndice B Ejemplos de cortafuegos y secuencias de comandos de soporte

```

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A ENTRADA -m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR
    $IPT -A SALIDA -m estado --estado ESTABLECIDO, RELACIONADO -j ACEPTAR

    $IPT -A INPUT -m estado --estado NO VÁLIDO -j REGISTRO \ --log-prefix "Entrada
        NO VÁLIDA:"
    $IPT -A INPUT -m estado --estado NO VÁLIDO -j DROP

    $IPT -A SALIDA -m estado --estado NO VÁLIDO -j REGISTRO \ --log-prefix "Salida
        NO VÁLIDA:"
    $IPT -A SALIDA -m estado --estado NO VÁLIDO -j DROP
ser

#####
# Suplantación de identidad de la dirección de origen y otras direcciones incorrectas

# Rechazar paquetes falsificados que pretenden ser de # la dirección IP de la
interfaz externa
$IPT -A ENTRADA -i $INTERNET -s $IPADDR -j DROP

# Rechazar paquetes que afirman ser de una red privada Clase A
$IPT -A ENTRADA -i $INTERNET -s $CLASE_A -j DROP

# Rechazar paquetes que afirman ser de una red privada Clase B
$IPT -A ENTRADA -i $INTERNET -s $CLASE_B -j DROP

# Rechazar paquetes que afirman ser de una red privada Clase C
$IPT -A ENTRADA -i $INTERNET -s $CLASE_C -j DROP
# Rechazar paquetes que afirman ser de la interfaz loopback
$IPT -A ENTRADA -i $INTERNET -s $LOOPBACK -j DROP

# Rechazar paquetes de difusión mal formados
$ IPT -A ENTRADA -i $ INTERNET -s $ BROADCAST_DEST -j REGISTRO
$ IPT -A ENTRADA -i $ INTERNET -s $ BROADCAST_DEST -j DROP

$IPT -A ENTRADA -i $INTERNET -d $BROADCAST_SRC -j REGISTRO
$IPT -A ENTRADA -i $INTERNET -d $BROADCAST_SRC -j DROP

si [ "$DHCP_CLIENT" = "0" ]; luego
    # Rechazar transmisiones dirigidas
    # Se utiliza para mapear redes y en ataques de denegación de servicio
    $IPT -A ENTRADA -i $INTERNET -d $SUBRED_BASE -j DROP
    $IPT -A ENTRADA -i $INTERNET -d $SUBNET_BROADCAST -j DROP

    # Rechazar transmisiones limitadas
    $ IPT -A ENTRADA -i $ INTERNET -d $ BROADCAST_DEST -j DROP
ser

# Rechazar direcciones de multidifusión Clase D # ilegal como
dirección de origen
$IPT -A ENTRADA -i $INTERNET -s $CLASS_D_MULTICAST -j DROP

$IPT -A ENTRADA -i $INTERNET ! -p UDP -d $CLASS_D_MULTICAST -j DROP

$IPT -A ENTRADA -i $INTERNET -p udp -d $CLASE_D_MULTICAST -j ACEPTAR

```

```
# Rechazar direcciones IP reservadas de Clase E
$IPT -A ENTRADA -i $INTERNET -s $CLASS_E_RESERVED_NET -j DROP

si [ "$DHCP_CLIENT" = "1" ]; luego
    $IPT -A INPUT -i $INTERNET -p udp \-s $
        BROADCAST_SRC --sport 67 \-d $
        BROADCAST_DEST --dport 68 -j ACEPTAR
ser

# rechazar direcciones definidas como reservadas por la IANA - No se pueden
Vincular redes locales # 192.0.2.0/24 que usan unilateralmente con DHCP # 0.*.* # 169.254.0.0/16 -

- RED DE PRUEBA

$IPT -A INPUT -i $INTERNET -s 0.0.0.0/8 -j DROP $IPT -A INPUT -i $INTERNET
-s 169.254.0.0/16 -j DROP $IPT -A INPUT -i $INTERNET -s 192.0.2.0/24 -j CAÍDA

#####
# Servidor de nombres DNS

# Servidor de nombres de reenvío de DNS o solicitudes de clientes

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p udp \-s $IPADDR --sport
        $UNPRIVPORTS \-d $SERVIDOR DE NOMBRES --
        dport 53 \-m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p udp \-s $IPADDR --sport
    $UNPRIVPORTS \-d $SERVIDOR DE NOMBRES --
    dport 53 -j ACEPTAR

$IPT -A INPUT -i $INTERNET -p udp \-s $SERVIDOR DE
    NOMBRES --sport 53 \-d $IPADDR --dport
    $UNPRIVPORTS -j ACEPTAR

#.....
# TCP se usa para respuestas grandes

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \-s $IPADDR --sport
        $UNPRIVPORTS \-d $NAMESERVER --dport 53 \-m
        estado --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \-s $IPADDR --sport
    $UNPRIVPORTS \-d $SERVIDOR DE NOMBRES --
    dport 53 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \-s $SERVIDOR DE
    NOMBRES --sport 53 \-d $IPADDR --dport
    $UNPRIVPORTS -j ACEPTAR
```

```
#.....
# Servidor de nombres de caché DNS (servidor local a servidor principal)

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p udp \ -s $IPADDR --
        sport 53 \ -d $SERVIDOR DE NOMBRES
        --dport 53 \ -m estado --estado NUEVO -j
        ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p udp \ -s $IPADDR --
    sport 53 \ -d $SERVIDOR DE NOMBRE
    --dport 53 -j ACEPTAR

$IPT -A INPUT -i $INTERNET -p udp \ -s $SERVIDOR
    DE NOMBRES --sport 53 \ -d $IPADDR --
    dport 53 -j ACEPTAR

#.....
# Solicitudes entrantes de clientes remotos a servidores locales

si [ "$ACCEPT_AUTH" = "1" ]; entonces si [ "$
    CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A INPUT -i $INTERNET -p tcp \ --sport
        $UNPRIVPORTS \ -d $IPADDR --dport
        113 \ -m estado --estado NUEVO -j
        ACEPTAR
    ser

$IPT -A INPUT -i $INTERNET -p tcp \ --sport
    $UNPRIVPORTS \ -d $IPADDR --dport
    113 -j ACEPTAR

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \ -s $IPADDR --sport
    113 \ --dport $UNPRIVPORTS -j ACEPTAR

demás
$IPT -A INPUT -i $INTERNET -p tcp \ --sport
    $UNPRIVPORTS \ -d $IPADDR --dport
    113 -j REJECT --reject-with tcp-reset
ser

#####
# Envío de correo a cualquier servidor de correo externo
# Use "-d $MAIL_SERVER" si se usa una puerta de enlace de correo ISP en su lugar

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
        sport $UNPRIVPORTS \ --dport 25 -m estado --
        estado NUEVO -j ACEPTAR
    ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
    sport $UNPRIVPORTS \ --dport 25 -j ACEPTAR
```

```
$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \ --sport 25 \ -d
    $IPADDR --dport $UNPRIVPORTS -j ACEPTAR

#####
# Recuperación de correo como cliente POP (puerto TCP 110)

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
        sport $UNPRIVPORTS \ -d $SERVIDOR_POP --
        dport 110 -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
    sport $UNPRIVPORTS \ -d $POP_SERVER --
    dport 110 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \
    -s $POP_SERVER --sport 110 \ -d
    $IPADDR --dport $UNPRIVPORTS -j ACEPTAR

##### # ssh (Puerto TCP 22)

# Solicitudes salientes de clientes locales a servidores remotos

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
        sport $SSH_PORTS \ --dport 22 -m estado --
        estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
    sport $SSH_PORTS \ --dport 22 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \ --sport 22 \ -d
    $IPADDR --dport $SSH_PORTS -j ACEPTAR

#.....
# Solicitudes entrantes de clientes remotos a servidores locales

si [ "$SSH_SERVER" = "1" ]; entonces si [ "$
    CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A INPUT -i $INTERNET -p tcp \ --sport
        $SSH_PORTS \ -d $IPADDR --dport 22 \
        -m estado --estado NUEVO -j ACEPTAR

ser

$IPT -A ENTRADA -i $INTERNET -p tcp \ --sport
    $SSH_PORTS \ -d $IPADDR --dport 22
    -j ACEPTAR
```

```

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \ -s $IPADDR --
    sport 22 \ --dport $SSH_PORTS -j ACEPTAR

ser

##### # ftp (Puertos TCP
21, 20)

# Solicitudes salientes de clientes locales a servidores remotos

# Conexión de control saliente al puerto 21 if
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR
        --sport $UNPRIVPORTS \ --dport 21 -m estado --
        estado NUEVO -j ACEPTAR

ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR
    --sport $UNPRIVPORTS \ --dport 21 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \ --sport 21 \ -d
    $IPADDR --dport $UNPRIVPORTS -j ACEPTAR

# Conexión de canal de datos de modo de puerto entrante desde el puerto 20 si [ "$
CONNECTION_TRACKING" = "1" ]; entonces # Esta regla no es necesaria si se usa
el módulo ip_contrack_ftp #.

$IPT -A INPUT -i $INTERNET -p tcp \ --sport 20 \ -d
    $IPADDR --dport $UNPRIVPORTS \ -m
    estado --estado NUEVO -j ACEPTAR

ser

$IPT -A INPUT -i $INTERNET -p tcp \ --sport 20 \ -d
    $IPADDR --dport $UNPRIVPORTS -j
    ACEPTAR

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \ -s $IPADDR --
    sport $UNPRIVPORTS \ --dport 20 -j ACEPTAR

# Conexión de canal de datos en modo pasivo saliente entre puertos sin privilegios si [ "$
CONNECTION_TRACKING" = "1" ]; entonces # Esta regla no es necesaria si se usa el módulo ip_contrack_ftp
#.

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR
    --sport $UNPRIVPORTS \ --dport $UNPRIVPORTS
    -m estado --estado NUEVO -j ACEPTAR

ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR
    --sport $UNPRIVPORTS \ --dport $UNPRIVPORTS
    -j ACEPTAR

```

```

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \ --sport
    $UNPRIVPORTS \ -d $IPADDR --dport
    $UNPRIVPORTS -j ACCEPTAR

#.....
# Solicitudes entrantes de clientes remotos a servidores locales

si [ "$FTP_SERVER" = "1" ]; luego

    # Conexión de control entrante al puerto 21 if
    [ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A INPUT -i $INTERNET -p tcp \ --sport
        $UNPRIVPORTS \ -d $IPADDR --dport
        21 \ -m estado --estado NUEVO -j
        ACCEPTAR
ser

$IPT -A INPUT -i $INTERNET -p tcp \ --sport
    $UNPRIVPORTS \ -d $IPADDR --dport
    21 -j ACCEPTAR

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \ -s $IPADDR --sport
    21 \ --dport $UNPRIVPORTS -j ACCEPTAR

# Conexión del canal de datos del modo de puerto saliente al puerto 20 si [ "$
CONNECTION_TRACKING" = "1" ]; luego
$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
    sport 20 \ --dport $UNPRIVPORTS -m
    estado --estado NUEVO -j ACCEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
    sport 20 \ --dport $UNPRIVPORTS -j
    ACCEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \
    --sport $UNPRIVPORTS \ -d
    $IPADDR --dport 20 -j ACCEPTAR

# Conexión de canal de datos de modo pasivo entrante entre puertos sin privilegios si [ "$ CONNECTION_TRACKING"
= "1" ]; luego
    $IPT -A INPUT -i $INTERNET -p tcp \ --sport
        $UNPRIVPORTS \ -d $IPADDR --dport
        $UNPRIVPORTS \ -m estado --estado NUEVO -j
        ACCEPTAR
ser

$IPT -A INPUT -i $INTERNET -p tcp \ --sport
    $UNPRIVPORTS \ -d $IPADDR --dport
    $UNPRIVPORTS -j ACCEPTAR

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \ -s $IPADDR --sport
    $UNPRIVPORTS \ --dport $UNPRIVPORTS -j
    ACCEPTAR
ser

```

```
#####
# Tráfico web HTTP (Puerto TCP 80)

# Solicitudes salientes de clientes locales a servidores remotos

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
        sport $UNPRIVPORTS \ --dport 80 -m estado --
        estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
    sport $UNPRIVPORTS \ --dport 80 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \ --sport 80 \ -d
    $IPADDR --dport $UNPRIVPORTS -j ACEPTAR

#.....
# Solicitudes entrantes de clientes remotos a servidores locales

si [ "$SERVIDOR_WEB" = "1" ]; entonces si [ "$
    CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A INPUT -i $INTERNET -p tcp \ --sport
        $UNPRIVPORTS \ -d $IPADDR --dport
        80 \ -m state --state NEW -j ACEPTAR
ser

$IPT -A INPUT -i $INTERNET -p tcp \ --sport
    $UNPRIVPORTS \ -d $IPADDR --dport
    80 -j ACEPTAR

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \ -s $IPADDR --sport
    80 \ --dport $UNPRIVPORTS -j ACEPTAR

ser

#####
# Tráfico web SSL (Puerto TCP 443)

# Solicitudes salientes de clientes locales a servidores remotos

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
        sport $UNPRIVPORTS \ --dport 443 -m estado --
        estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
    sport $UNPRIVPORTS \ --dport 443 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \ --sport 443 \ -d
    $IPADDR --dport $UNPRIVPORTS -j ACEPTAR
```



```
#.....
# Solicitudes entrantes de clientes remotos a servidores locales

si [ "$SSL_SERVER" = "1" ]; entonces si [ "$
    CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A INPUT -i $INTERNET -p tcp --sport
        $UNPRIVPORTS \ -d $IPADDR --dport
        443 \ -m estado --estado NUEVO -j
        ACEPTAR
ser

$IPT -A INPUT -i $INTERNET -p tcp --sport
    $UNPRIVPORTS \ -d $IPADDR --dport
    443 -j ACEPTAR

$IPT -A SALIDA -o $INTERNET -p tcp ! --syn \ -s $IPADDR --sport
    443 \ --dport $UNPRIVPORTS -j ACEPTAR

ser

##### # whois (Puerto TCP
43)

# Solicitudes salientes de clientes locales a servidores remotos

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
        sport $UNPRIVPORTS \ --dport 43 -m estado --
        estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p tcp \ -s $IPADDR --
    sport $UNPRIVPORTS \ --dport 43 -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p tcp ! --syn \ --sport 43 \ -d
    $IPADDR --dport $UNPRIVPORTS -j ACEPTAR

##### # Acceso a servidores
de hora de red remotos (UDP 123)
# Nota: Algunos clientes y servidores usan el puerto de origen 123 # cuando
consultan un servidor remoto en el puerto de destino 123.

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p udp \ -s $IPADDR --
        sport $UNPRIVPORTS \ -d $TIME_SERVER --
        dport 123 \ -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A SALIDA -o $INTERNET -p udp \ -s $IPADDR --
    sport $UNPRIVPORTS \ -d $TIME_SERVER --
    dport 123 -j ACEPTAR
```

```

$IPT -A INPUT -i $INTERNET -p udp \-s
    $TIME_SERVER --sport 123 \-d $IPADDR
    --dport $UNPRIVPORTS -j ACCEPTAR

#####
# Acceso al servidor DHCP de su ISP (Puertos UDP 67, 68)

# El cortafuegos ignora explícitamente algunos paquetes de difusión.
# Otros son eliminados por la política predeterminada.
# Las pruebas de DHCP deben preceder a las reglas relacionadas con la transmisión, ya que DHCP
se basa # inicialmente en el tráfico de transmisión.

si [ "$DHCP_CLIENT" = "1" ]; luego
    # Inicialización o reenlace: Sin concesión o Tiempo de concesión vencido.

$IPT -A SALIDA -o $ INTERNET -p udp \-s $
    BROADCAST_SRC --sport 68 \-d $
    BROADCAST_DEST --dport 67 -j ACCEPTAR

# DHCP OFFER entrante de los servidores DHCP disponibles

$IPT -A INPUT -i $ INTERNET -p udp \-s $
    BROADCAST_SRC --sport 67 \-d $
    BROADCAST_DEST --dport 68 -j ACCEPTAR

# Regresar a la inicialización # El cliente
conoce su servidor, pero ha perdido su concesión, # o necesita volver a confirmar la dirección
IP después de reiniciar.

$IPT -A SALIDA -o $INTERNET -p udp \-s
    $BROADCAST_SRC --sport 68 \-d
    $DHCP_SERVER --dport 67 -j ACCEPTAR

$IPT -A INPUT -i $ INTERNET -p udp \-s $
    DHCP_SERVER --sport 67 \-d $
    BROADCAST_DEST --dport 68 -j ACCEPTAR

# Como resultado de lo anterior, se supone que debemos cambiar nuestra dirección IP #
con este mensaje, que está dirigido a nuestra nueva # dirección antes de que el cliente
dhcp haya recibido la actualización.
# Según la implementación del servidor, la dirección de destino # puede ser la nueva dirección IP, la
dirección de subred o la # dirección de difusión limitada.

# Si la dirección de la subred de la red se usa como destino, # la siguiente regla debe
permitir los paquetes entrantes destinados a la # dirección de la subred, y la regla debe
preceder a cualquier regla general # que bloquee dichos paquetes de transmisión entrantes.

$IPT -A ENTRADA -i $INTERNET -p udp \-s
    $DHCP_SERVER --sport 67 \ --dport 68 -j
    ACCEPTAR

# Renovación de arrendamiento

```

```
$IPT -A SALIDA -o $INTERNET -p udp \ -s $IPADDR --
    sport 68 \ -d $DHCP_SERVER --dport 67 -j
    ACCEPTAR
$IPT -A INPUT -i $INTERNET -p udp \ -s $DHCP_SERVER
    --sport 67 \ -d $IPADDR --dport 68 -j ACCEPTAR

# Rechazar transmisiones dirigidas
# Se utiliza para mapear redes y en ataques de denegación de servicio iptables -A INPUT
-i $INTERNET -d $SUBNET_BASE -j DROP iptables -A INPUT -i $INTERNET -d
$SUBNET_BROADCAST -j DROP

# Rechazar transmisiones limitadas iptables
-A INPUT -i $INTERNET -d $BROADCAST_DEST -j DROP

ser
#####
# Mensajes de estado y control ICMP

# Registrar y soltar fragmentos ICMP iniciales
$IPT -A INPUT -i $INTERNET --fragment -p icmp -j LOG \ --log-prefix "ICMP fragmentado:"

$IPT -A ENTRADA -i $INTERNET --fragmento -p icmp -j DROP

$IPT -A INPUT -i $INTERNET -p icmp \ --icmp-type source-
quench -d $IPADDR -j ACCEPTAR

$IPT -A SALIDA -o $INTERNET -p icmp \ -s $IPADDR --icmp-
type source-quench -j ACCEPTAR

$IPT -A INPUT -i $INTERNET -p icmp \ --icmp-type parámetro-
problema -d $IPADDR -j ACCEPTAR

$IPT -A SALIDA -o $INTERNET -p icmp \ -s $IPADDR --icmp-
type parámetro-problema -j ACCEPTAR

$IPT -A INPUT -i $INTERNET -p icmp \ --icmp-type destino-
inalcanzable -d $IPADDR -j ACCEPTAR

$IPT -A SALIDA -o $INTERNET -p icmp \ -s $IPADDR --icmp-
type fragmentation-needed -j ACCEPT

# No registre los mensajes de error ICMP salientes descartados
$IPT -A SALIDA -o $INTERNET -p icmp \ -s $IPADDR --icmp-
type destino-inalcanzable -j DROP

# Respuestas intermedias de seguimiento de ruta
$IPT -A INPUT -i $INTERNET -p icmp \ --icmp-type time-
exceeded -d $IPADDR -j ACCEPTAR

# Permitir pings salientes a cualquier lugar if
[ "$CONNECTION_TRACKING" = "1" ]; luego
    $IPT -A SALIDA -o $INTERNET -p icmp \ -s $IPADDR --icmp-
type echo-request \ -m state --state NEW -j ACCEPTAR
```

ser

```

$IPT -A SALIDA -o $INTERNET -p icmp \
    -s $IPADDR --petición de eco de tipo icmp -j ACEPTAR

$IPT -A ENTRADA -i $INTERNET -p icmp \
    --icmp-type eco-respuesta -d $IPADDR -j ACEPTAR

# Permitir pings entrantes de hosts confiables
si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A ENTRADA -i $INTERNET -p icmp \
        -s $MI_ISP --solicitud de eco de tipo icmp -d $IPADDR \
        -m estado --estado NUEVO -j ACEPTAR
ser

$IPT -A ENTRADA -i $INTERNET -p icmp \
    -s $MI_ISP --solicitud de eco de tipo icmp -d $IPADDR -j ACEPTAR

$IPT -A SALIDA -o $INTERNET -p icmp \
    -s $IPADDR --icmp-type eco-respuesta -d $MI_ISP -j ACEPTAR

#####
# Registro de paquetes descartados
$IPT -A ENTRADA -i $INTERNET -p tcp \
    -d $IPADDR -j REGISTRO

$IPT -A SALIDA -o $INTERNET -j REGISTRO

salida 0

```

nftables Firewall para un sistema independiente del Capítulo 5

Esta sección contiene una secuencia de comandos nftables basada en el ejemplo que se muestra en el Capítulo 5. La secuencia de comandos se basa en el archivo `setup-tables` de ese capítulo, que espera encontrar en el mismo directorio. Estos son los contenidos del archivo `setup-tables` :

```

filtro de tabla {
    entrada de cadena {
        tipo filtro gancho entrada prioridad 0;
    }
    salida en cadena {
        tipo filtro gancho salida prioridad 0;
    }
}

```

Aquí está la secuencia de comandos del cortafuegos:

```
#!/bin/sh
```

```

NFT = "/usr/local/sbin/nft"
INTERNET="eth0"
LOOPBACK_INTERFACE="lo"
IPADDR="mi.dirección.ip"
MY_ISP="mi.isp.dirección.rango"
SUBNET_BASE="mi.subred.base"

```

```

# Ubicación de nft en su sistema
# Interfaz conectada a Internet
# Sin embargo, su sistema lo nombra
# Su dirección IP
# Servidor ISP y rango de direcciones NOC
# La dirección de red de su subred

```

```

SUBNET_BROADCAST="my.subnet.bcast" # Dirección de transmisión de su subred
LOOPBACK="127.0.0.0/8"              # Rango de direcciones de loopback reservado
NAMESERVER="isp.nombre.servidor.1"  # Dirección de un servidor de nombres remoto
SMTP_GATEWAY="isp.smtp.servidor"    # Dirección de una puerta de enlace de correo remoto
POP_SERVER="isp.pop.servidor"        # Dirección de un servidor pop remoto
IMAP_SERVER="isp.imap.servidor"      # Dirección de un servidor imap remoto
TIME_SERVER="tiempo.nist.gov"        # Dirección de un servidor NTP remoto
CLASE_A="10.0.0.0/8"                 # Redes privadas clase A
CLASE_B="172.16.0.0/12"              # Redes privadas clase B
CLASE_C="192.168.0.0/16"             # Redes privadas clase C
CLASE_D_MULTICAST="224.0.0.0/4"      # Direcciones de multidifusión de clase D
CLASS_E_RESERVED_NET="240.0.0.0/5"  # Direcciones reservadas Clase E
TRANSMITIR_SRC = "0.0.0.0"           # Dirección de origen de la transmisión
BROADCAST_DEST = "255.255.255.255"   # Dirección de destino de transmisión
PRIVPORTS="0-1023"                   # Rango de puertos privilegiado y bien conocido
UNPRIVPORTS="1024-65535"             # Rango de puertos sin privilegios

para i en `tablas de lista de $ NFT | awk '{imprimir $2}'`
hacer
    echo "Enjuagar ${i}"
    $NFT vaciar la tabla ${i} para j en
        '$NFT list table ${i} | cadena grep | awk '{imprimir $2}'`
    hacer
        echo "...Eliminando la cadena ${j} de la tabla ${i}"
        $ Cadena de eliminación de NFT $ {i} $ {j}
    hecho
    echo "Eliminando ${i}"
    $ Tabla de eliminación de NFT $ {i}
hecho

si [ "$1" = "parar" ]
luego
    echo "¡Firewall completamente detenido! ADVERTENCIA: ESTE HOST NO TIENE FIREWALL EN FUNCIONAMIENTO."
salida 0
ser

$NFT -f tablas de configuración

#bucle invertido
$ NFT agregar entrada de filtro de regla iifname para aceptar
$ NFT agregar salida de filtro de regla oifname para aceptar

#estado de conexión
$NFT agregar regla filtro entrada ct estado establecido, relacionado aceptar
$NFT agregar regla filtrar entrada ct estado prefijo de registro no válido "\Entrada NO VÁLIDA: \ " límite yvelocidad 3/segundo caída

$ NFT agregar regla filtro salida ct estado establecido, relacionado aceptar
$NFT añadir filtro de regla salida ct estado prefijo de registro no válido "\salida NO VÁLIDA: \ " ytasa de límite 3/segundo caída

#suplantación de dirección de origen
$NFT agregar entrada de filtro de regla si $!INTERNET ip saddr $!PADDR

#direcciones inválidas
$NFT agregar entrada de filtro de regla si $!INTERNET ip saddr $!CLASS_A drop
$NFT agregar entrada de filtro de regla si $!INTERNET ip saddr $!CLASS_B drop
$NFT agregar entrada de filtro de regla si $!INTERNET ip saddr $!CLASS_C drop
$NFT agregar entrada de filtro de regla si $!INTERNET ip saddr $!LOOPBACK drop

```

330 Apéndice B Ejemplos de cortafuegos y secuencias de comandos de soporte

```

#broadcast src y destino
$NFT agregar entrada de filtro de regla si $!INTERNET ip saddr $BROADCAST_DEST tasa de límite de registro y3/segundo caída

$NFT agregar entrada de filtro de regla si $!INTERNET ip saddr $BROADCAST_SRC tasa de límite de registro y3/segundo caída

#emisión dirigida
$NFT agregar entrada de filtro de regla si $!INTERNET ip daddr $SUBNET_BASE drop
$NFT agregar entrada de filtro de regla si $!INTERNET ip daddr $SUBNET_BROADCAST soltar

#emisión limitada
$NFT agregar entrada de filtro de regla iif $ !INTERNET ip daddr $ BROADCAST_DEST drop

#multidifusión
$NFT agregar entrada de filtro de regla si $!INTERNET ip saddr $CLASS_D_MULTICAST soltar
$NFT agregar entrada de filtro de regla si $!INTERNET ip daddr $CLASS_D_MULTICAST ip protocol y!= udp drop

$NFT agregar entrada de filtro de regla iif $!INTERNET ip daddr $CLASS_D_MULTICAST ip protocol yudp accept

#clase e
$NFT agregar entrada de filtro de regla si $!INTERNET ip saddr $CLASS_E_RESERVED_NET drop

#x ventanas
XWINDOW_PORTS="6000-6063"
$NFT agregar salida de filtro de regla oif $!INTERNET ct state new tcp dport $XWINDOW_PORTS yreject

$NFT agregar entrada de filtro de regla si $!INTERNET ct estado nuevo tcp dport $XWINDOW_PORTS ydrop

NFS_PORT="2049"                                # (TCP) NFS
CALCETINES_PORT="1080"                         # (TCP) calcetines
OPENWINDOWS_PORT="2000"                       # (TCP) Ventanas abiertas
SQUID_PORT="3128"                              # (TCP) calamar

$NFT añadir salida de filtro de regla oif $!INTERNET tcp dport {$NFS_PORT,$SOCKS_ yPORT,$OPENWINDOWS_PORT,
$SQUID_PORT} ct estado nuevo rechazo
$NFT agregar entrada de filtro de regla si $!INTERNET tcp dport {$NFS_PORT,$SOCKS_ yPORT,
$OPENWINDOWS_PORT,$SQUID_PORT} ct state new drop

NFS_PORT="2049"                                #NFS
LOCKD_PORT="4045"                             # RPC bloqueado para NFS
$NFT agregar salida de filtro de regla oif $!INTERNET upd dport {$NFS_PORT,$LOCKD_PORT} yrechazar

$NFT agregar entrada de filtro de regla si $!INTERNET upd dport {$NFS_PORT,$LOCKD_PORT} drop

#DNS
$NFT agregar salida de filtro de regla oif $!INTERNET ip saddr $IPADDR udp sport $UNPRIVPORTS yip daddr $NAMESERVER udp dport
53 ct state new accept
$NFT agregar entrada de filtro de regla iif $!INTERNET ip daddr $IPADDR udp dport $UNPRIVPORTS yip saddr $NAMESERVER udp
sport 53 aceptar

#tcp DNS
$NFT agregar salida de filtro de regla oif $!INTERNET ip saddr $IPADDR tcp sport $UNPRIVPORTS yip daddr $NAMESERVER tcp dport
53 ct state new accept
$NFT agregar entrada de filtro de regla iif $!INTERNET ip daddr $IPADDR tcp dport $UNPRIVPORTS yip saddr $NAMESERVER tcp sport
53 tcp flags != syn accept

```

```
#tcp smtp
$NFT agregar salida de filtro de regla oif $INTERNET ip daddr $SMTP_GATEWAY tcp dport 25 ip ysaddr $IPADDR
tcp sport $UNPRIVPORTS aceptar
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $SMTP_GATEWAY tcp sport 25 ip ydaddr $IPADDR
tcp dport $UNPRIVPORTS tcp flags != syn accept

$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $UNPRIVPORTS ytcp dport 25 aceptar

$NFT agregar entrada de filtro de regla iif $INTERNET ip daddr $IPADDR tcp sport 25 tcp dport y$UNPRIVPORTS tcp
flags != syn accept
$NFT agregar entrada de filtro de regla si $INTERNET tcp sport $UNPRIVPORTS ip daddr $IPADDR ytcp dport 25
aceptar
$NFT agregar salida de filtro de regla oif $INTERNET tcp sport 25 ip saddr $IPADDR tcp dport y$UNPRIVPORTS tcp
flags != syn accept

#tcppop3
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR ip daddr $POP_SERVER ytcp sport
$UNPRIVPORTS tcp dport 110 accept
$NFT agregar entrada de filtro de regla iif $INTERNET ip saddr $POP_SERVER tcp sport 110 ip ydaddr $IPADDR
tcp dport $UNPRIVPORTS tcp flags != syn accept

#tcp diablillos
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $UNPRIVPORTS yip daddr
$IMAP_SERVER tcp dport 993 aceptar
$NFT agregar entrada de filtro de regla iif $INTERNET ip saddr $IMAP_SERVER tcp sport 993 ip ydaddr $IPADDR
tcp dport $UNPRIVPORTS tcp flags != syn accept

#permitir que los clientes se conecten a su servidor IMAP
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr 0/0 tcp sport $UNPRIVPORTS ip ydaddr $IPADDR
tcp dport 993 aceptar
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport 993 ip daddr y0/0 tcp dport
$UNPRIVPORTS tcp flags != syn accept

#ssh
SSH_PORTS="1020-65535"
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $SSH_PORTS ytcp dport 22 accept

$NFT agregar entrada de filtro de regla iif $INTERNET tcp sport 22 ip daddr $IPADDR tcp dport y$SSH_PORTS tcp
flags != syn accept
$NFT agregar entrada de filtro de regla si $INTERNET tcp sport $SSH_PORTS ip daddr $IPADDR ytcp dport 22
aceptar
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport 22 tcp dport y$SSH_PORTS tcp flags !
= syn accept

#ftp
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR tcp sport $UNPRIVPORTS ytcp dport 21 accept

$NFT agregar entrada de filtro de regla si $INTERNET ip daddr $IPADDR tcp sport 21 tcp dport y$UNPRIVPORTS
aceptar
#asumir el uso del módulo de estado ct para ftp

#dhcp (esta máquina hace dhcp en dos interfaces, por lo que necesita más reglas)
$NFT añadir salida de filtro de regla oif $INTERNET ip saddr $BROADCAST_SRC udp sport 67-68 yip daddr
$BROADCAST_DEST udp dport 67-68 aceptar
$NFT agregar entrada de filtro de regla si $INTERNET udp sport 67-68 udp dport 67-68 aceptar
$NFT agregar regla filtrar salida udp sport 67-68 udp dport 67-68 aceptar
$NFT agregar entrada de filtro de regla udp sport 67-68 udp dport 67-68 aceptar
```

```
#ntp
$NFT agregar salida de filtro de regla oif $INTERNET ip saddr $IPADDR udp sport $UNPRIVPORTS yip daddr $TIME_SERVER udp dport
123 aceptar
$NFT agregar entrada de filtro de regla si $INTERNET ip saddr $TIME_SERVER udp sport 123 ip ydaddr $IPADDR udp dport
$UNPRIVPORTS aceptar

#registrar todo lo que haya llegado hasta aquí
$NFT agregar registro de entrada de filtro de regla
$NFT agregar registro de salida de filtro de regla

#política predeterminada:
$ NFT agregar entrada de filtro de regla
$NFT agregar regla filtrar salida rechazar
```

Firewall de iptables optimizado del Capítulo 6

Para la mayoría de los sistemas en DSL, módem por cable y conexiones de línea alquilada de menor velocidad, es muy probable que el código de red de Linux pueda manejar paquetes más rápido que la conexión de red. Particularmente debido a que las reglas del firewall dependen del orden y son difíciles de construir, organizar las reglas para que sean legibles es probablemente una victoria mayor que organizarlas para la velocidad.

Además del orden general de reglas, iptables admite listas de reglas definidas por el usuario, o cadenas, que puede usar para optimizar las reglas de su firewall. Pasar un paquete de una cadena a otra según los valores en el encabezado del paquete proporciona un medio para probar selectivamente el paquete contra un subconjunto de las reglas de ENTRADA, SALIDA o ADELANTE en lugar de probar el paquete contra cada regla en la lista hasta que se encuentre una coincidencia. fundar.

Según estos scripts particulares, un paquete de entrada de un servidor de tiempo NTP debe probarse contra numerosas reglas de entrada en el script de firewall no optimizado antes de que el paquete coincida con su regla ACCEPT . Usando cadenas definidas por el usuario para optimizar el firewall, el mismo paquete de entrada se prueba contra muchas menos reglas antes de que coincida con su regla ACCEPT . Con la adición del seguimiento del estado de la conexión, el mismo paquete de entrada se prueba solo con un puñado de reglas antes de que coincida con su regla ACCEPT .

Con las cadenas definidas por el usuario, las reglas se utilizan para pasar paquetes entre cadenas, así como para definir bajo qué condiciones se acepta o descarta el paquete. Si un paquete no coincide con ninguna regla en la cadena definida por el usuario, el control vuelve a la cadena de llamada. Si el paquete no coincide con una regla de selección de cadena de nivel superior, el paquete no se pasa a esa cadena para probarlo con las reglas de la cadena. El paquete simplemente se prueba contra la siguiente regla de selección de cadena.

El siguiente es el script de firewall del Capítulo 5, optimizado con cadenas definidas por el usuario:

```
#!/bin/sh

/sbin/modprobe ip_conntrack_ftp

CONEXIÓN_SEGUIMIENTO="1"
ACCEPT_AUTH="0"
DHCP_CLIENTE="0"
IPT="/sbin/iptables"
INTERNET="eth0"

# Ubicación de iptables en su sistema
# Interfaz conectada a Internet
```



```

LOOPBACK_INTERFACE="lo"                # Sin embargo, su sistema lo nombra
IPADDR="mi.dirección.ip"                # Su dirección IP
SUBNET_BASE="red.dirección"              # Dirección base del segmento de red ISP
SUBNET_BROADCAST="directed.broadcast"    # Dirección de difusión del segmento de red
MY_ISP="mi.isp.dirección.rango"          # Servidor ISP y rango de direcciones NOC

NAMESERVER_1="isp.nombre.servidor.1"     # Dirección de un servidor de nombres remoto
NAMESERVER_2="isp.nombre.servidor.2"     # Dirección de un servidor de nombres remoto
NAMESERVER_3="isp.nombre.servidor.3"     # Dirección de un servidor de nombres remoto
POP_SERVER="isp.pop.servidor"            # Dirección de un servidor pop remoto
MAIL_SERVER="isp.correo.servidor"        # Dirección de una puerta de enlace de correo remoto
NEWS_SERVER="isp.noticias.servidor"      # Dirección de un servidor de noticias remoto
TIME_SERVER="algún.time.servidor"        # Dirección de un servidor de tiempo remoto
DHCP_SERVER="isp.dhcp.servidor"          # Dirección del servidor dhcp de su ISP
SSH_CLIENT="algún.ssh.cliente"

LOOPBACK="127.0.0.0/8"                  # Rango de direcciones de loopback reservado
CLASE_A="10.0.0.0/8"                    # Redes privadas clase A
CLASE_B="172.16.0.0/12"                  # Redes privadas clase B
CLASE_C="192.168.0.0/16"                 # Redes privadas clase C
CLASE_D_MULTICAST="224.0.0.0/4"          # Direcciones de multidifusión de clase D
CLASS_E_RESERVED_NET="240.0.0.0/5"      # Direcciones reservadas Clase E
TRANSMITIR_SRC = "0.0.0.0"               # Dirección de origen de la transmisión
BROADCAST_DEST = "255.255.255.255"       # Dirección de destino de transmisión

PRIVPORTS="0:1023"                      # Rango de puertos privilegiado y bien conocido
UNPRIVPORTS="1024:65535"                 # Rango de puertos sin privilegios

# Traceroute generalmente usa -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS = "33434: 33523"

USER_CHAINS="EXT-entrada                 Salida EXT \
tcp-state-flags destino-                seguimiento de conexión \
dirección-fuente-comprobación-verificación \
consulta-servidor-dns-local respuesta-servidor-dns-remoto \
solicitud-cliente-tcp-local respuesta-servidor-tcp-remoto \
solicitud-cliente-tcp-remoto respuesta-servidor-tcp-local \
solicitud-cliente-udp-local respuesta-servidor-udp-remoto \
local-dhcp-client-query remote-dhcp-server-response \
EXT-icmp-salida                         Entrada EXT-icmp \
Inicio de sesión EXT                    EXT-cierre de sesión \
registro-tcp-estado"

#####

# Habilitar la protección de eco de transmisión
eco 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

# Deshabilitar paquetes enrutados de origen
para f en /proc/sys/net/ipv4/conf/*/accept_source_route; hacer
    eco 0 > $f
hecho

# Habilitar la protección de cookies TCP SYN
eco 1 > /proc/sys/net/ipv4/tcp_syncookies

```

334 Apéndice B Ejemplos de cortafuegos y secuencias de comandos de soporte

```
# Deshabilitar la aceptación de redirección ICMP para f
en /proc/sys/net/ipv4/conf/*accept_redirects; hacer echo 0 > $f hecho
```

```
# No envíe mensajes de redirección para f en /
proc/sys/net/ipv4/conf/*send_redirects; hacer echo 0 > $f hecho
```

```
# Descartar los paquetes falsificados que llegan a una interfaz, los cuales, si se responden, # darían como resultado
que la respuesta saliera por una interfaz diferente. para f en /proc/sys/net/ipv4/conf/*rp_filter; haz echo 1 > $f hecho
```

```
# Registra paquetes con direcciones imposibles. para f en /proc/
sys/net/ipv4/conf/*log_martians; haz echo 1 > $f hecho
```

```
#####
```

```
# Eliminar cualquier regla existente de todas las cadenas
```

```
$IPT --vaciar
```

```
$IPT -t natural --vaciar
```

```
$IPT -t mangle --vaciar
```

```
$IPT-X
```

```
$IPT -t nacional -X
```

```
$IPT -t destrozar -X
```

```
$IPT --política ENTRADA ACEPTAR $IPT --
política SALIDA ACEPTAR $IPT --política
ADELANTE ACEPTAR $IPT -t nat --política
PRE-ENRUTAMIENTO ACEPTAR $IPT -t nat --política SALIDA
ACEPTAR $IPT -t nat --política POSTERIOR AL ENRUTAMIENTO
ACCEPT $IPT -t mangle --policy PREROUTING ACCEPT $IPT -t
mangle --policy OUTPUT ACCEPT if [ "$1" = "stop" ] luego echo
"¡Firewall completamente detenido! ADVERTENCIA: ESTE HOST
NO TIENE FIREWALL FUNCIONANDO". salir 0 fi
```

```
# Tráfico ilimitado en la interfaz loopback
```

```
$IPT -A ENTRADA -i lo -j ACEPTAR
```

```
$IPT -A SALIDA -o lo -j ACEPTAR
```

```
# Establecer la política predeterminada para descartar
```

```
$IPT --política ENTRADA DROP
```

```
$IPT --política CAÍDA DE SALIDA
```

```
$IPT --política FORWARD DROP
```

```
# Crea las cadenas definidas por el usuario para i en
```

```
$USER_CHAINS; hacer
```

```
    $IPT -N $he
```

```
terminado
```

```
#####
# Servidor de nombres de almacenamiento en caché DNS (consulta al servidor primario remoto)

$IPT -A EXT-salida -p udp --sport 53 --dport 53 \
    -j local-dns-servidor-consulta

$IPT -A Entrada EXT -p udp --sport 53 --dport 53 \ -j respuesta-servidor-
    dns-remoto

# Servidor de nombres de caché DNS (consulta al servidor remoto a través de TCP)

$IPT -A EXT-salida -p tcp \
    --sport $UNPRIVPORTS --dport 53 \ -j local-dns-
    server-query

$IPT -A Entrada EXT -p tcp ! --syn \ --sport 53 --
    dport $UNPRIVPORTS \ -j respuesta-servidor-
    dns-remoto

#####
# Servidor de nombres de reenvío de DNS o solicitudes de clientes

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A local-dns-server-query \ -d
        $NAMESERVER_1 \ -m state --
        state NEW -j ACEPTAR

    $IPT -A local-dns-server-query \ -d
        $NAMESERVER_2 \ -m state --
        state NEW -j ACEPTAR

    $IPT -A local-dns-server-query \ -d
        $NAMESERVER_3 \ -m state --
        state NEW -j ACEPTAR
ser

$IPT -A local-dns-server-query \ -d
    $NAMESERVER_1 -j ACCEPT

$IPT -A local-dns-server-query \ -d
    $NAMESERVER_2 -j ACCEPT

$IPT -A local-dns-server-query \ -d
    $NAMESERVER_3 -j ACCEPT

# Respuestas del servidor DNS a solicitudes locales

$IPT -A respuesta-servidor-dns-remoto \ -s
    $SERVIDOR_NOMBRE_1 -j ACEPTAR

$IPT -A respuesta-servidor-dns-remoto \ -s
    $SERVIDOR_NOMBRE_2 -j ACEPTAR

$IPT -A respuesta-servidor-dns-remoto \ -s
    $SERVIDOR_NOMBRE_3 -j ACEPTAR
```

```
#####
# Cliente TCP local, servidor remoto

$IPT -A Salida EXT -p tcp \ --sport
        $UNPRIVPORTS \ -j solicitud-
        cliente-tcp-local

$IPT -A Entrada EXT -p tcp ! --syn \ --dport
        $UNPRIVPORTS \ -j respuesta-
        servidor-tcp-remoto

#####
# Salida del cliente TCP local y cadenas de entrada del servidor remoto

#cliente SSH

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A solicitud-cliente-tcp-local -p tcp \
        -d <host seleccionado> --dport 22 \ -m state --
        state NEW \ -j ACEPTAR

ser

$IPT -A solicitud-cliente-tcp-local -p tcp \
    -d <host seleccionado> --dport 22 \ -j
    ACEPTAR

$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \ -s <host seleccionado>
    --sport 22 \ -j ACEPTAR

#.....
# Reglas de cliente para solicitudes de control HTTP, HTTPS, AUTH y FTP

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A local-tcp-client-request -p tcp \ -m multiport --
        destination-port 80,443,21 \ --syn -m state --state NEW \ -j
        ACCEPT

ser

$IPT -A solicitud-cliente-tcp-local -p tcp \ -m multipuerto --
    puerto-destino 80,443,21 \ -j ACEPTAR

$IPT -A respuesta-servidor-tcp-remoto -p tcp \ -m multiport --
    source-port 80,443,21 ! --syn \ -j ACEPTAR

#.....
# cliente POP

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A solicitud-cliente-tcp-local -p tcp \ -d $SERVIDOR_POP
        --dport 110 \ -m estado --estado NUEVO \ -j
        ACEPTAR

ser
```

```

$IPT -A solicitud-cliente-tcp-local -p tcp \ -d $SERVIDOR_POP
--dport 110 \ -j ACEPTAR

$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \ -s $POP_SERVER --
sport 110 \ -j ACEPTAR #.....

#cliente de correo SMTP

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A solicitud-cliente-tcp-local -p tcp \ -d
        $SERVIDOR_CORREO --dport 25 \ -m estado
        --estado NUEVO \ -j ACEPTAR

ser

$IPT -A solicitud-cliente-tcp-local -p tcp \ -d
    $SERVIDOR_CORREO --dport 25 \ -j ACEPTAR

$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \ -s
    $SERVIDOR_DE_CORREO --sport 25 \ -j ACEPTAR

#.....
# Cliente de noticias de Usenet

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A local-tcp-client-request -p tcp \ -d $NEWS_SERVER
        --dport 119 \ -m state --state NEW \ -j ACEPTAR

ser

$IPT -A local-tcp-client-request -p tcp \ -d $NEWS_SERVER
    --dport 119 \ -j ACEPTAR

$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \ -s
    $NOTICIAS_SERVIDOR --sport 119 \ -j ACEPTAR

#.....
# Cliente FTP - conexión de canal de datos en modo pasivo

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A local-tcp-client-request -p tcp \ --dport
        $UNPRIVPORTS \ -m state --state NEW \ -j
        ACEPTAR

ser

$IPT -A solicitud-cliente-tcp-local -p tcp \ --dport
    $UNPRIVPORTS -j ACEPTAR

$IPT -A respuesta-servidor-tcp-remoto -p tcp ! --syn \ --sport $UNPRIVPORTS
    -j ACEPTAR

```

```
#####
# Servidor TCP local, cliente remoto

$IPT -A Entrada EXT -p tcp \ --sport
    $UNPRIVPORTS \ -j solicitud-
    cliente-tcp-remoto

$IPT -A Salida EXT -p tcp ! --syn \ --dport
    $UNPRIVPORTS \ -j local-tcp-server-
    response

# Kludge para conexiones de canales de datos FTP entrantes # desde
# servidores remotos usando el modo de puerto.
# Los módulos de estado tratan esta conexión como RELACIONADA # si el
# módulo ip_conntrack_ftp está cargado.

$IPT -A Entrada EXT -p tcp \ --sport 20
    --dport $UNPRIVPORTS \ -j ACEPTAR

$IPT -A Salida EXT -p tcp ! --syn \ --sport
    $UNPRIVPORTS --dport 20 \ -j ACEPTAR

#####
# Entrada del cliente TCP remoto y cadenas de salida del servidor local

# servidor SSH

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A remote-tcp-client-request -p tcp \ -s <host seleccionado>
        --destination-port 22 \ -m state --state NEW \ -j ACEPTAR

ser

$IPT -A solicitud-cliente-tcp-remota -p tcp \
    -s <host seleccionado> --destination-port 22 \ -j ACEPTAR

$IPT -A local-tcp-server-response -p tcp ! --syn \
    --source-port 22 -d <host seleccionado> \ -j ACEPTAR

#.....
# Servidor de identificación AUTH

si [ "$ACCEPT_AUTH" = "0" ]; luego
    $IPT -A solicitud-cliente-tcp-remota -p tcp \ --destination-port 113
        \ -j REJECT --reject-with tcp-reset

demás
    $IPT -A solicitud-cliente-tcp-remota -p tcp \ --destination-port 113
        \ -j ACEPTAR
```

```

$IPT -A local-tcp-server-response -p tcp ! --syn \
    --source-port 113 \ -j ACCEPTAR

ser

#####
# Cliente UDP local, servidor remoto

$IPT -A Salida EXT -p udp \ --sport
    $UNPRIVPORTS \ -j solicitud-
    cliente-udp-local

$IPT -A Entrada EXT -p udp \ --dport
    $UNPRIVPORTS \ -j respuesta-
    servidor-udp-remoto

#####
# cliente de tiempo NTP

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A solicitud-cliente-udp-local -p udp \ -d $SERVIDOR_HORA
        --dport 123 \ -m estado --estado NUEVO \ -j
        ACCEPTAR

ser

$IPT -A solicitud-cliente-udp-local -p udp \ -d $SERVIDOR_HORA
    --dport 123 \ -j ACCEPTAR

$IPT -A respuesta-servidor-udp-remoto -p udp \
    -s $SERVIDOR_HORA --sport 123 \ -j ACCEPTAR

#####
# ICMP

$IPT -A Entrada EXT -p icmp -j Entrada EXT-icmp

$IPT -A Salida-EXT -p icmp -j Salida-icmp-EXT

#####
# tráfico ICMP

# Registrar y soltar fragmentos ICMP iniciales
$IPT -A EXT-icmp-in --fragment -j LOG \ --log-prefix "ICMP
    entrante fragmentado:"

$IPT -A EXT-icmp-entrada --fragmento -j DROP

$IPT -A EXT-icmp-out --fragment -j LOG \ --log-prefix "ICMP
    saliente fragmentado:"

$IPT -A EXT-icmp-out --fragmento -j DROP

# ping saliente

```

```

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A EXT-icmp-out -p icmp \ --icmp-type echo-
        request \ -m state --state NEW \ -j ACCEPTAR

ser

$IPT -A EXT-icmp-salida -p icmp \
    --petición de eco de tipo icmp -j ACCEPTAR

$IPT -A EXT-icmp-entrada -p icmp \
    --icmp-type eco-respuesta -j ACCEPTAR

# Ping entrante

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    $IPT -A EXT-icmp-in -p icmp \ -s $MI_ISP \ --icmp-
        type echo-request \ -m state --
        state NEW \ -j ACCEPTAR

ser

$IPT -A EXT-icmp-in -p icmp \ --icmp-type echo-
    request \ -s $MY_ISP -j ACCEPTAR

$IPT -A EXT-icmp-out -p icmp \ --icmp-type eco-
    respuesta \ -d $MI_ISP -j ACCEPTAR

# Destino inalcanzable Tipo 3
$IPT -A EXT-icmp-out -p icmp \ --icmp-type
    fragmentación necesaria -j ACCEPTAR

$IPT -A EXT-icmp-entrada -p icmp \
    --icmp-type destino-inalcanzable -j ACCEPTAR

# Problema de parámetro
$IPT -A EXT-icmp-salida -p icmp \
    --icmp-type parámetro-problema -j ACCEPTAR

$IPT -A EXT-icmp-in -p icmp \ --icmp-type
    parámetro-problema -j ACCEPTAR

# Tiempo excedido
$IPT -A EXT-icmp-entrada -p icmp \
    --icmp-type time-exceded -j ACCEPTAR

# Fuente para enfriar
$IPT -A EXT-icmp-salida -p icmp \
    --icmp-type source-quench -j ACCEPTAR

#####

# Indicadores de estado de TCP

# Todos los bits se borran
$IPT -A tcp-state-flags -p tcp --tcp-flags TODO NINGUNO -j log-tcp-state
    
```



```
# SYN y FIN están configurados
$IPT -A tcp-state-flags -p tcp --tcp-flags SYN,FIN SYN,FIN -j log-tcp-state

# SYN y RST están configurados
$IPT -A tcp-state-flags -p tcp --tcp-flags SYN,RST SYN,RST -j log-tcp-state

# FIN y RST están configurados
$IPT -A tcp-state-flags -p tcp --tcp-flags FIN,RST FIN,RST -j log-tcp-state

# FIN es el único bit establecido, sin el esperado ACK que lo acompaña
$IPT -A tcp-state-flags -p tcp --tcp-flags ACK,FIN FIN -j log-tcp-state

# PSH es el único conjunto de bits, sin el esperado ACK que lo acompaña
$IPT -A tcp-state-flags -p tcp --tcp-flags ACK,PSH PSH -j log-tcp-state

# URG es el único conjunto de bits, sin el ACK que se espera que lo acompañe
$IPT -A tcp-state-flags -p tcp --tcp-flags ACK,URG URG -j log-tcp-state

#####
# Registrar y descartar paquetes TCP con combinaciones de mal estado

$IPT -A log-tcp-state -p tcp -j LOG \ --log-prefix "Estado TCP
    ilegal: " \ --log-ip-options --log-tcp-options

$IPT -A log-tcp-state -j DROP

#####
# Comprobación de reglas de omisión para intercambios ESTABLECIDOS

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    # Omite los filtros de firewall para intercambios establecidos
    $IPT -A seguimiento de conexión -m estado \ --estado
        ESTABLECIDO, RELACIONADO \ -j ACEPTAR

    $IPT -A seguimiento de conexión -m estado --estado NO VÁLIDO \ -j REGISTRO --log-prefix
        "Paquete NO VÁLIDO:"
    $IPT -A seguimiento de conexión -m estado --estado NO VÁLIDO -j DROP
ser

#####
# tráfico DHCP

# El cortafuegos ignora explícitamente algunos paquetes de difusión.
# Otros son eliminados por la política predeterminada.
# Las pruebas de DHCP deben preceder a las reglas relacionadas con la transmisión, ya que DHCP
se basa # inicialmente en el tráfico de transmisión.

si [ "$DHCP_CLIENT" = "1" ]; luego

    # Inicialización o reenlace: Sin concesión o Tiempo de concesión vencido.

    $IPT -A local-dhcp-client-query \ -s $ BROADCAST_SRC
        \ -d $ BROADCAST_DEST -j ACEPTAR

# DHCP OFFER entrante de los servidores DHCP disponibles
```

```

$IPT -A respuesta-del-servidor-dhcp-remoto \-s
        $BROADCAST_SRC \-d $BROADCAST_DEST
        -j ACCEPT

# Regresar a la inicialización # El cliente conoce
su servidor, pero ha perdido su concesión, # o necesita volver a confirmar la dirección IP después de
reiniciar.

$IPT -A local-dhcp-client-query \-s $BROADCAST_SRC
        \-d $DHCP_SERVER -j ACEPTAR

$IPT -A respuesta-del-servidor-dhcp-remoto \-s
        $DHCP_SERVER \-d $BROADCAST_DEST
        -j ACCEPT

# Como resultado de lo anterior, se supone que debemos cambiar nuestra dirección IP # con este
mensaje, que está dirigido a nuestra nueva # dirección antes de que el cliente dhcp haya recibido la
actualización.
# Según la implementación del servidor, la dirección de destino # puede ser la nueva dirección IP, la dirección
de subred o la # dirección de difusión limitada.

# Si la dirección de la subred de la red se usa como destino, # la siguiente regla debe permitir los
paquetes entrantes destinados a la # dirección de la subred, y la regla debe preceder a cualquier
regla general # que bloquee dichos paquetes de transmisión entrantes.

$IPT -A respuesta-del-servidor-dhcp-remoto \-s
        $DHCP_SERVER -j ACCEPT

# Renovación de arrendamiento

$IPT -A local-dhcp-client-query \-s $IPADDR \-d
        $DHCP_SERVER -j ACEPTAR

#####
# Comprobaciones falsificadas de dirección de origen

# Descartar paquetes que fingen ser originarios de la interfaz receptora
$IPT -A verificación-dirección-fuente -s $IPADDR -j DROP

# Rechazar paquetes que afirman ser de redes privadas

$IPT -A verificación-dirección-fuente -s $CLASS_A -j DROP
$IPT -A verificación-dirección-fuente -s $CLASS_B -j DROP
$IPT -A verificación-dirección-fuente -s $CLASS_C -j DROP
$IPT -A verificación-dirección-fuente -s $CLASS_D_MULTICAST -j DROP
$IPT -A verificación-dirección-fuente -s $CLASS_E_RESERVED_NET -j DROP
$IPT -A verificación-dirección-fuente -s $LOOPBACK -j DROP

$IPT -A verificación-dirección-fuente -s 0.0.0.0/8 -j DROP $IPT -A verificación-
dirección-fuente -s 169.254.0.0/16 -j DROP $IPT -A verificación-dirección-fuente -s 192.0
.2.0/24 -j CAIDA

```

```
#####
# Comprobaciones de puerto y dirección de destino incorrectas

# Bloquear transmisiones dirigidas desde Internet

$IPT -A verificación-dirección-destino -d $BROADCAST_DEST -j DROP $IPT -A verificación-dirección-
destino -d $SUBNET_BASE -j DROP $IPT -A verificación-dirección-destino -d $SUBNET_BROADCAST
-j DROP $IPT -j Un control de dirección de destino! -pupp\

        -d $CLASS_D_MULTICAST -j DROP

#####
# Reglas de registro antes del abandono por la política predeterminada

# reglas ICMP

$IPT -A EXT-iniciar sesión -p icmp \! --icmp-type
        echo-request -m límite -j LOG

# reglas TCP

$IPT -A EXT-iniciar sesión -p tcp \ --dport 0:19
        -j LOG

# Saltar ftp, telnet, ssh
$IPT -A EXT-iniciar sesión -p tcp \ --dport 24 -j
        LOG

# Saltar smtp
$IPT -A EXT-iniciar sesión -p tcp \ --dport 26:78
        -j LOG

# Saltar dedo, www
$IPT -A EXT-iniciar sesión -p tcp \ --dport
        81:109 -j LOG

# Saltar pop-3, sunrpc
$IPT -A EXT-iniciar sesión -p tcp \ --dport
        112:136 -j LOG

# Saltar NetBIOS
$IPT -A EXT-iniciar sesión -p tcp \ --dport
        140:142 -j LOG

# Saltar imagen
$IPT -A EXT-iniciar sesión -p tcp \ --dport
        144:442 -j LOG

# Omitir web_segura/SSL
$IPT -A EXT-iniciar sesión -p tcp \ --dport
        444:65535 -j REGISTRO

#reglas UDP

$IPT -A EXT-iniciar sesión -p udp \ --dport
        0:110 -j LOG
```

```

# Saltar sunrpc
$IPT -A EXT-iniciar sesión -p udp \ --dport
112:160 -j LOG

# Saltar snmp
$IPT -A EXT-iniciar sesión -p udp \ --dport
163:634 -j LOG

# Omitir NFS montado
$IPT -A EXT-iniciar sesión -p udp \ --dport
636:5631 -j LOG

# Saltar pcAnywhere
$IPT -A EXT-iniciar sesión -p udp \ --dport
5633:31336 -j REGISTRO

# Omitir los puertos predeterminados de traceroute
$IPT -A EXT-iniciar sesión -p udp \
--sport $TRACEROUTE_SRC_PORTS \ --dport
$TRACEROUTE_DEST_PORTS -j REGISTRO

# Saltar el resto
$IPT -A EXT-iniciar sesión -p udp \
--dport 33434:65535 -j REGISTRO

# paquetes salientes

# No registrar paquetes de destino inalcanzables ICMP salientes rechazados
$IPT -A EXT-cierre de sesión -p icmp \
--icmp-type destino-inalcanzable -j DROP

$IPT -A EXT-cierre de sesión -j LOG

#####
# Instale las cadenas definidas por el usuario en el integrado
# Cadenas de ENTRADA y SALIDA

# Si es TCP: compruebe si hay patrones comunes de estado TCP de escaneo sigiloso
$IPT -A INPUT -p tcp -j tcp-state-flags
$IPT -A SALIDA -p tcp -j tcp-state-flags

si [ "$CONEXIÓN_SEGUIMIENTO" = "1" ]; luego
    # Omita los filtros de firewall para intercambios establecidos
    $IPT -A INPUT -j seguimiento de conexión
    $IPT -A SALIDA -j seguimiento de conexión
ser

si [ "$DHCP_CLIENT" = "1" ]; luego
    $IPT -A INPUT -i $INTERNET -p udp \ --sport 67 --dport 68
    -j respuesta-servidor-dhcp-remoto
    $IPT -A SALIDA -o $INTERNET -p udp \ --sport 68 --dport
    67 -j local-dhcp-client-query
ser

# Prueba de direcciones de origen y destino ilegales en los paquetes entrantes $IPT -A INPUT ! -p tcp -j verificación de
dirección de origen $IPT -A INPUT -p tcp --syn -j verificación de dirección de origen $IPT -A INPUT -j verificación de
dirección de destino

```

```
# Prueba de direcciones de destino ilegales en paquetes salientes
$IPT -A SALIDA -j verificación-dirección-destino

# Comience las pruebas de firewall estándar para los paquetes dirigidos a este host
$IPT -A ENTRADA -i $INTERNET -d $IPADDR -j Entrada EXT

# Tráfico de multidifusión #####
¡ELIJA SI DESCARGAR O ACEPTAR!
$IPT -A INPUT -i $INTERNET -p udp -d $CLASS_D_MULTICAST -j [ DROP | ACEPTAR ]
$IPT -A SALIDA -o $INTERNET -p udp -s $IPADDR -d $CLASS_D_MULTICAST \ -j [ DROP | ACEPTAR ]

# Comience las pruebas de firewall estándar para los paquetes enviados desde este host.
# La suplantación de la dirección de origen por parte de este host no está permitida debido a la # prueba
de la dirección de origen en esta regla.
$IPT -A SALIDA -o $INTERNET -s $IPADDR -j Salida EXT

# Registre cualquier cosa de interés que falle, # antes de que la política
predeterminada descarte el paquete.
$IPT -A INPUT -j EXT-log-in $IPT -A OUTPUT
-j EXT-log-out

salida 0
```

nftables Firewall del Capítulo 6

Recuerde que en el Capítulo 6 construimos un cortafuegos nftables optimizado que usaba varios archivos diferentes para declarar variables y crear reglas. Esta sección muestra el contenido de cada uno de esos archivos. El único que debe ejecutarse es el `rc.firewall` principal. El resto debe estar en el mismo directorio.

Aquí están los contenidos de `rc.firewall` :

```
#!/bin/sh

NFT = "/usr/local/sbin/nft" # Ubicación de nft en su sistema

# Habilitar protección de eco de transmisión eco 1 > /
proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Deshabilite los paquetes enrutados de origen
para f en /proc/sys/net/ipv4/conf/*/accept_source_route; hacer echo 0 > $f hecho

# Habilitar protección de cookies TCP SYN echo 1 > /
proc/sys/net/ipv4/tcp_syncookies
# Deshabilitar la aceptación de redirección ICMP para f
en /proc/sys/net/ipv4/conf/*/accept_redirects; hacer echo 0 > $f hecho

# No envíe mensajes de redirección para f en /
proc/sys/net/ipv4/conf/*/send_redirects; do echo 0 > $f done # Soltar paquetes
falsificados que llegan a una interfaz, que, si se responde, # daría como resultado
que la respuesta saliera por una interfaz diferente.
```

```
para f en /proc/sys/net/ipv4/conf/*/rp_filter; do echo 1 > $f done # Registra
paquetes con direcciones imposibles. para f en /proc/sys/net/ipv4/
conf/*/log_martians; haz echo 1 > $f hecho
```

```
para i en `tablas de lista de $ NFT | awk '{imprimir $2}'`
```

```
hacer
```

```
echo "Enjuagar ${i}"
```

```
$NFT vaciar la tabla ${i} para j
```

```
en `"$NFT list table ${i} | cadena grep | awk '{imprimir $2}'`
```

```
hacer
```

```
echo "...Eliminando la cadena ${j} de la tabla ${i}"
```

```
$ Cadena de eliminación de NFT $ {i} $ {j}
```

```
hecho
```

```
echo "Eliminando ${i}"
```

```
$ Tabla de eliminación de NFT $ {i}
```

```
hecho
```

```
if [ "$1" = "stop" ] luego echo
```

```
"¡Firewall completamente
```

```
detenido! ADVERTENCIA: ESTE HOST NO TIENE FIREWALL FUNCIONANDO". exit 0 fi $NFT -f tablas de configuración
```

```
$NFT -f política de host local $NFT -f política de estado de conexión
```

```
$NFT -f política no válida
```

```
$NFT -f política-dns
```

```
$NFT -f política-cliente-tcp
```

```
$NFT -f política-servidor-tcp
```

```
$NFT -f icmp-política
```

```
$NFT -f log-policy #default
```

```
drop
```

```
$NFT -f política predeterminada
```

Estos son los contenidos de nft-vars:

```
define int_loopback = lo define
```

```
int_internet = ethN define ip_external =
```

```
define subnet_external = define
```

```
subnet_bcast = define net_loopback =
```

```
127.0.0.0/8 define net_class_a =
```

```
10.0.0.0/8 define net_class_b = 172.16.0.0/16
```

```
define net_class_c = 192.168.0.0/16 define
```

```
net_class_d = 224.0.0.0/4 define net_class_e =
```

```
240.0.0.0/5 define broadcast_src = 0.0.0.0 define
```

```
broadcast_dest = 255.255.255.255
```

```
definir puertos_priv = 0-1023
definir ports_unpriv = 1024-65535

definir nameserver_1 = definir
nameserver_2 = definir nameserver_3
=

definir servidor_smtp =
```

Estos son los contenidos de connectionstate-policy:

```
filtro de tabla {
    entrada de cadena {
        ct estado establecido, relacionado aceptar
        Prefijo de registro no válido de estado de ct "Entrada NO VÁLIDA:" tasa de límite 3/segundo y caída

    }
    salida en cadena {
        ct estado establecido, relacionado aceptar
        Prefijo de registro no válido de estado de ct "Salida NO VÁLIDA: y 3/ "límite de tasa
        segundo caída
    }
}
```

Estos son los contenidos de la política predeterminada:

```
filtro de tabla {
    entrada de cadena {
        soltar
    }
    salida en cadena {
        soltar
    }
}
tabla nat {
    cadena posterior al enrutamiento {
        soltar
    }
    enrutamiento previo de la cadena {
        soltar
    }
}
```

Estos son los contenidos de dns-policy:

```
incluir "nft-vars"
filtro de tabla {
    entrada de cadena {
        ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } udp sport y53 udp dport 53 aceptar

        ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } tcp sport y53 tcp dport $ports_unpriv aceptar

        ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } udp sport y53 udp dport $ports_unpriv aceptar
    }
    salida en cadena {
        ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } udp sport y53 udp dport 53 aceptar
```

```

        ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } tcp sport $ports_unpriv tcp dport 53
accept ip daddr { $nameserver_1,$nameserver_2,$nameserver_3 } udp sport $ports_unpriv udp dport 53 accept }

}

```

Estos son los contenidos de icmp-policy:

```

incluir filtro de tabla "nft-
vars" { entrada de cadena {

        tipo icmp { respuesta de eco, destino inalcanzable, parámetro- y problema, extinción de
la fuente, tiempo excedido} aceptar

        } cadena de salida {
                tipo icmp {petición de eco, problema de parámetro, extinción de fuente} aceptar
        }
}

```

Estos son los contenidos de la política inválida:

```

incluir filtro de tabla "nft-
vars" { entrada de cadena {

        iif $int_internet ip saddr $ip_external drop iif $int_internet ip saddr
$net_class_a drop iif $int_internet ip saddr $net_class_b drop iif
$int_internet ip saddr $net_class_c drop iif $int_internet ip protocol
udp ip daddr $net_class_d accept iif $int_internet ip saddr
$net_class_e drop iif $int_internet ip saddr $net_loopback drop iif $int_internet ip daddr
$broadcast_dest drop

        } cadena de salida { }

}

```

Estos son los contenidos de la política localhost:

```

incluir filtro de tabla "nft-
vars" { entrada de cadena {

        iifname $int_loopback aceptar

        } cadena de salida
        { oifname $int_loopback aceptar
        }

}

```

Estos son los contenidos de la política de registro:

```

incluir filtro de tabla "nft-
vars" { entrada de cadena {

        prefijo de registro "Paquete de ENTRADA descartado:" velocidad límite 3/segundo

        }
}

```



```
        cadena de salida
            {prefijo de registro "Paquete de SALIDA descartado:" velocidad límite 3/segundo
        }
    }
```

Estos son los contenidos de las tablas de configuración:

```
incluir filtro de tabla "nft-
vars" { entrada de cadena {

        tipo filtro gancho entrada prioridad 0;

    } cadena de salida
        { tipo filtro gancho salida prioridad 0;
    }
}
```

Estos son los contenidos de tcp-client-policy:

```
incluir filtro de tabla "nft-
vars" { entrada de cadena
    { } salida de cadena
    { tcp dport

        {21,22,80,110,143,993,995,443} tcp sport $ports_unpriv yaccept ip daddr $server_smtp tcp dport 25 tcp sport
        $ports_unpriv accept

    }
}
```

Estos son los contenidos de tcp-server-policy:

```
incluir filtro de tabla "nft-
vars" { entrada de cadena {

        #ELIGE SI ACEPTAR O BAJAR! ip daddr
        $ip_external tcp sport $ports_unpriv tcp dport {22} [ yacceptar | soltar ]

    } cadena de salida { }
}
```

Esta página se dejó en blanco intencionalmente

C

Glosario

Esta El glosario define los términos y siglas utilizados en el libro. Los términos de varias palabras se ordenan alfabéticamente en el sustantivo principal del término, seguido de una coma y el resto del término.

ACCEPT Una decisión de regla de filtrado de firewall para pasar un paquete a su próximo destino.

política de aceptar todo por defecto Una política que acepta todos los paquetes que no coinciden con una regla de firewall en la cadena. Por lo tanto, la mayoría de las reglas de firewall son reglas DENY que definen las excepciones a la política ACCEPT predeterminada .

ACK El indicador TCP que acusa recibo de un segmento TCP recibido previamente.

puerta de enlace a nivel de aplicación Véase *también* proxy, nivel de aplicación. A menudo denominado ALG, puerta de enlace de nivel de aplicación (o capa) es un término sobrecargado. En términos de firewall, ALG a menudo se refiere a módulos de soporte específicos de la aplicación que inspeccionan la carga útil de la aplicación en busca de direcciones y puertos integrados, y que reconocen flujos secundarios asociados con la sesión.

Puerto de servicio AUTH TCP 113, asociado al servidor de autenticación de usuario identd .

autenticación El proceso de determinar que una entidad es quién o qué dice ser.

autorización El proceso de determinar qué servicios y recursos puede utilizar una entidad.

bastión Véase cortafuegos, bastión.

BIND Berkeley Internet Name Domain, la implementación de Berkeley del DNS protocolo.

BOOTP Bootstrap Protocol, que utilizan las estaciones de trabajo sin disco para descubrir su dirección IP y la ubicación del servidor de arranque, y para iniciar la descarga del sistema a través de TFTP antes de arrancar. BOOTP fue desarrollado para reemplazar RARP.

BOOTPC Servicio UDP puerto 68, asociado a los clientes BOOTP y DHCP.

bootpd El programa de servidor BOOTP.

BOOTPS Servicio UDP puerto 67, asociado a los servidores BOOTP y DHCP.

enrutador de borde Un dispositivo para enrutar paquetes que reside en el borde o límite de un la red.

difusión Un paquete IP que se direcciona y envía a todas las interfaces conectadas a la misma red o subred.

Equipo de Respuesta a Emergencias Informáticas del **CERT** , un centro de coordinación de información y centro de prevención de emergencias de seguridad en Internet formado en el Instituto de Ingeniería de Software de la Universidad Carnegie Mellon después del incidente del Gusano de Internet en 1988.

cadena La lista de reglas que definen qué paquetes pueden entrar y cuáles pueden salir a través de una interfaz de red.

suma de comprobación Un número producido al realizar algún cálculo aritmético en el valor numérico de cada byte en un archivo o paquete. Si se cambia el archivo o se daña el paquete, una segunda suma de verificación producida para el mismo objeto no coincidirá con la suma de verificación original.

estrangulador Véase cortafuegos, estrangulador.

chroot Tanto un programa como una llamada al sistema que define un directorio para que sea la raíz del sistema de archivos, y que luego ejecuta un programa para que se ejecute confinado a ese sistema de archivos virtual.

puerta de enlace de circuito Véase proxy, nivel de circuito.

clase, dirección de red Históricamente, una de las cinco clases de direcciones de red. Una dirección IPv4 es un valor de 32 bits. El espacio de direcciones se divide en direcciones de Clase A a Clase E, según el valor de los primeros 4 bits más significativos en el valor de 32 bits.

El espacio de direcciones de red Clase A mapea 128 redes separadas, cada una de las cuales se dirige a más de 16 millones de hosts. El espacio de direcciones de red Clase B mapea 16 384 redes, cada una de las cuales se dirige a 64 534 hosts. El espacio de direcciones de red Clase C mapea alrededor de 2 millones de redes, cada una de las cuales se dirige a 254 hosts. La clase D se utiliza para direcciones de multidifusión.

La clase E está reservada para fines no especificados o experimentales. Las clases de red se han convertido en gran medida en un artefacto con la introducción de CIDR. Las personas se refieren a ellos por familiaridad y porque sus características de límite de bytes los hacen convenientes para usar en ejemplos.

Enrutamiento entre dominios sin clase CIDR reemplaza el concepto de dirección de red clases para asignación de espacio con el concepto de campos de red de longitud variable. Una extensión conceptual de la idea de las máscaras de subred de longitud variable, CIDR tiene como objetivo mejorar la escalabilidad de la tabla de enrutadores y resolver los problemas de asignación causados por el agotamiento del espacio de direcciones con clase para las organizaciones medianas.

modelo cliente/servidor El modelo de servicios de red distribuidos, en el que un programa centralizado, un servidor, brinda un servicio a programas cliente remotos que solicitan ese servicio, ya sea que el servicio esté recibiendo una copia de una página web, descargando un archivo de un servidor central repositorio, realizar una búsqueda en la base de datos, enviar o recibir correo electrónico,

realizar algún tipo de cálculo en los datos proporcionados por el cliente, o establecer conexiones de comunicación humana entre dos o más personas.

daemon Un servidor de servicios básicos del sistema que se ejecuta en segundo plano.

Agencia de Proyectos de Investigación Avanzada de Defensa **DARPA** .

Capa de enlace de datos En el modelo de referencia OSI, la segunda capa, que representa la entrega de señales de datos punto a punto entre dos dispositivos de red adyacentes, como la entrega de una trama Ethernet desde una computadora a un enrutador externo. (En el modelo de referencia de TCP/IP, esta funcionalidad se incluye como parte de la primera capa, la capa de subred).

política predeterminada Una política para un conjunto de reglas de firewall, ya sea para una cadena de ENTRADA , una SALIDA cadena, o una cadena FORWARD en la tabla de filtros , que define la disposición de un paquete cuando el paquete no coincide con ninguna regla en el conjunto. *Consulte también* la política de aceptar todo de forma predeterminada y la política de denegar todo de forma predeterminada.

Ataque de denegación de servicio (DoS) Un ataque basado en la idea de enviar datos inesperados o inundar un sistema con paquetes para interrumpir o degradar gravemente el servicio, inmovilizar servidores locales hasta el punto de que las solicitudes legítimas no se pueden cumplir, o en el peor de los casos, bloquear un sistema o sistemas por completo.

política denegar todo por defecto Una política que elimina silenciosamente todos los paquetes que no coinciden con una regla de firewall en la cadena. La mayoría de las reglas del cortafuegos son reglas de ACEPTAR que definen las excepciones a la política de DENEGAR predeterminada .

Protocolo de configuración dinámica de host **DHCP** , que se utiliza para asignar dinámicamente direcciones IP y proporcionar información de servidor y enrutador a clientes sin direcciones IP registradas. DHCP fue desarrollado para reemplazar BOOTP.

DMZ La zona desmilitarizada, una red perimetral que contiene máquinas que alojan servicios públicos, separada de una red privada local. Los servidores públicos menos seguros están aislados de la LAN privada.

Servicio de nombres de dominio **DNS** , un servicio de base de datos de Internet global que proporciona principalmente host Mapeo de IP a IP y de IP a host.

DROP Una decisión de regla de filtrado de firewall para descartar silenciosamente un paquete sin devolver ninguna notificación al remitente. DROP es idéntico a DENY en tecnologías de firewall de Linux anteriores.

dual-homed Una computadora que tiene dos interfaces de red. *Véase también* multitarjeta.

dirección asignada dinámicamente dirección IP asignada temporalmente a una interfaz de red cliente por un servidor central, como un servidor DHCP.

Trama Ethernet En una red Ethernet, los datagramas IP se encapsulan en Ethernet marcos

filtro, cortafuegos Una regla de filtrado de paquetes del cortafuegos que define las características de la IP del paquete y los encabezados de transporte que, si coinciden, determina si el paquete se permitirá a través de la interfaz de red o se descartará. Los filtros se definen en términos de campos tales como direcciones de origen y destino de un paquete, puertos de origen y destino, tipo de protocolo, estado de conexión TCP y tipo de mensaje ICMP.

dedo Un programa de búsqueda de información del usuario.

cortafuegos Un dispositivo o grupo de dispositivos que aplica una política de control de acceso entre redes

cortafuegos, bastión Con frecuencia, un cortafuegos que tiene dos o más interfaces de red y es la puerta de enlace o el punto de conexión entre esas redes, normalmente entre un sitio local e Internet. Debido a que un firewall bastión es el único punto de conexión entre redes, el bastión está protegido en la mayor medida posible. De manera más general, un bastión es un firewall al que los sitios remotos tienen acceso directo, ya sea que el host conecte redes o proteja un servidor que brinda servicios públicos.

firewall, choke Un firewall de LAN que tiene dos o más interfaces de red y es el puerto de enlace o punto de conexión entre esas redes. Un lado se conecta a una red perimetral DMZ entre el cortafuegos de estrangulación y un cortafuegos de puerta de enlace bastión. La otra interfaz de red se conecta a una LAN privada interna.

cortafuegos, doble alojamiento Un cortafuegos de puerta de enlace de host único que requiere que los usuarios locales Conéctese específicamente a la máquina de firewall para acceder a Internet o proxies de todos los servicios remotos accesibles desde el sitio. En un sistema de cortafuegos de puerta de enlace de doble base, no se permite el paso de tráfico entre la LAN e Internet.

cortafuegos, host filtrado Casi idéntico a un cortafuegos de doble alojamiento, el cortafuegos de host único no se encuentra directamente entre Internet y la red local. El servidor de seguridad del host filtrado está separado de la red pública por un enrutador intermedio y un filtro de paquetes. Los usuarios locales deben conectarse específicamente a la máquina del cortafuegos para acceder a Internet o utilizar servidores proxy en la máquina del cortafuegos. El enrutador de filtrado garantiza que todo el tráfico entre redes, o al menos tipos específicos de tráfico, pase por el host filtrado. La diferencia entre el firewall de host filtrado y el firewall de host dual radica principalmente en la ubicación del firewall dentro de la red local.

cortafuegos, subred filtrada Un sistema de cortafuegos que incorpora un cortafuegos de puerta de enlace, una red DMZ que alberga servidores públicos y un cortafuegos de estrangulación interno que filtra la LAN tanto de la DMZ como del acceso directo a Internet. Los servicios públicos no están alojados desde el

cortafuegos de estrangulamiento.

inundación, paquete Un ataque de denegación de servicio en el que el host o la red de la víctima recibe más paquetes de un tipo dado de los que la víctima puede acomodar.

reenviar Para enrutar paquetes de una red a otra en el proceso de entregar un paquete de una computadora a otra.

fragmento Un paquete IP que contiene una parte de un segmento TCP.

Protocolo de transferencia de archivos **FTP** . El protocolo y los programas utilizados para copiar archivos entre computadoras en red.

FTP, servicio FTP anónimo accesible para cualquier cliente que solicite el servicio.

FTP, servicio FTP autenticado accesible a cuentas predefinidas, que deben autenticarse antes de utilizar el servicio.

puerta de enlace Una computadora o programa que sirve como conducto o punto de terminación y relevo entre dos redes.

hosts.allow, hosts.deny Los archivos de configuración de los contenedores TCP son /etc/hosts.allow y /etc/hosts.deny.

CÓMO Además de las páginas de manual estándar, Linux incluye documentación en línea proporcionada por el usuario sobre numerosos temas, en muchos idiomas y en múltiples formatos. Los documentos HOWTO son coordinados y mantenidos por el Proyecto de Documentación de Linux.

Protocolo de transferencia de hipertexto **HTTP** , utilizado por servidores web y navegadores.

concentrador Repetidor de señal de hardware que se utiliza para conectar físicamente varios segmentos de red, ampliar la distancia de una red física o conectar segmentos de red de diferentes tipos físicos.

Autoridad de Números Asignados de Internet de la **IANA** .

Protocolo de mensajes de control de Internet **ICMP** . A Estado y control de IP de capa de red mensaje.

identd El servidor de autenticación de usuario (AUTH) .

Protocolo de acceso a mensajes de Internet **IMAP** , utilizado para recuperar correo de hosts de correo que se ejecutan un servidor IMAP.

inetd Un superservidor de red que escucha las conexiones entrantes a los puertos de servicio utilizados por los servidores que administra. Cuando llega una solicitud de conexión, inetd inicia una copia del servidor de solicitudes para manejar la conexión. Por defecto, inetd ha sido reemplazado por una versión extendida llamada xinetd.

Datagrama IP Un paquete de capa de red IP.

ipchains Con la introducción de la implementación más reciente del mecanismo de firewall IPFW en Linux, el programa de administración de firewall que reemplazó a iptwadm. iptables se suministra con un módulo de compatibilidad con ipchains para sitios que desean continuar usando sus scripts de firewall existentes.

Mecanismo de firewall IP **IPFW** , ahora reemplazado por Netfilter.

ipfwadm Antes de la introducción de ipchains, el programa de administración del cortafuegos IPFW de Linux. iptables se suministra con un módulo de compatibilidad con ipfwadm para sitios que desean continuar usando sus scripts de firewall existentes.

iptables El programa de administración de cortafuegos que comienza en el núcleo de la serie 2.4.

klogd El demonio de registro del kernel que recopila mensajes de error y de estado del sistema operativo de los búferes de mensajes del kernel y, junto con syslogd, escribe los mensajes en un archivo de registro del sistema.

LAN Red de área local.

localhost El nombre simbólico que suele darse a la interfaz loopback de una máquina en /etc/hosts.

interfaz de bucle invertido Una interfaz de red de software especial utilizada por el sistema para entregar mensajes de red generados localmente destinados a la máquina local, sin pasar por la interfaz de red de hardware y el controlador de red asociado.

página de manual El formato estándar de documentación en línea de Linux. Las páginas de manual están escritas para casi todos los programas de administración de usuarios y sistemas, así como llamadas al sistema, llamadas a la biblioteca, tipos de dispositivos y formatos de archivos del sistema.

enmascaramiento El proceso de reemplazar la dirección de origen local de un paquete saliente con la del firewall o la máquina de la puerta de enlace para que las direcciones IP de la LAN permanezcan ocultas. En el mecanismo de firewall de IPFW, el enmascaramiento se refería a la funcionalidad NAT de origen implementada en Linux. En Netfilter, el enmascaramiento se refiere a una forma especializada de NAT de origen para usar con conexiones a las que se les asigna dinámicamente direcciones IP temporales que tienden a cambiar con cada conexión.

MD5 Algoritmo de suma de control criptográfico utilizado para garantizar la integridad de los datos mediante la creación firmas digitales, llamadas resúmenes de mensajes, de objetos.

Unidad de transmisión máxima de MTU, el tamaño máximo de paquete basado en el subyacente la red.

multidifusión Un paquete IP dirigido especialmente a una dirección IP de multidifusión Clase D. Los clientes de multidifusión se registran con los enrutadores intermedios para recibir paquetes dirigidos a una dirección de multidifusión particular.

multitarjeta Una computadora que tiene dos o más interfaces de red. Véase también doble base.

servidor de nombres, primario Un servidor autorizado para un dominio o una zona del espacio de dominio. El servidor mantiene una base de datos completa de nombres de host y direcciones IP para esta zona.

servidor de nombres, secundario Una copia de seguridad o igual a un servidor de nombres primario.

Traducción de direcciones de red NAT, el proceso de reemplazar la dirección de origen o de destino de un paquete con la de alguna otra interfaz de red. NAT está destinado principalmente a

permitir el tráfico entre espacios de direcciones de red incompatibles, como entre Internet y una LAN a la que se asignan direcciones privadas internamente.

Netfilter El mecanismo de firewall incluido a partir del kernel de Linux 2.4.

nft El programa de administración para un cortafuegos nftables .

nftables El mecanismo de firewall incluido a partir del kernel de Linux 3.13.

netstat Un programa que informa varios tipos de estado de la red en función de los diversos tablas del núcleo relacionadas con la red.

Capa de red En el modelo de referencia OSI, la tercera capa, que representa la comunicación de extremo a extremo entre dos computadoras, como el enrutamiento y la entrega de un diagrama de datos IP desde una computadora de origen a una computadora de destino externa. En el modelo de referencia de TCP/IP, esto se conoce como la segunda capa, la capa de Internet.

Sistema de archivos de red **NFS** , utilizado para compartir sistemas de archivos entre computadoras en red.

Nmap Una herramienta de auditoría de seguridad de red (es decir, exploración de puertos) que incluye muchas de las técnicas de exploración más nuevas que se utilizan en la actualidad.

Protocolo de transferencia de noticias de red **NNTP** , utilizado por Usenet.

Protocolo de tiempo de red **NTP** , utilizado por ntpd y ntpdate.

Modelo de **referencia OSI (Open System Interconnection)** Un modelo de siete capas desarrollado por la Organización Internacional de Normalización (ISO) para proporcionar un marco o guía para los estándares de interconexión de redes.

OSPF El protocolo de enrutamiento Open Shortest Path First para TCP/IP, que es el protocolo de enrutamiento más utilizado en la actualidad.

paquete Un datagrama de red IP.

filtrado de paquetes Véase cortafuegos.

PATH La variable ambiental del shell que define en qué directorios debe buscar el shell comandos ejecutables no calificados y en qué orden debe buscar el shell en esos directorios.

peer-to-peer Un modo de comunicación utilizado para la comunicación entre dos programas de servidor. Un protocolo de comunicación punto a punto es a menudo, pero no siempre, diferente del protocolo utilizado para comunicarse entre el servidor y un cliente.

Capa física En el modelo de referencia OSI, la primera capa, que representa el medio físico utilizado para transportar las señales entre dos dispositivos de red adyacentes, como cobre, fibra óptica, radio por paquetes o infrarrojos. En el modelo de referencia TCP/IP, esto se incluye como parte de la primera capa, la capa de subred.

358 Apéndice C Glosario

ID de proceso PID, que es el identificador numérico único de un proceso en el sistema, generalmente asociado con la ranura del proceso en la tabla de procesos del sistema.

ping Una herramienta simple de análisis de red utilizada para determinar si un host remoto es accesible y responde. ping envía un mensaje de solicitud de eco ICMP. El host del destinatario devuelve un mensaje de respuesta de eco ICMP en respuesta.

Protocolo de oficina de correos **POP**, utilizado para recuperar correo de hosts de correo que ejecutan un servidor POP.

puerto En TCP o UDP, el designador numérico de una comunicación de red particular canal. Las asignaciones de puertos son administradas por IANA. Algunos puertos se asignan a protocolos de comunicación de aplicaciones particulares como parte del estándar de protocolo. Algunos puertos están registrados como asociados con un servicio en particular por convención. Algunos puertos no están asignados y pueden asignarse dinámicamente para que los utilicen clientes y programas de usuario:

• **Privilegiado:** un puerto en el rango de 0 a 1023. Muchos de estos puertos están asignados a protocolos de aplicación por estándar internacional. En un sistema Linux, el acceso a los puertos privilegiados requiere privilegios a nivel del sistema.

• **Sin privilegios:** un puerto en el rango de 1024 a 65535. Algunos de estos puertos están registrados para que los usen ciertos programas por convención. Cualquier puerto en este rango puede ser utilizado por un programa cliente para establecer una conexión con un servidor en red.

exploración de puertos Una sonda de todos o un conjunto de puertos de servicio de una computadora host, generalmente puertos de servicio que a menudo se asocian con vulnerabilidades de seguridad.

portmap Un demonio administrador de RPC, usado para mapear entre un servicio RPC particular número al que un cliente solicita acceso y el puerto de servicio al que está vinculado el servidor asociado.

sonda Enviar algún tipo de paquete a un puerto de servicio en una computadora host. El propósito de una sonda es determinar si se genera una respuesta desde el host de destino.

proxy Un programa que crea y mantiene una conexión de red en nombre de otro programa, proporcionando un conducto a nivel de aplicación entre un cliente y un servidor. El cliente y el servidor reales no tienen comunicación directa. El proxy parece ser el servidor para el programa cliente y parece ser el cliente para el programa servidor. Los proxies de aplicaciones generalmente se clasifican como pasarelas de aplicaciones o pasarelas de circuitos.

proxy, nivel de aplicación Un servidor proxy para un servicio en particular. Los proxies de puerta de enlace a nivel de aplicación entienden el protocolo de aplicación particular que representan. El proxy es capaz de inspeccionar la carga útil de la aplicación y tomar decisiones basadas en la información a nivel de la aplicación, en lugar de tomar decisiones simplemente a nivel de IP y transporte.

proxy, a nivel de circuito Un servidor proxy que se puede implementar como aplicaciones separadas para cada servicio que se utiliza como proxy o como un único relé de conexión generalizado. Un proxy a nivel de circuito no tiene ningún conocimiento específico sobre los protocolos de aplicación.

El proxy toma decisiones basadas en la misma información de transporte y de IP que un cortafuegos de filtrado de paquetes, con la posible adición de cierta cantidad de funcionalidad de autenticación de usuario.

Calidad de servicio **QoS** .

Protocolo de resolución de dirección inversa **RARP** , desarrollado para permitir que las máquinas sin disco pregunte a los servidores por su dirección IP en función de su dirección de hardware MAC.

RECHAZAR Una decisión de regla de filtrado de firewall para descartar un paquete y devolver un mensaje de error a el remitente.

resolver El lado del cliente de DNS. El resolutor se implementa como código de biblioteca que está vinculado a programas que requieren acceso a la red. El archivo de configuración del cliente DNS es `/etc/resolv.conf`.

Solicitud de comentarios de **RFC** , una nota o memorando publicado a través de Internet Society o el Grupo de Trabajo de Ingeniería de Internet. Algunos RFC se convierten en estándares. Los RFC generalmente se refieren a un tema relacionado con Internet o el conjunto de protocolos TCP/IP.

Protocolo de información de enrutamiento **RIP** , un protocolo de enrutamiento más antiguo que aún se usa en la actualidad, especialmente dentro de una LAN grande. El demonio enrutado utiliza RIP.

RPC Llamada a procedimiento remoto.

regla Véase cortafuegos y filtro, cortafuegos.

runlevel Un concepto de arranque y estado del sistema tomado de System V UNIX. Un sistema ni Funciona normalmente en uno de los niveles de ejecución 2, 3 o 5. El nivel de ejecución 3 es el estado del sistema multiusuario, normal y predeterminado. El nivel de ejecución 2 es similar al nivel de ejecución 3, sin los servicios xinetd, portmap o NFS en ejecución. El nivel de ejecución 5 es el mismo que el nivel de ejecución 3, con la adición de X Window Display Manager, que presenta una pantalla de inicio de sesión y selección de host basada en X.

host filtrado Véase cortafuegos, host filtrado.

subred filtrada Véase cortafuegos, subred filtrada.

script Un archivo ASCII que puede contener comandos de programa shell o Linux. Estos scripts son interpretados por programas shell como sh, csh, bash, zsh o ksh, o por programas como perl, awk o sed.

segmento, TCP Un mensaje TCP.

setgid Un programa que, cuando se ejecuta, asume el ID de grupo del propietario del programa en lugar del ID de grupo del proceso que ejecuta el programa.

setuid Un programa que, cuando se ejecuta, asume la ID de usuario del propietario del programa en lugar de la ID de usuario del proceso que ejecuta el programa.

shell Un intérprete de comandos, como sh, csh, bash, zsh y ksh.

360 Apéndice C Glosario

Protocolo simple de transferencia de correo SMTP , utilizado para intercambiar correo entre servidores de correo y entre programas de correo y servidores de correo.

Protocolo simple de administración de red SNMP , utilizado para administrar la configuración de dispositivos de red desde una estación de trabajo remota.

socket El punto de conexión de red único definido por el emparejamiento de una dirección IP con un puerto de servicio TCP o UDP particular.

SOCKS Un paquete proxy de puerta de enlace de circuito disponible de NEC.

suplantación de identidad, dirección de origen Falsificación de la dirección de origen en el encabezado de un paquete IP para que parezca ser la de alguna otra dirección.

Protocolo SSH Secure Shell, utilizado para conexiones de red cifradas y autenticadas.

Protocolo SSL Secure Socket Layer, utilizado para la comunicación cifrada. SSL es utilizado más comúnmente por servidores web y navegadores para intercambiar información personal para comercio electrónico

dirección asignada estáticamente Direcciones IP codificadas y asignadas de forma permanente, ya sean direcciones registradas públicamente o direcciones de clase privada.

capa de subred En el modelo de referencia TCP/IP, la primera capa, que representa los medios físicos utilizados para transportar las señales entre dos dispositivos de red adyacentes y la entrega de señales de datos punto a punto entre dos dispositivos de red adyacentes, como la entrega de una trama Ethernet desde una computadora a un enrutador externo.

SYN El indicador de solicitud de sincronización de conexión TCP. Un mensaje SYN es el primer mensaje enviado desde un programa que busca abrir una conexión con otro programa en red.

syslog.conf El archivo de configuración del demonio de registro del sistema.

syslogd El daemon de registro del sistema, que recopila mensajes de error y de estado generados por programas del sistema que publican mensajes utilizando la llamada al sistema syslog() .

Protocolo de control de transmisión TCP , utilizado para conexiones de red continuas y confiables entre dos programas.

Modelo de referencia de TCP/IP Un modelo de comunicación de red informal desarrollado cuando TCP/IP se convirtió en el estándar de facto para la comunicación de Internet entre las máquinas UNIX a finales de los años setenta y principios de los ochenta. En lugar de ser un ideal académico formal, el modelo de referencia TCP/IP se basa en lo que los fabricantes y desarrolladores finalmente acordaron para la comunicación a través de Internet.

tcp_wrapper Un esquema de autorización utilizado para controlar qué servicios locales están disponibles para qué hosts remotos en la red.

TFTP Trivial File Transfer Protocol, el protocolo utilizado para descargar una imagen de arranque en un estación de trabajo sin disco o enrutador. El protocolo es una versión simplificada de FTP basada en UDP.

apretón de manos de tres vías El protocolo de establecimiento de conexión TCP. Cuando un programa cliente envía su primer mensaje a un servidor, el mensaje de solicitud de conexión, el SYN se establece la bandera y se acompaña de un número de secuencia de sincronización que el cliente utilizará como punto de partida para numerar el resto de mensajes que enviará el cliente. El servidor responde con un acuse de recibo (ACK) al mensaje SYN , junto con su propia solicitud de sincronización (SYN). El servidor incluye el número de secuencia del cliente incrementado por el número de bytes de datos contiguos recibidos, más 1. El propósito del acuse de recibo es reconocer el mensaje al que se refirió el cliente por su número de secuencia. Al igual que con el primer mensaje del cliente, el indicador SYN va acompañado de un número de secuencia de sincronización. El servidor pasa su propio número de secuencia inicial para su mitad de la conexión. El cliente responde con un ACK del SYN-ACK del servidor, incrementando el número de secuencia del servidor por el número de bytes de datos contiguos recibidos, más 1 para indicar la recepción del mensaje. Se establece la conexión.

Tipo de servicio de **TOS** , el campo en el encabezado del paquete IP destinado a proporcionar una pista de la política de enrutamiento preferida o la preferencia de enrutamiento de paquetes.

traceroute Una herramienta de análisis de red utilizada para determinar la ruta de una computadora a otro a través de la red.

Capa de transporte En el modelo de referencia OSI, la cuarta capa, que representa la comunicación de extremo a extremo entre dos programas, como la entrega de un paquete desde un programa cliente a un programa servidor. En el modelo de referencia de TCP/IP, esto se conoce como la tercera capa, también la capa de transporte. Sin embargo, la abstracción del nivel de transporte de la capa 3 de TCP/IP incluye el concepto de la capa de sesión de la capa 5 de OSI, que incluye los conceptos de un intercambio de mensajes ordenado y sincronizado.

TTL Time to Live, un campo de encabezado de paquete IP que es un recuento máximo del número de enrutadores por los que puede pasar el paquete antes de llegar a su destino.

Protocolo de datagramas de usuario **UDP** , utilizado para enviar mensajes de red individuales entre programas, sin ninguna garantía de entrega ni orden de entrega.

unidifusión Un paquete IP enviado punto a punto, desde la interfaz de red de una computadora a de otro

Protocolo de copia **UUCP** de UNIX a UNIX.

Objetos del sistema de archivos **legibles en todo el mundo** (**archivos**, directorios y sistemas de archivos completos) que son legible por cualquier cuenta o programa en el sistema.

Objetos de sistema de archivos **de escritura mundial** (**archivos**, directorios y sistemas de archivos completos) que son escritura por cualquier cuenta o programa en el sistema.

X Windows El sistema de visualización de ventanas de la interfaz gráfica de usuario de Linux.

Esta página se dejó en blanco intencionalmente



Libre de GNU

Documentación Licencia1

Versión 1.3, 3 de noviembre de 2008

0. Preámbulo

El propósito de esta Licencia es hacer que un manual, libro de texto u otro documento funcional y útil sea "libre" en el sentido de libertad: para asegurar a todos la libertad efectiva de copiarlo y redistribuirlo, con o sin modificarlo, ya sea comercial o no comercialmente. .

En segundo lugar, esta Licencia preserva para el autor y el editor una forma de obtener crédito por su trabajo, sin ser considerado responsable de las modificaciones realizadas por otros.

Esta Licencia es una especie de "copyleft", lo que significa que los trabajos derivados del documento deben ser libres en el mismo sentido. Complementa la Licencia Pública General GNU, que es una licencia copyleft diseñada para software libre.

Hemos diseñado esta Licencia para usarla en manuales de software libre, porque el software libre necesita documentación libre: un programa libre debe venir con manuales que brinden las mismas libertades que el software. Pero esta Licencia no se limita a los manuales de software; se puede utilizar para cualquier trabajo textual, independientemente del tema o si se publica como un libro impreso. Recomendamos esta Licencia principalmente para trabajos cuyo propósito sea instructivo o de referencia.

1. Aplicabilidad y Definiciones

Esta Licencia se aplica a cualquier manual u otro trabajo, en cualquier medio, que contenga un aviso colocado por el titular de los derechos de autor que indique que puede distribuirse según los términos de esta Licencia. Dicho aviso otorga una licencia libre de regalías en todo el mundo, de duración ilimitada, para usar ese trabajo bajo las condiciones establecidas en este documento. El "Documento", a continuación, se refiere a cualquier manual u obra de este tipo. Cualquier miembro del público es un licenciatario y se trata de "usted". Usted acepta la licencia si copia, modifica o distribuye el trabajo de una manera que requiera permiso bajo la ley de derechos de autor.

1. Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Todos pueden copiar y distribuir copias textuales de este documento de licencia, pero no se permite cambiarlo.

Una "Versión Modificada" del Documento significa cualquier trabajo que contenga el Documento o una parte del mismo, ya sea copiado palabra por palabra, o con modificaciones y/o traducido a otro idioma.

Una "Sección Secundaria" es un apéndice con nombre o una sección preliminar del Documento que trata exclusivamente de la relación de los editores o autores del Documento con el tema general del Documento (o con asuntos relacionados) y no contiene nada que pueda caer directamente dentro de ese tema general. (Por lo tanto, si el Documento es en parte un libro de texto de matemáticas, una Sección Secundaria puede no explicar ninguna matemática.) La relación podría ser una cuestión de conexión histórica con el tema o con asuntos relacionados, o de carácter legal, comercial, filosófico, ético. o posición política respecto de ellos.

Las "Secciones Invariantes" son ciertas Secciones Secundarias cuyos títulos se designan, como los de las Secciones Invariantes, en el aviso que dice que el Documento se publica bajo esta Licencia. Si una sección no se ajusta a la definición anterior de Secundaria, no se permite designarla como Invariable. El Documento puede contener cero Secciones Invariantes.

Si el Documento no identifica ninguna Sección Invariable, entonces no hay ninguna.

Los "Textos de Portada" son ciertos pasajes cortos de texto que se enumeran, como Textos de Portada o Textos de Contraportada, en el aviso que dice que el Documento se publica bajo esta Licencia. Un texto de portada puede tener un máximo de 5 palabras y un texto de contraportada puede tener un máximo de 25 palabras.

Una copia "Transparente" del Documento significa una copia legible por máquina, representada en un formato cuya especificación está disponible para el público en general, que es adecuada para revisar el documento directamente con editores de texto genéricos o (para imágenes compuestas de píxeles) pintura genérica programas o (para dibujos) algún editor de dibujo ampliamente disponible, y que sea adecuado para la entrada a formateadores de texto o para la traducción automática a una variedad de formatos adecuados para la entrada a formateadores de texto. Una copia realizada en un formato de archivo transparente cuyo marcado, o ausencia de marcado, se ha dispuesto para impedir o desalentar la modificación posterior por parte de los lectores no es transparente. Un formato de imagen no es transparente si se usa para una cantidad sustancial de texto. Una copia que no es "Transparente" se llama "Opaca".

Los ejemplos de formatos adecuados para copias transparentes incluyen ASCII simple sin marcado, formato de entrada Texinfo, formato de entrada LaTeX, SGML o XML usando un DTD disponible públicamente, y HTML, PostScript o PDF simple que cumple con los estándares y está diseñado para la modificación humana. Los ejemplos de formatos de imagen transparente incluyen PNG, XCF y JPG. Los formatos opacos incluyen formatos patentados que solo pueden leerse y editarse mediante procesadores de texto patentados, SGML o XML para los cuales las herramientas de procesamiento y/o DTD generalmente no están disponibles, y el HTML, PostScript o PDF generado por máquina producido por algunos procesadores de texto. solo para fines de salida.

La "Página de título" significa, para un libro impreso, la página de título en sí, más las páginas siguientes que sean necesarias para contener, de manera legible, el material que esta Licencia requiere que aparezca en la página de título. Para obras en formatos que no tienen ninguna página de título como tal, "Página de título" significa el texto cerca de la aparición más destacada del título de la obra, que precede al comienzo del cuerpo del texto.

El "editor" significa cualquier persona o entidad que distribuya copias del Documento al público.

Una sección "Titulado XYZ" significa una subunidad nombrada del Documento cuyo título ya sea precisamente XYZ o contiene XYZ entre paréntesis a continuación del texto que traduce XYZ en otro idioma. (Aquí XYZ representa un nombre de sección específico que se menciona a continuación, como "Agradecimientos", "Dedicatorias", "Aprobaciones" o "Historia"). "Conservar el título" de dicha sección cuando modifica el Documento significa que sigue siendo una sección "Titulado XYZ" según esta definición.

El Documento puede incluir Exenciones de garantía junto al aviso que establece que esta Licencia se aplica al Documento. Se considera que estas renunciaciones de garantía se incluyen por referencia en esta licencia, pero solo en lo que respecta a la renuncia de garantías: cualquier otra implicación que estas renunciaciones de garantía puedan tener es nula y no tiene ningún efecto sobre el significado de esta licencia.

2. Copia textual

Puede copiar y distribuir el Documento en cualquier medio, ya sea comercial o no comercial, siempre que esta Licencia, los avisos de derechos de autor y el aviso de licencia que dice que esta Licencia se aplica al Documento se reproduzcan en todas las copias y que no agregue ningún otro condición alguna a las de esta Licencia. No puede utilizar medidas técnicas para obstruir o controlar la lectura o la copia posterior de las copias que realice o distribuya.

Sin embargo, usted puede aceptar compensación a cambio de las copias. Si distribuye un número suficientemente grande de copias, también debe seguir las condiciones de la sección 3.

También puede prestar copias, en las mismas condiciones establecidas anteriormente, y puede exhibir copias públicamente.

3. Copiar en cantidad

Si publica copias impresas (o copias en medios que comúnmente tienen portadas impresas) del Documento, que suman más de 100, y el aviso de licencia del Documento requiere Textos de Portada, debe adjuntar las copias en portadas que lleven, de forma clara y legible, todos estos Textos de portada: Textos de portada en la portada y Textos de contraportada en la contraportada. Ambas portadas también deben identificarlo de manera clara y legible como el editor de estas copias. La portada debe presentar el título completo con todas las palabras del título igualmente prominentes y visibles. Además, puede agregar otro material en las portadas. La copia con cambios limitados a las cubiertas, siempre que conserven el título del Documento y cumplan con estas condiciones, puede ser tratada como copia literal en otros aspectos.

Si los textos requeridos para cualquiera de las portadas son demasiado voluminosos para caber de manera legible, debe colocar los primeros de la lista (tantos como quepan razonablemente) en la portada real y continuar con el resto en las páginas adyacentes.

Si publica o distribuye copias opacas del Documento que numera más de 100, debe incluir una copia transparente legible por máquina junto con cada copia opaca, o indicar en o con cada copia opaca una ubicación de red informática desde la cual el público general que usa la red tiene acceso para descargar usando protocolos de red estándar públicos a Copia transparente completa del Documento, libre de material añadido.

Si utiliza la última opción, debe tomar medidas razonablemente prudentes, cuando comience a distribuir copias opacas en cantidad, para asegurarse de que esta copia transparente permanezca accesible en el lugar indicado hasta al menos un año después de la última vez que distribuya una copia. Copia opaca (directamente o a través de sus agentes o distribuidores) de esa edición al público.

Se solicita, pero no se requiere, que se comunique con los autores del Documento mucho antes de redistribuir una gran cantidad de copias, para darles la oportunidad de brindarle una versión actualizada del Documento.

4. Modificaciones

Usted puede copiar y distribuir una Versión Modificada del Documento bajo las condiciones de las secciones 2 y 3 anteriores, siempre que publique la Versión Modificada bajo precisamente esta Licencia, con la Versión Modificada desempeñando el papel del Documento, autorizando así la distribución y modificación de la Versión Modificada a quien posea una copia de la misma. Además, debe hacer estas cosas en la versión modificada:

- A. Usar en la Portada (y en las portadas, si las hay) un título distinto al de la Documento, y de las de versiones anteriores (que deberían, si las hubiere, estar listadas en la sección Historial del Documento). Puede usar el mismo título que una versión anterior si el editor original de esa versión da permiso.
- B. Listar en la Portada, como autores, una o más personas o entidades responsables de la autoría de las modificaciones en la Versión Modificada, junto con al menos cinco de los autores principales del Documento (todos sus autores principales, si tiene menos de cinco), a menos que lo liberen de este requisito.
- C. Indique en la Portada el nombre del editor de la Versión Modificada, como editor.
- D. Conservar todos los avisos de derechos de autor del Documento.
- E. Agregue un aviso de derechos de autor apropiado para sus modificaciones junto a los demás avisos de derechos de autor.
- F. Incluir, inmediatamente después de los avisos de derechos de autor, un aviso de licencia que otorgue al público permiso para usar la Versión Modificada según los términos de esta Licencia, en el formato que se muestra en el Anexo a continuación.
- G. Conservar en ese aviso de licencia las listas completas de las Secciones Invariantes y los Textos de Cubierta requeridos que se proporcionan en el aviso de licencia del Documento.
- H. Incluir una copia inalterada de esta Licencia. • I.

Preservar la sección titulada "Historia", Preservar su Título y agregarle un ítem indicando al menos el título, el año, los nuevos autores y el editor de la Versión Modificada tal como se indica en la Portada. Si no hay una sección titulada "Historia" en el Documento, cree una que indique el título, el año, los autores y el editor del Documento tal como se indica en su Página de título, luego agregue un elemento que describa la Versión modificada como se indica en la oración anterior.

ÿ J. Preservar la ubicación de la red, si la hubiere, proporcionada en el Documento para el acceso público a una copia Transparente del Documento, así como las ubicaciones de la red proporcionadas en el Documento para las versiones anteriores en las que se basó. Estos pueden colocarse en la sección "Historial". Puede omitir una ubicación de red para un trabajo que se publicó al menos cuatro años antes que el Documento en sí, o si el editor original de la versión a la que se refiere da permiso.

ÿ K. Para cualquier sección titulada "Agradecimientos" o "Dedicatorias", Preservar la Título de la sección, y conservar en la sección toda la sustancia y tono de cada uno de los agradecimientos y/o dedicatorias de los contribuyentes que allí se dan.

ÿ L. Conservar todas las Secciones Invariantes del Documento, inalteradas en su texto y en sus títulos. Los números de sección o su equivalente no se consideran parte de los títulos de sección.

ÿ M. Elimine cualquier sección titulada "Respaldos". Tal sección puede no estar incluida en la Versión Modificada.

ÿ N. No cambie el título de ninguna sección existente para que se titule "Respaldos" o entre en conflicto en título con cualquier Sección

Invariable. ÿ O. Conservar las renunciaciones de garantía.

Si la Versión modificada incluye nuevas secciones preliminares o apéndices que califican como Secciones secundarias y no contienen material copiado del Documento, puede, a su elección, designar algunas o todas estas secciones como invariables. Para hacer esto, agregue sus títulos a la lista de Secciones Invariantes en el aviso de licencia de la Versión Modificada. Estos títulos deben ser distintos de cualquier otro título de sección.

Puede agregar una sección titulada "Aprobaciones", siempre que no contenga más que aprobaciones de su Versión modificada por parte de varias partes, por ejemplo, declaraciones de revisión por pares o que el texto haya sido aprobado por una organización como la definición autorizada de un estándar.

Puede agregar un pasaje de hasta cinco palabras como Texto de portada y un pasaje de hasta 25 palabras como Texto de contraportada al final de la lista de Textos de portada en la Versión modificada. Solo un pasaje del Texto de la Portada y uno del Texto de la Contraportada pueden ser agregados por (o mediante arreglos hechos por) cualquier entidad. Si el Documento ya incluye un texto de portada para la misma portada, agregado previamente por usted o por arreglo hecho por la misma entidad en cuyo nombre está actuando, no puede agregar otro; pero puede reemplazar el anterior, con el permiso explícito del editor anterior que agregó el anterior.

El(los) autor(es) y editor(es) del Documento no dan permiso por esta Licencia usar sus nombres para publicidad o para afirmar o implicar el respaldo de cualquier Versión Modificada.

5. Combinación de documentos

Puede combinar el Documento con otros documentos publicados bajo esta Licencia, según los términos definidos en la sección 4 anterior para versiones modificadas, siempre que

incluya en la combinación todas las Secciones Invariantes de todos los documentos originales, sin modificar, y enumérelas todas como Secciones Invariantes de su trabajo combinado en su aviso de licencia, y que conserve todas sus Exenciones de Garantía.

El trabajo combinado solo necesita contener una copia de esta Licencia, y múltiples Secciones Invariantes idénticas pueden ser reemplazadas con una sola copia. Si hay varias Secciones Invariantes con el mismo nombre pero contenido diferente, haga que el título de cada sección sea único agregando al final, entre paréntesis, el nombre del autor o editor original de esa sección, si se conoce, o de lo contrario un número único. Haga el mismo ajuste a los títulos de las secciones en la lista de Secciones Invariantes en el aviso de licencia de la obra combinada.

En la combinación, debe combinar cualquier sección titulada "Historia" en los diversos documentos originales, formando una sección titulada "Historia"; combine igualmente las secciones tituladas "Agradecimientos" y las secciones tituladas "Dedicatorias". Debe eliminar todas las secciones tituladas "Aprobaciones".

6. Colecciones de Documentos

Puede hacer una colección que consista en el Documento y otros documentos publicados bajo esta Licencia, y reemplazar las copias individuales de esta Licencia en los diversos documentos con una sola copia que se incluye en la colección, siempre que siga las reglas de esta Licencia. para la copia textual de cada uno de los documentos en todos los demás aspectos.

Puede extraer un solo documento de dicha colección y distribuirlo individualmente. aliado bajo esta Licencia, siempre que inserte una copia de esta Licencia en el documento extraído, y siga esta Licencia en todos los demás aspectos con respecto a la copia literal de ese documento.

7. Agregación con Obras Independientes

Una compilación del Documento o sus derivados con otros documentos u obras separados e independientes, en o sobre un volumen de un medio de almacenamiento o distribución, se denomina "agregado" si los derechos de autor resultantes de la compilación no se utilizan para limitar los derechos legales. de los usuarios de la compilación más allá de lo que permiten las obras individuales. Cuando el Documento se incluye en un agregado, esta Licencia no se aplica a los otros trabajos en el agregado que no son en sí mismos trabajos derivados del Documento.

Si el requisito del Texto de Portada de la sección 3 es aplicable a estas copias del Documento, entonces si el Documento es menos de la mitad del agregado total, los Textos de Portada del Documento pueden colocarse en portadas que encuadren el Documento dentro del agregado, o el equivalente electrónico de las cubiertas si el Documento está en formato electrónico. En caso contrario, deberán aparecer en cubiertas impresas que enmarquen todo el árido.

8. Traducción

La traducción se considera un tipo de modificación, por lo que puede distribuir traducciones del Documento según los términos de la sección 4. Sustitución de secciones invariantes por traducciones

requiere un permiso especial de los propietarios de los derechos de autor, pero puede incluir traducciones de algunas o todas las Secciones invariantes además de las versiones originales de estas Secciones invariantes. Puede incluir una traducción de esta Licencia y todos los avisos de licencia en el Documento, y cualquier Descargo de responsabilidad de la garantía, siempre que también incluya la versión original en inglés de esta Licencia y las versiones originales de esos avisos y descargos de responsabilidad.

En caso de desacuerdo entre la traducción y la versión original de esta Licencia o un aviso o descargo de responsabilidad, prevalecerá la versión original.

Si una sección del Documento se titula "Agradecimientos", "Dedicatorias" o "Sutory", el requisito (sección 4) para preservar su título (sección 1) generalmente requerirá cambiar el título real.

9. Terminación

No puede copiar, modificar, otorgar sublicencias ni distribuir el Documento excepto según lo dispuesto expresamente en esta Licencia. Cualquier intento de copiarlo, modificarlo, otorgar sublicencias o distribuirlo de otro modo es nulo y automáticamente rescindirá sus derechos bajo esta Licencia.

Sin embargo, si cesa toda violación de esta Licencia, entonces su licencia de un titular de derechos de autor en particular se restablece (a) provisionalmente, a menos y hasta que el titular de los derechos de autor rescinda su licencia de manera explícita y definitiva, y (b) permanentemente, si el titular de los derechos de autor no le notifica la violación por algún medio razonable antes de los 60 días posteriores a la cesación.

Además, su licencia de un titular de derechos de autor en particular se restablece de forma permanente si el titular de los derechos de autor le notifica la violación por algún medio razonable, esta es la primera vez que recibe un aviso de violación de esta Licencia (para cualquier trabajo) de ese titular de los derechos de autor, y usted subsana la violación antes de los 30 días posteriores a la recepción de la noticia.

La rescisión de sus derechos en virtud de esta sección no rescinde las licencias de las partes que hayan recibido copias o derechos de usted en virtud de esta Licencia. Si sus derechos han sido rescindidos y no restablecidos de forma permanente, la recepción de una copia de parte o la totalidad del mismo material no le otorga ningún derecho a usarlo.

10. Futuras revisiones de esta licencia

La Free Software Foundation puede publicar versiones nuevas y revisadas de la Licencia de documentación libre GNU de vez en cuando. Estas nuevas versiones serán similares en espíritu a la versión actual, pero pueden diferir en detalles para abordar nuevos problemas o inquietudes.

Consulte <http://www.gnu.org/copyleft/>.

Cada versión de la Licencia recibe un número de versión distintivo. Si el Documento especifica que se aplica una versión numerada particular de esta Licencia "o cualquier versión posterior", tiene la opción de seguir los términos y condiciones de esa versión especificada o de cualquier versión posterior que haya sido publicada (no como borrador) por la Fundación de Software Libre. Si el Documento no especifica un número de versión de esta Licencia, puede elegir cualquier versión publicada (no como borrador) por la Free Software Foundation. Si el

El Documento especifica que un apoderado puede decidir qué versiones futuras de esta Licencia se pueden usar, la declaración pública de aceptación de una versión de ese apoderado lo autoriza permanentemente a elegir esa versión para el Documento.

11. Renovación de la licencia

"Sitio de colaboración masiva de varios autores" (o "Sitio MMC") hace referencia a cualquier servidor de la World Wide Web que publique obras protegidas por derechos de autor y también proporcione instalaciones destacadas para que cualquier organismo edite esas obras. Un wiki público que cualquiera puede editar es un ejemplo de dicho servidor. Una "Colaboración masiva de varios autores" (o "MMC") contenida en el sitio significa cualquier conjunto de obras protegidas por derechos de autor publicadas en el sitio de MMC.

"CC-BY-SA" se refiere a la licencia Creative Commons Attribution-Share Alike 3.0 publicada por Creative Commons Corporation, una corporación sin fines de lucro con sede comercial principal en San Francisco, California, así como las futuras versiones con copyleft de esa licencia publicada por esa misma organización.

"Incorporar" significa publicar o volver a publicar un Documento, en su totalidad o en parte, como parte de otro Documento.

Un MMC es "elegible para renovar la licencia" si tiene una licencia bajo esta Licencia y si todo funciona que se publicaron por primera vez bajo esta Licencia en otro lugar que no sea este MMC, y posteriormente se incorporaron en su totalidad o en parte en el MMC, (1) no tenían textos de portada o secciones invariables, y (2) se incorporaron antes del 1 de noviembre de 2008.

El operador de un Sitio MMC puede volver a publicar un MMC contenido en el sitio bajo CC-BY-SA en el mismo sitio en cualquier momento antes del 1 de agosto de 2009, siempre que el MMC sea elegible para renovar la licencia.

Índice

: (dos puntos), sintaxis de línea de comandos, 62

[] (corchetes), sintaxis de línea de comandos, 62 # (signo almohadilla), indicador de comentario, 297 | (símbolo de tubería), sintaxis de línea de comando, 62

< > (corchetes angulares), sintaxis de línea de comandos, 62

0.0.0.0 dirección IP

 definición, 110

 negar paquetes dirigidos a, 109

255.255.255.255 Dirección IP, negando paquetes originados en, 109

A

-una opción, 224 -a

(-handle) opción, 85–86

-Una opción, 224

Aceptar paquete y detener procesamiento, 87

ACEPTAR regla

 definición de una política predeterminada, 106

 definición, 351 declaración de aceptación, 87

política de aceptar todo por defecto,

29–30, 351. Consulte también Políticas predeterminadas.

Indicador ACK, 16, 351

agregar comando

 nftables, sintaxis de cadena, 86–

 87 nftables, sintaxis de tabla, 85–86

agregando

 cadenas a una tabla, 86–87

 reglas, 87 tablas, 85–86

Dirección familias, 84

Información de dirección, que se muestra numéricamente, 85–86

Protocolo de resolución de direcciones (ARP), 17–18

Direcciones. *Ver direcciones Ethernet; Direcciones IP.*

extensión de coincidencia de tipo de dirección, 77

AIDE (Detección avanzada de intrusos)

Ambiente)

cambiar la salida del informe, 303–306

verificaciones de suma de verificación, 310

archivos de configuración, 297–300 configurar, 297–301 definir macros, 306–307 verificaciones agrupadas, 309 inicializar la base de datos, 300 instalar, 296–297 monitorear, 301–302 verbosidad del informe, 305–306 ejecutándose automáticamente, 301 comprobaciones estándar, 308–309 tipos de comprobaciones, 307–310 actualizando la base de datos, 302–303 Alertas del programa Snort, 290–291 ALG (puerta de enlace a nivel de aplicación). Véase también Proxies, nivel de aplicación.

definición, 351

descripción, 25–26

Corchetes angulares (< >), sintaxis de línea de comandos, 62

Capa de aplicación, 6

Criptografía aplicada, 310

Operadores aritméticos, 271

ARP (Protocolo de resolución de direcciones), 17–18

familia de direcciones arp, 84

Expresiones de encabezado ARP, 91

paquetes ARP, 18

falsificación de ARP, 264

Demonio ARPWatch, 265, 291–293

Detección de ataques. Véase también Intrusión detección.

ARP spoofing, 264 captura

de tráfico de red. *Ver resoplar programa.*

entorno de concentrador *frente* a conmutador, 263 puertos espejo, 263–264 supervisión del tráfico ARP. *Ver demonio ARPWatch.*

descripción general, 263–

264 captura y análisis de paquetes. *Consulte Volcado TCP.* puertos de tramo, 263–264

Puerto AUTH, 351

autenticación, definición, 351

Encabezado de autenticación, en VPN, 230–231

Autorización, definición, 351

B

NAT básico, 58, 199

Bastilla Linux, 258

Cortafuegos de bastión

definición, 3, 354

limitaciones de, 179–180

reenvío de paquetes, 179–180

NAT bidireccional, 58, 199

BIND (dominio de nombre de Internet de Berkeley), 351

Indicadores de bits, TCP, 15–16

Bloqueo

de transmisiones dirigidas, 110

transmisiones limitadas, 110

servicios TCP locales, 113–115 sitios

problemáticos, 33–34

Libros y publicaciones

Criptografía aplicada, 310

"Denegación de servicio", 40

"Bombardeo de correos electrónicos y spam", 46

Tipos de cadena, nftables, 87

Cadenas. Consulte también Cadenas definidas por el usuario.

agregar a una tabla, 86–87
 incorporado, 61 borrar, 85–86
 crear, 86 definición, 352 eliminar,
 86 mostrar para tablas, 85–86
 ADELANTE, 60–61 ENTRADA,
 60–61 tabla natural, 61 en
 nftables . Consulte nftables, sintaxis
 de cadena.

operaciones en, 62–63
 SALIDA, 59–61
 POSTERIOR AL RUTEADO, 59–61
 ENRUTAMIENTO PREVIO, 59–61
 cambio de nombre, 86 definido por el
 usuario, 54–55

Sumas de comprobación

cheques AIDE, 310
 definición, 352
 TCP (control de transmisión
 Protocolo), 15

programa chkrootkit

descarga, 251 falsos
 negativos, 252 falsos
 positivos, 252 informes de
 infección, 253 limitaciones,
 253–254 programación de
 ejecución, 255 ejecución,
 251–253 uso seguro, 254–255

Estrangular cortafuegos, 181–182, 354

programa chroot/llamada al sistema, 352

CIDR (interdominio sin clase)
 Enrutamiento), 352

Pasarela del circuito. Consulte Proxies, nivel de circuito.

Clase, dirección de red, 352

Cadenas y reglas de compensación, 85

Modelo cliente/servidor, 352–353

estado CERRADO, 17

CLOSE_WAIT estado, 17

Dos puntos (:), sintaxis de línea de comandos, 62

Entrada de línea de comando, habilitación, 85

Opciones de línea de comandos, nftables, 85

Comandos, tabla de filtros, 62–67

Comandos y subcomandos, nftables, 83

Máquinas comprometidas. Consulte Detección de
 ataques; Detección de intrusos.

Equipo de Respuesta a Emergencias Informáticas
 (CERT), 352

Archivos de configuración, AIDE, 297–300

Configuración de AIDE, 297–301 servicios

de correo electrónico. Consulte Servicios

de correo electrónico, inicializando. cortafuegos

Consulte Inicialización de cortafuegos;

Instalación de cortafuegos. LAN internas, 191–
 192 LAN múltiples, 192–195 Programa

Snort, 288–289

Estado de conexión, inicialización de
 cortafuegos, 107

Expresiones de seguimiento de conexión, 88–89

Sin conexión frente a orientado a la conexión
 protocolos, 7

archivo de política de estado de conexión, 170, 173

cadena de seguimiento de conexión, 151, 166

Expresiones de Contrack, 88–89

Conservación de direcciones, 10

Continuar procesando paquetes, 87

continuar declaración, 67

Contramidas. Consulte Prevención de intrusiones.

crear comando, 86

bandera CWR, 16

D

demonios. Véase también demonios específicos.

- definición, 353 escucha en puertos de servicio, 19 modelo de red DARPA, 6

Base de datos, AYUDA

- inicializando, 300
- actualizando, 302–303

Capa de enlace de datos, 6, 353

Debian, inicializando cortafuegos, 140 --

opción de depuración, 85

depuración

- habilitación, 85
- scripts de cortafuegos, 139–140

Depuración, reglas de cortafuegos

- consejos de desarrollo de cortafuegos, 211–212
- comando fuser, 226–227 iptables -L INPUT, 214–215 iptables -n -L INPUT, 215–216 lista de tablas de iptables, 213–214 iptables -v -L INPUT, 216 lista de reglas de firewall, 213–217 prioridades de mensajes de registro, 218 mensajes de registro, interpretación, 220–223 auditoría de seguridad de red, 227 ejemplo de listado de nftables, 216–217

Herramienta Nmap,

- 227 puertos abiertos, comprobación de, 223–227
- convenciones de generación de informes de salida, 226 procesos vinculados a puertos, comprobación de, 226–227
- configuración de syslog, 217–220 registros del sistema, 217–223 verbosidad, 216
- políticas predeterminadas. Véase también

aceptar todo

- política por defecto; Política denegar todo de forma predeterminada. aceptar-todo-por-predeterminado, 29–30

- definición, 353
- denegar todo por defecto, 29–30 filtrado de paquetes, 31–32

borrar comando, nftables

- sintaxis de cadena, 86
- sintaxis de reglas, 87
- sintaxis de tablas, 85

Borrando

- cadena, 86
- reglas, 87
- tablas, 85

Zona Desmilitarizada (DMZ), 180, 353

demultiplexado, 6

"Denegación de servicio", 40

Ataques de denegación de servicio (DoS). Ver DoS (Ataques de denegación de servicio).

Política denegar todo de forma predeterminada. Ver también Políticas predeterminadas.

- depuración, 211–212
- definición, 353 descripción,
- 30 deficiencias, 114

Denegar paquetes frente a rechazar, 31

Direcciones de destino, NAT, 202

NAT de destino (DNAT), 205–206, 209–210

cadena de verificación de dirección de destino, 151, 168

Detección de intrusiones. Consulte Detección de intrusos.

DHCP (configuración dinámica de host) Protocolo), 353

Mensajes DHCPACK, 135

Mensajes DHCPDECLINE, 135

Mensajes DHCPDISCOVER, 135

Mensajes DHCPINFORM, 135

Mensajes DHCPNAK, 135

Mensajes DHCPOFFER, 135

Mensajes DHCPRELEASE, 135

Mensajes DHCPREQUEST, 135

Emisiones dirigidas, 12, 110

Calificador de dirección, TCPDump, 270–271

Directorios, incluso en una ruta de búsqueda, 85

Rechazar paquetes entrantes, 108–109

DMZ (zona desmilitarizada), 180, 353

DNAT (NAT de destino), 205–206, 209–210

Extensiones de objetivo DNAT, 56, 58, 80–81

DNS (Servicio de nombres de dominio)

- definición, 18, 353
- Búsquedas de DNS, como cliente, 120–121
- Búsquedas de DNS, como servidor de reenvío, 121–122
- habilitación, 117–122
- transferencias de zona, 118

Uso del puerto DNS BIND, 121

Búsquedas de DNS, 120–122

protocolo DNS, 119

trafico DNS, identificacion, 157 archivo

dns-policy, 170

Documentación. Ver Libros y publicaciones.

dominios, 18

Ataques DoS (denegación de servicio). Ver también Filtrado de paquetes entrantes.

- desbordamientos de búfer, 45
- contramedidas, 41 definición, 353
- vulnerabilidades de correo electrónico, 46
- activación del módulo de cookies
- SYN, 41 desbordamiento del sistema de archivos, 45
- bombas de fragmentación, 44–45 ping, inhabilitación, 42
- inundación de ping, 41–42 Ping of Death, 42– 43 bombas de redireccionamiento, 45 ataque Smurf, 41 filtrado de direcciones de origen, 40–41 inundación TCP SYN, 40–41

Ataque de lágrima, 44

inundaciones UDP, 43

Notación decimal con puntos, 8

Notación cuádruple punteada, 8

Descartar paquete y detener el procesamiento, 88

regla DROP

- definición de una política predeterminada, 106
- definición, 353

soltar declaración, 88

Descartar paquetes, 108, 112, 138

Computadora de doble base, 353. Véase también Computadora de base múltiple.

protocolo de configuración huésped dinámico (DHCP), 353

Dirección asignada dinámicamente, 353

Y

bandera de la CEPE, 16

“Bombardeo de correos electrónicos y spam”, 46

Servicios de correo electrónico, ataques DoS (denegación de servicio), 46

Servicios de correo electrónico, inicializando

- alojar un servidor de correo, 127–128
- protocolos de correo, 124 descripción general, 123 recibir correo, 125–127
- retransmitir correo saliente, 124 enviar correo a servidores de correo externos, 125 enviar a través de SMTP, 123

Encapsulación, 6

Transparencia de extremo a extremo, 4

Error de mensajes

- mensaje de error ICMP tipo 3, 35, 44 iptables, selección, 57 flujo de salida, 304

STDERR, flujo de error estándar, 304

ESP (carga útil de seguridad encapsulada), 231–232

paquetes ESTABLECIDOS, 73

estado ESTABLECIDO, 17

expresión estatal establecida, 89

direcciones Ethernet, 18

Tarjetas Ethernet, identificación, 18

marco Ethernet, 353

Explotaciones y ataques. Consulte Detección de ataques;
Detección de intrusos.

Expresiones, nftables

Expresiones de encabezado ARP, 91

Expresiones de carga útil de IPv4, 90

Expresiones de encabezado IPv6, 90

Expresiones de encabezado TCP, 90

Expresiones de encabezado UDP, 91

Expresiones, TCPODump, 269–271

Extensiones. Ver Declaraciones.

Archivos de reglas externas, 170

Cadena de entrada EXT-icmp, 152, 164

Cadena de salida EXT-icmp, 152, 164

Cadenas de entrada EXT, 151, 157

Cadena de inicio de sesión EXT, 152, 168–170

Cadena de cierre de sesión EXT, 152, 168–170

Cadenas de salida EXT, 151, 157

F

opción -f (–archivo), 85

Instalaciones, 217–218

Otoño, Kevin R., 7

Falsos negativos/positivos, 252

Sintaxis de archivo, nftables, 92

Protocolo de transferencia de archivos (FTP). Consulte FTP (Protocolo
de transferencia de archivos).

Archivos, incluyendo, 85

Comprobaciones básicas de

integridad de la integridad del sistema de

archivos , 295 sumas de comprobación,

295–296 definición, 295–296

indicaciones de intrusión, 240

software para verificación, 255–256. Véase
también AIDE (Entorno de detección de intrusos
avanzado).

Desbordamiento del sistema de archivos, DoS (denegación de servicio)
ataques, 46

Filtro, regla de cortafuegos, 354

cadena de filtros, 87

tabla de filtros

descripción, 54–55

extensiones de características,

56 vaciado de la cadena, 103

extensiones de coincidencia,

56 sintaxis. Consulte la sintaxis de iptables, tabla
de filtros. extensiones objetivo, 56

Filtrado de paquetes entrantes. Véase también DoS

(Ataques de denegación de servicio; Cortafuegos de filtrado
de paquetes.

bloqueo de sitios problemáticos, 33–34

direcciones ilegales, 32–33 transmisión

limitada, 34 limitación de paquetes

entrantes a hosts seleccionados, 34

por dirección de destino local, 34–35 por puerto

de destino local, 35 sondas, 36–39 por dirección

de origen remota, 31–34 por puerto de origen

remoto, 35 escaneos, 36–39 suplantación de

dirección de origen, 32–33 paquetes enrutados

de origen, 46 por estado de conexión TCP, 35–

36

Filtrado de mensajes de registro de iptables, 57

Filtrado de paquetes salientes. Véase también Cortafuegos de
filtrado de paquetes.

por dirección de origen local, 47 por

puerto de origen local, 48 por estado

de conexión TCP saliente,
48–49

Filtrado de paquetes salientes. Véase también Cortafuegos de filtrado de paquetes. *(continuado)*

- descripción general, 46–
- 47 por dirección de destino remoto, 47–48 por puerto de destino remoto, 48

Bandera FIN,

programa de 16 dedos, 354

FIN_WAIT_2 estado, 17

Programa de administración de cortafuegos. Consulte Iptables.

Inicialización de firewall, ejemplo de optimización, 153–154, 170–172

Registros de cortafuegos. Consulte Registro.

Reglas de firewall, listado, 213–217

cortafuegos

- básico. *Consulte* Cortafuegos Bastion.

- bastión. *Consulte* Cortafuegos Bastion.

- choke, 181–182, 354

- combinación con VPN, 233–234 definición, 3, 25, 354 consejos de desarrollo, 211–212 dual-homed,

- 354 inicialización. *Consulte* Inicialización de cortafuegos. instalando *Consulte* Instalación de cortafuegos.

- Enrutadores habilitados para NAT como, 4 sin estado. *Consulte* Cortafuegos sin estado.

- filtrado de paquetes. *Consulte* Cortafuegos de filtrado de paquetes.

- propósito de, 3–4

- dispositivos de enrutador

- como, 4 host apantallado,

- 354 subred apantallada, 354

- independiente. *Consulte* Cortafuegos Bastion.

- con estado, 25

- sin estado, 25

- transparencia, 4

Cortafuegos, ejemplos (listados de códigos)

- iptables, 315–328 nftables, 328–332

Paquete de

- inundación , 354

- ping, 41–42

- TCP SYN, 40–41

- UDP, 43

comando de descarga, nftables

- sintaxis de cadena, 86

- sintaxis de tabla, 85

Vaciar la definición de

- cadena , 103–104 efecto en

- la política predeterminada, 211 nftables,

- sintaxis de cadena, 86 nftables, sintaxis

- de tabla, 85

Delantero, definición, 354

Cadenas FORWARD, mesa mangle, 61

ganchos delanteros, 85

Reenvío. Consulte también Reenvío de paquetes. host, 209–210 NAT, 201 puerto, 59

Fragmento, definición, 355

Bombas de fragmentación, 44–45

Marcos, OSI (Sistema Abierto Interconexión), 6

FTP (Protocolo de transferencia de archivos)

- anónimo, 355 autenticado,

- 355 definición, 355

- inicializando cortafuegos,

- 130–133 en puertos sin privilegios, 114

NAT completo, 201

comando del fusor, 226–227

GD&AD

Puerta de enlace, definición, 355

Configuraciones de cortafuegos de puerta de enlace, reenvío de paquetes, 181–182

puerta de enlace primitiva, 271

Encapsulación de enrutamiento genérico, 230

instrucción goto, 88 primitivo mayor, 271

Cheques agrupados, AIDE, 309

H

opción -h (--ayuda), 85

trucos Consulte Detección de ataques; Intrusión detección.

opción --handle (-a), 85–86

expresiones de encabezado

ARP, 91

IPv6, 90

TCP, 90

UDP, 91

Indicadores de encabezado, TCP, 283

Encabezados

autenticación, VPN, 230–231 Direccionamiento

IPv4, 8 IPv6, 8 TCP, 15 Ayuda, visualización, 85

Reenvío de host, ejemplo, 209–210 Nombres de

host, direccionamiento IP, 18 hosts.allow, archivo

de configuración de contenedores TCP, 355

hosts.deny, archivo de configuración de contenedores TCP, 355

documentos COMO, 355

programa hping3, 260

HTTP (Protocolo de transferencia de hipertexto), 355

Conversaciones HTTP, captura, 273–277

Entorno concentrador frente a conmutado, 263

concentradores

definición, 355

detección de intrusos, 250

I

-i (--interactivo) opción, 85

-I (--includepath) opción, 85

IANA (Autoridad de Números Asignados en Internet), 19–20, 355

ICMP (Protocolo de mensajes de control de Internet), 12–14, 355

opciones de coincidencia de la tabla de filtros icmp, 66–67

Tráfico ICMP, ejemplo de optimización, 163–165

Mensaje de error ICMP tipo 3, 35, 44 archivo de

política icmp, 170

servidor identificado, 355

IGMP (Administración de grupos de Internet Protocolo), 111

IKE (intercambio de claves de Internet), 232

Direcciones ilegales, 32–33

IMAP (Acceso a mensajes de Internet Protocolo), 355

Direcciones imposibles, registro, 102

Reporte de incidentes, intrusiones detectadas.

Consulte Respuesta a intrusiones, informes de incidentes.

Incluyendo archivos, 85

paquetes de multidifusión entrantes, 111 paquetes

entrantes. Consulte Filtrado de paquetes entrantes.

familia de direcciones inet, 84

servidor inetd, definición, 355

Informes de infecciones, 253

Inicialización de cortafuegos. Consulte también Instalación de cortafuegos.

estado de conexión, 107

en Debian, 140

DNS (Servicio de nombres de dominio), habilitación, 117–122

descarga de la cadena, 103–104

FTP, 130–133

servicio TCP genérico, 133–134

direcciones imposibles, registro, 102

Inicializando cortafuegos. Consulte también

Instalación de cortafuegos. (*continuación*)

bloqueo involuntario, 100 servicios de

Internet, habilitación, 117–122 supervisión

del kernel, habilitación, 101–102 registro, 108,

109 comando `log_martians`, 102 interfaz de

bucle invertido, habilitación, 105 reglas

preexistentes, eliminación de cadenas. *Consulte*

Lavado de la cadena. en Red Hat, 140 mensajes

de redireccionamiento, desactivación, 102 de

forma remota, 100 verificación de reglas,

omisión, 107 invocaciones de reglas, 99–100

escalabilidad, 107 validación de dirección de

origen, desactivación, 102 paquetes enrutados

en origen, desactivación, 101 suplantación de

direcciones de origen, 108– 112 SSH (Secure

Shell), 128–130 detención del cortafuegos, 104–

105 en SUSE, 140 cookies SYN, habilitación,

102 servicios TCP, habilitación, 122–128 tiempos

de espera, 107 servicios UDP, habilitación, 134–

138

Inicializando cortafuegos, direcciones incorrectas

dirección 0.0.0.0, 109–110 dirección

255.255.255.255, 109 difusiones dirigidas, 110

no permitir paquetes entrantes, 108–109 descartar

paquetes, 108, 112, 138 paquetes de multidifusión entrantes, 111

difusiones limitadas, 110 registrar paquetes descartados , 138

paquetes multicast con protocolo no UDP, 111

registro y enrutamiento de multidifusión,
111–112

paquetes de red de multidifusión falsificados,
110–111

Inicialización de cortafuegos, políticas predeterminadas

definir, 106

restablecer, 104–105

reglas, 106

Inicialización de cortafuegos, servicios de correo electrónico

alojar un servidor de correo, 127–128

protocolos de correo, 124 descripción

general, 123 recibir correo, 125–127

retransmitir correo saliente, 124 enviar

correo a servidores de correo externos,

125 enviar a través de SMTP, 123

Inicialización de cortafuegos, script de shell

ejecución, 99–100

constantes simbólicas para nombres y
direcciones, 100

Cadenas INPUT, mangle table, 61

ganchos de entrada, 84

insertar comando, 87

Instalando

AYUDA, 296–297

Programa Snort, 287–288

TCPDump, 266–267

cadenas definidas por el usuario, ejemplo
de optimización, 155–156

Instalación de cortafuegos. Ver también Inicializar
cortafuegos

con direcciones IP dinámicas, 141

guión de cortafuegos, 139–140

argumento de inicio, 139 iniciar y

detener el cortafuegos, 140–141

dejar de discutir, 139

Autoridad de Números Asignados en Internet (IANA), 19–20, 355

Protocolo de mensajes de control de Internet (ICMP), 12–14, 355

Protocolo de gestión de grupos de Internet (IGMP), 111

Intercambio de claves de Internet (IKE), 232

Protocolo de acceso a mensajes de Internet (IMAP), 355

Protocolo de Internet (IP), 7, 12–14

Seguridad de protocolo de Internet (IPSec), 230. Consulte también direcciones IP.

Servicios de Internet, habilitación, 117–122

Detección de intrusos. Véase también Ataque
detección; Respuesta de intrusión.
papel humano en, 237–238
resumen, 237–238

Detección de intrusión, síntomas

indicaciones del sistema de archivos,
240 descripción general, 238–239

indicaciones de la herramienta de auditoría de
seguridad, 241 indicaciones de configuración del
sistema, 239–240

indicaciones de registro del sistema,
239 indicaciones de rendimiento del sistema, 241
indicaciones de cuenta de usuario, 240–241

Kit de herramientas de detección de intrusos

establecimiento de líneas base de tráfico, 250
software de integridad del sistema de archivos, 255–
256 concentradores, 250 limitaciones de herramientas,
253–254 monitoreo de registros, 256–257 monitoreo
de fallas de inicio de sesión SSH, 256–257

rastreadores de red, 249

herramientas de red, 249–250

programa ntop, 250 verificadores

de rootkit, 251

rootkits, 251

Resoplar, 249–250

Programa de muestras, 256–257

interruptores, 250

volsado TCP, 249

Kit de herramientas de detección de intrusos, programa
Chkrootkit

descarga, 251 falsos

negativos, 252 falsos

positivos, 252 informes de

infección, 253 limitaciones,

253–254 programación de

ejecución, 255 ejecución, 251–

253 uso seguro, 254–255

Prevención de intrusiones

Bastilla Linux, 258

Ataques DoS (denegación de servicio), 41

programa hping3, 260

programa nikto, 260

Programa Nmap (Mapeador de redes), 259–260

puertos abiertos, pruebas para, 260

descripción general, 257 pruebas de

penetración, 259–260 asegurar a menudo,

257–258 probar a menudo, 259–260

actualizar a menudo, 258–259 servidores

web, pruebas, 260

Respuesta de intrusión

lista de verificación, 242–

243 documentar sus acciones, 242 llevar un

registro, 242 descripción general, 241–243

tomar una instantánea de los registros del

sistema, 242

Respuesta a intrusiones, informes de incidentes

designación de destinatario de informe, 246

- Respuesta a intrusiones, notificación de incidentes (*continuación*) tipos de incidentes notificables, 244–245 descripción general, 243 motivos, 243–244 información recomendada, 246–247
- paquetes NO VÁLIDOS, 73
- expresión de estado inválida, 89
- archivo de política inválida, 170, 173
- IP (Protocolo de Internet), 7, 12–14
- familia de direcciones IP, 84
- Direcciones IP. Véase también IPsec.
 - 0.0.0.0, 109–110
 - 255.255.255.255, 109
 - direcciones incorrectas. *Consulte* Inicialización de cortafuegos, direcciones incorrectas.
 - difusión, 8, 9, 11 difusión, 11–12 conservación de direcciones, 10 difusiones dirigidas, 12
 - DNS (Servicio de nombres de dominio), 18 dominios, 18 notación decimal con puntos, 8 notación cuádruple con puntos, 8 direcciones Ethernet, 18 nombres de host, 18 ilegales , 32–33 para IPv4, 8–9 para IPv6, 8 transmisiones limitadas, 8, 12 vinculación de dispositivos físicos a direcciones IP. *Consulte* ARP (Protocolo de resolución de direcciones). loopback, 8 enmascaramiento, 99 multidifusión, 9–10 multidifusión, 11–12 red, 8 dominios de red, 18 transmisiones dirigidas por red, 8 descripción general, 8–11 especial, 7 división en subredes, 8–11 suscriptores, 11–12 nombres simbólicos para, 18, 98 sintaxis, 9 unidifusión, 9
- datagramas IP
 - definición, 355
 - tamaño máximo, 11 MTU (Unidad máxima de transmisión), 11 división. *Ver* fragmentación de IP.
- fragmentación de IP, 11
- familia de direcciones ip6, 84
- módulo de cadenas de ip
 - definición, 355 en distribuciones anteriores, 96 vs. iptables, 52
- Mecanismo IPFW (cortafuegos IP). Véase también módulo Ipfwadm.
 - definición, 355 *frente* a Netfilter, 51–54. Véase *también* cortafuegos Netfilter.
 - en distribuciones más antiguas, 96 recorrido de paquetes, 52–54
- módulo ipfwadm, 96, 356
- extensión de coincidencia iprange, 77–78
- IPsec (Seguridad del Protocolo de Internet), 230. *Ver* también direcciones IP.
- \$IPT, 97
- iptables
 - definición, 356
 - mensajes de error, selección, 57 filtrar mensajes de registro, 57 filtrar tabla, 54–56

ejemplo de cortafuegos (lista de códigos), 315–328 vs.	extensiones de destino de tabla nacional, 79–82
ipchains, 52 -L INPUT, 214–215 función de carga,	Cadenas de SALIDA, 61
posibles errores, 140–141 mangle table, 54–55, 56, 60–	POSTRRUTEADO cadenas, 61
61 extensiones de coincidencia, 56 -n - L INPUT, 215–	Cadenas PREROUTING, 61 tablas
216 NATP (traducción de direcciones y puertos de red),	primarias, 61
58	RECHAZAR extensión de destino, 68
	sintaxis de iptables, tabla de filtros
frente a nftables, 83	extensión de coincidencia addrtpe, 77
coincidencia de paquetes, 57	cadenas integradas, 61 comandos, 62–67
destino de COLA, 57 destino	extensión de coincidencia de iprange, 77–
de RECHAZO, 57 destino de	78 extensión de coincidencia de longitud, 78–
RETORNO, 58 característica	79 extensión de coincidencia de límite, 70–71
de guardado, errores potenciales, 140–141 secuencia	
de comandos, ejemplo de optimización, 151–152	Extensión de destino LOG, 67
secuencia de comandos de shell, línea shebang (primera	extensión de coincidencia mac, 75
línea), 97 sintaxis , 54–55 campo TOS, 57 cadenas	extensión de coincidencia de marca,
definidas por el usuario, 54–55	76 extensiones de coincidencia, 68–
	79 operaciones de coincidencia, 62,
comando iptables	64 extensión de coincidencia multipuerto, 69–70
definición de reglas, 97	operaciones en cadenas completas, 62–63
definición, 54 habilitación	operaciones en reglas, 62 extensión de coincidencia
de comandos de tabla de filtro, 62 ubicación,	de propietario , 75–76
configuración, 97 sintaxis, 55	REJECT extensión de destino, 68 opciones
sintaxis de iptables	de regla, 64–65 extensión de coincidencia
cadena de comandos en individuo	de estado, 71–75 extensiones de destino,
reglas, 64	67–68 opciones de coincidencia de tcp, 65
Cadenas FORWARD, 61 opciones	extensión de coincidencia de tos, 76–77
de coincidencia de la tabla de filtros icmp, 66–67	opciones de coincidencia de udp, 66
Cadenas de ENTRADA,	
61 comando de cadena de lista, opciones, 63	extensión de objetivo ULOG, 57, 68 extensión
extensión de destino LOG, 67 mesa	de coincidencia sucia, 77
mangle, 61, 81–82 extensión de	sintaxis de iptables, tabla nat
destino de marca, 81–82 tabla nat, 61	Extensiones de objetivo DNAT, 80–81
	Extensiones de diana MASQUERADE, 80
	REDIRECT extensiones de destino, 81
	extensiones de destino SNAT, 79–80
	extensiones de destino, 79–81

listado de tablas de iptables, 213–214

iptables -v -L ENTRADA, 216

Escasez de

- direcciones de

- direccionamiento IPv4 , 4

- clases, 8–9 notación decimal con

- puntos, 8 notación cuádruple con

- puntos, 8 encabezado, 8

- Direccionamiento IP, 8–9

Expresiones de carga útil de IPv4, 90

IPv6

- encabezado,

- 8 expresiones de encabezado, 90

- direccionamiento IP, 8

j

-j objetivo LOG, 108, 138

declaración de salto, 88

PARA

Supervisión del kernel, habilitación, 101–102 demonio

klogd, 356

I

-L mando, 223

L2TP (Protocolo de Túnel de Capa 2), 229–230

Ataque TERRESTRE, 284

LAN (redes de área local)

- definición, 356

- Ejemplo de NAT, 209–210

- seguridad, reenvío de paquetes, 182–183

LAN, reenvío de paquetes en una mayor o menor

de confianza

- configurar una LAN interna, 191–192 configurar

- múltiples LAN, 192–195 crear múltiples redes, 188–190

división del espacio de direcciones, 188–190

descripción general, 188 acceso interno

selectivo, 190–195 división en subredes, 188–

190

LAN, reenvío de paquetes en un servidor de confianza

reenvío de tráfico local, 186–188

Acceso LAN al cortafuegos de la puerta de enlace,
184–186

múltiples LAN, 186–188

extensión de coincidencia de longitud, 78–79

menos primitivo, 271

Libreswan program, 233

límite de extensión de coincidencia, 70–71

Se alcanzó el límite de coincidencias recibidas
paquetes, 88

declaración de límite, 88

Emisiones limitadas

- bloqueo, 110

- transmisión, 12

- definición, 8 filtrado de

- paquetes entrantes, 34

Limitación de paquetes entrantes a seleccionados
anfitriones, 34

linux

- flujos de salida, 304

- VPN (redes privadas virtuales), 232–233

Programa de administración de cortafuegos de Linux. Consulte
iptables.

Kernel de Linux, personalizado frente a stock, 97–98

comando de cadena de lista, opciones, 63

comando de lista, nftables

- sintaxis de cadena, 86

- sintaxis de tabla, 85–86

Dirección de destino local, filtrado por,
34–35

Puerto de destino local, filtrado de paquetes entrantes, 35

Dirección de origen local, filtrado de paquetes salientes,
47

Puerto de origen local, filtrado saliente
paquetes, 48

cadena local_dhcp_client_query, 166–167

cadena local_dns_client_request, 159–161

cadena local_dns_server_query, 151, 158

anfitrión local, definición, 356

archivo de política localhost, 170, 172

cadena local_tcp_client_request, 152

cadena local_tcp_server_response, 152,
161–162

cadena local_udp_client_request, 152, 163

demonio bloqueado, 116

Bloqueo, involuntario, 100, 139–140

Filtrado de mensajes

de registro , 57

interpretación, 220–223

prioridades, 218

Supervisión de registros para la detección de intrusos,
256–257

Paquetes de registro,

88 declaración de registro, 88

Extensión de destino LOG, 67

Registro (diario del administrador), 242

Registro (registros del sistema)

como herramientas de depuración, 217–

223 paquetes descartados, 138, 168–170, 175–
176

inicialización de cortafuegos, 108, 109

indicaciones de intrusión, 239 escaneos

de puertos, 38 creación de instantáneas

como respuesta a una intrusión, 242

comando log_martians, 102 archivo de

política de registro, 170 cadena de

estado log-tcp, 152, 165–166

Direcciones de bucle invertido, 8

Interfaz de bucle invertido, 105, 356

METRO

direcciones MAC

definición, 17

identificación de tarjetas Ethernet, 18

cortafuegos de filtrado de paquetes, 27

extensión de coincidencia mac, 75

Macros, en AIDE, 306–307

Páginas man, definición, 356

mangle table

cadenas integradas,

61 sintaxis de comandos, 81–

82 descripción, 54–55

Cadenas FORWARD, 60

Cadenas de ENTRADA, 60

extensión MARK, 56, 81–82 marcado,

60

Cadenas de SALIDA, 60

Cadenas POSTROUTING, 60

Cadenas PREROUTING, 60

extensiones de destino, 56

extensión TOS, 56

campo TOS, 60

extensión MARCA, 56

extensión de coincidencia de marcas, 76

marcar la extensión del objetivo, 81–82

Enmascaramiento de direcciones IP, 99

MASCARADA, 203–204

Extensiones de objetivo MASQUERADE, 57, 59, 80

Enmascaramiento. Véase también NAPT (Traducción de
direcciones y puertos de red); NAT (traducción de direcciones
de red).

definición, 356

descripción, 201 en

versiones anteriores de Linux, 52

en iptables, 59

Tráfico LAN a Internet, ejemplo,
206–208

Extensiones de coincidencia, tabla de filtros, 68–79

Operaciones de coincidencia, tabla de filtros, 62, 64

algoritmo MD5, 356

Metaexpresiones, 89

Puertos de espejo, 263–264

Monitoreo

AIDE, tráfico ARP

301–302. *Ver* demonio ARPWatch.

para la detección automática de intrusos. *Ver*

programa Snort. supervisión del núcleo,

habilitación, 101–102 registros, 256–257 redes

con demonio ARPWatch,

291–293

sesiones, 72

Alertas de resoplido, 290–291

Errores de inicio de sesión SSH, 256–

257 registros del sistema, 256–257

MSS (Tamaño máximo de segmento), 17

MTU (Unidad Máxima de Transmisión), 11, 356

Direcciones de multidifusión, 9–10

"Multidifusión sobre TCP/IP CÓMO", 111

Paquetes de multidifusión, 111, 356

Registro y enrutamiento de multidifusión, 111–112

Multidifusión, 11–12

Computadora con múltiples terminales, 356. Véase también

Computadora con dos terminales.

extensión de coincidencia multipuerto, 69–70, 114

none

opción -n, 224

-n (–numérico) opción, 85–86

Servidor de nombres, principal, 356

Servidor de nombres, secundario, 356

Nombrar scripts de cortafuegos, 139

NAPT (Network Address and Port Translation),

58, 199. Véase también Enmascaramiento.

NAT (traducción de direcciones de red). *Ver* también

Enmascarado.

ventajas de, 199–200 básico,

199 bidireccional, 199

definición, 4, 356–357

direcciones de destino, 202

desventajas de, 200 DNAT

(destino NAT), 205–206 reenvío,

201 completo, 201 introducción, 197–198 en

iptables, 52 MASQUERADE, 203–204

masquerading, 201 NAPT (Network Address
and Port Translation), 199. *Consulte también*

Masquerading. sintaxis de tabla nat, 203 NAT

de destino REDIRECT, 205–206

semántica, 201–206

SNAT (NAT de origen), 203–204 NAT

de origen. *Consulte* SNAT (NAT de

origen). tradicional, 198–199 con modo

de transporte IPSec, 200 dos veces, 199

NAT (traducción de direcciones de red), ejemplos

DNAT (NAT de destino), 209–210 reenvío de

host, 209–210

LAN, 209–210

enmascarando el tráfico LAN al
Internet, 206–208

apoderados, 209–210

SNAT y LAN privadas, NAT estándar 206–

208, tráfico LAN en el

internet, 208

nftables (continuación)

- familia de direcciones arp,
- 84 familia de direcciones de
- punto, 84 sintaxis de cadena,
- 86–87 tipos de cadena, 87 borrar
- cadena, 86 opciones de línea
- de comando, 85 crear comando,
- 86 crear cadenas, 86 definición,
- 357 borrar comando, 86 borrar
- cadena, 86 visualización de reglas
- en una cadena, 86 sintaxis de
- archivo, 92 como programa de
- administración de cortafuegos, 96
- ejemplo de cortafuegos (lista de
- códigos), 328–332
- comando flush, 86
- enlaces directos, 85
- familia de direcciones inet, 84
- enlaces de entrada, 84 familia
- de direcciones ip, 84 familia
- de direcciones ip6, 84 vs.
- iptables, 83 comando list, 86
- listado, ejemplo, 216–217
- sintaxis de comandos nft, 83
- enlaces de salida , 85 enlaces
- posteriores al enrutamiento, 85
- enlaces previos al enrutamiento,
- 84 comando de cambio de nombre,
- 86 cadenas de cambio de nombre,
- 86 subcomando de regla, 84
- secuencia de comandos, ejemplo
- de optimización, 170 subcomando de tabla,
- 84 comandos y subcomandos típicos, 83

nftables, operaciones básicas

- Expresiones de encabezado ARP, 91
- Expresiones de carga útil de IPv4, 90
- Expresiones de encabezado IPv6, 90
- Expresiones de encabezado TCP, 90
- Expresiones de encabezado UDP, 91

nftables, opciones de línea de comandos

- a (--handle) opción, 85–86
- información de dirección, que se muestra
- numéricamente, 85–86
- opción de depuración,
- 85 depuración, habilitación, 85
- directorios, incluso en una ruta de búsqueda,
- 85 habilitación de entrada de línea de
- comando, 85 opción -f (--file), 85 opción -h (--
- help), 85 ayuda, visualización , 85 -i (--
- interactivo) opción, 85

- I (--includepath) opción, 85 incluyendo
- archivos, 85 -n (--numeric) opción, 85–
- 86 número de versión nft, visualización,
- 85 información de puerto, visualización
- numérica, 85–86 identificadores de reglas,
- visualización, 85–86 -v (--versión) opción,
- 85

tablas, expresiones

- expresiones de seguimiento de conexión,
- 88–89
- expresión de estado establecida, 89
- expresión de estado no válido, 89
- metaexpresiones, 89 nueva expresión
- de estado, 89 expresiones de carga útil,
- 88 expresión de estado relacionado, 89
- expresiones de estado, 89 expresión de
- estado sin seguimiento, 89

nftables, sintaxis de reglas

aceptar paquete y detener procesamiento, 87 aceptar
declaración, 87 agregar comando, 87 agregar reglas,
87 continuar procesando paquetes, 87 continuar
declaración, 87 borrar comando, 87 eliminar reglas,
87 descartar paquete y detener procesamiento, 88
descartar declaración, 88 declaración goto, 88
comando de inserción, 87 declaración de salto, 88
límite alcanzado en los paquetes recibidos
coincidentes, 88 declaración de límite, 88 paquetes
de registro, 88 declaración de registro, 88 anteponer
una regla en una cadena, 87 declaración de cola, 88
declaración de rechazo, 88 procesamiento de retorno
a la llamada cadena, 88 declaración de devolución,
88 enviar procesamiento a una cadena específica,
no devolver, 88 enviar procesamiento a una cadena
específica, devolver, 88

sentencias y veredictos, 87–88 detener y
rechazar el paquete, 88 detener y enviar
paquetes al proceso de espacio de usuario, 88 nftables,
comando agregar sintaxis de tabla, 85–86 agregar
una tabla, 85–86 borrar todas las cadenas y reglas para un
mesa, 85

tablas predeterminadas, 85–

86 comando de eliminación,
85 eliminación de una tabla,
85 visualización de cadenas y reglas para una tabla,
85–86
comando de descarga, 85
comando de lista, 85–86
sintaxis de tabla, 85–86

archivo nft-vars, 170

programa nikto, 260

Nmap (mapeador de redes)

definición, 357
descripción, 227
identificación de puertos abiertos y dispositivos
disponibles, 259–260, 281–282 detección de
intrusos, 259–260

NNTP (Protocolo de transferencia de noticias de red), 357

bandera NS, 16

programa ntop, 250

NTP (protocolo de tiempo de red), 137, 357

demonio ntpd, 137

--opción numérica (-n), 85–86

O

Abrir primero la ruta más corta (OSPF), 357

Interconexión de sistemas abiertos (OSI)

capas del modelo. Consulte las capas del modelo

OSI (interconexión de sistemas abiertos).

programa openswan, 233

programa OpenVPN, 233

Mejoramiento

archivos de reglas externas, 170

objetivo de, 176–177 tráfico

ICMP, 163–165 iptables, 176–

177 nftables, 177 script rc.firewall,

173–175

Optimización (*continuación*)

comprobación de dirección de origen, omisión, 162–163

TCP, habilitación del tráfico del servidor local, 161

Tráfico TCP, habilitación desde clientes locales, 159–161

UDP, tráfico de clientes locales, 162

Optimización, ejemplos

cadena de seguimiento de conexión, 166 cadena

de verificación de dirección de destino, 168

Cadena de entrada EXT-icmp, 164

Cadena de salida EXT-icmp, 164

Cadenas de entrada EXT, edificio, 157

Cadena de inicio de sesión EXT, 168–170

Cadena de cierre de sesión EXT, 168–170

Cadenas de salida EXT, construcción, 157

inicialización del cortafuegos, 153–154, 170–172

Tráfico ICMP, 163–165 instalación

de cadenas definidas por el usuario, 155–156 cortafuegos

iptables (listado de códigos), 332–345 script iptables, 151–

152 cadena local_dhcp_client_query, 166–167

cadena local_dns_client_request, 159–161

cadena local_dns_server_query, 158 cadena

local_tcp_server_response, 161–162

local_udp_client_request chain, 163 registro de

paquetes perdidos, 168–170 log-tcp-state chain, 165–

166 cortafuegos nftables (listado de códigos), 345–349

secuencia de comandos

nftables, 170 secuencia de comandos

rc.firewall, 345–349 cadena de
respuesta_servidor_dhcp_remoto, 166–167

cadena_consulta_servidor_dns_remoto, 158

cadena_respuesta_servidor_dns_remoto, 159–161

cadena remote_tcp_client_request, 161–162

remote_udp_server_response chain, 163
comprobación de dirección de origen,

omisión, 162–163

cadena de verificación de dirección de origen, 167–168

TCP, habilitación del tráfico del servidor local, 161

Tráfico TCP, habilitación desde clientes locales,
159–161 cadena tcp-state-flags, 165

UDP, tráfico de clientes locales, 162

Optimización, organización de reglas

creación de archivos de reglas, 172–176

omisión de reglas de suplantación de

identidad, 146 estado de conexión, habilitación,

173 creación de tablas, 172 archivos de reglas

externas, 170 servicios muy utilizados, 147

Reglas ICMP, colocación, 147

Tráfico ICMP, 175 reglas

de paquetes entrantes, ubicación, 146–147

tráfico no válido, eliminación, 173 tráfico de

cliente local, sobre TCP, 174 tráfico de servidor

local, sobre TCP, 175 tráfico de host local, habilitación,

172 secuencia de comandos rc.firewall, 173–175

módulo de estado para ESTABLECIDO y

partidos RELACIONADOS, 146

tráfico, habilitación, 173–174

flujo de tráfico para determinar la regla

colocación, 147–148 protocolos

de transporte, 146–147

Reglas UDP, colocación, 147 por

dónde empezar, 145

Optimización, cadenas definidas por el usuario

- bifurcación, 149
- características de, 150–151
- seguimiento de conexión, 151, 166
- verificación de dirección de destino, 151, 168
- tráfico DNS, identificación, 157
- EXT-icmp-entrada, 152, 164
- EXT-icmp-salida, 152, 164
- Entrada EXT, 151
- Inicio de sesión EXT, 152, 168–170
- Cierre de sesión EXT, 152, 168–170
- Salida EXT, 151
- local_dhcp_client_query, 166–167
- local_dns_client_request, 159–161
- local_dns_server_query, 151, 158
- local_tcp_client_request, 152
- local_tcp_server_response, 152, 161–162
- local_udp_client_request, 152, 163 registro de paquetes descartados, 168–170, 175–176
- log-tcp-state, 152, 165–166
- remote_dhcp_server_response, 152, 166–167
- consulta_servidor_dns_remoto, 158
- respuesta_servidor_dns_remoto, 151, 159–161
- solicitud_cliente_tcp_remoto, 152, 161–162
- remote_tcp_server_response, 152
- remote_udp_server_response, 152, 163 source-address-check, 151, 167–168 tcp-state-flags, 151, 165
- Variable USER_CHAINS, 151

Modelo OSI (interconexión de sistemas abiertos)

- capas
 - Aplicación, 6 sin
 - conexión *frente* a protocolos orientados a la conexión, 7

- Enlace de
 - datos, 6 definición, 357 demultiplexación, 6
 - encapsulación, 6
 - tramas, 6 Red, 6
 - descripción general, 5–6 Físico, 6
 - Presentación, 6
 - Sesión, 6 Capa de transporte, 6 Protocolos de transporte. *Ver* TCP

- (Protocolo de Control de Transmisión); UDP (Protocolo de datagramas de usuario).

OSPF (Abrir primero la ruta más corta), 357

Estado de conexión TCP saliente, filtrado de paquetes salientes, 48–49

Cadenas de SALIDA

- mesa mangle, 61
- mesa nat, 59, 61

ganchos de salida, 85

extensión de coincidencia de propietario, 75–76

PAGS

opción -p, 224

reenvío de paquetes

- cortafuegos de estrangulación, 181–182
- DMZ (zona desmilitarizada), 180 configuraciones de cortafuegos de puerta de enlace, 181–182
- Seguridad LAN, 182–183
- limitaciones de un firewall bastión, 179–180
- redes perimetrales, 180

Reenvío de paquetes, en un servidor más grande o menos confiable LAN

- configurar una LAN interna, 191–192 configurar múltiples LAN, 192–195 crear múltiples redes, 188–190

Reenvío de paquetes, en un servidor más grande o menos confiable
LAN (*continuación*)

división del espacio de direcciones, 188–190

descripción general, 188 acceso interno

selectivo, 190–195 división en subredes, 188–

190

Reenvío de paquetes, en una LAN doméstica de confianza

que reenvía tráfico local, 186–188

Acceso LAN al cortafuegos de la puerta de enlace,
184–186

varias LAN, 186–188 Coincidencia

de paquetes, iptables, 57 Cortafuegos

de filtrado de paquetes. Consulte también Filtrado de
paquetes entrantes; Filtrado de paquetes salientes.

aceptar-todo-por-política predeterminada,
29–30

política predeterminada, 29–

30 política denegar todo por defecto,
29–30

Filtrado de direcciones MAC, 27

descripción general, 26–28 paquetes

de rechazo *frente* a paquetes de denegación, 31

Paquetes

definición, 357

dirección de destino, especificación, 98–99 descartado,

registro, 138 descartado, 108, 112, 138 filtrado.

Consulte Filtrado de paquetes entrantes; Filtrado de

paquetes salientes; Cortafuegos de filtrado de
paquetes. fragmentos, 44 registro, 88 dirección de
origen, especificación, 98–99

variable RUTA, 357

Expresiones de carga útil, 88

Protocolo de comunicación entre pares, 357.

Prueba de penetración, 259–260

penetraciones Consulte Detección de ataques; Detección de
intrusos.

Redes perimetrales, reenvío de paquetes,
180

Dispositivos físicos, vinculación a direcciones IP.

Consulte ARP (Protocolo de resolución de
direcciones).

Capa física, 6, 357

PID (ID de proceso), 358

inundación de ping, 41–42

Ping of Death, 42–43 captura

de pings, 279 definición, 358

desactivación, 42

Símbolo de canalización (|), sintaxis de línea de comandos, 62

Protocolo de tunelización punto a punto (PPTP),
229–230, 233

Valores predeterminados de la política. Consulte Políticas predeterminadas.

POP (Protocolo de oficina de correos), 358

reenvío de puertos, 59

Información del puerto, que se muestra numéricamente,
85–86

Números de puerto

asignación a nombres de servicios, 19–20

numérico *frente* a simbólico, 96–97

Exploraciones de puertos

definición, 36–38, 358 en

registros de firewall, 38 general,

36 para puertos abiertos, 281–

282 en respuesta, 38–39 furtivo,

38 dirigido, 36–38 nivel de

amenaza, 38–39

demonio del mapa de puertos

definición, 358

descripción, 113

Puertos

objetivos de escaneo comunes, 36–38

definición, 358

privilegiado, 358. Véase *también* Unprivileged
puertos

puertos de servicio, 19–23

puertos, abiertos

buscando, 223–227 buscando,

281–282 probando, 260

Protocolo de oficina postal (POP), 358

Cadenas POSTROUTING, 59, 61

ganchos de postenrutamiento, 85

Signo de libra (#), indicador de
comentario, 297

PPTP (Protocolo de tunelización punto a punto),
229–230, 233

demonio pptpd, 233

Anteponer una regla en una cadena, 87

Cadenas PREROUTING, 59, 61

ganchos de enrutamiento previo, 84

Capa de presentación, 6

Primitivas para TCPEDump, 271

Redes privadas frente a públicas, 49–50

Sondas, definición, 36, 358

ID de proceso (PID), 358

Calificador de protocolo, TCPEDump, 271

apoderados

nivel de aplicación, 358. Véase *también* ALG
(puerta de enlace a nivel de aplicación). nivel
de circuito, 358–359 definición, 358 ejemplo,
209–210

Bandera PSH, 16

Redes públicas frente a privadas, 49–50

q

QoS (Calidad de Servicio), 359

Consultas, capturas, 279

declaración de cola, 88

objetivo COLA, 57

R

RARP (Resolución de dirección inversa
Protocolo), 359

script rc.firewall, 173–175 Registro

de tráfico, 284–286 Red Hat,

inicialización de firewalls, 140

Redirigir bombas, 45

NAT de destino REDIRECT, 205–206

redirigir mensajes, deshabilitar, 102

REDIRECT extensiones de destino, 57, 81

RECHAZO regla, 106, 359

declaración de rechazo, 88

REJECT extensión de destino, 57, 68

Rechazo de paquetes frente a denegación, 31

Paquetes RELACIONADOS, 73 expresión de
estado relacionado, 89 Dirección de destino
remota, filtrado de paquetes salientes, 47–48

Puerto de destino remoto, filtrado de paquetes salientes, 48

Llamada a procedimiento remoto (RPC), 359

Dirección de origen remota, filtrado de paquetes entrantes,
31–34

Puerto de origen remoto, filtrado de paquetes
entrantes, cadena 35 remote_dhcp_server_response,
152, 166–167

cadena de consulta_servidor_dns_remoto, 158

cadena de respuesta_servidor_dns_remoto, 151, 159–
161

Inicialización remota de cortafuegos, 100

cadena remote_tcp_client_request, 152,
161–162

cadena_respuesta_servidor_tcp_remoto,
152

cadena de respuesta_servidor_udp_remoto, 152,
163

renombrar comando, 86

Cadenas de cambio de nombre, 86

- Salida del informe, AIDE, 303–306
- Verbosidad del informe, AIDE, 305–306
- Notificación de intrusiones. Consulte Respuesta a intrusiones, informes de incidentes.
- Solicitud de comentarios (RFC), 359
- Resolver, 359
- Recursos. Ver Libros y publicaciones.
- Responder a las intrusiones. Ver Intrusión respuesta.
- Procesamiento de retorno a la cadena de llamada, 88
- declaración de devolución, 88
- RETORNO objetivo, 58
- Protocolo de resolución de dirección inversa (RARP), 359
- RFC (Solicitud de comentarios), 359
- RFC 1112 "Extensiones de host para IP Multidifusión", 111
- RFC 1122 "Requisitos para Internet Anfitriones: capas de comunicación", 102
- RFC 1458 "Requisitos para multidifusión Protocolos", 111
- RFC 1631 "La dirección de red IP Traductor (NAT)", 197
- RFC 1700 "Números asignados", 113
- RFC 1812 "Requisitos para IP Versión 4 Enrutadores", 102
- RFC 2196 "Manual de seguridad del sitio", 238
- RFC 2236 "Gestión de grupos de Internet Protocolo Versión 2", 111
- RFC 2474, "Definición de la diferenciación Campo de Servicios (DS Field) en el IPv4 y Encabezados IPv6", 77
- RFC 2475, "Una arquitectura para servicios diferenciados Servicios", 77
- RFC 2588 "Multidifusión IP y cortafuegos", 111
- RFC 2647 "Terminología comparativa para Rendimiento del cortafuegos", 25
- RFC 2663 "Dirección de red IP Traductor (NAT) Terminología y Consideraciones", 198
- RFC 2827 "Filtrado de ingreso de red: Derrotar los ataques de denegación de servicio. . . , 47
- RFC 2990 "Próximos pasos para IP QoS arquitectura", 77
- RFC 3022 "Dirección de red IP tradicional traductor (NAT tradicional)", 197
- RFC 3168 "La adición de explícita Notificación de congestión (ECN) a IP", 77
- RFC 3260 "Nueva terminología y Aclaraciones para Diffserv", 77
- RFC 3704 "Filtrado de ingreso para multitarjeta Redes", 47
- RIP (Protocolo de información de enrutamiento), 359
- Inspectores de rootkits, 251
- Rootkits, 251
- cadena de rutas, 87
- Dispositivos enrutadores como cortafuegos, 4
- Protocolos de enrutamiento, 19
- RPC (llamada a procedimiento remoto), 359
- RST bandera, 16
- Organización de reglas
- archivos de reglas de construcción, 172–176
 - eludir las reglas de suplantación de identidad, 146
 - estado de conexión, habilitación, 173
 - creación de tablas, 172
 - archivos de reglas externas, 170
 - servicios muy utilizados, 147
 - Reglas ICMP, colocación, 147
 - tráfico ICMP, 175
 - reglas de paquetes entrantes, colocación, 146–147
 - tráfico inválido, caída, 173
 - tráfico de cliente local, sobre TCP, 174
 - tráfico del servidor local, sobre TCP, 175
 - tráfico localhost, habilitación, 172
 - secuencia de comandos rc.firewall, 173–175
 - módulo de estado para ESTABLECIDO y partidos RELACIONADOS, 146
 - tráfico, habilitación, 173–174

flujo de tráfico para determinar la regla
 colocación, 147–148
 protocolos de transporte, 146–147
 Reglas UDP, colocación, 147
 por dónde empezar, 145
 subcomando regla, 84
 Normas. Véase también Filtro, cortafuegos; cortafuegos;
 nftables, sintaxis de reglas.
 agregar, 87
 verificar, omitir, 107 borrar, 85
 orden de definición, 96
 eliminar, 87 filtrar operaciones
 de tabla en, 62 vaciar la
 cadena, 103 identificadores,
 mostrar, 85–86 invocaciones,
 inicializar cortafuegos, 99–100
 enumerar, 85–86, 213–217 opciones, tabla de
 filtros, 64–65 direcciones de paquetes,
 especificación, 98–99 anteponer en una
 cadena, 87 eliminar de cadenas. *Consulte*
 Lavado de la cadena.

Nivel de ejecución, 359

S

Escalabilidad, inicialización de cortafuegos, 107
 Escaneo de puertos abiertos. Consulte Exploraciones de puertos.
 Schneier, Bruce, 310
 Anfitrión filtrado. Consulte Firewalls, host filtrado.
 Subred filtrada. Ver Firewalls, filtrado
 subred.
 Guiones, definición, 359
 Protocolo Secure Shell (SSH). Ver SSH (Secure
 Shell) protocolo.
 Protocolo de capa de sockets seguros (SSL), 360
 Asegurar a menudo, como prevención de intrusiones,
 257–258

Asociaciones de seguridad, VPN, 232
 Herramientas de auditoría de seguridad,
 indicaciones de intrusión, 241
 Segmentos, TCP, 15, 359
 Enviar procesamiento a una cadena específica, no devolver,
 88
 Enviar procesamiento a una cadena especificada,
 retorno, 88
 Programas de servidor. Véase Demonios.
 Nombres de servicio, asignación a números de puerto,
 19–20
 Puertos de servicio, 19–23
 Capa de sesión, 6
 Supervisión de sesión, mantenimiento de información
 de estado, 72
 programa setgid, 359
 programa setuid, 359
 archivo de configuración de tablas, 170, 172
 Shebang línea (primera línea de guión), 97
 Concha, definición, 359
 SMTP (Protocolo simple de transferencia de correo), 360
 Conversaciones SMTP, captura, 277–278
 Ataque de pitufo, 41, 282–283
 Instantánea de los registros del sistema, 242
 SNAT (NAT de origen)
 y LAN privadas, ejemplo, 206–208 semántica,
 203–204 extensiones de destino, tabla nat,
 56, 58, 59,
 79–80
 SNMP (Administración de red simple
 Protocolo), 360
 Configuración del
 programa Snort , 288–289
 descripción, 249–250, 265 instalación,
 287–288
 Programa Snort, obtención automatizada de
 monitoreo de intrusión, 287–288 descripción
 general, 286

Programa Snort, monitoreo automatizado de intrusiones (*continuación*) recepción de alertas, 290–291 con Swatch, 290–291 pruebas, 289–290

Zócalo, definición, 360

Paquete CALCETINES, 360

Comprobación de la dirección de origen, omisión, 162–163

Direcciones de origen

filtrado, 41 suplantación

de identidad. Consulte Direcciones de origen de suplantación de identidad.

validación, inhabilitación, 102

cadena de verificación de dirección de origen, 151, 167–168

Paquetes enrutados de origen

inhabilitación,

101 filtrado, 46

Puertos de tramo, 263–264

direcciones especiales, 7

División de datagramas IP. Ver fragmentación de IP.

Definición de direcciones de origen

de suplantación de identidad,

360 inicialización de cortafuegos, 108–112

paquetes de red de multidifusión, 110–111 descripción general, 32–33

Corchetes ([]), sintaxis de línea de comandos, 62
caché web de squid, bloqueo de servicios

basados en TCP, 114

Conversaciones SSH, captura, 278

Protocolo SSH (Secure Shell)

definición, 360

inicializando firewalls, 128–130

fallas de inicio de sesión, monitoreo, 256–257

Protocolo SSL (Secure Sockets Layer), 360

Cortafuegos independiente. Consulte Cortafuegos bastión.

Cheques estándar, AIDE, 308–309

NAT estándar, tráfico LAN en el internet, 208

iniciar argumento, 139

Iniciar y detener cortafuegos, 140–141

Expresiones de estado, 89

extensión de coincidencia estatal, 71–75

Cortafuegos sin estado, 25

Declaraciones, sintaxis de reglas, 87–88

Declaraciones y veredictos, 87–88

Dirección asignada estadísticamente, 360

STDERR, flujo de error estándar, 304

STDIN, flujo de entrada estándar, 304

STDOUT, flujo de salida estándar, 304

Escaneo de puertos ocultos, 38

"Pasos para recuperar desde un UNIX o NT
Compromiso del sistema", 238

Stevens, Richard W., 7

dejar de discutir, 139

Detener reglas de procesamiento y

rechazar el paquete, 88 enviar

paquetes al proceso de espacio de usuario,
88

Detención e inicio de cortafuegos, 104–105, 140–141

Capa de subred, definición, 360

División en subredes, 10–11

Suscriptores, 11–12

SUSE, inicialización de cortafuegos, 140

Programa de muestras, 256–257, 290–291

Entorno conmutado frente a concentrador, 263

Interruptores, detección de intrusos, 250

Nombres simbólicos de

hosts, 98

Direcciones IP, 18, 98

números de puerto, 96–97

Cookies SYN, habilitación, 41, 102

bandera SYN, 16, 360

paquetes SYN, 16

segmentos SYN, 16

paquetes SYN_ACK, 17

SYN_ACK segmentos, 17

Estado SYN_RCVD, 17

estado SYN_SENT, 16

configuración de syslog, 217–220 archivo

syslog.conf, 360 demonio syslogd, 360

configuración del sistema, indicación de

intrusión, 239–240

Registros del

- sistema como herramientas de depuración, 217–223 paquetes descartados, 138, 168–170, 175–176
- inicialización de cortafuegos, 108, 109
- indicaciones de intrusión, 239 escaneos de puertos, 38 creación de instantáneas como respuesta a una intrusión, 242

Rendimiento del sistema, intrusión indicaciones, 241

T

subcomando de tabla, 84

Sintaxis de tablas, nftables, 85–86

Mesas

- agregar, 85–86
- borrar cadenas y reglas, 85 eliminar, 85 mostrar cadenas y reglas, 85–86

Target extensions filter

- table, 67–68 mangle
- table, 56 nat table, 56, 56–58, 79–81

Escaneos de puertos dirigidos, 36–38

TCP (Protocolo de control de transmisión)

- una conexión típica, banderas de 20–23 bits, 15–16. *Ver también banderas específicas.*

sumas de control, 15

estado CERRADO, 17

CLOSE_WAIT estado, 17

conexiones, 16–17 definición, 14–15, 360 habilitación, 122–128

estado ESTABLECIDO, 17

FIN_WAIT_2 estado, 17

genérico, inicializando cortafuegos, 133–134 encabezado, 15

MSS (Tamaño máximo de segmento), 17

tablas de protocolo, 116–117 segmentos, 15

paquetes SYN, 16

segmentos SYN, 16

paquetes SYN_ACK, 17

SYN_ACK segmentos, 17

Estado SYN_RCVD, 17

estado SYN_SENT, 16

protocolo de enlace de tres vías, 16

Estado TIME_WAIT, 17

tráfico, habilitación desde clientes locales, 159–161

TCP, habilitación del tráfico del servidor local, 161

Estado de conexión TCP, filtrado de paquetes entrantes, 35–36

Expresiones de encabezado TCP, 90

Indicadores de encabezado TCP, 283

opciones de coincidencia de tcp, 65

Inundación TCP SYN, 40–41

“Ataques de suplantación de IP y de inundación TCP SYN,”41

archivo tcp-client-policy, 170

volcado TCP

- operadores aritméticos, 271
- primitiva de difusión, 271
- descripción, 249, 265 calificador de dirección, 270–271

TCPDump (*continuación*)

expresiones, 269–271 primitiva
de puerta de enlace, 271 primitiva
mayor, 271 instalación, 266–267
menos primitiva, 271 obtención,
266–267 opciones, 266–269
descripción general, 265–266
primitivas, 271 calificador de
protocolo, 271 calificador de tipo ,
269–270

TCPDump, detección de ataques

Ataque TERRESTRE, 284
Nmap (Network Mapper), 281–282 descripción
general, 280–281 registro de tráfico, 284–286
escaneo de puertos abiertos, 281–282

Ataques de pitufos, 282–283

Indicadores de encabezado TCP, 283

Ataque del árbol de Navidad, 283

volcado TCP, capturando

conversaciones HTTP, 273–277 otros
protocolos basados en TCP, 278–279

pings, 279

consultas, 279

Conversaciones SMTP, 277–278

conversaciones SSH, 278

Modelo de referencia TCP/IP

definición, ubicación
de firewall 360, 27 capas, 6

archivo tcp-server-policy, 170

cadena tcp-state-flags, 151, 165

esquema tcp_wrapper, 360

Ataque de lágrima, 44

Pruebas

como prevención de intrusiones, 259–260 para
puertos abiertos, 260 penetración, 259–260

Programa Snort, 289–290 servidores
web, 260

TFTP (Protocolo trivial de transferencia de archivos), 360

Apretón de manos de tres vías, 16, 361

Tiempo de vida (TTL), 361

Tiempos de espera, inicialización de cortafuegos, 107

Estado TIME_WAIT, 17

TOS (Tipo de Servicio), 361

extensión TOS, 56

campo TOS, 57

extensión del partido tos, 76–77

herramienta de trazado de ruta, 361

NAT tradicional, 198–199

Saliente unidireccional tradicional
NAT, 58

Líneas de base de tráfico, establecimiento, 250

Protocolo de control de transmisión (TCP).

Consulte TCP (Protocolo de control de
transmisión).

Transparencia, cortafuegos, 4

Capa de transporte, 6

Capa de transporte, definición, 361

Modo de transporte, VPN, 231

Protocolos de transporte. Consulte TCP (Protocolo de control
de transmisión); UDP (Protocolo de datagramas de usuario).

Protocolo trivial de transferencia de archivos
(TFTP), 360

TTL (tiempo de vida), 361

Modo túnel, VPN, 231

tuplas, 72

Dos veces NAT, 58, 199

Tipo de Servicio (TOS), 361

Calificador de tipo, 269–270

tu

Definición de UDP (Protocolo de datagramas de usuario) , 14, 361	
inundación, 43 tráfico de cliente	
local, 162 tablas de protocolo, 116–117	
UDP (Protocolo de datagramas de usuario), que permite acceder al servidor DHCP de su ISP, 134–136	
Tipos de mensajes DHCP, 135	
Protocolo DHCP, 136	
descripción general, 134–138	
servidores de hora de red remota, acceso, 136–138	
Expresiones de encabezado UDP, 91	
Opciones de coincidencia de UDP, 66	
"Ataque de denegación de servicio de puerto UDP", 43	
Extensión de objetivo ULOG, 57, 68	
extensión de partido sucio, 77	
Unidifusión, definición, 361	
Direcciones de unidifusión, 9	
Puertos sin privilegios	
definición, 358	
asignaciones de números de puertos oficiales, 113	
rango de puertos, sintaxis, 114 escaneos de puertos, 113 propósito de, 19	
Puertos sin privilegios, servicios de protección en	
bloquear servicios TCP locales, 113–115 servicios TCP locales comunes, 113–115 servicios UDP locales comunes, 116 denegar por defecto, deficiencias, 114 prohibir conexiones, 114–115	
FTP, 114	
asignaciones de números de puertos oficiales, 113	
descripción general, 112–113 rango de puertos, sintaxis, 114	

escaneos de puertos, 113
Tablas de protocolos de servicio TCP, 116–117
Tablas de protocolo de servicio UDP, 116–117
expresión de estado sin seguimiento, 89
Actualización, como prevención de intrusiones, 258–259
bandera URG, 16
Cuentas de usuario, indicaciones de intrusión, 240–241
Protocolo de datagramas de usuario (UDP). Consulte UDP (Protocolo de datagramas de usuario).
Cadena de variables USER_CHAINS, 151
Cadenas definidas por el usuario
bifurcación, 149
características de, 150–151
seguimiento de conexión, 151, 166
verificación de dirección de destino, 151, 168
tráfico DNS, identificación, 157
EXT-icmp-entrada, 152, 164
EXT-icmp-salida, 152, 164
Entrada EXT, 151
Inicio de sesión EXT, 152, 168–170
Cierre de sesión EXT, 152, 168–170
Ext-Output, 151
local_dhcp_client_query, 166-167
local_dns_client_request, 159-161
local_dns_server_query, 151, 158
local_tcp_client_request, 152
local_tcp_server_response, 152, 161-162
local_udp_client_request, 152, 163 Paquetes de registro, 168-170, 175-176
log-tcp-state, 152, 165–166
remote_dhcp_server_response, 152, 166–167
consulta_servidor_dns_remoto, 158
respuesta_servidor_dns_remoto, 151, 159–161
solicitud_cliente_tcp_remoto, 152, 161–162

Cadenas definidas por el usuario

- (*continuación*) remote_tcp_server_response, 152
- remote_udp_server_response, 152, 163
- source-address-check, 151, 167–168
- tcp-state-flags, 151, 165
- Variable USER_CHAINS, 151

protocolo UUCP, 361

V

-v (--versión) opción, 85

Verbosidad de salida, 216

VPN (redes privadas virtuales)

- encabezado de autenticación, 230–231
- combinación con cortafuegos, 233–234
- ESP (carga útil de seguridad encapsulada), 231–232
- encapsulación de enrutamiento genérico, 230
- IKE (intercambio de claves de Internet), 232
- IPSec (Protocolo de Seguridad de Internet), 230
- L2TP (Protocolo de Túnel de Capa 2), 229–230

Libreswan program, 233

Linux, 232–233

programa openswan, 233

Programa OpenVPN, 233

descripción general, 229

PPTP (tunelización punto a punto) protocolo, 229–230, 233

protocolos, 229–232

asociaciones de seguridad, 232

modo de transporte, 231

modo túnel, 231

EN

Servidores web, pruebas, 260

Mundialmente legible, 361

Mundialmente escribible, 361

X

Ventanas X, 361

Ataque del árbol de Navidad, 283

CON

Transferencias de zona, 118

PEARSON

InformIT is a brand of Pearson and the online presence for the world's leading technology publishers. It's your source for reliable and qualified content and knowledge, providing access to the leading brands, authors, and contributors from the tech community.

 Addison-Wesley
  Cisco Press
  IBM Press
  Microsoft Press

PEARSON
IT CERTIFICATION

 PRENTICE
HALL

 que

 SAMS

 vmware PRESS

LearnIT at InformIT

Looking for a book, eBook, or training video on a new technology? Seeking timely and relevant information and tutorials. Looking for expert opinions, advice, and tips? **InformIT has a solution.**

- Learn about new releases and special promotions by subscribing to a wide variety of monthly newsletters. Visit informit.com/newsletters.
- FREE Podcasts from experts at informit.com/podcasts.
- Read the latest author articles and sample chapters at informit.com/articles.
- Access thousands of books and videos in the Safari Books Online digital library. safari.informit.com.
- Get Advice and tips from expert blogs at informit.com/blogs.

Visit informit.com to find out all the ways you can access the hottest technology content.

Are you part of the **IT** crowd?

Connect with Pearson authors and editors via RSS feeds, Facebook, Twitter, YouTube and more! Visit informit.com/socialconnect.





REGISTER



ESTE PRODUCTO

informit.com/registrar

Registre los productos Addison-Wesley, Exam Cram, Prentice Hall, Que y Sams que posee para desbloquear grandes beneficios.

Para comenzar el proceso de registro, simplemente vaya a **informit.com/register** para iniciar sesión o crear una cuenta.

Luego se le pedirá que ingrese el ISBN de 10 o 13 dígitos que aparece en la contraportada de su producto.

El registro de sus productos puede desbloquear los siguientes beneficios:

- Acceso a contenido complementario, incluidos capítulos adicionales, código fuente o archivos de proyectos.
- Un cupón para usar en su

próxima compra.

Los beneficios de registro varían según el producto.

Los beneficios se enumerarán en la página de su cuenta en Productos registrados.

Acerca de InformIT : LA FUENTE DE APRENDIZAJE TECNOLÓGICO DE CONFIANZA

INFORMIT ES EL HOGAR DE LAS IMPRESAS EDITORIALES LÍDERES EN TECNOLOGÍA Addison-Wesley Professional, Cisco Press, Exam Cram, IBM Press, Prentice Hall Professional, Que y Sams. Aquí obtendrá acceso a contenido y recursos confiables y de calidad de los autores, creadores, innovadores y líderes de la tecnología. Ya sea que esté buscando un libro sobre una nueva tecnología, un artículo útil, boletines informativos oportunos o acceso a la biblioteca digital Safari Books Online, InformIT tiene una solución para usted.

informarIT.com

LA FUENTE DE APRENDIZAJE TECNOLÓGICO DE CONFIANZA

Addison Wesley | Prensa de Cisco | examen intensivo
Prensa de IBM | Que | Pasillo de Prentice | sams

LIBROS DE SAFARI EN LÍNEA

LUNES



Linux

MARTES



Kali
Linux

MIÉRCOLES



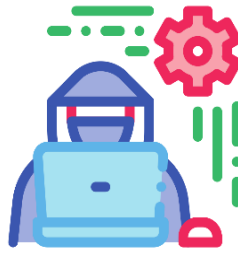
Ciberseguridad

JUEVES



Hacking

VIERNES



Pentesting

SÁBADO



Python

LIBROS DE INFORMÁTICA

EN APRECKING

LIBROS POR SEMANA

