Firewalls Linux

Guía Avanzada

Robert L. Ziegler

Prentice Hall

New Riders

Índice de contenido

Acerca del autor

Robert L. Ziegler se graduó en la Universidad de Wisconsin-Madison con una licenciatura de grado medio en Psicología, además de terminar las licenciaturas de Alemán y Filosofía. Después de dedicarse a la educación y a las carreras estudiadas, decidió hacer de su carrera un hobby y estudió un master en Ciencias de la computación, también en la Universidad de Wisconsin-Madison.

Una vez fuera de la universidad, Bob se convirtió en una de las dos personas de un equipo de programadores de sistemas operativos UNIX que trabajan para una compañía que desarrollaba una mini-supercomputadora. Desarrolló una versión multiprocesador del UNIX BSD 4.3 como un proyecto paralelo a los esfuerzos de desarrollo uniprocesador que tenía en marcha el equipo. Desde entonces, ha trabajado como programador del núcleo de sistemas operativos UNIX para compañías de I+D en el área de Boston.

La llegada de Linux y el acceso del consumidor a la conectividad Internet 24/7 dio a Bob las claves para un sueño que albergaba desde 1982: tener su propio servidor y su propia LAN UNIX en su hogar. Lo que empezó como un esfuerzo pragmático por realizar su propio sistema seguro en Internet, rápidamente creció como una pasión por el usuario particular de UNIX. Ofrece gratuitamente al público, servicios de diseño de firewall (*cortafuegos* o *servidor de seguridad*) de Linux basados en Web, al igual que un firewall popular y unas FAQ de LAN para ayudar a las personas a conseguir configurar rápidamente sus sistemas LINUX de forma segura.

Ahora es ingeniero jefe de Nokia, y está diseñando y programando productos de firewall para la familia de productos Ipsilon de Nokia.

Acerca de los revisores

Debbie Dailey es actualmente administradora de sistemas UNIX para un líder mundial en productos de conexión. Trabajó con mainframes durante seis años antes de volver a la universidad para estudiar Ciencias de la computación, especializándose en seguridad de redes. Debbie empezó su carrera en UNIX como estudiante de prácticas de administración de sistemas mientras asistía a clase. Recientemente ha recibido su licenciatura en Ciencias de la computación por la Universidad de Purdue.

El **Dr. Craig Hollabaugh** ha sido un devoto de UNIX desde su primera conferencia en 1985. Administró estaciones de trabajo de Sun a la vez que seguía un doctorado en ingeniería eléctrica sobre simulación analógica interactiva en el Instituto de Tecnología de Georgia. Luego desarrolló herramientas de interfaz de C++ para el solucionador interactivo de gradientes paralelos distribuido en el laboratorio de dinámica de fluidos de computación de la Universidad de Texas. En 1995, cuando creó su primera compañía (que nació con un capital nulo), Wireless Scientific, desarrolló una aplicación de telemetría industrial inalámbrica centrada en Linux. Posteriormente, su brillante carrera como consultor de Craig le llevó a Cambridge, Massachussets, donde ahora es el vicepresidente de ingeniería en Kiava Systems. Se dedica a supervisar la integración de Kiava y el desarrollo del hardware DSP incrustado. Su interés personal por la investigación se centra en la simulación interactiva en matemáticas elementales y en la enseñanza de la ciencia.

Uno de mis dos amigos de toda la vida murió mientras escribía este libro. A Gloria Frawley, quien a lo largo de la vida ha sido amiga, amante, hermano, hermana, padre, hijo, marido, esposa. Hasta que volvamos a caminar juntos, gracias por el amor más puro, más verdadero y más desinteresado que he conocido, y por ser la mejor compañera que yo nunca pudiera haber esperado.

Agradecimientos

Nunca he comprendido por qué la gente escribe las secciones de agradecimientos. ¿Quién se molesta en leerlas? Bien, ahora comprendo mejor el porqué.

Así, quiero agradecer a Bill Sommerfeld por ofrecerme el primer tutorial cuando se dio cuenta de lo negado que era; a Gary Zaidenweber por recalcarme la necesidad de la seguridad y por ayudarme, en un primer momento, a configurar mi sistema; a mis otros amigos de Hewlett-Packard, que me dieron ánimos para ir detrás de mi pasión, especialmente a Mary MacGregor y a mi administradora Cindy Buhner, quienes son mucho más agradecidas de lo que pueden imaginarse; a Paul Fox por enviarme una copia de su propio firewall cuando estaba empezando esta andadura; a Craig Hollabaugh por su generosidad a la hora de dedicar semanas a la revisión editorial de las FAQ en las que se basa este libro; a Karl Runge por ofrecer semanas de revisión a las reglas de firewall; a numerosos clientes del grupo de noticias NorthEast Mediaone; a mi madre, quien confió en mí para que encontrase mi propio camino cuando pasaba apuros con los problemas propios después de terminar la carrera durante un año; a Regina Weyer por estar siempre allí, a millas de distancia y durante todos los años; a Jonathan Kaplan por ser una fuente de esperanza y apoyo; a Old Person por ser un amigo constante y una fuente de ánimo durante los últimos años; a Alan Small por rehacer amablemente las partes más difíciles del texto de este libro, y, por último, pero no menos importante, a Kitty Jarrett, mi editora, quien de alguna forma sobrevivió a toda mi inexperiencia. Kitty habla de forma suave y sin malicia. Simplemente observa y pregunta, a menudo con humor.

Introducción

Linux disfruta de una popularidad creciente entre los particulares aficionados y entre las pequeñas empresas que se dirigen desde casa. El acceso directo y continuo a Internet está extendiéndose en las casas conforme el modem cable y los servicios de conexión ADSL se expanden en el mercado de consumo.

UNIX no es sólo una plataforma de servidor popular, especialmente para servidores web, sino que es también excelente como pasarela a una LAN particular. Detrás de esa pasarela, continuamente conectada a Internet, hay otras máquinas UNIX, plataformas Windows y NT, Macintosh e impresoras compartidas. Como resultado, los usuarios de sistemas pequeños deben afrontar cuestiones de seguridad en las que nunca antes habían tenido que pensar.

La seguridad de la red es una cuestión especialmente importante para los usuarios de Linux con conexiones directas a Internet. Al contrario que un sistema sencillo de equipos personales, UNIX es un sistema operativo de pleno derecho y potente. Su propósito fundamental y su filosofía fue promover el poder compartir información en un entorno de investigación y desarrollo. Como tal, creció para ser grande y críptico, y no para que lo usaran personas sin experiencia o sin protección.

Conectar un sistema UNIX a Internet es como anunciar una casa abierta al público, dejando la puerta principal abierta de par en par e irse a unas largas vacaciones. Si no se toman precauciones, los intrusos no deseados entrarán en ambos casos y esto ocurrirá más temprano que tarde.

El usuario medio de un sistema particular o de un sistema pequeño no tiene el tiempo, el interés o la paciencia necesaria para aprender todos los aspectos necesarios de la seguridad. El objetivo de este libro es ayudar a los usuarios de Linux y de pequeños negocios a conseguir rápidamente medidas de seguridad de Internet, sin necesidad de convertirse en un experto en seguridad de redes. Las precauciones necesarias no son difíciles de implementar, pero encontrar toda la información en un sitio, poniendo especial énfasis en *cómo hacerlo,* no es una tarea sencilla.

Lo que se explica en este libro

Para los usuarios de pequeños sistemas, las cuestiones de seguridad están relacionadas casi exclusivamente con la seguridad externa, con protegerse a ellos mismos contra accesos de red no deseados desde el exterior. Por ejemplo, algunas familias pueden estar preocupadas en limitar cierta clase de accesos al sistema y a Internet a

sus hijos, pero eso suele ser sólo una parte del problema. Para la mayoría, el entorno particular se considera como un entorno seguro.

Este libro guía a los usuarios particulares y de pequeños negocios en los pasos básicos que deben seguir para diseñar e implementar un firewall de filtrado de paquetes. Sin embargo, un firewall es sólo un paso más para crear un sistema seguro. También son necesarias las medidas de seguridad de alto nivel.

La seguridad del equipo requiere una aproximación de varios niveles. Un solo nivel de un esquema de seguridad no es suficiente por sí mismo. Cada nivel sucesivo depende de las protecciones que ofrecen los niveles inferiores. Los servicios que se habilitan, además del firewall, forman la base para la seguridad del sistema. Por ello, este libro se preocupa de deshabilitar servicios innecesarios, seleccionar los servicios que se harán públicos e identificar los servicios locales peligrosos que debe proteger el firewall.

Se examinan los tipos de protección de firewall que puede soportar fácil y económicamente un sistema pequeño. Entre otros temas se trata la idea que subyace al filtrado a nivel de paquete, cómo configurar su propio firewall, cómo configurar algunos servicios de forma más segura en términos del firewall y protocolos de comunicación, enmascaramiento IP para esconder las identidades de los equipos internos cuando acceden a Internet, y asegurar que el firewall funcione.

Aunque no es el principal tema del libro, la Parte III, "Supervisión y seguridad y a nivel de sistema", explica las formas de más alto nivel de control de acceso. Entre los temas se incluyen listas de control de acceso compatibles con tcp_wrappers y portmap, la configuración del servidor, los servidores proxy y las prácticas generales de administración de sistemas.

Por último, el libro trata la seguridad del sistema y la supervisión de la integridad, detectando sondeos e intentos de acceso no autorizados antes de que tenga lugar una intrusión. También se explican las herramientas para detectar signos de una situación comprometida y cómo solucionarla si se descubre una situación comprometida.

El texto y los ejemplos de este libro se basan en la versión Linux 6.0 de Red Hat. Los ejemplos están escritos con la semántica ipchains. Como la conversión de ipfwadm a ipchains está en proceso actualmente, ya que los sistemas Linux están usando ambas versiones en la actualidad, los ejemplos con semántica ipfwadm se incluyen en el Apéndice B, "Ejemplos de firewalls y secuencias de comandos compatibles".

Lo que no se explica en este libro

Las directivas y los procedimientos de seguridad que una gran empresa necesita enfatizar son casi opuestos a los de un usuario de un sistema pequeño. La seguridad externa de Internet representa sólo un pequeño porcentaje de las cuestiones de seguridad de una empresa grande. Se estima que sobre el 90 % de las violaciones de seguridad de nivel empresarial se originan desde el interior de las LAN corporativas, no desde Internet.

Este libro no intenta solucionar las cuestiones relativas a la seguridad interna del sistema, de los grandes sistemas, de la seguridad LAN multiusuario, de complejas configuraciones de proxy, de métodos y tecnologías de autenticación de nivel corporativo, de redes privadas virtuales, de cifrado, ni de firewall y arquitecturas de red a nivel comercial.

Intentos de piratería informática: ámbito del problema

Resulta difícil conseguir las estimaciones actuales del número de intentos de intrusión. Quizá esto se deba a que los intentos sin éxito en gran parte pasan desaper-

cibidos y muchos sitios se han habituado a ellos y han empezado a considerarlos como habituales en Internet. Las estimaciones en los documentos del CERT oscilan entre un crecimiento constante con el crecimiento de Internet hasta un crecimiento exponencial en 1998.

Cualesquiera que sean los números reales, no hay duda que los intentos globales de piratería en Internet y su nivel de sofisticación están creciendo. Los modelos de las exploraciones de puerto han cambiado desde simples sondeos de unos pocos indicadores de seguridad hasta una exploración de todo el dominio del intervalo completo del puerto del servicio. Las últimas herramientas de piratería se comparten en Internet a través de sitios web, listas de distribución y grupos de noticias. Varios grupos de hackers aficionados usan IRC (Internet Relay Chat) para coordinar exploraciones y ataques en grupo de forma corporativa, a menudo para reducir el riesgo de detección. Algunas de las debilidades recientemente descubiertas se publican rápidamente en Internet e inmediatamente las aprovechan los hackers. Los fabricantes y las organizaciones de supervisión de seguridad se encuentran en una constante carrera con la comunidad hacker, en la que ambos intentan mantenerse un paso por delante del otro.

Qué puede conseguir un hacker informático

Así que, ¿quiénes son esos hackers y qué esperan conseguir? No existe una respuesta sencilla.

Gran parte de la motivación de un intento de piratería es realmente el resultado de la curiosidad, un error, software pobremente escrito y sistemas mal configurados. Mucha de la actividad hacker la originan los adolescentes y estudiantes curiosos. Otra se origina desde sistemas comprometidos, especialmente en sitios universitarios. El propietario del sistema no es consciente de que su equipo lo está usando como base de operaciones un individuo no invitado. También está el grupo poco estructurado de hackers que cooperan entre sí mencionado anteriormente, personas para las que esta es la mejor forma de pasar un buen rato. El espionaje corporativo y político puede ser otra de las motivaciones, incluso para el usuario particular.

Como ejemplo de lo que el hacker espera obtener, algunas personas buscan el reto de resolver el puzzle. Algunos quieren permitirse el lujo de presumir. A otros simplemente les gusta entrar y destruir. También se pueden obtener ganancias tangibles. Una nueva base de operaciones desde la que lanzar posteriores ataques, para que este sitio se vea comprometido como el culpable, es un hallazgo muy importante. De igual forma, un sitio comprometido ofrece una base o recursos de sistema para enviar correos electrónicos de forma masiva. Un objetivo más hostil es encontrar un sitio donde establecer un almacén WAREZ. Por último, está el obvio objetivo del robo, de robar software u otra propiedad intelectual.

Qué tiene que perder

Cuando un sistema está comprometido, al usuario particular medio se le causan molestias y se suele asustar. La perdida de datos es un problema habitual, porque muchos hackers se introducen para borrar los discos duros de las personas. La perdida de datos también afecta a cualquier archivo del que no se haya hecho copia de seguridad, ya que un sistema comprometido debe volver a cargarse desde cero, se haya realizado daño o no.

La perdida del servicio suele ser otro problema común. Un ISP normalmente desconectará la cuenta hasta que se corrija el problema. El propietario del sistema debe primero averiguar el defecto de seguridad y aprender cómo hacer el sistema seguro antes de implementar el procedimiento de seguridad crítico. Esto lleva tiempo. Para

un negocio pequeño, estas dos consecuencias significan la pérdida de ingresos, así como inconvenientes.

No sólo el ISP verá al propietario como sospechoso, sino que el propietario puede sufrir una pérdida de reputación ante cualquiera a quien haya molestado alguna acción lanzada por el hacker desde su sistema. Si el ISP no cree en su inocencia personal, lo eliminará como cliente. Si su sitio ha sido identificado como un sitio WAREZ, o si el hacker atacó los sitios equivocados, puede enfrentarse a problemas legales y a la denuncia pública.

Por último, se puede borrar o difundir la información personal y la información propietaria.

Firewalls frente a la piratería informática en un mundo ideal

Desde un punto de vista conceptual, muchos o la mayoría de los intentos de piratería los puede detener el ISP o los proveedores de servicio de pasarela en el origen. El uso de un conjunto estándar de procedimientos de filtrado, aplicados en enrutadores y pasarelas, terminarán con la mayoría de este tipo de intentos de violación de la seguridad. Por desgracia, esto sólo es práctico en un mundo ideal, por ahora. No sólo será necesario convencer a todos los proveedores de servicio de cualquier lugar de su función y responsabilidad en el esfuerzo, sino que habrá que habilitar a los enrutadores de red para manejar la carga extra de filtrado de paquetes a escala masiva. Todavía no nos hemos referido al hardware.

Sin embargo, estas clases de procedimientos de filtrado pueden implementarse fácilmente en sistemas particulares y de pequeños negocios sin ninguna degradación palpable del rendimiento. Estos procedimientos no sólo ayudarán a mantener un sitio más seguro, sino que también protegerán a otras personas de los errores.

I

Consideraciones previas

1 Conceptos básicos subyacentes a los firewalls de filtrado de paquetes.

1

Conceptos básicos subyacentes a los firewalls de filtrado de paquetes

Un sitio pequeño puede tener acceso a Internet a través de cable modem, una línea ASDL o, a menudo, a través de una conexión PPP a una cuenta de acceso telefónico. El equipo conectado directamente a Internet es el centro de las cuestiones de seguridad. Si se dispone de un equipo o una red de área local (LAN, Local Area Network) pequeña de equipos conectados, el centro de un sitio pequeño será la máquina que tiene la conexión directa a Internet. Esta máquina será la máquina firewall.

El término *firewall (cortafuegos o servidor de seguridad)* tiene varios significados dependiendo de su implementación y de su propósito. En este momento inicial del libro, utilizaremos el término firewall para referirnos a la máquina conectada a Internet. Aquí es donde se implementarán las directivas de seguridad. La tarjeta de interfaz de red externa de la máquina firewall es el punto de conexión, o pasarela, a Internet. El objetivo de un firewall es proteger lo que hay en el lado del usuario de esta pasarela de lo que hay al otro lado.

Una configuración simple de firewall suele recibir el nombre de *firewall bastión*, debido a que es la principal línea de defensa contra cualquier ataque desde el exterior. Todas las medidas de seguridad se implementan desde este defensor de su entorno. En consecuencia, es el que hace todo lo posible para proteger el sistema. Es el primer y único bastión de defensa.

Detrás de esta línea de defensa se encuentra el equipo o el grupo de equipos del usuario. El propósito de la máquina firewall puede ser simplemente

servir como punto de conexión a Internet para otras maquinas de la LAN. Quizá detrás de este firewall, se ejecuten servicios privados locales, como una impresora compartida o un sistema de archivos compartido. Quizá se quiera que todos los equipos tengan acceso a la World Wide Web y que una de las máquinas albergue los registros financieros privados. También se puede querer tener acceso a Internet desde esta máquina, pero que nadie más disponga de este servicio. En algún momento, se querrán ofrecer servicios propios a Internet. Una de las máquinas podría estar albergando un sitio web propio para Internet. La configuración y objetivos particulares determinarán las directivas de seguridad.

El propósito del firewall es hacer cumplir unas determinadas directivas de seguridad. Estas directivas reflejan las decisiones que se han tomado sobre qué servicios de Internet deben ser accesibles a los equipos, qué servicios se quieren ofrecer al exterior desde los equipos, qué servicios se quieren ofrecer a usuarios remotos o sitios específicos y qué servicios y programas se quieren ejecutar localmente para uso privado. Todas las directivas de seguridad están relacionadas con el control de acceso y uso autenticado de servicios privados o protegidos y de programas y archivos en los equipos.

Los sistemas particulares y de pequeños negocios no se enfrentan directamente con todas las cuestiones de seguridad de un gran sitio corporativo, aunque las ideas básicas y los pasos sean los mismos: simplemente no existen tantos factores a considerar y se presta especial atención a la protección del sitio de los intentos ilícitos de acceso desde Internet. Un sitio corporativo debe prestar especial atención a la seguridad interna, cosa que no sucede con un sitio particular. Un firewall para filtrado de paquetes es una aproximación común y una pieza de la seguridad de red y del control del acceso desde el exterior.

Antes de entrar en detalles de cómo programar un firewall, este capítulo explica los conceptos subyacentes y mecanismos básicos en los que se basa un firewall de filtrado de paquetes. Estos conceptos incluyen un marco general o referencia en el que se analiza para qué sirve la comunicación de redes, cómo se identifican los servicios basados en red, qué es un paquete y los tipos de mensajes e información que se envían entre los equipos de una red.

El modelo de referencia TCP/IP para redes

Con el fin de poder ofrecer un entorno de trabajo para este capítulo, y para el resto del libro, vamos a usar algunos términos antes de definirlos en las siguientes secciones de este capítulo. Estas definiciones son ya conocidas en el mundo informático y no serán ajenas para las personas que trabajan con redes, pero pueden ser nuevas para los menos iniciados técnicamente. Si todo esto es nuevo para el lector, no se preocupe. Ahora mismo, lo único que pretendemos es ofrecerle un marco conceptual donde colocar las próximas definiciones, de forma que tengan más sentido.

Si tiene conocimientos académicos formales en redes, estará familiarizado con el modelo de referencia (ISO) de interconexión de sistemas abiertos. El modelo de referencia ISO se desarrolló a finales de los 70 y principios de los 80 para ofrecer un marco para los estándares de interconexión de redes. El modelo de referencia ISO es un modelo académico, formal y elegante. Los libros de texto y los académicos usan este modelo como marco conceptual cuando hablan de redes.

Las redes comenzaron a implantarse a finales de los 70 y principios de los 80, de forma que el mundo siguió su curso durante los siete años que el modelo de referencia ISO se estuvo preparando. Como TCP/IP se convirtió en el estándar de facto para la comunicación en Internet entre máquinas UNIX durante este tiempo, se desarrolló un segundo modelo informal llamado modelo de referencia TCP/IP. En vez de ser un ideal académico, el modelo de referencia TCP/IP lo elaboraron los fabricantes y los programadores que, por fin, llegaron a un acuerdo sobre las comunicaciones en Internet. Además, este modelo es más sencillo que el modelo ISO, por que desarrolla TCP/IP desde el punto de vista del programador, partieno de lo que es el mundo real y práctico. Por tanto, donde ISO traza explícitamente siete capas, el modelo TCP/IP las agrupa en cuatro.

Este libro usa el modelo de referencia TCP/IP. Como la mayoría de las personas con cierto bagaje en informática, tendemos a usar el vocabulario ISO, pero para referirnos al modelo conceptual de TCP/IP.

La comunicación de red se explica como un modelo de capas, donde la comunicación se realiza entre capas adyacentes sobre un equipo individual, y entre capas paralelas sobre equipos que se comunican. El programa que se ejecuta (por ejemplo, un navegador Web) se encuentra en la parte superior, en la capa de aplicación, comunicándose con otro programa sobre otro equipo (por ejemplo, un servidor web).

Para que la aplicación cliente del navegador web envíe una petición de una página Web a la aplicación del servidor web, tiene que usar llamadas de biblioteca y de sistema para obtener la información del navegador web y encapsularla en un mensaje apropiado para poder transportarlo entre dos programas por la red. Estos mensajes son segmentos TCP o datagramas UDP de la capa de transporte. Para construir estos mensajes, la capa de aplicación llama a la capa de transporte para que ofrezca este servicio. Los mensajes de la capa de transporte saben cómo entregar mensajes entre un programa de un equipo y un programa situado en el otro extremo de la red. Tanto el modelo ISO como el modelo TCP/IP llaman a esta capa la capa de transporte, aunque el modelo ISO divide esta capa en varias capas funcionales.

Para que los mensajes de la capa de transporte se entreguen entre los dos programas, es necesario enviar los mensajes entre los dos equipos. Para ello, la capa de transporte hace uso de funciones del sistema operativo que toman el mensaje de transporte TCP o UDP y lo encapsulan en un datagrama de Internet adecuado para enviarlo al otro equipo. Estos datagramas son paquetes IP. Los paquetes IP de Internet se envían entre los dos equipos a través de Inter-

net. La capa de Internet sabe cómo comunicarse con el equipo situado en el otro extremo de la red. El modelo de referencia TCP/IP llama a esta capa la capa de Internet. Para esta capa se suele usar el vocabulario del modelo de referencia ISO, por lo que se suele llamar la capa de red. Los dos son uno y el mismo.

Debajo de la capa de red está la capa de subred. De nuevo, el paquete se encapsula en un encabezado Ethernet. En la capa de subred, el mensaje se llama ahora trama de Ethernet. Desde el punto de vista de TCP/IP, la capa de subred es un conjunto de todo lo que sucede para conseguir que el paquete se entregue al siguiente equipo. Este conjunto incluye todo el direccionamiento y los detalles de entrega asociados con el enrutamiento de la trama entre los equipos, de un enrutador al siguiente, hasta que, por último, se alcanza el equipo destino. Esta capa incluye la traducción de la trama de red de una clase de red a otra a lo largo del camino. La mayoría de las redes actuales son redes Ethernet, pero también existen redes ATM, redes FIDI, redes Token Ring, etc., es decir, cualquier tecnología de red que se esté usando para transportar las tramas entre dos equipos. Este grupo incluye el hardware, los cables físicos que conectan dos equipos, las señales y el cambio del voltaje, que representan los bits individuales de una trama y la información de control necesaria para crear una trama a partir de un byte individual.

La idea general, como se muestra en la Figura 1.1, es que la capa de aplicación representa la comunicación entre dos programas. La capa de transporte representa cómo se realiza la comunicación entre los dos programas. Los programas se identifican por números llamados *puertos de servicio.* La capa de red representa cómo se transporta esta comunicación entre los dos equipos terminales. Los equipos, o sus tarjetas de interfaz de red individuales, se identifican por números llamados *direcciones IP.* La capa de subred representa cómo se transporta la comunicación entre cada equipo individual a lo largo de la trayectoria. En una red Ethernet, estas interfaces de red del equipo se identifican por números llamados *direcciones Ethernet,* con las que puede que el usuario esté familiarizado, como la dirección MAC de hardware impresa en la tarjeta de red.

La siguiente sección muestra la información que deben intercambiar estas capas y que termina usando un firewall de filtrado de paquetes.

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 los que pueden acceder otros equipos de la red. Los puertos de servicio identifican los programas y las sesiones individuales o las conexiones que se realizan. Los puertos de servicio son los nombres numéricos para los diferentes servicios de red. También se usan como identificadores numéricos para los extremos finales de una conexión particular. Los números de puerto de servicio van desde 0 hasta el 65535.

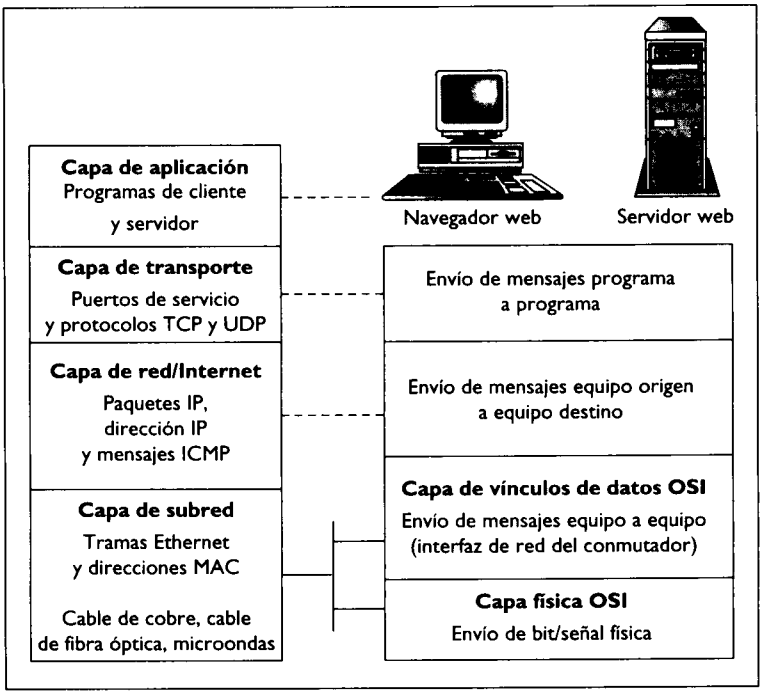

Figura 1.1. Modelo de referencia TCP/IP.

Los programas de servidor (por ejemplo, los *demonios*) escuchan las conexiones entrantes en un puerto de servicio asignado a ellos. Por convención histórica, los servicios de red principales están asignados a números de puerto bien conocidos, en el intervalo inferior, desde 1 a 1023. Estas asignaciones numéricas de puerto a servicio las coordina la Autoridad de asignación de números de Internet (IANA, *Internet Assigned Numbers Authoriry*) como un conjunto de convenciones o estándares universales de mutuo acuerdo.

Estos puertos de intervalo inferior se llaman *puertos privilegiados* porque sus propietarios son programas que ejecutan privilegios del nivel de sistema (es decir, superusuario o root). La idea es incrementar la confianza en que el programa del cliente está realmente conectado al servicio anunciado que se desea. Esta es la intención, pero no una garantía. El usuario nunca puede tener la absoluta certeza de que una máquina remota o un servicio remoto es quien o lo que dice ser.

Un servicio anunciado es simplemente un servicio disponible en Internet desde su puerto asignado. Si una máquina no ofrece un servicio particular y alguien intenta conectarse al puerto asociado con dicho servicio, no ocurrirá nada. Están llamando a la puerta, pero nadie vive allí para abrirle. Por ejemplo, los servidores web están asignados al puerto 80. Si una máquina no ejecuta un servidor web y alguien intenta conectarse al puerto 80, el programa

cliente recibe un mensaje de error procedente de la máquina indicando que ese servicio no está disponible.

Los números de puerto superiores, desde 1024 hasta 65535, se llaman *puertos no privilegiados*. Sirven para un propósito doble. Casi siempre, estos puertos se asignan de forma dinámica al cliente de una conexión. La combinación de pares de números de puerto cliente y servidor, junto con sus respectivas direcciones IP, identifican unívocamente la conexión.

Además, los puertos en el intervalo de 1024 hasta 49151 están registrados por el IANA. Estos puertos pueden usarse como parte del conjunto no privilegiado general, pero también se asocian a servicios particulares como SOCKS o servidores X Window. En un principio, la idea era que esos servicios que se ofrecían en los puertos superiores no se ejecutaran con privilegios de root. Eran para que los usaran los programas no privilegiados, de nivel de usuario. En cada caso concreto, se puede mantener o no la convención.

Asignación de nombres de servicio a número de puerto

Las versiones de Linux incluyen una lista de los 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 a él, el protocolo (TCP o UDP) sobre el que se ejecuta el servicio y cualquier alias opcional para el servicio. La siguiente tabla muestra algunas asignaciones habituales de nombre de servicio a número de puerto, tomada de la versión 6.0 de Red Hat.

Nombre del puerto	Número de puerto/Protocolo	Alias
ftp	21/tcp	
telnet	23/tcp	
smtp	25/tcp	mail
whois	43/tcp	nicname
domain	53/tcp	nameserver
domain	53/udp	nameserver
finger	79/tcp	
pop-3	110/tcp	
nntp	119/tcp	readnews
www	80/tcp	http
auth	113/tcp	ident
ntp	123/udp	
https	443/tcp	

Observe que los nombres simbólicos asociados con los números de puerto varían según la versión de Linux. Los nombres y los alias difieren. Los números de puerto no difieren.

Tenga también en cuenta que los números de puerto están asociados con un protocolo. La IANA ha intentado asignar el mismo número de puerto de servicio tanto al protocolo TCP como al protocolo UDP, sin tener en cuenta si un servicio particular usa ambos protocolos. La mayoría de los servicios usan un protocolo u otro. El servicio *domain* usa ambos protocolos.

Paquetes: mensajes de red IP

El término *paquete* hace referencia a un mensaje de red (IP) del protocolo de Internet. El estándar IP define la estructura del mensaje que se envía entre dos equipos sobre la red. Es el nombre que se asigna a un mensaje discreto y sencillo o la pieza de información que se envía sobre una red Ethernet. Estructuralmente, un paquete contiene un encabezado de información y un cuerpo de mensaje que contiene los datos que se transfieren. El cuerpo del paquete IP, sus datos, forman un solo mensaje de un protocolo de nivel superior o una pieza (un fragmento) de éste.

El mecanismo firewall IP (IPFW) que se incluye en Linux es compatible con tres tipos de mensajes IP: ICMP, UDP y TCP. Un paquete ICMP (Internet Control Message Protocol, Protocolo de mensajes de control de Internet) es un mensaje IP de control y estado del nivel de red. A nivel de red, los mensajes ICMP contienen información sobre la comunicación entre los dos equipos finales. Un paquete IP UDP transporta datos de la capa de transporte UDP entre dos programas basados en red, sin ninguna garantía de éxito en cuanto a la entrega o sobre el orden de la entrega de paquetes. El envío de un paquete UDP a otro programa es como enviar una postal. Un paquete IP TCP (Transmisión Control Protocol, Protocolo de control de transmisión) también transporta datos de la capa de transporte TCP entre dos programas de red, pero el encabezado del paquete contiene información adicional de estado para mantener una conexión confiable y activa. El envío de un paquete TCP es muy parecido a mantener una conversación telefónica con otro programa. La mayoría de los servicios de red de Internet usan el protocolo de comunicación TCP en vez del protocolo de comunicación UDP. En otras palabras, la mayoría de los servicios de Internet se basan en la idea de una conexión activa con comunicación bidireccional entre un programa cliente y un programa servidor.

Todos los encabezados de paquetes IP contienen las direcciones origen y destino IP y el tipo de mensaje de protocolo IP (ICMP, UDP o TCP) que contiene el paquete. Además de esto, el encabezado de un paquete contiene campos algo distintos en función del tipo de protocolo. Los paquetes ICMP contienen un campo tipo que identifica el tipo de mensaje de control o estado, junto con un segundo campo código que define el mensaje de forma más específica. Los paquetes UDP y TCP contienen números de puerto de servicio origen y destino. Los paquetes TCP contienen información adicional sobre el estado de la conexión e identificadores únicos para cada paquete. El encabezado también contiene otros campos. Los demás campos no son visibles o útiles generalmente en la capa de filtrado de paquetes IP. (casi todos los libros sobre redes TCP/IP define los restantes campos del encabezado. Una referencia es ***TCP/IP Clearly Explained*** de Pete Loshin, publicado por Academic Press).

Tipos de mensajes IP: ICMP

Un paquete ICMP es un mensaje de control y estado IP de la capa de red. Su encabezado contiene las direcciones IP origen y destino, el identificador

de protocolo ICMP y un tipo de mensaje ICMP. Un tipo de mensaje ICMP indica si el paquete es un comando, una respuesta a un comando, información de estado o una condición de error. Los tipos de mensaje individuales ICMP se explican en el Capítulo 3, "Creación e instalación de un firewall". Los mensajes ICMP no contienen puertos origen y destino. No se envían entre programas, sino entre los equipos origen y destino.

Un ejemplo de mensaje ICMP es un mensaje de estado de error. Si intenta conectarse a un servidor web remoto y el servidor web no se está ejecutando en el host remoto, el host remoto devolverá un mensaje de error ICMP indicando que el servicio no existe. Los campos de interés del encabezado del paquete se muestran en la Figura 1.2.

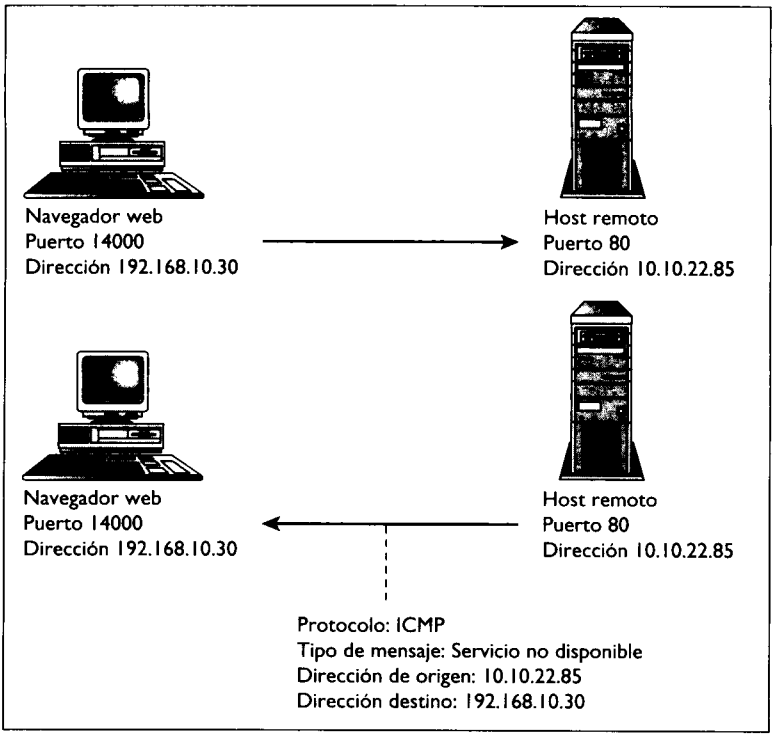

Figura 1.2. Mensaje de error ICMP entrante.

Tipos de mensajes IP: UDP

UDP es un protocolo de entrega de transporte sin estado (es decir, sin conexión) y no confiable. Un programa envía un mensaje UDP, que puede recibirse o no y responder a él o no. No se devuelve acuse de recibo. No se ofrece control de flujo, por lo que un datagrama UDP simplemente se elimina si no se puede procesar durante el camino.

Un encabezado de paquete UDP contiene las direcciones IP origen y destino, el tipo de protocolo UDP y los números de puerto de servicio origen y destino.

¿No es UDP confiable?

UDP se conoce como un *servicio de datagrama no confiable.* No confiable no implica ninguna connotación negativa para la utilidad de UDP. No confiable, en este caso, significa que no se realiza ningún esfuerzo para asegurar la entrega correcta. Por ello, sin la sobrecarga de TCP orientada a la conexión y confiable, la trasferencia de datos basada en UDP es varias veces más rápida que la transferencia de datos basada en TCP. Por ello, UDP se presta a tipos de comunicación sencillas de petición y respuesta.

Como ejemplo, la máquina puede intentar periódicamente contactar con un servidor remoto de tiempo de la red. Los servidores de tiempo de Internet tienen asignado el puerto de servicio UDP 123. El servidor de tiempo devolverá la hora actual. El sistema del usuario puede actualizar el reloj con la información más precisa procedente del servidor de tiempo. Si el intercambio no tiene éxito porque el servidor de tiempo no está activo en la máquina remota, o porque la máquina remota está apagada en ese momento, se devuelve un mensaje de error ICMP, Servicio o host no disponible.

Como se muestra en la Figura 1.3, el programa cliente del sistema inicia la petición. Se asigna un puerto no privilegiado a la petición. Se construye una petición de paquete UDP saliente con el puerto no privilegiado como puerto origen. El puerto destino es el conocido puerto de servicio de tiempo, 123.

El servidor de tiempo responde con un paquete UDP que contiene la hora actual. Llegará un paquete de respuesta UDP entrante desde la dirección IP del servidor originado en el puerto 123, dirigido a la dirección IP y al puerto no privilegiado destino proporcionado inicialmente.

Tipos de mensajes IP: TCP

La gran mayoría de servicios de red se ejecutan sobre TCP. Los mensajes se envían de forma confiable en ambas direcciones como parte de una conexión activa entre dos programas; se entregan sin error, sin pérdidas o duplicaciones, y de forma ordenada. Cada segmento TCP se confirma después de recibirse. Cada segmento TCP se identifica mediante un numero de secuencia único. Se usan los bits indicadores para definir el estado de la conexión.

Si un segmento TCP IP encapsulado es más largo que la unidad de transmisión máxima (MTU, Maximum Transmission Unit) de la red subyacente, el segmento se divide en fragmentos (como en la mayoría de las redes Ethernet, la MTU es de 1500 bytes como máximo por trama Ethernet). Los fragmentos se identifican como pertenecientes a un segmento particular y se envían in-

dividualmente, para que el equipo destino final los reconstruya en el segmento TCP original.

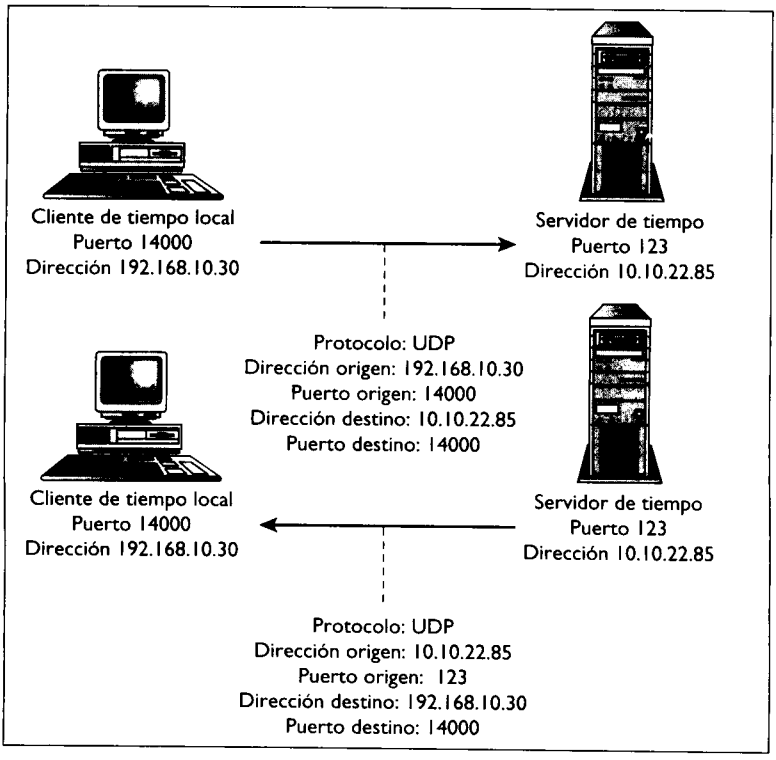

Figura 1.3. Petición y respuesta UDP.

Cuando un programa cliente inicia una conexión a un servidor, se selecciona un puerto del grupo no privilegiado del cliente. La combinación de la dirección IP del cliente y del número de puerto definen el socket del cliente. En el lado del servidor, la combinación de dirección IP de host y el número de puerto conocido del servidor forman el socket del servidor. La conexión entre cliente y servidor se define de forma unívoca mediante este par de sockets.

Cada conexión individual entre un cliente dado y un servidor, quizá sólo una en un conjunto de conexiones simultaneas a dicho servidor (por ejemplo, un servidor web), se define unívocamente mediante la dirección origen y el número de puerto del cliente, junto con la dirección IP del servidor y el número de puerto asignado.

El encabezado de un paquete TCP contiene las direcciones origen y destino IP, el tipo de mensaje de protocolo TCP, los puertos de servicio destino y los números de secuencia y de confirmación, así como los indicadores de control que se usan para crear y mantener un circuito virtual confiable, o conexión bidireccional activa.

Una conexión TCP típica: cómo visitar un sitio web remoto

Un ejemplo habitual de conexión TCP consiste en dirigirse a un sitio web usando el navegador de Netscape (es decir, conectarse a un servidor web). Esta sección muestra los aspectos de establecimiento de conexión y de comunicación activa que serán importantes para el filtrado de paquetes IP en los capítulos posteriores.

¿Qué sucede? Como se muestra en la Figura 1.4, se ejecuta un servidor web en una máquina situada en cualquier lugar, esperando una petición de conexión sobre el puerto de servicio TCP 80. El usuario hace clic en el vínculo de un URL en Netscape. Parte del URL se analiza sintácticamente como un nombre de host, el nombre de host se traduce en la dirección IP del servidor web; y se asigna un puerto no privilegiado al explorador (por ejemplo, 14000) para la conexión. Se crea un mensaje HTTP para el servidor web. El mensaje se encapsula en un mensaje TCP, dentro de un encabezado de paquete IP y se envía al exterior. En este caso, el encabezado contiene los campos que se muestran en la Figura 1.4.

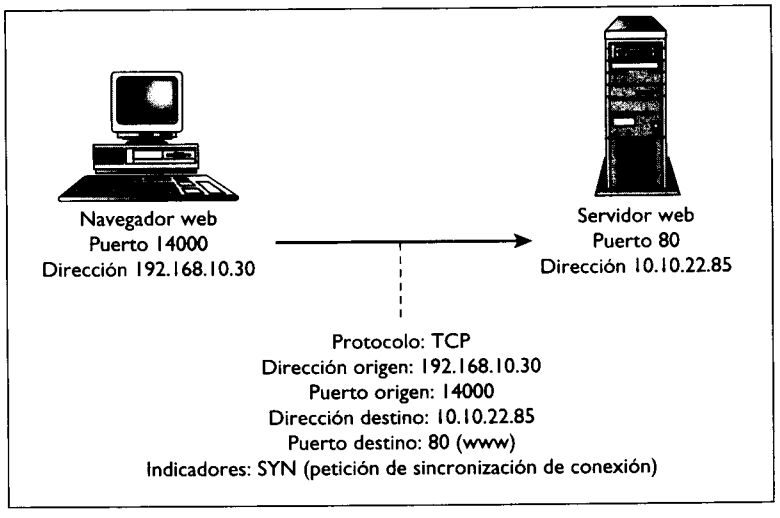

Figura 1.4. Petición de conexión de un cliente TCP.

En el encabezado se incluye información adicional que no es visible a nivel de filtrado de paquetes. Sin embargo, describir los números de secuencia asociados con los indicadores SYN y ACK ayuda a clarificar lo que ocurre durante el saludo de tres vías. Cuando el programa cliente envía el primer mensaje de petición de conexión, se incluye un número de secuencia de sincronización con el indicador SYN. El cliente solicita una conexión con el

servidor y la envía junto con un número de secuencia inicial que usará como base para numerar el resto de los mensajes que debe enviar.

El servidor recibe el paquete. Se envía al puerto de servicio 80. El servidor escucha el puerto 80, por lo que se le notifica una petición de conexión entrante (el indicador de petición de sincronización SYN) desde la dirección IP origen y del par de puertos de socket (192.168.10.30). El servidor asigna un nuevo socket en su extremo (10.10.22.85) y lo asocia con el socket del cliente.

El servidor web responde con una confirmación (ACK) al mensaje SYN, junto con su propia petición de sincronización (SYN), como se muestra en la Figura 1.5. Ahora la conexión está semiabierta.

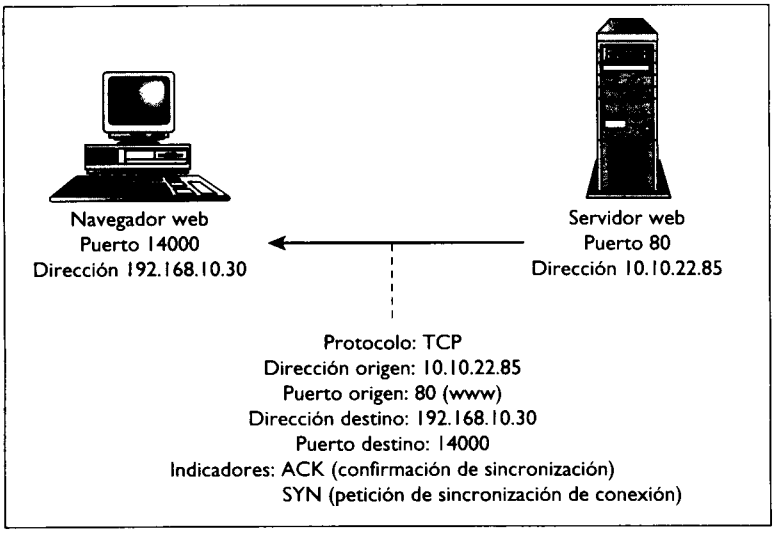

Figura 1.5. Confirmación de petición de conexión del servidor TCP.

Dos campos no visibles a nivel de filtrado de paquetes se incluyen en el encabezado SYN-ACK. Junto con el indicador ACK, el servidor incluye el número de secuencia del cliente incrementado en uno. El propósito de la confirmación es reconocer el mensaje al que el cliente hace referencia por su número de secuencia. El servidor lo confirma incrementando el número de secuencia del cliente, lo que indica que recibió el mensaje, y el número de secuencia +1 es el siguiente mensaje que el servidor espera recibir. El cliente puede enviar una copia del mensaje SYN original, ahora que el servidor ha recibido un acuse de recibo del mismo.

El servidor también define el indicador SYN en el primer mensaje. Con el primer mensaje del cliente, se incluye un numero de secuencia de sincronización con el indicador SYN. El servidor envía 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 activado. Éste y todos los mensajes siguientes tienen activado el indicador ACK. La presencia del indicador ACK en todos los mensajes de servidor, cuando se compara con la ausencia de un indicador ACK en el primer mensaje de cliente, será una diferencia importante cuando queramos obtener la información disponible para construir un firewall.

La máquina del usuario recibe este mensaje y responde a él con su propia confirmación, después de lo cual, se establece la conexión. La Figura 1.6 muestra un esquema de esta situación. De ahora en adelante, tanto el cliente como el servidor activan el indicador ACK. El indicador SYN no lo volverá a activar ningún programa.

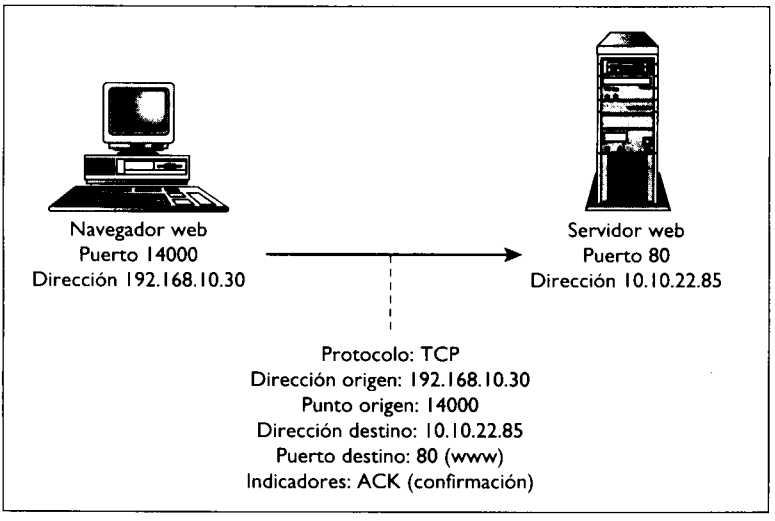

Figura 1.6. Establecimiento de la conexión TCP.

Con cada confirmación, los programas cliente y servidor incrementan su número de secuencia del proceso asociado en el número de paquetes contiguos secuenciales recibidos desde que se envió el último ACK, indicando la recepción de los paquetes, así como el siguiente mensaje que el programa espera recibir.

A medida que el navegador recibe la página Web, la máquina recibe mensajes de datos desde el servidor web con encabezados de paquete, como se muestra en la Figura 1.7.

Al final, se envían los mensajes adicionales con indicadores para cerrar la conexión, pero estos indicadores no están disponibles a nivel de filtrado de paquetes. Los indicadores importantes son SYN y ACK. El indicador SYN se activa cuando un cliente y un servidor intercambian los primeros dos mensajes durante el establecimiento de la conexión. Todos los mensajes siguientes entre cliente y servidor tienen el conjunto de indicadores ACK.

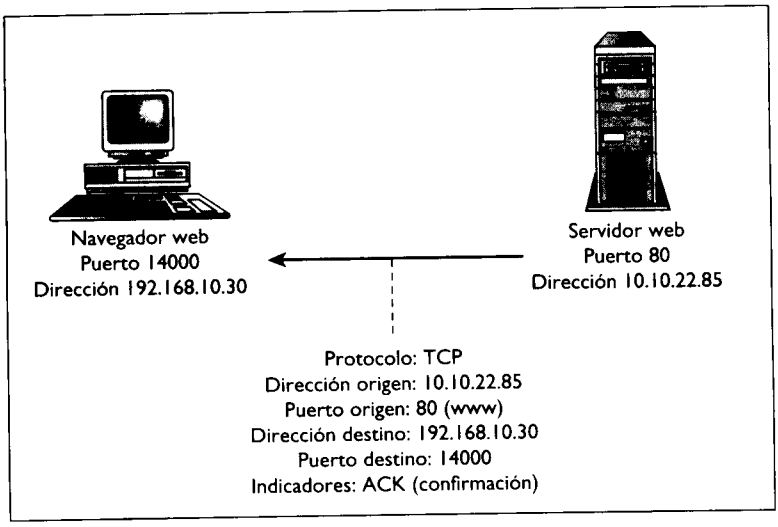

Figura 1.7. Conexión servidor a cliente TCP en marcha.

Resumen

Los sencillos ejemplos de este capítulo muestran la información en la que se basan los firewalls de filtrado de paquetes. El Capítulo 2, "Conceptos de filtrado de paquetes", profundiza en esta introducción, describiendo cómo se usan los tipos de mensaje ICMP, UDP y TCP y los números de puerto de servicio para definir un firewall de filtrado de paquetes.

II

Filtrado de paquetes y medidas básicas de seguridad

2 Conceptos del filtrado de paquetes.
3 Creación e instalación de un firewall.
4 Redes de perímetro, firewalls múltiples y problemas con los LAN.
5 Depuración de las reglas del firewall.

2

Conceptos del filtrado de paquetes

El término *firewall* tiene varios significados, dependiendo del mecanismo que se use para implementarlo, el nivel de la pila del protocolo TCP/IP sobre el que funciona el firewall y las arquitecturas de red y enrutamiento que se usen. Tres de los significados más comunes se refieren a un firewall de filtrado de paquetes, a una pasarela de aplicación, también llamada un firewall de host explorado, y a una pasarela de circuito del nivel de aplicación, también llamada firewall proxy.

Un firewall de filtrado de paquetes se suele implementar dentro del sistema operativo y funciona en las capas de transporte y red de la red IP. Protege el sistema realizando las decisiones de enrutamiento después de filtrar los paquetes basándose en la información del encabezado del paquete IP.

Una pasarela de aplicación, o firewall de host explorado, se implementa en los niveles de arquitectura de red y configuración del sistema. El tráfico de red nunca pasa a través de la máquina de aplicación de pasarela. Todo el tráfico, interno y externo, debe realizarse a través de la máquina de pasarela. Los usuarios locales deben iniciar la sesión en la máquina de pasarela y acceder a Internet desde allí. Además, la máquina de pasarela puede estar protegida mediante un firewall de filtrado de paquetes tanto en la interfaz externa como en la interna.

Un firewall proxy se suele implementar como aplicación independiente para cada servicio con el que se desea usar un proxy. Cada aplicación proxy aparece ante el servidor como el programa cliente, y ante el cliente como el servidor real. Los programas cliente especiales, o programas de cliente configurados especialmente, se conectan al servidor proxy en vez de a un servi-

dor remoto. El proxy establece la conexión al servidor remoto en el lado de la aplicación cliente, después de sustituir la dirección origen del cliente por la suya propia. Las aplicaciones proxy pueden asegurar la integridad de los datos, es decir, que se están intercambiando los datos apropiados para el servicio, que se filtran contra virus y se hacen cumplir las directivas de alto nivel de control de acceso detallado.

Este libro explica las ideas en las que se basa un firewall de filtrado de paquetes. Las tres perspectivas controlan a qué servicios se puede acceder y por quién. Cada perspectiva tiene sus puntos fuertes y ventajas, que se basan en la distinta información disponible en las diferentes capas del modelo de referencia TCP/IP. Los grandes productos firewall comerciales suelen incorporar alguna combinación de filtrado de paquetes, host explorados protegidos y funcionalidad proxy para aplicaciones en un paquete de seguridad multinivel.

El Capítulo 1, "Conceptos básicos subyacentes a los firewalls de filtrado de paquetes", explicó los conceptos y la información en la que se basa un firewall. Este capítulo explica cómo se usa esta información para implementar reglas de firewall. Gran parte de este capítulo se centra en el filtrado de paquetes entrantes, los tipos de paquetes que se deben filtrar y por qué. Se hace menos énfasis en el filtrado de paquetes salientes ya que un sitio pequeño es un entorno razonablemente seguro sin las cuestiones de seguridad internas adicionales a las que debería hacer frente una organización más grande. De hecho, no todos los firewall comerciales ofrecen la capacidad de filtrar paquetes salientes. Esto se debe en parte al apreciable coste de rendimiento que el filtrado de paquetes impone a los grandes enrutadores. El hardware no es lo suficientemente rápido, todavía.

Tanto el Capítulo 1 como este capítulo explican con detalle los puertos de servicio TCP y UDP. La última sección de este capítulo, "Servicios de red públicos frente a privados", describe los servicios de red comunes que existen en una máquina UNIX y qué hacen, así como algunas recomendaciones, por si quisiera ejecutar estos servicios sobre una máquina firewall.

Un Firewall de filtrado de paquetes

Un firewall de filtrado de paquetes IPFW consta de una lista de reglas de aceptación y denegación. Estas reglas definen explícitamente los paquetes que se permiten pasar y los que no a través de la interfaz de red. Las reglas del firewall usan los campos del encabezado del paquete, explicadas en el Capítulo 1, para decidir si enrutar un paquete hacia su destino, eliminar el paquete o bloquear un paquete y devolver una condición de error a la máquina emisora. Estas reglas se basan en la tarjeta de interfaz de red específica y en la dirección IP del host, las direcciones IP origen y destino del nivel de red, los puertos de servicio UDP y TCP de la capa de transporte, los indicadores de conexión TCP, los tipos de mensaje ICMP del nivel de red y en si el paquete es entrante o saliente.

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

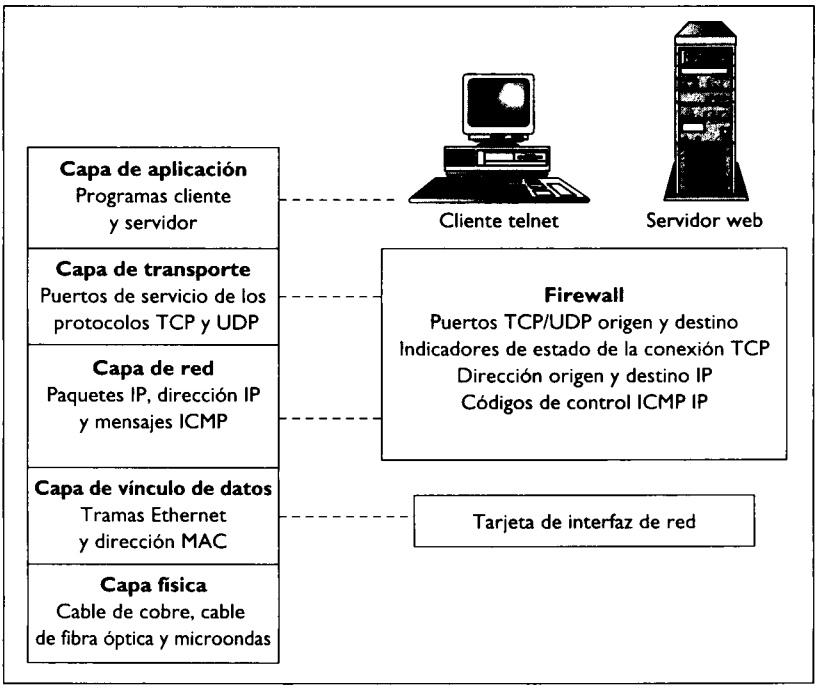

Figura 2.1. Ubicación del firewall en el modelo de referencia TCP/IP.

La idea general es que el usuario debe controlar con mucho cuidado lo que sucede entre Internet y la máquina que se ha conectado directamente a Internet. Sobre la interfaz externa a Internet, el usuario filtrará individualmente lo que procede del exterior y lo que sale de la máquina de forma tan precisa y explícita como sea posible.

Para una configuración de máquina sencilla, podría servir de ayuda pensar en la interfaz de red como en un par E/S. El firewall filtra independientemente lo que entra y lo que sale a través de la interfaz. El filtrado de entrada y el filtrado de salida pueden tener reglas completamente diferentes. Las listas de reglas que definen lo que puede entrar y lo que puede salir se llaman cadenas. El par E/S es la lista de reglas tanto de la cadena de entrada como de la cadena de salida. Las listas se llaman cadenas porque se compara un paquete con cada regla de la lista, una a una, hasta que se encuentra una coincidencia o la lista se termina, como se describe en la Figura 2.2.

Esto suena bastante fuerte, y lo es, pero no es un mecanismo de seguridad infalible. Es sólo parte del problema, una capa de todo el esquema de seguridad. No todos los protocolos de comunicación de aplicación se prestan

para el filtrado de paquetes. Este tipo de filtrado pertenece a un nivel demasiado bajo como para permitir autenticación y control de acceso preciso. Estos servicios de seguridad deben proporcionarse a niveles más altos. IP no tiene la capacidad de verificar que el emisor es quien o lo que dice ser. La única información de identificación disponible en este nivel es la dirección origen del encabezado de paquete IP. La dirección origen se puede modificar fácilmente. Ni la capa de red ni la de transporte pueden verificar que los datos de la aplicación son correctos. Sin embargo, el nivel de paquete permite un control más preciso y sencillo sobre el acceso directo a un puerto, el contenido del paquete y los protocolos de comunicación correctos que se pueden establecer fácilmente o de forma adecuada, en niveles superiores.

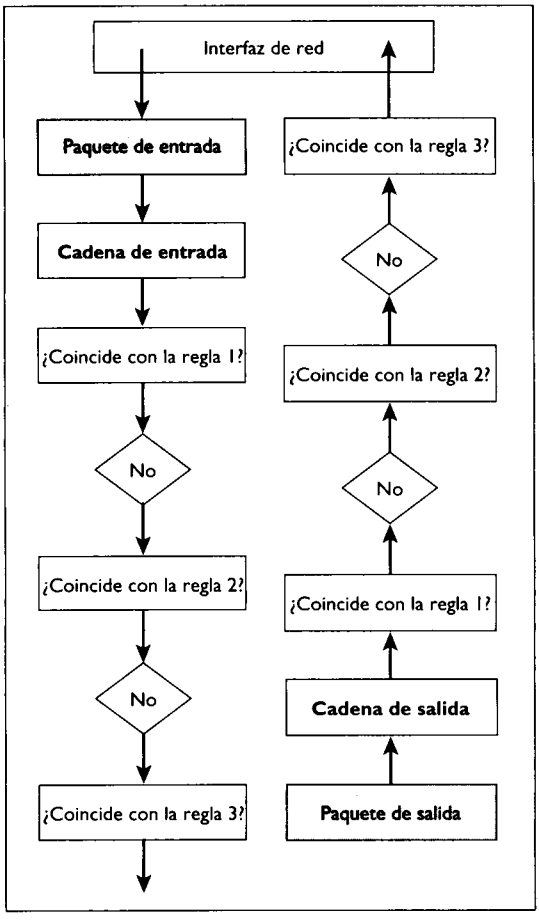

Figura 2.2. Cadenas de entrada y salida.

Sin filtrado a nivel de paquete, el filtrado de alto nivel y las medidas de seguridad proxy son inútiles o probablemente ineficaces. Hasta cierto punto,

deben basarse en la exactitud del protocolo de comunicación subyacente. Cada nivel de la pila del protocolo de seguridad agrega otra pieza que los demás niveles no pueden ofrecer fácilmente.

Elección de una directiva predeterminada de filtrado de paquetes

Cada cadena del firewall tiene una directiva predeterminada y una colección de acciones a realizar en respuesta a tipos de mensajes específicos. Cada paquete se compara, uno a uno, con cada regla de la lista hasta que se encuentra una coincidencia. Si el paquete no coincide con ninguna regla, fracasa y se aplica la directiva predeterminada al paquete.

Hay dos perspectivas básicas para un firewall:

- Denegar todo de forma predeterminada y permitir que pasen paquetes seleccionados de forma explícita.
- Aceptar todo de forma predeterminada y denegar que pasen paquetes seleccionados de forma explícita.

La directiva de denegar todo es la propuesta que se recomienda. Esta aproximación facilita la configuración de un firewall seguro, pero es necesario habilitar explícitamente cada servicio y la transacción de protocolo relacionada que quiera el usuario (véase la Figura 2.3). Esto significa que se debe comprender el protocolo de comunicación para cada servicio que se habilite. La propuesta denegar todo requiere preparar el terreno para habilitar el acceso de Internet. Algunos productos de firewall comerciales sólo son compatibles con la directiva denegar todo.

La directiva aceptar todo facilita mucho la configuración y la puesta en funcionamiento de un firewall, pero obliga a prever todo tipo de acceso imaginable que quiera deshabilitar (véase la Figura 2.4). El peligro es que no preverá un tipo de acceso peligroso hasta que sea demasiado tarde, o posteriormente habilitará un servicio no seguro sin bloquear primero el acceso externo al mismo. En definitiva, programar un firewall seguro para aceptar todo, implica más trabajo, mayor dificultad y, por tanto, es más propenso a errores.

Rechazar frente a denegar un paquete

El mecanismo del firewall IPFW ofrece la opción de rechazar o denegar los paquetes. ¿Cuál 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 deniega un paquete, simplemente se descarta el paquete sin ningún tipo de notificación al remitente.

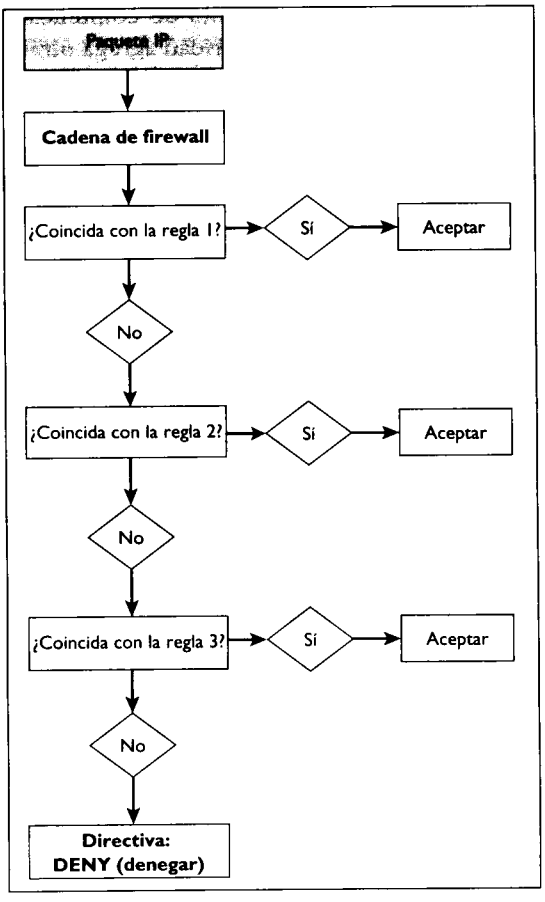

Figura 2.3. Directiva denegar todo de forma predeterminada.

La denegación es casi siempre la mejor elección. Hay tres razones para esto. Primero, enviar una respuesta de error duplica el trafico de red. La mayoría de los paquetes se descartan porque son malévolos, no porque representen un intento inocente de acceder a un servicio que no se le ha ocurrido ofrecer. Segundo, cualquier paquete al que responda se puede usar en un ataque por denegación de servicio. Tercero, cualquier respuesta, incluso un mensaje de error, ofrece información potencialmente útil a quien podría ser un hacker.

Cómo filtrar los paquetes entrantes

El lado de entrada del par E/S de la interfaz externa, la cadena de entrada, es el más interesante a la hora de asegurar un sitio. Como se mencionó anteriormente, se puede filtrar basándose en la dirección origen, la dirección destino, el puerto origen, el puerto destino y el indicador de estado TCP. Se usará toda esta información en un momento u otro de las siguientes secciones.

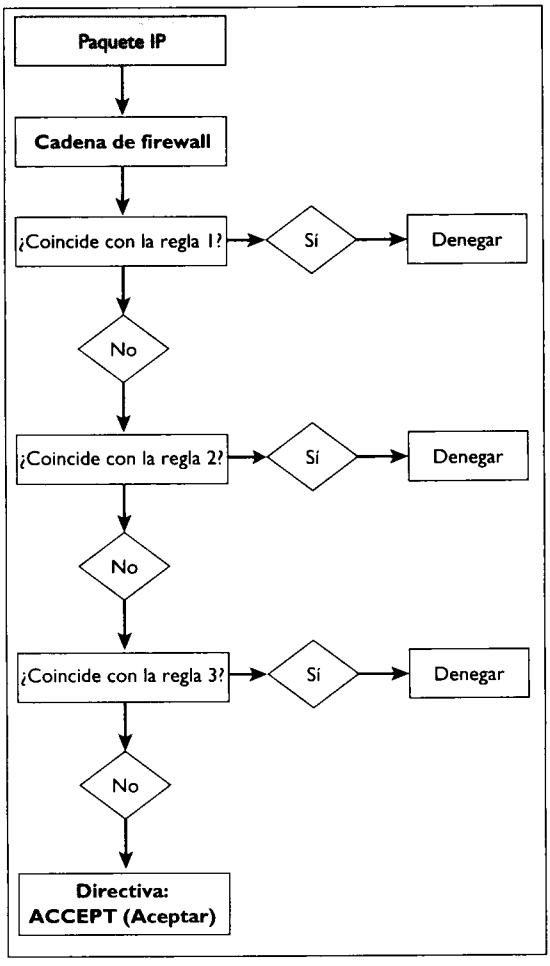

Figura 2.4. Directiva aceptar todo de forma predeterminada.

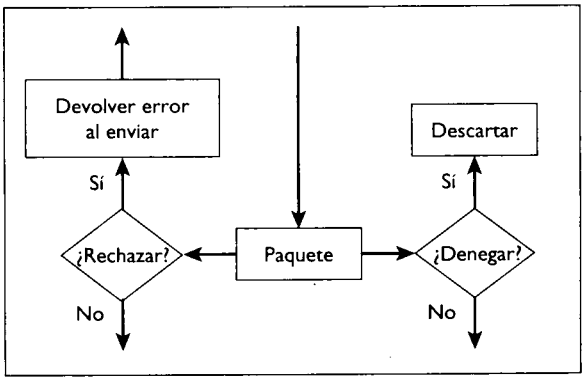

Figura 2.5. Cómo rechazar un paquete frente a denegarlo.

Filtrado de dirección origen remota

A nivel de paquete, el único medio de identificar el remitente del paquete IP es la dirección origen del encabezado del paquete. Este hecho abre la puerta al *spoofing*, o usurpamiento de dirección origen, donde el remitente coloca una dirección incorrecta, en vez de la suya propia, en el campo origen. La dirección puede ser una dirección inexistente o puede ser una dirección legítima perteneciente a otra persona. Esto puede permitir varias formas desagradables de romper el sistema y hacerse pasar por el usuario mientras atacan otros sitios, fingiendo ser otra persona cuando están atacando, o hacerle creer que es el origen de los mensajes entrantes.

Usurpamiento de dirección origen y direcciones ilegales

Hay seis clases principales de direcciones origen que siempre se deben denegar en la interfaz externa. Estas direcciones son las de paquetes entrantes que dicen ser una de las siguientes direcciones:

- Su dirección IP: Nunca se verán paquetes entrantes legales que indiquen proceder de su máquina. Como la dirección origen es la única información disponible, y ésta se puede modificar, es la única forma de usurpamiento que se puede detectar a nivel de filtrado de paquetes. Los paquetes entrantes que dicen proceder de su máquina pertenecen al usurpamiento de direcciones. No es posible saber con certeza si otros paquetes entrantes proceden de donde dicen proceder.
- Direcciones IP privadas de clase A, B y C: Un conjunto de direcciones en cada uno de los intervalos de las clases A, B y C son reservadas para su uso en las LAN privadas. No se pueden usar en Internet. Como tales, estas direcciones las puede usar cualquier sitio de forma interna sin necesidad de comprar direcciones IP registradas. Su máquina nunca debe ver paquetes entrantes procedentes de estas direcciones origen. En realidad, las direcciones origen privadas se suelen ver con cierta frecuencia en subredes ISP locales debido a sistemas mal configurados, al igual que desde sitios donde se usa el usurpamiento de forma intencionada. Si alguien filtra estas direcciones, se verán.
  - Las direcciones privadas de clase A se asignan al intervalo de direcciones de 10.0.0.0 a 10.255.255.255.
  - Las direcciones privadas de clase B se asignan al intervalo de direcciones de 172.16.0.0 a 172.31.255.255.
  - Las direcciones privadas de clase C se asignan al intervalo de direcciones de 192.168.0.0 a 192.168.255.255.
- Direcciones IP multidifusión de clase D, Las direcciones IP en el intervalo de la clase D se reservan para su uso como direcciones destino cuando se participa en una difusión en una red multidifusión, como una difusión de sonido o vídeo. El intervalo comienza en la dirección 224.0.0.0 y termina en 239.255.255.255. Su máquina nunca debería ver paquetes procedentes de estas direcciones origen.

- Direcciones reservadas de la clase E: Las direcciones IP en el intervalo de la clase E se han reservado para usos futuros y experimentales y no se asignan públicamente. El intervalo comienza en la dirección 240.0.0.0 y termina en 247.255.255.255. Su máquina nunca debería ver paquetes de estas direcciones origen y es muy probable que no lo haga. Las redes de defensa e inteligencia son lo bastante buenas como para no perder sus paquetes.
- Direcciones de interfaz de bucle invertido: La interfaz de bucle invertido es una interfaz de red privada que usa el sistema UNIX para servicios locales basados en red. En lugar de enviar el 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 forma de mejorar el rendimiento. Por definición, el tráfico de bucle invertido apunta al sistema que lo generó. No sale fuera de la red. El intervalo de las direcciones de bucle invertido es 127.0.0.0 a 127.255.255.255. Normalmente se verá como 127.0.0.1, localhost o la interfaz de bucle invertido lo.
- Direcciones de difusión mal formadas, 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 origen de difusión especial. La dirección origen de difusión será 0.0.0.0 o una dirección IP normal. Los clientes DHCP verán los paquetes de difusión entrantes procedentes de la dirección origen 0.0.0.0. Ignoro personalmente cualquier otra situación donde se puede ver la dirección de difusión 0.0.0.0. No es una dirección origen punto a punto legítima. Cuando se ve como la dirección origen en un paquete no de difusión regular, la dirección está falsificada.

Cómo bloquear sitios problemáticos

Otro esquema común de filtrado de dirección, pero que se usa menos, consiste en bloquear todos los accesos desde una máquina seleccionada, o de forma más normal, desde todo un bloque de direcciones IP de red. Así es como la comunidad de Internet suele tratar los sitios problemáticos y a los ISP que no vigilan a sus usuarios. Si un sitio se gana la reputación de ser un mal vecino de Internet, otros sitios tienden a bloquearlo sin excepciones.

A nivel individual, es conveniente bloquear todos los accesos desde redes seleccionadas cuando los individuos de la red remota no hacen más que ocasionar problemas.

Cómo limitar los paquetes entrantes a aquellos procedentes de los host remotos seleccionados

Puede que se quiera aceptar ciertas clases de paquetes entrantes procedentes sólo de sitios externos específicos o personas concretas. En estos casos, las reglas del firewall definirán direcciones IP específicas o un intervalo limitado de direcciones origen IP desde las que se aceptarán estos paquetes.

La primera clase de paquetes entrantes procede de servidores remotos que responden a las peticiones del usuario. Aunque algunos servicios, como los servicios Web o FTP, se puede esperar que procedan de cualquier lugar, otros servicios tendrán una procedencia legítima sólo si proceden del ISP o de los host seguros elegidos. Algunos ejemplos de servidores que es probable que sólo ofrezca el ISP son el servicio de correo POP, la asignación de dirección IP dinámica DHCP y, quizá, las respuestas del servidor de nombres de dominio (DNS, Domain Name Server).

La segunda clase de paquetes entrantes es la que procede de clientes remotos que acceden a los servicios que se ofrecen desde su sitio. De nuevo, aunque algunas conexiones de servicio entrante, como las conexiones al servidor web, se puede esperar que procedan de cualquier sitio, otros servicios locales sólo se ofrecerán a unos pocos usuarios o amigos de confianza. Algunos ejemplos de servicios locales restringidos pueden ser telnet, ssh y finger.

Filtrado de dirección destino local

El filtrado de paquetes entrantes basándose en la dirección destino no suele ser problemático. La tarjeta de interfaz de red ignora los paquetes habituales que no se dirigen a ella. La excepción son los paquetes de difusión, que se difunden a todos los *host* de la red.

La dirección 255.255.255.255 es la dirección destino de difusión general. Esta dirección se puede definir de forma más explícita como la dirección de red seguida por el número 255 en las restantes tuplas de dirección. Por ejemplo, si la dirección de red del ISP es 192.168.0.0 y su dirección IP es 192.168.10.30, se podrían ver paquetes de difusión dirigidos a 192.168.255.255 o a 255.255.255.255 procedentes del ISP.

La dirección de difusión a destino 0.0.0.0 se parece a la situación de los paquetes punto a punto que decían proceder de la dirección origen de difusión mencionada anteriormente en la sección "Usurpamiento de direcciones origen y direcciones ilegales". En este caso, los paquetes de difusión se dirigen a la dirección origen 0.0.0.0 en vez de a la dirección destino, 255.255.255.255. Por tanto, existe algo extraño sobre la intención del paquete. Esto es un intento de identificar el sistema como una máquina UNIX. Por razones históricas, el código de redes procedente del UNIX BSD devuelve un mensaje ICMP de error tipo 3 en respuesta a la utilización de la dirección 0.0.0.0 como la dirección destino de difusión. Otros sistemas operativos simplemente descartan el paquete. Como tal, este es un buen ejemplo de por qué es diferente denegar que rechazar un paquete. En este caso, el propio mensaje de error es lo que está buscando el hacker.

Filtrado de puerto origen remoto

El puerto origen en los paquetes entrantes identifica el programa del host remoto que envía el mensaje. En general, todas las peticiones entrantes des-

de clientes remotos a sus servicios siguen el mismo modelo, y todas las respuestas entrantes de servidores remotos al cliente local siguen un modelo diferente.

Las peticiones y las conexiones de entrada desde clientes remotos a los servidores locales tendrán un puerto origen del intervalo no privilegiado. Si se alberga un servidor web, todas las conexiones entrantes al servidor web deberán tener un puerto origen entre 1024 y 65535.

Las respuestas entrantes de los servidores remotos con los que se ha conectado tendrán el puerto origen asignado al servicio particular. Si se conecta a un sitio web remoto, todos los mensajes entrantes procedentes del servidor remoto tendrán el puerto origen establecido a 80, que es el número de puerto del servicio http.

Filtrado de puerto destino local

El puerto destino en los paquetes entrantes identifica el programa o el servicio del equipo al que se dirige el paquete. Como ocurre con el puerto origen, en general, todas las peticiones entrantes procedentes de clientes remotos a sus servicios siguen el mismo modelo, y todas las respuestas entrantes procedentes de servicios remotos a sus clientes locales siguen un modelo diferente.

Las peticiones entrantes y las conexiones entrantes procedentes de clientes remotos a los servidores locales establecerán el puerto destino al número de servicio asignado al servicio particular. Un paquete entrante dirigido a un servidor web local tendrá el puerto destino establecido a 80, que es el número del puerto del servicio http.

Las respuestas entrantes de los servidores remotos con los que se ha conectado tendrán el puerto destino en el intervalo no privilegiado. Si se conecta a un sitio web remoto, todos los mensajes entrantes deberán tener un puerto destino entre 1024 y 65535.

Filtrado del estado de la conexión TCP entrante

Las reglas de aceptación del paquete TCP entrante pueden hacer uso de los indicadores de estado de la conexión asociados con las conexiones TCP. Todas las conexiones TCP se adhieren al mismo conjunto de estados de conexión. Estos estados difieren entre cliente y servidor debido al saludo de tres vías que se realiza durante el establecimiento de la conexión.

Los paquetes TCP entrantes procedentes de clientes remotos tendrán el indicador SYN activado en el primer paquete recibido como parte del saludo de establecimiento de la conexión de tres vías. La primera petición de conexión tendrá el indicador SYN activado, pero no el indicador ACK. Todos los paquetes entrantes después de la primera petición de conexión tendrán sólo el indicador ACK activado. Las reglas del firewall del servidor local permitirán paquetes entrantes, sin tener en cuenta el estado de los indicadores SYN y ACK.

Los paquetes entrantes procedentes de servidores remotos siempre serán respuestas a la petición de conexión inicial que comienza en el programa cliente local. Cada paquete recibido desde un servidor remoto tendrá el indicador ACK activado. Las reglas del firewall cliente local solicitarán que todos los paquetes entrantes procedentes de servidores remotos tengan el indicador ACK activado. Los servidores legítimos no intentarán iniciar conexiones a programas cliente.

Sondeos y exploraciones

Un sondeo es un intento de conectar o de obtener una respuesta desde un puerto de servicio individual. Una exploración es una serie de sondeos a un conjunto de diferentes puertos de servicio. Las exploraciones suelen estar automatizadas.

En todo momento, los sondeos y las exploraciones son inofensivos. En Internet, la única forma de descubrir si un sitio ofrece un servicio concreto es sondear el puerto. ¿Cómo saber si un sitio tiene un servidor web si no se conoce su URL? Intente acceder a su sitio web. En otras palabras, debe sondear el puerto http.

Por desgracia, los sondeos y las exploraciones rara vez son inocentes. Es más probable que sean la fase inicial de recopilación de información, que busca debilidades interesantes antes de lanzar un ataque de un hacker. En 1998, en particular, se produjo un incremento exponencial de las exploraciones todo el mundo. Las herramientas de exploración automatizadas son de uso generalizado, y los esfuerzos coordinados por grupos de hackers son habituales.

Exploraciones de puerto generales

Las exploraciones de puerto generales son sondeos indiscriminados a lo largo de un bloque de puertos de servicio, probablemente todo el intervalo (véase la Figura 2.6). Estas exploraciones, posiblemente generadas por herramientas antiguas de mantenimiento de red, como satan o strobe, se están convirtiendo en menos frecuentes a medida que aparecen otras herramientas más sofisticadas y específicas como mscan, sscan y nscan.

Exploraciones de puerto dirigidas

Las exploraciones de puerto dirigidas buscan debilidades específicas (véase la Figura 2.7). Las herramientas más nuevas y sofisticadas intentan identificar el hardware, el sistema operativo y las versiones de software. Estas herramientas están diseñadas para anular por completo las debilidades conocidas de objetivos específicos.

Destinos comunes en los puertos de servicio

Los destinos comunes se suelen sondear y explorar de forma individual. El hacker puede estar buscando una debilidad específica, como un servidor de correo inseguro o un demonio portmap de RPC abierto.

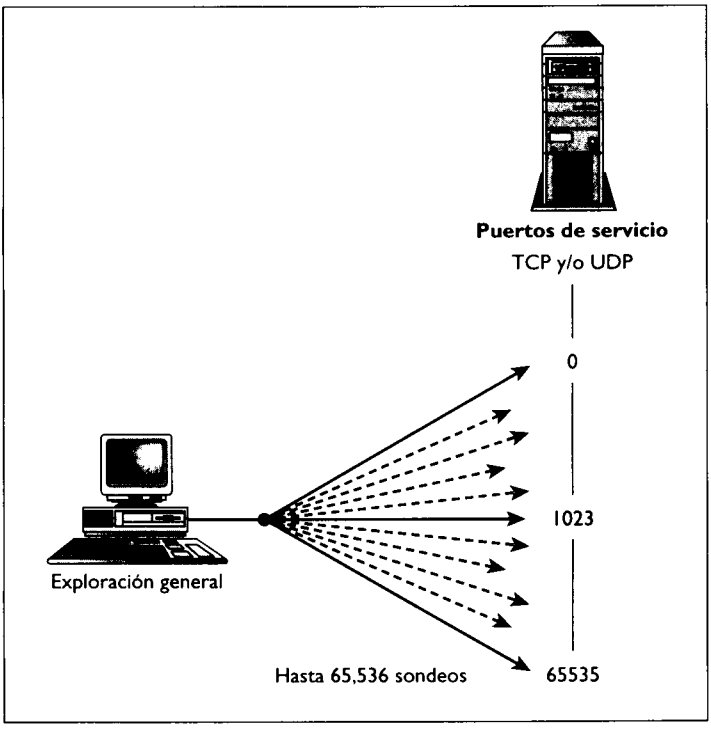

Figura 2.6. Exploración de puerto general.

En la sección "Interpretación de los registros del sistema" del Capítulo 6, "Cómo verificar que el sistema funciona como se espera", se muestra una extensa lista de puertos. Aquí, y para que tenga una idea general, mencionaremos tan sólo unos pocos puertos comunes:

- Los paquetes entrantes procedentes del puerto 0 reservado son siempre falsos. Este puerto no se usa de forma legítima.
- Los sondeos de puertos TCP del 0 al 5 son una firma del programa sscan.
- telnet (23/tcp), smtp (25/tcp), pop-3 (110/tcp), sunrpc (111/udp/tcp), imap (143/tcp), snmp (161/udp), route (520/udp) y mount (635/udp) son los puertos destino favoritos. Representan algunas de las debilidades más importantes de un sistema. Como estos servicios son tan comunes, representan buenos ejemplos de por qué no se quieren ofrecer al mundo exterior o de la necesidad de controlar con mucho cuidado el acceso exterior a estos servicios.
- Los sondeos NetBIOS (137, 138/tcp/udp, 139/tcp), Netbus (12345/tcp) y Back Orifice (31337/udp) son muy comunes. No suponen una amenaza para un sistema UNIX. En este caso, el destino es un sistema Windows.

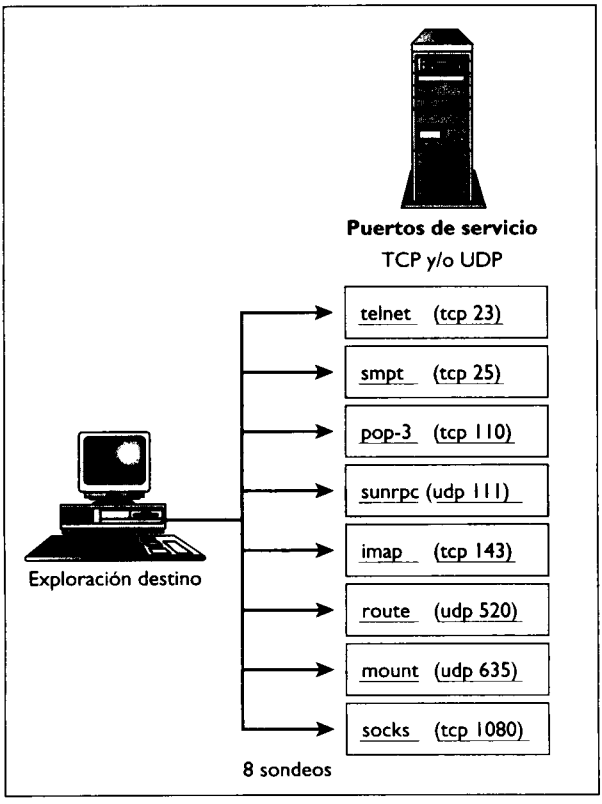

Figura 2.7. Exploración de puerto dirigida.

Cómo evitar la paranoia: responder a las exploraciones de puerto

Prácticamente todos los días, los registros del firewall, que se encuentran en /var/log/messages, muestran todas las clases de intentos de conexión fallidos.

¿Intentan las personas acceder de forma ilegal al sistema con esta regularidad? Sí lo hacen. ¿Está el sistema comprometido? No, no lo está. Los puertos están bloqueados. El firewall está realizando su trabajo. Son intentos de conexión fallidos que el firewall denegó.

¿En qué momento decide informar personalmente de ello? ¿En que momento es lo suficientemente importante dedicar algo de tiempo para informar de ello? ¿En qué momento decir basta y seguir adelante normalmente, o debería escribir a la dirección abuse@algun.sistema cada vez que sucede esto?

No hay respuestas "correctas". La forma de responder es una llamada a su juicio personal. Es uno de esos temas que puede provocar discusiones bizantinas. Para sondeos y exploraciones claras, no existe una respuesta bien definida. Dependerá de su propia personalidad, de su nivel de comodidad, de la forma en que defina un sondeo serio y de su conciencia social.

Teniendo esto presente, éstas son algunas pautas a seguir.

Los intentos más comunes son una combinación de sondeos automatizados, errores, intentos legítimos basados en la historia de Internet, en la ignorancia, en la curiosidad y el software que se comporta de forma errónea.

Deben ignorarse los intentos de conexión individuales, restringidos a telnet, ssh, ftp, finger o a cualquier otro puerto para un servicio común que no se ofrezca. Los sondeos y las exploraciones son un hecho del día a día en Internet, todos muy frecuentes y que no suelen suponer ningún riesgo. Son una clase de vendedores que van de puerta en puerta, de llamadas telefónicas comerciales, de números de teléfono equivocados y de la propaganda en el correo. No hay suficiente tiempo en un día para responder a cada uno de ellos.

Por otro lado, algunas personas que se dedican a sondear son más persistentes. Puede que sea necesario agregar reglas de firewall para bloquearlos completamente, o quizá incluso bloquear todo el espacio de direcciones de dominio si el dominio tiene una mala reputación o si usan múltiples direcciones origen.

Las exploraciones de un subconjunto de los puertos conocidos como posibles agujeros de seguridad son normalmente los precursores de un intento de piratería si se encuentra un puerto abierto. Las exploraciones más globales suelen ser parte de una exploración más amplia en busca de aberturas en un dominio o de una subred. Las herramientas de piratería actuales sondean, uno tras otro, un subconjunto de estos puertos.

En algunas ocasiones, aparecerán intentos serios de piratería. Este es, sin ninguna duda, el momento de realizar las acciones. Escríbales. Denúncielos. Duplique la comprobación de la seguridad. Observe lo que hacen. Bloquéelos. Bloquee su bloque de direcciones IP. Ponga su sistema fuera de línea, si es necesario.

Puertos históricamente peligrosos

Para obtener más información acerca de puertos históricamente peligrosos, consulte el documento *"Packet Filtering for Firewall Systems"* (Filtrado de paquetes para sistemas de firewall), que puede encontrar en www.cert.org.

Algunos administradores de sistemas consideran serias todas las apariciones, porque incluso si una máquina es segura, las máquinas de otras personas pueden no serlo. La siguiente persona puede que ni siquiera tenga la capacidad de saber que está siendo sondeado. Crear informes de sondeos es una responsabilidad social que, por el bien de todos, hay que tomar.

¿Cómo debería responder a las exploraciones de puerto? Si escribe a estas personas, al administrador, al NOC proveedor de servicio de vínculo ascendente al coordinador de bloques de direcciones de red, intente ser educado. Concédales el beneficio de la duda. Suelen ser muy frecuentes las reacciones exageradas. Lo que puede parecer un intento de piratería serio para el usuario es a menudo un niño curioso jugando con un nuevo programa. Unas palabras educadas al abuse, root o postmaster normalmente solucionarán el problema. Es necesario enseñar a las personas *Netiquette* (buenos modales en Internet)

en lugar de rescindirles las cuentas. Y, pueden ser inocentes de todo. Con mucha frecuencia, el sistema de la persona está comprometido y no tienen ni idea de lo que sucede. Estarán agradecidos por la información.

Ataques por denegación de servicio

Los ataques por denegación de servicio se basan en la idea de inundar un sistema con paquetes de forma que afecte o degrade seriamente la conexión de Internet, inmovilizando los servidores locales hasta el extremo de no poder atender las peticiones legítimas, o en el peor de los casos, rompiendo el sistema totalmente. Los dos resultados más comunes son mantener al sistema demasiado ocupado para hacer nada útil e inmovilizar los recursos críticos del sistema.

No es posible protegerse completamente contra los ataques por denegación de servicio. Toman tantas formas diferentes como permite la imaginación del hacker. Cualquier cosa que produzca una respuesta del sistema, cualquier cosa que produzca peticiones de recursos en el sistema, cualquier cosa que induzca a un sitio remoto a dejar de comunicarse con el usuario, todo se puede usar en un ataque por denegación de servicio.

Sin embargo, estos ataques suelen implicar alguno de los diferentes modelos clásicos, incluyendo inundación SYN TCP, inundación ping, inundación UDP y bombas de redirección de enrutamiento ICMP.

Ataques por denegación de servicio

Para obtener más información acerca de ataques por denegación de servicio, consulte el documento "Denial of Service" (Denegación de servicio), que puede encontrar en www.cert.org.

Inundación SYN TCP

Un ataque de inundación SYN TCP consume los recursos del sistema hasta que no es posible establecer más conexiones TCP entrantes. El ataque hace uso del protocolo de saludo de tres vías TCP durante el establecimiento de la conexión, junto con usurpamiento de la dirección origen IP.

El atacante usurpa la dirección origen e inicia una conexión a uno de los servicios basados en TCP. Como si fuera un cliente que intenta una conexión TCP, el atacante envía un mensaje SYN. Su máquina responde enviando una confirmación, un SYN-ACK. Sin embargo, en este caso, la dirección a la que contesta el usuario no es la dirección del atacante. La etapa final del establecimiento de conexión TCP, que consiste en recibir un ACK como respuesta, nunca tendrá lugar. En consecuencia, se consumen los recursos finitos de conexión de red. La conexión permanece en un estado semiabierto hasta que la conexión alcanza su tiempo de espera. El hacker inunda su puerto con una petición de conexión tras otra, más rápido que los tiempos de espera TCP liberan los recursos. Si esto continúa, todos los recursos estarán en uso y no

se podrán aceptar más peticiones de conexión entrantes. Si el objetivo es el puerto smtp, no se podrá recibir correo electrónico. Si el objetivo es el puerto http, las personas no podrán conectar con su sitio web.

Existen varias ayudas para los usuarios de Linux. La primera es el filtrado de direcciones origen descrito anteriormente. Éste filtra las direcciones origen usurpadas que más se suelen usar, pero no hay garantía de que la dirección usurpada pertenezca a las categorías que se pueden anticipar y filtrar. La segunda consiste en compilar el núcleo de Linux con las cookies SYN habilitadas; este es un retardo específico para la inundación SYN. Las cookies SYN están habilitadas de forma predeterminada en Red Hat 6.0. No es necesario hacer nada. Las primeras versiones necesitan que se configure explícitamente la opción en el núcleo usando make config, make menuconfig o make xconfig, y luego volver a compilar e instalar el núcleo.

Inundación ping

Cualquier mensaje que provoque una respuesta de la máquina se puede usar para degradar la conexión de red obligando al sistema a gastar la mayor parte del tiempo respondiendo. El mensaje de petición de eco ICMP que se ha enviado mediante ping suele ser un posible culpable.

Además, un viejo programa de piratería llamado *Ping de la muerte* se dedicaba a enviar grandes paquetes de ping. Como resultado, los sistemas vulnerables podrían dejar de funcionar. Linux no es vulnerable a este programa, como tampoco lo son otros sistemas operativos UNIX actuales. Si un firewall protege sistemas viejos, estos sistęmas sí pueden ser vulnerables.

La proeza de *Ping de la muerte* da una idea de cómo puede usar el hacker creativo los protocolos más simples y las interacciones de los mensajes. No todos los intentos de piratería son intentos de introducirse en un equipo, algunos son simplemente destructivos. En este caso, el objetivo es bloquear la máquina.

ping es una herramienta básica de redes muy útil. Puede que no sea necesario deshabilitar ping completamente. En el entorno de Internet actual, la gente conservadora recomienda deshabilitar el ping entrante, o al menos limitar de forma severa de quién se pueden aceptar peticiones de eco. Como la historia del ping está involucrada en ataques por denegación de servicio, muchos sitios ya no responden a peticiones ping externas.

Inundación SYN y usurpamiento de dirección IP

Para obtener más información acerca de la inundación SYN y del usurpamiento de direcciones IP, consulte el CERT_Advisory_CA-96.21, "TCP SYN Flooding and IP Spoofing Attacks" *(Inundación SYN TCP y ataques de usurpamiento de direcciones IP)* en ftp://info.cert.org/pub/cert_advisories/CA-96.21.tcp_SYN_flooding.

Inundación UDP

El protocolo UDP es especialmente útil como herramienta de denegación de servicio. Al contrario que TCP, UDP es sin estado. No se incluyen meca-

nismos de control de flujo. No hay indicadores de estado de conexión. No se usan los números de secuencia del datagrama. No se mantiene información sobre el paquete que se espera a continuación. Es relativamente fácil mantener a un sistema tan ocupado respondiendo a sondeos UDP entrantes que no quede ancho de banda para el tráfico de red legítimo.

Como los servicios UDP son inherentemente menos seguros que los servicios TCP, muchos sitios deshabilitan todos los puertos UDP que no son absolutamente necesarios. Como se mencionó anteriormente, casi todos los servicios de Internet habituales se basan en TCP. El firewall que construiremos en el Capítulo 3, "Creación e instalación de un firewall", restringe con cuidado el tráfico UDP sólo a aquellos host que proporcionan servicios UDP necesarios.

Bombas de redirección ICMP

El mensaje de redirección de ICMP tipo 5 indica al sistema destino que cambie sus tablas de enrutamiento por una ruta más corta. Si se ejecuta routed o gated y puede redireccionar mensajes, es posible para un hacker engañar al sistema para que piense que la máquina del hacker es una de las máquinas locales o una de las máquinas del ISP, o incluso engañar al sistema para que lance todo el tráfico a algún otro host remoto.

Ataques por denegación de servicio y otros recursos del sistema

La conectividad de red no es la única preocupación en los ataques por denegación de servicio. A continuación se muestran algunos ejemplos de otras áreas que se deben tener en cuenta a la hora de configurar el sistema:

- El sistema de archivos puede desbordarse si se obliga al sistema a escribir gran cantidad de mensajes en los registros de errores, o si se inunda el sistema con muchas copias de mensajes de correo electrónico grandes. Puede que sea necesario configurar los límites de los recursos y una partición independiente para sistemas de archivos que crezcan o cambien rápidamente.
- Los servidores pueden bloquearse si se envía gran cantidad de datos y se inundan los búfer de entrada, o si se envían datos inesperados. Los guiones CGI son especialmente vulnerables a menos que se tomen precauciones. Muchas de las debilidades actuales en los servidores se deben a desbordamientos de búfer. Es importante mantener actualizados e instalar todos los parches y revisiones de software más recientes.
- La memoria del sistema, las ranuras de la tabla de procesos, los ciclos de CPU y otros recursos pueden agotarse debido a repetidas y rápidas invocaciones de servicios de red. Poco se puede hacer aparte de establecer cualquier límite que se pueda configurar para cada servicio individual, habilitando cookies SYN y denegar en vez de rechazar los paquetes enviados a puertos de servicio no compatibles.

Ataques por denegación de servicio del puerto UDP

Para obtener más información acerca de explosiones por denegación de servicio usando estos servicios UDP, consulte el CERT Advisory CA-96.01 "UDP Port Denial-of-Service Attack" *(Ataque por denegación de servicio del puerto UDP)*, que puede encontrar en www.cert.org.

Explosiones por denegación de servicio de correo electrónico

Para obtener más información acerca de explosiones por denegación de servicio usando correo electrónico, consulte "Email Bombing and Spamming" (Bombas y correo masivo de correo electrónico), que se puede encontrar en www.cert.org.

Consideraciones varias sobre el filtrado de paquetes entrantes

El enrutamiento origen y la fragmentación no son cuestiones del firewall de filtrado de paquetes en la implementación Linux del IPFW, pero son cuestiones de seguridad relacionadas con los paquetes. Ambos se tratan a nivel del sistema operativo.

Paquetes enrutados en origen

Los paquetes enrutados en origen emplean una opción IP que se usa en raras ocasiones y que permite al creador definir la ruta entre dos máquinas, en vez de dejar que los enrutadores intermedios determinen la trayectoria. Al igual que con las redirecciones ICMP, esta característica puede permitir a un hacker engañar a su sistema para que piense que está hablando con una máquina local, una máquina ISP o con algún otro host seguro.

La versión Linux 6.0 de Red Hat descarta, de forma predeterminada, los paquetes enrutados en origen. Las versiones más antiguas deben volverse a compilar para denegar los paquetes enrutados en origen. Se recomienda encarecidamente deshabilitar esta característica en la configuración de compilación del núcleo. El enrutamiento en origen tiene pocos usos legítimos. Algunos enrutadores incluso ignoran esta opción.

Fragmentación del paquete

Las diferentes redes subyacentes (por ejemplo, Ethernet, ATM o Token Ring) definen diferentes límites en el tamaño de una trama. A medida que se pasa un paquete de un enrutador al siguiente a lo largo de la trayectoria desde la máquina origen hasta la máquina destino, los enrutadores de pasarela de red pueden tener que dividir el paquete en piezas más pequeñas, llamadas *fragmentos*, antes de pasarlos a una nueva red. El primer fragmento contiene los números usuales de puerto origen y puerto destino. Los siguientes fragmentos no los contienen.

Las máquinas de firewall y las máquinas que hacen enmascaramiento IP para otros host locales deben configurarse para que puedan volver a ensamblar los paquetes antes de entregarlos al destino local. Esta característica está habilitada automáticamente en Red Hat 6.0. Las versiones anteriores requieren que se compile explícitamente la característica de desfragmentación en el núcleo.

Explosiones por denegación de servicio CGI

Para obtener más información acerca de explosiones por denegación de servicio usando guiones CGI, consulte *"How to Remove Meta-Characters in CGI Scripts"* (Cómo quitar metacaracteres en guiones CGI), que puede encontrar en www.cert.org y en *"The World Wide Web Security FAQ"* en www.w3.org.

Cómo filtrar paquetes salientes

Si el entorno representa un entorno seguro, filtrar los paquetes salientes no es tan importante como filtrar los paquetes entrantes. El sistema no responderá a mensajes entrantes si no pasan a través del firewall. Aun así, el filtrado simétrico es más seguro. Este filtrado también protege a otras personas y al usuario, de posibles errores existentes en la máquina.

¿Qué problemas pueden aparecer en una máquina? En el peor de los casos, un hacker puede tener éxito y conseguir una cuenta en el sistema. Filtrar los paquetes salientes proporciona algo más de protección, al menos hasta que el hacker consiga acceso de superusuario o root y averigüe cómo deshabilitar el firewall.

Filtrar los paquetes salientes también permite ejecutar servicios LAN sin perder paquetes locales en Internet, a donde no pertenecen estos paquetes. No es sólo una cuestión de no permitir el acceso externo a los servicios LAN, es también una cuestión de no difundir información del sistema local en Internet. Ejemplos de esto serían si estuviese ejecutando un servidor local dhcpd, timed, routed o rwhod para uso interno.

Otra posibilidad es bloquear los daños originados por las máquinas. Hace un año, estaba realizando una aproximación algo caballeresca a los filtros salientes en una discusión de seguridad en Usenet. Alguien me escribió para fastidiarme diciéndome que obviamente yo no tenía hijos adolescentes...

Una última fuente de problemas que suele aparecer en la red de un ISP son las personas que ejecutan software de prueba o experimental. Se han traído un programa a casa del trabajo para probarlo o para trabajar con él, y el software no funciona correctamente.

Un origen relacionado es el viejo software de los equipos personales, que a veces ignora los protocolos de puerto de servicio de Internet y las asignaciones reservadas. Este es el equivalente en un equipo personal a ejecutar un programa diseñado para usarlo en una LAN sobre una máquina conectada a Internet.

Filtrado de dirección origen local

Filtrar los paquetes salientes basándose en la dirección origen es fácil. Para un pequeño sitio o un equipo independiente conectado a Internet, la dirección origen es siempre la dirección IP del equipo del usuario cuando funciona normalmente. No hay razón para permitir que un paquete saliente tenga otra dirección origen.

Para las personas cuya dirección IP la asigna de forma dinámica el ISP mediante DHCP, existe una única excepción durante la asignación de la dirección. Esta excepción es específica de DHCP y se explica en el Capítulo 3.

Para las personas con una LAN y múltiples máquinas de servidor públicas, cada una con su propia dirección IP asignada de forma estática, el tema no está tan claro. Los temas relacionados con las LAN se explican en el Capítulo 4, "Aspectos relacionados con las LAN, múltiples firewalls y redes de perímetro". Para las personas con una LAN cuya máquina firewall tenga una dirección IP asignada de forma dinámica, es obligatorio restringir los paquetes salientes a los que tengan la dirección origen de la dirección origen IP de la máquina firewall. Esto le protege de varios errores bastante comunes de configuración, que se explican en el Capítulo 4, y que aparecen como casos de usurpamiento o direcciones origen ilegales a host remotos.

Filtrado de dirección destino remota

Al igual que con los paquetes entrantes, puede que quiera permitir que ciertas clases de paquetes salientes sólo se dirijan a redes remotas o máquinas individuales específicas. En estos casos, las reglas del firewall definirán direcciones IP concretas o un intervalo restringido de direcciones IP destino donde se permitirán estos paquetes.

La primera clase de paquetes salientes a filtrar mediante la dirección destino son los paquetes destinados a servidores remotos con los que se ha contactado. Aunque puede esperar que algunos paquetes, como los destinados a servidores web o FTP, se dirijan a cualquier lugar de Internet, otros servicios remotos sólo los ofrecerán de forma legítima el ISP o a hosts seguros elegidos de forma concreta. Algunos de los servidores que probablemente sólo los ofrecerá el ISP son el servicio de correo POP, la asignación de direcciones IP dinámica DHCP y el servicio de noticias Usenet.

La segunda clase de paquetes salientes que se deben filtrar por la dirección destino son los paquetes destinados a clientes remotos que acceden a un servicio que se ofrece desde el sitio del usuario. De nuevo, aunque algunas conexiones de servicio salientes pueden ir a cualquier lugar, como las respuestas desde su servidor web local, otros servicios locales sólo se ofrecerán a unos pocos sitios remotos seguros o amigos. Algunos ejemplos de servicios locales restringidos pueden ser telnet, ssh y finger. No sólo las reglas de firewall denegarán las conexiones entrantes generales a estos servicios, sino que tampoco permitirán respuestas salientes de estos servicios a nadie.

Filtrado de puerto origen local

Definir explícitamente los puertos de servicio que se pueden usar en su extremo para conexiones salientes tiene dos propósitos, el primero para los programas cliente y el segundo para los programas servidor. Especificar los puertos origen que se permiten para las conexiones salientes ayuda a asegurar que los programas se comportan correctamente, y protege a otras personas de cualquier tráfico de red local que no pertenezca a Internet.

Las conexiones salientes desde los clientes locales casi siempre se originan desde un puerto origen no privilegiado. Restringir los clientes a los puertos no privilegiados en las reglas de firewall ayuda a proteger a otras personas de posibles errores en su extremo, asegurando que los programas cliente se comportan como se espera.

Los paquetes salientes desde los programas servidor locales se originarán siempre desde el puerto de servicio asignado. Si se restringen los servidores a los puertos asignados en el nivel de firewall, se asegura que los programas de servidor funcionan correctamente a nivel de protocolo. Lo más importante es que todo esto sirve de ayuda para proteger todos los servicios privados de red local que se estén ejecutando desde un acceso exterior. También ayuda a proteger los sitios remotos, para que el tráfico de red que debería permanecer confinado a los sistemas locales, no ocasione molestias.

Filtrado de puerto destino remoto

Los programas cliente locales están diseñados para conectarse a servidores de red que ofrecen sus servicios desde los puertos de servicio asignados. Desde esta perspectiva, restringir los clientes locales para que sólo se puedan conectar a los puertos de servicio asociados asegura la exactitud del protocolo. Restringir las conexiones de los clientes a puertos destino específicos también sirve para un par de propósitos. En primer lugar, ayuda a vigilar contra los programas de red privados y locales que de forma inadvertida intentan acceder a servidores en Internet. Segundo, hace todo lo posible para no permitir los errores salientes, las exploraciones de puerto y otras acciones incorrectas que se originan en el sitio.

Los programas servidor locales casi siempre participarán en las conexiones que se originan desde puertos no privilegiados. Las reglas de firewall restringen el tráfico saliente de servidores sólo a puertos destino no privilegiados.

Filtrado saliente de estado de la conexión TCP

Las reglas de aceptación de paquetes TCP salientes pueden hacer uso de los indicadores de estado de la conexión asociados con las conexiones TCP, igual que lo hacen las reglas de entrada. Todas las conexiones TCP se adhieren al mismo conjunto de estados de conexión, que es diferente para el cliente y el servidor.

Los paquetes TCP salientes procedentes de clientes locales tendrán el indicador SYN definido en el primer paquete que se envíe como parte del saludo de establecimiento de conexión de tres vías. La petición de conexión inicial tendrá definido el indicador SYN, pero no el indicador ACK. Todos los paquetes salientes posteriores a la primera petición de conexión sólo tendrán definido el indicador ACK. Las reglas de firewall de cliente local permitirán los paquetes salientes con el indicador SYN o el indicador ACK activado.

Los paquetes salientes procedentes de servidores locales serán siempre respuestas a una petición inicial de conexión iniciada desde un programa cliente remoto. Cada paquete que se envía desde sus servidores tendrá activo el indicador ACK. Las reglas de firewall de servidor local solicitarán que todos los paquetes salientes de los servidores tengan activo el indicador ACK.

Servicios de red privados frente a públicos

Una de las formas más sencillas de permitir el acceso de personas no invitadas de forma inadvertida es permitir el acceso exterior a servicios locales que están diseñados sólo para que se usen en una LAN. Algunos servicios, si se ofrecen localmente, nunca deberían cruzar la frontera entre la LAN e Internet. Algunos de estos servicios molestan a los vecinos, algunos proporcionan información que sería mejor guardarla para usted y otros representan grandes agujeros de seguridad si están disponibles fuera de la LAN.

Algunos de los primeros servicios de red, los comandos de acceso remoto BSD en particular, se diseñaron para compartirse de forma local y facilitar el acceso a través de múltiples máquinas de laboratorio en un entorno seguro. Otros servicios se han destinado al acceso a Internet, pero se diseñaron cuando Internet era básicamente una amplia comunidad de académicos e investigadores. Internet era un sitio seguro y relativamente abierto. Conforme Internet fue derivando hacia una red global que permitía el acceso al público en general, se fue convirtiendo en un entorno completamente inseguro.

Muchos servicios de red de UNIX están diseñados para ofrecer información sobre cuentas de usuario en el sistema, qué programas se están ejecutando y qué recursos se utilizan, el estado del sistema, el estado de la red e información parecida procedente de otras máquinas conectadas a la red. No todos estos servicios de información representan agujeros de seguridad por sí mismos. No es que un hacker pueda usarlos directamente para tener acceso no autorizado al sistema. La cuestión es que ofrecen información sobre el sistema y las cuentas de usuario, que pueden ser útiles para un hacker que busca puntos débiles. También pueden ofrecer información, como los nombres de usuario, direcciones, números de teléfono, etc., que no sería nada recomendable que estubieran a disposición de cualquiera que las pidiera.

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

Algunos servicios son difíciles de configurar correctamente y otros son difíciles de configurar de forma segura. Se han escrito libros enteros sobre cómo configurar algunos de los servicios UNIX más complicados. La configuración específica de servicios queda fuera del objetivo de este libro.

Otros servicios no tienen sentido en una configuración particular o de pequeña empresa. Unos están dirigidos a administrar grandes redes, proporcionar servicio de enrutamiento de Internet, servicios de información de grandes bases de datos, soportar cifrado y autenticación bidirecionales, etc.

Cómo proteger servicios locales no seguros

La forma más sencilla de protegerse a sí mismo es no ofrecer el servicio. Pero, ¿qué sucede si necesita alguno de estos servicios de forma local? No todos los servicios se pueden proteger de forma adecuada en el nivel de filtrado de paquetes. Los servicios multiconexión, como RealAudio e ICQ, y los servicios RPC basados en UDP son bastante difíciles de asegurar a nivel de filtrado de paquetes.

Una forma de mantener seguro el equipo es no albergar servicios de red en la máquina firewall que no quiere que use el público. Si el servicio no está disponible, no existe forma posible de que un cliente remoto se conecte a él. Los sitios pequeños, como los personales, es probable que no tengan un conjunto adicional de equipos disponibles para hacer cumplir las directivas de seguridad de acceso ejecutando servicios privados en otras máquinas. Es necesario tomar decisiones.

Otra forma de salvaguardar el equipo consiste en implementar un firewall de algún tipo. Un firewall de filtrado de paquetes agrega protección a nivel de acceso al puerto de servicio. En concreto, esto permite ejecutar muchos de los servicios de red locales e inseguros, particularmente servicios TCP, con menos peligro de que alguien acceda desde el exterior. Además, esto puede ayudar a controlar el tráfico externo que puede acceder a servicios locales, y haciendo cumplir los protocolos de comunicación de bajo nivel, pueden ayudar a asegurar que los programas sólo hablan lo que se espera y con quien se espera.

Sin embargo, un firewall de filtrado de paquetes no ofrece seguridad completa. Algunos programas requieren medidas de seguridad de más alto nivel que pueden proporcionarse a nivel de filtrado de paquetes. Algunos programas son demasiado problemáticos para arriesgarse a ejecutarlos sobre una máquina firewall.

Cómo seleccionar los servicios que se desean ejecutar

Cuando todo está dicho y hecho, sólo puede decidir los servicios que necesita o quiere. El primer paso a la hora de asegurar un sistema es decidir qué servicios y demonios desea ejecutar sobre la máquina firewall. Cada servicio tiene sus propias consideraciones de seguridad. Cuando se seleccionan los

servicios a ejecutar bajo UNIX, la regla general es ejecutar sólo servicios de red que necesite y comprenda. Es importante entender un servicio de red, qué hace y a quién está dirigido, antes de ejecutarlo, especialmente en una máquina conectada directamente a Internet.

Las siguientes secciones listan los servicios de red más comunes que se encuentran disponibles en un sistema Linux de Red Hat. Los servicios están ordenados según cómo y cuándo se inician. Dicho de otra forma, los servicios se ordenan por el mecanismo que usa UNIX para iniciarlos, que también determina cómo están organizados estos servicios desde el punto de vista de la administración del sistema.

Los servicios de red se inician desde tres sitios principales en un sistema UNIX. El primero, los servicios básicos, los administra el administrador de nivel de ejecución y se inician automáticamente cuando se arranca el sistema desde secuencias de comandos del shell que se encuentran en el directorio /etc/rc.d. Segundo, algunos de los servicios menos importantes de red se administran mediante inetd y se inician sobre peticiones desde un programa cliente. Estos servicios se definen en el archivo de configuración de inetd, /etc/inetd.conf. El tercero, algunos servicios de red son locales a su sitio específico y deben iniciarse explícitamente a través de las secuencias de comandos de configuración local. Estos servicios no suelen ser parte de la distribución Linux estándar. Por el contrario, suelen ser programas descargados desde algún almacén de archivos e instalados por el propio usuario.

Administrador de nivel de ejecución

Un *administrador de nivel de ejecución* es un concepto de inicio y estado del sistema tomado de UNIX System V. Red Hat Linux incorpora siete niveles de ejecución diferentes. Los subdirectorios que hay por debajo de /etc/rc.d están asociados con los diferentes niveles de ejecución y contienen vínculos simbólicos a secuencias de comandos de configuración en /etc/rc.d/init.d. El vínculo simbólico sigue una convención de nombres que indica si un servicio debe detenerse (los nombres que empiezan con K) o iniciarse (los nombres que empiezan por S) cuando se entra a un nivel de ejecución, y el orden en que deben ejecutarse las secuencias de comandos.

Nunca se usarán los siete niveles de ejecución. El sistema del usuario funcionará en uno de los niveles de ejecución 2, 3 ó 5. El nivel de ejecución predeterminado es el 3, normal, estado de sistema multiusuario. El nivel de ejecución 2 es igual que el 3, pero sin disponibilidad de los servicios del sistema de archivos de red (NFS). El nivel de ejecución 5 es lo mismo que el 3, con la adición del X Window Display Manager, que muestra un inicio de sesión basado en X Window y pantallas de selección de host (idealmente, una máquina firewall no debería ejecutar el X Window Display Manager).

Para los curiosos, los restantes niveles de ejecución representan estados especiales del sistema. El nivel de ejecución 0 define las acciones de limpieza final que se deben realizar antes de detener el sistema. El nivel de ejecu-

ción 1 define las acciones a realizar cuando se entra y se sale del modo monousuario. El nivel de ejecución 4 no se usa. Se puede configurar un estado personalizado. El nivel de ejecución 6 define las acciones de limpieza finales a realizar antes de reiniciar el sistema.

Cuando instale un sistema por primera vez y, posteriormente, cuando esté usando el administrador de nivel de ejecución al que se puede tener acceso a través del programa control-panel, se podrán definir los servicios de sistema que se iniciarán automáticamente cuando el sistema arranca. Ambas herramientas de configuración proporcionan interfaces gráficas para administrar los archivos de inicio del sistema en /etc/rc.d.

Para saber algo sobre el tema, cuando el sistema se inicia, el proceso de sistema maestro, init, lee el archivo de configuración, /etc/inittab. Este archivo indica a init que ejecute varias secuencias de comandos en el directorio /etc/rc.d, incluyendo rc.sysinit y rc. /etc/rc.d/rc.sysinit que se ejecutan una vez en tiempo de inicio para inicializar el sistema, cargar módulos, comprobar los sistemas de archivo, leer valores de configuración de todo el sistema, etc. /etc/rc.d/rc se ejecuta cada vez que se cambia el nivel de ejecución. Su trabajo consiste en detener los servicios que no están definidos para ejecutarse en este nivel de ejecución, y luego lanzar los servicios que *están* definidos para ejecutarse en este nivel de ejecución particular (nivel de ejecución 3 ó 5, normalmente). El nivel de ejecución predeterminado se define en el archivo /etc/inittab. Está preconfigurado como id:3:initdefault. El nivel de ejecución 2 es la mejor elección para una máquina firewall.

Cuando se inicia el editor de nivel de ejecución, aparece una ventana dividida en nueve listas de nombres ordenadas por orden alfabético, que son los nombres de las secuencias de comandos de inicialización de los servicios del sistema. La lista situada más a la izquierda se titula "Available" (Disponible) y contiene los nombres de las secuencias de comandos disponibles en /etc/rc.d/init.d. El resto de la ventana se divide en dos columnas, etiquetadas "Start" (Inicio) y "Stop" (Parada). Las dos filas se dividen en cuatro columnas etiquetadas de 2 a 5, que hacen referencia a servicios que se detienen o inician al entrar a cada uno de los niveles de ejecución 2 a 5. La Tabla 2.1 lista las secuencias de comandos de inicialización de servicio disponibles en la columna situada más a la izquierda. El contenido exacto depende de los paquetes de servicio específicos que se hayan instalado en el sistema.

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución

	amd
Descripción	Habilita el demonio de montaje automático de NFS, amd.
Consideraciones	NFS se diseñó como un servicio LAN. Contiene numerosas debilidades de seguridad si se usa sobre Internet. No permite acceso a Internet a los sistemas de archivos montados NFS. De hecho, no ejecuta NFS en la máquina firewall. NFS lo forman varios demonios. Dentro del editor de nivel de ejecución, las secuencias de comandos de inicialización se llaman amd, nfs y nfsfs.
Recomendación	No ejecute amd en la máquina firewall.

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución *(continuación)*

	arpwatch
Descripción	Permite al demonio arpwatch registrar y crear una base de datos de los pares de direcciones IP/direcciones Ethernet que se ven en una interfaz LAN.
	arpwatch
Consideraciones	Si se configura correctamente, el demonio arpwatch no representa un problema de seguridad. arpwatch escucha en un socket de dominio privado de UNIX. Sin embargo, arpwatch colocará, de forma predeterminada, la interfaz de red externa en modo promiscuo, como un rastreador de paquetes, permitiendo a la interfaz examinar los paquetes que no van dirigidos a ella. Poner la interfaz de red en modo promiscuo es un signo evidente de un sistema comprometido. Si la interfaz se encuentra en modo promiscuo como algo normal del estado del sistema, se pierde la información de seguridad y no aprenderá nada que no supiera ya sobre la LAN.
Recomendación	No ejecute arpwatch en la máquina firewall.
	autofs
Descripción	Habilita el proceso de administración de montaje automático, automount.
Consideraciones	NFS es un servicio LAN basado en RPC. Tanto NFS como cualquier servicio de red que se basa en el demonio portmap son posibles agujeros de seguridad importantes y no deben ser accesibles desde Internet. Idealmente, estos servidores no deben ejecutarse en una máquina firewall. Además, automount se basa en el servicio de información de red (NIS), si el servicio está disponible. NIS es otro servicio LAN que es mejor no ejecutar en una máquina firewall.
Recomendación	No ejecute automount en la máquina firewall.
	bootparamd
Descripción	Habilita el servidor de parámetros de inicio.
Consideraciones	El demonio bootparamd proporciona información relacionada con el inicio de estaciones sin disco sobre una LAN. Un sistema particular o de una pequeña oficina no es probable que tenga estaciones sin disco. En cualquier caso, este servidor no se debe ejecutar en una máquina firewall.
Recomendación	No ejecute bootparamd.
	dhcpd
Descripción	Inicia un servidor DHCP local.
Consideraciones	Este servicio asigna direcciones IP asignadas dinámicamente a los host clientes. Este servicio se suele usar en un ISP o una LAN corporativa. Si el ISP asigna direcciones IP de forma dinámica, ejecutar un servidor DHCP local es una forma rápida de conseguir que se cierre la cuenta del ISP. Algunas personas tienen buenas razones para ejecutar un servidor local, pero debe tener cuidado en el nivel de firewall y en el nivel de configuración de servidor. No habilite el servidor dhcpd a menos que sepa lo que hace.

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución *(continuación)*

Recomendación	No ejecute dhcpd a menos que realmente lo necesite. No lo ejecute hasta que comprenda su configuración. No lo ejecute sin un firewall en el sitio.
	gated
Descripción	Habilita el demonio de enrutamiento de pasarela.
Consideraciones	El demonio gated maneja los protocolos de enrutamiento de red. El enrutamiento en una pequeña LAN es mejor y más seguro si se administra usando direcciones IP estáticas. Las máquinas firewall deben usar sólo enrutamiento estático. Si debe ejecutar un demonio de enrutamiento, elija gated en vez del más antiguo y menos seguro routed, porque gated es más compatible con el nuevo protocolo de enrutamiento OSPF; routed sólo es compatible con el protocolo más antiguo de enrutamiento RIP. Las máquinas firewall no deben funcionar como enrutadores dinámicos. gated no debe ejecutarse en la máquina firewall.
Recomendación	No ejecute gated.
	httpd
Descripción	Inicia el servidor web Apache para albergar un sitio web.
Consideraciones	La seguridad de los servidores web queda fuera del objetivo de este libro. Sin embargo, como en cualquier clase de servicio de acceso a archivos, la cuestión básica es limitar el acceso sólo a aquellas partes del sistema de archivos que quiere hacer público, y ejecutar los servidores como usuarios no privilegiados. Además, se debe evitar cualquier guión CGI a menos que comprenda los problemas de seguridad que pueden representar. Como los guiones CGI pueden ejecutar programas en el sistema, las cuestiones básicas implican una comprobación rigurosa de la entrada, ejecutarlos como usuario no privilegiado y usar nombres de ruta de acceso completos a cualquier programa o guión que ejecute el programa CGI.
Recomendación	Ejecute httpd posteriormente si decide albergar un sitio web. Lea primero la documentación del servidor web Apache y los comentarios en los archivos de configuración.

Más información acerca de problemas con el servidor web

Para obtener más información acerca de problemas con el servidor web, consulte el sitio web de Apache en www.apache.org, ***Apache Server for Dummies*** de Ken Coar (IDG Books) y *"How to Remove Meta-Characters from User-Supplied Data in CGI Scripts"* en www.cert.org.

	inet
Descripción	El demonio inetd es el fundamento para proporcionar muchos servicios de red. En lugar de tener, como mínimo, un demonio de cada servicio ejecutándose continuamente, se use o no el servicio, inetd reemplaza estos demonios con un solo programa, él mismo. inetd escucha las conexiones entrantes al puerto de servicio que administra, decide a qué servicio debería conectarse la

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución *(continuación)*

	petición, posiblemente usa un programa de ayuda para realizar las comprobaciones de permiso de acceso, inicia la ejecución del programa solicitado si no se ejecuta continuamente, y establece la conexión saliente entre el programa cliente que realiza la petición y el programa servidor que atiende la petición.
Consideraciones	Es necesario inetd si usa servicios comunes, como ftp o telnet, localmente, o si ofrece estos servicios a sitios remotos.
Recomendación	Ejecute inetd.
	innd
Descripción	Habilita un servidor de noticias Usenet local.
Consideraciones	No tiene que preocuparse de nada si dispone de un firewall en el sitio. Pocos sitios pequeños necesitan usar un servidor de noticias local. La configuración de un servidor de noticias es difícil. Si necesita un servidor local, asegúrese de que deniega el acceso remoto a cualquiera excepto a sitios remotos seguros específicos.
Recomendación	No ejecute innd.
	linuxconf
Descripción	Permite configurar la máquina usando un servidor web local como interfaz de usuario.
Consideraciones	linuxconf escucha en el puerto 98 TCP. De forma predeterminada, linuxconf sólo escucha la interfaz de bucle invertido. No es necesario preocuparse de nada si mantiene la configuración predeterminada. No cambie la configuración de acceso de linuxconf para permitir acceso externo.
Recomendación	Ninguna.
	lpd
Descripción	Habilita el servidor de impresión.
Consideraciones	A primera vista, puede pensar que el acceso a una impresora se restringe necesariamente a la máquina local. Con UNIX, las impresoras se tratan como dispositivos de red. Asegúrese de que tanto los archivos de configuración de acceso a la impresora como el firewall bloqueen el acceso remoto a la impresora, si dispone de alguna.
Recomendación	Ejecute lpd si se comparte una impresora entre máquinas UNIX.
	mars-nwe
Descripción	Habilita el archivo mars-nwe y el servidor de impresión para clientes Novell NetWare para Windows sobre la LAN.
Consideraciones	Los servidores de archivo local y de impresión no deben ejecutarse en una máquina firewall. Para habilitar el soporte NetWare, tendrá que volver a compilar el núcleo para que sea compatible con el nivel de red IPX, compatible con el nivel de transporte SPX y compatible para el sistema de archivos SPX. Como indica el documento de ayuda de la configuración del núcleo, si no lo comprende, no lo habilite.
Recomendación	No ejecute mars-nwe en la máquina firewall.

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución *(continuación)*

	mcserv
Descripción	Habilita el servidor de archivos Midnight Commander.
Consideraciones	mcserv administra el acceso al sistema de archivo en red Midnight Commander. Es un servidor de archivos no seguro destinado al uso en LAN. No permite el acceso de Internet externo al sistema de archivos Midnight Commander. mcserv es un servicio basado en RPC de UDP, que se basa en el demonio portmap. Idealmente, el servicio no debería ejecutarse en la máquina firewall.
Recomendación	No ejecute mcserv en la máquina firewall.
	named
Descripción	El demonio named proporciona la mitad del servidor DNS de red, traduciendo entre nombres de máquina simbólicos y sus direcciones IP numéricas. La parte del cliente de DNS, la que resuelve las traducciones, no es visible como un programa independiente. Es parte de las bibliotecas de red compiladas dentro de los programas.
Consideraciones	Casi con toda seguridad, la mayoría de los servicios DNS los proporcionará el ISP. Ejecutar un servidor local puede mejorar el rendimiento de red. Los servidores DNS simples no suponen riesgos de seguridad si se configuran correctamente.
Recomendación	Ejecute named después de comprender el servidor y cómo configurarlo. La configuración DNS puede ser complicada y misteriosa. Mientras tanto, dirija el traductor de nombres a los servidores de nombre del ISP.
	netfs
Descripción	Monta sistemas de archivos en red NFS, Samba, y NetWare.
Consideraciones	netfs no es un demonio de servicio. Es un guión de shell que se ejecuta una vez para montar los sistemas de archivos conectados en red de forma local. Tenga presente la idea general de que, normalmente, los servicios de sistemas de archivos de red no se deben usar en una máquina firewall.
Recomendación	No ejecute netfs en la máquina firewall.
	network
Descripción	La secuencia de comandos de configuración network se ejecuta en tiempo de inicio para activar las interfaces de red que se ha configurado. No es un servidor en sí.
Consideraciones	Debe ejecutar esta secuencia de comandos.
Recomendación	Ejecute network.
	nfs
Descripción	Habilita servicios NFS.
Consideraciones	NFS se diseñó como un servicio LAN. Contiene gran cantidad de debilidades de seguridad si se usa sobre Internet. No permite acceso a Internet a los sistemas de archivos montados NFS. De hecho, no ejecuta NFS en la máquina firewall. NFS lo forman varios demonios. Dentro del editor de nivel de ejecución, las secuencias de comandos de inicialización se llaman amd, nfs y nfsfs.
Recomendación	No ejecute nfs en la máquina firewall.

Más información acerca de DNS

Para obtener más información acerca de DNS, consulte DNS and BIND de Paul Albitz y Cricjet Liu (O´Reilly), el "DNS HOWTO" de Nicolai Langfeidt, disponible en la documentación en línea que puede encontrar en /usr/doc y las páginas man named(8), resolver(5) y hostname(7).

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución *(continuación)*

	nscd
Descripción	Habilita el demonio Name Switch Cache.
Consideraciones	nscd es un servicio para compatibilidad NIS que introduce en la caché las contraseñas de usuario y los miembros del grupo. NIS es un servicio LAN inherentemente seguro que no debe ejecutarse en una máquina firewall.
Recomendación	No ejecute nscd en la máquina firewall.
	portmap
Descripción	Habilita el administrador portmap RPC.
Consideraciones	El demonio portmap RPC se parece a inetd. Administra conexiones a servicios basados en RPC, como NFS y NIS. Si no se usan estos servicios localmente, deshabilite el demonio portmap. Si se utilizan estos servicios, asegúrese de que el firewall bloquea el acceso exterior a portmap. Puede consultar la información de control de acceso más actual de portmap en /etc/hosts.allow y /etc/hosts.deny.
Recomendación	No ejecute portmap en la máquina firewall.
	postgresql
Descripción	Inicia un servidor de base de datos SQL local.
Consideraciones	SQL es un servicio basado en TCP asociado con el puerto 5432. El servidor principal, postmaster, puede configurarse para usar sockets de dominio de Internet o sockets de dominio UNIX locales. Como servicio diseñado para acceder a archivos locales, SQL no se suele ofrecer desde una máquina firewall sin precauciones de firewall y configuraciones de seguridad. Consulte las páginas man postmaster(1), postgres(1) y psql(1), y la documentación en línea PostgreSQL-HOWTO.
Recomendación	No ejecute postgresql en la máquina firewall.

Debilidades que se suelen explotar

El demonio portmap representa una familia de las debilidades que más se suelen explotar en un sistema UNIX.

	routed
Descripción	Habilita el demonio routed para actualizar automáticamente el núcleo dinámico de las tablas de enrutamiento.

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución *(continuación)*

Consideraciones	Tanto routed como gated representan serios agujeros de seguridad, routed incluso más que gated. Es improbable que necesite administrar sus propias tablas de enrutamiento de forma dinámica usando RIP. Este es un servicio que ofrece el ISP. Simplemente use direccionamiento IP estáticos localmente.
Recomendación	No ejecute routed.
	rstatd
Descripción	Habilita el demonio rstatd para coleccionar y proporcionar información del sistema para otras máquinas de la LAN.
Consideraciones	La información de estado del sistema no debe compartirse con las máquinas de Internet remotas. El servicio no debe ejecutarse en una máquina firewall.
Recomendación	No ejecute rstatd en la máquina firewall.
	ruserd
Descripción	Habilita el servicio de localización de usuarios. Este es un servicio basado en RPC que ofrece información sobre usuarios individuales que tienen actualmente una sesión abierta en una de las máquinas de la LAN.
Consideraciones	Un sitio pequeño no tiene necesidad de este servicio LAN. Además, el demonio rpc.rusersd depende de los servicios RPC, que no deben usarse en una máquina firewall.
Recomendación	No ejecute ruserd en la máquina firewall.
	RWALLD
Descripción	Habilita el demonio de servicio rpc.rwalld. Este es un servicio basado en RPC que permite a los usuarios escribir mensajes a las terminales de los demás usuarios que tienen iniciada una sesión en una máquina de la LAN.
Consideraciones	Un sitio pequeño no tiene necesidad de este servicio LAN. Además, el demonio rpc.rwalld depende de servicios RPC, que no deben usarse en una máquina firewall.
Recomendación	No ejecute rwalld en la máquina firewall.
	rwhod
Descripción	Habilita el demonio de servicio rwhod. El demonio rwhod es compatible con los servicios rwho y ruptime para una LAN. Como tales, el servicio ofrece información sobre quién tiene una sesión iniciada, qué están haciendo, qué sistemas se están ejecutando y están conectados a la LAN, etc.
Consideraciones	Un sitio pequeño tiene poca necesidad de este servicio LAN. Idealmente, el demonio rwhod no debería usarse en una máquina firewall.
Recomendación	No ejecute rwhod en la máquina firewall.
	sendmail
Descripción	El servicio de correo local se controla mediante sendmail.

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución *(continuación)*

Consideraciones	sendmail es necesario si alberga servicios propios de correo. Correctamente configurado, sendmail es, actualmente, relativamente seguro. Sin embargo, sendmail sigue estando en el punto de mira de los hackers. Las actualizaciones de seguridad se ponen a disposición a medida que se descubren y corrigen los problemas. Los servicios SMTP tienen una larga historia de debilidades de seguridad, tanto en términos de permitir acceso al sistema general como en términos de usarse como modo de envío de correo masivo. Se ha realizado un gran esfuerzo para crear versiones actualizadas de sendmail más seguras. Tal y como se incluye con Red Hat, la configuración predeterminada es bastante segura, al menos en términos de envío de correo no autorizado.
Recomendación	Ejecute sendmail si quiere disponer de servicios de correo locales independientes de los que ofrece el ISP. No ejecute sendmail si usa exclusivamente el servicio de correo que proporciona el ISP.
	smb
Descripción	Permite el servicio Samba para compartir archivos, así como para compartir impresoras.
Consideraciones	Los servicios del sistema de archivos y para compartir dispositivos son servicios LAN y no se deben ejecutar en una máquina firewall. Los servicios Samba no deben estar disponibles de forma remota bajo ninguna circunstancia.
Recomendación	No ejecute smb en una máquina firewall.
	snmpd
Descripción	Habilita el demonio simple de administración de red. El demonio snmpd controla la administración de red SNMP.
Consideraciones	SNMP es un servicio de administración LAN. Por motivos de seguridad, como es un servicio UDP local, snmpd no debe ejecutarse en una máquina firewall. Es muy improbable que una LAN pequeña lo necesite. Si es necesario usarlo, debe considerarse como un servicio peligroso y asegúrese de bloquear todo el tráfico entre la LAN e Internet. No es deseable que las personas extrañas administren la red, además de que no apreciarán la posibilidad de ver los paquetes originados desde el demonio SNMP.
Recomendación	No ejecute snmpd.
	squid
Descripción	Habilita la Squid Internet Object Cache. Si no se ejecuta el servidor web Apache localmente, squid puede servir como un servidor proxy HTTP local y como caché web local para las páginas web obtenidas de sitios remotos. Es necesario algún esfuerzo para configurar squid.
Consideraciones	Si se configura correctamente, squid no implica especiales consideraciones de seguridad si tiene un firewall en un sitio.
Recomendación	Ejecute squid posteriormente si quiere que las páginas web se inserten en la caché localmente y si usa las características de caché del servidor Apache.

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución *(continuación)*

	syslog
Descripción	La secuencia de comandos de configuración syslog inicia los demonios de registro del sistema syslogd y klogd en tiempo de inicio. Este servicio es necesario para que el estado del sistema y los mensajes de error se escriban en los archivos de registro. syslogd puede configurarse para ser ejecutado como un servicio LAN.
Consideraciones	Este servicio es necesario.
Recomendación	Ejecute syslog.
	xfs
Descripción	Habilite el servidor de fuentes de X Window, que es nuevo en la versión 6.0 de Red Hat, para servir fuentes a servidores X remotos y locales.
Consideraciones	Aunque xfs puede configurarse para escuchar en un socket de dominio de Internet de TCP para servidores remotos, en su configuración predeterminada, xfs escucha en un socket de domino UNIX privado. Como tal, xfs no representa en sí un riesgo para la seguridad. El servidor de X Window depende de xfs. Idealmente, el servicio no debe ejecutarse en una máquina firewall.
Recomendación	No ejecute xfs en una máquina firewall, si es posible.
	xntpd
Descripción	Habilita un servidor de tiempo de red local.
Consideraciones	No tendrá que preocuparse de nada si dispone de un firewall en el sitio. Algunos sitios programan cron para ejecutar periódicamente el programa cliente ntpd o ntpdate, con el fin de conseguir la hora actual desde un servidor de tiempo remoto oficial. El servidor xntpd local se ejecuta para distribuir la hora del sistema actual entre máquinas locales de una LAN interna.
Recomendación	Ejecute xntpd después de comprender las cuestiones de la configuración, si desea un servidor de tiempo local para máquinas LAN.
	ypbind
Descripción	Habilita el demonio ypbind para máquinas que se ejecutan como clientes NIS.
Consideraciones	El paquete NIS incluye las secuencias de comandos de configuración en tiempo de inicio ypbind, yppasswdd y ypserv. NIS es un servicio LAN que ofrece funciones de red centralizadas de usuario y de administración de máquina. No es probable que una pequeña LAN lo utilice. Si es necesario usarlo, considérelo como un servicio peligroso y asegúrese de bloquear todo el trafico NIS y RPC entre la LAN e Internet.
Recomendación	No ejecute ypbind en la máquina firewall.
	yppasswdd
Descripción	Habilita el servidor de contraseñas NIS.
Consideraciones	El paquete NIS incluye las secuencias de comandos de configuración en tiempo de inicio ypbind, yppasswdd y ypserv. NIS es un servicio LAN que ofrece funciones de red centralizadas de usua-

Tabla 2.1. Servicios de red disponibles en los diferentes niveles de ejecución *(continuación)*

	rio y de administración de máquina. No es probable que una pequeña LAN lo utilice. Si es necesario usarlo, considérelo como un servicio peligroso y asegúrese de bloquear todo el trafico NIS y RPC entre la LAN e Internet.
Recomendación	No ejecute yppasswdd en la máquina firewall.
	ypserv
Descripción	Habilita el servidor maestro de NIS.
Consideraciones	El paquete NIS incluye las secuencias de comandos de configuración en tiempo de inicio ypbind, yppasswdd y ypserv. NIS es un servicio LAN que ofrece funciones de red centralizadas de usuario y de administración de máquina. No es probable que una pequeña LAN lo utilice. Si es necesario usarlo, considérelo como un servicio peligroso y asegúrese de bloquear todo el trafico NIS y RPC entre la LAN e Internet.
Recomendación	No ejecute ypserv en la máquina firewall.

Servicios administrados por inetd

Ahora que tenemos un conocimiento más amplio sobre el administrador de nivel de ejecución y sobre qué servicios de sistema básicos se inician automáticamente cuando la máquina arranca, podemos pasa a explicar los servicios de red. Los servicios de red pueden ser tanto locales como públicos, y se inician mediante inetd. Algunos servicios de red que se ofrecen desde un sitio se hacen disponibles a través del superservidor inetd. Los servicios que administra inetd se especifican en el archivo de configuración /etc/inetd.conf.

Prácticamente todos los servicios de este archivo son servicios LAN y serán necesarios en una máquina firewall. En algunas ocasiones, /etc/inetd.conf varía según la versión de Linux. La configuración predeterminada que se ofrece con las versiones recientes de Red Hat Linux, incluyendo la versión 6.0, no son seguras para un sistema conectado directamente a Internet.

El contenido exacto del archivo /etc/inetd.conf difiere según el fabricante y la versión. En general, el archivo /etc/inetd.conf contiene los servicios que se describen en las siguientes secciones, en este orden.

Servicios para probar la red

El primer conjunto de servicios lo ofrece internamente el demonio inetd para usarlo en pruebas de redes y solución de problemas. Es probable que nadie que lea este libro necesite nunca estas utilidades de prueba:

```
#echo        stream  tcp     nowait   root   internal
#echo        dgram   udp     wait     root   internal
#discard     stream  tcp     nowait   root   internal
#discard     dgram   udp     wait     root   internal
#daytime     stream  tcp     nowait   root   internal
#daytime     dgram   udp     wait     root   internal
```

```
#chargen     stream  tcp     nowait  root   internal
#chargen     dgram   udp     wait    root   internal
#time        stream  tcp     nowait  root   internal
#time        dgram   udp     wait    root   internal
```

Si se leen detenidamente, los servicios parecen inofensivos. Sin embargo, discard y chargen pueden usarse conjuntamente en un ataque por denegación de servicio UDP bien orquestado. Consulte la sección "Inundación UDP", anteriormente en este capítulo, como una muestra de la advertencia del CERT que describe estos ataques.

Servicios estándar

Varios servicios UNIX estándar se administran bastante bien mediante inetd, Los servidores ftp y telnet:

- ftp—ftp es uno de los medios más comunes de compartir archivos sobre Internet. Sin embargo, ftp está lleno de agujeros de seguridad y se ha explotado frecuentemente cuando no se configura de forma segura. Una configuración correcta es minuciosa y requiere dedicación:

  ```
  #ftp    stream tcp  nowait root  /usr/sbin/tcpd in.ftpd -l -a
  ```

  Si desea ofrecer servicios FTP generales a Internet, deberá consultar primero la documentación de configuración y alguno de los documentos del CERT que explican los temas de seguridad de FTP. Si lo que se quiere es ofrecer unos pocos archivos del sitio a usuarios anónimos, debe considerar usar un servidor web en vez de hacer disponibles estos archivos. Los servicios de FTP anónimo están más predispuestos a violaciones de seguridad, debido a configuraciones erróneas, que los servicios FTP autenticados. Se recomienda incluso no instalar el paquete FTP anónimo. Idealmente, no se ofrece ftp autenticado a usuarios remotos desde una máquina firewall.
  Si se desean ofrecer servicios FTP sólo a la LAN, será necesario habilitar el servicio en el archivo /etc/inetd.conf. El acceso exterior puede deshabilitarse en los archivos de configuración de ftp, del firewall y en la configuración de tcp_wrapper. Puede encontrar la documentación de tcp_wrapper en las páginas man tcpd(8), hosts_access(5), inetd.conf(5), hosts_options(5) y syslog.conf(5).
- telnet—telnet es uno de los medios más comunes de iniciar sesiones en sistemas remotos, tanto sobre Internet como localmente entre máquinas UNIX y máquinas no UNIX. Si tiene una LAN, es probable que quiera habilitar estos servicios a menos que pueda usar ssh en todas las máquinas locales. Si quiere acceder a una máquina desde cuentas remotas sobre Internet, es necesario telnet a menos que el host remoto proporcione algún servicio más seguro, como SSH:

  ```
  #telnet stream tcp  nowait root  /usr/sbin/tcpd in.telnetd
  ```

telnet se considera como inseguro sobre Internet porque pasa información en formato de texto ASCII sin cifrar, incluyendo el nombre de inicio de sesión y la contraseña de inicio de sesión. Los rastreadores de paquetes pueden capturar este tipo de información.
Si necesita tener acceso telnet de forma interna, pero no quiere permitir el acceso a inicio de sesión desde sitios remotos, el acceso puede limitarse a las máquinas locales mediante el firewall, mediante tcp_wrappers y en el archivo /etc/security/access.conf.

- gopher. El servicio de obtención de información gopher es un servicio estándar. Algunas personas todavía lo usan, pero ha sido ampliamente reemplazado por servidores web y motores de búsqueda. Actualmente, es casi innecesario ofrecer este servicio. gopher ha sido eliminado del archivo /etc/inetd.conf en la versión 6.0 de Red Hat:

```
#gopher stream tcp  nowait root  /usr/sbin/tcpd gn
```

Cuestiones relacionados con la seguridad FTP

Para obtener información acerca de los problemas relacionados con FTP, consulte los siguientes documentos disponibles en www.cert.org: *"Anonymous FTP Configurations Guidelines"* (Directrices generales acerca de la configuración FTP anónima), *"Anonymous FTP Abuses"* (Abusos de FTP anónimo) y *"Problems with the FTP PORT Command"* (Problemas con el comando PORT de FTP).

Servicios de acceso remoto BSD

Los servicios BSD se desarrollaron como parte de la versión estándar del UNIX de Berkeley con el fin de hacer más práctico el acceso entre cuentas en múltiples máquinas LAN:

- Se accede a los servicios de shell remotos, servicios de acceso remoto BSD (llamados shell, login y exec en el archivo /etc/inetd.conf) a través de los programas rsh y rlogin y la llamada a la biblioteca rexec. Son servicios LAN diseñados para facilitar el acceso local entre máquinas a un mismo usuario, sin necesidad de volver a autenticarse. Un sistema firewall no debería usar estos servicios. Asegúrese de que están deshabilitados en la máquina firewall. Si no son necesarios, asegúrese de bloquear todos los accesos externos mediante el firewall y en la configuración de tcp_wrapper. No se debe permitir nunca acceso externo a estos servicios, o los hackers accederán inmediatamente:

```
#shell stream tcp   nowait root   /usr/sbin/tcpd in.rshd
#login stream tcp   nowait root   /usr/sbin/tcpd in.rlogind
#exec  stream tcp   nowait root   /usr/sbin/tcpd in.rexecd
```

- Servicios Talk. Los servicios Talk no son tan inseguros en sí mismos, pero ¿realmente es necesario que la gente en Internet sea capaz de es-

cribir mensajes en la terminal, o incluso que conozcan las cuentas que existen en el sistema? Un sistema pequeño probablemente no los necesitará. Los servicios talk se han reemplazado en Internet por servicios como IRC o Instant Messenger:

```
#comsat dgram  udp   wait  root   /usr/sbin/tcpd in.comsat
#talk   dgram  udp   wait  root   /usr/sbin/tcpd in.talkd
#ntalk  dgram  udp   wait  root   /usr/sbin/tcpd in.ntalkd
#dtalk  stream tcp   wait  nobody /usr/sbin/tcpd in.dtalkd
```

Explosiones *sunrpc* y *mountd*

Para obtener información acerca de las explosiones sunrpc(111/udp/tcp) y mountd(635/udp), consulte www.cert/org/advisories/CA-98.12.mountd.html. Para obtener más información acerca de las explosiones imap(143/tcp), consulte www.cert/org/advisories/CA-98.09.imapd_vul.html. Para obtener más información acerca de las explosiones pop-3(110/tcp), consulte www.cert/org/advisories/CA-98.08.qpopper_vul.html.

Servicios de entrega de correo

Los servidores de entrega de correo, tanto los servicios POP como los IMAP, son agujeros de seguridad importantes cuando no se configuran correctamente. Además de los servidores portmap y mountd, los servidores pop-3 e imap son los tres servidores que más se suelen comprometer actualmente:

- pop-2—pop-2 se ha reemplazado casi completamente por el más reciente protocolo pop-3. No hay probablemente ninguna razón para que alguien ofrezca este servicio:

```
#pop-2  stream tcp   nowait root   /usr/sbin/tcpd ipop2d
```

- pop-3. Puede que sea necesario ofrecer servicios POP, aunque no es muy probable. Algunos querrán ofrecer servicios POP a unos pocos usuarios remotos, o quizá ofrecer servicios POP de forma local como medio de obtener el correo electrónico desde un servidor de correo central. De nuevo, si sólo es necesario un servidor POP local, asegúrese de bloquear el acceso externo en el firewall, en el tcp_wrapper y en los archivos de configuración de POP en el directorio /etc/ppp:

```
#pop-3  stream tcp   nowait root   /usr/sbin/tcpd ipop3d
```

- imap, Al igual que con pop-3, puede que sea necesario ofrecer servicios de correo imap, si bien la mayoría de la gente no necesitará habilitar este servicio:

```
#imap  stream tcp   nowait root   /usr/sbin/tcpd imapd
```

uucp

¡No habilite uucp! El servicio UUCP ha sido una forma muy habitual de enviar archivos entre máquinas remotas. Algunos servidores de noticias se ba-

san en uucp para conseguir las noticias. Unos pocos sitios todavía lo usan como alternativa a ftp, bajo condiciones de seguridad controladas con mucho cuidado, pero está cayendo en desuso:

```
#uucp  stream tcp  nowait uucp  /usr/sbin/tcpd  /usr/lib/uucp/uucico -l
```

Servicios de inicio remoto

Los servicios de inicio remoto se usan para asignar direcciones IP y arrancar las máquinas sin disco en la LAN. Las estaciones sin disco y los enrutadores usan estos servicios:

- tftp. Como indica el comentario en el archivo /etc/inetd.conf, ¡no habilite este servicio! Su propósito es cargar el sistema operativo en un sistema remoto sin disco o en un enrutador sobre la red segura local. Por desgracia, bastante gente usa tftp como una alternativa a ftp, pensando que es más seguro porque no utiliza autenticación. Justo lo contrario es lo correcto:

  ```
  #tftp  dgram  udp   wait  root   /usr/sbin/tcpd in.tftpd
  ```

- bootps. BOOTP lo usan las estaciones de trabajo sin disco para descubrir la dirección IP propia, la ubicación del servidor de inicio y para iniciar la descargar el sistema sobre tftp antes de iniciar. No puedo imaginar por qué una instalación de usuario medio particular o de pequeña empresa pueda necesitar este servicio. Déjelo desactivado:

  ```
  #bootps dgram  udp   wait  root   /usr/sbin/tcpd bootpd
  ```

Servicios de información

Los servicios considerados como servicios de información en el archivo /etc/inetd.conf, proporcionan información sobre la cuenta del usuario, el proceso, la conexión de red y la configuración. Si se usan estos servicios, deben limitarse a otras máquinas de la LAN. Estos servicios escuchan en dos conjuntos de sockets, los sockets de dominio para host remotos inet y los sockets de dominio UNIX para peticiones locales. Si se deshabilitan estos servicios en el archivo /etc/inetd.conf, se deshabilitará el acceso de host remotos. El acceso local no se ve afectado:

- finger—finger, es un servicio de información. Se utiliza tan poco que es realmente un agujero para la seguridad, pero proporciona información sobre las cuentas de usuario, tiempos de inicio de sesión, correo pendiente, nombres de máquinas, etc. Aunque esta información no es necesariamente sensible, puede usarla un hacker. No se recomienda permitir el acceso al servicio finger en el entorno actual de Internet:

  ```
  #finger   stream tcp   nowait root   /usr/sbin/tcpd in.fingerd
  ```

- Servicios de información del sistema. Junto con finger, cfinger, systat y netstat, ofrecen información de la cuenta del usuario y del sistema que

posiblemente sea útil para un hacker. Si se deshabilitan estos servicios en el archivo /etc/inetd.conf, se deshabilita el acceso de red a estos servicios. El acceso local sigue disponible:

```
#cfinger  stream  tcp  nowait root  /usr/sbin/tcpd  in.cfingerd
#systat   stream  tcp  nowait guest /usr/sbin/tcpd  /bin/ps -
auwwx
#netstat  stream  tcp  nowait  guest  /usr/sbin/tcpd   /bin/netstat
-f inet
```

RARP, BOOTP y DHCP

Los servidores bootpd y dhcpd comparten el mismo puerto de servicio en /etc/services/, aunque son servicios distintos.
Históricamente, las máquinas sin disco conocían su dirección hardware a partir de la interfaz Ethernet, pero no conocían su dirección software IP. El protocolo de resolución de direcciones inversas (RARP, Reverse Address Resolution Protocol) se desarrolló para permitir a las máquinas sin disco preguntar a un servidor su dirección IP basándose en su dirección hardware MAC. RARP se reemplazó por BOOTP, el protocolo de secuencia de inicio (Bootstrap Protocol). BOOTP no sólo proporciona la dirección IP, sino que también proporciona la dirección del servidor de archivos para descargar la imagen de inicio de la estación de trabajo mediante tftp. BOOTP no se ha reemplazado, pero ha evolucionado hasta DHCP, el protocolo de configuración dinámica de host (Dynamic Host Configuration Protocol). DHCP incluye un superconjunto de las características de BOOTP, que proporciona información adicional de la dirección del servidor y del enrutador situado más allá de la dirección IP del host, así como asignación dinámica de direcciones IP reutilizables.

Autenticación

El servicio AUTH se parece algo a finger en el sentido que ofrece información del usuario, normalmente a servidores de correo y de noticias, con el fin de iniciar una sesión cuando se envía correo o se coloca un artículo en las noticias. También, es poco frecuente que los servidores FTP soliciten una búsqueda de identd válida. Si decide habilitar el servicio de autenticación de usuario AUTH identd, aténgase a las consecuencias. El tema está abierto a debate, con gente que apoya cada una de las perspectivas y con buenas razones para su elección. Algunos insisten en habilitar el servicio como una cuestión de cortesía. Otros lo deshabilitan para mantener la privacidad de la información de la cuenta de usuario:

```
auth stream tcp  nowait nobody /usr/sbin/in.identd in.identd -l -e -o
```

Servicios definidos localmente

Además del servicio automático y de la administración de inicio de demonio que hacen el administrador de nivel de ejecución e inetd, es necesario iniciar de forma explícita otros servicios y demonios que no son parte de la distribución Linux estándar. Estos programas se suelen iniciar desde el archivo /etc/rc.d/rc.local.

Durante el inicio, se ejecuta la secuencia de comandos /etc/rc.d/rc.local para ejecutar cualquier opción de configuración local que se haya definido. De forma predeterminada, se crean los archivos que se usan para mostrar el titular de inicio de sesión. Otros posibles usos son el inicio de programas locales, como el servidor sshd, o actualizar la hora del sistema desde un servicio de tiempo remoto.

Resumen

Una vez analizados los fundamentos de los protocolos IP y las consideraciones sobre los servidores ya explicados, en el Capítulo 3 mostraremos el proceso de creación de un firewall real para un sitio. La secuencia de comandos del firewall usa la información que se encuentra disponible en los paquetes de red descritos en el Capítulo 1 para construir las reglas de entrada y salida específicas del firewall que implementen los conceptos explicados en este capítulo.

3

Creación e instalación de un firewall

En el Capítulo 2, "Conceptos del filtrado de paquetes", se han explicado las ideas en que se basa y los conceptos subyacentes a un firewall de filtrado de paquetes. Cada cadena de regla de firewall tiene su propia directiva predeterminada. Cada regla se aplica no sólo a una cadena input u output individual, sino también a una interfaz de red específica, a un tipo de protocolo de mensaje (como TCP, UDP o ICMP) y a un número de puerto de servicio. Las reglas individuales de aceptación, denegación y rechazo las definen las cadenas input, output y forward, sobre las que aprenderemos más al final de este capítulo y en el Capítulo 4, "Redes de perímetro, firewalls múltiples y problemas con las LAN". Este capítulo reúne esos conceptos para mostrar cómo crear un sistema simple de una sola máquina firewall diseñado a medida para un sitio.

El firewall que crearemos en este capítulo se basa en una directiva denegar todo predeterminada. Es decir, que todo el tráfico de red se bloquea de forma predeterminada y los servicios se habilitan de forma individual como excepciones a la directiva.

Después de crear el firewall de sistema de una sola máquina, el capítulo termina mostrando cómo extender el firewall independiente a un firewall bastión formal. Un firewall bastión tiene al menos dos interfaces de red. El firewall bastión aísla una LAN interna de la comunicación directa con Internet. Sólo son necesarias pequeñas extensiones para hacer funcionar al firewall de sistema de una sola máquina como un simple firewall bastión de doble tarjeta. Esto protege la LAN interna aplicando las reglas de filtrado de paquetes en la interfaz externa, que actúa como una pasarela proxy entre la LAN e Internet.

Los firewall de sistema de una sola máquina y el firewall bastión son las formas menos seguras de la arquitectura firewall. Si se comprometiera el host firewall, cualquier máquina local estaría abierta a un ataque. Al ser un firewall independiente, es una proposición de todo o nada. Como este libro está destinado a los usuarios particulares y o de pequeñas empresas, se supone que la mayoría de los usuarios particulares tienen un único equipo conectado a Internet, o una sola máquina firewall que protege a una pequeña LAN privada. Sin embargo, hablar de "formas menos seguras", implica que estos firewall sean inseguro, auqnue no sean tan flexibles que las arquitecturas más complejas que implican múltiples máquinas. El Capítulo 4 introduce configuraciones más flexibles que permiten una protección adicional de la seguridad interna para configuraciones más complicadas de LAN y de servidor de las que se puede conseguir con un firewall de sistema de una sola máquina.

ipchains: El programa de administración de firewall de Linux

Este libro se basa en la versión Linux 6.0 de Red Hat. Linux incluye un mecanismo de firewall llamado IPFW (firewall IP). Las principales versiones de Linux tienen el programa ipchains o están en el proceso de convertirse a ipchains, una reescritura de la versión 4 del programa IPFW. Esta nueva versión se suele conocer con el nombre de ipchains, el nombre de su programa de administración. Las versiones anteriores de Linux usaban una antigua implementación de IPFW. Su mecanismo de firewall se suele conocer con el nombre de ipfwadm, el nombre del programa de administración de la versión más antigua.

En los ejemplos de este libro se usa ipchains. Como ipfwadm se sigue usando bastante en los sistemas LINUX más antiguos, las versiones ipfwadm de los ejemplos se presentan en el Apéndice B, "Ejemplos de firewalls y secuencias de comandos compatibles". Aunque la sintaxis difiere entre ipchains e ipfwadm, son funcionalmente iguales. ipchains incluye el conjunto de características de ipfwadm, además de características adicionales que se encuentran en la nueva implementación de IPFW. En este libro no se usan las nuevas características de ipchains. En los ejemplos sólo se usan las características comunes a ambas versiones.

Como programa de administración de firewall que es, ipchains crea las reglas de filtrado de paquetes individuales para las cadenas input y output que componen el firewall. Uno de los aspectos más importantes a la hora de definir las reglas del firewall es tener presente que, a la hora de definir las reglas, el orden es importante.

Las reglas de filtrado de paquetes se almacenan en tablas del núcleo, en una cadena input, output o forward, en el orden en que se definen. Las reglas individuales se insertan al principio de la cadena o se agregan al final de la cadena. El orden en que se definen las reglas es el orden en que se agregarán a

las tablas del núcleo y, por tanto, el orden de comparación de cada regla con cada paquete.

A medida que llega a la interfaz de red cada paquete originado externamente, sus campos de encabezado se comparan con cada regla de la cadena input de la interfaz hasta que se encuentra una coincidencia. A la inversa, conforme se envía cada paquete generado internamente a una interfaz de red, sus campos de encabezado se comparan con cada regla de la cadena output de la interfaz hasta que se encuentra una coincidencia. En ambas direcciones, cuando se encuentra una coincidencia, la comparación se detiene y se aplica la disposición de paquetes de la regla: ACCEPT, REJECT o DENY. Si el paquete no coincide con ninguna regla de la cadena, se aplica la directiva predeterminada de dicha cadena. Lo más importante es que la primera regla que coincide gana.

Los puertos de servicio, en todos los ejemplos de reglas de este capítulo, se usan mediante sus números, en vez de por sus nombres simbólicos, como se listan en el archivo /etc/services. El programa ipchains es compatible con los nombres de puerto de servicio simbólicos. Los ejemplos de este capítulo usan los valores numéricos, ya que los nombres simbólicos no son consistentes entre las diferentes versiones de Linux, o incluso entre una versión y la siguiente. Se pueden usar los nombres simbólicos en las reglas por motivos de claridad, pero recuerde que el firewall puede ser inservible con la siguiente actualización del sistema.

ipchains es un programa C compilado. Debe llamarse una vez por cada regla de firewall individual que se defina. Esto se hace desde una secuencia de comandos del shell. Los ejemplos de este capítulo suponen que la secuencia de comandos del shell se llama /etc/rc.d/rc.firewall. En los casos donde la semántica del shell difiera, los ejemplos se escriben con la semántica de los shell Bourne (sh), Korn (ksh) o Bourne Again (bash).

Los ejemplos no están optimizados. Se explican detalladamente por motivos de claridad y para mantener la compatibilidad conceptual y de conjunto de las características entre ipchains e ipfwadm. Los dos programas usan diferentes argumentos de línea de comandos para hacer referencia a características similares y ofrecen atajos y capacidades de optimización ligeramente diferentes. Los ejemplos se presentan usando las características comunes a ambos programas.

Opciones del programa ipchains que se usan en la secuencia de comandos del firewall

En este capítulo no se explican las opciones del programa ipchains de forma completa. Sólo se explican las características que se usan en los ejemplos de este libro y las características comunes a ipfwadm. La Tabla 3.1 lista los argumentos de la línea de comandos ipchains que se usan aquí:

```
ipchains -A¦I [cadena] [-i interfaz] [-p protocolo] [ [!] -y]
         [-s dirección [puerto[:puerto]]]
         [-d dirección [puerto[:puerto]]]
         -j directiva [-l]
```

Para obtener una descripción del conjunto completo de características de ipchains, consulte la página man *on-line* ipchains e IPCHAINS-HOWTO.

Tabla 3.1. Opciones de ipchains que se usan en la secuencia de comandos del firewall

Opción	Descripción
-A [*cadena*]	Agrega una regla al final de una cadena. Los ejemplos usan las cadenas internas input, output y forward. Si no se especifica una cadena, la regla se aplica a todas las cadenas.
-I [*cadena*]	Inserta una regla al principio de una cadena. Los ejemplos usan las cadenas internas, input, output y forward. Si no se especifica una cadena, la regla se aplica a todas las cadenas.
-i <*interfaz*>	Especifica la interfaz de red a la que se aplica la cadena. Si no se especifica una interfaz la regla se aplica a todas las interfaces. Nombres comunes de interfaz son eth0, eth1, o y ppp0.
-p <*protocolo*>	Especifica el protocolo al que se aplica la regla. Si no se usa la opción -p, la regla se aplica a todos los protocolos. Los nombres de los protocolos compatibles son tcp, udp, icmp y all. También se permite cualquiera de los nombres de protocolos o números de protocolos del archivo /etc/protocols.
-y	El indicador SYN debe estar activado y el indicador ACK debe ponerse a cero en un mensaje TCP, indicando una petición de establecimiento de conexión. Si no se incluye -y como argumento, no se comprueban los bits indicadores de TCP.
! -y	El indicador ACK debe estar activado en un mensaje TCP, indicando una respuesta inicial a una petición de conexión o una conexión establecida y activa. Si no se incluye -y como argumento, no se comprueban los bits indicadores de TCP.
-s <*dirección*> [<*puerto*>]	Especifica la dirección origen del paquete. Si no se especifica una dirección origen, están implicadas todas las direcciones origen unidifusión. Si se da un puerto o un intervalo de puertos, la regla se aplica sólo a dichos puertos. Sin un especificador de puerto, la regla se aplica a todos los puertos origen. Un intervalo de puertos se define mediante un número de puerto inicial y un número de puerto final, separados por dos puntos (por ejemplo, 1024:65535). Si se da un puerto, se debe especificar una dirección.
-d <*dirección*> [<*puerto*>]	Especifica la dirección destino del paquete. Si no se especifica una dirección destino, están implicadas las direcciones destino unidifusión. Si se da un puerto o un intervalo de puertos, la regla sólo se aplica a dichos puertos. Sin un especificador de puerto, la regla se aplica a todos los puertos destino. Un intervalo de puertos se define mediante un número de puerto inicial y un número de puerto final separados por dos puntos (por ejemplo, 1024:65535). Si se da un puerto, debe especificarse una dirección.

Tabla 3.1. Opciones de ipchains que se usan en la secuencia de comandos del firewall *(continuación)*

Opción	Descripción
-j <*directiva*>	Especifica la directiva de disposición de paquete para esta regla: ACCEPT, DENY o REJECT. La cadena forward puede utilizar también la directiva MASQ (enmascarada).
-l	Escribe un mensaje de información en el núcleo (KERN_INFO) en el registro del sistema, /var/log/messages de forma predeterminada, siempre que un paquete coincida con en esta regla.

Opciones de direccionamiento origen y destino

Tanto la dirección origen como la dirección destino de un paquete pueden especificarse en una regla de firewall. Sólo los paquetes con dichas direcciones específicas origen o destino coincidirán con la regla. Las direcciones pueden ser una dirección IP específica, un nombre de host totalmente cualificado, un nombre de (dominio) red, un intervalo restringido de direcciones, o todo a la vez.

Una dirección IP es un valor numérico de 32 bits, dividido en cuatro bytes de 8 bits individuales, que van de 0 a 255. En notación decimal separada por puntos, cada uno de los cuatro bytes que forman el valor de 32 bits se representa como una de las cuatro tuplas de la dirección IP. La dirección IP privada de clase C, 192.168.10.30, se usa como la dirección de host local en las figuras que aparecen a lo largo de este libro.

El programa ipchains permite poner un sufijo a la dirección con un especificador de máscara de bits. El valor de la máscara puede variar entre 0 y 32, indicando el número de bits a enmascarar. Los bits se cuentan desde la izquierda, o tupla más significativa. Este especificador de máscara indica cuántos de los bits de la dirección, empezando a contar por la izquierda, deben coincidir con la dirección IP definida en la regla.

Una máscara 32, /32, significa que deben coincidir todos los bits. La dirección debe coincidir exactamente con la dirección que se 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 definida de forma predeterminada. No será necesario especificarla.

Un ejemplo del uso del enmascaramiento sería permitir sólo un tipo de conexión entre el usuario y las máquinas del servidor ISP. Supongamos que el ISP usa direcciones en el intervalo de direcciones 192.168.24.0 a 192.168.27.255 para el espacio de direcciones del servidor. En este caso, el par dirección/máscara será 192.168.24/22. cómo se muestra en la Figura 3.1, los primeros 22 bits de todas las direcciones en este intervalo son idénticas, por lo que cualquier dirección que encaje en los primeros 22 bits coincidirá con la regla. En efecto, se indica que se permiten conexiones al servicio sólo

cuando lo ofrezcan máquinas que pertenezcan al intervalo de direcciones que va desde la dirección 192.168.24.0 hasta la dirección 192.168.27.255.

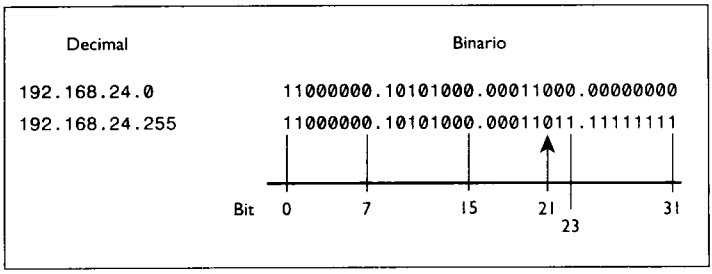

Figura 3.1. Los primeros 22 bits que coinciden con la dirección IP enmascarada pertenecen al intervalo 192.168.24.0/22

Una máscara 0, /0, significa que no es necesario que los bits de la dirección coincidan. En otras palabras, como no es necesario que coincidan los bits, usar /0 es lo mismo que no especificar una dirección. Cualquier dirección de unidifusión coincide. ipchains tiene un alias interno para la dirección 0.0.0.0/0, any/0.

Direcciones IP expresadas como nombres simbólicos

Las redes y host remotos se pueden especificar como nombres de host o red completamente cualificados. El uso de un nombre de host es especialmente recomendable para las reglas del firewall que se aplican a un host remoto individual. Esto es especialmente verdadero para los host cuya dirección IP puede cambiar, o que representan de forma invisible varias direcciones IP, como a veces ocurre con un servidor de correo del ISP. Sin embargo, en general las direcciones remotas se expresan en notación de tuplas separadas por puntos, debido a la posibilidad de usurpamiento de direcciones IP

No es posible resolver nombres de host simbólicos hasta que se habilite el tráfico DNS en las reglas del firewall. Si se usan nombres de host en las reglas del firewall, dichas reglas deben verificar las reglas que habiliten el tráfico DNS.

Inicialización del firewall

Un firewall se implementa como una serie de reglas de filtrado de paquetes definidas mediante opciones en la línea de comandos de ipchains. ipchains se ejecuta una vez para cada regla individual (firewalls diferentes pueden tener desde una docena de reglas hasta cientos).

Las llamadas al programa ipchains deben hacerse desde una secuencia de comandos del shell ejecutable y no directamente desde la línea de comandos. Es necesario llamar a todas las secuencias de comandos del shell del firewall. No intente llamar a reglas ipchains específicas desde la línea de comandos porque esto puede provocar que el firewall acepte o deniegue paquetes de forma incorrecta. Cuando las cadenas se inicializan y se habilita la directiva de-

negar predeterminada, todos los servicios de red se bloquean hasta que se define un filtro de aceptación para permitir el servicio individual.

De igual forma, la secuencia de comandos debe ejecutarse desde el shell desde la consola. No ejecute la secuencia de comandos del shell desde una máquina remota o desde una sesión xterm de X Window. No sólo se bloqueará el tráfico de red remoto, sino que el acceso a la interfaz de bucle invertido que usa X Window se bloqueará hasta que se vuelva a habilitar explícitamente el acceso a la interfaz.

Además, recuerde que los filtros del firewall se aplican en el orden en que se definen en la cadena input o en la cadena output. Las reglas se agregan al final de la cadena en el orden en que se definen. La primera regla que coincida gana. Por ello, las reglas del firewall deben definirse en orden jerárquico empezando por las reglas más específicas para terminar con las más generales.

La *inicialización del firewall* se usa para abarcar una extensa área, incluyendo la definición de constantes globales que se usan en la secuencia de comandos del shell, vaciar y ordenar cualquier regla existente en las cadenas del firewall, definir directivas predeterminadas para las cadenas input y output, volver a habilitar la interfaz de bucle invertido para que el sistema funcione normalmente, denegar el acceso desde cualquier host o red específicos que haya decidido bloquear, y definir algunas reglas básicas para protegerse de direcciones incorrectas y para proteger ciertos servicios que se ejecutan en puertos no privilegiados.

Constantes simbólicas que se usan en los ejemplos de firewall

Una secuencia de comandos del shell del firewall es más fácil de leer y mantener si se usan constantes simbólicas para nombres y direcciones recurrentes. Las siguientes constantes se usan en los ejemplos de este capítulo, o bien son constantes universales definidas en los estándares de redes:

```
EXTERNAL_INTERFACE="eth0"              # interfaz conectada a Internet
LOOPBACK_INTERFACE="lo"                # sin tener en cuenta cómo la llame
                                       # el sistema

IPADDR="my.ip.address"                 # su dirección IP
ANYWHERE="any/0"                       # coincide cualquier dirección IP
MY_ISP="my.isp.address.range"          # servidor ISP e intervalo de
                                       # direcciones NOC

LOOPBACK="127.0.0.0/8"                 # intervalo de direcciones
                                       # reservadas de bucle invertido
CLASS_A="10.0.0.0/8"                   # redes privadas de la clase A
CLASS_B="172.16.0.0/12"                # redes privadas de la clase B
CLASS_C="192.168.0.0/16"               # redes privadas de la clase C
CLASS_D_MULTICAST="224.0.0.0/4"        # direcciones de la clase D
                                       # multidifusión
CLASS_E_RESERVED_NET="240.0.0.0/5"     # direcciones de clase E reservadas
BROADCAST_SRC="0.0.0.0"                # direcciones origen de difusión
```

```
BROADCAST_DEST="255.255.255.255"      # direcciones destino de difusión
PRIVPORTS="0:1023"                    # intervalo de puerto privilegiado
                                      # bien conocido
UNPRIVPORTS="1024:65535"              # intervalo de puerto no
                                        privilegiado
```

Las constantes que no se listan aquí se definen dentro del contexto de las reglas específicas en las que se usan.

Cómo quitar cualquier regla que exista previamente

Lo primero que se debe hacer cuando se define un conjunto de reglas de filtrado es eliminar cualquier regla existente de las cadenas. De lo contrario, cualquier regla nueva que defina se agregará al final de las reglas existentes. Los paquetes pueden fácilmente coincidir con una regla preexistente antes de alcanzar el punto en la cadena donde se ha empezado a definir la nueva regla.

Quitar una cadena se llama *eliminar* la cadena. Sin un argumento de dirección que haga referencia a una cadena específica, el siguiente comando vacía las reglas de las tres cadenas internas, input, output y forward, de una vez:

```
# Elimina cualquier regla existente de las cadenas
ipchains -F
```

Las cadenas están vacías. Se parte de cero. El sistema está en su estado predeterminado de directiva para aceptar todo.

Cómo definir la directiva predeterminada

Un efecto secundario que produce la eliminación de todas las reglas es que se devuelve al sistema a su estado predeterminado, incluyendo la directiva predeterminada de aceptar todo para cada cadena. Hasta que se definan nuevas directivas predeterminadas, el sistema permite que pase todo a través de las interfaces de red. No se realiza filtrado.

De forma predeterminada, lo mejor es que el firewall deniegue todo lo que venga y rechace todo lo que sale. A menos que se defina una regla para permitir explícitamente que un paquete coincidente pase por él, los paquetes entrantes simplemente se deniegan sin una notificación para el remitente remoto, y los paquetes salientes se rechazan y se devuelve un mensaje de error ICMP al remitente. La diferencia para el usuario final es que, por ejemplo, si alguien en un sitio remoto intenta conectar con un servidor web, el navegador de dicha persona se colgará hasta que el sistema devuelva una condición de tiempo de espera TCP sobrepasado. El usuario no sabe nada de si existe el sitio o el servidor web. Si, por el contrario, se intenta conectar con

un servidor web remoto, el navegador recibirá una condición de error inmediata indicando que se permite la operación:

```
# Establecer la directiva predeterminada a denegar
ipchains -P input   DENY
ipchains -P output  REJECT
ipchains -P forward REJECT
```

En este momento, todo el trafico de red está bloqueado.

Cómo habilitar la interfaz de bucle invertido

Es necesario habilitar el tráfico de bucle invertido no restringido. Esto permite ejecutar cualquier servicio de red que se elija, o aquellos de los que depende el sistema, sin tener que preocuparse de especificar todas las reglas del firewall.

El bucle invertido se habilita de forma inmediata en la secuencia de comandos del firewall. No es una interfaz disponible para el exterior. Los servicios basados en redes locales, como el sistema X Window, se colgarán hasta que se permita el tráfico de bucle invertido a través de ellos.

Las reglas son sencillas cuando se permite todo. Sólo es necesario deshacer el efecto de las directivas predeterminadas denegar para la interfaz de bucle invertido aceptando todo en dicha interfaz:

```
# Tráfico ilimitado en la interfaz de bucle invertido
ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT
```

Los registros del sistema, el X Window, y los demás servicios locales basados en socket del dominio UNIX vuelven a estar disponibles.

Usurpamiento de dirección origen y otras direcciones incorrectas

Esta sección establece algunos filtros de la cadena input basados en direcciones origen y destino. Estas direcciones nunca se verán en un paquetes entrante legítimo procedente de Internet.

El núcleo de Linux ofrece alguna compatibilidad contra los paquetes entrantes de usurpamiento, además de lo que se debe hacer a nivel de firewall. Además, si no se ha habilitado la protección Cookie SYN de TCP, las siguientes líneas habilitan los dos módulos de compatibilidad del núcleo:

```
echo 1 >/proc/sys/net/ipv4/tcp_syncookies

# Configurar la protección contra usuarpamiento IP
# en Source Address Verification
for f E /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 > $f
done
```

Reglas de directiva predeterminada y la primera regla que coincide gana

Las directivas predeterminadas parecen ser excepciones del escenario de la primera regla que coincida gana. Los comandos de la directiva predeterminada no dependen de la posición. No son reglas, por sí mismos. Una directiva predeterminada de una cadena se aplica después de comparar un paquete con cada regla de la cadena y de que éste no haya coincidido con ninguna de ellas.
Las directivas predeterminadas se definen primero en la secuencia de comandos para definir la disposición predeterminada del paquete antes de que se definan reglas contrarias. Si los comandos de la directiva se ejecutaron al final de la secuencia de comandos, y la secuencia de comandos del firewall contenía un error de sintaxis que provocó que la secuencia de comandos terminara de forma prematura, la directiva predeterminada accept all seguirá funcionando. Si un paquete no coincide con una regla (y las reglas suelen ser reglas aceptar en un firewall de denegar todo de forma predeterminada), el paquete llegará al final de la cadena y se aceptará de forma predeterminada. Las reglas del firewall no realizan nada útil en este caso.

En el nivel de filtrado de paquetes, uno de los pocos casos de usurpamiento de direcciones origen que se puede identificar con certeza es la falsificación de su propia dirección IP. Esta regla deniega los paquetes entrantes que dicen proceder de su dirección:

```
# Rechazar los paquetes usurpados que dicen proceder
# de la dirección IP de la interfaz externa
ipchains -A input  -i $EXTERNAL_INTERFACE -s $IPADDR -j DENY -l
```

No es necesario bloquear los paquetes salientes destinados a usted mismo. No volverán diciendo ser de usted y aparentando ser usurpados. Recuerde que si se envían paquetes a la propia interfaz externa, estos paquetes llegarán a la cola de entrada de la interfaz de bucle invertido, no a la cola de entrada de la interfaz externa. Los paquetes que contienen su dirección como dirección origen nunca llegan a la interfaz externa, incluso aunque envíe paquetes a la interfaz externa.

Como se explica en el Capítulo 2, las direcciones IP privadas que no se usan se colocan aparte en cada uno de los intervalos de las clases A, B y C, para su uso en las LAN privadas. No están destinadas para usarlas en Internet. Los enrutadores no están pensados para enrutar paquetes con direcciones origen privadas. Los enrutadores no pueden enrutar paquetes con direcciones destino privadas. Sin embargo, muchos enrutadores permiten que pasen los paquetes con direcciones origen privadas a través de ellos.

Además, si alguien de la subred del ISP (es decir, en el lado del enrutador compartido del usuario) está filtrando paquetes con direcciones IP destino privadas, se verán incluso si el enrutador no las reenvía. Las máquinas situadas en la LAN propia también pueden filtrar direcciones origen privadas si se ha configurado correctamente el enmascaramiento IP o la configuración del proxy.

Los tres conjuntos de siguientes reglas no permiten los paquetes entrantes y salientes que utilizan cualquiera de las direcciones de red privada de las

clases A, B o C como sus direcciones origen o destino. Ninguno de estos paquetes debe verse fuera de una LAN privada:

```
# Rechazar paquetes que digan proceder o ir a direcciones privadas
# de red de clase A
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_A -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_A -j DENY -l

# Rechazar paquetes que digan proceder o ir a direcciones privadas
# de red de clase B
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_B -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_B -j DENY -l

# Rechazar paquetes que digan proceder o ir a direcciones privadas
# de red de clase C
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_C -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_C -j DENY -l
```

Si la tabla de enrutamiento se ha configurado correctamente, la interfaz de red externa no reconocerá una dirección destino diferente de la suya propia. No obstante, si se ha configurado el enrutamiento automático y se tiene una LAN que usa estas direcciones, y alguien en la subred del ISP estaba filtrando la pérdida de paquetes, el firewall puede reenviar perfectamente los paquetes a la LAN.

Las siguientes reglas rechazan los paquetes con una dirección origen reservada para la interfaz de bucle invertido:

```
# Rechazar paquetes que digan proceder de la interfaz de bucle invertido
ipchains -A input  -i $EXTERNAL_INTERFACE -s $bucle invertido -j DENY-
ipchains -A output -i $EXTERNAL_INTERFACE -s $bucle invertido -j DENY -l
```

Como las direcciones de bucle invertido se asignan a una interfaz de software local, que el software del sistema maneja internamente, cualquier paquete que pretenda proceder de tal dirección es intencionadamente falsa. Observe que se ha escogido registrar el suceso si un usuario local intenta usurpar la dirección.

Igual que las direcciones que se reservan para usarlas en las LAN privadas, los enrutadores no sirven para reenviar paquetes originados desde el intervalo de direcciones de bucle invertido. Un enrutador no puede reenviar un paquete con una dirección destino de bucle invertido.

Las siguientes reglas bloquean los paquetes de difusión que contienen direcciones difusión origen o destino ilegales. La directiva predeterminada del firewall es denegar todo. Por tanto, las direcciones destino de difusión se deniegan de forma predeterminada y deben habilitarse explícitamente en los casos donde sea necesario:

```
# Rechazar paquetes de difusión mal formados
ipchains -A input  -i $EXTERNAL_INTERFACE -s $BROADCAST_DEST -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -d $BROADCAST_SRC  -j DENY -l
```

La primera de estas reglas registra y deniega cada paquete que dice proceder de la dirección 255.255.255.255, la dirección reservada como la dirección destino de difusión. Un paquete nunca será originado legalmente desde la dirección 255.255.255.255.

La segunda de estas reglas registra y deniega cualquier paquete dirigido a la dirección destino 0.0.0.0, la dirección reservada como dirección origen de difusión. Dicho paquete no es un error; es un sondeo específico dirigido a identificar una máquina UNIX que ejecuta software derivado de BSD. Como la mayoría del código de red del sistema operativo UNIX se deriva del de BSD, este sondeo está destinado a identificar máquinas que ejecutan el sistema operativo UNIX.

Las direcciones de multidifusión son legales sólo como direcciones destino. El siguiente par de reglas deniega y registra los paquetes de red multidifusión usurpados:

```
# Rechazar las direcciones de multidifusión de clase D
# que sean sólo ilegales como dirección origen
# La multidifusión usa UDP
ipchains -A input  -i $EXTERNAL_INTERFACE \
         -s $CLASS_D_MULTICAST -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE \
         -s $CLASS_D_MULTICAST -j REJECT -l
```

Los paquetes multidifusión legítimos son siempre paquetes UDP. Como tales, los mensajes multidifusión se envían punto a punto, al igual que cualquier otro mensaje UDP. La diferencia entre paquetes unidifusión y multidifusión es la clase de dirección destino que se usa. La siguiente regla rechaza los paquetes multidifusión salientes desde la máquina del usuario:

```
ipchains -A output -i $EXTERNAL_INTERFACE \
         -d $CLASS_D_MULTICAST -j REJECT -l
```

La funcionalidad de la multidifusión es una opción que se puede configurar cuando se compila el núcleo, y puede inicializar la tarjeta de interfaz de red para reconocer direcciones multidifusión. La funcionalidad está habilitada de forma predeterminada en Red Hat 6.0, pero no en las versiones anteriores. Puede que quiera habilitar estas direcciones si se suscribe a un servicio de conferencia en red que ofrezca difusiones multidifusión de sonido y vídeo.

No se verán direcciones destino multidifusión legítimas a menos que se haya registrado a sí mismo como receptor. Los paquetes multidifusión se envían a múltiples destinos, pero específicos, elegidos de antemano. Sin embargo, se pueden ver paquetes multidifusión que se envían al exterior de las máquinas de la subred local de un ISP. Se pueden bloquear totalmente las di-

recciones de multidifusión si no se suscribe a servicios multidifusión. La siguiente regla rechaza los paquetes multidifusión entrantes:

```
ipchains -A input -i $EXTERNAL_INTERFACE -d $CLASS_D_MULTICAST -j REJECT -l
```

Clarificación del significado de la dirección IP 0.0.0.0

La dirección 0.0.0.0 está reservada para el uso de direcciones origen de difusión. La convención IPFW para especificar una coincidencia de cualquier dirección, any/0.0.0.0 o 0.0.0.0/0.0.0.0, no coincide con la dirección origen de difusión. La razón es que el paquete de difusión tiene activado un bit en el encabezado del paquete que indica que es un paquete de difusión destinado a todas las interfaces de la red, en lugar de ser un paquete unidifusión punto a punto dirigido a un destino particular. Los paquetes de difusión se manejan de forma diferente que los paquetes que no son de difusión. No existe ninguna dirección IP 0.0.0.0 legítima que sea de no difusión.

El registro y el enrutamiento multidifusión son un proceso complicado y que administra un protocolo de control propio de nivel de IP, el protocolo de Internet de administración de grupo (IGMP, Internet Group Management Protocol). Si desea más información sobre la comunicación multidifusión, consulte un excelente documento, "How IP Multicast Works" (Cómo funciona la multidifusión IP), de Vicki Johnson y Marjory Johnson. El documento está disponible en la dirección http://www.ipmulticast.com/community/whitepapers/howipmcworks.html.

Las direcciones IP de la clase D pertenecen al intervalo que empieza en la dirección 224.0.0.0 y llega hasta la dirección 239.255.255.255. La constante CLASS_D_de multidifusión, 224.0.0.0/4, se define para hacer coincidir los primeros cuatro bits de la dirección. Como se muestra en la Figura 3.2, en binario, los valores decimales 224 (11100000B) a 239 (11101111B) son idénticos en cuanto a los primeros cuatro bits (1110B).

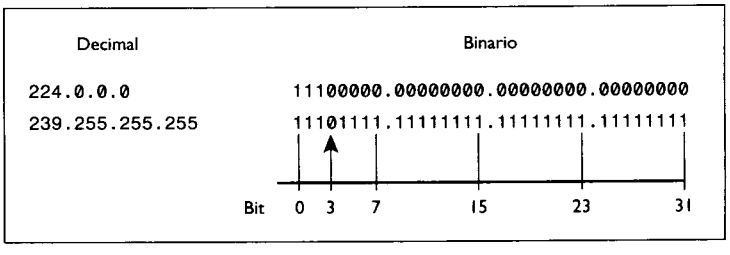

Figura 3.2. Los primeros cuatro bits coincidentes en el intervalo 224.0.0.0/4 de direcciones de multidifusión enmascaradas de clase D.

La siguiente regla de esta sección deniega y registra los paquetes que dicen proceder de la red reservada de clase E:

```
# Rechazar direcciones reservadas IP de clase E
ipchains -A input  -i $EXTERNAL_INTERFACE \
         -s $CLASS_E_RESERVED_NET -j DENY -l
```

Las direcciones IP de clase E pertenecen al intervalo que va desde 240.0.0.0 a 247.255.255.255. La constante cLASS_E_RESERVED_NET, 240.0.0.0/5, se define para hacer coincidir los primeros cinco bits de la dirección. Como se muestra en la Figura 3.3, en binario, los valores decimales 240 (11110000B) a 247 (11110111B) son idénticos en sus cinco primeros bits (1111 0B).

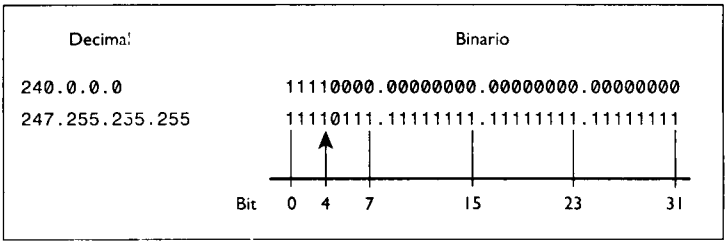

Figura 3.3. Los primeros cinco bits coincidentes en el intervalo 224.0.0.0/5 de direcciones de multidifusión enmascaradas de clase D.

En última instancia, es el IANA quien administra la reserva y el registro del espacio de direcciones IP a nivel mundial. Si busca más información sobre la asignación de direcciones IP, visite la página web http://www.isi.edu/in-notes/iana/assignments/ipv4-address-space. El IANA define algunos bloques de direcciones como reservados. Estas direcciones no deben aparecer en Internet. El conjunto final de reglas deniega esta clase de paquetes potencialmente usurpados:

```
# rechazar direcciones definidas como reservadas por el IANA
# 0.*.*.*, 1.*.*.*, 2.*.*.*, 5.*.*.*, 7.*.*.*, 23.*.*.*, 27.*.*.*
# 31.*.*.*, 37.*.*.*, 39.*.*.*, 41.*.*.*, 42.*.*.*, 58-60.*.*.*

ipchains -A input -i $EXTERNAL_INTERFACE -s 1.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 2.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 5.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 7.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 23.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 27.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 31.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 37.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 39.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 41.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 42.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 58.0.0.0/7 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 60.0.0.0/8 -j DENY -l

# 65: 01000001    - /3 includes 64 - need 65-79 spelled S
ipchains -A input -i $EXTERNAL_INTERFACE -s 65.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 66.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 67.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 68.0.0.0/8 -j DENY -l
```

```
ipchains -A input -i $EXTERNAL_INTERFACE -s 69.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 70.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 71.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 72.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 73.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 74.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 75.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 76.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 77.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 78.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 79.0.0.0/8 -j DENY -l

# 80: 01010000    - /4 masks 80-95
ipchains -A input -i $EXTERNAL_INTERFACE -s 80.0.0.0/4 -j DENY -l

# 96: 01100000  - /4 makses 96-111
ipchains -A input -i $EXTERNAL_INTERFACE -s 96.0.0.0/4 -j DENY -l

# 126: 01111110  - /3 includes 127 - need 112-126 spelled S
ipchains -A input -i $EXTERNAL_INTERFACE -s 112.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 113.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 114.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 115.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 116.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 117.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 118.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 119.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 120.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 121.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 122.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 123.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 124.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 125.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 126.0.0.0/8 -j DENY -l

# 217: 11011001  - /5 includes 216 - need 217-219 spelled S
ipchains -A input -i $EXTERNAL_INTERFACE -s 217.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 218.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 219.0.0.0/8 -j DENY -l

# 223: 11011111  - /6 masks 220-223
ipchains -A input -i $EXTERNAL_INTERFACE -s 220.0.0.0/6 -j DENY -l
```

Cómo filtrar los mensajes de estado y de control ICMP

Los mensajes de control ICMP se generan en respuesta a varias condiciones de error y los provocan los programas de análisis de red, como ping y traceroute. La Tabla 3.2 lista los tipos de mensajes ICMP comunes más interesantes para un sitio pequeño.

Tabla 3.2. Tipos habituales de mensajes ICMP

Tipo numérico	Nombre simbólico	Descripción
0	echo-reply	Una respuesta de ping.
3	destination-unreachable	Un mensaje de estado de error general; un enrutador a lo largo de la trayectoria hasta el destino es incapaz de entregar el paquete al siguiente destino; lo usa traceroute.
4	source-quench	Flujo de nivel de red IP entre dos enrutadores, o entre un enrutador y un host.
5	redirect	Un mensaje de enrutamiento que se devuelve al remitente cuando un enrutador determina que existe una trayectoria más corta.
8	echo-request	Una petición de ping.
11	time-exceeded	Un mensaje de enrutamiento que se devuelve cuando el contador de saltos máximos de un paquete (TTL) se sobrepasa; lo usa traceroute.
12	parameter-problem	Aparecen valores inesperados en el encabezado del paquete IP.

Mensajes de control y de estado de error

Cuatro mensajes ICMP de control y de estado necesitan pasar a través del firewall: Source Quench, Parameter Problem, incoming Destination Unreachable y outgoing Destination Unreachable, subtype Fragmentation Needed. Otros cuatro tipos de mensajes ICMP son opcionales: Echo Request, Echo Reply, other outgoing Destination Unreachable subtypes y Time Exceeded. Los demás tipos de mensajes se pueden ignorar, y hacer que la directiva predeterminada los filtre en el exterior.

De los tipos de mensajes que pueden, o deben, ignorarse, sólo Redirect aparece en la Tabla 3.2, a causa de la función que desempeña en los ataques por denegación de servicio como una bomba de redirección (consulte el Capítulo 2 si busca más información acerca de bombas de redirección.) Al igual que con Redirect, los tipos de mensajes ICMP restantes son mensajes de control y estado destinados al uso entre enrutadores.

Diferencias en los códigos ICMP entre ipchains e ipfwadm

El programa ipchains de la versión Red Hat 6.0 es compatible con el uso del tipo de mensaje numérico ICMP o del nombre simbólico alfabético. Las versiones anteriores que usaban ipfwadm sólo soportaban el tipo de mensaje numérico.
ipchains también es compatible con el uso de los subtipos de mensajes, o códigos. Esto es especialmente útil para un control de filtrado más fino sobre los mensajes destination-unreachable (destino inalcanzable) de tipo 3. Por ejemplo, se pueden rechazar de forma específica los mensajes salientes port-unreachable (puerto inalcanzable) para deshabilitar un traceroute (seguimiento de la ruta) entrante, o permitir de forma específica sólo los mensajes salientes de estado que necesitan fragmentación. ipfwadm no soporta los códigos de subtipo de mensaje.

Para ver una lista de todos los nombres simbólicos ICMP compatibles con ipchains, ejecute ipchains –h icmp. Para ver las asignaciones de la RFC oficial, visite la dirección http://www.isi.edu/in-notes/iana/assignments/icmp-parameters.

Las siguientes secciones describen con más detalle los tipos de mensajes importantes para una máquina host terminal, en contraposición a un enrutador medio.

Mensajes de control Source Quench (Tipo 4)

Los mensajes ICMP tipo 4, Source Quench, se envían cuando un origen de conexión, normalmente un enrutador, está enviando datos a una velocidad mayor que la que el siguiente enrutador destino puede manejar. Source Quench se usa como una forma primitiva de control de flujo a nivel de red IP, normalmente entre dos máquinas punto a punto adyacentes:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 4 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 4 -d $ANYWHERE -j ACCEPT
```

El siguiente salto o máquina destino del enrutador envía un comando Source Quench. El enrutador que originó el mensaje responde enviando paquetes a una velocidad menor, incrementando gradualmente la velocidad hasta que recibe otro mensaje Source Quench.

Mensajes de estado Parameter Problem (Tipo 12)

Los mensajes ICMP tipo 12, Parameter Problem, se envían cuando se recibe un paquete que contiene datos ilegales o inesperados en el encabezado, o cuando la suma de comprobación del encabezado no coincide con la suma de comprobación generada por la máquina receptora:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 12 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 12 -d $ANYWHERE -j ACCEPT
```

Mensajes de error Destination Unreachable (Tipo 3)

El tipo 3 de mensaje ICMP, Destination Unreachable, es un mensaje de estado de error general:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 3 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 3 -d $ANYWHERE -j ACCEPT
```

Los mensajes del encabezado del paquete ICMP para el tipo 3, Destination Unreachable, contienen un campo de código de error que identifica el tipo

particular de error. Idealmente, puede que se quieran eliminar los mensajes salientes de tipo 3. Este tipo de mensaje es el que se envía en respuesta a un intento de un hacker de asignar sus puertos de servicio o su espacio de direcciones. Un atacante puede crear una condición de denegación de servicio obligando al sistema a generar un gran número de estos mensajes bombardeando los puertos UDP que no se usan. Peor aún, un atacante puede usurpar la dirección origen, obligando al sistema a que los envíe a los host usurpados. Por desgracia, el mensaje Destination Unreachable crea una situación Catch-22. Uno de los subtipos de mensajes, Fragmentation Needed, se usa para negociar el tamaño de fragmentación del paquete. El rendimiento de la red puede verse seriamente degradado sin esta negociación.

Si se quiere responder a peticiones entrantes traceroute, deben permitirse los mensajes salientes ICMP Destination Unreachable, con el código de subtipo Port Unreachable.

Mensajes de estado Time Exceeded (Tipo 11)

El mensaje ICMP tipo 11, Time Exceeded, indica una condición de tiempo de espera sobrepasado, o más exactamente, que se ha sobrepasado el máximo número de saltos de un paquete. En las redes actuales, los mensajes entrantes Time Exceeded se reciben principalmente como la respuesta ICMP a una petición saliente traceroute de UDP:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 11 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 11 -d $MY_ISP -j ACCEPT
```

Si se desea responder a las peticiones entrantes traceroute, es necesario permitir los mensajes salientes ICMP Time Exceeded. En las reglas anteriores, sólo se permiten los traceroute procedentes de las máquinas del ISP. Si se quiere usar traceroute por sí mismo, deben permitirse los mensajes entrantes ICMP Time Exceeded. Como la máquina es un enrutador intermedio, no puede dar otro uso a los mensajes Time Exceeded.

Mensajes de control ping Echo Request (Tipo 8) y Echo Reply (Tipo 0)

El programa ping usa dos tipos de mensajes ICMP. El mensaje de petición, Echo Request, es un mensaje tipo 8. El mensaje de contestación, Echo Reply, es un mensaje tipo 0. El programa ping es una sencilla herramienta de análisis de red que data de la DARPANET original. El nombre ping se tomó de la idea del ping sonoro que devolvían los sistemas de sonar (DARPA es, después de todo, la Defense Advanced Research Projects Agency, Agencia de defensa de proyectos avanzados de investigación). Parecido al funcionamiento de un sonar, se difunde un mensaje Echo Request a todas las máquinas en un espa-

cio de direcciones de red y genera mensajes Echo Reply, que devuelve desde todos los *host* que responden en la red.

Sólo para ipchains: cómo usar los códigos de subtipo de mensaje ICMP

Sólo para usuarios de ipchains: se puede reemplazar el conjunto general de reglas compatibles con el programa ipfwadm mencionado anteriormente con reglas más específicas permitiendo cualquier mensaje saliente de tipo 3 hacia el ISP, y los mensajes Fragmentation Needed hacia cualquier dirección:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 3 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 3 -d $MY_ISP -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR fragmentation needed -d $ANYWHERE -j ACCEPT
```

Cómo hacer ping *a host* remotos

El siguiente par de reglas permite hacer un ping a cualquier *host* de Internet:

```
# permitir realizar ping a cualquier sitio
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 8 -d $ANYWHERE -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 0 -d $IPADDR -j ACCEPT
```

ping entrante de *host* remotos

La aproximación que se muestra aquí sólo permite a los *host* externos seleccionados realizar un ping a una dirección:

```
# permitir hacer ping entrantes de host en los que se confía
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $MY_ISP 8 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 0 -d $MY_ISP -j ACCEPT
```

Para conseguir el propósito del ejemplo, los *host* externos que tienen permiso para hacer un ping a la máquina del usuario, son máquinas que pertenecen al mismo ISP. Es probable que el centro de operaciones de red, o el soporte de clientes, quiera hacer ping a la interfaz externa. Las peticiones de eco entrantes que no procedan de los vecinos de la red local se deniegan. El programa ping se usa en diferentes tipos de ataques por denegación de servicio.

Cómo bloquear los ataques smurf entrantes y salientes

Los ataques smurf usan, históricamente, los paquetes ping, difundiendo continuamente mensajes Echo Request a host intermedios con la dirección

origen usurpada como si fuera la dirección IP de la víctima. Como resultado, todas las máquinas de la red del intermediario bombardean continuamente la máquina víctima con mensajes Echo Request, cortando todo el ancho de banda disponible.

Las siguientes reglas registran los ataques smurf. Como los paquetes ICMP de difusión no se permiten explícitamente, la directiva de firewall denegar todo de forma predeterminada siempre elimina estos paquetes. Observe que se deniegan todos los tipos de mensajes ICMP, en lugar de denegar sólo los mensajes Echo Request. Los paquetes ping se suelen utilizar en ataques smurf, pero también se puede utilizar otros tipos de mensajes ICMP. Nunca se será demasiado cuidadoso en un conjunto de reglas de firewall:

```
# ataque smurf
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -d $BROADCAST_DEST -j DENY -l

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -d $BROADCAST_DEST -j REJECT -l
# ataque smurf - máscara de red
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp\
         -d $NETMASK -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp\
         -d $NETMASK -j REJECT -l
# ataque smurf - dirección de red
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp\
         -d $NETWORK -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp\
         -d $NETWORK -j REJECT -l
```

Cómo proteger los servicios en los puertos no privilegiados asignados

Los servicios LAN, en concreto, se ejecutan a menudo sobre puertos no privilegiados. Para servicios basados en el protocolo TCP, un intento de conexión a uno de estos servicios puede distinguirse de una conexión activa que tenga un cliente usando uno de estos puertos no privilegiados a través del estado de los bits SYN y ACK. Por razones de seguridad propia, deberían bloquearse los intentos de conexión entrantes a estos puertos. Es probable que se quieran bloquear los intentos de conexión saliente para protegerse a sí mismo y a otros de errores en su lado, y para registrar los posibles problemas de seguridad interna.

¿Contra qué clases de errores puede necesitar protección? El error más grave es ofrecer servicios peligrosos al exterior, ya sea de forma intencionada o por descuido, lo que se explica en el Capítulo 2. Un error común es ejecutar servicios de red local que se filtran a Internet y molestan a otras personas. Otro error es permitir tráfico saliente dudoso, como las exploraciones de puerto, ya sea generado por accidente o porque alguien lo envíe de forma intencionada

desde la máquina del usuario. Una directiva de firewall predeterminada denegar todo ofrece protección contra casi todos los errores de este tipo.

Una directiva de firewall predeterminada denegar todo permite ejecutar muchos servicios privados detrás del firewall sin correr riesgos. Estos servicios deben permitirse explícitamente a través del firewall para que los clientes remotos puedan tener acceso a ellos. Sin embargo, esta generalización es sólo una aproximación a la realidad. Aunque los servicios TCP sobre puertos privilegiados son bastante seguros contra los hacker, menos para uno con gran nivel y perseverancia, los servicios UDP son inherentemente menos seguros, y algunos servicios se asignan para que se ejecuten sobre puertos no privilegiados. Los servicios RPC, que normalmente se ejecutan sobre UDP, son incluso más problemáticos. Los servicios basados en RPC son exteriores a algún puerto, a menudo un puerto no privilegiado. El demonio portmap asigna entre el número de servicio RPC y el número de puerto real. Un examen de puerto puede mostrar dónde se encuentran estos servicios basados en RPC sin tener que pasar por el demonio portmap.

Ataques smurf

No se debe difundir nada al exterior hacia Internet. La difusión ping, mencionada anteriormente, es el fundamento del ataque por denegación de servicio smurf IP. Consulte el documento CA-98.01.smurf del CERT en la dirección www.cert.org, si desea obtener más información sobre ataques smurf.

Asignaciones oficiales de número de puerto de servicio

Los números de puerto se asignan y se registran en el IANA. La información se basa originalmente en el RFC 1700, "Assigned Numbers (Números asignados)". Dicha RFC es ahora obsoleta. El IANA mantiene información actualizada de forma dinámica en la dirección http://www.isi.edu/in-notes/iana/assignments/port-numbers.

Servicios TCP locales comunes asignados a puertos no privilegiados

Algunos servicios, normalmente los servicios LAN, se ofrecen a través de un puerto no privilegiado, bien conocido y registrado de forma oficial. Además, algunos servicios, como los servicios FTP e IRC, usan protocolos de comunicación complicados que no se prestan al filtrado de paquetes. Las reglas que se describen en las siguientes secciones rechazan los programas cliente locales o remotos que inician una conexión a uno de estos puertos.

FTP es un buen ejemplo de cómo la directiva denegar de forma predeterminada no es siempre suficiente para tratar todos los casos posibles. El protocolo FTP se trata posteriormente en este capítulo. Por ahora, lo importante es que FTP permite conexiones entre dos puertos no privilegiados. Como algunos servicios escuchan en puertos no privilegiados registrados, y las peticiones de conexión entrantes a estos servicios se originan en un

puerto de cliente no privilegiado, las reglas que, por descuido, habilitan el servicio FTP, también permiten conexiones entrantes a estos servicios locales. Esta situación es también un ejemplo de cómo las reglas del firewall son jerárquicas y dependientes del orden. Las reglas que protegen explícitamente un servicio LAN, que se ejecuta sobre un puerto no privilegiado, deben preceder a las reglas de FTP permitiendo el acceso a todo el intervalo de puertos no privilegiados.

Como resultado, algunas de estas reglas parecen ser redundantes, y serán redundantes para algunas personas. Para las personas que ejecutan otros servicios, las siguientes reglas son necesarias para proteger los servicios privados que se ejecutan en puertos no privilegiados.

No permitir las conexiones a Open Window (Puerto TCP 2000)

No deben permitirse las conexiones de cliente salientes a un administrador remoto de Open Window. Especificando el indicador -y, que indica el bit indicador SYN, sólo se rechazan los intentos de establecimiento de conexión realizados desde la máquina del usuario. Otras conexiones activas con programas cliente remotos que estén usando el puerto no privilegiado 2000, no se verán afectadas por la regla, porque los puertos no privilegiados remotos son el punto final de una conexión iniciada por un cliente remoto a un servidor desde la máquina del usuario.

La siguiente regla bloquea los clientes locales que inician una petición de conexión a un administrador remoto de Open Window:

```
OPENWINDOWS_PORT="2000"                    # (TCP) OpenWindows

# Open Windows: intentando establecer una conexión
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
         -s $IPADDR \
         -d $ANYWHERE $OPENWINDOWS_PORT -j REJECT
```

No es necesario bloquear explícitamente las conexiones entrantes al puerto 2000. Linux incluye el administrador de Open Window.

El problema de las exploraciones de puerto

Las exploraciones de puerto no son dañinas por sí mismas. Las generan las herramientas de análisis de red. Actualmente, el problema con las exploraciones de puerto es que las suelen generar personas con intenciones poco menos que honorables. Están "analizando" su red, no la de ellos. Por desgracia, esto hace que los curiosos también parezcan culpables.

No permitir las conexiones a X Window (Puertos TCP 6000:6063)

Las conexiones a los servidores remotos de X Window deben realizarse sobre SSH, que es compatible automáticamente con las conexiones a X Window. Si se especifica el indicador -y, que indica el bit SYN, sólo se re-

chaza el establecimiento de conexión al puerto del servidor remoto. Esto no afecta a otras conexiones iniciadas usando el puerto como un puerto cliente.

La asignación del puerto de X Window empieza en el puerto 6000 con el primer servidor en ejecución. Si existen más servidores en ejecución, se asigna cada uno al siguiente puerto. Si se dispone de un sitio pequeño, es probable que sólo se ejecute un servidor X, por lo que el servidor sólo escuchará el puerto 6000. El puerto 6063 es el puerto más alto que se asigna, permitiendo que se ejecuten 64 administradores independientes de X Window en una sola máquina:

```
XWINDOW_PORTS="6000:6063"                    # (TCP) X Window
```

La primera regla asegura que no se realiza ningún intento de conexión saliente a los administradores de X Window remotos desde la máquina del usuario:

```
# X Window: estableciendo una conexión remota
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
         -s $IPADDR \
         -d $ANYWHERE $XWINDOW_PORTS -j REJECT
```

La siguiente regla registra y bloquea los intentos de conexión entrantes al administrador de X Window del usuario. Esto no afecta a las conexiones locales, ya que las conexiones locales se realizan sobre la interfaz de bucle invertido:

```
# X Window: intento de conexión entrante
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $XWINDOW_PORTS -j DENY -l
```

No permitir conexiones al servidor SOCKS (Puerto TCP 1080)

SOCKS es un servidor proxy local disponible en la dirección http://www.socks.nec.com/. Los programas cliente con capacidad SOCKS se conectan al servidor en vez de conectarse directamente a los servidores remotos. El servidor SOCKS se conecta a los servidores remotos, como un cliente, en su nombre.

Los intentos de conexión a servidores SOCKS remotos son bastante habituales y a menudo implican peligros de intrusión. Las siguientes reglas permiten el uso del puerto 1080 como un puerto local o un puerto cliente remoto, pero rechazan el uso del puerto 1080 como puerto local o puerto de servidor remoto:

```
SOCKS_PORT="1080"                            # socks (TCP)
```

La primera regla asegura que no se realiza ningún intento de conexión saliente a servidores SOCKS remotos desde la máquina del usuario:

```
# SOCKS: estableciendo una conexión
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
         -s $IPADDR \
         -d $ANYWHERE $SOCKS_PORT -j REJECT -l
```

La siguiente regla bloquea los intentos de conexión entrantes a su servidor de SOCKS:

```
# SOCKS: conexión entrante
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR SOCKS_PORT -j DENY -l
```

Servicios UDP locales habituales que se asignan a puertos no privilegiados

Debido al protocolo de establecimiento de conexión de TCP, las reglas del protocolo TCP se pueden manejar de forma más precisa que las reglas del protocolo UDP. Como servicio de datagrama, UDP no tiene un estado de conexión asociado con él. Simplemente se debe bloquear el acceso a los servicios UDP. Las excepciones se realizan de forma explícita para ajustar el servicio de DNS y cualquiera de los pocos servicios de Internet que se basan en UDP y que puede utilizar el usuario. Afortunadamente, los servicios comunes UDP de Internet suelen ser del tipo de los que se usan entre un cliente y un servidor específico. Las reglas de filtrado pueden permitir a menudo los intercambios con un host remoto específico.

El servicio NFS es el principal servicio UDP del que debe preocuparse. El servicio se ejecuta en el puerto no privilegiado 2049. Al contrario que los anteriores servicios basados en TCP, el servicio NFS es principalmente un servicio basado en UDP. Se puede configurar para ejecutarlo como un servicio basado en TCP, pero normalmente no se hace.

No permitir conexiones al servicio NFS (Puerto UDP/TCP 2049)

La primera regla bloquea el puerto UDP 2049 asociado al servicio NFS para cualquier acceso entrante. La regla es innecesaria si no se ejecuta el servicio NFS. No debería ejecutar el servicio NFS en una máquina firewall, pero si lo hace, deniegue el acceso externo:

```
el servicio NFS_PORT="2049"            # (TCP/UDP) el servicio NFS

# el servicio NFS: conexiones UDP
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR $el servicio NFS_PORT -j DENY -l
```

Las dos reglas TCP siguientes tratan el modo de conexión TCP del servicio NFS, que se suele utilizar muy poco. Se bloquean tanto los intentos de es-

tablecimiento de conexión entrantes como salientes, de la misma forma que en las secciones UDP anteriores:

```
# NFS: conexión TCP
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $NFS_PORT -j DENY -l

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $ANYWHERE $NFS_PORT -j DENY -l
```

Las tablas de protocolo de servicio TCP y UDP

El resto del libro se dedica a definir reglas para permitir el acceso a servicios específicos. La comunicación cliente/servidor, tanto para servicios basados en TCP como para los basados en UDP, implica cierta clase de comunicación bidireccional que usa un protocolo específico al servicio. Como tal, las reglas de acceso se representan siempre como un par E/S. El programa cliente realiza una petición y el servidor envía una respuesta. Las reglas para un servicio dado se ordenan por categorías, como reglas cliente o reglas de servidor. La categoría cliente representa la comunicación que necesitan los clientes locales para acceder a servidores remotos. La categoría servidor representa la comunicación que necesitan los clientes remotos para acceder a los servicios que albergan sus máquinas.

Los mensajes de aplicación se encapsulan en mensajes de protocolo de transporte, TCP o UDP. Como cada servicio usa un protocolo de aplicación específico propio, las características particulares del intercambio TCP o UDP son, de alguna manera, únicas del servicio dado.

Las reglas de firewall describen explícitamente el intercambio entre cliente y servidor. Parte del propósito de las reglas de firewall es asegurar la integridad del protocolo a nivel de paquete. Sin embargo, las reglas de firewall, expresadas en la sintaxis ipchains, no son demasiado legibles para el usuario normal. En cada una de las siguientes secciones se presenta el protocolo de servicio a nivel de filtrado de paquetes como una tabla de información de estado, seguido por las reglas ipchains que expresan estos estados.

Cada fila de la tabla lista un tipo de paquete involucrado en el intercambio del servicio. Se define una regla de firewall para cada tipo de paquete individual. La tabla se divide en columnas:

- Descripción contiene una breve descripción que indica si el paquete se origina desde el cliente o desde el servidor y el propósito del mismo.
- Protocolo es el protocolo de transporte que se usa actualmente, TCP o UDP, o el protocolo IP para control de mensajes, ICMP.
- Dirección remota es la dirección legal, o intervalo de direcciones, que el paquete puede contener en el campo de dirección remota.
- Puerto remoto es el puerto legal, o intervalo de puertos, que el paquete puede contener en el campo puerto remoto.
- Entrada/Salida describe la dirección del paquete; es decir, si llega al sistema procedente de una ubicación remota o si sale del sistema hacia una ubicación remota.
- Dirección local es la dirección legal, o intervalo de direcciones, que el paquete puede contener en el campo dirección local.
- Puerto local es el puerto legal, o intervalo de puertos, que el paquete puede contener en el campo puerto local.
- Los paquetes del protocolo TCP contienen una columna final, el indicador TCP, que define los estados legales de SYN-ACK que puede tener.

La tabla describe paquetes tanto entrantes como salientes. Las direcciones y los puertos se describen como remotos o locales, relativos a la interfaz de red de su máquina. Tenga en cuenta que para paquetes entrantes, la dirección remota y el puerto remoto se refieren a los campos origen en el encabezado del paquete IP. La dirección local y el puerto local se refieren a los campos destino del encabezado del paquete IP. Para paquetes salientes, la dirección remota y el puerto remoto se refieren a los campos destino del encabezado del paquete IP. La dirección local y el puerto local se refieren a los campos origen del encabezado del paquete IP.

Por último, en las pocas situaciones donde el protocolo de servicio implica mensajes ICMP, observe que los paquetes ICMP del nivel de red IP no están asociados con el concepto de un puerto origen o destino, como es el caso de los paquetes TCP o UDP de la capa de transporte. En su lugar, los paquetes ICMP usan el concepto de un tipo de mensaje de control o estado. Los mensajes ICMP no se envían a programas enlazados a puertos de servicio particulares. Por el contrario, los paquetes ICMP se envían de un equipo a otro. Por consiguiente, las pocas entradas de paquetes ICMP que se presentan en las tablas usan la columna puerto origen para contener el tipo de mensaje. Para paquetes ICMP entrantes, la columna puerto origen es la columna puerto remoto. Para paquetes ICMP salientes, la columna puerto origen es la columna puerto local.

Cómo habilitar los servicios básicos necesarios de Internet

En realidad sólo son necesarios dos servicios: el servicio de nombres de dominio (DNS, Domain Name Service) y el servicio de identificación de usuario IDENT. DNS traduce entre nombres de host y sus direcciones IP asociadas. No es posible localizar un host remoto sin el servicio DNS. identd proporciona el nombre de usuario o el identificador asociado con una conexión. Esta información suele solicitarla un servidor de correo remoto cuando se envía correo electrónico. No es necesario ofrecer el servicio identd, pero debe contabilizar las peticiones de conexión entrantes de alguna forma para evitar tiempos de espera prolongados.

Cómo permitir el servicio DNS (Puerto UDP/TCP 53)

El servicio DNS usa un protocolo de comunicación que se basa tanto en UDP como en TCP. Los modos de conexión incluyen conexiones regulares cliente a servidor, tráfico de igual a igual entre servidores de reenvío y servidores dedicados, y conexiones de servidor de nombre primario y secundario.

Las solicitudes de bucle invertido se realizan normalmente sobre el protocolo UDP, tanto para búsquedas cliente a servidor como para búsquedas de servidor de igual a igual. La comunicación UDP puede fallar para una búsqueda cliente a servidor si la información que se devuelve es demasiado grande para que quepa en un solo paquete DNS de 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 nuevo intento sobre TCP. La Figura 3.4 muestra la relación entre UDP y TCP durante una búsque-

da de DNS. En la práctica, TCP no suele ser necesario para peticiones. TCP se usa normalmente para transferencias administrativas de zona entre servidores de nombres primarios y secundarios.

Las transferencias de zona son las transferencias de la información completa de un servidor de nombres sobre una red, o la parte (zona) de una red para la que el servidor está autorizado (es decir, es el servidor oficial). Se hace referencia al servidor de nombres autorizado como el servidor de nombres primario. Los servidores de nombres secundarios, o copia de seguridad, solicitan transferencias de zona de forma periódica desde los servidores primarios para mantener actualizada la caché de DNS.

Por ejemplo, uno de los servidores de nombres del ISP es el primario, el servidor autorizado para el espacio de direcciones del ISP. Los ISP suelen tener múltiples servidores DNS para equilibrar la carga, así como por motivos de redundancia de copia de seguridad. Los demás servidores de nombre son servidores de nombres secundarios, que actualizan su información desde la copia maestra del servidor primario

Las transferencias de zona quedan fuera del alcance de este libro. Es probable que un sistema pequeño no sea un servidor de nombres autorizado para un espacio de nombres de dominio público, ni probablemente un servidor de copia de seguridad público para dicha información.

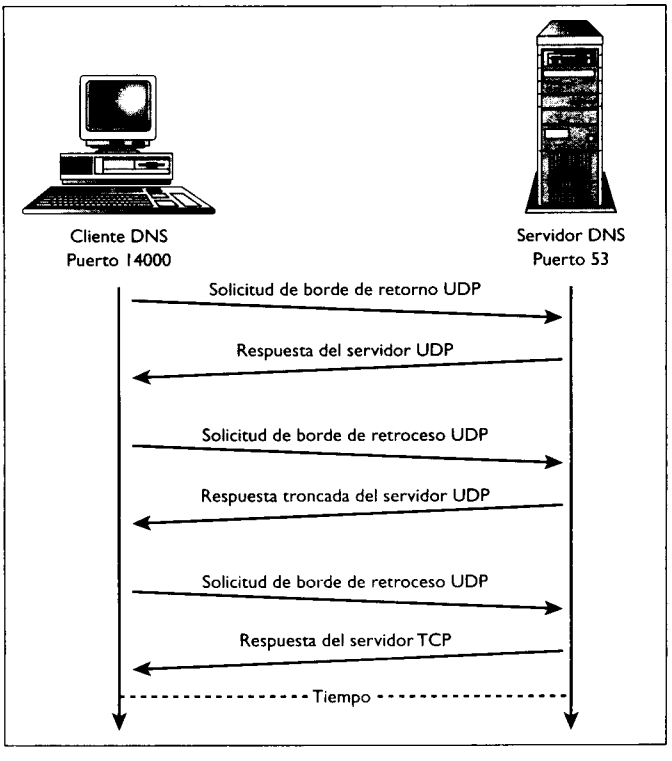

Figura 3.4. Búsqueda DNS cliente a servidor.

La Tabla 3.3 lista el protocolo DNS completo para las reglas de cuentas del firewall.

Tabla 3.3. Protocolo DNS

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	UDP	NAMESERVER	53	S	IPADR	1024:65535	-
Respuesta del servidor remoto	UDP	NAMESERVER	53	E	IPADR	1024:65535	-
Petición del cliente local	TCP	NAMESERVER	53	S	IPADR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	NAMESERVER	53	E	IPADR	1024:65535	ACK
Petición del servidor local	UDP	NAMESERVER	53	S	IPADR	53	-
Repuesta del servidor remoto	UDP	NAMESERVER	53	E	IPADR	53	-
Petición de transferencia de zona local	TCP	Primario	53	S	IPADR	1024:65535	Cualquiera
Petición de transferencia de zona remota	TCP	Primario	53	E	IPADR	1024:65535	ACK
Petición del cliente remoto	UDP	Cliente DNS	1024:65535	E	IPADR.	53	-
Respuesta del servidor local	UDP	Cliente DNS	1024:65535	S	IPADR	53	-
Petición del cliente remoto	TCP	Cliente DNS	1024:65535	E	IPADR	53	Cualquiera
Respuesta del servidor local	TCP	Cliente DNS	1024:65535	S	IPADR	53	ACK
Petición del cliente remoto	UDP	Cliente DNS	53	E	IPADR	53	-
Respuesta del servidor local	UDP	Cliente DNS	53	S	IPADR	53	-
Petición de transferencia de zona remota	TCP	Secundario	1024:65535	E	IPADR	53	Cualquiera
Respuesta de transferencia de zona local	TCP	Secundario	1024:65535	S	IPADR	53	ACK

Cómo permitir búsquedas DNS como cliente

El cliente que resuelve DNS no es un programa específico. El cliente está incluido en el código de biblioteca de red que se compila en programas de red. Cuando un nombre de host debe realizar una búsqueda, el equipo que resuelve direcciones solicita la búsqueda desde un servidor named. Muchos sistemas pequeños sólo están configurados como clientes del servicio DNS. El servidor se ejecuta en una máquina remota. Para un usuario particular, el servidor de nombres suele ser una máquina propiedad del ISP.

DNS envía una petición de búsqueda en forma de datagrama UDP:

```
NAMESERVER ="mi.nombre.servidor"          # (TCP/UDP) DNS

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NAMESERVER 53 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Si se produce un error porque los datos que se devuelven son demasiado grandes y no caben en un datagrama UDP, DNS vuelve a probar usando una conexión TCP.

Las siguientes reglas se incluyen para la extraña ocasión en que la respuesta de búsqueda no quepa en un datagrama UDP de DNS. Esto no suele ocurrir en operaciones cotidianas. Se puede ejecutar el sistema sin problemas durante meses, sin necesidad de usar las reglas TCP. Por desgracia, de vez en cuando, quizá una o dos veces al año, las búsquedas de DNS se bloquean sin estas reglas debido a un servidor DNS remoto mal configurado. Suele ser más normal que estas reglas las use un servidor de nombres secundario que solicita una transferencia de zona desde el servidor de nombres primario:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d <mi.dns.primario> 53 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s <mi.dns.primario> 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo permitir búsquedas DNS como servidor de igual a igual

Las transacciones de igual a igual son intercambios entre dos servidores. En el caso del servicio DNS, cuando el servidor de nombres local no dispone de la información solicitada localmente por un cliente, éste contacta con un servidor remoto y reenvía la petición.

La configuración de un servidor de nombres de reenvío local puede suponer una importante ganancia en cuanto a rendimiento. Como se muestra en la Figura 3.5, cuando se configura named como un servidor de nombres de reenvío y de caché, funciona tanto como servidor local como cliente de

un servidor DNS remoto. La diferencia entre un intercambio directo cliente a servidor y un intercambio servidor local reenviado a un servidor remoto (de igual a igual) está en los puertos origen y destino que se usan. En vez de iniciar un intercambio desde un puerto no privilegiado, named inicia el intercambio desde su propio puerto, el 53. Una segunda diferencia es que las búsquedas de servidor de igual a igual de este tipo siempre se realizan sobre UDP.

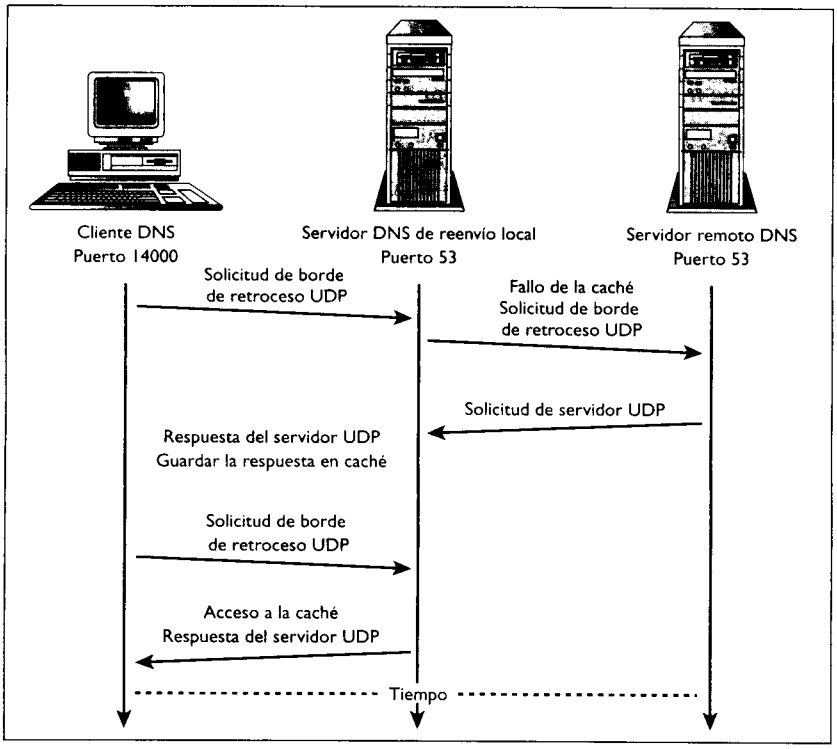

Figura 3.5. Un servidor DNS de reenvío y una búsqueda de igual a igual.

Las peticiones locales de cliente se envían al servidor DNS local. La primera vez, named no tendrá la información de la búsqueda, por lo que reenvía la petición a un servidor de nombres remoto. named guarda en la caché la información que se devuelve y la pasa al cliente. La próxima vez que se solicite la información, named la encuentra en la caché local y no solicita una petición remota:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d $NAMESERVER 53 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER 53 \
         -d $IPADDR 53 -j ACCEPT
```

Cómo permitir búsquedas DNS remotas al servidor

Un sitio normal de un usuario tiene pocas razones para ofrecer el servicio DNS a máquinas remotas. A menos que el sitio sea un ISP, los clientes serán máquinas locales de la LAN, la subred, el enrutador, el espacio de direcciones de red o cualquier término que se utilice para el tamaño de la red y la organización que se busca.

Si es un particular o un pequeño negocio que ofrece el servicio DNS al mundo exterior, es conveniente restringir los clientes a un grupo selecto. No se deberían permitir conexiones de cualquier lugar:

```
# transacción DNS cliente/servidor
ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s <mis.clientes.dns> $UNPRIVPORTS \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d <mis.clientes.dns> $UNPRIVPORTS -j ACCEPT

# transacción DNS de igual a igual
ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s <mis.clientes.dns> 53 \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d <mis.clientes.dns> 53 -j ACCEPT
```

Las siguientes reglas se aplican a los reintentos de petición de cliente cuando los datos son demasiado largos y no se pueden colocar en un paquete DNS de UDP. También se aplican a un servidor de nombres secundario que solicita transferencias de zona desde un servidor de nombre primario. El protocolo TCP se utiliza casi exclusivamente para transferencias de zona, y las transferencias de zona representan posibles agujeros de seguridad. Si se utiliza, es importante restringir las direcciones origen de cliente:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <mis.dns.secundarios> $UNPRIVPORTS \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 53 \
         -d <mis.dns.secundarios> $UNPRIVPORTS -j ACCEPT
```

Filtrado del servicio de identificación de usuario AUTH (Puerto TCP 113)

El servicio de identificación de usuario AUTH o IDENTD se suele utilizar cuando se envía correo o se envía un artículo de Usenet. Algunos sitios FTP también se configuran para solicitar una búsqueda AUTH que se pueda resolver. Con fines de registro, el servidor devuelve una petición a la máquina

del usuario para conseguir el nombre de la cuenta del usuario que inició la conexión de correo o de noticias. La Tabla 3.4 lista el protocolo completo de conexión cliente/servidor para el servicio AUTH.

Transferencias de zona sobre TCP

Los servicios de red a gran escala, como las transferencias de zona DNS, no deben permitirse en pequeños sitios. Indudablemente, puede haber excepciones. Para las excepciones, y para las personas que "van a hacerlo de todas formas", restrinja la lista de los sitios secundarios desde los que se aceptan conexiones. Comparta las tablas DNS sólo con sitios remotos seguros.

Tabla 3.4. Protocolo identd

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	113	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	113	E	IPADDR	1024:65535	ACK
Petición de cliente remoto	TCP	ANYWHERE	1024:65535	E	IPADDR	113	Cualquiera
Respuesta del servidor local	TCP	ANYWHERE	1024:65535	S	IPADDR	113	ACK

Cómo permitir que las peticiones AUTH salientes actúen como cliente

Una máquina actuará como cliente AUTH si se ejecuta un servidor de correo o de FTP. No hay ninguna razón para no permitir a los sistemas ser un cliente AUTH:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 113 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 113 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo filtrar peticiones AUTH entrantes a un servidor

Ofrecer los servicios AUTH es el tema actual de debate. Parece que no existen argumentos convincentes para hacer caso a ninguna de las partes, salvo que un número creciente de sitios FTP lo necesiten, y AUTH proporciona información de la cuenta de usuario. Tanto si se ofrece el servicio como si no, se recibirán peticiones entrantes para el servicio cada vez que se envíe correo.

Si se ejecuta el servidor identd sin el archivo /etc/inetd.conf, las siguientes reglas habilitan las peticiones de conexión identd entrantes:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 113 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 113 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Si se decide no ofrecer el servicio, no será posible denegar las peticiones entrantes. El resultado será una larga espera cada vez que se intente enviar correo o un artículo de Usenet. Nadie notificará al cliente de correo que el correo o el artículo se ha recibido para entregarlo, hasta que la petición identd sobrepase el tiempo de respuesta. Por el contrario, se debe rechazar la petición de conexión para evitar esperar el tiempo de respuesta de la conexión TCP. Este es el único caso, de los ejemplos estudiados, en que se rechaza un paquete entrante en vez de denegarse:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE \
         -d $IPADDR 113 -j REJECT
```

Cómo habilitar servicios TCP habituales

Es posible que nadie quiera habilitar todos los servicios listados en esta sección, pero sí habilitar algún subconjunto de los mismos. Estos son los servicios que más se utilizan sobre Internet en la actualidad. Como tales, esta sección es más una sección de referencia que otra cosa. Proporciona reglas para:

- Correo electrónico
- Usenet
- telnet
- ssh
- ftp
- Servicios web
- finger
- whois
- Servicio de información gopher
- Servicio de información de área extensa (WAIS, Wide Area Information Service)

Existen otros muchos servicios disponibles que no se explican aquí. Algunos de ellos se utilizan en servidores especializados, algunos por grandes empresas y organizaciones y otros están diseñados para usarlos en redes privadas locales.

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

El correo electrónico es un servicio que casi todo el mundo solicita. La forma de configurar el correo depende del ISP, del tipo de conexión y de las elecciones particulares que haga el usuario. El correo electrónico se envía a través de la red usando el protocolo SMTP asignado al puerto de servicio TCP 25. El correo electrónico se suele recibir de forma local a través de uno de los tres protocolos diferentes, SMTP, POP o IMAP, dependiendo de los servicios que proporcione el ISP y de la configuración local.

SMTP es el protocolo de correo general. El correo se entrega a la máquina host destino. El servidor de correo destino determina si se puede entregar el correo (si está dirigido a una cuenta de usuario válida en la máquina) y la entrega al buzón de correo local del usuario.

POP e IMAP son servicios de obtención de correo. POP se ejecuta en el puerto TCP 110. IMAP se ejecuta en el puerto TCP 143. Los ISP suelen poner a disposición el correo entrante a sus clientes usando uno de estos dos servicios. Ambos servicios son autenticados. Se asocian con la cuenta y la contraseña del cliente del ISP. Por lo que respecta a la entrega de correo, la diferencia entre SMTP y POP o IMAP es que SMTP recibe el correo entrante y lo encola en el buzón de correo del usuario. POP e IMAP entregan el correo en el programa de correo local del usuario desde el ISP del usuario, donde el correo se ha colocado en cola de forma remota en el buzón de correo del usuario del ISP. La Tabla 3.5 lista los protocolos de conexión completos cliente/servidor para SMTP, POP e IMAP.

Tabla 3.5. Protocolos de correo SMTP, POP e IMAP

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Enviar correo saliente	TCP	ANYWHERE	25	S	IPADDR	1024:65535	Cualquiera
Respuesta servidor remoto	TCP	ANYWHERE	25	E	IPADDR	1024:65535	ACK
Recibir correo entrante	TCP	ANYWHERE	1024:65535	E	IPADDR	25	Cualquiera
Respuesta del servidor local	TCP	ANYWHERE	1024:65535	S	IPADDR	25	ACK
Repuesta de cliente local	TCP	SERVIDOR POP	110	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	SERVIDOR POP	110	E	IPADDR	1024:65535	ACK
Petición de cliente remoto	TCP	CLIENTE POP	110	E	IPADDR	1024:65535	Cualquiera
Respuesta del servidor local	TCP	CLIENTE POP	110	S	IPADDR	1024:65535	ACK

Tabla 3.5. Protocolos de correo SMTP, POP e IMAP *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	SERVIDOR IMAP	143	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	SERVIDOR IMAP	143	E	IPADDR	1024:65535	ACK
Petición de cliente remoto	TCP	CLIENTE IMAP	143	E	IPADDR	1024:65535	Cualquiera
Respuesta del servidor local	TCP	CLIENTE IMAP	143	S	IPADDR	1024:65535	ACK

Cómo enviar correo sobre SMTP (Puerto TCP 25)

El correo se envía sobre SMTP. Pero ¿qué servidor SMTP se usa para recoger el correo y enviarlo? Los ISP ofrecen servicio de correo SMTP a sus clientes. El servidor de correo del ISP actúa como pasarela de correo. Sabe cómo recoger el correo, encuentra el host receptor y envía el correo. Si lo desea, con UNIX puede albergar un servidor propio. El servidor será responsable de enrutar el correo al destino.

Cómo transmitir correo saliente a través de un servidor STMP de pasarela (ISP) externo

Cuando se transmite el correo saliente a través de un servidor SMTP de pasarela externo, el programa de correo cliente envía todo el correo saliente al servidor de correo del ISP. El ISP actúa como pasarela de correo para el resto del mundo. El sistema del usuario no necesita saber cómo localizar los destinos o las trayectorias a ellos. La pasarela de correo del ISP realiza las funciones de estación transmisora.

Las siguientes reglas permiten transmitir correo a través de la pasarela SMTP del ISP:

```
SMTP_GATEWAY="mi.isp.servidor"      # servidor externo de correo
                                    # o transmisión

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $SMTP_GATEWAY 25 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $SMTP_GATEWAY 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo enviar correo a cualquier servidor de correo externo

También se puede omitir el servidor de correo del ISP y albergar uno propio. El servidor local será el responsable de recoger el correo saliente, realizar la búsqueda DNS del nombre de host destino y transmitir el correo al des-

tino. El programa de correo cliente apunta al servidor SMTP local en vez de al servidor del ISP.

Las siguientes reglas permiten enviar correo directamente a destinos remotos:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 25 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Servidores proxy tanto cliente como servidor

El servidor de correo actual de SMTP es sendmail, que es un servidor proxy. Actúa como servidor para el programa cliente de correo. Actúa como cliente para el servidor remoto al que se envía el correo. Los términos cliente y servidor se pueden confundir en este contexto. sendmail actúa como ambos, dependiendo del programa con el que se comunique.

Cómo recibir correo

La forma en que se recibe el correo depende de la situación. Si se ejecuta un servidor propio de correo local, se puede recoger el correo entrante directamente en la máquina Linux. Si se recibe el correo desde la cuenta del ISP, se podrá recoger o no el correo como cliente POP o IMAP, dependiendo de cómo se haya configurado la cuenta de correo electrónico del ISP y dependiendo de los servicios de entrega de correo que ofrece el ISP.

Obtención de correo como servidor SMTP local (Puerto TCP 25)

Si se desea recibir el correo directamente que se envía directamente a las máquinas locales desde cualquier lugar en el mundo, es necesario ejecutar sendmail y usar estas reglas de servidor:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 25 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 25 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Si se prefiere mantener la cuenta de correo electrónico local como dirección pública, se pueden configurar las cuentas de trabajo y de correo electrónico del ISP para que reenvíen el correo desde el servidor local. En este caso, es necesario reemplazar las dos reglas anteriores, para que acepten conexiones desde cualquier lugar, con reglas específicas separadas para cada reenviador de correo.

Cómo recibir correo como cliente POP (Puerto TCP 110)

La conexión a un servidor POP es una forma muy habitual de recoger el correo desde un ISP remoto o desde una cuenta de trabajo. Si el ISP usa un servidor para la entrega del correo del cliente, debe permitir las conexiones salientes cliente a servidor.

La dirección del servidor será un nombre de host o una dirección específicos, en vez del indicador global ANYWHERE. Las cuentas POP son cuentas de usuario asociadas con un usuario y una contraseña específicos:

```
POP_SERVER="mi.isp.pop.servidor"          # servidor pop externo,
                                          # si existe

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $POP_SERVER 110 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $POP_SERVER 110 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Obtención de correo como cliente IMAP (Puerto TCP 143)

La conexión a un servidor IMAP es otra forma habitual de recibir correo desde un ISP remoto o desde una cuenta de trabajo remota. Si el ISP usa un servidor IMAP para recibir el correo del cliente, debe permitir las conexiones salientes cliente a servidor.

La dirección del servidor será una dirección o un nombre de host específicos, en vez del indicador global ANYWHERE. Las cuentas IMAP son cuentas de usuario asociadas con un usuario y una contraseña específicos:

```
IMAP_SERVER="mi.servidor.imap"          # servidor imap externo,
                                        # si existe

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $IMAP_SERVER 143 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IMAP_SERVER 143 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Ejemplos reales de combinaciones de correo electrónico de servidor y cliente

En esta sección se describen cuatro aproximaciones comunes a las combinaciones de correo electrónico entre cliente y servidor:

- Enviar correo como cliente SMTP y recibir el correo como cliente POP.
- Enviar correo como cliente SMTP y recibir correo como cliente IMAP.
- Enviar correo como cliente SMTP y recibir correo como servidor SMTP.
- Enviar correo como servidor SMTP y recibir correo como un servidor SMTP.

Las dos primeras combinaciones son útiles si se usan los servicios de correo electrónico SMTP o IMAP del ISP. El tercer ejemplo es una aproximación mixta, donde se transmite el correo a través del servidor de correo SMTP del ISP, pero se recibe directamente a través del servidor SMTP local. La cuarta aproximación es totalmente compatible con la ejecución de un servidor propio independiente de correo, tanto para correo entrante como saliente.

Cómo enviar correo como cliente SMTP y recibir correo como cliente POP

Si se envía correo como cliente SMTP y se recibe correo como cliente POP, se basa completamente en un sitio remoto para los servicios de correo. El sitio remoto alberga tanto el servidor SMTP para transmitir el correo saliente, como el servidor POP para obtener el correo local:

```
SMTP_GATEWAY="mi.isp.servidor"          # transmisión o servidor
                                        # de correo externo

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $SMTP_GATEWAY 25 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $SMTP_GATEWAY 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

POP_SERVER="mi.isp.pop.servidor"        # servidor pop externo, si
                                        # existe

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $POP_SERVER 110 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $POP_SERVER 110 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo enviar correo como cliente SMTP y recibir correo como cliente IMAP

Si se envía correo como cliente SMTP y se recibe correo como cliente IMAP, se basa completamente en un sitio remoto para los servicios de correo. El sitio remoto alberga tanto el servidor SMTP para entregar el correo saliente, como el servidor IMAP para obtener el correo local:

```
SMTP_GATEWAY="mi.isp.servidor"          # transmisión o servidor de
                                        # correo externo

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $SMTP_GATEWAY 25 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $SMTP_GATEWAY 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

```
IMAP_SERVER="mi.servidor.imap"        # servidor imap externo, si
                                      # existe

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $IMAP_SERVER 143 -j ACCEPT
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IMAP_SERVER 143 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo enviar correo como cliente SMTP y recibir correo como servidor SMTP

Si se envía correo como cliente SMTP y se recibe correo como cliente SMTP, se basa completamente en un sitio remoto para ofrecer servicios SMTP para reenviar el correo saliente a destinos remotos. Es necesario ejecutar sendmail localmente como servidor SMTP local que permite a los host remotos enviar correo directamente a la máquina del usuario. El correo saliente se enviará a través del ISP, pero el demonio local sendmail sabe cómo entregar el correo entrante a las cuentas de usuario locales:

```
SMTP_GATEWAY="mi.isp.servidor"          # transmisión o servidor
                                        # de correo externo

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $SMTP_GATEWAY 25 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $SMTP_GATEWAY 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 25 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 25 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Cómo enviar correo como servidor SMTP y recibir correo como servidor SMTP

Si se envía correo como cliente SMTP y se recibe correo como servidor SMTP, se basa completamente en servicios propios de correo. El demonio local sendmail se configura para entregar el correo saliente a los *host* destino por sí mismo, al igual que para recoger y entregar el correo entrante:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 25 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 25 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 25 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Cómo albergar un servidor de correo para clientes remotos

Es poco usual que un sistema pequeño albergue servicios públicos POP o IMAP. Puede que se quiera hacer si se ofrecen servicios de correo remoto a unos cuantos amigos, por ejemplo, o si el servicio de correo del ISP estaba temporalmente inoperativo. En cualquier caso, es importante restringir los clientes de los que el sistema aceptará conexiones, tanto a nivel de filtrado de paquetes como a nivel de configuración del servidor. También debe considerarse el uso de un método de autenticación cifrado o permitir la obtención de correo sobre una conexión SSH.

Cómo albergar un servidor POP para clientes remotos

Los servidores POP son uno de los tres puntos de entrada más comunes y que se han utilizado con más éxito para todo tipo de piratería informática.

Si suele usar un sistema local como servidor de correo central y ejecutar un servidor popd local para proporcionar acceso de correo a máquinas locales sobre una LAN, no necesita las reglas de servidor de este ejemplo. Debe denegar las conexiones entrantes desde Internet. Si necesita albergar el servicio POP para un número limitado de individuos remotos, las siguientes reglas permiten las conexiones entrantes a su servidor POP. Las conexiones se restringen a las direcciones IP de clientes específicos:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <mi.pop.clientes> $UNPRIVPORTS \
         -d $IPADDR 110 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 110 \
         -d <mi.pop.clientes> $UNPRIVPORTS -j ACCEPT
```

Cómo albergar un servidor IMAP para clientes remotos

Los servidores IMAP son uno de los tres puntos de entrada más comunes y que se han utilizado con más éxito para todo tipo de piratería informática.

Si suele usar un sistema local como servidor de correo central y ejecutar un servidor imapd local para proporcionar acceso de correo a máquinas locales en una LAN, no necesita una regla de servidor. Debe denegar las conexiones entrantes procedentes de Internet. Si necesita albergar el servicio IMAP para un número limitado de individuos remotos, las siguientes reglas permiten las conexiones entrantes al servidor IMAP. Las conexiones se restringen a las direcciones IP de clientes específicos:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <mi.imap.clientes> $UNPRIVPORTS \
         -d $IPADDR 143 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 143 \
         -d <mi.imap.clientes> $UNPRIVPORTS -j ACCEPT
```

Cómo acceder a los servicios de noticias de Usenet (Puerto TCP NNTP 119)

Se accede a las noticias de Usenet sobre NNTP, que se ejecuta en la cima de TCP, a través del puerto de servicio 119. La lectura de noticias y el envío de artículos lo controla el cliente de noticias local. Pocos sistemas requieren las reglas de servidor. La Tabla 3.6 lista el protocolo completo de conexión cliente/servidor para el servicio de noticias de Usenet NNTP.

Tabla 3.6. Protocolo NNTP

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	NEWS SERVER	119	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	NEWS SERVER	119	E	IPADDR	1024:65535	ACK
Petición de cliente remoto	TCP	Clientes TNP	1024:65535	E	IPADDR	119	Cualquiera
Respuesta del servidor local	TCP	Clientes TNP	1024:65535	S	IPADDR	119	ACK
Petición del servidor local	TCP	Noticias	119	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	Noticias	119	E	IPADDR	1024:65535	ACK

Cómo leer y enviar un artículo como cliente de Usenet

Las reglas de cliente permiten conexiones al servidor de noticias del ISP. El manejo tanto de la lectura de noticias como del envío de artículos se controla con estas reglas:

```
NEWS_SERVER="my.news.server"          # servidor externo de
                                      # noticias, si existe

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NEWS_SERVER 119 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $NEWS_SERVER 119 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo albergar un servidor de noticias de Usenet para clientes remotos

Es improbable que un sitio pequeño albergue un servidor de noticias para uso ajeno. Incluso es improbable que albergue un servidor de noticias local. En pocas ocasiones será necesario configurar las reglas de servidor para permitir las conexiones entrantes sólo desde un conjunto seleccionado de clientes:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <mis.clientes.noticias> $UNPRIVPORTS \
         -d $IPADDR 119 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 119 \
         -d <mis.clientes.noticias> $UNPRIVPORTS -j ACCEPT
```

Cómo permitir que el principal reciba noticias para un servidor de Usenet local

Es improbable que un pequeño sitio particular tenga una relación de igual a igual de servidor de noticias con un ISP. Aunque los servidores de noticias solían ser bastante accesibles para Internet, pocos servidores de noticias siguen disponibles debido al SPAM y a los problemas relacionados con la carga del servidor.

Si el sitio es lo suficientemente grande o lo suficientemente generoso como para albergar un servidor de Usenet general, debe conseguir las noticias de otro lugar. Las siguientes reglas permiten al servidor de noticias local recibir noticias desde un servidor remoto. El servidor local contacta con el servidor remoto como un cliente. La única diferencia entre las reglas de noticias de igual a igual y las reglas de cliente habituales es el nombre o la dirección del host remoto:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d <my.news.feed> 119 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s <my.news.feed> 119 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

telnet (Puerto TCP 23)

telnet ha sido el medio estándar de facto para iniciar sesiones sobre Internet durante muchos años. Conforme ha cambiado la naturaleza de la comunidad de Internet, telnet se ha empezado a ver cada vez más como un servicio inseguro, porque se comunica en texto ASCII no cifrado. Sin embargo, telnet puede ser la única herramienta que tenga disponible para las conexiones remotas, dependiendo de las opciones de conexión disponibles en el

otro extremo. Si tiene la posibilidad, debe usar siempre un servicio cifrado, como ssh, en lugar de telnet.

Las reglas de cliente y de servidor permiten el acceso a y desde cualquier lugar. Si se usa telnet, casi con toda seguridad se podrán restringir las direcciones externas a un subconjunto selecto a nivel de filtrado de paquetes. La Tabla 3.7 lista el protocolo completo de conexión cliente/servidor para el servicio telnet.

Tabla 3.7. Protocolo TELNET

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	23	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	23	E	IPADDR	1024:65535	ACK
Petición de cliente remoto	TCP	Clientes Telnet	1024:65535	E	IPADDR	23	Cualquiera
Respuesta del servidor local	TCP	Clientes Telnet	1024:65535	S	IPADDR	23	ACK

Cómo permitir el acceso del cliente saliente a sitios remotos

Si es necesario usar telnet para acceder a cuentas en sistemas remotos, las siguientes reglas permiten las conexiones salientes a sitios remotos. Si el sitio tiene usuarios múltiples, se pueden restringir las conexiones salientes a los sitios específicos donde los usuarios tienen cuentas, en vez de permitir conexiones salientes a cualquier sitio:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 23 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 23 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo permitir el acceso de entrada al servidor local

Incluso aunque sea necesario acceder como cliente a servidores remotos, puede que no sea necesario permitir conexiones entrantes al servidor Telnet. Si lo hace, las siguientes reglas permiten las conexiones entrantes al servidor:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 23 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 23 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

En lugar de permitir conexiones procedentes de cualquier lugar, es preferible definir reglas de servidor para cada host o red específicos desde los que puede originarse una conexión entrante de forma legítima.

ssh (Puerto TCP 22)

SSH, o shell seguro, no se incluye en las versiones Linux debido a las limitaciones en la exportación de tecnología de cifrado, pero está disponible de forma gratuita en sitios de descarga de software en Internet. Es preferible usar SSH que telnet para iniciar sesiones remotas, pues ambos extremos de la conexión usan claves de autenticación, tanto para el *host* como para los usuario,s y cifran los datos. Además, SSH es algo más que un servicio de inicio de sesión remoto. Puede dirigir automáticamente conexiones de X Window entre sitios remotos, y FTP y otras conexiones basadas en TCP pueden dirigirse sobre la conexión más segura SSH. Como los otros extremos de la conexión permiten conexiones SSH, es posible enrutar todas las conexiones TCP a través del firewall usando SSH. Como tal, SSH es como la red privada virtual (VPN) de un pobre hombre.

Los puertos que usa SSH permiten configurarlos de muchas formas. Las reglas de este ejemplo se aplican al uso del puerto SSH. Las conexiones se inician, de forma predeterminada, entre un puerto no privilegiado de cliente y el puerto 22 asignado al servicio del servidor. El servidor crea una copia de sí mismo para la conexión, y luego se vuelve a asignar el extremo cliente de la conexión a un puerto privilegiado en el intervalo descendente que va de 1023 a 513 para poder soportar la autenticación de .rhosts y hosts.equiv. Se utiliza el primer puerto disponible. El cliente SSH usará opcionalmente los puertos no privilegiados exclusivamente. El servidor SSH aceptará conexiones desde los puertos privilegiados, o bien desde los puertos no privilegiados.

Las reglas de cliente y de servidor permiten el acceso a y desde cualquier lugar. En la práctica, restringirá las direcciones externas a un subconjunto selecto, particularmente porque ambos extremos de la conexión deben configurarse para la autenticación. La Tabla 3.8 lista el protocolo de conexión completo cliente/servidor para el servicio SSH.

Tabla 3.8. Protocolo SSH

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	22	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	22	E	IPADDR	1024:65535	ACK
Petición del cliente local	TCP	ANYWHERE	22	S	IPADDR	513:1023	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	22	E	IPADDR	513:1023	ACK

Tabla 3.8. Protocolo SSH *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición de cliente remoto	TCP	clientes SSH	1024:65535	E	IPADDR	22	Cualquiera
Respuesta del servidor local	TCP	clientes SSH	1024:65535	S	IPADDR	22	ACK
Petición de cliente remoto	TCP	clientes SSH	513:1023	E	IPADDR	22	Cualquiera
Respuesta del servidor local	TCP	clientes SSH	513:1023	S	IPADDR	22	ACK

Autenticación SSH, tcp_wrappers y rhost

No se puede iniciar directamente SSH con tcp_wrappers, pero se puede compilar para que cumpla la información de la lista de acceso en los archivos /etc/hosts.allow y /etc/hosts.deny.
La autenticación de .rshosts y hosts.equiv no debe estar disponible en una máquina firewall.
Las herramientas de análisis de seguridad del sistema, que se explican en el Capítulo 8, "Detección de intrusos e informe de incidentes", avisan de que éstos archivos existen en el sistema.
Para obtener más información sobre SSH, consulte la dirección http://www.ssh.fi/.

Cuando se selecciona un puerto de servidor privilegiado para la conexión actual, se usa el primer puerto libre entre los números 1023 y 513. El intervalo de los puertos que se permiten es equivalente al numero permitido de conexiones entrantes SSH simultaneas:

```
SSH_PORTS="1020:1023"          # (TCP) 4 conexiones simultáneas
```

Cómo permitir el acceso de cliente a servidores SSH remotos

Estas reglas permiten conectarse a sitios remotos usando ssh:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $SSH_PORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $IPADDR $SSH_PORTS -j ACCEPT
```

Cómo permitir acceso de cliente remoto al servidor SSH local

Estas reglas permiten conexiones entrantes al servidor sshd:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 22 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $SSH_PORTS \
         -d $IPADDR 22 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE $SSH_PORTS -j ACCEPT
```

ftp (Puertos TCP 21, 20)

FTP sigue siendo uno de los medios más habituales de transferir archivos entre dos máquinas en red. Las interfaces web para FTP se han convertido también en algo común.

FTP usa dos puertos privilegiados, uno para enviar comandos y otro para enviar datos. El puerto 21 se usa para establecer la conexión inicial al servidor y enviar comandos de usuario. El puerto 20 se usa para establecer un canal de datos sobre el que los archivos y los listados de directorio se envían como datos.

FTP tiene dos modos de intercambiar datos entre cliente y servidor: el modo de puerto de canal de datos normal y el modo pasivo de canal de datos. El modo de puerto normal es el mecanismo predeterminado, original, cuando se usa el programa cliente ftp y se conecta a un sitio FTP remoto. El modo pasivo es un mecanismo más reciente, y es el que se utiliza de forma predeterminada cuando se conecta con un navegador web. Algunas veces, puede encontrar un sitio FTP que sólo sea compatible con un modo u otro. La Tabla 3.9 lista el protocolo de conexión completo cliente/servidor para el servicio FTP.

Tabla 3.9. Protocolo FTP

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	21	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	21	E	IPADDR	1024:65535	ACK
Petición del canal de datos del puerto de servidor remoto	TCP	ANYWHERE	20	E	IPADDR	1024:65535	Cualquiera
Respuesta de canal de datos puerto de cliente local	TCP	ANYWHERE	20	S	IPADDR	1024:65535	ACK

Tabla 3.9. Protocolo FTP *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición de canal de datos pasivo de cliente local	TCP	ANYWHERE	1024:65535	S	IPADDR	1024:65535	Cualquiera
Respuesta de canal de datos pasivo de servidor remoto	TCP	ANYWHERE	1024:65535	E	IPADDR	1024:65535	ACK
Petición de cliente remoto	TCP	ANYWHERE	1024:65535	E	IPADDR	21	Cualquiera
Respuesta del servidor local	TCP	ANYWHERE	1024:65535	S	IPADDR	21	ACK
Respuesta de canal de datos puerto de cliente remoto	TCP	ANYWHERE	1024:65535	E	IPADDR	20	ACK
Petición de canal de datos puerto de servidor local	TCP	ANYWHERE	1024:65535	S	IPADDR	20	Cualquiera
Petición de canal de datos pasivo de cliente remoto	TCP	ANYWHERE	1024:65535	E	IPADDR	1024:65535	Cualquiera
Respuesta de canal de datos pasivo de servidor local	TCP	ANYWHERE	1024:65535	S	IPADDR	1024:65535	ACK

Cómo permitir el acceso saliente del cliente a servidores FTP remotos

Casi con toda seguridad, la mayoría de los sitios querrán disponer del acceso cliente FTP a almacenes de archivos remotos. La mayoría de las personas querrá habilitar las conexiones salientes del cliente a un servidor remoto.

Peticiones FTP salientes

Las siguientes reglas permiten una conexión saliente a un servidor FTP remoto:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 21 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 21 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Canales de datos FTP de modo puerto normal

Las siguientes reglas permiten la conexión del canal de datos estándar, en la que los servidores remotos vuelven a llamar para establecer la conexión de datos:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE 20 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 20 -j ACCEPT
```

Este comportamiento poco usual de devolución de llamada, en la que el servidor remoto establece la conexión secundaria con el cliente, es parte de lo que convierte a FTP en algo difícil de asegurar a nivel de filtrado de paquetes. No existe ningún mecanismo para asegurar que la conexión entrante se origina realmente desde el servidor FTP remoto con el que se ha contactado. A menos que se bloqueen de forma explícita las conexiones entrantes a servicios locales que se ejecutan en puertos no privilegiados, como un servidor X Window o de SOCKS, se permite el acceso remoto a estos servicios por las reglas de cliente FTP para los canales de datos de modo de puerto.

Canales de datos FTP de modo pasivo

Las siguientes reglas permiten usar el modo pasivo del canal de datos, que es más actual y el que usan los exploradores web:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp !*-y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR $UNPIRVPORTS -j ACCEPT
```

El modo pasivo se considera más seguro que el modo puerto porque el cliente ftp inicia las conexiones tanto de control como de datos, incluso aunque la conexión se realice entre dos puertos no privilegiados.

Cómo permitir accesos entrantes al servidor FTP local

La decisión de ofrecer servicios FTP al exterior es difícil. Aunque los sitios FTP abundan en Internet, la configuración de un servidor FTP requiere gran cuidado. Son posibles numerosos actos de piratería informática cuando un servidor FTP no esté configurado correctamente.

Si el objetivo es ofrecer acceso de sólo lectura general a algún conjunto de archivos de una máquina, quizá sea mejor hacer disponibles estos archivos a través de un servidor web. Si el objetivo es permitir la carga de ficheros a la máquina desde el exterior, debe restringir severamente el acceso al servidor FTP a nivel del firewall, a nivel de tcp_wrapper y a nivel de la configuración FTP.

En cualquier caso, si se decide ofrecer servicios FTP y se quieren permitir transferencias de archivos entrantes, no debe permitirse la escritura en el sistemas de archivos mediante FTP anónimo. El permiso de escritura remota a los sistemas de archivos sólo debe permitirse desde cuentas de usuario FTP específicas autenticadas, desde sitios remotos específicos y para áreas FTP controladas y limitadas y reservadas con mucho cuidado, en el sistema de ar-

chivos. El Capítulo 7, "Problemas a nivel de administración del sistema UNIX", explica estas cuestiones de FTP.

Peticiones FTP entrantes

Las siguientes reglas permiten conexiones entrantes al servidor FTP:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 21 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 21 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Respuestas del canal de datos FTP de modo de puerto normal

Las siguientes reglas permiten al servidor FTP devolver la llamada al cliente remoto y establecer la conexión del canal de datos secundario:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR 20 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 20 -j ACCEPT
```

Respuestas del canal de datos FTP de modo pasivo

Las siguientes reglas permiten al cliente FTP establecer la conexión del canal de datos secundario con el servidor local:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Servicios web

Los servicios web se basan en el protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol). Las conexiones de cliente y servidor usan las convenciones estándar de TCP. Se pueden usar varios protocolos de comunicación de más alto de nivel y de propósito especial, además del acceso HTTP general estándar, incluyendo acceso seguro sobre SSL y el acceso mediante proxy de servidor web que proporciona el ISP. Estos protocolos de acceso usan diferentes puertos de servicio.

Acceso HTTP estándar (Puerto TCP 80)

Normalmente, los servicios web están disponibles en el puerto de servicio 80 http. La Tabla 3.10 lista el protocolo completo de conexión cliente/servidor para el servicio web http.

Tabla 3.10. Protocolo HTTP

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	443	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	443	E	IPADDR	1024:65535	ACK
Petición de cliente remoto	TCP	ANYWHERE	1024:65535	E	IPADDR	443	Cualquiera
Respuesta del servidor local	TCP	ANYWHERE	1024:65535	S	IPADDR	443	ACK

Acceso a sitio web remotos como cliente

Es casi inconcebible en el mundo actual que un sitio particular no quiera acceder a la World Wide Web desde un navegador web. Las siguientes reglas permiten acceder a servidores web remotos:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 80 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 80 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Precaución: no use tftp en Internet

tftp ofrece una versión UDP del servicio FTP, simplificada y no autenticada. Está destinado a cargar el software de arranque en enrutadores y estaciones de trabajo sin discos sobre una red local desde host seguros. Algunas personas confunden tftp como una alternativa a ftp. No lo use sobre Internet, y punto.

Cómo permitir el acceso remoto a un servidor web local

Si se decide ejecutar un servidor web propio y albergar un sitio web para Internet, las reglas generales de servidor permiten todos los accesos entrantes típicos al sitio. Esto es todo lo que la mayoría de las personas necesitan para albergar un sitio web:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp-\
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 80 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 80 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Acceso web seguro (SSL) (Puerto TCP 443)

El nivel de socket seguro (SSL, Secure Socket Layer) se usa para acceso Web cifrado seguro. El protocolo SSL usa el puerto TCP 443. Se suele ver si visita un sitio web comercial para comprar algún artículo, usar servicios bancarios en línea o entrar a un área Web protegida, en la que se le solicita información personal. La Tabla 3.11 lista el protocolo de conexión completo cliente/servidor para el servicio SSL.

Tabla 3.11. Protocolo SSL

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	443	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	443	E	IPADDR	1024:56535	ACK
Petición de cliente remoto	TCP	ANYWHERE	1024:65535	E	IPADDR	443	Cualquiera
Respuesta del servidor local	TCP	ANYWHERE	1024:65535	S	IPADDR	443	ACK

Cómo acceder a sitio web remotos como cliente SSL

La mayoría de las personas querrá acceder a sitio web seguros en un momento u otro:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 443 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 443 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo permitir el acceso remoto a un servidor web SSL local

Si se dirige alguna forma de comercio electrónico, es probable que se quieran permitir conexiones entrantes a áreas SSL protegidas de un sitio web. De lo contrario, no serán necesarias reglas de servidor locales.

La distribución básica del servidor web Apache incluye compatibilidad SSL, pero no se incluyen los módulos SSL más seguros, debido a las leyes federales sobre cifrado. Sin embargo, tanto los paquetes de soporte SSL comerciales como los gratuitos están disponibles para el servidor web Apache. Visite la dirección www.apache.org si desea más información.

Las siguientes reglas permiten el acceso al servidor web usando el protocolo SSL:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 443 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 443 \
         -d $ANYWHERE $UNPRIVPORTS-j ACCEPT
```

Acceso a un proxy web (TCP 8 Puertos 008, 8080)

Suele ser muy habitual para un ISP que los servidores proxy de web estén accesibles públicamente. Como cliente, se debe configurar el navegador para usar un servicio de proxy remoto. Se suele acceder a los proxy de web a través de uno de los dos puertos no privilegiados asignados para este propósito, el puerto 8008 o el puerto 8080, según lo defina el ISP. Gracias a esto, se consigue tener acceso más rápido a las páginas web cuando las páginas están ya en la caché local del servidor del ISP y se conserva el anonimato del acceso mediante proxy a sitios remotos. Las conexiones no son directas, sino que el proxy del ISP las inicia por el usuario. La Tabla 3.12 lista el protocolo completo de conexión cliente local a servidor remoto para el servicio proxy de web.

Tabla 3.12. Protocolo proxy de web

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	WEB PROXY SERVER	WEB PROXY PORTS	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	WEB PROXY SERVER	WEB PROXY PORT	E	IPADDR	1024:65535	ACK

Si se usa un servicio proxy web que ofrece el ISP, la dirección de servidor y el número de puerto específicos los definirá el propio ISP. Las reglas de cliente son:

```
WEB_PROXY_SERVER="mi.www.proxy"      # servidor proxy de web del
                                     # ISP, si existe
WEB_PROXY_PORT="www.proxy.puerto"    # puerto proxy de web del
                                     # ISP, si existe
                                     # normalmente 8008 o 8080

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $WEB_PROXY_SERVER $WEB_PROXY_PORT -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $WEB_PROXY_SERVER $WEB_PROXY_PORT \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

finger (Puerto TCP 79)

Desde el punto de vista de la conexión, el servicio finger es inocuo. Dada la naturaleza cambiante de las cuestiones de seguridad en relación con el crecimiento y el cambio de la comunidad de Internet, en general se ha dejado de ofrecer el servicio finger. finger proporciona información de la cuenta de usuario, como el nombre de inicio de sesión, el nombre real, las sesiones actualmente activas, el correo pendiente y las ubicaciones de reenvío de correo. finger suele proporcionar información personal de usuario que también ofrece (en un archivo .plan) números de teléfono, direcciones particulares, tareas y planes, disponibilidad, etc. La Tabla 3.13 lista el protocolo completo de conexión cliente/servidor para el servicio finger.

Tabla 3.13. Protocolo del servicio finger

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	79	S	IPADDR	1024:56535	Cualquiera
Repuesta de servidor remoto	TCP	ANYWHERE	79	E	IPADDR	1024:65535	ACK
Petición de cliente remoto	TCP	Clientes finger	1024:65535	E	IPADDR	79	Cualquiera
Repuesta de servidor local	TCP	Clientes finger	1024:65353	S	IPADDR	79	ACK

Cómo acceder a servicios remotos finger como cliente

No existe ningún problema si se habilita el acceso saliente a servidores finger remotos, y estas son las reglas para hacerlo:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 79 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 79 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo permitir el acceso cliente remoto a un servidor finger local

Si se decide permitir el acceso exterior al servicio finger, se recomienda restringir el acceso a sitios cliente específicos. Se puede restringir el acceso finger tanto a nivel de firewall como a nivel tcp_wrapper.

Las siguientes reglas permiten conexiones entrantes al servidor finger, pero sólo tienen permiso de iniciar la conexión los host remotos seleccionados:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <my.finger.clients> $UNPRIVPORTS \
         -d $IPADDR 79 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 79 \
         -d <my.finger.clients> $UNPRIVPORTS -j ACCEPT
```

whois (Puerto TCP 43)

El programa whois accede la base de datos de los servicios de registro de InterNIC. Esto permite realizar búsquedas de direcciones IP, de dominio y de nombres de dominio en formato legible para el usuario. La Tabla 3.14 lista el protocolo de conexión cliente local a servidor remoto para el servicio whois.

Tabla 3.14. Protocolo del servicio whois

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	43	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	43	E	IPADDR	1024:65535	ACK

Las siguientes reglas permiten realizar peticiones oficiales a un servidor remoto:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 43 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 43 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

gopher (TCP Puerto 70)

El servicio de información GOPHER está todavía disponible para terminales ASCII de baja sobrecarga, pero su uso se ha reemplazado casi completamente por motores de búsqueda web y por vínculos de hipertexto. No es probable que un sistema Linux ofrezca servicio GOPHER local en vez de un sitio web. No se incluyen las reglas de servidor. La Tabla 3.15 lista el protocolo de conexión cliente local a servidor remoto para el servicio GOPHER.

Tabla 3.15. Protocolo del servicio gopher

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	70	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	70	E	IPADDR	1024:65535	ACK

Estas son las reglas de cliente que permiten conectarse a un servidor remoto:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 70 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 70 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

WAIS (Puerto TCP 210)

Los servidores de información de área extensa (WAIS, Wide Area Information Servers) se conocen ahora como motores de búsqueda. Los exploradores Web suelen ofrecer una presentación gráfica a WAIS. Netscape contiene el código de cliente WAIS necesario para conectarse a WAIS. La Tabla 3.16 lista el protocolo de conexión de cliente local a servidor remoto para el servicio WAIS.

Tabla 3.16. Protocolo del servicio WAIS

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente local	TCP	ANYWHERE	210	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	210	E	IPADDR	1024:65535	ACK

Las siguientes reglas permiten el acceso de cliente a servicios WAIS remotos:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 210 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 210 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo habilitar servicios UDP habituales

El protocolo UDP sin estado es inherentemente menos seguro que el protocolo TCP basado en la conexión. Por ello, muchos sitios, conscientes de la seguridad, deshabilitan completamente, o restringen todo lo que pueden, cualquier acceso a los servicios UDP. Obviamente, los intercambios DNS basados en UDP son necesarios, pero los servidores de nombres remotos pue-

den especificarse de forma explícita en las reglas del firewall. Por tanto, esta sección proporciona reglas para sólo tres servicios:

- traceroute
- Protocolo de configuración dinámica de host (DHCP, *Dynamic Host Configuration Protocol*).
- Protocolo de tiempo de red (NTP, *Network Time Protocol*).

traceroute (Puerto UDP 33434)

traceroute es un servicio UDP que provoca que los sistemas intermedios generen mensajes Time Exceeded de ICMP para recuperar información de contadores de saltos, y hace que el sistema destino devuelva un mensaje Destination Unreachable (port not found) (Destino inalcanzable, puerto no encontrado), que indica el punto final de la ruta al *host*. El firewall desarrollado en este capítulo bloquea de forma predeterminada los paquetes traceroute entrantes de UDP destinados al intervalo de puerto que traceroute usa normalmente. Esto provoca que no se envíen las respuestas ICMP salientes a peticiones traceroute entrantes. La Tabla 3.17 lista el protocolo de conexión cliente local a servidor remoto para el servicio traceroute.

Tabla 3.17. Protocolo del servicio traceroute

Descripción	Protocolo	Dirección remota	Puerto remoto/ tipo ICMP	E/S	Dirección local	Puerto local/ Tipo ICMP
Sondeo traceroute saliente	UDP	ANYWHERE	33434:33523	S	IPADDR	32769:65535
Tiempo sobrepasado (salto intermedio)	ICMP	ANYWHERE	11	E	IPADDR	-
Puerto no encontrado (terminación)	ICMP	ANYWHERE	3	E	IPADDR	-
Sondeo traceroute entrante	UDP	ISP	32769:65535	E	IPADDR	33434:33523
Tiempo sobrepasado (salto intermedio)	ICMP	ISP	-	S	IPADDR	11
Puerto no encontrado (terminación)	ICMP	ISP	-	S	IPADDR	3

Se puede configurar traceroute para que use cualquier puerto o intervalo de puertos. Por tanto, es difícil bloquear todos los paquetes entrantes traceroute mediante el listado de puertos específicos. Sin embargo, suele usar los puertos origen en el intervalo de 32769 a 65535 y los puertos destino en el intervalo de 33434 a 33523. Se definen constantes simbólicas para los puertos origen y destino predeterminados de traceroute:

```
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"
```

Cómo habilitar peticiones salientes de traceroute

Si se quiere usar traceroute por sí mismo, es necesario habilitar los puertos cliente de UDP. Tenga en cuenta que deben permitirse los mensajes Time Exceeded y Destination Unreachable de ICMP desde cualquier lugar para que funcione el traceroute saliente:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $TRACEROUTE_SRC_PORTS \
         -d $ANYWHERE $TRACEROUTE_DEST_PORTS -j ACCEPT
```

Cómo permitir peticiones traceroute entrantes

Como traceroute es un servicio UDP menos seguro y se puede utilizar para atacar otros servicios UDP, el siguiente ejemplo abre los traceroute entrantes sólo desde el ISP y su centro de operaciones de red asociados:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -s $MY_ISP $TRACEROUTE_SRC_PORTS \
         -d $IPADDR $TRACEROUTE_DEST_PORTS -j ACCEPT
```

Tenga en cuenta que debe permitir que los mensajes salientes Time Exceeded y Destination Unreachable de ICMP se dirijan al ISP para que el traceroute entrante pueda funcionar.

Cómo acceder al servidor DHCP del ISP (Puertos UDP 67, 68)

Los intercambios DHCP, si existen, entre un sitio y el servidor del ISP se intercambiarán necesariamente entre el cliente local y el servidor remoto. Los clientes DHCP reciben, de forma dinámica, direcciones IP temporales reservadas desde un servidor central que administra el espacio de direcciones IP de los clientes del ISP.

Si se dispone de una dirección IP reservada dinámicamente desde el ISP, debe ejecutarse el demonio del cliente DHCP (dhcpd o pump) en la máquina. No es raro que existan mensajes de servidor DHCP falsos en la subred local del ISP si alguien ejecuta el servidor de forma accidental. Por esta razón, es especialmente importante filtrar los mensajes DHCP para restringir el tráfico entre el cliente y el servidor DHCP del ISP específico tanto como sea posible.

La Tabla 3.18 lista las descripciones de los tipos de mensajes DHCP según se citan en la RFC 2131, "Dynamic Host Configuration Protocol" (Protocolo de configuración dinámica de *host*).

Tabla 3.18. Tipos de mensajes DHCP

Mensaje DHCP	Descripción
DHCPDISCOVER	El cliente realiza una difusión para localizar servidores disponibles.
DHCPOFFER	Del servidor al cliente en respuesta a DHCPDISCOVER con una oferta de parámetro de configuración.
DHCPREQUEST	Mensaje del cliente a los servidores para: (a) Pedir los parámetros que se ofrecen desde un servidor y declinar, de forma implícita, los que ofrecen todos los demás; (b) Confirmar la exactitud de las direcciones reservadas previamente después de, por ejemplo, reiniciar el sistema o (c) Extender la concesión de una dirección de red particular.
DHCPACK	Del servidor al cliente con parámetros de configuración, incluyendo dirección de red asignada.
DHCPNAK	Del servidor al cliente indicando que la noción del cliente de la dirección de red es incorrecta (por ejemplo, el cliente se ha desplazado a una nueva subred) o que ha caducado la concesión del cliente.
DHCPDECLINE	Del cliente al servidor indicando que la dirección de red ya está en uso.
DHCPRELEASE	Del cliente al servidor renunciando a la dirección de red y cancelando la concesión restante.
DHCPINFORM	Del cliente al servidor preguntando sólo los parámetros de configuración local; el cliente ya tiene la dirección configurada de forma externa (no compatible con Red Hat 6.0).

En esencia, cuando el cliente DHCP se inicializa, difunde una petición DHCPDISCOVER para descubrir si hay algún servidor DHCP disponible. Cualquier servidor que reciba la petición puede responder con un mensaje DHCPOFFER indicando su buena voluntad para funcionar como servidor para este cliente, incluyendo los parámetros de configuración que tienen que ofrecer. El cliente difunde un mensaje DHCPREQUEST tanto para aceptar uno de estos servidores como para informar a los servidores restantes que ha decidido declinar sus ofertas. El servidor elegido responde con un mensaje DHCPACK, que indica la confirmación de los parámetros ofrecidos en un principio. En este momento finaliza la asignación de la dirección. Periódicamente, el cliente enviará un mensaje DHCPREQUEST pidiendo una renovación de la concesión de la dirección IP. Si se renueva la concesión, el servidor responde con un mensaje DHCPACK. De lo contrario, el cliente vuelve al proceso de inicialización. La Tabla 3.19 lista el protocolo de intercambio de cliente local a servidor remoto para el servicio DHCP.

El protocolo DHCP es mucho más complicado que este breve resumen, pero el resumen describe la esencia del intercambio típico entre cliente y servidor.

Tabla 3.19. Protocolo del servicio DHCP

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local
DHCPDISCOVER; DHCPREQUEST	UDP	255.255.255.255	67	S	0.0.0.0	68
DHCPOFFER	UDP	0.0.0.0	67	E	255.255.255.255	68
DHCPOFFER DHCPREQUEST;	UDP	DHCP SERVER	67	E	255.255.255.255	68
DHCPDECLINE	UDP	DHCP SERVER	67	S	0.0.0.0	68
DHCPACK; DHCPNAK	UDP	DHCP SERVER	67	E	ISP/máscara de red	68
DHCPACK	UDP	DHCP SERVER	67	E	IPADDR	68
DHCPREQUEST; DHCPRELEASE	UDP	DHCP SERVER	67	S	IPADDR	68

Las siguientes reglas de firewall permiten la comunicación entre el cliente DHCP y un servidor remoto:

```
DHCP_SERVER="mi.dhcp.servidor"          # si se usa uno
# INIT o REBINDING: No existe concesión o La concesión ha caducado.
ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $BROADCAST_0 68 \
         -d $BROADCAST_1 67 -j ACCEPT

# Se renumeran

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $BROADCAST_0 67 \
         -d $BROADCAST_1 68 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $DHCP_SERVER 67 \
         -d $BROADCAST_1 68 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $BROADCAST_0 68 \
         -d $DHCP_SERVER 67 -j ACCEPT

# Como resultado de lo anterior, se supone que cambiaremos la
# dirección IP con este mensaje, que se envía a la nueva
# dirección antes de que el cliente dhcp haya recibido la
# actualización.

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $DHCP_SERVER 67 \
         -d $MY_ISP 68 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $DHCP_SERVER 67 \
         -d $IPADDR 68 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 68 \
         -d $DHCP_SERVER 67 -j ACCEPT
```

Tenga en cuenta que no es posible restringir completamente el trafico DHCP al servidor DHCP. Durante las secuencias de inicialización, cuando el cliente todavía no dispone de una dirección IP asignada o incluso no se conoce la dirección IP del servidor, los paquetes se difunden en vez de enviarse punto a punto.

Acceder a servidores remotos de tiempo de red (UDP 123)

Los servicios de tiempo de red del servidor de tiempo, como NTP, permiten el acceso a uno o más proveedores de tiempo públicos de Internet. Esto es útil para mantener ajustado el reloj del sistema, particularmente si el reloj interno tiende a variar, así como para establecer la hora correcta y la fecha cuando se inicia o después de un corte de corriente. Un sistema de usuario pequeño debería usar el servicio sólo como cliente. Pocos sitios pequeños, si es que lo hace alguno, tienen un enlace satélite a Greenwich, Inglaterra, un enlace de radio al reloj atómico de los Estados Unidos o un reloj atómico propio en los alrededores.

xntpd es el demonio de servidor. Además de proporcionar servicio horario a los clientes, xntpd también usa una relación de igual a igual entre servidores. Pocos sitios pequeños requieren la precisión extra que ofrece xntpd. ntpdate es el programa cliente y usa una relación cliente a servidor. El programa cliente es todo lo que necesitará un sitio pequeño. La Tabla 3.20 lista sólo el protocolo de intercambio cliente/servidor para el servicio NTP.

Tabla 3.20. Protocolo NTP

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local
Petición del cliente local	UDP	timeserver	123	S	IPADDR	1024:65535
Respuesta del servidor remoto	UDP	timeserver	123	E	IPADDR	1024:65535

Como cliente, se puede usar ntpdate para solicitar periódicamente una serie de proveedores de servicio de tiempo público desde una tarea cron. Estos host se especificarán individualmente en una serie de reglas de firewall:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $UNPRIVPORTS \
         -d <mi.tiempo.proveedor> 123 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s <mi.tiempo.proveedor> 123 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo registrar los paquetes entrantes denegados

Cualquier paquete que coincida con una regla se puede registrar agregando la opción -l a la regla ipchains. Algunas de las reglas mostradas anteriormente tenían el registro habilitado. Las reglas de usurpamiento de dirección IP son un ejemplo de esto.

Se pueden definir reglas con el propósito concreto de registrar ciertas clases de paquetes. Normalmente, los paquetes de interés son paquetes sospechosos que indican alguna clase de sondeo o exploración. Como de forma predeterminada se deniegan todos los paquetes, si se desea registrar cierto tipo de paquetes, es necesario definir reglas explícitas antes de que el paquete llegue al final de la cadena y surta efecto la directiva predeterminada. Fundamentalmente, de todos los paquetes denegados, se puede estar interesado en registrar algunos de ellos.

Es decisión del usuario el tipo de paquetes que deben registrarse. Algunas personas quieren registrar todos los paquetes denegados. Para otras personas, registrar todos los paquetes denegados podría suponer una sobrecarga excesiva para los registros del sistema. Algunas personas, convencidas de que los paquetes se deniegan, no se preocupan de ellos y no quieren saber nada de ellos. Otros están interesadas en las exploraciones obvias de puerto o en algún tipo de paquete particular.

Debido al comportamiento de la primera regla que coincida gana, se pueden registrar todos los paquetes entrantes denegados con una única regla:

```
ipchains -A input -i $EXTERNAL_INTERFACE -j DENY -l
```

Para algunas personas, esto producirá demasiadas entradas en el registro, o demasiadas entradas de registro no interesantes. Por ejemplo, puede que se quiera registrar y denegar el trafico ICMP entrante con la excepción del servicio ping, ya que es un servicio común, sin importar si el sitio responde a las peticiones ping:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 1:7 -d $IPADDR -j DENY -l

ipchains -A input -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 9:18 -d $IPADDR -j DENY -l
```

Puede que se quiera registrar el tráfico TCP entrante denegado a todos los puertos y el tráfico UDP denegado a los puertos privilegiados:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR -j DENY -l

ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR $PRIVPORTS -j DENY -l
```

Luego, puede que se quiera registrar de nuevo todos los accesos denegados a puertos privilegiados, con la excepción de los puertos que se suelen sondear y de los que nunca se ofrece servicio:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR 0:19 -j DENY -l

# omitir ftp, telnet, ssh
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR 24 -j DENY -l

# omitir smtp
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR 26:78 -j DENY -l

# omitir finger, www
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR 81:109 -j DENY -l

# omitir pop-3, sunrpc
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR 112:136 -j DENY -l

# omitir NetBIOS
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR 140:142 -j DENY -l

# omitir imap
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR 144:442 -j DENY -l

# omitir Web seguro/SSL
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR 444:65535 -j DENY -l

# reglas UDP
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR 0:110 -j DENY -l

# omitir sunrpc
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR 112:160 -j DENY -l

# omitir snmp
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR 163:634 -j DENY -l

# omitir el servicio NFS mountd
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR 636:5631 -j DENY -l

# omitir pcAnywhere
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR 5633:31336 -j DENY -l

# omitir BackOrifice
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR 31338:33433 -j DENY -l
```

```
# omitir traceroute
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -s $ANYWHERE 32679:65535 \
         -d $IPADDR 33434:33523 -j DENY -l

# omitir el resto
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR 33434:65535 -j DENY -l
```

Cómo denegar todo tipo de acceso a sitios problemáticos

Si algún sitio tiene la costumbre de explorar la máquina del usuario o, por cualquier otra razón, es un fastidio, puede que se quiera denegar totalmente el acceso, al menos hasta que se corrija el problema de comportamiento.

Una forma de hacer esto sin tener que modificar cada vez la secuencia de comandos rc.firewall, consiste en incluir un archivo independiente de reglas denegar específicas. Insertando reglas en la cadena input en vez de agregarlas, se bloqueará el sitio incluso aunque las siguientes reglas permitieran acceder a algún servicio. El archivo se llama /etc/rc.d/rc.firewall.blocked. Para evitar un posible error en tiempo de ejecución, es necesario comprobar la existencia del archivo antes de incluirlo:

```
# Rechazar paquetes que indiquen proceder de la lista prohibida
if [ -f /etc/rc.d/rc.firewall.blocked ]; then
     . /etc/rc.d/rc.firewall.blocked
fi
```

Un ejemplo de una regla denegación global en el archivo rc.firewall.blocked podría ser:

```
ipchains -I input -i $EXTERNAL_INTERFACE -s <dirección/máscara> -j DENY
```

Cualquier paquete procedente de este intervalo de direcciones se deniega, sin tener en cuenta el tipo de protocolo del mensaje o los puertos origen o destino.

En este momento, están definidas las reglas del firewall. Cuando las reglas de firewall se configuran en el núcleo como un firewall funcional, puede conectar la máquina Linux a Internet con una buena dosis de confianza en que el sistema es seguro contra la mayoría de los ataques externos.

Cómo habilitar el acceso LAN

Si la máquina firewall se sitúa entre Internet y una LAN, las máquinas de la LAN no tienen acceso ni a la interfaz de red interna de la máquina firewall ni a Internet. El Capítulo 4 explica con detalle las cuestiones del firewall de

LAN. Un sitio pequeño, particularmente un sitio personal, no necesitará, o no tendrá los recursos para implementar la arquitectura del firewall que se muestra en el Capítulo 4. Para un sitio particular normal, y también para sitios de pequeñas empresas, el firewall de una sola máquina desarrollado en este capítulo es suficiente.

Para soportar una LAN detrás del firewall, son necesarias unas cuantas reglas más para habilitar el acceso a la interfaz de red interna de la máquina firewall y para pasar el tráfico interno a través a Internet. Cuando la máquina firewall sirve para este propósito, con dos o más interfaces de red, se llama un firewall bastión o un firewall que explora la red.

Cómo habilitar el acceso LAN a la interfaz de red interna del firewall

Para una configuración particular o de un pequeño negocio, existen pocas razones para restringir el acceso directo a la máquina firewall desde la LAN interna. Este par de reglas permite comunicaciones abiertas entre la maquina firewall y la LAN:

```
LAN_INTERFACE_1="eth1"          # interfaz LAN interna

LAN_1="192.168.1.0/24"          # su (privado) intervalo de
                                # direcciones LAN
LAN__IPADDR_1="192.168.1.1"     # su dirección interna de red

ipchains -A input  -i $LAN_INTERFACE_1 \
         -s $LAN_1 -j ACCEPT

ipchains -A output -i $LAN_INTERFACE_1 \
         -d $LAN_1 -j ACCEPT
```

Tenga en cuenta que este par de reglas permiten el acceso de la LAN a la máquina firewall. La LAN no tiene todavía por definición acceso a Internet a través del firewall, por lo que no enruta el tráfico dinámica, o automáticamente usando rutas estáticas (a menos que la máquina esté mal configurada), y necesita reglas de firewall adicionales para enrutar el tráfico local.

Cómo habilitar el acceso de LAN a Internet: reenvío y enmascaramiento IP

En este momento, se abren los puertos seleccionados para comunicaciones de cliente o de servidor, o ambos, entre máquinas remotas y la interfaz de red externa de la máquina firewall. Las comunicaciones locales entre la LAN y la máquina firewall se abren completamente a través de la interfaz de red interna del firewall. Sin embargo, las máquinas internas de la LAN no tienen todavía acceso a Internet. Permitir el acceso a Internet es un proceso de dos pasos. La comunicación entre la LAN e Internet debe reenviarse y enmascararse.

El reenvío IP es un servicio del núcleo que permite a la máquina Linux actuar como un enrutador entre dos redes, reenviando el tráfico de una red a la otra. Con una LAN, se debe habilitar el reenvío IP en la sección de enrutamiento de la configuración de red. Sin embargo, con una directiva de firewall denegar todo predeterminada en funcionamiento, los paquetes reenviados no pueden cruzar las dos interfaces hasta que lo permitan reglas específicas.

Pocos sistemas particulares querrán reenviar tráfico interno directamente. Las direcciones IP obtenidas de los intervalos de direcciones privadas de las clases A, B o C requieren enmascaramiento IP (otro servicio del núcleo), o proxy del nivel de aplicación, para sustituir una dirección IP privada de LAN con la dirección IP pública de la interfaz externa de la máquina firewall. Los paquetes con direcciones origen privadas no deben cruzar más allá de la máquina firewall en dirección a Internet, y si lo hacen, no deben enrutarse a su destino indefinidamente. Incluso si el sitio ha registrado direcciones IP estáticas, el enmascaramiento IP y los servidores proxy del nivel de aplicación son dos de las mejores formas de asegurar y aislar de forma transparente las máquinas internas de Internet.

En el nivel de administración de ipchains, el reenvío y el enmascaramiento parecen ser dos aspectos diferentes del mismo servicio (de hecho, son mecanismos independientes, pero las interfaces de usuario se combinan en el programa de administración del firewall). El reenvío enruta el tráfico de LAN desde la interfaz interna del firewall al exterior a través de la interfaz externa a Internet. Antes de que los paquetes se coloquen en la cola de salida de la interfaz externa de la máquina firewall, el servicio de enmascaramiento reemplaza la dirección origen del paquete con la dirección IP pública de la interfaz externa de la máquina firewall. El reenvío y el enmascaramiento permiten actuar a la máquina firewall como un enrutador proxy de filtrado.

La siguiente regla muestra cómo reenviar al exterior y enmascarar todo el tráfico interno a través de la interfaz externa. Las reglas ACCEPT y DENY para la cadena output de la interfaz externa se aplican después de que se apliquen las reglas de reenvío, así que incluso aunque los paquetes tengan permiso de reenvío y enmascaramiento entre las dos interfaces de red, sólo aquellos paquetes que tienen permiso para salir por las reglas de firewall para la interfaz externa realmente pasarán por ella:

```
ipchains -A forward -i $EXTERNAL_INTERFACE -s $LAN_1 -j MASQ
```

Las reglas de enmascaramiento pueden tomar las direcciones origen y destino y los argumentos de puerto, igual que lo hacen las demás reglas. El argumento de la interfaz de red es el nombre de la interfaz (externa) de reenvío, no la interfaz de red local del paquete. Aunque las reglas del ejemplo permiten todos los servicios, se pueden definir fácilmente reglas específicas para enmascarar y reenviar sólo servicios específicos, sólo a tráfico TCP, etc.

Instalación del firewall

Igual que con una secuencia de comandos del shell, la instalación es simple. La secuencia de comandos debe ser propiedad del root:

```
chown root.root /etc/rc.d/rc.firewall
```

La secuencia de comandos debe tener permisos de escritura y ejecución sólo para el root. En principio, el usuario general no debe tener acceso de lectura:

```
chmod ug=rwx /etc/rc.d/rc.firewall
```

Para inicializar o reinicializar el firewall en cualquier momento, se ejecuta la secuencia de comandos desde la línea de comandos. No es necesario reiniciar:

```
sh /etc/rc.d/rc.firewall
```

La forma de ejecutarse la.secuencia de comandos en tiempo de arranque varía dependiendo de si se dispone de una dirección IP estática o dinámica registrada, o si la dirección IP se asigna mediante DHCP.

Instalación de un firewall con una dirección IP estática

Si se dispone de una dirección IP estática, la forma más sencilla de inicializar el firewall es modificar el archivo /etc/rc.d/rc.local y agregar la siguiente línea al final del archivo:

```
sh /etc/rc.d/rc.firewall
```

Si se usan nombres de host en las reglas del firewall, es importante recordar que named debe estar activo antes de ejecutar la secuencia de comandos del firewall. Si se configura un servidor de nombres local, el sistema inicia automáticamente named antes de ejecutar rc.local en tiempo de arranque, o posteriormente, si se cambian los niveles de ejecución.

Instalación de un firewall con una dirección IP dinámica

Si se dispone de una dirección IP dinámica, la posibilidad de instalar un firewall ya no se incluye en la versión Linux 6.0 de Red Hat. Con un poco de suerte, serán tantos los clientes DHCP furiosos que se quejarán, como para volver a incluir la compatibilidad de firewall. Mientras tanto, es necesario volver a configurar la instalación predeterminada del sistema para el servicio DHCP. Los siguientes pasos funcionan en versiones de /sbin/dhcpcd anterio-

res a Red Hat 6.0. Si se ha actualizado desde una versión anterior, estos pasos restablecerán el entorno anterior:

1. Red Hat 6.0 reemplazó el cliente DHCP, dhcpcd, con un nuevo cliente, pump. pump no proporciona un mecanismo para ejecutar una secuencia de comandos cuando la dirección IP se asigna o se vuelve a asignar. Por consiguiente, si no se modifica una de las secuencias de comandos de inicio de red, /sbin/ifup, no será posible guardar la información dinámica que el servidor DHCP ofrece, ni tampoco será posible reiniciar automáticamente la secuencia de comandos del firewall si se reasigna la dirección IP después de revocar la concesión. Es necesario modificar la secuencia de comandos ejecutable /sbin/ifup, para usar el antiguo /sbin/dhcpcd en lugar de /sbin/pump.
   Consulte la sección "Compatibilidad dhcpcd en el archivo /sbin/ifup", en el Apéndice B, para ver algunos ejemplos de código.
2. Cree una nueva secuencia de comandos ejecutable del shell, /etc/sysconfig/network-scripts/ifdhcpc-done. Este archivo se incluía como parte de la versión Red Hat hasta la versión 6.0. Lo ejecutaba dhcpcd después de la asignación o reasignación de la dirección IP. pump no es compatible con la ejecución de una secuencia de comandos.
   El objetivo primero, original del ifdhcpc-done, era proporcionar un mecanismo para informar a /sbin/ifup sobre si dhcpcd consiguió con éxito la información dinámica desde el servidor DHCP o no. Dependiendo de la versión de Red Hat disponible, el fichero también realizaba otras cuantas actualizaciones de archivos.
   ifdhcpc-done es el lugar perfecto desde donde ejecutar /etc/rc.d/rc.firewall, porque ifdhcpc-done se ejecuta cada vez que se asigna o se cambia la dirección IP. Es también un lugar útil desde donde realizar otras funciones. Entre estas funciones está el establecimiento del nombre de dominio del sistema, la actualización del archivo /etc/hosts con la dirección IP actual, la actualización del archivo /etc/resolv.conf si se dispone de un servidor de nombres propio, el lanzamiento de peticiones a los servidores de nombres del ISP y proporcionar un mecanismo para entregar la dirección actual IP y las direcciones de servidor de nombre a la secuencia de comandos del firewall.
   Consulte la sección "Actualizar direcciones dinámicas e instalar el firewall desde el archivo /etc/sysconfig/network-scripts/ifdhcpc-done", en el Apéndice B, para ver ejemplos de código.
3. Cree el directorio /etc/dhcpc, de configuración del dhcpcd:

   ```
   mkdir /etc/dhcpc
   ```

   pump no usa este directorio. dhcpcd espera que exista el directorio.
4. La propia secuencia de comandos del firewall debe incluir las constantes IPADDR y NAMESERVERs del archivo /etc/dhcpc/dhcpd-eth0.info. Estas direcciones las proporciona el servidor DHCP. Las direcciones de los servidores de nombres son bastante estables. La dirección IP puede cambiar con relativa frecuencia, dependiendo de la configuración del servidor DHCP del ISP.

Resumen

Este capítulo explica los procesos involucrados a la hora de programar un firewall independiente usando el programa ipchains. Se establece, de forma predeterminada, la directiva denegar. Se han mostrado los posibles problemas iniciales, al igual que el usurpamiento de direcciones origen y la protección de los servicios que se ejecutan en puertos no privilegiados. Se han manejado los mensajes ICMP, los mensajes de control y de estado que usa el nivel de red IP subyacente. También se muestra el servicio de nombres DNS, en el que se basan todos los servicios de red, y el servicio de identificación de usuario AUTH, que es compatible con varios servicios de red habituales. Se han mostrado ejemplos de reglas para servicios de red populares y ejemplos de cómo controlar el nivel de registro producido. Por último, se describen las cuestiones relacionadas con la instalación de un firewall, tanto para sitios con una dirección IP estática, como para sitios con una dirección IP asignada de forma dinámica.

Para terminar, se amplía ligeramente la secuencia de comandos del firewall para agregar compatibilidad, de forma que sirva como un firewall bastión para una pequeña LAN. En el Apéndice B se incluyen ejemplos completos de secuencias de comandos, tanto para ipchains como para ipfwadm.

El Capítulo 4 usa el firewall bastión como base para crear una arquitectura de firewall más complicada y describe una arquitectura de subred explorada usando dos firewalls que separan una red de perímetro DMZ. Es probable que una pequeña empresa tenga necesidad, y los recursos necesarios, para implementar esta configuración más compleja.

4

Redes de perímetro, firewalls múltiples y problemas con las LAN

Un firewall de filtrado de paquetes es un enrutador estático con reglas para examinar el tráfico, las cuales hacen cumplir las directivas locales relacionadas con paquetes que tienen permiso para pasar a través de las interfaces de red. El sistema de firewall único presentado en el Capítulo 3, "Crear e instalar un firewall", es un firewall bastión básico. Como enrutador de filtrado de paquetes sobre un host con doble tarjeta (doble tarjeta significa que la máquina firewall tiene dos o más interfaces de red, una conectada a Internet y otra conectada a la LAN), el firewall aplica reglas para decidir si reenvía o bloquea los paquetes que atraviesan las dos interfaces. En un sistema con doble tarjeta que enmascara el tráfico LAN, la máquina actúa como un simple firewall examinador de red como pasarela proxy del enmascaramiento IP y para enrutamiento estático.

Para configurar el host con doble tarjeta en una LAN, las reglas de firewall aplicadas a cada interfaz de red representan un par de E/S. En el caso de la LAN, se dispone de dos pares. El firewall filtra lo que entra y lo que sale a través de la interfaz externa. También filtra, en mayor o menor grado, lo que entra y lo que sale a través de la interfaz interna hacia la LAN. Las dos interfaces se controlan de forma independiente. Además, el tráfico no se enruta directamente entre Internet y la LAN. Las reglas de filtrado en las dos interfaces actúan como un firewall bastión y como un enrutador estático entre las dos redes.

La configuración de firewall que se muestra en el Capítulo 3 se adecua perfectamente a un sistema particular. Es prácticamente lo mejor que se pue-

de hacer con un sistema independiente y personal. Se puede hacer más con un firewall de doble tarjeta y una LAN, pero la pregunta es: ¿merece la pena el esfuerzo extra que es necesario realizar para una mejora de seguridad como la que se obtiene?

Al ser un firewall bastión, si la máquina firewall se ve alguna vez comprometida, se acabó. Incluso si se aplica un segundo conjunto de reglas de firewall a la interfaz interna, si el sistema ha estado comprometido, no pasará mucho tiempo hasta que el hacker pueda acceder como root. En ese momento, si no antes, los sistemas internos estarán también abiertos de par en par. Es probable que un sistema particular nunca tenga que enfrentarse a esta situación si se eligen con mucho cuidado los servicios que se ofrecen a Internet y se hace cumplir una directiva de firewall estricta. Aun así, un firewall bastión representa un punto de error. Es una situación de todo o nada.

Las organizaciones más grandes y las LAN corporativas no dependerán de una sola máquina, de la arquitectura firewall bastión. En su lugar, usarán una arquitectura de exploración de host sin enrutamiento directo, o una arquitectura de exploración de red con servicios proxy, junto con una red DMZ de perímetro creada entre el firewall (bastión) externo y un firewall (contención) secundario interno. Los servidores públicos de la red DMZ también tienen sus propios firewalls especializados. Esto significa que estos sitios tienen muchos más equipos a su disposición y una plantilla de trabajadores para administrarlos.

Este capítulo explica las cuestiones básicas subyacentes a la seguridad de las LAN. Las directivas de seguridad se definen de forma relativa según las necesidades de seguridad del sitio, la importancia de los datos que se protegen y el coste que implicaría la pérdida de datos o de la privacidad. Este capítulo comienza mostrando los tipos de cuestiones que debe contestar el creador de la directiva del sitio cuando elige la ubicación del servidor y las directivas de seguridad.

En primer lugar, se explican las opciones que se han usado en el Capítulo 3 para una pequeña LAN particular. Aunque la arquitectura firewall que se muestra en el Capítulo 3 es excelente para una configuración particular, algunas empresas pequeñas pueden necesitar medidas más elaboradas, especialmente si algunas máquinas LAN ofrecen servicios de Internet y otras se usan para programación interna y administración de la empresa. De esta forma, el ejemplo de firewall del Capítulo 3 se amplía para hacerlo compatible con las opciones que ofrecen mayor flexibilidad que la que pueda necesitar un sitio particular.

A continuación se usa el ejemplo de firewall del Capítulo 3 como base para programar un tipo de firewall de referencia, minucioso y formal. El firewall bastión tiene dos interfaces de red: una conectada a Internet y otra conectada a una red de perímetro, o DMZ. Los servicios de Internet públicos se ofrecen desde máquinas en la red DMZ. Un segundo firewall, un firewall de contención, también esta conectado a la red DMZ, separando las redes privadas internas de las máquinas de servidor casi públicas en la red de perímetro. Las máquinas privadas están protegidas detrás del firewall de contención

en la LAN interna. Si se produce un error en el firewall bastión, los servidores públicos de la DMZ pueden quedar desprotegidos. El firewall de contención protege la LAN interna de la máquina bastión desprotegida y de cualquier otra máquina de la red de perímetro.

DMZ: una red de perímetro con otro nombre

Una red de perímetro entre dos firewalls se conoce como una DMZ, o zona desmilitarizada. El propósito de una DMZ es establecer un espacio protegido desde el que poder ejecutar servidores públicos (o servicios) y aislar dicho espacio del resto de la LAN privada.

Cuestiones de seguridad relacionadas con las LAN

Las cuestiones sobre seguridad dependen en gran medida del tamaño de la LAN, de su arquitectura y de su finalidad. ¿Se ofrecerán servicios a Internet? ¿Se albergarán estos servicios en la máquina firewall o se albergan en máquinas internas? Por ejemplo, se puede ofrecer FTP anónimo desde la máquina firewall bastión, pero servir un sitio web desde una máquina interna de la DMZ. Cuando los servicios se albergan en máquinas internas, puede que se quieran colocar estas máquinas en una red de perímetro y aplicarles un filtrado de paquetes y unas directivas de acceso completamente diferentes. Si los servicios se ofrecen desde máquinas internas, ¿se puede ver esto desde el exterior, se controlan los servicios mediante un proxy, o se reenvían de forma transparente para que parezcan estar disponibles desde la máquina firewall?

¿Cuánta información se quiere ofrecer públicamente sobre las máquinas de la LAN? ¿Se van a albergar servicios DNS locales? ¿Estará disponible el contenido de la base de datos del servicio DNS desde la máquina firewall bastión?

¿Podrá la gente iniciar sesiones en sus máquinas desde Internet? ¿Cuántas máquinas locales y cuáles son accesibles para esta gente? ¿Tendrán los mismos permisos de acceso todos los usuarios? ¿Pasarán por el proxy las conexiones entrantes para realizar un control adicional del acceso?

¿Serán todas las máquinas internas accesibles de igual manera para los usuarios locales y desde todas las máquinas? ¿Serán los servicios externos accesibles de igual manera desde todas las máquinas internas? Por ejemplo, si se usara una arquitectura firewall de exploración de host, los usuarios no tendrían que iniciar la sesión en el firewall bastión directamente para tener acceso a Internet, ni se realizaría ningún tipo de enrutamiento.

¿Se ejecutarán los servicios LAN privados detrás del firewall? ¿Por ejemplo, se usa internamente el servicio NFS, o NIS, los servidores de tiempo de red o los comandos remotos Berkeley, como rsh, rlogin y rcp? ¿Es necesario mantener alguno de estos servicios como SNMP, DHCP, timed, xntpd, remote uptime o rwho lejos de la tentación de filtrar información a Internet? Manteniendo estos servicios detrás del firewall de contención secundario se asegura un aislamiento completo de estos servicios de Internet.

Las cuestiones relacionadas con los servicios diseñados para usarlos en una LAN son cuestiones sobre acceso local frente a los accesos externos a los servicios diseñados para uso en Internet. ¿Se ofrecerá FTP internamente pero no externamente, o es posible que se ofrezcan diferentes clases de servicio FTP a ambos? ¿Se ejecutará un servidor web privado o se configurarán diferentes partes del mismo sitio para hacerlas disponibles a usuarios locales en lugar de a usuarios remotos? ¿Se ejecutará un servidor local para enviar correo pero se usará un mecanismo diferente para recibir el correo entrante desde Internet? Es decir, ¿se entregará el correo directamente a las cuentas de usuario de la máquina o se recuperará explícitamente desde un ISP?

Opciones de configuración para una LAN particular segura

Hay dos clases de tráfico de red interno a tener en cuenta. Como se muestra en la Figura 4.1, la primera es el acceso local al firewall bastión, a través de la interfaz interna. La segunda es el acceso local a Internet, a través de la interfaz externa de la máquina bastión.

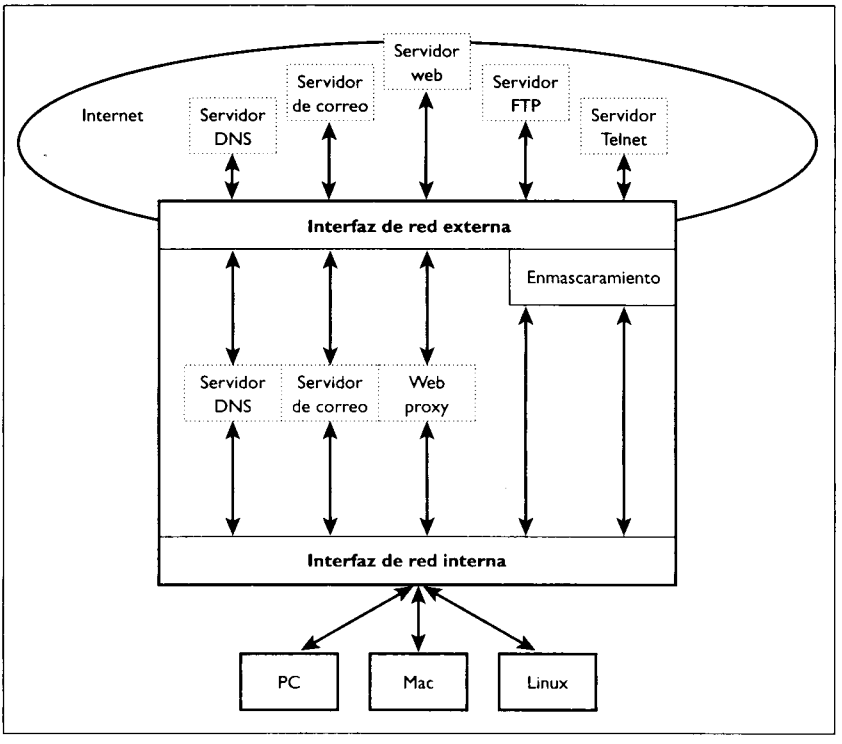

Figura 4.1. Tráfico LAN a la máquina firewall y a Internet.

Probablemente, la mayoría de los sistemas pequeños no tendrán necesidad de filtrar paquetes entre la red local y el firewall. Sin embargo, como la mayoría de los sitios particulares tienen asignada una única dirección IP, hay una excepción: el enmascaramiento IP. Es de suponer que sólo la acción relacionada con el filtrado que se realiza habilitará su propia forma de usurpamiento de dirección origen para enmascarar paquetes que se mueven entre las máquinas internas e Internet.

Si se dispone de una sola dirección IP pública para la máquina firewall, y todas las máquinas internas usan direcciones IP de clase privada, es necesario proporcionar una forma de proxy mediante el enmascaramiento IP. Si las máquinas internas tienen direcciones IP registradas, también se debería asegurar de no reenviar el tráfico de forma directa entre la LAN e Internet. Es mejor usar servidores proxy de nivel de aplicación o enmascaramiento IP para las conexiones externas.

En realidad, el enmascaramiento IP es una forma de bajo nivel de realizar un proxy. Un proxy realiza conexiones a servidores remotos en nombre de un cliente local. Todas las conexiones salientes parecen originadas desde el host que ejecuta el servidor proxy. Los paquetes que salen de la LAN se identificarán como si procediesen de la máquina bastión conectada directamente a Internet, y su dirección origen se reemplazará con la dirección de la interfaz de red externa del bastión. La dirección de los paquetes entrantes destinados a la LAN se volverán a traducir a las direcciones destino internas de los equipos.

Acceso LAN al firewall bastión

En un entorno particular, es probable que se quiera habilitar el acceso no restringido entre las máquinas LAN y el firewall bastión.

Considerando el firewall programado en el Capítulo 3 como referencia, son necesarias dos constantes más en el ejemplo de firewall para hacer referencia a la interfaz interna que conecta la LAN. Este ejemplo define la interfaz de red interna como `eth1`; la LAN se define incluyendo direcciones privadas de clase C en el intervalo de direcciones desde `192.168.1.0` hasta `192.168.1.255`:

```
LAN_INTERFACE_1="eth1"
LAN_1="192.168.1.0/24"
```

Permitir el acceso no restringido a través de la interfaz interna es una simple cuestión de permitir todos los protocolos y todos los puertos predeterminados:

```
ipchains -A input  -i $LAN_INTERFACE_1 -s $LAN_1 -j ACCEPT
ipchains -A output -i $LAN_INTERFACE_1 -d $LAN_1 -j ACCEPT
```

Acceso LAN a otras LAN: reenvío de tráfico local entre múltiples LAN

Si las máquinas de la LAN, o de múltiples LAN, necesitan servicios de enrutamiento entre ellas, será necesario permitir el acceso entre las máquinas para los puertos de servicio que necesiten, a menos que tengan trayectorias de conexión interna alternativas a través de un concentrador. En el primer caso, cualquier enrutamiento local que se realice entre las LAN se hará mediante el firewall.

Son necesarias dos constantes más. Este ejemplo define una segunda interfaz de red interna como 192.168.3.0 hasta 192.168.3.255:

```
LAN_INTERFACE_2="eth2"
LAN_2="192.168.3.0/24"
```

Las siguientes reglas permiten el acceso a la máquina firewall:

```
ipchains -A input  -i $LAN_INTERFACE_1 -s $LAN_1 -j ACCEPT
ipchains -A output -i $LAN_INTERFACE_1 -d $LAN_1 -j ACCEPT

ipchains -A input  -i $LAN_INTERFACE_2 -s $LAN_2 -j ACCEPT
ipchains -A output -i $LAN_INTERFACE_2 -d $LAN_2 -j ACCEPT
```

Las siguientes reglas reenvían el trafico en ambas direcciones entre las dos LAN, sin enmascaramiento:

```
ipchains -A forward -i $LAN_INTERFACE_2 \
         -s $LAN_1 -d $LAN_2 -j ACCEPT

ipchains -A forward -i $LAN_INTERFACE_1 \
         -s $LAN_2 -d $LAN_1 -j ACCEPT
```

Acceso LAN a Internet: reenvío frente a enmascaramiento

El reenvío enruta simplemente el tráfico entre las interfaces de red. El reenvío consiste en enrutar el tráfico entre redes. El tráfico se puede reenviar en cualquier dirección. Si las máquinas internas tienen direcciones IP registradas, se puede simplemente reenviar el tráfico y aparecerán como máquinas propias distintas a los servidores de Internet. De igual forma, se puede reenviar el trafico entrante a una máquina local particular. Por ejemplo, se puede reenviar todo el correo entrante a un servidor de correo de una máquina de la DMZ.

El enmascaramiento se sitúa por encima del reenvío, como un servicio independiente del núcleo. El tráfico se enmascara en ambas direcciones, pero no de forma simétrica. Sólo se permiten las conexiones salientes. Cualquier tráfico procedente de las máquinas de la LAN a un destino exterior se pasa a

través del firewall. En el proceso, la dirección de la máquina interna se reemplaza con la dirección de la interfaz de red externa de la máquina firewall. El proceso se invierte para las respuestas entrantes. Antes de que el paquete se reenvíe a la máquina interna, la dirección IP destino del firewall se reemplazará con la dirección IP real de la máquina interna que participa en la conexión.

No se puede reenviar una conexión entrante a una dirección interna enmascarada, porque la dirección interna es invisible para Internet. Una conexión entrante debe enrutarse directamente a una única interfaz interna pública, registrada oficialmente.

Tanto el reenvío como el enmascaramiento son servicios del nivel de núcleo; ambos deben configurarse y compilarse en el núcleo. Para usar el enmascaramiento, también debe estar habilitado el reenvío. Sin embargo, la configuración de los dos servicios es independiente. El reenvío se habilita en la configuración de red en el archivo /etc/sysconfig/network. Busque una línea que dice FORWARD_IPV4=yes. El reenvío IP se puede configurar de forma permanente, manualmente, en el archivo /etc/sysconfig/network, o mediante la interfaz GUI del panel de control. La opción de reenvió IP se encuentra en la sección de enrutamiento de los diálogos de configuración de red del panel de control. Ninguno de estos métodos de configuración tendrá efecto hasta que se reinicie la red. Si no está habilitado el reenvío IP, se puede habilitar inmediatamente escribiendo la siguiente línea como root:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

El enmascaramiento se habilita mediante el comando de enmascaramiento de ipchains.

Cuando se programa un firewall en Linux, el enmascaramiento se puede considerar como un caso especial de reenvío. En términos de la semántica ipchains, el enmascaramiento se trata como si fuera un caso especial de reenvío.

Para un sistema personal, la siguiente regla enmascara todo el tráfico procedente de máquinas de la LAN destinado a direcciones externas:

```
ipchains -A forward -i $EXTERNAL_INTERFACE \
         -s $LAN_1 -j MASQ
```

Sin embargo, el término *todo el tráfico* es relativo. Dado al comportamiento jerárquico del firewall, la primera regla que coincide gana, y sólo aquellos paquetes que se permiten a través de la interfaz de red externa se enrutarán en cualquier dirección.

Tanto si se usan direcciones IP registradas oficialmente como si se usan direcciones IP de clase privada para las máquinas locales, es mejor enmascarar que reenviar directamente, en el caso de las conexiones salientes. Enmascarar las direcciones locales es una medida de seguridad muy importante. No será posible establecer conexiones entrantes desde máquinas externas directamente a las máquinas locales. Las máquinas locales no son visibles. Todas ellas aparentan ser la máquina firewall.

Una máquina firewall no debe enrutar el trafico entrante de forma automática. Todas las direcciones LAN se deben enmascarar o pasar a través de un proxy en Internet.

Para servicios basados en protocolos de comunicación poco habituales, son necesarios otros servicios específicos, el módulo de enmascaramiento del núcleo o un proxy de nivel de aplicación independiente. Algunos servicios usan conexiones múltiples, como FTP, donde se inicia una conexión de datos secundaria mediante el servidor remoto en respuesta al flujo de control iniciado por el cliente local. Algunos servicios requieren que tanto el cliente como el servidor usen puertos no privilegiados para las comunicaciones. Este es un problema importante cuando el intercambio de información usa el protocolo UDP, donde no es posible supervisar el estado de la conexión.

Estos protocolos de comunicación poco habituales son ejemplos de por qué algunos servicios no se controlan tan fácilmente en el nivel de filtrado de paquetes. Tanto las servicios de conexión múltiple como los servicios UDP sin conexión pertenecen a esta categoría. Los dos son buenos candidatos para módulos de filtrado de paquetes o para servicios de proxy en el nivel de aplicación.

Las versiones de Linux cada vez incluyen más módulos de enmascaramiento para servicios específicos. Para la versión 6.0 de Red Hat, se incluye compatibilidad de módulos para FTP, Quake, CU-SeeMe, IRC, RealAudio y LiveVideo. Si se ha habilitado el reenvío y el enmascaramiento en la configuración de compilación del núcleo, estos módulos proxy especiales se compilan automáticamente. No obstante, es necesario cargar explícitamente los módulos que se quieran usar.

Como alternativa, o como medida adicional de seguridad, se puede usar un filtro de proxy de nivel de aplicación, como SOCKS. De nuevo, todo el tráfico saliente parecerá originado desde la máquina firewall. Además, si el sitio necesita un control de acceso más fino que el que ofrece el filtrado de paquetes, los servidores proxy de nivel de aplicación suelen ofrecer dicho control. Los servidores proxy casi siempre son específicos de una aplicación. Comprenden el protocolo de comunicación de la aplicación y pueden supervisar el origen y destino, cosa que no es posible a nivel de filtrado de paquetes.

Opciones de configuración para una mayor o menor LAN segura

Una gran empresa u organización usará mecanismos específicos más elaborados que las sencillas reglas de firewall de enmascaramiento y reenvío genérico que se acaban de mostrar en la última sección para una LAN particular segura. En entornos menos seguros, las máquinas firewall se protegen de los usuarios internos tanto como de los usuarios externos. Se definen reglas de firewall específicas del puerto para la interfaz interna, así como para la interfaz

externa. Estas reglas pueden ser una imagen reflejada de las reglas de la interfaz externa, o pueden ser más globales. Lo que se permite a través de las interfaces de red internas de la máquina firewall depende del tipo de sistemas que se ejecutan en la LAN y los tipos de servicios locales en ejecución en la máquina firewall, si existe alguno, a los que se quiere permitir el acceso.

Por ejemplo, puede que se quieran bloquear los mensajes de difusión locales para que no alcancen el firewall bastión. Si no se confía plenamente en todos los usuarios, se querrá restringir lo que pasa al firewall desde las máquinas internas tanto como lo que procede de Internet. Además, se debe mantener en el mínimo necesario el número de cuentas de usuario de la máquina firewall.

El enmascaramiento IP o las cuestiones de proxy son las mismas para una gran LAN que para una LAN particular. Una empresa particular puede tener una sola dirección IP, que necesite enmascaramiento de dirección de LAN. Sin embargo, los negocios suelen alquilar varias direcciones IP registradas públicamente, o un bloque entero de direcciones de red. Con direcciones IP públicas, las conexiones salientes se suelen reenviar y las conexiones entrantes se enrutan. En vez de crear una red de clase privada para servidores públicos internos, se define una subred local para crear una LAN DMZ pública y local.

División en subredes para crear múltiples redes

Las direcciones IP se dividen en dos partes: una dirección de red y una dirección de host dentro de dicha red. Las direcciones 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.

Crear subredes es una extensión local a la parte de la dirección de red de las direcciones IP locales. Se define una máscara de red local que trata algunos de los bits más significativos de la dirección del host como si fueran parte de la dirección de red. Estos bits adicionales de la dirección de red sirven para definir múltiples redes de forma local. Los sitios remotos no pueden ver las subredes locales. Ellos ven el intervalo de direcciones como direcciones normales de las clases A, B o C.

Por ejemplo, consideremos el bloque de direcciones privada de clase C 192.168.1.0. 192.168.1.0, que define la dirección como una dirección de red de clase C, que contiene hasta 254 host. La máscara de red para esta red es 255.255.255.0, que coincide de forma exacta con los primeros 24 bits, la dirección de red, de la red 192.168.1.0.

Esta red puede dividirse en dos subredes locales definiendo los primeros 25 bits, en vez de los primeros 24 bits, como la dirección de red. Los bits más significativos del campo dirección de host se tratan ahora como parte del campo de dirección de red. El campo host contiene ahora 7 bits, en lugar de 8. La máscara de red pasa a ser 255.255.255.128. Se definen dos subredes: 192.168.1.0,que direcciona los host 1 a 126, y 192.168.1.128, que direcciona los host 129 a 254. Cada subred pierde dos direcciones de host porque cada

subred usa las direcciones de host más bajas, 0 ó 129, como la dirección de red, y la dirección de host más alta, 127 ó 255, como la dirección de difusión. La Tabla 4.1 muestra esto.

Tabla 4.1. Red 192.168.1.0 de clase C dividida en dos redes

Subred	Dirección de red	Máscara de dirección	Primer *host* direccionable	Último *host* direccionable	Dirección difusión	Número total de *host*
0	192.168.1.0	255.555.255.0	192.168.1.1	192.168.1.254	192.168.1.255	254
1	192.168.1.0	255.555.255.128	192.168.1.1	192.168.1.126	192.168.1.127	126
2	192.168.1.128	255.555.255.128	192.168.1.129	192.168.1.254	192.168.1.255	126

Las subredes 192.168.1.0 y 192.168.1.128 se pueden asignar a dos tarjetas independientes internas de interfaz de red. Desde le punto de vista de Internet, el sitio consta de una sola red de hasta 254 *host*. Internamente, el sitio consta de dos redes independientes, donde cada una contiene hasta 126 *host*.

Dividir las redes permite crear múltiples redes internas, cada una conteniendo diferentes clases de máquinas cliente o servidor, cada una con su propio e independiente enrutamiento. Se pueden aplicar las diferentes directivas de firewall a las dos redes. Aunque es posible enmascarar la dirección LAN, el tráfico de las subredes se suele reenviar sin enmascaramiento.

Acceso selectivo por *host*, intervalo de direcciones o por puerto

El tráfico a través de una interfaz interna de una máquina firewall puede restringirse de forma selectiva, al igual que el tráfico a través de la interfaz externa. Por ejemplo, en vez de dejar que pase cualquier tráfico a través de la interfaz interna, el tráfico se puede restringir a los servicios DNS, SMTP, AUTH, POP y HTTP. En este caso, diremos que la máquina firewall ofrece estos servicios para la LAN. Las máquinas locales no tienen permiso para acceder a otros servicios exteriores que no sean éstos.

Punto de interés

En este ejemplo, los host locales se restringen a servicios específicos: DNS, SMTP, AUTH, POP y HTTP. Como POP es un servicio de obtención de correo local y los servicios DNS, SMTP y HTTP son, inherentemente, servicios de proxy, los clientes locales no acceden a Internet. En cada caso, los clientes locales se conectan a servidores locales. POP es un servicio LAN local. Los otros tres servidores establecen conexiones remotas en nombre del cliente.

El siguiente ejemplo considera una máquina firewall con una interfaz interna conectada a una LAN. Las constantes para la interfaz interna son:

```
LAN_INTERFACE="eth1"                    # interfaz interna a la LAN
firewall="192.168.1.1"                  # interna a la máquina firewall
                                        # dirección de interfaz
LAN_ADDRESSES="192.168.1.0/24"          # intervalo de direcciones que
                                        # se usa en la LAN
```

Las máquinas LAN apuntan a la interfaz interna de la máquina firewall como si fuera el servidor de nombres:

```
ipchains -A input -i $LAN_INTERFACE -p udp \
         -s $LAN_ADDRESSES $UNPRIVPORTS \
         -d $firewall 53 -j ACCEPT

ipchains -A output  -i $LAN_INTERFACE -p udp \
         -s $firewall 53 \
         -d $LAN_ADDRESSES $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $LAN_INTERFACE -p tcp \
         -s $LAN_ADDRESSES $UNPRIVPORTS \
         -d $firewall 53 -j ACCEPT

ipchains -A output  -i $LAN_INTERFACE ! -y -p tcp \
         -s $firewall 53 \
         -d $LAN_ADDRESSES $UNPRIVPORTS -j ACCEPT
```

Las máquinas LAN también apuntan al firewall como si fuera el servidor POP y SMTP:

```
# Enviar correo - SMTP

ipchains -A input  -i $LAN_INTERFACE -p tcp \
         -s $LAN_ADDRESSES $UNPRIVPORTS \
         -d $firewall 25 -j ACCEPT

ipchains -A output -i $LAN_INTERFACE ! -y -p tcp \
         -s $firewall 25 \
         -d $LAN_ADDRESSES $UNPRIVPORTS -j ACCEPT

# Recibir correo - POP

ipchains -A input  -i $LAN_INTERFACE -p tcp \
         -s $LAN_ADDRESSES $UNPRIVPORTS \
         -d $firewall 110 -j ACCEPT

ipchains -A output -i $LAN_INTERFACE ! -y -p tcp \
         -s $firewall 110 \
         -d $LAN_ADDRESSES $UNPRIVPORTS -j ACCEPT
```

El servidor sendmail iniciará una petición de búsqueda AUTH del cliente de correo:

```
ipchains -A output -i $LAN_INTERFACE -p tcp \
         -s $firewall $UNPRIVPORTS  \
         -d $LAN_ADDRESSES 113 -j ACCEPT
ipchains -A input  -i $LAN_INTERFACE ! -y -p tcp \
         -s $LAN_ADDRESSES 113 \
         -d $firewall $UNPRIVPORTS -j ACCEPT
```

Por último, se ejecuta un servidor web proxy local en la máquina firewall en el puerto 8080. Las máquinas apuntan al servidor web en el firewall como su proxy, y el servidor web reenvía cualquier petición saliente en su nombre, además de guardar en la caché las páginas obtenidas de Internet:

```
ipchains -A input  -i $LAN_INTERFACE -p tcp \
         -s $LAN_ADDRESSES $UNPRIVPORTS \
         -d $firewall 8080 -j ACCEPT

ipchains -A output -i $LAN_INTERFACE ! -y -p tcp \
         -s $firewall 8080 \
         -d $LAN_ADDRESSES $UNPRIVPORTS -j ACCEPT
```

Opciones de configuración para múltiples LAN

Agregar una segunda LAN interna permite desarrollar todavía más este ejemplo. Como se muestra en la Figura 4.2, los servicios DNS, SMTP, AUTH, POP y HTTP se ofrecen desde las máquinas servidor en una segunda LAN en vez de hacerlo desde la máquina firewall. En este caso, el tráfico se enruta entre las dos LAN mediante las interfaces internas de la máquina firewall (aunque el enrutamiento se puede hacer directamente mediante un concentrador o un conmutador).

Las siguientes variables se utilizan para definir las LAN, las interfaces de red y las máquinas servidor de este ejemplo:

```
CLIENT_LAN_INTERFACE="eth1"  # interfaz interna a la LAN
SERVER_LAN_INTERFACE="eth2"  # interfaz interna a la LAN
firewall_1="192.168.1.1"     # interna a la máquina firewall
                             # dirección de interfaz
firewall_2="192.168.3.1"     # interna a la máquina firewall
                             # dirección de interfaz
CLIENT_LAN="192.168.1.0/24"  # intervalo de direcciones que se
                             # usa en la LAN
SERVER_LAN="192.168.3.0/24"  # intervalo de direcciones que se
                             # usa en la LAN
DNS_SERVER="192.168.3.2"     # servidor DNS de la LAN
MAIL_SERVER="192.168.3.3"    # servidor POP y de correo de la LAN
POP_SERVER="192.168.3.3"     # servidor POP y de correo de la LAN
WEB_SERVER="192.168.3.4"     # servidor web de la LAN
```

Las máquinas internas apuntan a la dirección IP del servidor en la LAN del servidor como su servidor de nombres. Al igual que con las reglas entre las interfaces interna y externa del firewall, las reglas de acceso a servidor se definen para la interfaz LAN del cliente. Las reglas de acceso de cliente se definen para la interfaz LAN del servidor:

```
ipchains -A input -i $CLIENT_LAN_INTERFACE -p udp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $DNS_SERVER 53 -j ACCEPT

ipchains -A output -i $SERVER_LAN_INTERFACE -p udp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $DNS_SERVER 53 -j ACCEPT
```

```
ipchains -A input  -i $SERVER_LAN_INTERFACE -p udp \
         -s $DNS_SERVER 53 \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CLIENT_LAN_INTERFACE -p udp \
         -s $DNS_SERVER 53 \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT
```

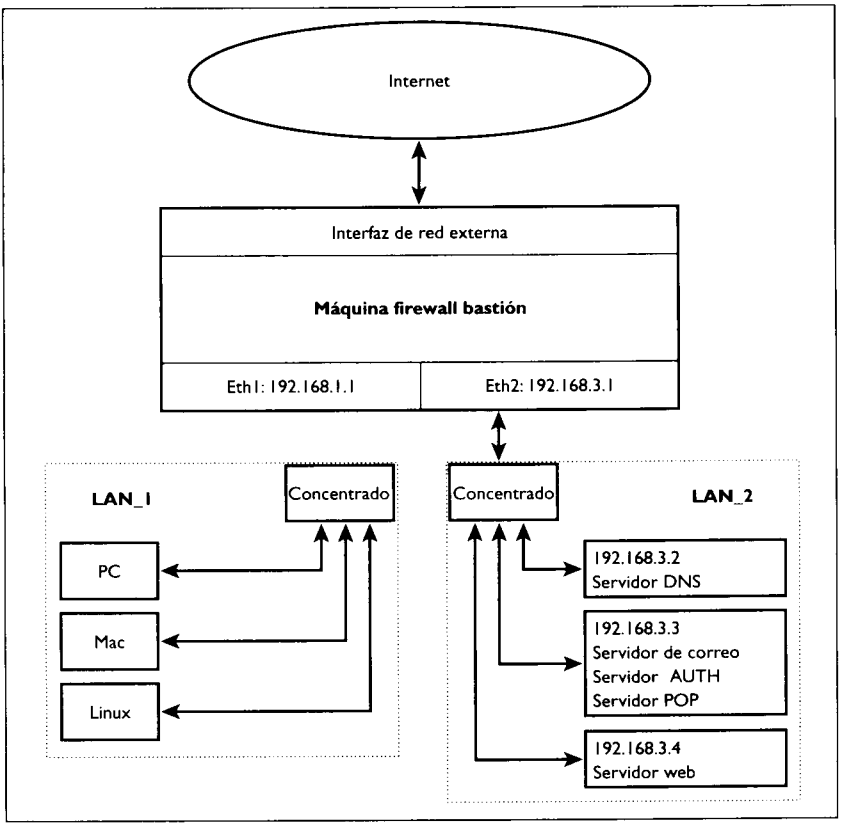

Figura 4.2. Separación de clientes y servidores en múltiples LAN.

Son necesarias reglas de reenvío para permitir el tráfico entre las dos interfaces. En este caso, las reglas de reenvío son específicas de la interfaz, del intervalo de direcciones y de los puertos del cliente y del puerto y direcciones del servidor:

```
ipchains -A forward -i $SERVER_LAN_INTERFACE \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $DNS_SERVER 53 -j ACCEPT

ipchains -A forward -i $CLIENT_LAN_INTERFACE \
         -s $DNS_SERVER 53  \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT
```

Sin embargo, la situación es algo más complicada. El servidor DNS de la segunda LAN necesita conseguir la información de un origen externo. Si el servidor interno fuera un servidor principal enmascarado para un servidor externo, que reenviara las búsquedas no resueltas al servidor externo, las reglas UDP del firewall para la interfaz LAN interna y la interfaz de Internet externa serían:

```
ipchains -A input    -i $SERVER_LAN_INTERFACE -p udp \
         -s $DNS_SERVER 53 \
         -d <servidor.nombres.externo> 53 -j ACCEPT

ipchains -A output   -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d <servidor.nombres.externo> 53 -j ACCEPT

ipchains -A input    -i $EXTERNAL_INTERFACE -p udp \
         -s <servidor.nombres.externo> 53 \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output   -i $SERVER_LAN_INTERFACE -p udp \
         -s <servidor.nombres.externo> 53 \
         -d $DNS_SERVER 53 -j ACCEPT

ipchains -A forward -i $EXTERNAL_INTERFACE -p udp \
         -s $DNS_SERVER 53 \
         -d <servidor.nombres.externo> 53 -j MASQ
```

Los clientes en la CLIENT_LAN apuntan al MAIL_SERVER como su servidor SMTP para enviar correo:

```
# Enviar correo - SMTP
# ----------

ipchains -A input    -i $CLIENT_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $MAIL_SERVER 25 -j ACCEPT

ipchains -A output   -i $SERVER_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $MAIL_SERVER 25 -j ACCEPT

ipchains -A input    -i $SERVER_LAN_INTERFACE ! -y -p tcp \
         -s $MAIL_SERVER 25 \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT

ipchains -A output   -i $CLIENT_LAN_INTERFACE ! -y -p tcp \
         -s $MAIL_SERVER 25 \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT

ipchains -A forward -i $SERVER_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $MAIL_SERVER 25 -j ACCEPT

ipchains -A forward -i $CLIENT_LAN_INTERFACE -p tcp \
         -s $MAIL_SERVER 25  \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT
```

El servidor SMTP en la SERVER_LAN necesita enviar el correo a destinos remotos. El servidor debe acceder a Internet a través del firewall:

```
ipchains -A input    -i $SERVER_LAN_INTERFACE -p tcp \
         -s $MAIL_SERVER $UNPRIVPORTS \
         -d $ANYWHERE 25 -j ACCEPT

ipchains -A output   -i $SERVER_LAN_INTERFACE ! -y -p tcp \
         -s $ANYWHERE 25 \
         -d $MAIL_SERVER $UNPRIVPORTS -j ACCEPT

ipchains -A forward -i $EXTERNAL_INTERFACE -p tcp \
         -s $MAIL_SERVER $UNPRIVPORTS \
         -d $ANYWHERE 25 -j MASQ
```

El servidor sendmail iniciará una petición de búsqueda AUTH al cliente de correo:

```
ipchains -A output -i $CLIENT_LAN_INTERFACE -p tcp \
         -s $MAIL_SERVER $UNPRIVPORTS  \
         -d $CLIENT_LAN 113 -j ACCEPT

ipchains -A input  -i $CLIENT_LAN_INTERFACE ! -y -p tcp \
         -s $CLIENT_LAN 113 \
         -d $MAIL_SERVER $UNPRIVPORTS -j ACCEPT
```

En realidad, el servidor SMTP de la SERVER_LAN también necesitará recibir correo de fuentes remotas. Como se muestra aquí, no es posible recibir correo entrante directamente mediante un demonio sendmail interno, porque el servidor de correo está enmascarado. Las soluciones a este problema se explican posteriormente en este capítulo. Por el bien de la explicación, la solución basada en la información que se ha mostrado hasta el momento en este capítulo sirve para ejecutar un servidor de correo en el bastión. El servidor bastión transmitirá el correo entre la máquina MAIL_SERVER y las máquinas remotas.

Los clientes de la CLIENT_LAN apuntan a la máquina POP_SERVER para obtener el correo:

```
# Recibir correo - POP
# ----------

ipchains -A input    -i $CLIENT_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $POP_SERVER 110 -j ACCEPT

ipchains -A output   -i $SERVER_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $POP_SERVER 110 -j ACCEPT

ipchains -A input    -i $SERVER_LAN_INTERFACE ! -y -p tcp \
         -s $POP_SERVER 110 \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT
```

```
ipchains -A output  -i $CLIENT_LAN_INTERFACE ! -y -p tcp \
         -s $POP_SERVER 110 \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT

ipchains -A forward -i $SERVER_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $POP_SERVER 110 -j ACCEPT

ipchains -A forward -i $CLIENT_LAN_INTERFACE -p tcp \
         -s $POP_SERVER_110  \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT
```

Por último, se ejecuta un servidor proxy web local en un servidor de la LAN de servidores, conectada al puerto 8080. Las máquinas internas apuntan al servidor web como su proxy de caché, y el servidor web reenvía cualquier petición saliente en su nombre:

```
# WWW PROXY
# ----------

ipchains -A input   -i $CLIENT_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $WEB_SERVER 8080 -j ACCEPT

ipchains -A output  -i $SERVER_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $WEB_SERVER 8080 -j ACCEPT

ipchains -A input   -i $SERVER_LAN_INTERFACE ! -y -p tcp \
         -s $WEB_SERVER 8080 \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT

ipchains -A output  -i $CLIENT_LAN_INTERFACE ! -y -p tcp \
         -s $WEB_SERVER 8080 \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT

ipchains -A forward -i $SERVER_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $WEB_SERVER 8080 -j ACCEPT

ipchains -A forward -i $CLIENT_LAN_INTERFACE -p tcp \
         -s $WEB_SERVER 8080  \
         -d $CLIENT_LAN $UNPRIVPORTS -j ACCEPT
```

El servidor web de la LAN de servidores también necesita acceder a Internet para los servidores remotos que escuchan en el puerto TCP 80:

```
ipchains -A input   -i $SERVER_LAN_INTERFACE -p tcp \
         -s $WEB_SERVER $UNPRIVPORTS \
         -d $ANYWHERE 80 -j ACCEPT

ipchains -A output  -i $SERVER_LAN_INTERFACE ! -y -p tcp \
         -s $ANYWHERE 80 \
         -d $WEB_SERVER $UNPRIVPORTS -j ACCEPT

ipchains -A forward -i $EXTERNAL_INTERFACE -p tcp \
         -s $WEB_SERVER $UNPRIVPORTS \
         -d $ANYWHERE 80 -j MASQ
```

Cómo enmascarar el tráfico LAN hacia Internet

Hasta el momento, se han usado varias reglas de reenvío y enmascaramiento. Hay disponible una serie de opciones en términos de lo específicas que se quieran hacer las reglas.

Enmascaramiento por interfaz

Para una LAN particular, lo más fácil es enmascarar simplemente todo el tráfico entre la LAN e Internet. La siguiente regla enmascara todo el tráfico que se permite a través de la interfaz externa por las reglas de entrada y salida del firewall:

```
ipchains -A forward -i $EXTERNAL_INTERFACE \
         -s $CLIENT_LAN -j MASQ
```

La regla de enmascaramiento general no permite todo el tráfico de cliente LAN a través de la interfaz externa. Las reglas para la interfaz externa definen el tráfico que puede pasar a través de la interfaz externa. Esta regla habilita el enmascaramiento de cualquier tráfico entre la LAN e Internet que las reglas de interfaz externa permitan pasar.

Enmascaramiento mediante servicio

Si se prefieren reglas más explícitas, o se quiere permitir el acceso LAN sólo a un subconjunto de los servicios externos disponibles a través de los filtros del firewall de la interfaz externa, se puede enmascarar mediante servicio. Cualquier cosa no especificada mediante una regla de reenvío y enmascaramiento no se enruta al exterior hacia Internet. La siguiente regla permite específicamente a los navegadores web internos acceder a servidores web externos:

```
ipchains -A forward -i $EXTERNAL_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $ANYWHERE 80 -j MASQ
```

Enmascaramiento mediante interfaz, servicio y host

Se puede conseguir que las reglas de firewall hacia las interfaces, puertos y host sean más específicas. En el ejemplo anterior que mostraba dos LAN, todas las máquinas de ambas LAN usaban una máquina particular en la LAN de servidores como el servidor de nombres local, DNS_SERVER. Sólo el DNS_SERVER puede acceder a un servidor de nombres DNS externo. Como esta máquina ejecuta un servidor de nombres de reenvío, usa el puerto UDP 53 a 53, comunicación de igual a igual con un servidor de nombres externo. Aunque SERVER_LAN_INTERFACE no se define explícitamente en la regla de enmascaramiento, la interfaz está involucrada debido al hecho de que el DNS_SERVER tiene una dirección específica relativa a SERVER_LAN_INTER-

FACE. Es decir, ninguna otra máquina en cualquiera de las LAN tiene permiso para acceder al servidor de nombres externo:

```
ipchains -A forward -i $EXTERNAL_INTERFACE -p udp \
         -s $DNS_SERVER 53 \
         -d <isp.dns.servidor> 53 -j MASQ
```

Si el servidor se ejecutara como cliente DNS para el servidor DNS externo, las reglas de enmascaramiento serían:

```
ipchains -A forward -i $EXTERNAL_INTERFACE -p udp \
         -s $DNS_SERVER $UNPRIVPORTS \
         -d <isp.dns.servidor> 53 -j MASQ

ipchains -A forward -i $EXTERNAL_INTERFACE -p tcp \
         -s $DNS_SERVER $UNPRIVPORTS \
         -d <isp.dns.servidor> 53 -j MASQ
```

Enmascaramiento y jerarquía de precedencia de reglas

El sistema de una sola máquina firewall independiente desarrollado en el Capítulo 3 usaba reglas específicas de entrada y salida en la interfaz externa. La configuración más sencilla para una LAN particular es permitir que todo entre y salga en la interfaz interna y reenviar y enmascarar todo lo destinado a una dirección externa remota. Las reglas de entrada, salida y reenvío las usa la LAN, pero las reglas que se aplican a la interfaz externa proporcionan el control de filtrado para todo el sitio.

Para cualquier cadena dada, sea input, output, forward o una cadena concreta del sitio, el paquete se compara con las reglas en el orden en que éstas se definieron. La primera regla que coincida gana. Con una LAN y reenvío o enmascaramiento, la cuestión se convierte en ¿qué cadena de reglas se aplica y cuándo?

Como se muestra en la Figura 4.3, cada paquete llega a la cadena input para dicha interfaz. Las reglas en la cadena input son las primeras en comprobarse. Si el filtro de entrada acepta el paquete, la siguiente cadena la determina el destino del paquete. Si el destino es la máquina local, el paquete se coloca en la cadena output de la interfaz de bucle invertido. Si el destino es una interfaz de una máquina diferente, el paquete se pasa primero a la cadena forward. Si el filtro de reenvío acepta el paquete, el paquete se enmascara y se coloca en la cadena output para la interfaz de pasarela. Por último, se comprueban los filtros de la cadena output. Si las reglas de filtro aceptan el paquete para dicha interfaz, se envía hacia su destino final (desde el punto de vista del firewall).

Precedencia de las reglas del firewall

Para una explicación más completa sobre precedencia de reglas de firewall, consulte al documento IPCHAINS-HOWTO.

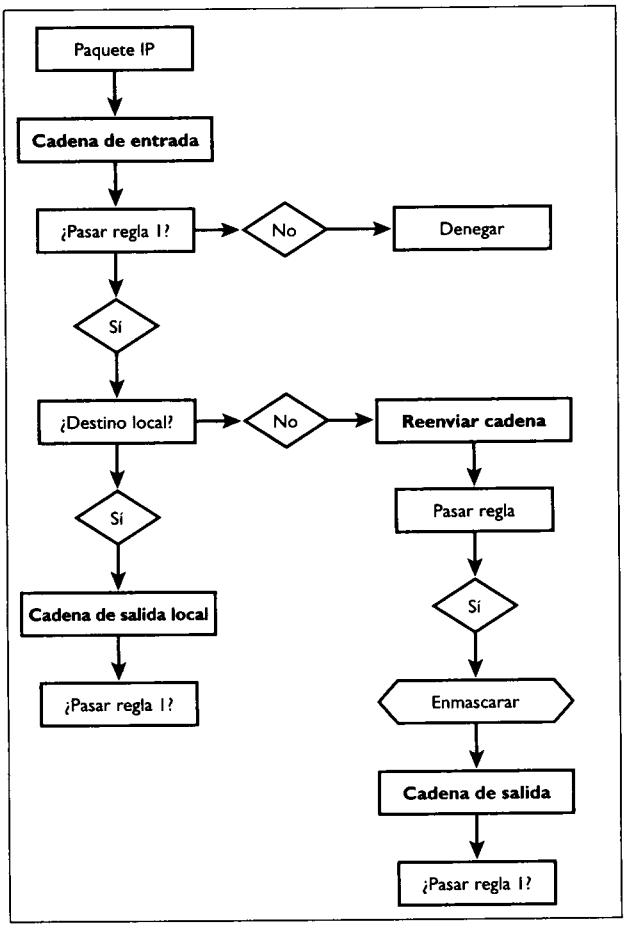

Figura 4.3. Jerarquía de enmascaramiento y precedencia de reglas.

Redirección de puerto: los proxy transparentes

Las características de redirección de puerto del programa ipchains son un caso especial limitado de reenvío y enmascaramiento de puerto local. Las conexiones no se reenvían o enmascaran entre las interfaces de red. Por el contrario, cualquier tráfico que cumpla la regla, sin tener en cuenta la dirección o el puerto destino del paquete, se redirecciona a un puerto local.

Esta característica es útil para algunos de los servidores proxy del nivel de aplicación que se pueden ejecutar. A menudo, los proxy del nivel de aplicación necesitan que se vuelva a configurar o reemplazar su software cliente con aplicaciones especiales que conocen el servicio proxy local. La redirección de puertos puede hacer la redirección invisible a los programas cliente si no se usa un protocolo especial de proxy (es decir, el protocolo HTTP di-

recto a un servidor web no es lo mismo que el protocolo HTTP enrutado a través de un proxy a un servidor proxy web).

Por ejemplo, supongamos que se ejecuta un servidor proxy telnet. Éste escucha en la interfaz interna de la LAN del puerto normal de telnet, el 23. Cualquier conexión saliente de telnet desde la LAN se redirecciona al proxy local. El servidor proxy realiza luego la conexión en nombre del programa cliente telnet local:

```
ipchains -A input -i $CLIENT_LAN_INTERFACE -p tcp \
         -s $CLIENT_LAN $UNPRIVPORTS \
         -d $ANYWHERE 23 -j REDIRECT 23
```

Reenvío de peticiones de conexión entrantes procedentes de Internet a servidores internos

Como las máquinas enmascaradas locales son invisibles al mundo exterior, los servicios que se están ejecutando en las máquinas locales no están disponibles para los clientes remotos. Ofrecemos el código experimental del núcleo para habilitar conexiones entrantes a servidores que se ejecutan en máquinas enmascaradas internas. La característica se habilita compilando la compatibilidad de módulo de enmascaramiento y la compatibilidad ipportfw masq en el núcleo. Estas características experimentales requieren el uso de ipmasqadm, una aplicación de otros fabricantes. ipmasqadm no se incluye en las versiones de Linux. Debe descargarse de forma independiente desde Internet.

Como ejemplo, supongamos que se quieren reenviar las conexiones entrantes al servidor web a un servidor que se ejecuta en una máquina enmascarada interna. Son necesarios dos comandos ipmasqadm además de las reglas de firewall normales. La primera regla elimina inicialmente la cadena de reenvío de puerto. La segunda regla reenvía las conexiones entrantes TCP dirigidas al puerto HHTP 80 de la máquina firewall a un servidor web interno en la dirección IP enmascarada 192.168.3.5:

```
ipmasqadm portfw -f
ipmasqadm portfw -a -P tcp -L $IPADDR 80 -R 192.168.3.5 80
```

Compatibilidad de núcleo proxy transparente

El proxy transparente necesita que se haya habilitado la compatibilidad de proxy transparente en la configuración de opciones de redes del núcleo. Esta característica está habilitada de forma predeterminada en la versión 6.0 de Red Hat. En versiones anteriores, debe habilitarse explícitamente y volver a compilar el núcleo.

Reenvío de puerto

Se puede encontrar información adicional sobre estos servicios experimentales en la dirección http://juanjox.linuxhq.com, http://www.monmouth.demon.uk/ipsubs/portforwarding.html, ftp://ftp.compsoc.net/users/steve/ipportfw/Linux21, http://ipmasq.cjb.net y ayuda en línea en el documento *"IP Masquerade HOWTO"*, que se encuentra en /usr/doc/HOWTO/mini/IP-Masquerade.

Una alternativa es usar un servidor proxy local para el servicio que se quiere hacer disponible. Se pueden usar los servidores proxy para conexiones tanto entrantes como salientes. A menudo, un servidor proxy es la idea de un programa de pasarela del nivel de aplicación que enmascara las conexiones salientes a servicios externos. Se suelen usar los servidores proxy para conexiones entrantes a servicios locales con el fin de hacer cumplir las especificaciones del protocolo de comunicación y para poder realizar un mejor control a nivel de filtrado de paquetes. El servidor proxy reenvía la conexión entrante a un servidor enmascarado local.

Existe una alternativa final para los sitios con múltiples direcciones IP registradas. La interfaz externa acepta los paquetes entrantes destinados a cualquier dirección del espacio de direcciones de la red local. El firewall envía tipos de conexiones especificas a servidores internos concretos. Por ejemplo, la siguiente regla reenvía las conexiones entrantes a un servidor web que se ejecuta en una LAN:

```
ipchains -A forward -i $SERVER_LAN_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $WEB_SERVER 80 -j ACCEPT
```

En esta configuración, sólo se permiten las conexiones externas al servidor web. Las máquinas remotas no tienen otro acceso a la máquina servidor interna.

Un firewall formal de exploración de subred

Es posible que una pequeña y mediana empresa tenga suficientes razones como para invertir en una arquitectura de firewall más elaborada. Una máquina con doble tarjeta sirve como firewall bastión entre Internet y la LAN interna, igual que se explicó al final del Capítulo 3. Sin embargo, la interfaz interna conecta a una red de perímetro, red DMZ, en vez de a la LAN privada. Los servicios públicos se encuentran en máquinas de la red de perímetro DMZ, cada uno con un firewall independiente y una directiva de seguridad. Los servidores públicos pueden tener o no interfaces públicamente visibles, dependiendo de si se han asignado direcciones IP públicas.

Una DMZ se suele configurar de una de las dos formas básicas. En el primer método, la máquina firewall bastión tiene tres interfaces de red, una que conecta a Internet y dos interfaces internas que se conectan con dos LAN independientes. Una LAN es una LAN DMZ para servidores públicos. La otra LAN es la LAN interna privada. La Figura 4.4 muestra este tipo de configuración DMZ.

La segunda configuración DMZ usa una segunda máquina firewall, llamada firewall de contención. La máquina de contención se sitúa en el otro extremo de la red de perímetro, formando una pasarela entre la red DMZ y la LAN privada. Su interfaz interna se conecta a la LAN privada. Las máquinas internas es-

tán enmascaradas, todas aparecen como la dirección de la interfaz externa del firewall de contención para la máquina bastión y las máquinas DMZ.

El firewall bastión enmascara todo el tráfico LAN privado, por lo que es innecesario el enmascaramiento de máquinas LAN en el firewall de contención. Sin embargo, las reglas del firewall bastión son más simples cuando la única dirección IP de la máquina de contención representa todas las máquinas privadas, distinguiéndolas de los servidores públicos de la DMZ.

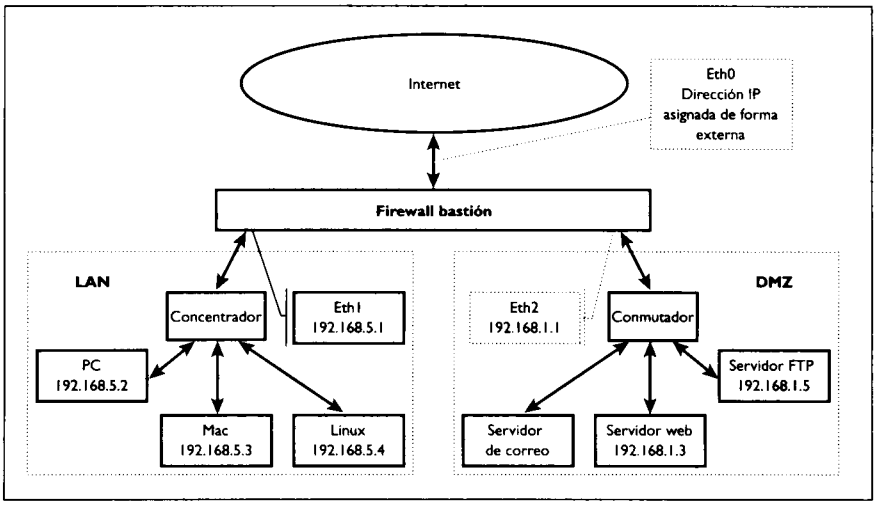

Figura 4.4. Una red DMZ independiente de una LAN privada.

Al contrario que el ejemplo de firewall del Capítulo 3, en esta configuración no existe ni un solo punto de posible error. Se pueden albergar servicios con diferentes directivas de seguridad y en diferentes zonas de seguridad dentro de las redes internas.

La idea principal es aislar físicamente la LAN privada de la máquina firewall bastión externa usando un firewall de contención interno. La red de perímetro que se muestra en la Figura 4.5 no tiene una red completa con sus propios servidores; es algo ideal. La red de perímetro se puede implementar de forma tan sencilla como un cable cruzado entre la interfaz interna del bastión y la interfaz externa de la máquina de contención.

Implementar una DMZ como un simple cable cruzado puede parecer algo ridículo. Merece la pena colocar dos firewalls en un sitio pequeño, como lo hace una red de perímetro. La instalación de dos firewalls no significa que no puedan existir errores. Los servicios de LAN locales se encuentran en la máquina de contención, completamente aislada del bastión o de Internet.

Usando un cable cruzado, la interfaz interna del bastión no necesita un conjunto completo de reglas de firewall independientes. Debería ser suficiente la interfaz externa de la máquina de contención.

El resto de este capítulo supone que el bastión y las máquinas de contención sirven como pasarelas a la red DMZ. La DMZ contiene servidores públicos y semipúblicos. Cada una de las interfaces de red de las dos máquinas firewall tienen sus propios conjuntos de reglas personalizados.

Esta configuración usa un mínimo de cuatro conjuntos de reglas de firewall, una para cada interfaz interna y externa de ambas máquinas firewall. Las reglas de la interfaz externa del firewall bastión son idénticas a las reglas de la interfaz externa del ejemplo del Capítulo 3 (las reglas de la interfaz interna del firewall de contención pueden ser idénticas a las reglas de la interfaz interna del ejemplo con doble tarjeta que se muestra en el Capítulo 3, aunque se han extendido ligeramente para ilustrar un servidor DHCP interno).

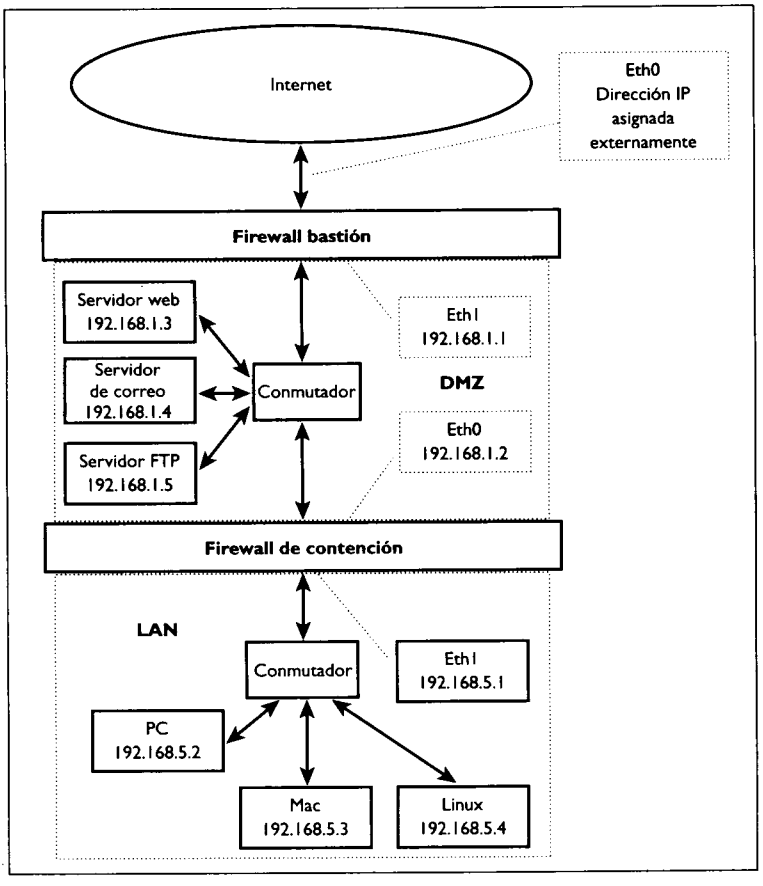

Figura 4.5. Una red de perímetro y una subred explorada detrás de un firewall de contención.

La diferencia real entre este ejemplo y el ejemplo del Capítulo 3 es la adición de la red de perímetro DMZ y las nuevas reglas que se aplican a la interfaz interna del bastión y a la interfaz externa del firewall. Las reglas para

estas dos interfaces son imágenes reflejadas una de la otra. Los servidores públicos en la red DMZ también tienen sus propias reglas de firewall personalizadas. Los servidores públicos de la DMZ suelen ser máquinas especializadas y dedicadas de un solo servicio. Las reglas de firewall son simples y muy restrictivas para el servicio concreto que albergan.

Las constantes simbólicas y las reglas iniciales del firewall de contención son muy parecidas a las del bastión. Como las reglas externas del bastión permanecen igual que en el ejemplo del Capítulo 3, este capítulo resalta las diferencias, si existen, entre los conjuntos de reglas de la interfaz externa. En este ejemplo se resaltan especialmente las reglas del firewall de contención y la simetría entre la interfaz interna del bastión y la interfaz externa de la máquina de contención.

Constantes simbólicas que se usan en los ejemplos de firewalls

Como en el ejemplo del firewall del Capítulo 3, la interfaz externa del bastión se asigna a eth0, cuando se dirigen a Internet. La dirección de la interfaz interna es 192.168.1.1 en eth1, dirigiéndose a la red de perímetro DMZ. La dirección externa de la máquina de contención es 192.168.1.2 en eth0, cuando se dirigen a la red de perímetro DMZ. La dirección de la interfaz interna de la máquina de contención es 192.168.5.1 en eth1, cuando se dirigen a la LAN privada.

Símbolos para el firewall bastión público

Las siguientes constantes se agregan en la secuencia de comandos del firewall del bastión para hacer referencia a la interfaz interna del bastión, la dirección IP de la máquina de contención y el bloque de direcciones de red para la DMZ:

```
BASTION_DMZ_INTERFACE="eth1"       # dirección interna de difusión
BASTION_DMZ_IPADDR="192.168.1.1"   # dirección interna de difusión
CHOKE_DMZ_IPADDR="192.168.1.2"     # firewall de contención interno
DMZ_ADDRESSES="192.168.1.0/24"     # su intervalo (privado)
DMZ_BROADCAST="192.168.1.255"      # dirección interna de difusión
```

Símbolos para el firewall de contención privado

Resulta más sencilla de mantener y leer una secuencia de comandos del shell del firewall si se usan constantes simbólicas para nombres y direcciones recurrentes. Las siguientes constantes se usan en los ejemplos de este capítulo o son constantes universales definidas en los estándares de redes:

```
CHOKE_DMZ_INTERFACE="eth0"          # cualquiera que se use
CHOKE_LAN_INTERFACE="eth1"          # cualquiera que se use
LOOPBACK_INTERFACE="lo"             # sea cual sea el nombre
                                    # que le asigne el sistema
```

```
CHOKE_DMZ_IPADDR="192.168.1.2"          # dirección de interfaz
                                        # externa
BASTION_DMZ_IPADDR="192.168.1.1"        # firewall bastión
DMZ_ADDRESSES="192.168.1.0/24"          # su intervalo (privado)
DMZ_BROADCAST="192.168.1.255"           # dirección de difusión
                                        # externa
CHOKE_LAN_IPADDR="192.168.5.1"          # su dirección de interfaz
                                        # externa
CHOKE_LAN_ADDRESSES="192.168.5.0/24"    # su intervalo (privado)
ANYWHERE="any/0"                        # coincidir con cualquier
                                        # dirección IP

LOOPBACK="127.0.0.0/8"                  # intervalo reservado de
                                        # direcciones de bucle
                                        # invertido
CLASS_A="10.0.0.0/8"                    # redes privadas de clase A
CLASS_B="172.16.0.0/12"                 # redes privadas de clase B
CLASS_C="192.168.0.0/16"                # redes privadas de clase C
CLASS_D_MULTICAST="224.0.0.0/4"         # direcciones de
                                        # multidifusión de clase D
CLASS_E_RESERVED_NET="240.0.0.0/5"      # direcciones reservadas
                                        # de clase E
BROADCAST_SRC="0.0.0.0"                 # dirección origen de
                                        # difusión
BROADCAST_DEST="255.255.255.255"        # dirección destino de
                                        # difusión
PRIVPORTS="0:1023"                      # bien conocido,
                                        # privilegiado
                                        # intervalo de puerto
UNPRIVPORTS="1024:65535"                # intervalo de puerto no
                                        # privilegiado
```

Las constantes que no aparecen aquí se definen dentro del contexto de las reglas específicas con las que se usan.

Cómo quitar cualquier regla que exista previamente del firewall de contención

Lo primero que se debe hacer cuando se define un conjunto de reglas de filtrado es quitar cualquier regla existente de las cadenas de reglas. De lo contrario, cualquier regla nueva que se defina se agregará al final de las reglas existentes. Los paquetes pueden coincidir fácilmente con una regla que exista antes incluso de alcanzar el punto de la cadena a partir del que se comienza a definir. El siguiente comando elimina de una sola vez las reglas de las tres cadenas internas, input, output y forward:

```
# Eliminar cualquier cadena existente de todas las cadenas
ipchains -F
```

Las cadenas están vacías. Se parte de cero y el sistema está en el estado de directiva predeterminado aceptar todo.

Cómo definir la directiva predeterminada del firewall de contención

La directiva de la máquina de contención es rechazar todo el tráfico en cualquier dirección. Se devuelve un mensaje de error ICMP 3. Esto provoca que se envíe inmediatamente cierto tipo de mensaje de error, en lugar de forzar al cliente local a esperar un cierto tiempo.

Los dos firewalls descartan todo de forma predeterminada, en vez de aceptar todo de forma predeterminada:

```
# Establecer la directiva predeterminada a rechazar
ipchains -P input   REJECT
ipchains -P output  REJECT
ipchains -P forward REJECT
```

En este momento, todo el tráfico de red está bloqueado.

Cómo habilitar la interfaz de bucle invertido de la máquina de contención

Es necesario habilitar el tráfico de bucle invertido no restringido. Esto permite ejecutar cualquier servicio de red local, o de los que dependa el sistema, sin tener que preocuparse de conseguir todas las reglas de firewall especificadas:

```
# Tráfico ilimitado en la interfaz de bucle invertido
ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT
```

Usurpamiento de dirección origen y otras direcciones incorrectas

Esta sección establece algunos filtros basándose en las direcciones origen y destino. Estas direcciones nunca serán visibles para un paquete legítimo en Internet.

En el nivel de filtrado de paquetes, uno de los pocos casos de usurpamiento de dirección origen que se puede identificar con certeza como una falsificación es su propia dirección IP. Esta regla deniega los paquetes entrantes que dicen proceder de su dirección:

```
# Rechazar los paquetes usurpados que dicen proceder de la
# dirección de la interfaz
ipchains -A input  -i $CHOKE_DMZ_INTERFACE \
         -s $CHOKE_DMZ_IPADDR -j REJECT -l
ipchains -A input  -i $CHOKE_LAN_INTERFACE \
         -s $CHOKE_LAN_IPADDR -j REJECT -l
```

Los dos conjuntos de reglas siguientes rechazan los paquetes entrantes y salientes con cualquier clase de dirección de red privada de las clases A o B como direcciones origen o destino. Ninguno de estos paquetes debe verse en el exterior de una red privada:

```
# Rechazar paquetes que digan ser o proceder de una red privada
# de clase A
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -s $CLASS_A -j REJECT -l
ipchains -A output -i $CHOKE_DMZ_INTERFACE -d $CLASS_A -j REJECT -l

ipchains -A input  -i $CHOKE_LAN_INTERFACE -s $CLASS_A -j REJECT -l
ipchains -A output -i $CHOKE_LAN_INTERFACE -d $CLASS_A -j REJECT -l

# Rechazar paquetes que digan ser o proceder de una red privada
# de clase B
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -s $CLASS_B -j REJECT -l
ipchains -A output -i $CHOKE_DMZ_INTERFACE -d $CLASS_B -j REJECT -l

ipchains -A input  -i $CHOKE_LAN_INTERFACE -s $CLASS_B -j REJECT -l
ipchains -A output -i $CHOKE_LAN_INTERFACE -d $CLASS_B -j REJECT -l
```

Las direcciones de red privada de la clase C se usan localmente. El firewall de contención no puede bloquear todas las direcciones privadas de clase C, porque tanto su interfaz externa como su interfaz interna tienen asignadas direcciones de clase C (ipfwadm no tiene la flexibilidad de bloquear todas las direcciones privadas de clase C, exceptuando la suya propia. ipchains dispone de esta flexibilidad usando cadenas definidas por el usuario).

Los dos conjuntos siguientes de reglas rechazan los paquetes con una dirección origen reservada para la interfaz de bucle invertido:

```
# Rechazar paquetes que digan ser o proceder de una interfaz de
# bucle invertido
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -s $LOOPBACK -j REJECT -l
ipchains -A input  -i $CHOKE_LAN_INTERFACE -s $LOOPBACK -j REJECT -l
```

Los dos conjuntos siguientes de reglas sirven principalmente para registrar los paquetes que coincidan. La directiva predeterminada del firewall es rechazar todo. Como tal, se rechazan todas las direcciones de difusión de forma predeterminada y deben habilitarse explícitamente cuando sean necesarias:

```
# Rechazar paquetes de difusión mal formados
ipchains -A input  -i $CHOKE_DMZ_INTERFACE \
         -s $BROADCAST_DEST -j REJECT -l
ipchains -A input  -i $CHOKE_DMZ_INTERFACE \
         -d $BROADCAST_SRC  -j REJECT -l

ipchains -A input  -i $CHOKE_LAN_INTERFACE \
         -s $BROADCAST_DEST -j REJECT -l
ipchains -A input  -i $CHOKE_LAN_INTERFACE \
         -d $BROADCAST_SRC  -j REJECT -l
```

Las direcciones de multidifusión sólo son legales como direcciones destino. Los siguientes pares de reglas deniegan y registran los paquetes de red multidifusión usurpada:

```
# Rechazar las direcciones experimentales y las direcciones
# multidifusión de clase D
ipchains -A input -i $CHOKE_DMZ_INTERFACE -s $MULTICAST -j REJECT -l
ipchains -A output -i $CHOKE_DMZ_INTERFACE -s $MULTICAST -j REJECT -l

ipchains -A input  -i $CHOKE_LAN_INTERFACE -s $MULTICAST -j REJECT -l
ipchains -A output -i $CHOKE_LAN_INTERFACE -s $MULTICAST -j REJECT -l
```

Las siguientes reglas de esta sección deniegan y registran los paquetes que dicen proceder de una red reservada de clase E:

```
# Rechazar las direcciones IP reservadas
ipchains -A input -i $CHOKE_DMZ_INTERFACE -s $RESERVED_NET -j REJECT -l
ipchains -A input -i $CHOKE_LAN_INTERFACE -s $RESERVED_NET -j REJECT -l
```

Cómo filtrar los mensajes de control y estado ICMP

Los mensajes ICMP se controlan de forma diferente cuando está habilitado el enmascaramiento IP. A menos que se habilite el enmascaramiento ICMP en la compilación del núcleo, además del enmascaramiento IP general, sólo se reenvían y enmascaran los mensajes de error ICMP para las conexiones existentes. Si el enmascaramiento ICMP también está habilitado, se pueden usar otros mensajes de control, como los que usan ping y traceroute, para enviarse a Internet desde máquinas enmascaradas internas. En caso contrario, estos mensajes ICMP sólo se pueden enviar entre host locales de la misma red. El enmascaramiento ICMP está habilitado de forma predeterminada en la versión 6.0 de Red Hat. También se puede compilar explícitamente el núcleo en versiones anteriores.

Igual que con el enmascaramiento general, si está habilitado el enmascaramiento ICMP, las máquinas locales pueden iniciar interacciones ICMP con máquinas externas, pero las máquinas externas no pueden iniciar un mensaje ICMP a ninguna de las máquinas locales. Sólo es visible directamente la interfaz externa en la máquina bastión.

Mensajes de control Source Quench (Tipo 4)

El mensaje ICMP de tipo 4, Source Quench, se envía cuando un origen de conexión, normalmente un enrutador, está enviando datos más rápidamente de lo que puede controlar el enrutador destino. Source Quench se usa como una forma primitiva de control de flujo en el nivel de red IP, normalmente entre dos máquinas punto a punto adyacentes.

Configuración DMZ del control Source Quench del bastión

Las reglas del bastión para aceptar todos los mensajes Source Quench son:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $BASTION_DMZ_IPADDR 4 -d $DMZ_ADDRESSES -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $DMZ_ADDRESSES 4 -d $BASTION_DMZ_IPADDR -j ACCEPT
```

Configuración DMZ del control Source Quench de contención

Las reglas de contención para aceptar todos los mensajes Source Quench son:

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $DMZ_ADDRESSES 4 -d $CHOKE_DMZ_IPADDR -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $CHOKE_DMZ_IPADDR 4 -d $DMZ_ADDRESSES -j ACCEPT
```

Mensajes de estado Parameter Problem (Tipo 12)

Los mensajes ICMP de tipo 12, Parameter Problem, se envían cuando se recibe un paquete y contiene datos ilegales o inesperados en el encabezado, o cuando la suma de comprobación del encabezado no coincide con la suma de comprobación generada por la máquina receptora.

Configuración DMZ del control Paremeter Problem de bastión

Las reglas del bastión para aceptar todos los mensajes Parameter Problem son:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $ANYWHERE 12 -d $DMZ_ADDRESSES -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $DMZ_ADDRESSES 12 -d $ANYWHERE -j ACCEPT
```

Configuración DMZ del control de contención Paremeter Problem

Las reglas de contención para aceptar todos los mensajes Parameter Problem son:

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $ANYWHERE 12 -d $CHOKE_DMZ_IPADDR -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $CHOKE_DMZ_IPADDR 12 -d $ANYWHERE -j ACCEPT
```

Mensajes de error Destination Unreachable (Tipo 3)

El mensaje ICMP de tipo 3, Destination Unreachable, es un mensaje general de estado de error.

Configuración DMZ del mensaje de error de bastión Destination Unreachable

Las reglas del bastión para aceptar todos los mensajes Destination Unreachable son:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $ANYWHERE 3 -d $DMZ_ADDRESSES -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $DMZ_ADDRESSES 3 -d $ANYWHERE -j ACCEPT
```

Configuración DMZ del mensaje de error de contención Destination Unreachable

Las reglas de contención para aceptar todos los mensajes Destination Unreachable son:

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $ANYWHERE 3 -d $CHOKE_DMZ_IPADDR -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $CHOKE_DMZ_IPADDR 3 -d $ANYWHERE -j ACCEPT
```

Mensajes de estado Time Exceeded (Tipo 11)

El mensaje ICMP de tipo 11, Time Exceeded, indica una condición de tiempo de espera, o más concretamente, que se ha sobrepasado el máximo número permitido de saltos de un paquete. En las redes actuales, el Time Exceeded entrante suele aparecer con más frecuencia que la respuesta ICMP a una solicitud UDP traceroute saliente.

Configuración DMZ del mensaje de estado de bastión Time Exceeded

Las reglas del bastión para aceptar mensajes Time Exceeded sólo con el firewall de contención son:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $BASTION_DMZ_IPADDR 11 -d $DMZ_ADDRESSES -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $DMZ_ADDRESSES 11 -d $BASTION_DMZ_IPADDR -j ACCEPT
```

Configuración DMZ del mensaje de estado de contención Time Exceeded

Las reglas de contención para aceptar mensajes Time Exceeded sólo con el firewall de bastión son:

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $BASTION_DMZ_IPADDR 11 -d $CHOKE_DMZ_IPADDR -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $CHOKE_DMZ_IPADDR 11 -d $BASTION_DMZ_IPADDR -j ACCEPT
```

Mensajes de control Echo Request (Tipo 8) y Echo Reply (Tipo 0)

ping usa dos tipos de mensajes ICMP. El mensaje de solicitud, Echo Request, es un mensaje de tipo 8. El mensaje de respuesta, Echo Reply, es un mensaje de tipo 0. ping es una herramienta sencilla de análisis de red que data de los primeros tiempos de DARPANET.

Configuración DMZ del bastión ping

El siguiente par de reglas permite a las máquinas de la DMZ hacer ping a cualquier host de Internet:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $DMZ_ADDRESSES 8 -d $ANYWHERE -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $ANYWHERE 0 -d $DMZ_ADDRESSES -j ACCEPT
```

La aproximación que se muestra aquí sólo permite a la máquina bastión hacer ping a las máquinas de la DMZ:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $BASTION_DMZ_IPADDR 8 -d $DMZ_ADDRESSES -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p icmp \
         -s $DMZ_ADDRESSES 0 -d $BASTION_DMZ_IPADDR -j ACCEPT
```

Configuración DMZ de contención ping

El siguiente par de reglas permite a la máquina de contención hacer ping a cualquier host de Internet:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $CHOKE_DMZ_IPADDR 8 -d $ANYWHERE -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $ANYWHERE 0 -d $CHOKE_DMZ_IPADDR -j ACCEPT
```

La aproximación que se muestra aquí sólo permite a las máquinas DMZ hacer ping a la máquina de contención:

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $DMZ_ADDRESSES 8 -d $CHOKE_DMZ_IPADDR -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $CHOKE_DMZ_IPADDR 0 -d $DMZ_ADDRESSES -j ACCEPT
```

Compatibilidad del núcleo con el enmascaramiento ICMP

Si se tiene una LAN y se ejecuta una versión Linux de Red Hat anterior a la versión 6.0, no se envían los mensajes ICMP de ping entre la LAN y las máquinas remotas, incluso aunque esté habilitado el envío IP. Si se quiere usar ping desde una máquina interna, es necesario configurar y compilar el núcleo para incluir el módulo de enmascaramiento ICMP. Se pueden enviar otros mensajes de control y error relacionados con la conexión sin este módulo especial.

Tablas de protocolo de servicio TCP y UDP

En cada una de las siguientes secciones, el protocolo de servicio a nivel de filtrado de paquetes se presenta primero como una tabla de información de estado, seguido por las reglas ipchains que muestran estos estados.
Cada fila de la tabla de protocolo de servicio lista un tipo de paquete involucrado en el intercambio de servicio. Se define una regla de firewall por cada tipo de paquete. La tabla se divide en columnas:

- *Descripción* contiene una breve descripción de si el paquete lo ha originado el cliente o el servidor y el propósito del mismo.
- *Protocolo* es el protocolo de transporte en uso, TCP o UDP, o los mensajes de control del protocolo IP, ICMP.
- *Dirección remota* es la dirección legal, o intervalo de direcciones, que puede contener el paquete en el campo dirección remota.
- *Puerto remoto* es el puerto legal, o intervalo de puertos, que puede contener el paquete en el campo puerto remoto.
- *E/S* describe la dirección del paquete, es decir, si entra al sistema desde una ubicación remota o si sale del sistema a una ubicación remota.
- *Dirección local* es la dirección legal, o intervalo de direcciones, que puede contener el paquete en el campo dirección local.
- *Puerto local* es el puerto legal, o intervalo de puertos, que puede contener el paquete en el campo puerto local.
- Los paquetes del protocolo TCP contienen una columna final, el *indicador TCP*, que define los estados legales SYN-ACK que puede tener el paquete.

La tabla describe los paquetes como entrantes o salientes. Las direcciones y los puertos se describen como remotos o locales, de forma relativa a la interfaz de red de la máquina. Es necesario tener en cuenta que para paquetes entrantes, las direcciones y el puerto remoto hacen referencia a los campos origen en el encabezado del paquete IP. La dirección y el puerto local se refieren a los campos destino en el encabezado del paquete IP. Para paquetes salientes, la dirección y el puerto remoto hacen referencia a los campos destino en el encabezado del paquete IP. La dirección y el puerto locales hacen referencia a los campos origen en el encabezado del paquete IP.
Por último, en las pocas situaciones en las que el protocolo de servicio implica mensajes ICMP, tenga en cuenta que los paquetes ICMP del nivel de red IP no están asociados con el concepto de puerto origen o destino, como en el caso de los paquetes TCP o UDP del nivel de transporte. En su lugar, los paquetes ICMP usan el concepto de un tipo de mensaje de control o de estado. Los mensajes ICMP no se envían a programas conectados a puertos de servicio particulares. En su lugar, los mensajes ICMP se envían de un equipo a otro. Por consiguiente, las pocas entradas de paquetes ICMP que aparecen en las tablas usan la columna puerto origen para contener el tipo de mensaje. Para paquetes ICMP entrantes, la columna puerto origen es la columna puerto remoto. Para paquetes ICMP salientes, la columna puerto origen es la columna puerto local.

Cómo habilitar el servicio DNS (Puerto UDP/TCP 53)

Es necesario el acceso al servicio de nombres de dominio remoto (DNS, Domain Name Service) para el acceso a Internet. El servicio DNS traduce entre nombres de host y sus direcciones IP asociadas. No es posible localizar un host remoto sin el servicio DNS.

El servicio DNS usa un protocolo de comunicaciones que se basa tanto en UDP como en TCP. Los modos de conexión incluyen conexiones cliente a servidor normales, tráfico de igual a igual entre servidores de reenvío y servidores dedicados y conexiones entre servidores dedicados primarios y secundarios.

Un usuario particular tiene varias opciones de configuración del servicio DNS. Si la máquina firewall no ejecuta el demonio named, pero funciona simplemente como un cliente DNS, las máquinas internas se configurarán para que apunten a los servidores de nombres del ISP, igual que lo hace la máquina firewall. Si la máquina firewall ejecuta un servidor de nombres, las máquinas internas se configurarán para apuntar a la interfaz interna de la máquina firewall como su servidor de nombres.

Configuración clásica del servicio DNS para una LAN

Algunos sitios necesitan la seguridad adicional que incluye una configuración clásica del servicio DNS y que oculta la información del host local. Lo interesante de esta configuración es que es sensible, local y personal, y que la información de las cuentas puede almacenarse de forma centralizada en la base de datos interna del servicio DNS.

La idea es que el firewall bastión ejecute su propio servidor DNS para el público. El servidor se configura como el origen autorizado para el sitio, aunque la información es incompleta. El servidor de nombres bastión no sabe nada acerca de las máquinas internas. Los clientes DNS de la máquina bastión no usan el servidor de nombres local. Por el contrario, el archivo /etc/resolv.conf del bastión apunta a la máquina de contención como servidor de nombres para los clientes locales del bastión. Las peticiones entrantes de Internet las maneja el servidor DNS público. Las peticiones locales las maneja el servidor DNS interno de la máquina de contención.

La máquina de contención alberga el servidor DNS real para el sitio. El servidor interno también se configura como el origen autorizado del sitio. En este caso, la información es correcta. Las peticiones locales, peticiones desde los clientes locales del bastión y las peticiones desde la LAN privada, se controlan todas mediante este servidor DNS privado interno. Como se muestra en la Figura 4.6, cuando el servidor no dispone de la información de la búsqueda solicitada, la solicita al servidor de la máquina bastión, lo que se traduce en una petición a un servidor de nombres externo.

Si las secuencias de comandos del firewall usan nombres de host simbólicos en algunos sitios, esta configuración mutuamente dependiente crea un dilema como el del huevo y la gallina cuando se inicializan los servidores. Cuando el bastión se inicia, intenta usar la máquina de contención como servidor de nombres. La máquina de contención intentará usar el bastión como servidor de nombres.

Cuando el bastión se inicia por primera vez, debe usar servidores de nombres externos para las peticiones de cliente. Cuando el servidor de nombres del firewall de contención se está ejecutando, los clientes del bastión pueden usar el servidor bastión en vez de servidores externos.

Configuración DMZ del bastión como servidor de nombres público

El bastión alberga el servidor de nombres público. Los clientes DNS locales que se ejecutan en el bastión usan el servidor de nombres privado que se ejecuta en la máquina de contención. Si el servidor de contención no puede servir una petición, reenvía la petición al servidor que se ejecuta en el bastión, que reenvía una por una las peticiones a los servidores remotos designados. Si ninguno de los servidores remotos responde, el servidor del bastión reenvía la petición como un cliente a servidores de nombre remotos más autorizados.

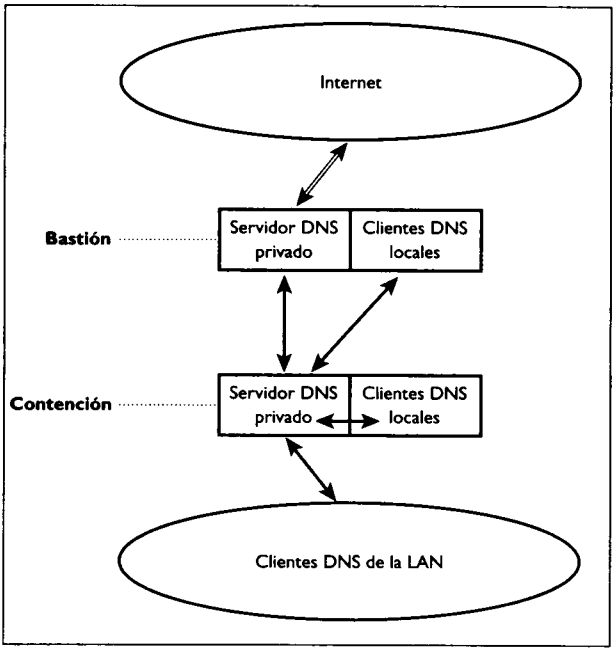

Figura 4.6. Servidores de nombres DNS públicos y privados.

La Tabla 4.2 lista el protocolo DNS que usa la interfaz de la DMZ del bastión.

Tabla 4.2. Protocolo DNS en la interfaz de la DMZ del bastión.

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del servidor de contención	UDP	CHOKE_DIMZ_IPADR	53	E	BASTION-DMZ-IPADDR	53	-
Respuesta del servidor bastión	UDP	CHOKE_DIMZ_IPADR	53	S	BASTION-DMZ-IPADDR	53	-

Tabla 4.2. Protocolo DNS en la interfaz de la DMZ del bastión *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente bastión	UDP	CHOKE_DIMZ_IPADR	53	S	BASTION-DMZ-IPADDR	1024:65535	-
Respuesta del servidor de contención	UDP	CHOKE_DIMZ_IPADR	53	E	BASTION-DMZ-IPADDR	1024:65535	-
Petición del cliente bastión	TCP	CHOKE_DIMZ_IPADR	53	S	BASTION-DMZ-IPADDR	1024:65535	Cualquiera
Respuesta del servidor de contención	TCP	CHOKE_DIMZ_IPADR	53	E	BASTION-DMZ-IPADDR	1024:65535	Ack

El servidor DNS del bastión escucha para recibir peticiones de igual a igual desde el servidor DNS de la máquina de contención:

```
# Servidor de nombres de reenvío y caché DNS (53)
# ----------------------------------

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR 53 \
         -d $BASTION_DMZ_IPADDR 53 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s $BASTION_DMZ_IPADDR 53 \
         -d $CHOKE_DMZ_IPADDR 53 -j ACCEPT
```

Los programas cliente del DNS bastión dirigen las peticiones cliente a servidor al servidor de nombres que se ejecuta en la máquina de contención:

```
# Solicitud local cliente a servidor DNS (53)
# ----------------------------------

ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 53 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR 53 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 53 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE ! -y -p tcp \
         -s $CHOKE_DMZ_IPADDR 53 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Después de estos dos conjuntos de reglas DNS, aparecen las reglas DNS normales para la interfaz externa y los servidores de nombre externos. La Ta-

bla 3.3 del Capítulo 3 muestra la tabla externa de protocolo DNS. El servidor de nombres del bastión se configura como un servidor DNS dedicado, que ofrece servicio DNS público limitado. Para peticiones locales, primero intenta enviar la petición al servidor de nombres del ISP, si existe alguno:

```
# Servidor de nombres de reenvío y caché DNS (53)
# ----------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d $NAME_SERVER 53 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAME_SERVER 53 \
         -d $IPADDR 53 -j ACCEPT


# Solicitud local cliente a servidor DNS (53)
# ----------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 53 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $ANYWHERE 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 53 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE ! -y -p tcp \
         -s $ANYWHERE 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# DNS de servidor local a cliente remoto (53)
# ----------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE ! -y -p tcp \
         -s $IPADDR 53 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ de contención como servidor de nombres privado

La máquina de contención alberga el servidor de nombres privado. Los clientes LAN, incluyendo los que se ejecutan en el bastión, usan el servidor

de nombres privado. Si el servidor de contención no puede dar servicio a una petición desde su caché de DNS, reenvía la petición al servidor que se ejecuta en el bastión.

La Tabla 4.3 lista el protocolo DNS que se usa mediante la interfaz DMZ de la máquina de contención.

Tabla 4.3. Protocolo DNS de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del servidor de contención	UDP	BASTION_DMZ_IP ADDR	53	S	CHOKE_DMZ_IPADDR	53	-
Respuesta del servidor bastión	UDP	BASTION_DMZ_IP ADDR	53	E	CHOKE_DMZ_IPADDR	53	-
Petición del cliente bastión	UDP	BASTION_DMZ_IP ADDR	53	E	CHOKE_DMZ_IPADDR	1024:65535	-
Respuesta del servidor de contención	UDP	BASTION_DMZ_IP ADDR	53	S	CHOKE_DMZ_IPADDR	1024:65535	-
Petición del cliente bastión	TCP	BASTION_DMZ_IP ADDR	53	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor de contención	TCP	BASTION_DMZ_IP ADDR	53	S	CHOKE_DMZ_IPADDR	1024:65535	Ack

Las reglas DNS del firewall de contención son la imagen espejo de las reglas del bastión. El servidor de DNS local reenvía peticiones de servidor de igual a igual al servidor DNS del bastión para peticiones que el servidor de contención no puede resolver:

```
# Servidor de nombres de reenvío y caché DNS (53)
# ----------------------------------

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR 53 \
         -d $DMZ_ADDRESSES 53 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $DMZ_ADDRESSES 53 \
         -d $CHOKE_DMZ_IPADDR 53 -j ACCEPT
```

El servidor DNS de contención recibe las peticiones de cliente desde los programas cliente que se ejecutan en las máquinas conectadas a la DMZ, incluyendo la máquina bastión:

```
# DNS de servidor local a cliente (53)
# ----------------------------------

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 53 -j ACCEPT
```

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR 53 \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 53 -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE ! -y -p tcp \
         -s $CHOKE_DMZ_IPADDR 53 \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT
```

Cómo filtrar el servicio de identificación de usuario AUTH (Puerto TCP 113)

Es necesario el servicio identd, o AUTH, y el servicio de identificación de usuario, para acceder a algunos servicios de Internet. identd proporciona el nombre de usuario o el Id. asociado con una conexión. Es habitual que esto lo solicite un servidor de correo remoto cuando alguien envía un correo electrónico. No es necesario ofrecer el servicio identd, pero sí se debe llevar cuenta de alguna forma de las peticiones entrantes para evitar tiempos de espera prolongados.

Tanto si se decide ofrecer el servicio de identificación de usuario IDENT en la interfaz de Internet externa como si no, no hay razón para no ofrecer el servicio de forma local. Esto significa que se puede habilitar AUTH en el archivo /etc/inetd.conf para ambas máquinas firewall.

Configuración DMZ del bastión AUTH

La Tabla 4.4 lista el protocolo completo de conexión cliente/servidor para el servicio AUTH.

Tabla 4.4. Protocolo del servicio identd en la interfaz de la DMZ del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente bastión	TCP	DMZ_ADDRESSES	113	S	BASTION_DMZ_IP ADDR	1024:65535	Cualquiera
Respuesta del servidor DMZ	TCP	DMZ_ADDRESSES	113	E	BASTION_DMZ_IP ADDR	1024:65535	ACK
Petición del cliente DMZ	TCP	DMZ_ADDRESSES	1024:65535	E	BASTION_DMZ_IP ADDR	113	Cualquiera
Respuesta del servidor bastión	TCP	DMZ_ADDRESSES	1024:65535	S	BASTION_DMZ_IP ADDR	113	ACK

Cómo permitir peticiones salientes AUTH del bastión como cliente

La máquina debe actuar como un cliente AUTH si se ejecuta un servidor de correo o de FTP. No hay razón para no permitir que el sistema sea un cliente AUTH:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $DMZ_ADDRESSES 113 -j ACCEPT

ipchains -A input -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $DMZ_ADDRESSES 113 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo permitir las peticiones AUTH entrantes del bastión como un servidor

Si se ejecuta el servidor de identd fuera del archivo /etc/inetd.conf, las siguientes reglas habilitan las peticiones de conexión identd entrantes:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR 113 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $BASTION_DMZ_IPADDR 113 \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ de contención AUTH

La Tabla 4.5 lista el protocolo completo de conexión cliente/servidor para el servicio AUTH.

Tabla 4.5. Protocolo identd de la interfaz DMZ de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente DMZ	TCP	DMZ_ADDRESSES	1024:65535	E	CHOKE_DMZ_IPADDR	113	Cualquiera
Respuesta del servidor de contención	TCP	DMZ_ADDRESSES	1024:65535	S	CHOKE_DMZ_IPADDR	113	ACK
Petición del cliente de contención	TCP	ANYWHERE	113	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	113	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Cómo permitir peticiones AUTH de contención entrantes como servidor

Si se ejecuta el servidor identd fuera del archivo /etc/inetd.conf, las siguientes reglas habilitan las peticiones de conexión identd entrantes:

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 113 -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR 113 \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT
```

Cómo permitir peticiones AUTH de contención salientes como cliente

La máquina actuará como un cliente AUTH si se ejecuta un servidor de correo o de FTP. No hay razón para no permitir que el sistema sea un cliente de AUTH:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 113 -j ACCEPT

ipchains -A input -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 113 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

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

Independientemente de cuánto correo se envíe y se reciba internamente entre el bastión e Internet, el correo lo suele controlar un servidor central SMTP. Como un ejemplo factible, esta sección supone que el bastión o una máquina de la DMZ es tanto la pasarela de correo local como el host de correo local. Los clientes locales recibirán el correo entrante desde el host de correo que usa un servidor POP o IMAP local.

En esta sección se describen cuatro aproximaciones habituales a las combinaciones de correo electrónico entre cliente y servidor:

- Enviar correo a través del transmisor bastión SMTP y recibir correo como cliente POP de bastión.
- Enviar correo a través del transmisor bastión SMTP y recibir correo como cliente IMAP del bastión.
- Enviar correo a través de un transmisor SMTP de la DMZ y recibir correo como cliente POP de la DMZ.
- Enviar correo a través de un transmisor SMTP de la DMZ y recibir correo como cliente IMAP de la DMZ.

Las dos primeras aproximaciones son útiles si el bastión hace de transmisor de correo para el servidor SMTP del ISP o para el destino receptor final. En cualquier caso, la búsqueda DNS no la realiza la máquina local que la originó. Las otras dos aproximaciones usan una pasarela de correo y un host de la DMZ. Ni un servidor POP ni un servidor IMAP se ejecutan en el bastión para obtener el correo local.

Cómo transmitir el correo saliente a través de la interfaz de la DMZ del bastión (Puerto TCP 25)

Cuando se transmite el correo saliente por medio de un transmisor a través de un servidor de pasarela remoto, el programa de correo cliente envía todo el correo saliente al servidor de correo del ISP. El ISP actúa como pasarela de correo al resto del mundo. El sistema no necesita saber cómo localizar el destino del correo o las rutas hacia ellos. La pasarela de correo ISP sirve como transmisor.

De igual forma, se puede ignorar el servidor de correo del ISP y albergar uno propio. El servidor local es responsable de recopilar el correo saliente, realizar la búsqueda de DNS sobre el nombre del host destino y transmitir el tráfico al destino. El programa de correo cliente apunta al servidor local en vez de al servidor del ISP.

Si se transmite el correo saliente como un servidor SMTP y se recibe el correo como un cliente SMTP, entonces se ofrecen servicios propios de correo. El demonio local sendmail está configurado para transmitir el correo a los host destino remotos por sí mismo, al igual que para recoger y entregar el correo entrante.

Configuración DMZ del servidor SMTP del bastión

La Tabla 4.6 lista el protocolo de conexión cliente/servidor para SMTP.

Tabla 4.6. Configuración DMZ del protocolo de correo SMTP del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Recibiendo correo entrante para transmisión	TCP	DMZ_ADDRESSES	1024:65535	E	BASTION_DMZ_IPADDR	25	Cualquiera
Respuesta del servidor bastión	TCP	DMZ_ADDRESSES	1024:65535	S	BASTION_DMZ_IPADDR	25	ACK

Este conjunto de reglas permite hacer de transmisor del correo saliente para las máquinas locales:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR 25 -j ACCEPT
```

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $BASTION_DMZ_IPADDR 25 \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ del cliente SMTP de contención

La Tabla 4.7 lista el protocolo de conexión cliente/servidor para SMTP.

Tabla 4.7. Configuración DMZ del protocolo de correo SMTP de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Enviar correo saliente	TCP	BASTION_DMZ_IP ADDR	25	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor bastión	TCP	BASTION_DMZ_IP ADDR	25	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Este conjunto de reglas permite enviar correo desde máquinas locales:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR 25 -j ACCEPT

ipchains -A input -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $BASTION_DMZ_IPADDR 25 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo recuperar correo como cliente POP mediante la interfaz DMZ del bastión (Puerto TCP 110)

La forma de recibir correo depende de la situación. Si se ejecuta un servidor propio de correo local, se puede recuperar el correo entrante directamente en la máquina Linux. Si se recibe el correo desde la cuenta del ISP, puede que se reciba o no correo como cliente POP o IMAP, dependiendo de cómo se haya configurado la cuenta de correo electrónico del ISP y de los servicios de entrega de correo que ofrezca el ISP.

De igual forma, si se prefiere mantener la cuenta de correo local relativamente privada y usar la cuenta de trabajo o de correo del ISP como dirección pública, se pueden configurar las cuentas de trabajo y de correo del ISP para enviar el correo al servidor local.

Conectarse a un servidor POP es un medio muy habitual de obtener correo desde un host de correo remoto. El siguiente ejemplo muestra las reglas de firewall necesarias para recuperar correo desde un servidor POP que se ejecuta en el bastión.

Configuración DMZ del servidor POP del bastión

La Tabla 4.8 lista el protocolo del servidor del bastión para el servicio POP.

Tabla 4.8. Configuración DMZ del protocolo de recuperación de correo POP de bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	BASTION_DMZ_IPADDR	110	Cualquiera
Respuesta del servidor bastión	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	BASTION_DMZ_IPADDR	110	ACK

El bastión alberga un servidor POP local para la LAN. No se permite el acceso externo sobre la interfaz externa:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR 110 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $BASTION_DMZ_IPADDR 110 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ del cliente POP de contención

La Tabla 4.9 lista el protocolo del cliente de contención para el servicio POP.

Tabla 4.9. Configuración DMZ del protocolo de recuperación de correo POP de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	BASTION_DMZ_IPADDR	110	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor bastión	TCP	BASTION_DMZ_IPADDR	110	S	CHOKE_DMZ_IPADDR	1024:65535	ACK

El correo se recupera desde el servidor POP que se ejecuta en el bastión:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR 110 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $BASTION_DMZ_IPADDR 110 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo recuperar correo como cliente IMAP mediante la interfaz DMZ del bastión (Puerto TCP 143)

Conectarse a un servidor IMAP es otra manera habitual de recuperar correo desde un host de correo remoto. El siguiente ejemplo muestra las reglas de firewall necesarias para recuperar correo desde un servidor IMAP que se ejecuta en el bastión.

Configuración DMZ del servidor IMAP del bastión

El bastión alberga un servidor IMAP local para la LAN. No se permite el acceso externo sobre la interfaz externa.

La Tabla 4.10 lista el protocolo de servidor del bastión para el servicio IMAP.

Tabla 4.10. Configuración DMZ del protocolo de recuperación de correo IMAP del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ _IPADDR	1024:65535	E	BASTION_DMZ_IPADDR	143	Cualquiera
Respuesta del servidor bastión	TCP	CHOKE_DMZ _IPADDR	1024:65535	S	BASTION_DMZ_IPADDR	143	ACK

El siguiente conjunto de reglas permite las conexiones de cliente entrantes desde la máquina de contención:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR 143 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $BASTION_DMZ_IPADDR 143 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ del cliente IMAP de contención

La Tabla 4.10 lista el protocolo de conexión del cliente de contención para IMAP.

Tabla 4.11. Configuración DMZ del protocolo de recuperación de correo IMAP de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	BASTION_DMZ_I PADDR	143	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor bastión	TCP	BASTION_DMZ_I PADDR	143	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

El correo se recupera desde el servidor IMAP que se ejecuta en el bastión:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR 143 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $BASTION_DMZ_IPADDR 143 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Transmitir correo saliente mediante un servidor SSMTP de la DMZ (Puerto TCP 25)

Existen posibles ventajas de seguridad a la hora de albergar servicios de correo local desde una máquina en la DMZ, en vez de en la máquina bastión. El servidor local es el responsable de recuperar el correo saliente, realizar la búsqueda DNS sobre el nombre de host destino y transmitir el correo al destino. El programa de correo cliente apunta al servidor SMTP local.

Si se transmite correo como un servidor SMTP y se recibe como cliente SMTP, entonces se ofrecen servicios de correo propio. El demonio local sendmail está configurado para transmitir el correo saliente a los host destino remotos por sí mismo, al igual que para recuperar y entregar el correo entrante:

```
MAIL_DMZ_INTERFACE="eth0"
MAIL_SERVER_DMZ_IPADDR="192.168.1.4"
```

Configuración DMZ del servidor SMTP

La Tabla 4.12 lista el protocolo completo de conexión cliente/servidor para SMTP.

Tabla 4.12. Protocolo de correo SMTP en un servidor DMZ

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Recibiendo correo entrante	TCP	ANYWHERE	1024:65535	E	MAIL_SERVER_DMZ_IPADDR	25	Cualquiera
Respuesta del servidor DMZ	TCP	ANYWHERE	1024:65535	S	MAIL_SERVER_DMZ_IPADDR	25	ACK
Enviar correo saliente	TCP	ANYWHERE	25	S	MAIL_SERVER_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	25	E	MAIL_SERVER_DMZ_IPADDR	1024:65535	ACK

El primer conjunto de reglas del servidor de correo DMZ supone que la máquina bastión se ejecuta como servidor de correo que recupera automáticamente el correo interno desde Internet al host de correo DMZ o envía las

conexiones entrantes directamente al servidor de la DMZ. Esto permite recibir correo entrante para las máquinas locales:

```
ipchains -A input  -i $MAIL_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $MAIL_SERVER_DMZ_IPADDR 25 -j ACCEPT

ipchains -A output -i $MAIL_DMZ_INTERFACE -p tcp ! -y \
         -s $MAIL_SERVER_DMZ_IPADDR 25 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

El siguiente conjunto de reglas permite al servidor de correo DMZ transmitir el correo local a los destinos remotos:

```
ipchains -A output -i $MAIL_DMZ_INTERFACE -p tcp \
         -s $MAIL_SERVER_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 25 -j ACCEPT

ipchains -A input  -i $MAIL_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 25 \
         -d $MAIL_SERVER_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Los conjuntos de reglas anteriores del servidor de correo DMZ también abarcan el caso en el que la máquina bastión no está ejecutando un servidor de transmisión de correo. Por el contrario, el bastión usa ipmasqadm portfw para enviar las conexiones de correo entrantes al servidor DMZ. Esto permite recibir directamente el correo entrante para las máquinas locales:

```
ipchains -A input  -i $MAIL_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $MAIL_SERVER_DMZ_IPADDR 25 -j ACCEPT

ipchains -A output -i $MAIL_DMZ_INTERFACE -p tcp ! -y \
         -s $MAIL_SERVER_DMZ_IPADDR 25 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ del cliente SMTP de contención

La Tabla 4.13 lista el protocolo de conexión del cliente de contención para SMTP.

Tabla 4.13. Protocolo de correo SMTP de contención en un cliente DMZ

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Enviar correo saliente	TCP	MAIL_SERVER_DMZ_IPADDR	25	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	MAIL_SERVER_DMZ_IPADDR	25	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Esto permite enviar correo desde máquinas locales:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $MAIL_SERVER_DMZ_IPADDR 25 -j ACCEPT

ipchains -A input -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $MAIL_SERVER_DMZ_IPADDR 25 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ del conducto SMTP del bastión

La Tabla 4.14 lista el protocolo de conexión cliente/servidor para SMTP.

Tabla 4.14. Protocolo de correo SMTP del bastión sobre un conducto DMZ

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Recibiendo correo entrante	TCP	ANYWHERE	1024:65535	E	IPADDR	25	Cualquiera
Respuesta del servidor DMZ	TCP	ANYWHERE	1024:65535	S	IPADDR	25	ACK
Enviar correo saliente	TCP	ANYWHERE	25	S	IPADDR	1024:65535	Cualquiera
Repuesta del servidor remoto	TCP	ANYWHERE	25	E	IPADDR	1024:65535	ACK

Esta sección abarca el caso en el que la máquina bastión no ejecuta un servidor de transmisión de correo. Por el contrario, el bastión usa ipmasqadm ipportfw para enviar las conexiones de correo entrante al servidor DMZ.

Los dos primeros conjuntos de reglas permiten conexiones de correo entrantes al servidor de correo que se ejecuta en la DMZ:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 25 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 25 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $MAIL_SERVER_DMZ_IPADDR 25 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $MAIL_SERVER_DMZ_IPADDR 25 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipmasqadm portfw -a -P tcp -L $IPADDR 25 -R $MAIL_SERVER_DMZ_IPADDR 25
```

Los siguientes conjuntos de reglas permiten al servidor de correo DMZ transmitir correo a los destinos remotos:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $MAIL_SERVER_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 25 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 25 \
         -d $MAIL_SERVER_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 25 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipmasqadm portfw -a -P tcp -L 192.168.1.4 25 -R $IPADDR 25
```

Cómo recuperar correo como cliente POP a un servidor DMZ (Puerto TCP 110)

Conectarse a un servidor POP es una forma muy habitual de recuperar correo desde un host de correo remoto. El siguiente ejemplo muestra las reglas de firewall necesarias para recuperar correo desde un servidor POP que se ejecuta en una máquina situada en la DMZ.

El servidor POP local se ejecuta en la misma máquina de la DMZ que alberga el servidor de correo local:

```
POP_DMZ_INTERFACE="eth0"
POP_SERVER_DMZ_IPADDR="192.168.1.4"
```

Configuración DMZ del servidor POP

La Tabla 4.15 lista el protocolo de conexión del servidor para un servidor POP.

Tabla 4.15. Protocolo del servidor POP desde un servidor DMZ

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ_IP ADDR	1024:65535	E	POP_SERVER_DMZ_IP ADDR	110	Cualquiera
Respuesta del servidor local	TCP	CHOKE_DMZ_IP ADDR	1024:65535	S	POP_SERVER_DMZ_IP ADDR	110	ACK

La máquina servidor de correo de la DMZ alberga un servidor POP local para la LAN. No se permite el acceso externo sobre la interfaz externa:

```
ipchains -A input  -i $POP_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $POP_SERVER_DMZ_IPADDR 110 -j ACCEPT

ipchains -A output -i $POP_DMZ_INTERFACE -p tcp ! -y \
         -s $POP_SERVER_DMZ_IPADDR 110 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ del cliente POP de contención

La Tabla 4.16 lista el protocolo de conexión del cliente para un servidor POP.

Tabla 4.16. Protocolo de recuperación de correo POP desde un servidor DMZ de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	POP_SERVER_DMZ_IPADDR	110	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor DMZ	TCP	POP_SERVER_DMZ_IPADDR	110	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Se recupera el correo desde el servidor POP que se ejecuta en el servidor de la DMZ:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $POP_SERVER_DMZ_IPADDR 110 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $POP_SERVER_DMZ_IPADDR 110 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo recuperar correo como cliente IMAP a un servidor DMZ (Puerto TCP 143)

Conectarse con un servidor IMAP es un medio muy habitual de obtener correo desde un host de correo remoto. El siguiente ejemplo muestra las reglas de firewall necesarias para obtener correo desde un servidor IMAP que se ejecuta en una máquina de la DMZ.

El servidor IMAP local se ejecuta en la misma máquina de la DMZ que alberga el servidor de coreo local:

```
IMAP_DMZ_INTERFACE="eth0"
IMAP_SERVER_DMZ_IPADDR="192.168.1.4"
```

Configuración DMZ del servidor IMAP

La Tabla 4.17 lista el protocolo de conexión del servidor DMZ para IMAP.

Tabla 4.17. Protocolo del servidor IMAP desde un servidor DMZ

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	IMAP_SERVER_DMZ_IPADDR	143	Cualquiera
Respuesta del servidor DMZ	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	IMAP_SERVER_DMZ_IPADDR	143	ACK

La máquina servidor de correo de la DMZ alberga un servidor local para la LAN. No se permite el acceso externo en la interfaz externa:

```
ipchains -A input  -i $IMAP_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $IMAP_SERVER_DMZ_IPADDR 143 -j ACCEPT

ipchains -A output -i $IMAP_DMZ_INTERFACE -p tcp ! -y \
         -s $IMAP_SERVER_DMZ_IPADDR 143 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ del cliente IMAP de contención

La Tabla 4.18 lista el protocolo de conexión de cliente de contención para IMAP.

Tabla 4.18. Protocolo de recuperación de correo IMAP desde un servidor DMZ de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	IPMAP_SERVER_DMZ_IPADDR	143	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor DMZ	TCP	IPMAP_SERVER_DMZ_IPADDR	143	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

El correo se recupera desde el servidor IMAP que se ejecuta en el servidor de la DMZ:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $IMAP_SERVER_DMZ_IPADDR 143 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $IMAP_SERVER_DMZ_IPADDR 143 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Cómo acceder a servicios de noticias de Usenet (Puerto TCP NNTP 119)

Se accede a las noticias de Usenet sobre el protocolo NNTP, que se ejecuta en la parte superior del protocolo TCP a través del puerto de servicio 119. La lectura de noticias y la colocación de artículos se controla mediante el cliente de noticias local.

No es probable que un sitio que no sea un ISP albergue un servidor de noticias para el mundo exterior. Ni siquiera es probable que un sitio pequeño albergue un servidor de noticias local. Como excepción, es necesario configurar la regla de servidor para permitir conexiones entrantes sólo desde un conjunto selecto de clientes externos. No es probable albergar un servidor público en la máquina bastión externa. Es más probable que un servidor de noticias se ejecute en una máquina de la DMZ y que el bastión envíe las conexiones NNTP entrantes al servidor interno.

Configuraciones del conducto NNTP del bastión y del servidor DMZ

La Tabla 4.19 lista el protocolo completo de conexión cliente/servidor para el servicio de noticias de Usenet NNTP.

Tabla 4.19. Protocolo de conducción del cliente/servidor NNTP del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente DMZ	TCP	NEWS_SERVER	119	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	NEWS_SERVER	119	S	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente remoto	TCP	Clientes NNTP	1024:65535	S	NEWS_SERVER_DMZ_IPADDR	119	Cualquiera
Respuesta del servidor DMZ	TCP	Clientes NNTP	1024:65535	E	NEWS_SERVER_DMZ_IPADDR	119	ACK
Petición del servidor local	TCP	NEWS_FEED	119	S	NEWS_SERVER_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	NEWS_FEED	119	E	NEWS_SERVER_DMZ_IPADDR	1024:65535	ACK

Las reglas de servidor permiten conexiones locales al servidor de noticias del ISP. Tanto la lectura de noticias como la colocación de artículos se maneja mediante este conjunto de reglas:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $NEWS_SERVER 119 -j ACCEPT
```

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $NEWS_SERVER 119 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Si se ejecuta un servidor de noticias local en una máquina de la DMZ, ofreciendo servicio público a clientes remotos concretos, debe definirse un conjunto de reglas de servidor que permita a los clientes locales conectarse a esta puerto NNTP de la máquina:

```
NEWS_SERVER_DMZ_IPADDR="192.168.1.6"

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s <mi.noticias.clientes> $UNPRIVPORTS \
         -d $NEWS_SERVER_DMZ_IPADDR 119 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $NEWS_SERVER_DMZ_IPADDR 119 \
         -d $<mi.noticias.clientes> $UNPRIVPORTS -j ACCEPT
```

Si a las máquinas DMZ se les asigna direcciones IP de clase privada, no es necesario el módulo de reenvío del puerto:

```
ipmasqadm portfw -a -P tcp -L <mi.noticias.clientes> 119 \
                 -R $NEWS_SERVER_DMZ_IPADDR 119
```

Si el servidor de noticias local proporciona grupos de noticias públicos de Usenet, así como grupos de noticias locales, el servidor local necesita recopilar noticias de un servidor remoto. Las siguientes reglas permiten el acceso a un servidor de noticias remoto que actúa como almacén de noticias:

```
NEWS_FEED="<mi.remoto.noticias.almacen>"

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $NEWS_SERVER_DMZ_IPADDR $UNPRIVPORTS \
         -d $NEWS_FEED 119 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $NEWS_FEED 119 \
         -d $NEWS_SERVER_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuraciones DMZ del cliente NNTP de contención

La Tabla 4.20 lista el protocolo completo de conexión cliente/servidor para el servicio de noticias NNTP de Usenet.

Tabla 4.20. Protocolo de cliente NNTP de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	NEWS_SERVER	119	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera

Tabla 4.20. Protocolo de cliente NNTP de contención *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Respuesta del servidor remoto	TCP	NEWS_SERVER	119	E	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente de contención	TCP	NEWS_SERVER_DMZ_IPADDR	119	S	IPADDR	1024:65535	Cualquiera
Respuesta del servidor DMZ	TCP	NEWS_SERVER_DMZ_IPADDR	119	E	IPADDR	1024:65535	ACK

Las dos reglas siguientes permiten a los clientes locales acceder a los servidores de noticias remotos:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS\
         -d $NEWS_SERVER 119 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $NEWS_SERVER 119 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS-j ACCEPT
```

Las dos reglas siguientes permiten a los clientes locales acceder a un servidor de noticias local de la DMZ:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS\
         -d $NEWS_SERVER_DMZ_IPADDR 119 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $NEWS_SERVER_DMZ_IPADDR 119 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS-j ACCEPT
```

Telnet (Puerto TCP 23)

Dependiendo del nivel de confianza, es probable que se llegue a la conclusión de que es perfectamente aceptable el servicio telnet interno. Dependiendo del tipo de máquinas no UNIX situadas en el otro extremo de la LAN, puede que telnet sea la única herramienta disponible.

Las reglas de cliente y de servidor permiten tener acceso a y desde cualquier sitio. Si se usa el programa telnet, se podrán restringir las direcciones externas a un subconjunto concreto. Si eso no resulta práctico, el programa telnet se debería restringir en el archivo /etc/hosts.allow.

Configuración DMZ de Telnet del bastión

La Tabla 4.21 lista el protocolo de conexión completo cliente/servidor para el servicio Telnet.

Tabla 4.21. Protocolo Telnet del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	ANYWHERE	23	Cualquiera
Respuesta del servidor	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	ANYWHERE	23	ACK
Petición del cliente bastión	TCP	DMZ_ADDRESSES	23	S	BASTION_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor DMZ	TCP	DMZ_ADRESSES	23	E	BASTION_DMZ_IPADDR	1024:65535	ACK

Observe que las conexiones telnet entrantes desde clientes sólo se permiten desde el firewall de contención, y no desde cualquier lugar. No se permite el acceso telnet desde otras máquinas de la DMZ:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 23 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 23 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Las dos reglas siguientes permiten el acceso telnet desde el bastión a cualquier máquina de la DMZ:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $DMZ_ADDRESSES 23 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $DMZ_ADDRESSES 23 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración DMZ de Telnet de contención

La Tabla 4.22 lista el protocolo completo de conexión cliente/servidor para el servicio Telnet.

Tabla 4.22. Protocolo Telnet de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	23	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera

Tabla 4.22. Protocolo Telnet de la máquina de contención *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Respuesta del servidor remoto	TCP	ANYWHERE	23	E	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente bastión	TCP	BASTION_DMZ_IPADDR	1024:65535	E	CHOKE_DMZ_IPADDR	23	Cualquiera
Respuesta del servidor de contención	TCP	BASTION_DMZ_IPADDR	1024:65535	S	CHOKE_DMZ_IPADDR	23	ACK

Tenga en cuenta que se permiten las conexiones telnet salientes a cualquier lugar, ofreciendo este servicio a clientes LAN que necesitan hacer telnet a sitios remotos en Internet:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 23 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 23 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Estas reglas permiten el acceso telnet desde el bastión a la máquina firewall de contención. No se permite el acceso entrante telnet desde otras máquinas de la DMZ:

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 23 -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR 23 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

SSH (Puerto TCP 22)

En este caso, las reglas de cliente y de servidor permiten el acceso a y desde cualquier lugar. En la práctica, es probable que un sitio particular restrinja las direcciones externas a un subconjunto selecto, particularmente teniendo en cuenta que ambos extremos se deben configurar para poder reconocer cada cuenta de usuario individual. Aunque no se recomienda usar inetd para iniciar sshd, se puede compilar sshd para que cumpla las listas de acceso de host en los archivos /etc/hosts.allow y /etc/hosts.deny.

Configuración DMZ del bastión del protocolo de conducción y del servidor SSH

La Tabla 4.23 lista el protocolo de conexión del servidor para el servicio SSH.

Tabla 4.23. Protocolo de servidor SSH de bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	ANYWHERE	22	Cualquiera
Respuesta del servidor	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	ANYWHERE	22	ACK
Petición del cliente de contención	TCP	CHOKE_DMZ_IPADDR	513:1023	E	ANYWHERE	22	Cualquiera
Respuesta del servidor	TCP	CHOKE_DMZ_IPADDR	513:1023	S	ANYWHERE	22	ACK

Estas reglas permiten las conexiones locales desde la máquina de contención tanto a un servidor sshd que se ejecuta en el bastión como a sitios externos:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $SSH_PORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $CHOKE_DMZ_IPADDR $SSH_PORTS -j ACCEPT
```

Configuración DMZ del bastión del protocolo cliente SSH

La Tabla 4.24 lista el protocolo de conexión de cliente para el servicio SSH.

Tabla 4.24. Protocolo del cliente SSH del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente bastión	TCP	DMZ_ADDRESSES	22	S	BASTION_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor DMZ	TCP	DMZ_ADDRESSES	22	E	BASTION_DMZ_IPADDR	1024:65535	ACK
Petición del cliente bastión	TCP	DMZ_ADDRESSES	22	S	BASTION_DMZ_IPADDR	513:1023	Cualquiera
Respuesta del servidor DMZ	TCP	DMZ_ADDRESSES	22	E	BASTION_DMZ_IPADDR	513:1023	ACK

Estas reglas permiten al bastión conectarse a todas las máquinas locales de la DMZ:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $DMZ_ADDRESSES 22 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $DMZ_ADDRESSES 22 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $SSH_PORTS \
         -d $DMZ_ADDRESSES 22 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $DMZ_ADDRESSES 22 \
         -d $BASTION_DMZ_IPADDR $SSH_PORTS -j ACCEPT
```

Configuración DMZ del protocolo cliente SSH de contención

La Tabla 4.25 lista el protocolo de conexión del cliente para el servicio SSH.

Tabla 4.25. Protocolo del cliente SSH de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	22	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	22	E	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente de contención	TCP	ANYWHERE	22	S	CHOKE_DMZ_IPADDR	513:1023	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	22	E	CHOKE_DMZ_IPADDR	513:1023	ACK

Estas reglas permiten conectarse a sitios remotos usando ssh:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $SSH_PORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $CHOKE_DMZ_IPADDR $SSH_PORTS -j ACCEPT
```

Configuración DMZ del protocolo servidor SSH de contención

La Tabla 4.26 lista el protocolo de conexión del servidor para el servicio SSH.

Tabla 4.26. Protocolo del servidor SSH de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente bastión	TCP	BASTION_DMZ_IPADDR	1024:65535	E	CHOKE_DMZ_IPADDR	22	Cualquiera
Respuesta del servidor de contención	TCP	BASTION_DMZ_IPADDR	1024:65535	S	CHOKE_DMZ_IPADDR	22	ACK
Petición del cliente bastión	TCP	BASTION_DMZ_IPADDR	513:1023	E	CHOKE_DMZ_IPADDR	22	Cualquiera
Respuesta del servidor de contención	TCP	BASTION_DMZ_IPADDR	513:1023	S	CHOKE_DMZ_IPADDR	22	ACK

Estas reglas permiten las conexiones entrantes desde el bastión al servidor sshd:

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $BASTION_firewall $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 22 -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR 22 \
         -d $BASTION_firewall $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $BASTION_firewall $SSH_PORTS \
         -d $CHOKE_DMZ_IPADDR 22 -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR 22 \
         -d $BASTION_firewall $SSH_PORTS -j ACCEPT
```

FTP (Puertos TCP 21 y 20)

FTP tiene dos modos para intercambiar datos entre un cliente y un servidor: el modo normal de puerto de canal de datos y el modo pasivo de canal de datos. El modo de puerto es el mecanismo predeterminado cuando se usa el programa cliente ftp y se conecta a un sitio FTP remoto. El modo pasivo es el mecanismo predeterminado cuando se conecta a un sitio FTP usando un navegador web. En algunas ocasiones, se puede encontrar un sitio FTP que sólo sea compatible con un modo o el otro.

En esta sección se describen tres aproximaciones a las combinaciones FTP de cliente y servidor:

- El bastión es una pasarela a servidores FTP remotos, al igual que un posible servidor local; la máquina de contención es un cliente.
- El bastión es un cliente a servidores internos; la maquina de contención es un servidor FTP.
- Un servidor FTP se ejecuta en la DMZ; las máquinas bastión y de contención son clientes.

El bastión como servidor FTP o conducto, la máquina de contención como cliente

Es casi un hecho que la mayoría de los sitios querrán tener acceso de cliente FTP a almacenes de archivos remotos. La primera sección permite a host internos conectarse a servidores FTP tanto en el bastión como en host remotos.

Conexiones entrantes del cliente local al servidor bastión

La Tabla 4.27 lista el protocolo de conexión del servidor para el servicio FTP.

Tabla 4.27. Protocolo del servidor FTP del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	ANYWHERE	21	Cualquiera
Respuesta del servidor bastión	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	ANYWHERE	21	ACK
Petición del puerto de canal de datos del servidor bastión	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	ANYWHERE	20	Cualquiera
Respuesta del puerto de canal de datos del cliente de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	ANYWHERE	20	ACK
Petición del canal de datos pasivo del cliente de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	ANYWHERE	1024:65535	Cualquiera
Respuesta del canal de datos pasivo del servidor bastión	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	ANYWHERE	1024:65535	ACK

Las siguientes reglas permiten las conexiones del cliente ftp desde la máquina de contención, así como desde máquinas situadas en la LAN privada detrás del firewall de contención:

```
# Solicitud entrante de cliente FTP
# ----------------------.

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 21 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 21 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

# Respuestas del canal de datos de modo puerto
# ----------------------------------.

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE 20 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 20 -j ACCEPT

# Respuestas del canal de datos de modo pasivo
# ----------------------------------.

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Conexiones salientes de la máquina de contención al servidor externo

La Tabla 4.28 lista el protocolo de conexión del cliente para el servicio FTP.

Tabla 4.28. Protocolo de cliente FTP de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	21	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	21	E	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del puerto de canal de datos del servidor remoto	TCP	ANYWHERE	20	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera

Tabla 4.28. Protocolo de cliente FTP de la máquina de contención *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Respuesta del puerto de canal de datos del cliente de contención	TCP	ANYWHERE	20	S	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del canal de datos pasivo del cliente de contención	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del canal de datos pasivo del servidor remoto	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones salientes del cliente ftp desde la máquina de contención, al igual que desde máquinas situadas en la LAN privada detrás del firewall de contención:

```
# Petición saliente del cliente FTP
# ----------------.

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 21 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 21 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

# Respuestas del canal de datos de modo puerto
# ----------------------------.

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE 20 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 20 -j ACCEPT

# Respuestas del canal de datos de modo pasivo
# ----------------------------.

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR $UNPIRVPORTS -j ACCEPT
```

Bastión como cliente, contención como servidor FTP interno

Suele ser conveniente para archivos ftp entre el bastión y las máquinas internas. La siguiente sección es un caso especial. La propia máquina bastión dispone de acceso ftp a la máquina de contención como cliente. El acceso cliente desde el bastión no suele permitirse en un entorno estrictamente seguro.

Conexiones salientes del cliente bastión al servidor FTP de contención

La Tabla 4.29 lista el protocolo de conexión de cliente para el servicio FTP.

Tabla 4.29. Protocolo del cliente FTP del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente bastión	TCP	CHOKE_DMZ_IPADDR	21	S	BASTION_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor de contención	TCP	CHOKE_DMZ_IPADDR	21	E	BASTION_DMZ_IPADDR	1024:65535	ACK
Petición del puerto de canal de datos del servidor de contención	TCP	CHOKE_DMZ_IPADDR	20	E	BASTION_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del puerto de canal de datos del cliente bastión	TCP	CHOKE_DMZ_IPADDR	20	S	BASTION_DMZ_IPADDR	1024:65535	ACK
Petición del canal de datos pasivo del cliente bastión	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	BASTION_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del canal de datos pasivo del servidor de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	BASTION_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones salientes del cliente ftp desde la máquina bastión a la máquina de contención:

```
# Petición saliente del cliente FTP
# ---------------------.

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 21 -j ACCEPT
```

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR 21 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

# Respuestas del canal de datos de modo puerto
# ----------------------------------

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR 20 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 20 -j ACCEPT

# Respuestas del canal de datos de modo pasivo
# ----------------------------------

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR $UNPIRVPORTS -j ACCEPT
```

Conexiones entrantes del cliente bastión al servidor de contención

La Tabla 4.30 lista el protocolo de conexión del servidor para el servicio FTP.

Tabla 4.30. Protocolo del servidor FTP de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente bastión	TCP	BASTION_DMZ_IPADDR	1024:65535	E	CHOKE_DMZ_IPADDR	21	Cualquiera
Respuesta del servidor de contención	TCP	BASTION_DMZ_IPADDR	1024:65535	S	CHOKE_DMZ_IPADDR	21	ACK
Petición del puerto de canal de datos del servidor de contención	TCP	BASTION_DMZ_IPADDR	1024:65535	S	CHOKE_DMZ_IPADDR	20	Cualquiera
Respuesta del puerto de canal de datos del cliente bastión	TCP	BASTION_DMZ_IPADDR	1024:65535	E	CHOKE_DMZ_IPADDR	20	ACK

Tabla 4.30. Protocolo del servidor FTP de la máquina de contención *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del canal de datos pasivo del cliente bastión	TCP	BASTION_DMZ_IPADDR	1024:65535	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del canal de datos pasivo del servidor de contención	TCP	BASTION_DMZ_IPADDR	1024:65535	S	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones entrantes del cliente ftp desde la máquina bastión a la máquina de contención:

```
# Solicitud entrante de cliente FTP
# ----------------.

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 21 -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR 21 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

# Respuestas del canal de datos de modo puerto
# --------------------.

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR 20 \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 20 -j ACCEPT

# Respuestas del canal de datos de modo pasivo
# ------------------.

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $BASTION_DMZ_IPADDR $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Servidor FTP de la DMZ, bastión y contención como clientes

Aunque se parece a la sección anterior por permitir a la propia máquina bastión tener acceso cliente a la máquina de contención, la siguiente sección permite las peticiones de cliente entrantes a un servidor FTP público de la DMZ.

Servidor FTP de la DMZ

La Tabla 4.31 lista el protocolo de conexión del servidor para el servicio FTP que se ejecuta en una máquina de la DMZ.

Tabla 4.31. Protocolo de servidor FTP de la DMZ

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente remoto	TCP	DMZ_ADDRESSES	1024:65535	E	FTP_SERVER_DMZ_IPADDR	21	Cualquiera
Respuesta del servidor DMZ	TCP	DMZ_ADDRESSES	1024:65535	S	FTP_SERVER_DMZ_IPADDR	21	ACK
Petición del canal de datos del puerto del servidor DMZ	TCP	DMZ_ADDRESSES	1024:65535	S	FTP_SERVER_DMZ_IPADDR	20	Cualquiera
Respuesta del canal de datos del puerto del cliente remoto	TCP	DMZ_ADDRESSES	1024:65535	E	FTP_SERVER_DMZ_IPADDR	20	ACK
Petición de canal de datos pasivo del cliente remoto	TCP	DMZ_ADDRESSES	1024:65535	E	FTP_SERVER_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta de canal de datos pasivo del servidor DMZ	TCP	DMZ_ADDRESSES	1024:65535	S	FTP_SERVER_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones entrantes del cliente ftp desde cualquier máquina situada en la red DMZ, incluyendo las máquinas bastión y contención:

```
# Solicitud entrante de cliente FTP
# ----------------------

ipchains -A input  -i $DMZ_INTERFACE -p tcp \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $FTP_SERVER_DMZ_IPADDR 21 -j ACCEPT

ipchains -A output -i $DMZ_INTERFACE -p tcp ! -y \
         -s $FTP_SERVER_DMZ_IPADDR 21 \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT

# Respuestas del canal de datos de modo puerto
# ----------------------------------

ipchains -A output -i $DMZ_INTERFACE -p tcp \
         -s $FTP_SERVER_DMZ_IPADDR 20 \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $DMZ_INTERFACE -p tcp ! -y \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $FTP_SERVER_DMZ_IPADDR 20 -j ACCEPT
```

```
# Respuestas del canal de datos de modo pasivo
# ---------------------.

ipchains -A input  -i $DMZ_INTERFACE -p tcp \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $FTP_SERVER_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $DMZ_INTERFACE -p tcp ! -y \
         -s $FTP_SERVER_DMZ_IPADDR $UNPRIVPORTS \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT
```

Máquina de contención como cliente para un servidor FTP de la DMZ

La Tabla 4.32 lista el protocolo de conexión de cliente para el servicio FTP.

Tabla 4.32. Protocolo de cliente FTP de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	FTP_SERVER_DMZ_IPADDR	21	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor DMZ	TCP	FTP_SERVER_DMZ_IPADDR	21	E	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del puerto de canal de datos del servidor DMZ	TCP	FTP_SERVER_DMZ_IPADDR	20	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del puerto de canal de datos del cliente de contención	TCP	FTP_SERVER_DMZ_IPADDR	20	S	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del canal de datos pasivo del cliente de contención	TCP	FTP_SERVER_DMZ_IPADDR	1024:65535	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del canal de datos pasivo del servidor DMZ	TCP	FTP_SERVER_DMZ_IPADDR	1024:65535	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones salientes del cliente ftp desde la máquina de contención a un servidor ftp en cualquier lugar:

```
# Petición saliente del cliente FTP
# ----------------.

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 21 -j ACCEPT
```

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 21 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

# Respuestas del canal de datos de modo puerto
# ----------------------.

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE 20 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 20 -j ACCEPT

# Respuestas del canal de datos de modo pasivo
# --------------------.

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR $UNPIRVPORTS -j ACCEPT
```

Servicios web

Los servicios web se suelen basar en el protocolo HTTP. Se usan varios protocolos de comunicación de alto nivel para propósitos especiales, incluyendo HTTPS para acceso seguro (SSL) y acceso proxy de servidor web.

En esta sección se describen tres aproximaciones a las combinaciones cliente y servidor web:

- El bastión como servidor o una pasarela conducto para servidores web remotos; la máquina de contención como cliente.
- El bastión como servidor o un conducto de pasarela; un servidor web se ejecuta en la DMZ y la máquina de contención es un cliente.
- El bastión como servidor o un conducto de pasarela; la máquina de contención es un servidor proxy web local.

El bastión como servidor web o conducto, la máquina de contención como cliente

Es casi inconcebible actualmente que un sitio particular no quiera acceder a la World Wide Web. La primera sección permite a los host internos conectarse a servidores web tanto en el bastión como en host remotos.

Conexiones de cliente internas entrantes al servidor bastión

La Tabla 4.33 lista el protocolo de conexión de servidor para le servicio web HTTP.

Tabla 4.33. Protocolo de servidor HTTP del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	ANYWHERE	80	Cualquiera
Respuesta del servidor	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	ANYWHERE	80	ACK

Las siguientes reglas permiten las conexiones entrantes de cliente HTTP desde la máquina de contención, así como desde máquinas situadas en la LAN privada detrás del firewall de contención:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 80 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 80 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.34 lista el protocolo de conexión del servidor para el servicio SSL.

Tabla 4.34. Protocolo del servidor SSL del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ_IPADDR	1024:65535	E	ANYWHERE	443	Cualquiera
Respuesta del servidor	TCP	CHOKE_DMZ_IPADDR	1024:65535	S	ANYWHERE	443	ACK

Las siguientes reglas permiten las conexiones de cliente SSL entrantes desde la máquina de contención, así como desde máquinas situadas en la LAN privada detrás del firewall de contención:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 443 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 443 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.35 lista el protocolo de conexión del servidor para el servicio proxy web.

Tabla 4.35. Protocolo del servidor proxy web del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	CHOKE_DMZ_IPADDR	WEB_PROXY_PORT	E	WEB_PROXY_SERVER	1024:65535	Cualquiera
Respuesta del servidor	TCP	CHOKE_DMZ_IPADDR	WEB_PROXY_PORT	S	WEB_PROXY_SERVER	1024:65535	ACK

Las siguientes reglas permiten las conexiones cliente de proxy web entrantes desde la máquina de contención, así como desde máquinas situadas en la LAN privada detrás del firewall de contención:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $WEB_PROXY_SERVER $WEB_PROXY_PORT -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_PROXY_SERVER $WEB_PROXY_PORT \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Conexiones salientes del cliente de contención a servidores externos

La Tabla 4.36 lista el protocolo de conexión de cliente para el servicio web HTTP.

Tabla 4.36. Protocolo de cliente HTTP de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	80	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor	TCP	ANYWHERE	80	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Esta sección permite a los host locales conectarse a servidores web tanto en el bastión como en host remotos:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 80 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 80 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.37 lista el protocolo de conexión de cliente para el servicio SSL.

Tabla 4.37. Protocolo de cliente SSL de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	443	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor	TCP	ANYWHERE	443	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Esta sección permite a los host locales conectarse a servidores web seguros tanto en el bastión como en host remotos:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 443 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 443 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.38 lista el protocolo de conexión de cliente para el servicio proxy Web.

Tabla 4.38. Protocolo del cliente proxy de Web

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	WEB_PROXY _SERVER	WEB_PROXY _PORT	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	WEB_PROXY _SERVER	WEB_PROXY _PORT	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Esta sección permite a los *host* locales conectarse a servidores proxy web seguros tanto en el bastión como en *host* remotos:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $WEB_PROXY_SERVER $WEB_PROXY_PORT -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_PROXY_SERVER $WEB_PROXY_PORT \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

El bastión como conducto, el servidor web público de la DMZ y la máquina de contención como clientes

El bastión es un conducto bidireccional: un servidor web se ejecuta en la DMZ y la máquina de contención es un cliente.

Conducto de bastión a servidores web remotos de la DMZ

La Tabla 4.39 lista el protocolo completo de conexión cliente/servidor para el servicio web HTTP.

Tabla 4.39. Protocolo del cliente/servidor HTTP del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente DMZ	TCP	ANYWHERE	80	E	DMZ_ADDRESSES	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	80	S	DMZ_ADDRESSES	1024:65535	ACK
Petición del cliente remoto	TCP	ANYWHERE	1024:65535	S	WEB_SERVER_DMZ_IPADDR	80	Cualquiera
Respuesta del servidor DMZ	TCP	ANYWHERE	1024:65535	E	WEB_SERVER_DMZ_IPADDR	80	ACK

Esta sección permite a los host locales conectarse a servidores web tanto en el bastión como en host remotos, y permite a los clientes remotos conectarse a un servidor web de la DMZ:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $ANYWHERE 80 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 80 \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $WEB_SERVER_DMZ_IPADDR 80 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_SERVER_DMZ_IPADDR 80 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.40 lista el protocolo completo de conexión cliente/servidor para el servicio SSL.

Tabla 4.40. Protocolo del cliente/servidor SSL del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente DMZ	TCP	ANYWHERE	443	E	DMZ_ADDRESSES	1024:65535	Cualquiera

Tabla 4.40. Protocolo del cliente/servidor SSL del bastión *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Respuesta del servidor remoto	TCP	ANYWHERE	443	S	DMZ_ADDRESSES	1024:65535	ACK
Petición del cliente remoto	TCP	ANYWHERE	1024:65535	S	WEB_SERVER_DMZ_IPADDR	443	Cualquiera
Respuesta del servidor DMZ	TCP	ANYWHERE	1024:65535	E	WEB_SERVER_DMZ_IPADDR	443	ACK

Esta sección permite a los host locales conectarse a servidores web seguros tanto en el bastión como en host remotos, y permite a los clientes remotos conectase a un servidor web seguro de la DMZ:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $DMZ_ADDRESSES $UNPRIVPORTS \
         -d $ANYWHERE 443 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 443 \
         -d $DMZ_ADDRESSES $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $WEB_SERVER_DMZ_IPADDR 443 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_SERVER_DMZ_IPADDR 443 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.41 lista el protocolo de conexión completo cliente/servidor para el servicio proxy Web.

Tabla 4.41. Protocolo del cliente/servidor proxy web del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	WEB_PROXY_SERVER	WEB_PROXY_PORT	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	WEB_PROXY_SERVER	WEB_PROXY_PORT	S	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente remoto	TCP	ANYWHERE	1024:65535	S	WEB_SERVER_DMZ_IPADDR	WEB_PROXY_PORT	Cualquiera
Respuesta del servidor DMZ	TCP	ANYWHERE	1024:65535	E	WEB_SERVER_DMZ_IPADDR	1024:65535	ACK

Esta sección permite a los host locales conectarse a servidores proxy web tanto en el bastión como en los host remotos, y permite a los clientes remotos conectarse a un servidor proxy web de la DMZ:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $WEB_PROXY_SERVER $WEB_PROXY_PORT -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_PROXY_SERVER $WEB_PROXY_PORT \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $WEB_SERVER_DMZ_IPADDR $WEB_PROXY_PORT -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_SERVER_DMZ_IPADDR $WEB_PROXY_PORT \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Servidor web público de la DMZ

Dos de los paquetes proxy web que se incluyen con la versión 6.0 de Red Hat son el módulo de proxy, que se incluye con el servidor Apache estándar, y un paquete independiente llamado Squid. Los dos paquetes permiten definir el puerto de servicio (el valor predeterminado de Squid es el puerto 3130).

Tanto el módulo proxy de Apache como el servidor proxy Squid se incluyen en las versiones recientes de Red Hat. Anteriormente, era necesario volver a compilar el código fuente de Apache con el módulo proxy habilitado. Squid era gratuito, pero todavía no se incluía en la versión estándar de Red Hat.

La Tabla 4.42 lista el protocolo de conexión del servidor para el servicio web HTTP.

Tabla 4.42. Protocolo del servidor HTTP de la DMZ

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	1024:65535	E	WEB_SERVER_DMZ_ IPADDR	80	Cualquiera
Respuesta del servidor DMZ	TCP	ANYWHERE	1024:65535	S	WEB_SERVER_DMZ_ IPADDR	80	ACK

Las siguientes reglas permiten las conexiones entrantes del cliente HTTP desde cualquier sitio, incluyendo host remotos, así como las máquinas bastión y contención:

```
WEB_DMZ_INTERFACE="eth0"
```

```
ipchains -A input  -i $WEB_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $WEB_SERVER_DMZ_IPADDR 80 -j ACCEPT
ipchains -A output -i $WEB_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_SERVER_DMZ_IPADDR 80 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.43 lista el protocolo de conexión de servidor para el servicio SSL.

Tabla 4.43. Protocolo del servidor SSL de la DMZ

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente remoto	TCP	ANYWHERE	1024:65535	E	WEB_SERVER_DMZ_IPADDR	443	Cualquiera
Respuesta del servidor DMZ	TCP	ANYWHERE	1024:65535	S	WEB_SERVER_DMZ_IPADDR	443	ACK

Las siguientes reglas permiten las conexiones entrantes del cliente web SSL desde cualquier sitio, incluyendo *host* remotos, así como las máquinas bastión y contención:

```
WEB_DMZ_INTERFACE="eth0"
ipchains -A input  -i $WEB_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $WEB_SERVER_DMZ_IPADDR 443 -j ACCEPT
ipchains -A output -i $WEB_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_SERVER_DMZ_IPADDR 443 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.44 lista el protocolo de conexión de servidor para el servicio proxy Web.

Tabla 4.44. Protocolo del servidor proxy web de la DMZ

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente remoto	TCP	Clientes proxy	1024:65535	E	WEB_SERVER_DMZ_IPADDR	WEB_PROXY_PORT	Cualquiera
Respuesta del servidor DMZ	TCP	Clientes proxy	1024:65535	S	WEB_SERVER_DMZ_IPADDR	WEB_PROXY_PORT	ACK

Es improbable que un sistema pequeño tenga razones suficientes como para ofrecer servicios proxy web sobre Internet. Resulta más normal que se ejecute un proxy local para proporcionar una caché local y para posibles conexiones proxy salientes. En este ejemplo no es necesaria una regla de fire-

wall para un proxy local. Para el mundo exterior, el servidor proxy aparece como un navegador Web normal. En casos excepcionales, un ejemplo de regla de servidor firewall sería:

```
WEB_DMZ_INTERFACE="etho"

ipchains -A input  -i $WEB_DMZ_INTERFACE -p tcp \
         -s <mi.Web.clientes_proxy> $UNPRIVPORTS \
         -d $WEB_SERVER_DMZ_IPADDR $WEB_PROXY_PORT -j ACCEPT

ipchains -A output -i $WEB_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_SERVER_DMZ_IPADDR $WEB_PROXY_PORT \
         -d <mi.Web.clientes_proxy> $UNPRIVPORTS -j ACCEPT
```

Máquina de contención como cliente a servidores remotos y servidores web de la DMZ

La Tabla 4.45 lista el protocolo de conexión del cliente para el servicio web HTTP.

Tabla 4.45. Protocolo del cliente HTTP de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	80	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	80	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones salientes de cliente HTTP desde la máquina de contención a un servidor web en cualquier sitio:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 80 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 80 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.46 lista el protocolo de conexión de cliente para el servicio SSL.

Tabla 4.46. Protocolo del cliente SSL de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	443	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	443	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones salientes de cliente SSL desde la máquina de contención a un servidor web seguro ubicado en cualquier parte:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 443 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 443 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

La Tabla 4.47 lista el protocolo de conexión de cliente para el servicio proxy Web.

Tabla 4.47. Protocolo de cliente proxy web de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	WEB_PROXY_SERVER	WEB_PROXY_PORT	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	WEB_PROXY_SERVER	WEB_PROXY_PORT	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones salientes del cliente desde la máquina de contención a servidores proxy web ubicados en cualquier parte:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $WEB_PROXY_SERVER $WEB_PROXY_PORT -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $WEB_PROXY_SERVER $WEB_PROXY_PORT \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

La máquina bastión como servidor web o conducto, la máquina de contención como servidor proxy web local

Aunque es posible ofrecer servicio web público desde un servidor interno centralizado, no se suele realizar dada la gran cantidad de posibles violaciones de seguridad que podría sufrir si se tienen servidores y secuencias de comandos CGI mal configurados, y a la tendencia de aislar información privada de la información pública. Es decir, los sitios tienen tanto un sitio web interno privado como un sitio web público que ejecutan múltiples servidores web en diferentes máquinas y en distintas LAN. Lo más normal será albergar el sitio web público tanto desde el firewall bastión como desde un host en la red de perímetro.

En el caso de un sitio personal o de una empresa pequeña, una posibilidad es ejecutar un servidor público en la máquina bastión y un segundo ser-

vidor proxy web privado en una máquina interna, el firewall de contención en este caso. En esta situación, el servidor público puede ofrecer o no servicio SSL. El servidor web interno ofrece servicio proxy. El servidor privado no es accesible desde la máquina bastión.

En este caso, no es necesario el uso de reglas adicionales. Desde la perspectiva del bastión, el proxy web interno no aparenta ser un cliente web. La única diferencia en esta situación es que no serán necesarias las reglas de acceso del proxy web que se han mostrado anteriormente, a no ser que el ISP necesite que el usuario use su servicio proxy web.

No es necesario realizar cambios en las reglas de interfaz de la DMZ, ni en la máquina bastión ni en la máquina de contención.

finger (Puerto TCP 79)

No existe ningún riesgo si se habilita el acceso saliente a los servidores remotos finger. Es probable que no sea muy útil ofrecer el servicio de finger interno a la máquina bastión. No sólo resulta desalentador ofrecer el servicio finger interno a Internet en un sistema enmascarado, sino que no es posible sin un gran esfuerzo adicional.

Configuración del bastión finger del conducto de la DMZ

La Tabla 4.48 lista el protocolo completo de conexión cliente/servidor para el servicio finger.

Tabla 4.48. Protocolo del conducto finger del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	79	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	79	S	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente remoto	TCP	Clientes finger	1024:65535	S	CHOKE_DMZ_IPADDR	79	Cualquiera
Respuesta del servidor de contención	TCP	Clientes finger	1024:65535	E	CHOKE_DMZ_IPADDR	79	ACK

No existe ningún peligro en habilitar el acceso saliente a servidores remotos finger, y éstas son las reglas para realizarlo:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 79 -j ACCEPT
```

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 79 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Las reglas de servidor no tienen mucho sentido en un sitio particular. Un pequeño ISP que alberga cuentas de usuario soportará los servidores de finger internos:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 79 -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR 79 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Configuración de cliente y servidor finger de contención de la DMZ

La Tabla 4.49 lista el protocolo completo de conexión cliente/servidor para el servicio finger.

Tabla 4.49. Protocolo finger de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de la máquina de contención	TCP	ANYWHERE	79	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	79	E	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente remoto	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	79	Cualquiera
Respuesta del servidor de contención	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	79	ACK

Las dos reglas siguientes habilitan las peticiones salientes del cliente finger desde la máquina de contención:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS  \
         -d $ANYWHERE 79 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 79 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Las dos reglas siguientes habilitan las peticiones entrantes del cliente finger a la máquina de contención:

```
ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 79 -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR 79  \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

whois (Puerto TCP 43)

El programa whois accede a la base de datos de los servicios de registro de ARIN. Este programa permite que se puedan realizar de forma legible búsquedas por dirección IP y por nombre de host y de dominio.

Configuración del conducto del cliente WHOIS del bastión de la DMZ

La Tabla 4.50 lista el protocolo de conexión de cliente para el servicio WHOIS.

Tabla 4.50. Protocolo del cliente WHOIS del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	43	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	43	S	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones del cliente WHOIS desde la máquina de contención, así como desde máquinas situadas en la LAN privada detrás del firewall de contención:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_firewall $UNPRIVPORTS \
         -d $ANYWHERE 43 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 43 \
         -d $CHOKE_firewall $UNPRIVPORTS -j ACCEPT
```

Configuración del cliente WHOIS de la DMZ

La Tabla 4.51 lista el protocolo de conexión del cliente para el servicio WHOIS.

Tabla 4.51. Protocolo del cliente WHOIS de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	43	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	43	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones salientes WHOIS desde la máquina de contención, así como desde máquinas situadas en la LAN privada detrás del firewall de contención:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 43 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 43 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

gopher (TCP Puerto 70)

El servicio de información Gopher todavía sigue disponible, pero su uso se ha reemplazado bastante por motores de búsqueda web. Sin embargo, Netscape incluye compatibilidad de proxy cliente de gopher.

Configuración del conducto del cliente gopher del bastión de la DMZ

La tabla 4.52 lista el protocolo de conexión del cliente para el servicio Gopher.

Tabla 4.52. Protocolo del cliente gopher del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	70	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	70	S	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las siguientes reglas permiten las conexiones entrantes de cliente gopher desde la máquina de contención, así como desde máquinas de la LAN privada situadas detrás del firewall de contención:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 70 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 70 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración del cliente gopher de contención de la DMZ

La tabla 4.53 lista el protocolo de conexión de cliente para el servicio Gopher.

Tabla 4.53. Protocolo del cliente gopher de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	70	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	70	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las dos reglas siguientes habilitan las peticiones salientes del cliente gopher desde la máquina de contención:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 70 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 70 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

WAIS (TCP Puerto 210)

Los servidores de información de área extensa (WAIS, Wide Area Information Servers) se conocen ahora como motores de búsqueda. Los servidores web suelen proporcionar una presentación gráfica a los servidores WAIS. Netscape incluye el código de cliente WAIS necesario para conectarse a servidores WAIS.

Configuración del conducto del cliente WAIS del bastión de la DMZ

La Tabla 4.54 lista el protocolo de conexión del cliente para el servicio WAIS.

Tabla 4.54. Protocolo del cliente WAIS del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	210	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	210	S	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las dos reglas siguientes permiten a los clientes internos acceder a servicios WAIS remotos:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 210 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 210 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración del cliente WAIS de contención de la DMZ

La Tabla 4.55 lista el protocolo de conexión del cliente para el servicio WAIS.

Tabla 4.55. Protocolo del cliente WAIS de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	210	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	210	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Las dos reglas siguientes permiten el acceso del cliente a servidores WAIS remotos:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 210 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 210 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

RealAudio y QuickTime (554)

Tanto RealAudio como QuickTime usan el protocolo de transmisión en tiempo real (RTSP, Real Time Streaming Protocol) sobre el puerto TCP 554, y el protocolo de transporte en tiempo real (RTP, Real Time Transport Protocol) sobre un par de puertos no privilegiados, TCP y UDP. RTSP proporciona el flujo de control. RTP proporciona el flujo de datos.

Los puertos particulares que se usan para el par de puertos no privilegiados los define el software de cliente particular que se usa. Por ejemplo, QuickTime de Apple usa los puertos 7070 y 7071 para el flujo de datos TCP. QuickTime usa el primer par de puertos disponibles en el intervalo desde 6970 a 6999 para el flujo de datos UDP.

Se puede acceder a los dos servicios sobre HTTP. En este caso, son necesarias reglas de firewall. También se pueden configurar los dos servicios para que usen flujo de datos TCP o UDP. HTTP y TCP ofrecen conexiones más seguras. UDP proporciona el mejor rendimiento.

El módulo proxy de RealAudio incluido en la versión de Red Hat se debe usar tanto para aceptar los datagramas UDP entrantes como para ayudar a proteger el tráfico de red basado en UDP de RealAudio y QuickTime. Además, también puede ser necesaria la utilidad ipmasqadm portfw para enviar los paquetes UDP entrantes a una máquina LAN enmascarada.

Configuración externa del cliente RealAudio del bastión

La Tabla 4.56 lista el protocolo de conexión del cliente en la interfaz de Internet externa del servicio RealAudio.

Tabla 4.56. Protocolo externo del cliente RealAudio del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	1024:65535	S	BASTION_EXTERNAL_IPADDR	554	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	1024:65535	E	BASTION_EXTERNAL_IPADDR	554	ACK
Petición del cliente de contención	TCP	ANYWHERE	1024:65535	S	BASTION_EXTERNAL_IPADDR	7070:7071	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	1024:65535	E	BASTION_EXTERNAL_IPADDR	7070:7071	ACK
Respuesta del servidor remoto	TCP	ANYWHERE	1024:65535	E	BASTION_EXTERNAL_IPADDR	6970:7170	

Las siguientes reglas permiten la comunicación de control del cliente entre la máquina bastión y los servidores remotos de RealAudio:

```
ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p tcp \
         -s $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 554 -j ACCEPT
ipchains -A input  -i $BASTION_EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 554 \
         -d $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS -j ACCEPT
```

Las siguientes reglas permiten la comunicación de datos del cliente entre la máquina bastión y los servidores de RealAudio remotos usando una conexión TCP:

```
ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p tcp \
         -s $BASTION_EXTERNAL_IPADDR 7070:7071 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
ipchains -A input  -i $BASTION_EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $BASTION_EXTERNAL_IPADDR 7070:7071 -j ACCEPT
```

Las siguientes reglas permiten la comunicación de datos del cliente entre la maquina bastión y los servidores de RealAudio remotos usando datagramas UDP:

```
ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s $BASTION_EXTERNAL_IPADDR 6970:6999 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
ipchains -A input -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $BASTION_EXTERNAL_IPADDR 6970:6999 -j ACCEPT
```

Configuración del conducto del cliente de RealAudio del bastión de la DMZ

La Tabla 4.57 lista el protocolo de conexión del conducto del cliente para el servicio RealAudio.

Tabla 4.57. Protocolo de conducto del cliente RealAudio del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	554	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	554	ACK
Petición del cliente de contención	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	7070:7071	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	7070:7071	ACK
Respuesta del servidor remoto	UDP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	6970:7170	-

RealAudio / QuickTime en un entorno enmascarado

Si las máquinas LAN están enmascaradas, son necesarias las siguientes líneas:

```
/sbin/modprobe ip_masq_raudio.o ports=554,7070, 7071,6970,6971
/usr/sbin/ipmasqadm portfw -f
/usr/sbin/ipmasqadm portfw -a -P udp -L $BASTION_EXTERNAL_IPADDR 6970 -R $CHOKE_DMZ_IPADDR 6970
/usr/sbin/ipmasqadm portfw -a -P udp -L $BASTION_EXTERNAL_IPADDR 6971 -R $CHOKE_DMZ_IPADDR 6971
```

Las siguientes reglas permiten a un cliente de la máquina de contención intercambiar información de control con un servidor de RealAudio remoto:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 554 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 554 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Las siguientes reglas permiten a un cliente de la máquina de contención intercambiar comunicación de datos con un servidor de RealAudio remoto usando conexiones TCP:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR 7070:7071 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 7070:7071 -j ACCEPT
```

Las siguientes reglas permiten a un cliente de la máquina de contención intercambiar comunicación de datos con un servidor de RealAudio remoto usando datagramas UDP:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR 6970:6999 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 6970:6999 -j ACCEPT
```

Configuración del cliente de RealAudio de contención de la DMZ

La Tabla 4.58 lista el protocolo de conexión del cliente para el servicio RealAudio.

Tabla 4.58. Protocolo del cliente de RealAudio de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	554	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	554	ACK
Petición del cliente de contención	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	7070:7071	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	7070:7071	ACK
Respuesta del servidor remoto	UDP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	6970:7170	-

Las siguientes reglas permiten a un cliente local intercambiar información de control con un servidor remoto:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 554 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 554 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Las siguientes reglas permiten a un cliente local intercambiar comunicación de datos con un servidor de RealAudio remoto usando conexiones TCP:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR 7070:7071 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 7070:7071 -j ACCEPT
```

Las siguientes reglas permiten a un cliente local intercambiar comunicación de datos con un servidor de RealAudio remoto usando datagramas UDP:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR 6970:6999 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR 6970:6999 -j ACCEPT
```

IRC (Puerto TCP 6667)

El puerto 6667 es el puerto predeterminado del servidor IRC. Los servidores tienen la opción de usar un puerto diferente. Si se accede a un servicio IRC sobre un puerto diferente, es necesario modificar las reglas de acuerdo con este cambio.

Se debe usar el módulo proxy de IRC que se incluye con la versión de Linux debido al protocolo de comunicación cliente a cliente, de puerto no privilegiado a puerto no privilegiado y para permitir conexiones entrantes desde los clientes remotos. Los firewalls de los sitios empresariales y comerciales no deben permitir el IRC a través del firewall debido al riesgo de seguridad inherente del protocolo.

Configuración externa del cliente IRC del bastión

La Tabla 4.59 lista el protocolo de conexión del cliente del bastión en la interfaz externa de Internet para el servicio IRC. Casi siempre se accede a un servicio de Internet Relay Chat externo. Por tanto, no se incluye aquí ninguna regla de servidor local.

Tabla 4.59. Protocolo externo del cliente IRC del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	6667	S	BASTION_EXTERNAL_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	6667	E	BASTION_EXTERNAL_IPADDR	1024:65535	ACK
Petición del cliente de contención al cliente remoto	TCP	ANYWHERE	1024:65535	S	BASTION_EXTERNAL_IPADDR	1024:65535	Cualquiera
Respuesta del cliente remoto al cliente de contención	TCP	ANYWHERE	1024:65535	E	BASTION_EXTERNAL_IPADDR	1024:65535	ACK
Petición del cliente remoto al cliente de contención	TCP	ANYWHERE	1024:65535	E	BASTION_EXTERNAL_IPADDR	1024:65535	ACK
Respuesta del cliente de contención al cliente remoto	TCP	ANYWHERE	1024:65535	S	BASTION_EXTERNAL_IPADDR	1024:65535	Cualquiera

El siguiente conjunto de reglas permite la comunicación cliente entre clientes IRC locales y servidores remotos, así como con clientes remotos:

```
ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p tcp \
         -s $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 6667 -j ACCEPT

ipchains -A input  -i $BASTION_EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 6667 \
         -d $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p tcp \
         -s $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $BASTION_EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $BASTION_EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p tcp ! -y \
         -s $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

Configuración del conducto del cliente IRC del bastión de la DMZ

La Tabla 4.60 lista el protocolo de conexión del conducto del cliente para el servicio IRC. Casi siempre se accede a un servicio de Internet Relay Chat externo. Por tanto, no se incluye aquí ninguna regla de servidor local.

Tabla 4.60. Protocolo del conducto del cliente IRC del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	6667	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	6667	S	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente de contención al cliente remoto	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del cliente remoto al cliente de contención	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente remoto al cliente de contención	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	1024:65535	ACK
Respuesta del cliente de contención al cliente remoto	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera

El siguiente conjunto de reglas permite la comunicación cliente entre clientes IRC de la máquina de contención local y servidores remotos, de la misma forma que con clientes remotos:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 6667 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 6667 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración del cliente IRC del bastión de la DMZ

La Tabla 4.61 lista el protocolo de conexión del cliente para el servicio IRC. Casi siempre se accede a un servicio de Internet Relay Chat externo. Por tanto, no se incluye aquí ninguna regla de servidor local.

Tabla 4.61. Protocolo del cliente IRC de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	TCP	ANYWHERE	6667	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	ANYWHERE	6667	E	CHOKE_DMZ_IPADDR	1024:65535	ACK
Petición del cliente de contención al cliente remoto	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Repuesta del cliente remoto al cliente de contención	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

Tabla 4.61. Protocolo del cliente IRC de la máquina de contención *(continuación)*

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente remoto al cliente de contención	TCP	ANYWHERE	1024:65535	E	CHOKE_DMZ_IPADDR	1024:65535	ACK
Respuesta del cliente remoto al cliente de contención	TCP	ANYWHERE	1024:65535	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera

El siguiente conjunto de reglas permite la comunicación del cliente entre clientes IRC locales y servidores remotos, así como con clientes remotos:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 6667 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 6667 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

CU-SeeMe (Puertos UDP 7648, 7649 y 24032; Puertos TCP 7648 y 7649)

El uso de CU-SeeMe requiere tener acceso a un servidor remoto. Por lo tanto, no se incluye una regla de servidor en el ejemplo.

Debido a los problemas de seguridad inherentes al protocolo UDP, es necesario un servidor CU-SeeMe de forma explícita. Además, se debe usar el módulo proxy de CU-SeeMe que se incluye en la versión Red Hat de Linux para los clientes enmascarados.

Configuración externa del cliente CU-SeeMe del bastión

La Tabla 4.62 lista el protocolo de conexión del cliente en la interfaz de Internet externa, para el servicio CU-SeeMe.

Tabla 4.62. Protocolo externo del cliente CU-SeeMe del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	UDP	Servidor CU-SeeMe	7648:7649	S	BASTIÓN_EXTERNAL_IPADDR	1024:65535	-
Respuesta del servidor remoto	UDP	Servidor CU-SeeMe	7648:7649	E	BASTIÓN_EXTERNAL_IPADDR	1024:65535	-
Petición del cliente de contención	UDP	Servidor CU-SeeMe	24032	S	BASTIÓN_EXTERNAL_IPADDR	1024:65535	-
Respuesta del servidor remoto	UDP	Servidor CU-SeeMe	24032	E	BASTIÓN_EXTERNAL_IPADDR	1024:65535	-
Petición del cliente de contención	TCP	Servidor CU-SeeMe	7648:7649	S	BASTIÓN_EXTERNAL_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	Servidor CU-SeeMe	7648:7649	E	BASTIÓN_EXTERNAL_IPADDR	1024:65535	ACK

CU-SeeMe

Para obtener más información sobre cuestiones de firewall relacionadas con CU-SeeMe, consulte las direcciones http://www.cu-seeme.com y http://www.wpine.com.

El siguiente conjunto de reglas permite la comunicación cliente entre la máquina bastión y los servidores remotos:

```
ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS \
         -d <servidor.cu-seeme> 7648:7649 -j ACCEPT

ipchains -A input -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s <servidor.cu-seeme> 7648:7649 \
         -d $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS \
         -d <servidor.cu-seeme> 24032 -j ACCEPT

ipchains -A input  -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s <servidor.cu-seeme> 24032 \
         -d $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p tcp \
         -s $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS \
         -d <servidor.cu-seeme> 7648:7649 -j ACCEPT
```

```
ipchains -A input -i $BASTION_EXTERNAL_INTERFACE -p tcp ! -y \
         -s <servidor.cu-seeme> 7648:7649 \
         -d $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración del cliente CU-SeeMe del bastión de la DMZ

La Tabla 4.63 lista el protocolo de conexión de cliente en la interfaz DMZ del bastión para el servicio CU-SeeMe.

Tabla 4.63. Protocolo del conducto del cliente CU-SeeMe del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	UDP	Servidor CU-SeeMe	7648:7649	E	CHOKE_DMZ_IPADDR	1024:65535	-
Respuesta del servidor remoto	UDP	Servidor CU-SeeMe	7648:7649	S	CHOKE_DMZ_IPADDR	1024:65535	-
Petición del cliente de contención	UDP	Servidor CU-SeeMe	24032	E	CHOKE_DMZ_IPADDR	1024:65535	-
Respuesta del servidor remoto	UDP	Servidor CU-SeeMe	24032	S	CHOKE_DMZ_IPADDR	1024:65535	-
Petición del cliente de contención	TCP	Servidor CU-SeeMe	7648:7649	E	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	Servidor CU-SeeMe	7648:7649	S	CHOKE_DMZ_IPADDR	1024:65535	ACK

El siguiente conjunto de reglas permite la comunicación cliente entre clientes CU-SeeMe locales y servidores remotos:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.cu-seeme> 7648:7649 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s <servidor.cu-seeme> 7648:7649 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.cu-seeme> 24032 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s <servidor.cu-seeme> 24032 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $BASTION_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.cu-seeme> 7648:7649 -j ACCEPT
```

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p tcp ! -y \
         -s <servidor.cu-seeme> 7648:7649 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración del cliente CU-SeeMe de contención de la DMZ

La Tabla 4.64 lista el protocolo de conexión del cliente para el servicio CU-SeeMe.

Tabla 4.64. Protocolo del cliente CU-SeeMe de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición del cliente de contención	UDP	Servidor CU-SeeMe	7648:7649	S	CHOKE_DMZ_IPADDR	1024:65535	-
Respuesta del servidor remoto	UDP	Servidor CU-SeeMe	7648:7649	E	CHOKE_DMZ_IPADDR	1024:65535	-
Petición del cliente de contención	UDP	Servidor CU-SeeMe	24032	S	CHOKE_DMZ_IPADDR	1024:65535	-
Respuesta del servidor remoto	UDP	Servidor CU-SeeMe	24032	E	CHOKE_DMZ_IPADDR	1024:65535	-
Petición del cliente de contención	TCP	Servidor CU-SeeMe	7648:7649	S	CHOKE_DMZ_IPADDR	1024:65535	Cualquiera
Respuesta del servidor remoto	TCP	Servidor CU-SeeMe	7648:7649	E	CHOKE_DMZ_IPADDR	1024:65535	ACK

El siguiente conjunto de reglas permite la comunicación cliente entre los clientes CU-SeeMe locales y los servidores remotos:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.cu-seeme> 7648:7649 -j ACCEPT

ipchains -A input -i $CHOKE_DMZ_INTERFACE -p udp \
         -s <servidor.cu-seeme> 7648:7649 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.cu-seeme> 24032 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p udp \
         -s <servidor.cu-seeme> 24032 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.cu-seeme> 7648:7649 -j ACCEPT
```

```
ipchains -A input -i $CHOKE_DMZ_INTERFACE -p tcp ! -y \
         -s <servidor.cu-seeme> 7648:7649 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Quake (Puertos UDP 26000 y 1025 hasta el 1200)

Los servidores Quake escuchan, de forma predeterminada, esperando peticiones de conexión de juego en el puerto UDP 26000. Cuando el servidor acepta una petición de un nuevo jugador, devuelve el número de puerto UDP que usará para comunicarse con el programa cliente del jugador. El puerto del servidor para el intercambio cliente/servidor actual suele ser un puerto UDP entre 1025 y 1200.

Debido a los problemas de seguridad inherentes al protocolo UDP, debe especificarse un servidor Quake explícito. Además se debe usar el módulo proxy Quake que se incluye en la versión Red Hat de Linux para enviar los paquetes UDP entrantes a los clientes enmascarados.

Configuración externa del cliente Quake bastión

La Tabla 4.65 lista el protocolo de conexión del cliente en la interfaz de Internet externa para el servicio Quake.

Tabla 4.65. Protocolo externo del cliente Quake del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición de conexión del cliente de contención	UDP	Servidor de Quake	26000	S	BASTIÓN_EXTERNAL_IPADDR	1024:65535	-
Respuesta de conexión del servidor remoto	UDP	Servidor de Quake	26000	E	BASTIÓN_EXTERNAL_IPADDR	1024:65535	-
Petición del cliente de contención	UDP	Servidor de Quake	1025:1200	S	BASTIÓN_EXTERNAL_IPADDR	1024:65535	-
Respuesta del servidor remoto	UDP	Servidor de Quake	1025:1200	E	BASTIÓN_EXTERNAL_IPADDR	1024:65535	-

Quake

Si se desea más información sobre las cuestiones de firewall relacionadas con Quake, consulte la dirección:

http://www.gamers.org/dEngine/quake/spec.

El siguiente conjunto de reglas permite la comunicación cliente entre la máquina bastión y los servidores Quake remotos:

```
ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS \
         -d <servidor.quake> 26000 -j ACCEPT

ipchains -A input -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s <servidor.quake> 26000 \
         -d $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS \
         -d <servidor.quake> 1025:1200 -j ACCEPT

ipchains -A input  -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s <servidor.quake> 1025:1200 \
         -d $BASTION_EXTERNAL_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración del cliente Quake del bastión de la DMZ

La Tabla 4.66 lista el protocolo de conexión de conducto del cliente del bastión para el servicio Quake.

Tabla 4.66. Protocolo de conducto del cliente Quake del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición de conexión del cliente de contención	UDP	Servidor de Quake	26000	E	CHOKE_DMZ_IPADDR	1024:65535	-
Respuesta de conexión del servidor remoto	UDP	Servidor de Quake	26000	S	CHOKE_DMZ_IPADDR	1024:65535	-
Petición del cliente de contención	UDP	Servidor de Quake	1025:1200	E	CHOKE_DMZ_IPADDR	1024:65535	-
Respuesta del servidor remoto	UDP	Servidor de Quake	1025:1200	S	CHOKE_DMZ_IPADDR	1024:65535	-

El siguiente conjunto de reglas permite la comunicación cliente entre clientes Quake locales y los servidores remotos:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.quake> 26000 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s <servidor.quake> 26000 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.quake> 1025:1200 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s <servidor.quake> 1025:1200 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Configuración del servidor Quake del bastión

No se puede enmascarar y ejecutar un servidor Quake sobre una máquina de la DMZ. Un servidor Quake debe ejecutarse en la máquina bastión o en un servidor público interno con una dirección IP registrada.

La Tabla 4.67 lista el protocolo de conexión del servidor del bastión en la interfaz de Internet externa para clientes remotos de Quake.

Tabla 4.67. Protocolo del servidor Quake del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición de conexión del cliente de contención	UDP	ANYWHERE	1024:65535	E	BASTION_EXTERNAL_IPADDR	26000	-
Respuesta de conexión del servidor remoto	UDP	ANYWHERE	1024:65535	S	BASTION_EXTERNAL_IPADDR	26000	-
Petición del cliente de contención	UDP	ANYWHERE	1024:65535	E	BASTION_EXTERNAL_IPADDR	1025:1200	-
Respuesta del servidor bastión	UDP	ANYWHERE	1024:65535	S	BASTION_EXTERNAL_IPADDR	1025:1200	-

El siguiente conjunto de reglas permite la comunicación cliente entre clientes Quake locales y servidores remotos:

```
ipchains -A input  -i $BASTION_EXTERNAL_INTERFACE -p udp \
          -s $ANYWHERE $UNPRIVPORTS \
         -d $BASTION_EXTERNAL_IPADDR 26000 -j ACCEPT

ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s $BASTION_EXTERNAL_IPADDR 26000 \
         -d $ANYWHERE  $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s $ANYWHERE  $UNPRIVPORTS \
         -d $BASTION_EXTERNAL_IPADDR 1025:1200 -j ACCEPT

ipchains -A output -i $BASTION_EXTERNAL_INTERFACE -p udp \
         -s $BASTION_EXTERNAL_IPADDR 1025:1200 \
         -d $ANYWHERE  $UNPRIVPORTS -j ACCEPT
```

Configuración del cliente Quake de contención de la DMZ

La Tabla 4.68 lista el protocolo de conexión del cliente para el servicio Quake.

Tabla 4.68. Protocolo del cliente Quake de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local	Indicador TCP
Petición de conexión del cliente de contención	UDP	Servidor de Quake	26000	S	CHOKE_DMZ_IPADDR	1024:65535	-
Respuesta de conexión del servidor remoto	UDP	Servidor de Quake	26000	E	CHOKE_DMZ_IPADDR	1024:65535	-
Petición del cliente de contención	UDP	Servidor de Quake	1025:1200	S	CHOKE_DMZ_IPADDR	1024:65535	-
Respuesta del servidor remoto	UDP	Servidor de Quake	1025:1200	E	CHOKE_DMZ_IPADDR	1024:65535	-

El siguiente conjunto de reglas permite la comunicación de cliente entre clientes Quake locales y servidores remotos:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.quake> 26000 -j ACCEPT

ipchains -A input -i $CHOKE_DMZ_INTERFACE -p udp \
         -s <servidor.quake> 26000 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d <servidor.quake> 1025:1200 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p udp \
         -s <servidor.quake> 1025:1200 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Servicio horario de red (Puerto UDP 123)

El servicio horario de red (NTP, Network Time Service) permite el acceso a proveedores de tiempo. Esto es útil para mantener ajustado el reloj del sistema, particularmente si el reloj interno de una máquina tiende a descontrolarse, así como para establecer la fecha y hora correctas cuando se inicia o después de un corte de corriente. xntpd también es útil como servicio interno para sincronizar los relojes de todas las máquinas locales.

Cómo habilitar servicios LAN privados

En el Capítulo 1, "Conceptos básicos subyacentes a los firewalls de filtrado de paquetes", se introduce la idea de que muchos servicios UNIX habituales están diseñados para su uso en una LAN interna. Estos servicios pueden presentar riesgos para la seguridad y molestar a los demás usuarios, si está permitido el acceso externo a ellos o si se filtran a Internet. En el Capítulo 3 se explicaron estos servicios desde una perspectiva de filtrado de paquetes y se incluían reglas de ejemplo redundantes para mostrar el tipo de cosas que el administrador no quiere que pasen a través de la interfaz externa hacia Internet. Algunos de estos servicios representan riesgos potencialmente tan peligrosos que no se suelen ejecutar en la máquina bastión, y algunos sitios eliminan por completo el software de la máquina bastión.
Aunque estos servicios se pueden usar en la máquina bastión detrás de la protección del firewall, la idea que subyace al bastión es que la máquina externa esté lo mejor asegurada posible. Un sistema independiente o una LAN particular pequeña pueden tener que realizar elecciones y compromisos. Lo más importante para el administrador es ser consciente de los riesgos y las posibles protecciones antes de hacer estas elecciones. Estos mismos servicios pueden ser muy útiles de forma interna. La LAN puede depender de ellos. Son uno de los ingredientes que hacen de UNIX un sistema operativo tan flexible y potente. La posibilidad de poder agregar un firewall de contención interno agrega una capa extra de seguridad y confidencialidad cuando se ofrecen estos servicios a una LAN, ya sea desde servidores LAN internos como desde el propio host firewall de la máquina de contención.
Como ejemplo, las siguientes reglas de ipchains se escriben desde la perspectiva del servicio que se ofrece desde la máquina de contención a los clientes internos.

Configuración del bastión como servidor NTP local de la DMZ

La Tabla 4.69 lista el protocolo de intercambio cliente/servidor de igual a igual para el servicio NTP.

Tabla 4.69. Protocolos de servidor NTP del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local
Petición del cliente de contención	UDP	BASTIÓN_DMZ_IPADDR	123	E	CHOKE_DMZ_IPADDR	1024:65535
Respuesta del servidor bastión	UDP	BASTIÓN_DMZ_IPADDR	123	S	CHOKE_DMZ_IPADDR	1024:65535
Petición del servidor de contención	UDP	BASTIÓN_DMZ_IPADDR	123	E	CHOKE_DMZ_IPADDR	123
Respuesta del servidor principal bastión	UDP	BASTIÓN_DMZ_IPADDR	123	S	CHOKE_DMZ_IPADDR	123

Como cliente, la máquina bastión realiza peticiones de forma periódica a un proveedor de servicio de tiempo público. La máquina bastión ejecuta xntpd como servidor local para difundir el tiempo a las máquinas internas.

Las reglas de ejemplo para acceder a un servidor remoto se explican en el Capítulo 3. Estas son las reglas de servidor que permiten a los clientes DMZ acceder al servidor del bastión local:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p udp \
         -s $DMZ_LAN_ADDRESSES $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR 123 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s $BASTION_DMZ_IPADDR 123 \
         -d $DMZ_LAN_ADDRESSES $UNPRIVPORTS -j ACCEPT
```

Los intercambios de servidor de igual a igual se usan entre servidores para sincronizar su idea del tiempo actual. Una LAN grande puede configurar una máquina interna adicional para que funcione como un servidor de tiempo local secundario, ejecutando también xntpd en dicha máquina:

```
ipchains -A input  -i $BASTION_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR 123 \
         -d $BASTION_DMZ_IPADDR 123 -j ACCEPT

ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s $BASTION_DMZ_IPADDR 123 \
         -d $CHOKE_DMZ_IPADDR 123 -j ACCEPT
```

Configuración del cliente NTP de contención y del servidor de igual a igual de la DMZ

La Tabla 4.70 lista el protocolo de intercambio de servidor de igual a igual para el servicio NTP.

Tabla 4.70. Protocolos cliente/servidor NTP de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local
Petición del cliente de contención	UDP	BASTION_DMZ_IPADDR	123	S	CHOKE_DMZ_IPADDR	1024:65535
Respuesta del servidor bastión	UDP	BASTION_DMZ_IPADDR	123	E	CHOKE_DMZ_IPADDR	1024:65535
Petición del servidor de contención	UDP	BASTION_DMZ_IPADDR	123	S	CHOKE_DMZ_IPADDR	123
Respuesta del servidor igual de bastión	UDP	BASTION_DMZ_IPADDR	123	E	CHOKE_DMZ_IPADDR	123

Las dos reglas siguientes permiten el acceso del cliente al servidor de tiempo xntpd del bastión usando el programa cliente ntpdate:

```
ipchains -A output  -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR $UNPRIVPORTS \
         -d $BASTION_DMZ_IPADDR 123 -j ACCEPT

ipchains -A input -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $BASTION_DMZ_IPADDR 123 \
         -d $CHOKE_DMZ_IPADDR $UNPRIVPORTS -j ACCEPT
```

Si la máquina de contención está configurada como un servidor interno que ejecuta xntpd, se puede usar el protocolo de comunicación de igual a igual con las dos reglas siguientes:

```
ipchains -A output -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $CHOKE_DMZ_IPADDR 123 \
         -d $BASTION_DMZ_IPADDR 123 -j ACCEPT

ipchains -A input  -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $BASTION_DMZ_IPADDR 123 \
         -d $CHOKE_DMZ_IPADDR 123 -j ACCEPT
```

Inicio de sesión remota en el sistema (Puerto UDP 514)

Los archivos de registro del sistema se pueden administrar mediante una máquina servidor de registros central. El archivo de registro tiene algunas ventajas en entornos que albergan muchos servidores. Los archivos de registro de sistema y la configuración de syslogd se explican en el Capítulo 6, "Verificar que el sistema se ejecuta como se espera". El registro remoto, específicamente, se explica en el Capítulo 7, "Problemas a nivel de administración del sistema de UNIX".

Como ejemplo, en este caso supondremos que se dispone de una configuración en la que la máquina bastión escribe una copia de las entradas del registro del sistema al servidor syslogd de la máquina de contención.

Configuración del bastión como escritor local syslog de la DMZ

La Tabla 4.71 lista el protocolo de intercambio del servidor de igual a igual del escritor para el servicio remoto syslog.

Tabla 4.71. Protocolo del escritor *syslog* del bastión

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local
Registro del servidor bastión	UDP	CHOKE_DMZ_IPADDR	514	S	BASTION_DMZ_IPADDR	514

El archivo /etc/syslog.conf del bastión contiene una línea de configuración adicional para escribir una copia de todas las entradas del registro a la máquina de contención:

```
*.*             @choke
```

El intercambio de servidor de igual a igual se usa entre servidores syslogd. Los registros del sistema bastión se copian al servidor syslogd de la máquina de contención:

```
ipchains -A output -i $BASTION_DMZ_INTERFACE -p udp \
         -s $BASTION_DMZ_IPADDR 514 \
         -d $CHOKE_DMZ_IPADDR 514 -j ACCEPT
```

Configuración de la máquina de contención como lector local syslog de la DMZ

La Tabla 4.72 lista el protocolo de intercambio de servidor de igual a igual del lector para el cliente remoto syslog.

Tabla 4.72. Protocolo de lector *syslog* de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local
Registro del servidor bastión	UDP	BASTION_DMZ_IPADDR	514	In	CHOKE_DMZ_IPADDR	514

La secuencia de comandos de inicialización de la máquina de contención, /etc/rc.d/init.d/syslog, contiene una línea de inicio modificada, a la que se le ha agregado la opción -r para indicar al servidor que escuche el socket de dominio de Internet en el puerto UDP 514, al igual que para que escuche en los socket de dominio UNIX normal:

```
daemon syslogd -r
```

La siguiente regla permite los mensajes de registro entrantes desde el servidor syslogd del bastión:

```
ipchains -A input -i $CHOKE_DMZ_INTERFACE -p udp \
         -s $BASTION_DMZ_IPADDR 514 \
         -d $CHOKE_DMZ_IPADDR 514 -j ACCEPT
```

La máquina de contención como servidor DHCP local (Puertos UDP 67 y 68)

La Tabla 4.73 lista el protocolo de intercambio de servidor para el servicio DHCP.

Tabla 4.73. Protocolo del servidor DHCP de la máquina de contención

Descripción	Protocolo	Dirección remota	Puerto remoto	E/S	Dirección local	Puerto local
DHCPDISCOVER; DHCPREQUEST	UDP	0.0.0.0	68	E	255.255.255.255	67
DHCPOFFER	UDP	255.255.255.255	68	S	0.0.0.0	67
DHCPOFFER	UDP	255.255.255.255	68	S	CHOKE_LAN_IPADDR	67
DHCPREQUEST; DHCEPDECLINE	UDP	0.0.0.0	68	E	CHOKE_LAN_IPADDR	67
DHCPACK; DHCPNAK	UDP	CHOKE_LAN_NETMASK	68	S	CHOKE_LAN_IPADDR	67
DHCPACK	UDP	CHOKE_DMZ_IPADDR	68	S	CHOKE_LAN_IPADDR	67
DHCPREQUEST; DHCPRELEASE	UDP	CHOKE_DMZ_IPADDR	68	E	CHOKE_LAN_IPADDR	67

Aunque nunca se deben enviar mensajes de servidor DHCP a Internet, algunas personas ejecutan un servidor DHCP privado para asignar direcciones IP a máquinas locales. DHCP puede ser útil no sólo para asignar direcciones IP en una LAN grande con muchas máquinas, sino también para pequeñas LAN personales. De hecho, algunas personas con un sistema independiente de una sola máquina ejecutan el servidor dhcpd de forma local si usan un equipo portátil entre casa y el trabajo. Si el entorno de trabajo asigna direcciones IP de forma dinámica, el uso de DHCP en casa facilita el transporte del equipo portátil entre redes.

Para este ejemplo, el servidor dhcpd se está ejecutando en la máquina de contención, ofreciendo asignación de direcciones IP dinámicas para máquinas de la LAN privada:

```
ipchains -A input  -i $CHOKE_LAN_INTERFACE -p udp \
         -s $BROADCAST_0 68 \
         -d $BROADCAST_1 67 -j ACCEPT

ipchains -A output -i $CHOKE_LAN_INTERFACE -p udp \
         -s $BROADCAST_0 67 \
         -d $BROADCAST_1 68 -j ACCEPT

ipchains -A output -i $CHOKE_LAN_INTERFACE -p udp \
         -s $CHOKE_LAN_IPADDR 67 \
         -d $BROADCAST_1 68 -j ACCEPT

ipchains -A input  -i $CHOKE_LAN_INTERFACE -p udp \
         -s $BROADCAST_0 68 \
         -d $CHOKE_LAN_IPADDR 67 -j ACCEPT

ipchains -A output -i $CHOKE_LAN_INTERFACE -p udp \
         -s $CHOKE_LAN_IPADDR 67 \
         -d $CHOKE_LAN_ADDRESSES/CHOKE_LAN_NETMASK 68 -j ACCEPT
```

```
ipchains -A output -i $CHOKE_LAN_INTERFACE -p udp \
         -s $CHOKE_LAN_IPADDR 67 \
         -d $CHOKE_LAN_ADDRESSES 68 -j ACCEPT

ipchains -A input  -i $CHOKE_LAN_INTERFACE -p udp \
         -s $CHOKE_LAN_ADDRESSES 68 \
         -d $CHOKE_LAN_IPADDR 67 -j ACCEPT
```

Cómo habilitar el acceso LAN a la máquina firewall de contención

Para una configuración particular o de una pequeña empresa, existen pocas razones para restringir el acceso directo a la máquina firewall de contención desde la LAN interna. Este par de reglas permite abrir la comunicación entre la máquina de contención y la LAN privada:

```
ipchains -A input  -i $CHOKE_LAN_INTERFACE \
         -s $CHOKE_LAN_ADDRESSES -j ACCEPT

ipchains -A output -i $CHOKE_LAN_INTERFACE \
         -d $CHOKE_LAN_ADDRESSES -j ACCEPT
```

Cómo habilitar el enmascaramiento IP

No es necesario enmascarar el tráfico procedente de la LAN interna. El firewall bastión enmascara actualmente todo el tráfico interno procedente de la DMZ que cruza la interfaz externa. Para no olvidarnos de nada, se puede decir que el tráfico de LAN privado también se enmascara en la DMZ mediante el firewall de contención, en vez de simplemente enviarlo. Dependiendo de lo específica que sea la regla complementaria en el bastión, el enmascaramiento puede simplificar la regla del bastión si no se esperan direcciones procedentes de múltiples redes internas. La siguiente regla enmascara todo el trafico procedente de la LAN privada:

```
ipchains -A forward -i $CHOKE_DMZ_INTERFACE \
         -s $CHOKE_LAN_ADDRESSES -j MASQ
```

De igual forma, si se prefieren reglas de enmascaramiento más explícitas, las dos reglas siguientes habilitan el enmascaramiento para TCP e ICMP, pero no para UDP:

```
ipchains -A forward -i $CHOKE_DMZ_INTERFACE -p tcp \
         -s $CHOKE_LAN_ADDRESSES -j MASQ
ipchains -A forward -i $CHOKE_DMZ_INTERFACE -p icmp \
         -s $CHOKE_LAN_ADDRESSES -j MASQ
```

Registro

Registrar los paquetes descartados en las interfaces internas puede que no sea especialmente útil en los entornos relativamente seguros en los que se

centra este libro. Sin embargo, el registro es una herramienta principal para depurar problemas de firewalls y comprender los protocolos de comunicación. Como se permite el tráfico entre la LAN privada y la máquina de contención, el registro se habilitará de forma especifica en función del puerto.

Resumen

En este capítulo se explican algunas de las opciones disponibles a la hora de proteger una LAN. Las directivas de seguridad se definen de forma relativa al nivel de seguridad necesaria del sitio, la importancia de los datos que se están protegiendo y el coste que supondría la perdida de datos o de privacidad. Se empieza con una LAN particular sencilla y con el firewall programado en el Capítulo 3; posteriormente se explican las opciones de configuración de la LAN y del firewall, incrementado la complejidad de las configuraciones.

Lo que más se resalta en este capítulo es cómo usar el ejemplo de firewall del Capítulo 3 como base para programar un tipo de firewall de referencia, elaborado y formal. El firewall bastión tiene dos interfaces de red: una conectada a Internet y otra conectada a una red de perímetro o DMZ. Se ofrecen los servicios de Internet públicos desde máquinas que pertenecen a la red DMZ. Un segundo firewall, el firewall de contención, también se conecta a la red DMZ, separando las redes privadas internas de las máquinas de servidor casi públicas en la red de perímetro. Las máquinas privadas se protegen detrás del firewall de contención situado en la LAN interna. El firewall de contención protege la LAN interna de la máquina bastión comprometida y de cualquier otra máquina situada en la red de perímetro.

Algunos servicios, como IRC o RealAudio, no se prestan para filtrar paquetes, debido a sus protocolos de comunicación de aplicación, tales como solicitar conexiones entrantes desde el servidor o intercambios cliente/servidor múltiples tanto sobre TCP como UDP. Estos tipos de servicios requieren ayuda adicional, ya sea desde los módulos de compatibilidad de enmascaramiento que incluye el núcleo o desde servidores proxy del nivel de aplicación. También se presentan en este capítulo varios conjuntos de reglas de firewall para algunos de estos servicios, los que no se pueden proteger de forma sencilla mediante los ejemplos de firewall del Capítulo 3.

5

Depuración de las reglas del firewall

Ahora que está configurado, instalado y activado el firewall, ¡no funciona! El usuario se encuentra bloqueado y, peor aún, X Window tampoco funciona. ¿Quién sabe lo que sucede? ¿Qué hacer ahora? ¿Por dónde comenzar?

Es bastante difícil hacer que las reglas de firewall funcionen correctamente. Si se programa a mano, resulta inevitable que surjan errores. Incluso si se crea una secuencia de comandos de firewall con una herramienta de generación de firewall automática, seguro que se necesitará algo de personalización.

Este capítulo introduce características de creación de informes adicionales de la herramienta ipchains y de otras herramientas del sistema. La información no tiene precio cuando se depuran las reglas de firewall. Este capítulo explica lo que esta información puede mostrar acerca del firewall. Las herramientas son rudimentarias y ell proceso es tedioso. ¡Queda advertido!

Si necesita información adicional sobre las características de creación de informes de ipchains, consulte la página man sobre ipchains y el IPCHAINS-HOWTO.

Sugerencias generales sobre la programación de firewalls

A continuación se explica cómo realizar un seguimiento detallado y meticuloso de un problema existente en el firewall. No hay atajos a la hora de de-

purar las reglas cuando algo sale mal. En general, los siguientes consejos pueden facilitar algo el proceso:

- Ejecute siempre las reglas desde una secuencia de comandos completa de prueba. Asegúrese de que la secuencia de comandos purga todas las reglas existentes y restablece primero las directivas predeterminadas. Si se hace de otra forma, no se puede estar seguro de que reglas se ejecutan o en qué orden lo harán.
- No ejecute nuevas reglas desde la línea de comandos. Especialmente no ejecute las reglas de directiva predeterminada desde la línea de comandos. Se le impedirá el acceso inmediatamente si tiene una sesión iniciada en X Window, o una sesión iniciada de forma remota desde otro sistema, incluso si el sistema pertenece a la LAN.
- Ejecute la secuencia de comandos de prueba desde la consola. No pruebe a depurar desde una máquina remota. Trabajar desde X Window en la consola puede ser más cómodo, pero sigue presente el peligro de perder el acceso a X Window de forma local. Prepárese por si es necesario cambiar a una consola virtual para volver a obtener el control.
- Siempre que sea posible, trabaje en un servicio cada vez. Agregue reglas una a una o como pares de reglas de entrada y salida. Pruébelas conforme las introduce. Esto facilita mucho poder aislar directamente las áreas problemáticas de las reglas.
- Recuerde que la primera regla que coincide gana. El orden es importante. Use los comandos de listado de ipchains conforme trabaja para comprobar cómo se ordenan las reglas. Haga el seguimiento de un paquete imaginario a través de la lista.
- Recuerde que existen al menos dos cadenas independientes: input y output. Si las reglas input parecen correctas, el problema puede estar en la cadena output, o viceversa.
- Si la secuencia de comandos parece que se bloquea, es probable que exista una regla que hace referencia a un nombre de host simbólico en vez de a una dirección IP antes de haber habilitado las reglas DNS. Cualquier regla que use un nombre de host en vez de una dirección IP debe colocarse después de las reglas DNS.
- Realice una doble comprobación de la sintaxis de ipchains. Es fácil equivocarse en la dirección de la regla, invertir las direcciones o los puertos origen y destino, o cambiar opciones sensibles a las mayúsculas.
- Si encuentra un error de sintaxis, la secuencia de comandos del firewall finaliza sin instalar las siguientes reglas y los mensajes de error del programa ipchains son enigmáticos. Si tiene dificultades para identificar la regla problemática, ejecute la secuencia de comandos con la opción de shell -x o -v para listar las reglas conforme se ejecuta la secuencia de comandos; por ejemplo, sh -v /etc/rc.d/rc.firewall. La opción -v imprime la línea de la secuencia de comandos según la lee

el intérprete de comandos del shell. La opción -x imprime la línea de la secuencia de comandos a medida que la ejecuta el shell.

- Cuando un servicio no funciona, use la opción de registro de ipchains, -l, para registrar todos los paquetes denegados en ambas direcciones. ¿Muestran las entradas del registro en /var/log/messages que se ha denegado algo cuando se prueba el servicio?
- Si tiene acceso a Internet desde la máquina firewall pero no desde la LAN, tendrá que realizar una doble comprobación para verificar que el envío IP esté habilitado en el archivo /etc/sysconfig/network. Busque una línea que diga FORWARD_IPV4=yes. El envío IP se puede configurar manualmente de forma permanente en el archivo /etc/sysconfig/network o a través de la interfaz GUI del panel de control. La opción de envío IP se encuentra en la sección de enrutamiento de los diálogos de configuración de red del panel de control. Ninguno de estos métodos de configuración surte efecto hasta que la red se reinicia. Si el envío IP no estaba habilitado, se puede habilitar inmediatamente escribiendo la siguiente línea como root:

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

- Si un servicio funciona en la LAN pero no de forma externa, active el registro de paquetes aceptados en la interfaz interna. Use el servicio de forma breve para ver qué puertos, direcciones, indicadores, etc, están en uso en ambas direcciones. Es probable que no quiera registrar los paquetes aceptados durante mucho tiempo, o quizá tenga cientos o miles de entradas de registro en el archivo /var/log/messages.
- Si un servicio no funciona correctamente, inserte temporalmente reglas de entrada y salida al principio de la secuencia de comandos del firewall para aceptar todo en ambas direcciones y registre todo el tráfico con la opción -l. ¿Está ahora disponible el servicio? Si es así, revise las entradas del registro en el archivo /var/log/messages para ver qué puertos están en uso.

Cómo listar las reglas del firewall

Es una buena idea listar las reglas que se han definido para revisar que están instaladas y en el orden que se espera. La opción -L lista las reglas reales según existen en la tabla interna del núcleo. Las reglas se listan en el orden en que se comparan con un paquete.

El formato básico del comando list es:

```
ipchains -L input
ipchains -L output
ipchains -L forward
```

En lugar de usar ipchains para definir reglas reales, se puede utilizar para listar las reglas existentes desde la línea de comandos. La salida se dirige al terminal o puede redirigirse a un archivo.

En las siguientes secciones se usan cuatro reglas de ejemplo de la cadena input para mostrar las diferencias entre las distintas opciones de formato de listado disponibles y para explicar lo que significan los campos output. Usando las diferentes opciones de formato de listado, se listan las mismas cuatro reglas de ejemplo con diferentes grados de detalle y legibilidad. Las opciones de formato de listado son las mismas para las cadenas input, output y forward.

Entrada ipchains -L

La siguiente es una lista abreviada de cuatro reglas para una tabla de la cadena input usando las opciones de listado predeterminadas:

```
> ipchains -L input

1. Chain input (policy DENY):
2. target  prot  opt     source           destination      ports
3. DENY    all   --l-  mi.host.dominio  anywhere         n/a
4. ACCEPT  icmp  ----  anywhere         mi.host.dominio  echo-reply
5. ACCEPT  all   ----  anywhere         anywhere         n/a
6. ACCEPT  tcp   !y--  anywhere         mi.host.dominio  www-> 1024:65535
```

La línea 1 identifica el listado como procedente de la cadena input. La directiva predeterminada de la cadena input es DENY.

La línea 2 contiene los siguientes encabezados de columna:

- target se refiere a la disposición del destino de un paquete que coincide con la regla, ACCEPT, DENY o REJECT.
- prot es la abreviatura de protocolo, que puede ser all, tcp, udp o icmp.
- opt es la abreviatura de opciones de paquete, o bits de indicador. Los cuatro indicadores más habituales que se pueden ver serán none, l (registra un paquete que coincida con la regla), y (debe activarse el indicador SYN) e !y (debe activarse el indicador ACK).
- source es la dirección origen del paquete.
- destination es la dirección destino del paquete.
- ports lista tanto el puerto origen como el destino, el tipo de mensaje ICMP, o n/a si no se han especificado puertos en la regla.

La línea 3 es la regla de usurpamiento IP básica, que deniega los paquetes entrantes que dicen proceder de la máquina del usuario, con el registro habilitado.

La línea 4 acepta contestaciones entrantes ping a las peticiones salientes ping.

La línea 5 muestra cómo el comando -L de listado, sin argumentos que lo califiquen, carece de un detalle importante. La regla aparenta aceptar todos los paquetes entrantes, tcp, udp e icmp, procedentes de cualquier lugar. El de-

talle que se olvidó en este caso es la interfaz, lo. Esta es la regla que acepta todas las entradas en la interfaz de bucle invertido.

La línea 6 acepta los paquetes procedentes de servidores web remotos con los que se ha contactado. El protocolo es el tcp. Las respuestas entrantes de servidor hacia el navegador deben tener el bit ACK activado (!y). El puerto origen del servidor web es el puerto 80, www. El navegador web, como programa cliente que es, contactó con el servidor desde uno de los puertos no privilegiados, 1024–65535.

ipchains -L input -n

La opción -n muestra todos los campos como valores numéricos en vez de como nombres simbólicos. Esta opción puede ahorrar tiempo si las reglas del firewall usan muchos nombres de host simbólicos, pues de lo contrario sería necesario realizar búsquedas DNS antes de listarlas. Además, un intervalo de puertos indica mejor si se lista como 23:79 en lugar de telnet:finger.

Si se usan las mismas cuatro reglas de ejemplo de la cadena input , los siguientes muestran cómo aparece la salida del listado usando la opción numérica -n:

```
> ipchains -L input -n

1. Chain input (policy DENY):
2. target   prot   opt      source          destination     ports
3. DENY     all    ----l-   192.168.10.30   0.0.0.0/0       n/a
4. ACCEPT   icmp   ------   0.0.0.0/0       192.168.10.30   0 -> *
5. ACCEPT   all    ------   0.0.0.0/0       0.0.0.0/0       n/a
6. ACCEPT   tcp    !y---    0.0.0.0/0       192.168.10.30   80  -> 1024:65535
```

La línea 1 identifica el listado como procedente de la cadena input. La directiva predeterminada de la cadena input es DENY.

La línea 2 contiene los siguientes encabezados de columna:

- target se refiere a la disposición del destino de un paquete que coincide con la regla, ACCEPT, DENY o REJECT.
- prot es la abreviatura de protocolo, que puede ser all, tcp, udp o icmp.
- opt es la abreviatura de opciones de paquete, o bits de indicador. Los cuatro indicadores más habituales que se pueden ver serán none, l (registra un paquete que coincida con la regla), y (debe activarse el indicador SYN) e !y (debe activarse el indicador ACK).
- source es la dirección origen del paquete.
- destination es la dirección destino del paquete.
- ports lista tanto el puerto origen como el destino, el tipo de mensaje ICMP, o n/a si no se han especificado puertos en la regla.

La línea 3 es la regla de usurpamiento básica de IP, que deniega los paquetes entrantes que dicen proceder de una máquina cuya la dirección IP es 192.168.10.30, con el registro habilitado.

La línea 4 acepta contestaciones entrantes ping a las peticiones salientes ping.

La línea 5 muestra cómo el comando -L de listado, sin argumentos que lo califiquen, carece de un detalle importante. La regla aparenta aceptar todos los paquetes entrantes, tcp, udp e icmp, procedentes de cualquier lugar. El detalle que se olvidó en este caso es la interfaz, lo. Esta es la regla que acepta todas las entradas en la interfaz de bucle invertido.

La línea 6 acepta los paquetes procedentes de servidores web remotos con los que se ha contactado. El protocolo es el tcp. Las respuestas entrantes de servidor hacia el navegador deben tener el bit ACK activado (!y). El puerto origen del servidor web es el puerto 80, www. El navegador web, como programa cliente que es, contactó con el servidor desde uno de los puertos no privilegiados, 1024–65535.

ipchains -L input -v

La opción -v produce una salida más detallada, que incluye el nombre de la interfaz. La presentación del nombre de la interfaz es de especial ayuda cuando la máquina tiene más de una interfaz de red.

Usando las mismas cuatro reglas de ejemplo de la cadena input, las siguientes muestran el aspecto de la salida del listado usando la opción de detalle -v:

```
> ipchains -L input -v

1. Chain input (policy DENY: 60018 packets, 4120591 bytes):
2.  pkts bytes target prot opt   tosa tosx ifname * source  destination
ports
3.     0    0   DENY   all    ----l- 0xFF 0x00 eth0     mi.host.dominio    anywhere
n/a
4.     0    0   ACCEPT icmp   ------ 0xFF 0x00 eth0     anywhere           mi.host.dominio
echo-reply
5.  61004 5987K ACCEPT all    ------ 0xFF 0x00 lo       anywhere           anywhere n/a
6.   2332 1597K  ACCEPTtcp    !y---- 0xFF 0x00 eth0     anywhere           mi.host.dominio
www -> 1024:65535
```

La línea 1 identifica el listado como procedente de la cadena input. La directiva predeterminada de la cadena input es DENY. Han pasado 60018 paquetes a través de las reglas de la cadena input, contabilizando 4120591 bytes de tráfico de red.

La línea 2 contiene los siguientes encabezados de columna:

- pkts es el número de paquetes que coinciden con la regla.
- bytes es el número de bytes contenidos en los paquetes que han coincidido con la regla.
- target se refiere a la disposición del destino de un paquete que coincide con la regla, ACCEPT, DENY o REJECT.
- prot es la abreviatura de protocolo, que puede ser all, tcp, udp o icmp.

- opt es la abreviatura de opciones de paquete, o bits de indicador. Los cuatro indicadores más habituales que se pueden ver serán none, l (registra un paquete que coincida con la regla), y (debe activarse el indicador SYN) e !y (debe activarse el indicador ACK).
- Las máscaras Tipo de servicio (TOS, Type of Service), tosa y tosx, son máscaras and y xor, respectivamente. Cuatro de los bits corresponden a cuatro formas diferentes de manejar la prioridad de entrega del paquete: Minimum Delay (Demora mínima), Maximum Throughput (Máximo rendimiento), Maximum Reliability (Máxima confiabilidad) y Minimum Cost (Coste mínimo). Se puede especificar uno de los bits de prioridad de servicio y éste se activa en la máscara and para seleccionar el bit. Al resultado de la operación and se le hace un xor con la máscara xor para activar o borrar el indicador.
- ifname es el nombre de la interfaz de red, como eth0, eth1, lo o ppp0, a la que se aplica esta regla. Sólo los paquetes de esta interfaz específica de red coincidirán con la regla. Este campo es importante si se tiene una LAN con reglas de firewall independientes para las distintas interfaces.
- * es un contenedor para los dos campos que no se usan y se incluyen en la salida. En este caso se han reemplazado estos campos, mark y outsize, para ahorrar espacio:
  - mark no se usa o no está bien documentado todavía.
  - outsize no está documentado.
- source es la dirección origen del paquete.
- destination es la dirección destino del paquete.
- ports lista tanto el puerto origen como el destino, el tipo de mensaje ICMP, o n/a si no se han especificado puertos en la regla.

La línea 3 es la regla de usurpamiento básica de IP, que deniega los paquetes entrantes que dicen proceder de la máquina del usuario, con el registro habilitado.

La línea 4 acepta contestaciones entrantes ping a las peticiones salientes ping.

La línea 5 muestra cómo el comando -L de listado, con la opción de detalle, proporciona algún detalle clarificador cuando las reglas se refieren a múltiples interfaces. Con la opción -v, está claro que la regla se refiere a la interfaz de bucle invertido, lo. Esta es la regla que acepta todas las entradas en la interfaz de bucle invertido.

La línea 6 acepta los paquetes procedentes de servidores web remotos con los que se ha contactado. El protocolo es el tcp. Las respuestas entrantes de servidor hacia el navegador deben tener el bit ACK activado (!y). El puerto origen del servidor web es el puerto 80, www. El navegador web, como programa cliente que es, contactó con el servidor desde uno de los puertos no privilegiados, 1024–65535.

Indicadores Tipo de servicio (TOS, Type of Service)

Los bits TOS del encabezado del paquete no se suelen utilizar. Además, se puede usar como sugerencia o indicación de precedencia. Es probable que el emisor no cumpla los bits TOS. Si todo esto sólo sirve de confusión, ¡olvídelo! Casi nadie usará las máscaras TOS.

Los indicadores TOS forman parte del estándar oficial del protocolo IP. Se ha reservado espacio para ellos por adelantado para cuando llegue el día de usarlos. En su lugar, IP ha evolucionado ha evolucionado hacia un protocolo más sencillo y que requiera menos esfuerzo. Los estándares actuales de enrutadores sugieren volver a utilizar estos bits para una nueva información de calidad de servicio (QoS, Quality of Service) del enrutador.

Campos que no se usan, mark y outsize

mark y outsize son marcadores de posición para soportar funcionalidad que todavía no se ha implementado. *'Marcar'* un paquete está relacionado con la calidad de servicio (QoS, Quality of Service). Los bits TOS no se usan todavía, y las organizaciones de desarrollo y los comités de estándares están estudiando la posibilidad de reutilizar el campo TOS con nuevos bits de QoS, lo que se conoce con el nombre de *marcar paquetes*. QoS ofrecerá a los proveedores de servicio la posibilidad de ofrecer distintas capas de rendimiento de calidad de servicio a los clientes del negocio, así como a distinguir entre la necesidad de sesiones de telnet y un flujo de vídeo en tiempo real de gran ancho de banda.

ipchains -L input -nv

Usando las mismas reglas de ejemplo de la cadena input, los siguientes muestran cómo aparece la salida del listado si se usan juntas las opciones numérica -n y detallada -v:

```
> ipchains -L input -nv

1. Chain input (policy DENY: 60018 packets, 4120591 bytes):
2.  pkts bytes target prot opt    tosa tosx ifname * source        destination
ports
3.     0     0 DENY   all  ----l- 0xFF 0x00 eth0      192.168.10.30 0.0.0.0/0
n/a
4.     0     0 ACCEPT icmp ------ 0xFF 0x00 eth0      0.0.0.0/0     192.168.10.30 0
-> *
5. 61004 5987K  ACCEPT all  ------ 0xFF 0x00 lo        0.0.0.0/0     0.0.0.0/0
n/a
6.  2332 1597K  ACCEPT tcp  !y---- 0xFF 0x00 eth0      0.0.0.0/0     192.168.10.30
80 -> 1024:65535
```

La línea 1 identifica el listado como procedente de la cadena input. La directiva predeterminada de la cadena input es DENY. Han pasado 60018 paquetes a través de las reglas de la cadena input, contabilizándose 4120591 bytes de tráfico de red.

La línea 2 contiene los siguientes encabezados de columna:

- pkts es el número de paquetes que coinciden con la regla.
- bytes es el número de bytes contenidos en los paquetes que coinciden con la regla.
- target se refiere a la disposición del destino de un paquete que coincide con la regla, ACCEPT, DENY o REJECT.
- prot es la abreviatura de protocolo, que puede ser all, tcp, udp o icmp.
- opt es la abreviatura de opciones de paquete, o bits de indicador. Los cuatro indicadores más habituales que se pueden ver serán none, l (registra un paquete que coincida con la regla), y (debe activarse el indicador SYN) e !y (debe activarse el indicador ACK).
- Las máscaras Tipo de servicio (TOS, Type of Service), tosa y tosx, son máscaras and y xor, respectivamente. Cuatro de los bits corresponden a cuatro formas diferentes de manejar la prioridad de entrega del paquete: Minimum Delay (Demora mínima), Maximum Throughput (Máximo rendimiento), Maximum Reliability (Máxima confiabilidad) y Minimum Cost (Coste mínimo). Se puede especificar uno de los bits de prioridad de servicio y éste se activa en la máscara and para seleccionar el bit. El resultado de la operación and se hace un xor con la máscara xor para activar o borrar el indicador.
- ifname es el nombre de la interfaz de red, como eth0, eth1, lo o ppp0, a la que se aplica esta regla. Sólo los paquetes de esta interfaz específica de red coincidirán con la regla. Este campo se convierte en algo importante si se tiene una LAN con reglas de firewall independientes para las distintas interfaces.
- * es un contenedor para los dos campos que no se usan y se incluyen en la salida. En este caso se han reemplazado estos campos, mark y outsize, para ahorrar espacio:
  - mark no se usa o no está todavía bien documentado.
  - outsize no está documentado.
- source es la dirección origen del paquete.
- destination es la dirección destino del paquete.
- ports lista tanto el puerto origen como el destino, el tipo de mensaje ICMP, o n/a si no se han especificado puertos en la regla.

La línea 3 es la regla de usurpamiento básica de IP, que deniega los paquetes entrantes que dicen proceder de la máquina del usuario, con el registro habilitado.

La línea 4 acepta contestaciones entrantes ping a las peticiones salientes ping.

La línea 5 muestra cómo el comando -L de listado, con la opción de detalle, proporciona algún detalle clarificador cuando las reglas se refieren a múltiples interfaces. Con la opción -v, está claro que la regla se refiere a la interfaz de bucle invertido, lo. Esta es la regla que acepta todas las entradas en la interfaz de bucle invertido.

La línea 6 acepta los paquetes procedentes de servidores web remotos con los que se ha contactado. El protocolo es el tcp. Las respuestas entrantes

de servidor hacia el navegador deben tener el bit ACK activado (!y). El puerto origen del servidor web es el puerto 80, www. El navegador web, como programa cliente que es, contactó con el servidor desde uno de los puertos no privilegiados, 1024–65535.

Comprobación de las reglas de entrada, salida y reenvío

Una vez visto cómo se presenta el listado de la cadena de un firewall y las opciones de formato disponibles, se examinarán unas breves listas de entrada, salida y reenvío. Las reglas de ejemplo son representativas de algunas de las reglas que es probable que se usen con más frecuencia.

Comprobación de las reglas de entrada

Las reglas de entrada son principalmente reglas ACCEPT cuando la directiva predeterminada es DENY. Se deniega todo de forma predeterminada y es necesario definir explícitamente lo que se acepta. El siguiente ejemplo contiene un ejemplo representativo de reglas de aceptación de entrada:

```
> ipchains -L input
    Chain input (policy DENY):
    target prot opt    source          destination    ports
1.  ACCEPT icmp ------ anywhere        mi.host.dominio
    destination-unreachable
2.  ACCEPT icmp ------ anywhere        mi.host.dominio source-quench
3.  ACCEPT icmp ------ anywhere        mi.host.dominio time-exceeded
4.  ACCEPT icmp ------ anywhere        mi.host.dominio parameter-problem
5.  ACCEPT udp  ------ isp.name.server mi.host.dominio domain -> domain
6.  ACCEPT udp  ------ isp.name.server mi.host.dominio domain ->
    1024:65535
7.  ACCEPT tcp  !y---- isp.name.server mi.host.dominio domain ->
    1024:65535
8.  REJECT tcp  ------ anywhere        mi.host.dominio 1024:65535 -> auth
9.  ACCEPT tcp  !y---- anywhere        mi.host.dominio auth -> 1024:65535
10. ACCEPT tcp  ------ anywhere        mi.host.dominio 1024:65535 -> www
11. ACCEPT tcp  !y---- anywhere        mi.host.dominio www -> 1024:65535
12. ACCEPT tcp  !y---- isp.news.server mi.host.dominio nntp -> 1024:65535
13. ACCEPT tcp  ------ anywhere        mi.host.dominio 1024:65535 -> smtp
14. ACCEPT tcp  !y--- anywhere         mi.host.dominio smtp -> 1024:65535
```

La directiva predeterminada para los paquetes entrantes es DENY. Los paquetes denegados simplemente se descartan sin devolver ninguna notificación a las direcciones origen. Hay 14 reglas en la cadena:

- Línea 1: Los mensajes de error ICMP Destination Unreachable entrantes se aceptan, procedan de donde procedan.

- Línea 2: Los mensajes de flujo de control ICMP Source Quench entrantes se aceptan, procedan de donde procedan..
- Línea 3: Los mensajes de error ICMP Time Exceeded entrantes se aceptan, procedan de donde procedan.
- Línea 4: Los mensajes de error ICMP Parameter Problem entrantes se aceptan, procedan de donde procedan.
- Línea 5: Se acepta un paquete UDP con la misma dirección origen que el servidor de nombres del ISP si los puertos origen y destino son ambos el puerto 53 del servicio domain. Este tipo de paquete debería ser una comunicación servidor a servidor, en la que el servidor de nombres local envía una petición de búsqueda al servidor de nombres del ISP. Este paquete contiene la respuesta a la petición.
- Línea 6: Se acepta un paquete con la dirección origen igual al servidor de nombres del ISP si el puerto origen es el puerto 53 del servicio domain, dirigido a esta interfaz y a un puerto destino no privilegiado. Este tipo de paquete debería ser una respuesta UDP servidor a cliente con la información de la búsqueda DNS que solicitó un cliente de la máquina.
- Línea 7: Un paquete TCP con la dirección origen del servidor de nombres del ISP se acepta si el puerto origen es el puerto 53 del servicio domain, el bit ACK está activado en el campo indicador del paquete, y el puerto destino pertenece al intervalo de los puertos no privilegiados. Si la máquina está configurada como un cliente DNS, este paquete debería ser una respuesta TCP de servidor a cliente con la información de la búsqueda DNS que ha solicitado un cliente de la máquina, después de una petición inicial UDP errónea.
- Línea 8: Se rechazan todos los paquetes destinados al servidor local identd en el puerto 113 del servicio auth. Se devuelve una notificación de error ICMP de tipo 3 a la dirección origen.
- Línea 9: Se aceptan todos los paquetes entrantes TCP procedentes del puerto 113 del servicio auth y dirigidos a un puerto destino no privilegiado, suponiendo que el bit ACK está activado. Estos paquetes contienen respuestas a peticiones auth que inició uno de los clientes.
- Línea 10: Se aceptan los paquetes TCP entrantes procedentes de cualquier dirección origen con un puerto origen no privilegiado si están dirigidos al puerto de servicio del servidor web, el puerto 80. Se aceptan los paquetes con el indicador SYN o sin el indicador ACK.
- Línea 11: Se aceptan los paquetes de respuesta TCP entrantes desde cualquier dirección origen y del puerto origen 80, si están dirigidos a un puerto no privilegiado local y el bit ACK está activado. Estos paquetes son respuestas entrantes de servidores web remotos con los que se ha contactado a través del navegador web.
- Línea 12: Se aceptan los paquetes de respuesta entrantes TCP procedentes de una dirección origen igual al servidor de noticias del ISP y del puerto origen nntp 119, si están dirigidos a un puerto no privilegiado local y el bit ACK está activado. Estos paquetes son respuestas

entrantes procedentes del servidor de noticias del ISP cuando se leen noticias de Usenet.

- Línea 13: Se aceptan los paquetes TCP entrantes procedentes de cualquier dirección origen con un puerto origen no privilegiado, si están dirigidos al puerto de servicio smtp del servidor de correo, el puerto 25. Se aceptan los paquetes con los indicador SYN o ACK para permitir las conexiones entrantes al servidor y poder aceptar correo entrante.
- Línea 14: Se aceptan los paquetes de respuesta entrantes TCP procedentes de cualquier dirección origen y con el puerto origen smtp 25, si están dirigidos a un puerto no privilegiado local y el bit ACK está activado. Estos paquetes son respuestas entrantes procedentes de servidores de correo con los que se ha contactado para enviar correo saliente.

Comprobación de las reglas de salida

Las reglas de salida son principalmente reglas ACCEPT cuando la directiva predeterminada es REJECT o DENY. Se bloquea todo de forma predeterminada. Es necesario definir explícitamente lo que se aceptará. El siguiente ejemplo contiene un ejemplo representativo de reglas de aceptación de salida:

```
> ipchains -L output
    Chain output (policy REJECT):
    target prot opt    source          destination       ports
1.  ACCEPT icmp ------ mi.host.dominio anywhere          destination-unreachable
2.  ACCEPT icmp ------ mi.host.dominio anywhere          source-quench
3.  ACCEPT icmp ------ mi.host.dominio isp.address.range time-exceeded
4.  ACCEPT icmp ------ mi.host.dominio anywhere          parameter-problem
5.  ACCEPT udp  ------ mi.host.dominio isp.name.server   domain -> domain
6.  ACCEPT udp  ------ mi.host.dominio isp.name.server   1024:65535-> domain
7.  ACCEPT tcp  ------ mi.host.dominio isp.name.server   1024:65535-> domain
8.  ACCEPT tcp  ------ mi.host.dominio anywhere          1024:65535 -> auth
9.  ACCEPT tcp  !y---- mi.host.dominio anywhere          www -> 1024:65535
10. ACCEPT tcp  ------ mi.host.dominio anywhere          1024:65535 -> www
11. ACCEPT tcp  ------ mi.host.dominio isp.news.server   1024:65535 -> nntp
12. ACCEPT tcp  !y---- mi.host.dominio anywhere          smtp -> 1024:65535
13. ACCEPT tcp  ------ mi.host.dominio anywhere          1024:65535 -> smtp
```

La directiva predeterminada para la cadena output es REJECT. Los programas locales recibirán una observación de error inmediatamente si una conexión no tiene permiso. La cadena tiene 13 reglas:

- Línea 1: Se permiten los mensajes de error ICMP de tipo 3 salientes a cualquier parte. Aunque el mensaje se identifica como un mensaje Destination Unreachable, en realidad es un tipo de mensaje de error general. El mensaje contiene un campo que especifica el tipo específico de código de error.
- Línea 2: Se permite que los mensajes de flujo de control ICMP Source Quench se dirijan a cualquier parte.

- Línea 3: Los mensajes ICMP Time Exceeded sólo se pueden enviar a las máquinas del ISP. Esta regla restringe las respuestas a solicitudes traceroute sólo al ISP.
- Línea 4: Se permite que los mensajes de error ICMP Parameter Problem se dirijan a cualquier parte.
- Línea 5: Se acepta un paquete UDP con la dirección destino igual a la dirección del servidor de nombres del ISP, si los puertos origen y destino son los dos el puerto 53 del servicio domain. Este tipo de paquete debería ser una comunicación servidor a servidor, en la que el servidor de nombres local envía una petición de búsqueda al servidor de nombres del ISP después de que falle una búsqueda en la caché local.
- Línea 6: Se acepta un paquete UDP con la dirección destino igual a la dirección del servidor de nombres del ISP, si el puerto origen es un puerto no privilegiado y el puerto destino es el puerto 53 del servicio domain, Este tipo de paquete debería ser una petición UDP cliente a servidor de información de búsqueda DNS desde un cliente de la máquina del usuario al servidor remoto.
- Línea 7: Se acepta un paquete TCP con la dirección destino igual a la dirección del servidor de nombres del ISP, si el puerto origen es un puerto no privilegiado, el puerto destino es el puerto 53 del servicio domain y están activados el indicador SYN o el indicador ACK en el campo indicador del paquete. Si se ha configurado la máquina como un cliente DNS, este tipo de paquete debería ser una petición TCP cliente a servidor de información de búsqueda DNS procedente de un cliente de la máquina del usuario después de una petición UDP inicial fallida.
- Línea 8: Se aceptan todos los paquetes salientes TCP procedentes de un puerto origen no privilegiado y dirigidos al puerto 113 del servicio auth. Estos paquetes contienen peticiones a un servidor auth identd que ha iniciado alguno de los clientes.
- Línea 9: Se aceptan los paquetes respuesta TCP salientes a cualquier dirección destino con un puerto de servicio destino no privilegiado, si se originaron en el puerto 80 del servicio del servidor web. El bit ACK debe estar activado, porque estos paquetes son las respuestas del servidor web a las peticiones entrantes.
- Línea 10: Se aceptan los paquetes petición TCP salientes dirigidos a cualquier lugar y con el puerto destino 80, si se originan en un puerto no privilegiado local. Estos paquetes son conexiones salientes con servidores web remotos que se ha contactado a través del navegador web.
- Línea 11: Se aceptan los paquetes petición TCP salientes dirigidos a la dirección IP del servidor de noticias del ISP y al puerto destino 119 del servicio nntp, si se originan desde un puerto no privilegiado local. Estos paquetes son peticiones salientes y conexiones activas con el servidor de noticias del ISP cuando se leen noticias de Usenet.
- Línea 12: Se aceptan los paquetes TCP salientes procedentes del puerto 25 del servicio smtp del servidor de correo, si están dirigidos a un

puerto destino no privilegiado y el bit ACK está activado. Estos paquetes son respuestas salientes a clientes de correo remotos que han contactado con el usuario para entregar el correo entrante.

- Línea 13: Se aceptan los paquetes TCP salientes procedentes de un puerto no privilegiado, si están dirigidos al puerto destino 25 del servicio smtp. Estos paquetes son peticiones de conexión salientes y conexiones activas con servidores de correo remotos con los que se ha contactado para enviar correo saliente.

Comprobación de las reglas de reenvío

Las reglas de reenvío se encuentran, evidentemente, entre las reglas de entrada y de salida. Un paquete entrante debe aceptarse primero en la cadena input. Si la dirección destino del paquete es diferente de la dirección de la interfaz por la que llegó el paquete, el paquete se pasa a la cadena forward. Si el paquete coincide con una regla de reenvío, el paquete se pasa a la siguiente cadena output de la interfaz. Si la cadena output acepta el paquete, se envía en última instancia al exterior hacia su destino.

En estos ejemplos, la regla de firewall que se muestra a continuación envía y enmascara sólo el tráfico TCP procedente de la interfaz de red interna. El tráfico UDP no se enruta. El tráfico ICMP general no se enruta.

Esta sección se basa en una regla cualquiera de ejemplo de reenvío. Como se puede ver, no es necesario un conjunto minucioso de reglas de reenvío en una configuración particular pequeña:

```
ipchains -A forward -i $EXTERNAL_INTERFACE -p tcp -s $INTERNAL_LAN_ADDRESSES -j
MASQ

> ipchains -L forward -v

Chain forward (policy REJECT: 0 packets, 0 bytes):
 pkts bytes target prot opt    tosa tosx ifname source destination ports
   80  4130 MASQ   tcp  ------ 0xFF 0x00 eth0   choke  anywhere    any->any
```

La directiva predeterminada es REJECT, de forma que no se enviará ningún paquete sin una regla explícita.

Enmascaramiento ICMP en Linux

Hasta ahora, los mensajes ICMP que no fueran mensajes de error no se enmascaraban, a menos que se compilara de forma explícita un servicio de enmascaramiento en el núcleo. El servicio no estaba habilitado de forma predeterminada. En la versión 6.0 de Red Hat, el enmascaramiento ICMP está habilitado de forma predeterminada. Esto significa que es posible que una única máquina haga ping a un host remoto o ejecute traceroute sin necesidad de volver a compilar el núcleo.

En este caso, la opción -v también puede ser de ayuda. La regla dice que los paquetes que llegan desde la LAN interna, $INTERNAL_LAN_ADDRES-

SES, se reenvían a la interfaz externa, eth0. Se reenvían a la interfaz externa sólo en aquellos casos en que la dirección destino sea una dirección remota. La interfaz LAN interna no puede definirse explícitamente con la semántica ipchains. La interfaz depende de la dirección origen LAN del paquete. Un paquete con dirección origen LAN, que llega a la interfaz de red interna, no pertenece a la misma red que la interfaz de red externa. Si la dirección destino del paquete no es la dirección de la interfaz de red interna en esta máquina, la dirección destino es necesariamente una dirección remota, relativa a la interfaz interna.

Tenga en cuenta que la interfaz externa rechazará estos paquetes si no estuviese habilitado el enmascaramiento. Las reglas de enmascaramiento se aplican cuando el paquete abandona la cola de entrada de la interfaz interna, antes de transferir el paquete a la cola de salida de la interfaz externa. Si se supone que el firewall rechaza los paquetes salientes que tienen direcciones origen privadas de red, se descartarán los paquetes procedentes de la LAN en la interfaz externa si el enmascaramiento no estuviese habilitado en la regla de reenvío.

Lo siguiente a tener en cuenta es que sólo el protocolo TCP está enmascarado. Ni los paquetes UDP ni los mensajes de control ICMP se enviarán al exterior a través de la interfaz externa. De nuevo, incluso si estos paquetes tienen permiso para salir de la máquina firewall, no tienen permiso para salir en nombre de la máquina interna, porque no se han especificado estos paquetes en la regla de reenvío y enmascaramiento.

Un sitio pequeño puede usar cómodamente la siguiente regla más general para enmascarar todo el tráfico interno destinado a direcciones remotas, en vez de sólo los paquetes TCP. Cualquier tipo de tráfico que las reglas de salida permiten salir al exterior para la interfaz externa, tendrá también permiso para salir a través de la máquina interna:

```
ipchains -A forward -i $EXTERNAL_INTERFACE -s $INTERNAL_LAN_ADDRESSES
    -j MASQ
```

Cómo comprobar un paquete concreto con las reglas de firewall

Se pueden probar tipos individuales de paquetes con las reglas de firewall usando la opción –c. ipchains mostrará un informe en la salida estándar stdout si se aceptó, denegó, rechazó o enmascaró el paquete, basándose en las reglas que se aplican en este momento.

Usar la opción –c, en lugar de usar una opción de insertar -I o de agregar -A, indica a ipchains que se está creando una descripción del paquete. Es decir, se quiere saber cómo manejarán las reglas de firewall instaladas actualmente este tipo de paquete.

Existen algunas diferencias en la sintaxis de ipchains entre la definición de las reglas reales y la definición de descripciones de paquetes de prueba. La

sintaxis –c está cayendo en desuso. Las diferencias pueden confundir al usuario, porque las ilegalidades que se producen usando la opción –c son perfectamente legales a la hora de definir una regla real.

ipchains no implica los valores predeterminados cuando se usa la opción –c. Es necesario especificar valores de argumentos de línea de comandos exactos. Las descripciones de los paquetes de prueba deben usar la opción -i para especificar una interfaz. Deben especificarse puertos origen y destino explícitos individuales. No se pueden utilizar intervalos, no están permitidos. Por lo tanto, en los siguientes ejemplos, la constante $UNPRIVPORTS se reemplaza por el puerto 5000, un puerto individual de dicho intervalo. Como es necesario especificar los puertos origen y destino, también debe especificar las direcciones origen y destino. No se permite el operador de negación, !. Se puede usar la opción -y, pero no se puede usar la opción ! -y. Se pueden utilizar tanto los identificadores numéricos como los nombres simbólicos para las direcciones y los puertos origen y destino.

Los siguientes ejemplos -C de ipchains suponen que se tiene un firewall instalado, según se describe en el Capítulo 3, "Creación e instalación de un firewall."

Un paquete que coincida con la siguiente descripción se deniega, incluso si se alberga un servidor web. Las reglas recomendadas contra el usurpamiento de direcciones IP no permiten los paquetes entrantes que dicen proceder de la dirección origen:

```
> ipchains -C input -i eth0 -p tcp -y \
           -s <mi.host.dominio> 5000 \
           -d <mi.host.dominio> 80

denied
```

Un paquete que coincida con esta descripción se acepta si se alberga un servidor web. Este tipo de paquete se espera como parte de la conexión activa entre el servidor y un cliente web remoto:

```
> ipchains -C output -i eth0 -p tcp \
           -s <mi.host.dominio> 80 \
           -d any/0 5000

accepted
```

Un paquete saliente que coincida con esta descripción se rechaza en lugar de denegarse. Los ejemplos de firewall de este libro deniegan los paquetes entrantes y rechazan los paquetes salientes. La opción -y indica que el indicador SYN debe estar activado, lo que significa que el servidor web solicita la conexión. El servidor web no puede iniciar una conexión a un cliente remoto:

```
> ipchains -C output -i eth0 -p tcp -y \
           -s <mi.host.dominio> 80 \
           -d any/0 5000

rejected
```

Un paquete que coincida con esta descripción está enmascarado. Cualquier paquete procedente de una máquina de la LAN y destinada a una dirección irresoluble (es decir, una dirección remota) se enmascara y se envía al exterior a través de la interfaz externa:

```
> ipchains -C forward -i eth0 -p tcp \
           -s <my.lan.ip.address> 5000 \
           -d any/0 80

masqueraded
```

El ejemplo de enmascaramiento muestra otra diferencia entre la sintaxis de la regla de ipchains normal y la sintaxis de la regla –c. No se permite -j MASQ. Sólo se puede especificar la cadena forward.

Comprobación de los puertos abiertos

Listar las reglas del firewall con ipchains -L es la principal herramienta disponible para comprobar los puertos abiertos. Los puertos abiertos se definen para que se abran por las reglas ACCEPT. La versión Linux de Red Hat no incluye ninguna otra herramienta relacionada, además de netstat, para identificar puertos abiertos. Sin embargo, se pueden conseguir muchas herramientas de otros fabricantes en Internet.

netstat se usa para varios propósitos. En la siguiente sección se usará para comprobar los puertos activos y volver a comprobar que los puertos TCP y UDP en uso son los puertos que contabilizan las reglas de firewall. netstat vuelve a aparecer en el Capítulo 6, "Comprobación de que el sistema funciona como se espera", donde se usa para un propósito algo diferente.

A continuación, se introducen dos herramientas de exploración de otros fabricantes, strobe y nmap.

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

netstat crea un informe con distinta información de estado de la red. Documentaremos unas cuantas opciones de la línea de comandos para seleccionar la información que mostrará netstat. Las siguientes opciones son útiles para identificar puertos abiertos, informando si están en uso actualmente y por quién, e informando sobre qué programas y procesos específicos escuchan en los puertos:

- -a lista todos los puertos que están en uso actualmente o escuchan los servidores locales.
- -n muestra los nombres de host y los identificadores de puerto en formato numérico. Sin la opción -n, los nombres de los host y los identificadores de los puertos aparecen como nombres simbólicos, tantos como quepan en 80 columnas. El uso de la opción -n evita una espera potencialmente larga mientras los nombres de host se traducen a di-

recciones físicas. Si no se usa la opción –n, se producirá un listado más legible.

- -p lista el nombre del programa que está escuchando en el socket. Es necesario tener una sesión abierta como root para usar la opción -p.
- -A inet especifica la familia de direcciones sobre la que se informa. El listado incluye los puertos en uso según se asocian con las tarjetas de interfaz de red. No se informa sobre las conexiones de la familia de socket de direcciones UNIX locales, incluyendo las conexiones locales basadas en la red que usan otros programas, como cualquier programa X Window que se pueda estar ejecutando.

La siguiente salida del programa netstat se restringe a los sockets de dominio INET. El listado informa de todos los puertos que escuchan los servicios de red, incluyendo el nombre del programa y de los identificadores de proceso específicos de los programas agentes:

```
> netstat -a -p -A inet

1.  Active Internet connections (servers and established)
2.  Proto Recv-Q Send-Q Local Address     Foreign Address State    PID/
    Program name

3.  tcp      0    143 internal:telnet     macintosh:62360 ESTABLISHED
    15392/in.telnetd
4.  tcp      0      0 *:smtp              *:*             LISTEN
    3674/sendmail: acce
5.  tcp      0      0 mi.host.dominio:www *:*             LISTEN   638/httpd
6.  tcp      0      0 internal:domain     *:*             LISTEN   588/named
7.  tcp      0      0 localhost:domain    *:*             LISTEN   588/named
8.  tcp      0      0 *:auth              *:*             LISTEN   574/inetd
9.  tcp      0      0 *:pop-3             *:*             LISTEN   574/inetd
10. tcp      0      0 *:telnet            *:*             LISTEN   574/inetd
11. tcp      0      0 *:ftp               *:*             LISTEN   574/inetd
12. udp      0      0 *:domain            *:*                      588/named
13. udp      0      0 internal:domain     *:*                      588/named
14. udp      0      0 localhost:domain    *:*                      588/named
```

La línea 1 indica que el listado incluye conexiones de servidores locales y conexiones activas de Internet. Esta selección se le indicó al programa netstat con la opción –A inet.

La línea 2 contiene los siguientes encabezados de columna:

- Proto se refiere al protocolo de transporte sobre el que se ejecuta el servicio, TCP o UDP.
- Recv-Q es el número de bytes recibidos desde el host remoto, pero que todavía no se han entregado al programa local.
- Send-Q es el número de bytes enviados desde el programa local que todavía no se han confirmado por el host remoto.
- Local Address es el socket local, la interfaz de red y el par de puertos de servicio.

- Foreign Address es el socket remoto, la interfaz de red remota y el par de puertos de servicio.
- State es el estado de conexión del socket local para los socket que usan el protocolo TCP, ESTABLISHED (conexión establecida), LISTENing para una petición de conexión, así como un número de estados de establecimiento de conexión intermedia y de cierre.
- PID/Program name es el identificador del proceso (PID) y el nombre del programa al que pertenece el socket local.

La línea 3 muestra que existe una conexión telnet activa sobre la interfaz LAN de la red interna procedente de un Macintosh. El comando netstat se ha introducido desde esta conexión.

La línea 4 muestra que el programa sendmail escucha el correo entrante en el puerto del servicio SMTP asociado con todas las interfaces de red, incluyendo la interfaz externa conectada a Internet, la interfaz LAN interna y la interfaz del host local de bucle invertido.

La línea 5 muestra que existe un servidor web que escucha las conexiones en la interfaz externa a Internet.

La línea 6 muestra que el servidor de nombres escucha, en la interfaz LAN interna, las peticiones de conexión de búsqueda DNS desde máquinas locales sobre TCP.

La línea 7 muestra que el servidor de nombres escucha, en la interfaz de bucle invertido, las peticiones de conexión de búsqueda DNS desde clientes en esta maquina sobre TCP.

La línea 8 muestra que inetd escucha las conexiones en el puerto del servicio auth asociado con todas las interfaces a favor de identd.

La línea 9 muestra que inetd escucha las conexiones en el puerto pop-3 asociado con todas las interfaces, a favor de popd (inetd escucha en todas las interfaces las conexiones POP entrantes. Si llega una petición de conexión, inetd inicia un servidor popd para atender la petición). Tanto el firewall como los mecanismos de seguridad de más alto nivel en el nivel tcp_wrapper y el nivel de configuración de popd restringen las conexiones entrantes a las máquinas LAN.

La línea 10 muestra que inetd escucha las conexiones en el puerto telnet asociado con todas las interfaces a favor de telnetd (inetd escucha en todas las interfaces las conexiones telnet entrantes. Si llega una petición de conexión, inetd inicia un servidor telnetd para atender la petición.) Tanto el firewall como los mecanismos de seguridad de alto nivel a nivel de tcp_wrapper restringen las conexiones entrantes a las máquinas LAN.

La línea 11 muestra que inetd escucha las conexiones en el puerto ftp asociado con todas las interfaces a favor de ftpd. (inetd escucha en todas las interfaces las conexiones telnet entrantes. Si llega una petición de conexión, inetd inicia un servidor telnetd para atender la petición.) Tanto el firewall como los mecanismos de seguridad de alto nivel a nivel de tcp_wrapper y el nivel de configuración del servidor ftpd restringen las conexiones entrantes a las máquinas LAN.

La línea 12 muestra que el servidor de nombres escucha, en todas las interfaces, las comunicaciones servidor a servidor DNS y acepta las peticiones de búsqueda locales sobre UDP.

La línea 13 muestra que el servidor de nombres escucha la interfaz de red LAN interna esperando las comunicaciones servidor a servidor DNS y las peticiones de búsqueda sobre UDP.

La línea 14 muestra que el servidor de nombres escucha la interfaz de bucle invertido esperando las peticiones de búsqueda DNS desde clientes locales en esta máquina sobre UDP.

Los servidores ociosos que escuchan el protocolo TCP se muestran como LISTENing para una petición de conexión. Los servidores detenidos que escuchan sobre el protocolo UDP se muestran en blanco. UDP no tiene estado. La salida de netstat simplemente está haciendo una distinción entre el protocolo TCP con estado y el protocolo UDP sin estado.

Las siguientes secciones describen dos herramientas de otros fabricantes disponibles en Internet, strobe y nmap.

Convenciones para los informes de salida de netstat

En la salida de netstat, las direcciones locales y externas (es decir, remotas) se listan como <dirección:puerto>. Debajo de la columna Local Address, la dirección es el nombre o dirección IP de una de las tarjetas de interfaz de red. Cuando la dirección aparece en el listado como *, significa que el servidor escucha en todas las interfaces de red, en lugar de en una única interfaz. El puerto es el identificador, numérico o simbólico, del puerto de servicio que usa el servidor. Debajo de la columna Foreign Address, la dirección es el nombre de la dirección IP del cliente remoto que participa actualmente en una conexión. Cuando el puerto está ocioso, aparece *.* en el listado. El puerto es el puerto del cliente en su extremo.

---

strobe

strobe es una exploración de puerto TCP simple. Se debe usar para hacer un informe sobre qué puertos TCP están abiertos en las interfaces de red. Se puede conseguir la herramienta strobe en la dirección http://metalab.unc.edu/pub/Linux/system/network/admin.

La siguiente salida de ejemplo de strobe informa sobre los puertos TCP en los que strobe ha encontrado servidores escuchando. La salida predeterminada de strobe incluye el nombre del host explorado y la entrada de /etc/services que describe el puerto. Si se tiene un firewall instalado, puede haber servidores adicionales ejecutándose en la máquina, así como otros ocultos detrás de puertos bloqueados de forma pública:

```
> strobe firewall

strobe 1.02 (c) 1995 Julian Assange -Proff- (proff@suburbia.apana.org.au).
firewall     ssh         22/tcp # SSH Protocolo de inicio de sesión remoto
firewall     smtp        25/tcp mail
firewall     domain      53/tcp nameserver    # servidor nombre-dominio
firewall     www         80/tcp http          # WorldWideWeb HTTP
firewall     auth       113/tcp authentication tap ident
```

nmap

nmap es una herramienta de auditoría de red bastante nueva que incluye muchas de las nuevas técnicas de exploración que se usan actualmente. Es necesario revisar la seguridad del sistema con nmap. Es un hecho que otras personas lo harán. Se puede conseguir la herramienta nmap en la dirección http://metalab.unc.edu/pub/Linux/system/network/admin/.

La siguiente salida de ejemplo de nmap informa sobre el estado de todos los puertos TCP y UDP. Como no se usa la versión completa, nmap sólo informa sobre los puertos que están abiertos o que tienen servidores escuchando en ellos. La salida de nmap incluye el nombre del host explorado, la dirección IP, el puerto, el estado abierto o cerrado, el protocolo de transporte en uso en dicho puerto y el nombre del puerto de servicio simbólico del archivo /etc/services. Como sebastion es un host interno, los puertos adicionales de telnet y de X11 están abiertos para acceso LAN interno:

```
> nmap -sT -sU sebastion

WARNING:  -sU is now UDP scan — for TCP FIN scan use -sF

Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Interesting ports on sebastion.firewall.lan (192.168.1.2):
Port       State          Protocol     Service
21         open           tcp          ftp
22         open           tcp          ssh
23         open           tcp          telnet
25         open           tcp          smtp
53         open           udp          domain
53         open           tcp          domain
113        open           tcp          auth
123        open           udp          ntp
6000       open           tcp          X11

nmap run completed — 1 IP address (1 host up) scanned in 3 seconds
```

Depuración de SSH: un ejemplo de la vida real

Al principio, un amigo me recalcó la importancia de usar ssh en vez de telnet. Todavía no había descubierto la dicha y las maravillas de mi biblia de firewalls, *Building Internet Firewalls*, escrito por *D. Brent Chapman* y *Elizabeth D. Zwicky* (O'Reilly & Associates, 1995), ni había descubierto el placer de leer los estándares RFC (Request for Comment, Petición de comentarios) en la dirección ww.ietf.cnri.reston.va.us/rfc.html. Si una regla de ejemplo de firewall no existía en la ayuda de Linux, los HOWTO, era necesario averiguar todo acerca del protocolo por cuenta propia, usando las herramientas de sistema disponibles y habilitando el registro del firewall. El protocolo SSH fue uno de los primeros.

Descargué, compilé e instalé ssh. Configuré las claves de autenticación localmente en uno de los ISP. Las conexiones salientes de ssh hacia el ISP funcionaron sin ningún problema. Las conexiones entrantes de ssh procedentes del ISP funcionaron sin problemas. Todo iba sobre ruedas.

No tardé mucho tiempo en descubrir que las conexiones entrantes ssh procedentes de un empleado no funcionaban. Al intentar utilizar ssh desde el trabajo para conectar con la máquina de casa, me solicitó la contraseña. Fue aceptada. Apareció el titular de inicio de sesión local y una línea de comandos de shell bajo mi cuenta de inicio de sesión. Luego, nada. El teclado no respondía. Por el contrario, si usaba ssh para conectarme al ISP desde el trabajo y luego usaba ssh para iniciar una sesión en el equipo de casa desde la máquina ISP, todo funcionaba como era de esperar. Era posible iniciar una sesión desde el ISP. Se podía iniciar una sesión en el ISP desde el trabajo. Pero no era posible iniciar una sesión directamente desde el trabajo.

La única opción era utilizar ssh en el ISP y, desde allí, utilizar ssh para conectar con la máquina local, y usar las herramientas del sistema, ps y netstat, para ver las diferencias que había entre usar ssh desde el ISP y usar ssh desde el trabajo.

No entendía el protocolo SSH, por lo que no entendía lo que sucedía. En realidad, interpreté de forma errónea todo lo que sucedía, pero ¿que sucedía con las reglas del firewall que yo mismo había creado?

El primer pensamiento fue que fallaba el servidor local sshd. Usando ps -ax, comprobé que había tres copias del servidor sshd en ejecución. Una era el demonio del servidor maestro. Otra era el servidor que se había falsificado para manejar la conexión activa desde el ISP. La tercera era el servidor que se había iniciado para manejar la conexión activa desde el trabajo, la conexión entrante que se había bloqueado. Correcto. Los servidores estaban en ejecución.

El siguiente paso era usar netstat -a -A inet para ver el estado de las conexiones. El servidor sshd maestro escuchaba las nuevas conexiones en el puerto 22, como era de esperar. El servidor que manejaba la conexión en marcha desde el ISP escuchaba el puerto 22, y el cliente ISP remoto usaba un puerto no privilegiado, de nuevo, como era de esperar. El servidor que manejaba la conexión activa desde el trabajo escuchaba el puerto 22, pero el cliente de trabajo remoto usaba el puerto privilegiado, ¡1023! Esto era confuso. El cliente del trabajo usaba un puerto privilegiado y el cliente ISP usaba un puerto no privilegiado.

Luego, ¿qué sucede con SSH? Mirando en el archivo /etc/services, SSH aparenta ser un servicio TCP estándar que usa el puerto 22. Si fuera así, el siguiente conjunto de reglas de E/S para conexiones entrantes funcionaría:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp  \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 22 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

De hecho, estas reglas funcionaron para la conexión ISP. Las reglas eran correctas, pero ssh se bloqueaba desde el trabajo después de iniciar la sesión. A medida que se quitaban, estas dos reglas fueron también las que permitieron llegar desde el trabajo hasta la solicitud de contraseña, el titular de inicio de sesión y la línea de comandos del shell. Estas dos reglas no eran suficientes para abarcar todo el protocolo SSH.

Para volver a revisar lo que veía, habilité el registro para todo el tráfico entrante y saliente relacionado con el puerto 22, tanto el que se aceptaba como el que se denegaba. El registro de paquetes del firewall comprobó la situación. Todos los paquetes salientes procedentes del puerto 22 se aceptaban. Los paquetes salientes procedentes del puerto 22 al puerto no privilegiado se aceptaban. Los paquetes salientes del puerto 22 al puerto privilegiado se aceptaban. Los paquetes entrantes procedentes del puerto privilegiado, 1023, se denegaban. Al final, esto explicaba por qué el inicio de sesión de ssh desde el trabajo devolvió el eco del titular de inicio de sesión y el símbolo del sistema del shell.

En ese momento, las reglas del firewall eran bastante primitivas. El firewall usaba una directiva denegar todo lo que entra de forma predeterminada y una directiva permitir que salga todo de forma predeterminada. Se permitía mucho más de lo que se permite actualmente. Las reglas del protocolo TCP estándar eran suficientes para la conexión desde el ISP. La combinación de las directivas denegar todo lo que entra y permitir todo lo que sale, complicaban el asunto para la conexión desde el trabajo. Entonces el firewall permitía los paquetes salientes del servidor en los puertos privilegiados, pero no permitía los paquetes entrantes del cliente remoto. El servidor era capaz de enviar el titular y el símbolo del sistema del shell. El cliente no pudo responder al servidor.

Como nunca se probó a tener más de una o dos conexiones SSH entrantes, tenía la impresión de que sshd hacía la conexión entre un puerto no privilegiado y el puerto 22 y luego, cuando se conectaba al trabajo, cambiaba a una conexión entrante con la copia falsificada de sí mismo entre el puerto 1022 ó 1023 y el puerto 22. Por supuesto, el servidor no elegía el puerto del cliente. El cliente elegía el puerto, privilegiado o no privilegiado.

Hasta entonces no había leído el código fuente de ssh. Dependiendo de los parámetros de la configuración local, el cliente ssh se inicia en el puerto 1023 y busca un puerto en sentido descendente hasta el puerto 513, asignando la conexión al primer puerto libre que encuentra. Con otros parámetros de configuración, el cliente usa un puerto no privilegiado. El servidor acepta las conexiones activas desde cualquiera de ellos:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp  \
         -s $ANYWHERE 1022:1023 \
         -d $IPADDR 22 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE 1022:1023 -j ACCEPT
```

En este caso, el registro de todos los paquetes dirigidos y procedentes del puerto 22, tanto los aceptados como los denegados, fue suficiente para diagnosticar el problema, al menos en términos de reglas de firewall. Se tardó más tiempo en comprender lo que contabilizaban las reglas.

Resumen

En este capítulo se ha introducido el mecanismo de listado de ipchains, la información disponible del demonio de red y de puerto de Linux mediante netstat, así como unas cuantas herramientas de otros fabricantes que se pueden conseguir para comprobar que las reglas de firewall están instaladas y ordenadas como se espera. También se ha explicado la opción de coincidencia de paquetes de ipchains para comprobar tipos de paquetes concretos con las reglas de firewall instaladas.

En este capítulo se hace especial énfasis en las reglas de firewall y en los puertos que protegen. El siguiente capítulo desplaza el centro de la atención ligeramente para enfatizar qué programas de servidor usan estos puertos, así como el firewall, el sistema y la información de servidor que generan los archivos de registro del sistema.

III

Supervisión y seguridad a nivel de sistema

6

Comprobación de que el sistema funciona como se espera

El Capítulo 5, "Depuración de las reglas del firewall", hace especial énfasis en el uso del programa ipchains como una herramienta de diagnóstico para validar y probar las reglas del firewall y los puertos de servicio que éstas protegen, con el fin de determinar que las reglas están en su sitio y funcionan correctamente. Este capítulo se centra en los programas de servidor que usan estos puertos. Es necesario determinar que los únicos programas y servicios en ejecución son aquellos que se espera encontrar. Después de comprobar los programas y puertos que se usan, se debe determinar lo que indican estos programas y el firewall cuando introducen informes de mensajes de estado y de error en los archivos de registro del sistema. En este capítulo se presentan herramientas de administración adicionales e información de los archivos de registro que puede utilizar el usuario.

Nunca es posible tener una certeza absoluta de que un sistema UNIX se ejecuta correctamente. Sólo se puede tener un grado razonable de confianza basándose en el hecho de que el sistema se ejecuta como se espera, tal y como se intenta. Los sistemas UNIX son demasiado complejos, las cuestiones de la configuración cruzan demasiadas fronteras, como para estar completamente seguro de que todo se ejecuta correctamente. Los archivos de registro ayudan a comprobar que el sistema se ejecuta como se espera e informa de sucesos y estados que se apartan del comportamiento normal.

Cómo comprobar las interfaces de red con ifconfig

La principal función de ifconfig es configurar y activar las interfaces de red. Se ejecuta desde una secuencia de comandos de inicio de red que gestiona el administrador del nivel de ejecución, cuando el sistema se inicia. Posteriormente, ifconfig es útil como herramienta de depuración para informar sobre el estado de las interfaces de red.

Si se usa sólo ifconfig, sin ninguna opción, informa del estado de todas las interfaces de red activas. Con la opción -a, ifconfig informa del estado de todas las interfaces de red, activas o no. Cuando todas las interfaces están activas, las dos opciones de informe producen la misma salida.

Si no se espera que una interfaz esté desactivada, esto es normalmente un signo de que existe un problema de configuración de la red. Revise las configuraciones de la interfaz mediante el panel de control o a través de los programas linuxconf.

La siguiente salida se creó desde una máquina con una única tarjeta de interfaz de red. ifconfig informa del estado de la interfaz física, eth0, y del estado de la interfaz de bucle invertido, lo:

```
> ifconfig

eth0      Link encap:Ethernet  HWaddr 00:A0:CC:40:9B:A8
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:266027 errors:0 dropped:0 overruns:0 frame:0
          TX packets:202290 errors:0 dropped:0 overruns:0 carrier:0
          collisions:17805 txqueuelen:100
          Interrupt:9 Base address:0xec00

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:51997 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51997 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

Cuando existe un problema, lo principal que se querrá saber es si la interfaz está activa. La tercera línea de cada informe de la interfaz indica el estado de la interfaz y su configuración. La interfaz eth0, UP, en este ejemplo está activa.

Observe que ifconfig muestra otra información de estado en la que puede estar interesado en algún momento. Se informa de la dirección Ethernet hardware MAC, Hwaddr, la dirección IP, inet addr, la dirección de difusión, Bcast, y la máscara de red, Mask. La restante información que muestra suele ser menos útil, como el tamaño de la trama máxima predeterminada, MTU, el número de paquetes recibidos, RX, y el número de paquetes transmitidos, TX.

Fragmentación y unidad de transmisión máxima (MTU, *Maximum Transmission Unit*)

Lo mejor es establecer la MTU al valor 1500, que es el valor predeterminado del sistema. La mayoría de las redes actuales son redes Ethernet. La MTU de la Ethernet es de 1500 bytes. Otros protocolos de la capa de enlace (como ATM o Token Ring) tienen tamaños de MTU recomendados más grandes. Establecer la MTU a un valor mayor de 1500 bytes garantiza la fragmentación del paquete en algún momento de la transmisión, cuando el paquete cruza los límites de la red. Las conexiones TCP negocian su MTU cuando la conexión se establece por primera vez.
En la actualidad, el ancho de banda no es tanto un problema de las redes troncales, como lo es la fragmentación. El tiempo de procesamiento necesario para la fragmentación es un factor de rendimiento importante. Se consigue un mejor rendimiento con más paquetes pequeños que con pocos paquetes grandes que deben fragmentarse a lo largo del camino.

Como comprobar la conexión de red con ping

Para revisar la conectividad de red, ping es la herramienta ideal. Si la interfaz de red externa está activa pero no es posible conectarse a un host remoto, ping puede indicar si los paquetes pasan a través de la interfaz. Por supuesto, es necesario habilitar el tráfico ping en las reglas de firewall. Una respuesta negativa de ping no prueba que un sitio remoto no esté activo. El sitio puede que no responda a los mensajes ICMP Echo Request. Una respuesta positiva de ping prueba que los paquetes se transmiten y que el host remoto responde.

Con un solo nombre de host o una dirección IP como argumento, ping envía paquetes de forma indefinida hasta que se mata el proceso. En dicho momento ping muestra las estadísticas de resumen finales:

```
> ping <smtp.my.isp.domain>

PING smtp.my.isp.domain (10.10.22.85): 56 data bytes
64 bytes from 10.10.22.85: icmp_seq=0 ttl=253 time=4.2 ms
64 bytes from 10.10.22.85: icmp_seq=1 ttl=253 time=4.4 ms
64 bytes from 10.10.22.85: icmp_seq=2 ttl=253 time=4.1 ms
64 bytes from 10.10.22.85: icmp_seq=3 ttl=253 time=5.4 ms
64 bytes from 10.10.22.85: icmp_seq=4 ttl=253 time=3.9 ms

> ^C

--- smtp.my.isp.domain ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.9/4.4/5.4 ms
```

Hacer ping a una máquina no significa que los servicios en dicha máquina estén en ejecución o que estén disponibles para su uso. Simplemente significa que un paquete ping atravesó la red y que una respuesta ha encontrado el camino de vuelta. Si un servicio de red en particular no está disponible, o un

host remoto en particular no responde, por lo menos se sabe que la red está activa y funcionando, si algún host en algún sitio responde.

En este ejemplo, se sabe que la máquina del servidor de correo del ISP está funcionando. No se sabe si el programa del servidor de correo está en ejecución. Es decir, si el servicio de correo está bloqueado y se hace un ping a la máquina del servidor de correo del ISP, la salida anterior indicará que el servidor está en línea. Es posible que el propio programa servidor de correo no se esté ejecutando.

Contestación a solicitudes ping

Tenga en cuenta que el host remoto puede no responder a los mensajes ping Echo Request desde un sitio remoto de Internet, incluso aunque el host esté activo. Como ping es una herramienta básica y sencilla, también tiene una larga historia de uso como herramienta de piratería para denegación de servicio. Como tal, los ejemplos de firewall que se incluyen este libro contestan a las peticiones ping sólo desde los servidores del ISP, como cortesía añadida de la necesidad de análisis interno de red que tienen los ISP.

Cómo comprobar los procesos de red con *netstat*

En el Capítulo 5 se hizo uso de netstat -a -p -A inet para comprobar los programas que se estaban ejecutando y escuchando y en qué interfaces de red, sobre qué protocolos de transporte (TCP o UDP) y en qué puertos de servicio. La opción -A inet restringía el informe a los servicios y puertos relacionados con comunicaciones de red remotos. Sin la opción -A, netstat informa sobre los socket de dominios INET y sobre los socket de dominios UNIX.

netstat -a -p -A unix muestra la misma información para los socket de dominios UNIX activos, en particular los socket que usan los servicios locales. El usuario debería ser capaz de explicar todas las entradas del informe:

```
> netstat -a -p

1.  Active UNIX domain sockets (servers and established)
2.  Proto RefCnt Flags   Type   State      I-Node PID/Program name Path

3.  unix  1      [ ]     STREAM CONNECTED 938    588/named     @0000007b
4.  unix  0      [ ACC ] STREAM LISTENING 500119 521/syslogd   /dev/log
5.  unix  1      [ ]     STREAM CONNECTED 864    532/klogd     @00000075
6.  unix  0      [ ACC ] STREAM LISTENING 1083   661/xfs
/tmp/.font-unix/fs-1
7.  unix  0      [ ACC ] STREAM LISTENING 939    588/named
/var/run/ndc
8.  unix  1      [ ]     STREAM CONNECTED 492327 3674/sendmail @00000722
9.  unix  0      [ ]     STREAM CONNECTED 129    1/init        @00000016
10. unix  1      [ ]     STREAM CONNECTED 1349   893/sshd      @0000008a
11. unix  1      [ ]     STREAM CONNECTED 549784 521/syslogd   /dev/log
```

```
12. unix  1      [ ]      STREAM CONNECTED 492328 521/syslogd   /dev/log
13. unix  1      [ ]      STREAM CONNECTED 1350   521/syslogd   /dev/log
14. unix  1      [ ]      STREAM CONNECTED 953    521/syslogd   /dev/log
15. unix  1      [ ]      STREAM CONNECTED 865    521/syslogd   /dev/log
```

La línea 1 identifica el informe como un listado de los socket de dominios UNIX que usan los servidores locales y las conexiones activas.

La línea 2 contiene estos encabezados de columna:

- Proto se refiere al protocolo de transporte sobre el que se ejecuta el servicio, unix en este caso.
- RefCnt es el número de procesos adjuntos a este socket.
- Flags es ACC o está en blanco. ACC indica que un proceso está esperando aceptar una petición de conexión entrante.
- Type es la clase de socket que usa el servicio. STREAM indica una conexión, parecido a una conexión TCP. Un socket DGRAM sería parecido a UDP, un protocolo sin conexión.
- State es el estado de la conexión del socket local, conexión establecida (ESTABLISHED), escuchando para una petición de conexión (LISTEN), así como un número de estados intermedios de establecimiento de la conexión intermedia y de apagado (por ejemplo, SYN SENT, SYN RECV, FIN WAIT, FIN SENT).
- I-Node es el número de inodo del objeto del sistema de archivos que el programa usó para adjuntarse al socket. Un inodo es la forma en que el sistema define e identifica un archivo.
- PID/Program name es el ID del proceso (PID) y el nombre del programa que son los propietarios del socket local.
- Path es la trayectoria del objeto que el programa usó para adjuntarse al socket.

La línea 3 muestra que el servidor de nombres, named, está actualmente conectado a un segundo servidor de nombres interno.

Las líneas 4 y 11-15 muestran que el demonio de registro del sistema, syslogd, está escuchando o conectado a seis socket. Esta máquina en particular está configurada para crear seis archivos de salida de syslog independientes. La salida de netstat no es suficiente para saber el socket que está asociado con cada registro. La configuración de syslog se explicará posteriormente en este capítulo.

La línea 5 muestra que el demonio de registro del núcleo, klogd, está escuchando mensajes desde los búfer de mensajes del núcleo y que están disponibles en el archivo /proc/kmsg. klogd funciona junto con syslogd.

La línea 6 muestra que xfs está escuchando, esperando peticiones locales de fuentes de X Window.

La línea 7 muestra que named está escuchando, esperando búsquedas locales y peticiones del servidor.

La línea 8 muestra que sendmail está conectado para mandar, transmitir y recibir correo.

La línea 9 muestra que init, el padre de todos los procesos UNIX, está escuchando para reiniciar las sesiones de terminal y aceptar los comandos de cambio de nivel de ejecución.

La línea 10 muestra que sshd está conectado para aceptar sesiones ssh locales y para iniciar y activar servidores individuales para cada conexión.

El sistema funciona como se esperaba. init debe estar en ejecución. syslogd y klogd deberían estar en ejecución. Se intenta que los restantes servidores opcionales, named, xfs, sendmail y sshd, estén en ejecución en esta máquina.

Cómo comprobar todos los procesos con ps -ax

ps informa sobre el resto de los procesos. La opción -a selecciona todos los procesos con terminales de control, normalmente los programas de usuario que se ejecutan de forma interactiva en segundo plano. La opción -x selecciona los procesos sin terminales de control, que suelen ser demonios de sistema permanentes que se ejecutan automáticamente en segundo plano. Usando las dos opciones a la vez, informan de todos los procesos UNIX, incluyendo sus identificadores de proceso, terminales de control tty, estado de ejecución, tiempo de sistema usado y el nombre del programa. Si se agrega la opción –u, se crea información adicional orientada al usuario, incluyendo el nombre de inicio de sesión del usuario.

ps -ax lista todos los procesos del sistema UNIX. Igual que con netstat, debe familiarizarse con todos los programas que se ejecutan en el sistema y con el motivo por el que se ejecuta el programa. Con la excepción de unos cuantos demonios especiales del sistema UNIX, particularmente init, kflushd, kpiod, kswapd, mdrecoveryd y mingetty, los demás demonios deben ser servicios que se hayan habilitado explícitamente con el administrador de nivel de ejecución, /etc/inetd.conf o /etc/rc.d/rc.local. Cualquier otro programa que se ejecute debe ser un programa de usuario que se pueda identificar. Lo que es más importante, ps -ax no debe informar sobre ningún proceso que no se espera ver.

A continuación se muestra una salida de ejemplo típica de ps -ax:

```
> ps -ax
1.    PID  TTY    STAT   TIME  COMMAND
2.      1  ?      S      0:03  init
3.      2  ?      SW     0:02  [kflushd]
4.      3  ?      SW     0:00  [kpiod]
5.      4  ?      SW     0:01  [kswapd]
6.      5  ?      SW<    0:00  [mdrecoveryd]
7.    202  ?      S      0:00  /sbin/dhcpcd
    -c /etc/sysconfig/network-scripts/ifdhcp
8.    521  ?      S      0:03  syslogd -m 0
9.    532  ?      S      0:00  klogd
```

```
10.   546  ?      S      0:00  /usr/sbin/atd
11.   560  ?      S      0:01  crond
12.   574  ?      S      0:01  inetd
13.   588  ?      S      0:10  named
14.   638  ?      S      0:02  httpd
15.   661  ?      S      0:01  xfs
16.   893  ?      S      0:07  /usr/local/sbin/sshd
17.   928  tty2   S      0:00  /sbin/mingetty tty2
18.   930  ?      S      0:04  update (bdflush)
19.  1428  tty1   S      0:00  /sbin/mingetty tty1
20.  3674  ?      S      0:00  sendmail: accepting connections on port 25
21. 17531  ?      S      0:00  in.telnetd -n
22. 17532  pts/1  S      0:00  login — bob
23. 17533  pts/1  S      0:00  -ksh
24. 17542  pts/1  R      0:00  ps -ax
```

La línea 1 contiene estos encabezados de columna:

- PID es el Id. único del proceso.
- TTY es el terminal de control del proceso, si existe alguno.
- STAT es el estado de ejecución del proceso. En el listado, ps está en ejecución, Runnable. Los demás procesos están Sleeping (durmiendo), es decir, no están en cola de ejecución porque están esperando algún suceso que manejar o al que responder.
- TIME es la cantidad de tiempo de CPU que ha consumido el proceso.
- COMMAND es el nombre del programa del proceso.

La línea 2 muestra que init está en ejecución, el padre de todos los demás procesos. Siempre está en ejecución. Casi todos los tipos de UNIX se bloquean si init muere.

La línea 3 muestra que kflushd está en ejecución. kflushd no es un proceso, es un subproceso del núcleo que devuelve a disco periódicamente los búfer del sistema de archivos modificados.

La línea 4 muestra que kpiod está en ejecución. kpiod no es un proceso, es un subproceso del núcleo que administra la demanda de paginación.

La línea 5 muestra que kswapd está en ejecución. kswapd no es un proceso. Es un subproceso del núcleo que selecciona páginas de memoria física para paginarlas desde memoria a espacio de paginación en disco, con el fin de liberar memoria para otros procesos.

La línea 6 muestra que mdrecoveryd está en ejecución. Mdrecoveryd no es un proceso. Es un subproceso del controlador del núcleo que administra múltiples dispositivos de disco como si fueran una sola unidad, es decir, RAID.

La línea 7 muestra que dhcpcd está en ejecución, es el cliente DHCP que recibe una asignación de memoria dinámica desde un servidor DHCP.

La línea 8 muestra que syslogd está en ejecución, es quien registra los mensajes del sistema. Recopila los mensajes de los programas del sistema y los escribe en los archivos de registro especificados, en la consola, en los terminales, etc.

La línea 9 muestra que klogd está en ejecución, es quien registra los mensajes del núcleo. Recopila los mensajes desde los búfer de mensajes del núcleo y los escribe en los archivos de registro especificados y en la consola, junto con syslogd.

La línea 10 muestra que atd está en ejecución, es el demonio que planifica los programas de usuario para ejecutarlos en momentos predefinidos. atd es el equivalente en el nivel de usuario de crond.

La línea 11 muestra que crond está en ejecución, es el demonio que ejecuta programas de sistema y administrativos en momentos predefinidos.

La línea 12 muestra que inetd está en ejecución, es el superservidor de servicio de red que escucha las conexiones en nombre de otros servicios, de forma que los demonios de servicio individuales no necesitan estar en ejecución en segundo plano cuando no hay peticiones de servicios activas.

La línea 13 muestra que named está en ejecución, éste es el servidor de nombres DNS.

La línea 14 muestra que httpd está en ejecución, es el servidor de Web Apache. inetd no administra el servidor, por lo que el proceso se ejecuta permanentemente en segundo plano.

La línea 15 muestra que xfs está en ejecución, es el servidor de fuentes de X Window.

La línea 16 muestra que sshd está en ejecución, es el servidor SSH. La administración de SSH no la realiza inetd, por lo que el proceso se ejecuta permanentemente en segundo plano.

La línea 17 muestra que mingetty (tty2) está en ejecución. mingetty se inicia desde el archivo /etc/inittab para escuchar los inicios de sesión en las líneas de terminal. En este caso, tty2 es una consola virtual.

La línea 18 muestra que update (bdflush) está en ejecución. update es el componente del espacio de usuario de kflushd para volver a escribir búfers modificados del sistema de archivos.

La línea 19 muestra que mingetty (tty1) está en ejecución. mingetty se inicia desde el archivo /etc/inittab para escuchar los inicios de sesiones en las líneas de terminal. En este caso, tty1 es la consola física.

La línea 20 muestra que sendmail está en ejecución; es el servidor de correo.

La línea 21 muestra que in.telnetd está en ejecución. inetd inició in.telnetd cuando se hizo un telnet al sistema para generar la salida anterior de ps.

La línea 22 muestra que login está en ejecución. login es la sesión de inicio de sesión creada después de iniciar la sesión en el sistema para generar la salida de ps.

La línea 23 muestra que ksh está en ejecución; es el programa de shell bajo el que se ejecuta la sesión de inicio de sesión.

La línea 24 muestra que ps está en ejecución; es la instancia de ps que creó la salida.

Si aparecen procesos no familiares o inesperados, alguien podría haber tenido acceso a la máquina. El Capítulo 8, "Detección de intrusos e informe de incidentes", explica los pasos que se deben realizar en dicho caso.

Cómo interpretar los registros del sistema

syslogd es el demonio de servicio que registra los sucesos del sistema. El archivo de registro del sistema principal es /var/log/messages. Muchos programas usan los servicios de registro estándares de syslogd. Otros programas, como el servidor Web Apache, mantienen sus propios archivos de registro independientes.

¿Qué se registra y dónde?

Los archivos de registro del sistema se escriben, de forma predeterminada, en el directorio /var/log. En el archivo de configuración de syslog, /etc/syslog.conf, se definen los archivos que se escriben y lo que se escribe en cada uno de ellos. Esto varía según las versiones y distribuciones de UNIX. La distribución de Linux está preconfigurada para escribir mensajes, como mínimo, en el archivo messages. Red Hat 6.0 está preconfigurado para escribir la información de registro del sistema en cuatro archivos independientes: messages, secure, maillog y spooler:

- /var/log/messages es el archivo del sistema que registra todo. En realidad, puede ser el único archivo de registro que se use. Contiene una copia de cualquier mensaje que se escriba en la consola, cualquier mensaje del sistema operativo que se escriba en el búfer del registro interno del núcleo y cualquier mensaje producido por los programas que usan la llamada de sistema syslog(), como named, sendmail y login.
- /var/log/secure contiene informes de inicios de sesión como root, inicios de sesión de usuario e intentos de su a otros usuarios. Aquí también se escriben informes de conexiones desde otros sistemas e inicios de sesión fallidos como root. Se registra todo inicio de sesión.
- /var/log/maillog contiene un registro del tráfico de correo entrante y saliente y el estado del servidor.
- /var/log/spoole no lo utilizarán la mayoría de los sistemas. El archivo contiene mensajes de error procedentes de los demonios de uucp y del servidor de noticias (innd).

Configuración de syslog

No todos los mensajes tienen la misma importancia ni el mismo interés. Aquí es donde aparece el archivo /etc/syslog.conf. Se puede personalizar la salida del registro para que se ajuste a sus necesidades y preferencias. El archivo de configuración /etc/syslog.conf permite confeccionar la salida del registro para que se ajuste a sus necesidades.

Definición de archivos de registro del sistema no predeterminados

Se pueden redireccionar o duplicar los mensajes del sistema en otros archivos para clasificarlos por tema o por importancia.

Los mensajes se clasifican mediante el subsistema que los produce. En las páginas man, estas categorías se llaman utilidades (consulte la Tabla 6.1).

Tabla 6.1. Categorías de utilidades de registro de *syslog*

Utilidad	Categoría del mensaje
auth o security	Seguridad/autorización.
authpriv	Seguridad privada/autorización.
cron	Mensajes del demonio cron.
daemon	Mensajes generados por el demonio del sistema.
ftp	Mensajes del servidor FTP.
kern	Mensajes del núcleo.
lpr	Subsistema de impresión.
mail	Subsistema de correo.
news	Subsistema de noticias de red.
syslog	Mensajes generados por syslogd.
user	Mensajes generados por un programa de usuario.
uucp	Subsistema UUCP.

Dentro de cada categoría de utilidad, los mensajes de registro se dividen en tipos de prioridad. En la Tabla 6.2 se listan las prioridades en orden creciente de importancia.

Tabla 6.2. Prioridades de los mensajes de registro de *syslog*

Prioridad	Tipo de mensaje
debug	Mensajes de depuración.
info	Mensajes que informan de estado.
notice	Condiciones normales pero importantes.
warning o warn	Mensajes de alerta.
err o error	Mensajes de error.
crit	Condiciones críticas.
alert	Se solicita atención inmediata.
emerg o panic	No se puede usar el sistema.

Una entrada en el archivo syslog.conf especifica una utilidad de registro, su prioridad y el lugar donde se escriben los mensajes. No es obvio que la prioridad sea inclusiva. Se toma para resaltar todos los mensajes de dicha prioridad o superiores. Si se especifican mensajes en la prioridad error, por ejemplo, se incluyen todos los mensajes en la prioridad error y en las superiores.

Los registros pueden escribirse en dispositivos, como la consola, en archivos y en máquinas remotas.

Estas dos entradas escriben todos los mensajes del núcleo, tanto en la consola como en el archivo /var/log/messages. Los mensajes pueden duplicarse en múltiples destinos:

```
kern.*                                          /dev/console
kern.*                                          /var/log/messages
```

Esta entrada escribe mensajes de alerta en todas las ubicaciones predeterminadas, incluyendo el archivo /var/log/messages, la consola y todas las sesiones de terminal de usuario:

```
*.emerg                                         *
```

Las dos entradas siguientes escriben información de autenticación relacionada con el privilegio de root y con las conexiones, en el archivo /var/log/secure, e información de autorización de usuario en el archivo /var/log/auth. Con la prioridad definida en el nivel info, los mensajes debug no se escribirán:

```
authpriv.info                                   /var/log/secure
auth.info                                       /var/log/auth
```

Las dos entradas siguientes escriben información del demonio general en el archivo /var/log/daemon, e información del tráfico de correo en el archivo /var/log/maillog:

```
daemon.notice                                   /var/log/daemon
mail.info                                       /var/log/maillog
```

Los mensajes de los demonios de las prioridades debug e info y los mensajes de correo de la prioridad debug no se registran (estas son las preferencias de quien escribe este libro). named y la revisión sistemática del correo producen, de forma regular, mensajes de información sin interés.

La última entrada registra todas las categorías de mensajes de prioridad info, o de mayor prioridad, en el archivo /var/log/messages, con la excepción de auth, authpriv, daemon y mail. En este caso, las últimas cuatro utilidades de mensajes se establecen a none (ninguno), porque los mensajes se dirigen a sus propios archivos de registro dedicados:

```
*.info;auth,authpriv,daemon,mail.none           /var/log/messages
```

Sugerencias sobre los archivos de registro de /var/log

syslogd no crea archivos. Sólo escribe en archivos existentes. Si no existe un archivo de registro, se puede crear con el comando touch y asegurarse luego que pertenece al root. Por motivos de seguridad, los archivos de registro no suelen ser legibles por el usuario normal. El archivo histórico de seguridad, /var/log/secure, en particular, sólo puede leerlo el root (administrador).

Más información sobre la configuración de syslog

Si desea una descripción más completa de las opciones de configuración de syslog y otras configuraciones de ejemplo, consulte las siguientes páginas man: syslog.conf(5) y sysklogd(8).

Mensajes de registro del firewall: ¿qué significan?

Para generar registros del firewall, el núcleo debe estar compilado con el registro de firewall habilitado. La versión 6.0 de Red Hat tiene habilitado el registro del firewall de forma predeterminada. Las versiones anteriores requieren que se vuelva a compilar el núcleo.

Los paquetes que coinciden de forma individual se registran como mensajes kern.info para las reglas de firewall que tengan activada la opción -l. Se informa de la mayoría de los campos del encabezado de un paquete IP cuando un paquete coincide con una regla y está habilitado el registro. Los mensajes registrados del firewall se escriben, de forma predeterminada, en el archivo /var/log/messages y en la consola.

Se pueden duplicar los mensajes registrados del firewall a un archivo diferente creando un nuevo archivo de registro y agregando una línea al archivo /etc/syslog.conf:

```
kern.info                                        /var/log/fwlog
```

Asimismo, todos los mensajes del núcleo se dirigen a la consola y al archivo /var/log/messages. Cuando se inicia, el núcleo no genera muchos mensajes de información, aparte de los mensajes de registro del firewall.

Como ejemplo, esta regla que deniega el acceso al puerto 111 de portmap/sunrpc producirá el siguiente mensaje en el archivo /var/log/messages:

```
ipchains -A input -p udp -i $EXTERNAL_INTERFACE \
        -d $IPADDR 111 -j DENY -l

  (1)    (2)      (3)            (4)          (5)  (6)  (7)    (8)
Jun  9 14:07:01 firewall kernel: Packet log: input DENY eth0 PROTO=17

             (9)     (10)     (11)      (12)
          10.10.22.85:14386 192.168.10.30:111

          (13)   (14)    (15)    (16)    (17)
          L=316 S=0x00 I=14393 F=0x0000 T=52
```

Se numeran los campos del mensaje de registro para explicarlos:

- El campo 1 es la fecha, 9 de Junio.
- El campo 2 es la hora en que se escribió el registro, 14:07:01.
- El campo 3 es el nombre de host del equipo, firewall.
- El campo 4 es la utilidad de registro que genera el mensaje, kernel. La rutina de registro IPFW anexa Packet log al nombre de la utilidad de registro.

- El campo 5 es la cadena de firewall a la que está asociada la regla, input. Las cadenas incorporadas son input, output y forward.
- El campo 6 es la acción que se realiza con este paquete, DENY. Las opciones son ACCEPT, REJECT, DENY, MASQ, REDIRECT y RETURN.
- El campo 7 es la interfaz de red por la que llega o sale el paquete, eth0.
- El campo 8 es el tipo de protocolo de mensaje que contiene el paquete, PROTO=17. Los valores del campo pueden ser 6 (TCP), 17 (UDP), 1 (ICMP/<code>) y PROTO=<número> para otros tipos de protocolos.
- El campo 9 es la dirección origen del paquete, 10.10.22.85.
- El campo 10 es el puerto origen del paquete, 14386.
- El campo 11 es la dirección destino del paquete, 192.168.10.30.
- El campo 12 es el puerto destino del paquete, 111.

Los campos restantes no tienen un interés especial desde la perspectiva del registro:

- El campo 13 es la longitud total del paquete en bytes, L=316, incluyendo tanto el encabezado del paquete como los datos.
- El campo 14 es el campo tipo de servicio (TOS, Type of Service), S=0x00.
- El campo 15 es el identificador del datagrama de paquete, I=14393. El identificador el datagrama es el Id. del paquete o el segmento al que pertenece este fragmento TCP.
- El campo 16 es el desplazamiento del byte del fragmento, F=0x0000. Si este paquete contiene un fragmento TCP, el desplazamiento del fragmento indica a qué parte del segmento reconstruido pertenece este fragmento.
- El campo 17 es el campo tiempo de vida (TTL, Time to live) del paquete, T=52. El tiempo de vida es el máximo número de saltos (es decir, enrutadores visitados) que quedan antes de que el paquete caduque.

A la hora de interpretar el registro, los campos más interesantes son

```
Jun  9 14:07:01 input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:111
```

Esto indica que el paquete denegado es un paquete UDP procedente de la interfaz externa, eth0, desde un puerto no privilegiado en la dirección 10.10.22.85. Iba dirigido al puerto 111 de esta máquina (192.168.10.30), el puerto sunrpc/portmap (este mensaje se puede ver con frecuencia, ya que portmap es uno de los servicios que más se suelen usar).

Puertos que se sondean habitualmente

Si se registran los paquetes que deniega el firewall, se verá que, normalmente, sólo se sondea un pequeño subconjunto de todo el intervalo del puerto 65536 (los sondeos furtivos más nuevos no generarán un mensaje de re-

gistro, incluso aunque el registro esté habilitado para el puerto). A menudo, la persona que realiza el sondeo sólo prueba un único puerto en el que tiene un interés particular.

Clasificaciones de hostilidad de la Tabla 6.3

Las clasificaciones de hostilidad son estimaciones generales. Cualquier sondeo puede estar motivado por una curiosidad inocente. En el mejor de los casos, las clasificaciones son estimaciones subjetivas basadas en una combinación del peligro potencial para el sistema si se permitiese el acceso al puerto, la probabilidad de ser hostil si dicho puerto fuera el único puerto sondeado como un incidente aislado y el uso histórico del puerto como un objetivo potencial. Al final, el usuario debe decidir la importancia del sondeo, basándose en las circunstancias de cada caso. Sin embargo, existe una gran diferencia entre que alguien haga un ping o un traceroute en el sistema o que busque un servidor ftp o un servidor web, y entre que se sondee buscando un demonio abierto de pop, imap o portmap.

De todos los puertos posibles, la Tabla 6.3 lista los puertos que se sondean habitualmente. Los puertos de la Tabla 6.3 están asociados con servicios LAN inherentemente inseguros, que se sabe históricamente que albergan fallos de seguridad y son los objetivos de clases específicas de ataques o forman parte de las típicas herramientas de piratería. Los puertos de la Tabla 6.3 son los puertos de los que más se habla en los documentos del CERT y los puertos que, personalmente, he visto que se sondean con más frecuencia en los últimos años, e incluye muchos de los puertos sobre los que alerta el libro **Firewalls and Internet Security—Repelling the Wily Hacker** (Addison-Wesley), escrito por Cheswick y Bellovin.

Tabla 6.3. Puertos que se sondean habitualmente

Servicio	Puerto	Protocolo	Hostilidad	Explicación
reserved	0	TCP/UDP	Alta	Origen o destino; uso no legítimo.
0-5	TCP	Alta	Firma de sscan	
echo	7	TCP/UDP	Alta	Ataque UDP.
systat	11	TCP	Alta	Información del proceso de usuario (ps).
netstat	15	TCP	Alta	Estado de la red: conexiones abiertas, tablas de enrutamiento, etc.
chargen	19	TCP/UDP	Alta	Ataque UDP.
ftp	21, 20	TCP	Baja-Alta	Servicio FTP.
ssh	22	TCP	Media-Alta	Servicio SSH.
ssh	22	UDP	Baja	Versión antigua de PC Anywhere
telnet	23	TCP	Alta	Servicio telnet.
smtp	25	TCP	Alta	Busca transmisión SPAM o antiguas debilidades.
domain	53	TCP	Alta	Transferencia de zona TCP: usurpamiento DNS.

Tabla 6.3. Puertos que se sondean habitualmente *(continuación)*

Servicio	Puerto	Protocolo	Hostilidad	Explicación
bootps	67	UDP	Baja	Posible error.
tftpd	69	UDP	Media-Alta	Alternativa al FTP inseguro.
finger	79	TCP	Baja	Información de usuario.
link	87	TCP	Alta	Enlace tty; lo suelen utilizar los intrusos.
pop-3	110, 109	TCP	Alta	Uno de los tres puertos que más se explota.
sunrpc	111	TCP/UDP	Alta	El puerto que más se explota.
nntp	119	TCP	Media-Alta	Noticias públicas o transmisión SPAM.
ntp	123	UDP	Baja	Sincronización de la hora de la red; correcto, pero poco educado.
netbios-ns	137	TCP/UDP	Baja-Alta	Inofensivo para UNIX.
netbios-dgm	138	TCP/UDP	Baja-Alta	Inofensivo para UNIX.
netbios-ssn	139	TCP	Baja-Alta	Inofensivo para UNIX.
imap	143	TCP	Alta	Uno de los tres puertos que más se explota.
NeWS	144	TCP	Alta	Sistema de administración de Window.
snmp	161, 162	UDP	Media	Peticiones y administración de la red remota.
xdmcp	177	UDP	Alta	X Display Login Manager (Administrador de inicio de sesión X).
exec	512	TCP	Alta	Sólo intranet.
biff	512	UDP	Alta	Sólo intranet.
login	513	TCP	Alta	Sólo intranet.
who	513	UDP	Alta	Sólo intranet.
shell	514	TCP	Alta	Sólo intranet.
syslog	514	UDP	Alta	Sólo intranet.
printer	515	TCP	Alta	Sólo intranet.
talk	517	UDP	Media	Sólo intranet.
ntalk	518	UDP	Media	Sólo intranet.
route	520	UDP	Alta	Tablas de enrutamiento.
uucp	540	TCP	Media	Servicio UUCP.
mount	635	UDP	Alta	mountd explotado.
socks	1080	TCP	Alta	Transmisión de SPAM; explosión del servidor proxy.
SQL	1114	TCP	Alta	Firma de sscan.
openwin	2000	TCP	Alta	OpenWindows.
NFS	2049	TCP/UDP	Alta	Acceso a archivo remoto.
pcanywhere	5632	UDP	Baja	PC Anywhere.
X11	6000+n	TCP	Alta	Sistema X Window.
NetBus	12345,	TCP	Alta	Inofensivo para UNIX 12346, 20034.
BackOrifice	31337	UDP	Alta	Inofensivo para UNIX.

Tabla 6.3. Puertos que se sondean habitualmente *(continuación)*

Servicio	Puerto	Protocolo	Hostilidad	Explicación
traceroute	33434-	UDP	Baja	traceroute entrante 33523.
ping	8	ICMP	Baja-Alta	ping entrante.
redirect	5	ICMP	Alta	Bomba de redirección.
traceroute	11	ICMP	Baja	Respuesta traceroute saliente.
UNIX OS probe	0	TCP/UDP	Alta	Difusión a la destino. dirección 0.0.0.0.

Ejemplos de exámenes del registro de puertos habituales

A continuación se listan los mensajes registrados que aparecerán con más frecuencia, procedentes de los puertos que se sondean habitualmente. Si está acostumbrado a la salida del registro del firewall, pase a la siguiente sección. Si nunca antes ha visto una entrada del registro del firewall, estas son las entradas que aparecerán con más frecuencia. Se resumen las entradas del registro para evitar que las líneas se salgan de la página:

- 22/UDP—PC Anywhere (versión antigua):
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:2
  ```
- 23/TCP—telnet:
  ```
  input DENY eth0 PROTO=6 10.10.22.85:14386 192.168.10.30:23
  ```
- 25/TCP—smtp:
  ```
  input DENY eth0 PROTO=6 10.10.22.85:14386 192.168.10.30:25
  ```
- 79/TCP—finger:
  ```
  input DENY eth0 PROTO=6 10.10.22.85:14386 192.168.10.30:79
  ```
- 110/TCP—pop-3:
  ```
  input DENY eth0 PROTO=6 10.10.22.85:14386 192.168.10.30:110
  ```
- 111/UDP—sunrpc:
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:111
  ```
- 119/TCP—nntp:
  ```
  input DENY eth0 PROTO=6 10.10.22.85:14386 192.168.10.30:119
  ```
- 123/UDP—ntp:
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:123
  ```
- 143/TCP—imap:
  ```
  input DENY eth0 PROTO=6 10.10.22.85:14386 192.168.10.30:143
  ```
- 161/UDP —snmp:
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:161
  ```
- 520/UDP—route:
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:520
  ```
- 635/UDP—mount:
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:635
  ```

- 1080/TCP—socks:
  ```
  input DENY eth0 PROTO=6 10.10.22.85:14386 192.168.10.30:1080
  ```
- 2049/UDP—nfs:
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:2049
  ```
- 5632/UDP—PC Anywhere:
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:5632
  ```
- 12345/TCP—NetBus:
  ```
  input DENY eth0 PROTO=6 10.10.22.85:14386 192.168.10.30:12345
  ```
- 31337/UDP—BackOrifice:
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:31337
  ```
- 33434:33523/UDP—traceroute:
  ```
  input DENY eth0 PROTO=17 10.10.22.85:14386 192.168.10.30:33434
  ```
- 8/ICMP—ping echo_request:
  ```
  input DENY eth0 PROTO=1 10.10.22.85:8 192.168.10.30:0
  ```

Paquetes de análisis automatizado del registro

Los paquetes de análisis del registro son herramientas que supervisan lo que se escribe en los registros del sistema, informando de entradas anómalas o realizando alguna clase de acción predefinida. Estas herramientas pueden ejecutarse de forma continua en segundo plano, ejecutarse periódicamente por crond, o ejecutarse manualmente. Los paquetes identifican posibles problemas de seguridad y avisan al usuario cuando aparece una entrada dudosa en el registro.

La versión Red Hat de Linux se distribuye con una herramienta de análisis de registros, swatch. También se pueden encontrar herramientas de análisis de otros fabricantes en la Web. Destacan tres paquetes por su utilidad y facilidad de localización: autobuse, logcheck y swatch. Todos se pueden configurar y pueden enviar notificaciones por correo de sucesos no usuales.

autobuse

autobuse examina periódicamente las nuevas entradas del registro para identificar sondeos habituales. Los resultados dudosos se envían por correo electrónico. Grant Taylor es el propietario de los derechos de autor de autobuse desde 1998 y puede encontrarse esta herramienta en la dirección http://www.picante.com/~gtaylor/download/autobuse/.

logcheck

logcheck examina periódicamente las nuevas entradas en el registro buscando violaciones de la seguridad y actividad inusual. Los resultados dudosos se envían por correo electrónico. logcheck es un clon de un paquete parecido procedente del paquete de firewall Gauntlet de TIS. logcheck viene pre-

configurado para reconocer muchos modelos diferentes de entradas de registros (el autor del libro lo usa. ¡Es un gran paquete!). logcheck lo ha escrito Craig H. Rowland (crowland@psionic.com) y puede encontrarse en la dirección http://www.psionic.com.

swatch

swatch es un sencillo programa observador, un filtro y un supervisor del archivo de registro que se configura con facilidad. swatch supervisa los archivos de registros y actúa para realizar de forma selectiva una o más acciones que especifique el usuario basándose en los modelos de coincidencia que aparecen en las entradas del registro. Se puede ejecutar periódicamente con crond o de forma continua en segundo plano como un supervisor del registro en tiempo real. swatch se incluye como parte de la distribución Red Hat.

Resumen

Este capítulo se centra en cómo comprobar que se ejecutan los programas y servicios que ha seleccionado el usuario, así como en comprobar los puertos de servicio en que escuchan estos programas. Algunas de las herramientas útiles son ifconfig, ping, netstat y ps. Los programas de servidor y el núcleo informan de mensajes de estado y error en los archivos de registro del sistema, que se encuentran en /var/log. Se explica brevemente el archivo de configuración del demonio de registro del sistema, /etc/syslog.conf, como introducción para explicar lo que indican las entradas del registro del firewall. Se explican los puertos que se sondean habitualmente. Es decir, los puertos a los que se hará referencia, con más frecuencia, en el registro del firewall. Para finalizar, se describe la supervisión del registro y la herramienta de análisis del registro que se incluye con la distribución de Linux, además de dos herramientas adicionales de otros fabricantes. Las herramientas de supervisión del registro pueden configurarse para que avisen o realicen alguna acción en respuesta a modelos que aparecen en los archivos de registro.

Ahora que se conocen los servicios que se ejecutan, los puertos que usan y lo que indican los mensajes del registro del firewall, el Capítulo 7, "Problemas a nivel de administración del sistema de UNIX", abandona todas estas cuestiones de menor nivel y examina medidas de seguridad de mayor nivel. El firewall de filtrado de paquetes no es una solución completa de seguridad. A nivel de administración del sistema, se puede proteger mejor cada servicio en el nivel de configuración de la aplicación. Aunque el nivel de filtrado de paquetes se centra en el acceso de conexión a puertos específicos, el nivel de aplicación se centra en un host específico y en el control de acceso de usuario a los programas que se ejecutan en dichos puertos.

7

Problemas a nivel de administración del sistema UNIX

Este capítulo abandona el nivel inferior que subyace a las tareas del firewall y se adentra en algunas de las medidas de mayor nivel que se pueden afrontar. Un firewall de filtrado de paquetes no es una solución completa de seguridad. A nivel de administración del sistema, cada servicio puede protegerse fuertemente a nivel de configuración de la aplicación. El nivel de filtrado de paquetes se centra en el acceso de conexión entre las direcciones IP y los puertos de servicio, y el nivel de aplicación se centra en la red, el *host* y el control de acceso de usuario, específicos para los programas individuales que se ejecutan en dichos puertos y en los archivos individuales a los que acceden estos programas.

Autenticación: comprobación de la identidad

El nivel de red IP no ofrece autenticación. La única información de identificación disponible es la dirección origen del encabezado del paquete IP. Como se ha explicado, las direcciones origen pueden falsificarse. La autenticación, es decir, determinar que el origen es quien o lo que dice ser, se controla en niveles superiores. En el modelo de referencia TCP/IP, la autenticación se controla mediante el nivel de aplicación.

Últimamente, la autenticación de usuario UNIX se proporciona mediante el uso de contraseñas secretas que sólo conoce el usuario individual al que

pertenece la cuenta. Los mecanismos LAN derivados que se basan en el host desde el que se conecta el usuario y en bases de datos administradas de forma centralizada, se construyen en la cima del mecanismo de contraseña básico. La autenticación de usuario es uno de los pilares de la seguridad UNIX. Al igual que todos los sistemas que evolucionan, el sistema de autenticación de contraseña de UNIX ha necesitado, en algunas ocasiones, mejoras de seguridad. En esta sección se examinan algunas de las mejoras: contraseñas secundarias y cifrado MD5. Dos de los mecanismos de autenticación LAN más comunes, construidos en la cima del mecanismo de contraseñas, la autenticación rhost y la administración NIS, se explican en términos de lo que son y de lo inseguros y peligrosos que son en un entorno de Internet.

Contraseñas secundarias

De forma predeterminada, UNIX cifra y almacena las contraseñas en el archivo /etc/passwd. El archivo /etc/passwd es públicamente legible (es decir, cualquier cuenta o cualquier programa del sistema puede leer el archivo), ya que contiene información necesaria de la cuenta de usuario para diferentes programas, incluyendo el Id. del usuario, el Id. del grupo del usuario, la contraseña, el nombre de usuario, campos para localización física y número de teléfono, el directorio raíz del usuario y el shell predeterminado, el preferido por el usuario.

Como el archivo es públicamente legible, un hacker puede aplicar métodos extraordinarios para buscar las contraseñas, probando todas las combinaciones de hasta ocho caracteres e intentando hacer coincidir dichas combinaciones cifradas con la contraseña cifrada. Este método suele ser extremadamente caro en un equipo normal. A medida que los equipos han aumentado su potencia, ahora resulta sencillo ejecutar una herramienta para buscar y descifrar las contraseñas como algo rápido y que se realiza en segundo plano.

Las contraseñas secundarias son un intento de evitar esta debilidad, desplazando la contraseña cifrada a un archivo de contraseña secundario creado para este propósito. La información de la cuenta de usuario permanece en el archivo /etc/passwd de lectura pública. La contraseña real se almacena en el archivo /etc/shadow, que es un archivo que sólo puede leer el root.

La versión Linux de Red Hat ofrece, desde hace tiempo, cierto nivel de compatibilidad con las contraseñas secundarias. En la versión 6.0, la compatibilidad con las contraseñas secundarias se integra en las interfaces GUI de instalación y configuración. Se pueden habilitar las contraseñas secundarias simplemente activando la opción en una casilla de verificación. Las versiones anteriores de Linux exigían que el usuario convirtiera manualmente las contraseñas a contraseñas secundarias usando el programa pwconv.

Acceso remoto al archivo de contraseñas

Como /etc/passwd es legible públicamente, los usuarios remotos pueden obtener una copia del archivo de forma no autenticada. Un servidor FTP mal configurado es un

buen punto de acceso. Un sistema de archivos root montado NFS y accesible de forma remota puede ser otro punto de acceso. El acceso remoto a la base de datos NIS es un regalo para el hacker. El acceso con direcciones usurpadas a través del mecanismo de autenticación rhost es otra posibilidad. El acceso clandestino mediante el shell a través del servidor sendmail es otro mecanismo. Cada una de estas tres posibilidades se explica en este capítulo.

Contraseña MD5 de hash

Las contraseñas secundarias son un intento de ayudar a solucionar la debilidad provocada por un archivo de contraseñas legible públicamente. La contraseña MD5 de hash es un intento de ayudar a solucionar la debilidad provocada por la relativa facilidad con que actualmente se pueden averiguar las contraseñas de ocho caracteres.

Las contraseñas MD5 de hash tienen una longitud máxima de 256 caracteres. El resultado del MD5 de hash aplicado a cualquier objeto es un valor de 128 bitss. Se considera que este valor es imposible de reproducir con equipos informáticos, o al menos tan imposible de reproducir actualmente con equipos informáticos como lo era hace diez o quince años el estándar de ocho caracteres, que eran las contraseñas cifradas DES.

La versión 6.0 de Red Hat es compatible con la biblioteca de MD5. La compatibilidad con las contraseñas cifradas MD5 se integra dentro del programa de instalación y de las interfaces GUI de configuración linuxconf y panel de control. Se puede habilitar el cifrado de contraseñas MD5 simplemente activando la opción en una casilla de verificación.

Autenticación rhost de Berkeley: hosts.equiv y .rhosts

La idea subyacente a la autenticación basada en host es que una vez que un usuario inicia una sesión autenticada en un sistema, dicho usuario puede autenticarse en todos los equipos conectados en red de forma local por asociaciones. A nivel de sistema global, dichas asociaciones las define el administrador del sistema, que es quien agrega los nombres de host a un archivo, /etc/hosts.equiv. Los usuarios particulares pueden personalizar su propia lista de asociaciones agregando los nombres de host a su propio archivo, ~/.rhosts.

El método de autenticación rhost se introdujo en la versión 4.2 de BSD. Junto con los comandos remotos rlogin y rsh de Berkeley, cuando el usuario inicia una sesión en un equipo, puede acceder cómodamente a sus cuentas en todos los equipos de la LAN. Lo que se solía hacer en entornos débiles, por el administrador del sistema, así como por el usuario, era lo más cómodo; en vez de listar cada nombre de host individual, se usó un carácter comodín, +, para permitir el acceso entre todas las máquinas en las que el usuario tenía una cuenta. Para la cuenta del root esto era particularmente cómodo. El administrador del sistema tenía cuentas en todas las máquinas.

Obviamente, la autenticación rhost es un servicio LAN, no un servicio de Internet. Las identidades de las máquinas locales conectadas a la red se aseguran de forma implícita basándose en sus nombres de host. En Internet, tanto las direcciones origen como las tablas de las bases de datos pueden falsificarse, lo que explica los avisos que se han mostrado a lo largo del libro para que se deshabiliten los servicios remotos Berkeley en una máquina firewall.

La autenticación rhost resulta de gran utilidad en un entorno LAN en el que los usuarios tienen cuentas en varias máquinas. La clave es que se debe deshabilitar la autenticación rhost en una máquina accesible desde Internet. Las máquinas firewall, en particular, deben tener deshabilitada la autenticación rhost, ya sea eliminando todos los archivos rhost del sistema como deshabilitando el acceso a los comandos r de Berkeley.

Acceso compartido a la autenticación central: el servicio de información de red

El servicio de información de red (NIS, Network Information Service) es un servicio LAN para centralizar la administración del usuario y la autenticación de host. El servicio NIS se incluye en las distribuciones Red Hat de Linux y puede habilitarse en las interfaces GUI de instalación y de configuración, junto con las contraseñas secundarias y el cifrado MD5. Al igual que ocurría con la autenticación rhost y con los comandos remotos de Berkeley, no se debe habilitar el servicio NIS en una máquina firewall.

Un servidor NIS ofrece servicios de búsqueda de host, usuario, contraseña y pertenencia a grupo a los host clientes. Los clientes pueden preguntar por otros clientes, preguntando quién es su servidor NIS. Los clientes pueden hacer peticiones a clientes y servidores de otros dominios. Sin las restricciones que se pueden configurar basadas en la dirección origen IP, el servidor contesta a todos los que preguntan, ya que éstos saben el nombre del dominio del servidor. Los usuarios no pueden ver las contraseñas cifradas. Los usuarios pueden cambiar la información relacionada con su contraseña de forma remota.

NIS es un servicio UDP al que se accede a través del demonio portmap. Al igual que con cualquier otro servicio al que se accede a través de portmap, el servicio permanece abierto incluso si el firewall deniega el acceso al puerto 111 del servicio portmap. Una exploración general de puerto puede localizar el puerto real al que está conectado el servidor NIS.

Desde un punto de vista de la seguridad de red externa, NIS incorpora lo peor de todo: archivos de contraseñas legibles remotamente para la autenticación de usuario y tablas /etc/hosts legibles remotamente para autenticación de host. El filtrado de paquetes no puede proteger el servicio a menos que se deshabilite el acceso a todos los puertos UDP no dedicados.

Autorización: cómo definir los derechos de acceso a identidades

Cuando se comprueba la identidad del usuario, después de que el usuario está autenticado como quien o lo que dice ser, la siguiente tarea es cosa de los recursos del equipo a los que el usuario tiene permiso para acceder. A nivel de sistema global, las tareas de autorización se centran alrededor de quién y qué tiene permiso para acceder a los recursos que son privilegios del root, quién y qué tiene permiso para asumir la identidad de otra cuenta, qué sistemas remotos tienen permiso para acceder a determinados servicios de red locales, y qué servicios tienen permisos de lectura, escritura y ejecución en determinados archivos y directorios del sistema.

Acceso con cuenta root

Por definición, la cuenta root del superusuario tiene autorización para hacer y acceder a todo. Obviamente, el acceso a la cuenta root se tiene que autenticar de forma severa. Por consiguiente, el root no debe tener permiso para iniciar una sesión desde un host remoto, ni otros servicios autenticados, como ftp, deben permitir el acceso root remoto.

Las ubicaciones desde las que el root puede originar un inicio de sesión se controlan en el archivo /etc/securetty. De forma predeterminada, los inicios de sesión del root se restringen al terminal consola físico y a cualquier consola virtual configurada para dicho terminal, los dispositivos tty. No se deben agregar otros terminales, dispositivos de red ttyp, al archivo /etc/securetty.

Cómo limitar el acceso de cuenta a su

El programa su permite a un usuario cambiar de cuenta de usuario, asumiendo el usuario y la identidad de grupo de una cuenta diferente a la suya en el sistema. Incluso aunque su requiera que el usuario suministre la contraseña para la cuenta intercambiada, algunos administradores de sistemas prefieren limitar el número de personas que pueden usar el programa su.

En Linux, el mecanismo para controlar quién puede ejecutar este tipo de programa se controla mediante la pertenencia a grupos. El grupo wheel existe para controlar el acceso al programa su y a otros programas que se consideran como programas de administración del sistema. Esta aproximación por capas para controlar el acceso es parte de una filosofía de seguridad profunda. Al igual que los usuarios deben conocer la contraseña destino para usar su, en principio, los usuarios no pueden usar la mayoría de los programas de administración, porque los usuarios particulares no tienen permiso de escritura para los recursos del sistema implicados. Controlar el permiso de ejecución de estos programas es un nivel extra de protección.

Para restringir el acceso al programa su a unos pocos usuarios selectos, el esquema general a seguir será agregar los usuarios seleccionados al grupo wheel en el archivo /etc/group. Luego se cambiará la pertenencia del grupo del programa su y sus permisos de acceso:

1. Agregue los usuarios seleccionados al grupo wheel en el archivo /etc/group.
2. Cambie el propietario del grupo del programa su al grupo wheel:

   ```
   chgrp wheel /bin/su
   ```

3. Cambie los bits de permiso de acceso del programa su para establecer el Id. del usuario (el propósito del programa su). Asigne al propietario del programa su, root, permisos de lectura, escritura y ejecución. No permita ningún otro acceso por parte de nadie:

   ```
   chmod 4750 /bin/su
   ```

tcp_wrappers

El programa empaquetador de TCP, tcpd, proporciona la capacidad de definir listas de control de acceso, permitiendo o denegando el acceso desde redes, sistemas o usuarios específicos. Como tal, tcp_wrappers permite definir exactamente qué sistemas remotos tienen o no tienen permiso para acceder a cada servicio de red individual que se ofrece desde el sistema. tcpd es un programa empaquetador. Esto significa que inetd está configurado para ejecutar tcpd en vez del programa de servidor específico pedido. tcpd realiza varias comprobaciones de autorización y luego ejecuta él mismo el servicio. Como tcpd es un programa filtro que se ejecuta sólo entre inetd y el programa servidor solicitado, tcpd no es un demonio de sistema que se ejecute continuamente. Nos referimos a todo el mecanismo de autorización como tcp_wrappers. tcp_wrappers es tanto una herramienta de autenticación como una herramienta de autorización.

En la versión Red Hat de Linux, el archivo /etc/inetd.conf está preconfigurado para usar tcp_wrappers sobre los servicios que se pueden aplicar. Los principales servicios que se pueden aplicar son ftp, telnet, acceso a correo remoto pop-3, acceso a correo remoto imap, finger y el servicio in.identd auth.

TCP no puede empaquetar todos los servicios. Por lo general, inetd debe administrar el servicio. Esto significa que inetd debe ejecutar una instancia individual del programa por cada conexión TCP entrante o por cada petición de datagrama UDP. Es posible administrar servidores, como el servidor web Apache y el servidor SSH de otros fabricantes, bajo inetd, pero estos servidores se ejecutan continuamente como demonios. httpd se ejecuta en modo independiente por razones de rendimiento, ya que la conexión inetd puede sobrepasar el tiempo de espera en sistemas más lentos cuando se usa un cifrado más potente. Sin embargo, sshd tiene la opción de aceptar los archivos hosts_access de su propiedad.

tcp_wrappers proporciona niveles de comprobación de autorización mucho más potentes para los servicios basados en TCP que para los servicios basados en UDP.

Para los servicios TCP, si la máquina del cliente ofrece el servicio identd, tcpd realizará búsquedas basándose en el nombre del usuario para registrar el nombre del usuario junto con el nombre del host y el nombre del servicio. Si se invierten las búsquedas por nombres de host, se ofrece cierto grado de protección contra el usurpamiento de la dirección IP origen y del nombre de host DNS. El nombre de host que devuelve el servicio DNS de una búsqueda de dirección IP a nombre de host, debe coincidir con la dirección IP que devuelve el servicio DNS de una búsqueda posterior de nombre de host a dirección. No se permiten las opciones de la conexión de socket enrutados desde el origen.

Para los servicios UDP, sólo está disponible el control de acceso básico basado en el nombre de host o en la dirección, incluyendo las búsquedas inversas del servicio DNS para protegerse contra el usurpamiento de dirección origen y del nombre de host DNS. Sin embargo, el control de acceso no es tan riguroso para los servicios UDP. No se puede configurar inetd para iniciar los servidores UDP con una opción wait, que indica al servidor que espere unos cuantos minutos después de recibir el último datagrama. El propósito es evitar la sobrecarga de reiniciar el servidor si, poco tiempo después, se recibe otra petición. Si llegan nuevas peticiones desde algún host mientras el servidor está esperando, tcp_wrappers no tendrá efecto en los posteriores intercambios cliente-servidor.

Al igual que con las reglas de firewall, la primera regla de tcp_wrappers que coincida gana, y el control de acceso toma el valor predeterminado de aceptar. La lista de permisos siempre se procesa antes que la lista de denegaciones. A continuación se muestra el orden en que se aplican las listas de acceso:

1. Si una petición entrante coincide con una entrada del archivo /etc/hosts.allow, se permite el acceso.
2. Si una petición entrante no coincide con una regla permitir, entonces si coincide con alguna entrada en el archivo /etc/hosts.deny, se deniega el acceso.
3. Si una petición entrante no coincide ni con una regla permitir ni con una regla denegar, se permite el acceso.

A la hora de configurar los archivos hosts.allow y hosts.deny, el objetivo es el mismo que con las reglas del firewall. Se desea una directiva denegar todo de forma predeterminada, con reglas aceptar para las excepciones explícitas que se quieren permitir.

Los campos principales de una lista de control de acceso son un nombre de servidor y una lista de clientes, separados por dos puntos.

Las listas de control de acceso pueden utilizar comodines. Los dos comodines que más se usan son ALL y LOCAL.

ALL se explica por sí mismo. Se usa en el campo servidor, e incluye todos los servicios tcp_wrappered. Si se usa en el campo lista de clientes, se incluyen todos los clientes.

LOCAL se refiere a la interfaz de bucle invertido y a nombres de host sin cualificar, es decir, los host sin un punto en su nombre, o los nombres de host sin un nombre de dominio. Los nombres de host sin cualificar se refieren a las máquinas que comparten el mismo dominio que la maquina local. Tenga en cuenta, sin embargo, que tcpd coincide con el primer nombre del archivo /etc/hosts. LOCAL no coincide con los siguientes campos alias que están disponibles, de forma opcional, en el archivo /etc/hosts. Para las máquinas particulares que no tienen su propio nombre de dominio local, la máquina firewall tendrá un nombre de dominio asignado por el ISP. Las máquinas LAN internas no serán LOCAL. Éstas pertenecen a un dominio privado interno.

PARANOID coincide con cualquier host cuyo nombre no coincida con su dirección. El modo predeterminado de tcpd es que se compile con la comprobación PARANOID activa todo el tiempo. Esto significa que se deniega el acceso, a los host que no coinciden, a los servicios tcp_wrappered, antes de consultar las listas de control de acceso. Si se vuelve a compilar tcpd con PARANOID desactivado, el comodín PARANOID se usará para aplicar una comprobación PARANOID selectiva a servicios individuales.

Las listas de control de acceso pueden coincidir con modelos. Los dos modelos más comunes son un punto seguido de un nombre de dominio y una dirección de red terminada en un punto. Estos dos modelos coinciden con cualquier host en el dominio o red coincidente.

A continuación se muestra un ejemplo del archivo /etc/hosts.allow:

```
1. ALL: LOCAL .internal.lan
2. in.ftpd: amigo@confianza.host.red
3. sshd : 10.30.27.
4. ipop3d: 10.30.27.45 EXCEPT PARANOID
```

La línea 1 permite el acceso a todos los servicios empaquetados de TCP desde la máquina local y desde todos los host de la LAN interna.

La línea 2 permite además el acceso al servidor FTP desde una cuenta de usuario remota específica. La especificación de un nombre de usuario no funcionará si el sitio remoto no ejecuta el servicio IDENT. La especificación de un nombre de usuario no funcionará si la máquina remota es un equipo personal, un PC o un Macintosh.

La línea 3 permite además el acceso al servidor SSH desde cualquier host de la red remota 10.30.27.0.

La línea 4 permite además el acceso al servidor POP desde una dirección IP remota específica. Si se ha vuelto a compilar el tcpd sin la funcionalidad PARANOID habilitada globalmente, la excepción PARANOID habilitará la funcionalidad PARANOID para esta regla de acceso. La línea 4 indica que se permiten las conexiones al servidor POP desde la dirección IP 10.30.27.45, pero sólo cuando se determina que la dirección IP y el nombre del host coin-

ciden, después de una comprobación cruzada con una búsqueda inversa de dirección a nombre.

Este es un archivo ejemplo /etc/hosts.deny:

```
ALL: ALL
```

Deniega el acceso a todos los servicios tcp_wrappered desde todos los clientes. Como las reglas del archivo hosts.deny se aplican después de las reglas del archivo hosts.allow, esta entrada define, en realidad, una directiva denegar todo de forma predeterminada. Las reglas del archivo hosts.allow definen las excepciones a esta directiva.

Si se desea más información sobre los tcp_wrappers, consulte las siguientes páginas man: tcpd(8), hosts_access(5), hosts_options(5), tcpdchk(8), tcpdmatch(8), inetd(8) e inetd.conf(5).

Permisos de archivos y directorios

Junto con la autenticación de usuario y la autorización de servicio general, está el concepto subyacente de quién tiene acceso a los objetos del sistema de archivos. Los permisos de acceso a archivos de UNIX se controlan en los niveles de lectura, escritura y ejecución para el propietario del archivo y la pertenencia a un grupo, y de forma global para cualquier usuario. Quién tiene el acceso de escritura a un archivo dado o a un directorio es la cuestión más evidente. Las cuestiones de acceso se convierten en algo más sutil cuando hablamos de servicios que se ejecutan con privilegios de sistema en nombre del usuario. UNIX tiene la posibilidad de restringir los permisos de acceso de los servidores por sí mismo, al igual que restringir la visión del servidor del sistema de archivos subyacente.

Archivos legibles públicamente

Ningún archivo requiere permiso de escritura pública. Los archivos que requieren permisos de escritura por varias personas, pueden manejarse incluyendo todos los usuarios implicados en un grupo sencillo creado para el propósito de compartir dicho recurso. Los archivos legibles públicamente se pueden encontrar en el sistema ejecutando el programa find con los siguientes argumentos:

```
find / -perm -0002 -fstype ext2 -type f -print
```

Dependiendo de lo que se haya instalado con la versión Linux, se encontrarán unos cuantos archivos legibles públicamente. Los archivos de puntuación de juegos que aparecen en el directorio /var/lib/games, suelen ser legibles públicamente. Este es un defecto en la seguridad de las actuales versiones del Linux de Red Hat. En algunas ocasiones, se encontrarán archivos fuente o archivos de documentación legibles públicamente. Los permisos

de acceso a archivos abiertos en dichos archivos son un fallo en la seguridad por parte de quien se ocupa del mantenimiento.

RPC y tcp_wrappers

Los servicios basados en RPC no los puede iniciar tcpd, pero el demonio portmap de RPC puede acceder a las listas de control de acceso en /etc/hosts.allo y /etc/hosts.deny por su cuenta. El nombre del servicio de lista de acceso es *portmap*, que no tiene nada que ver con el nombre actual del demonio portmap (algunas veces se hace referencia al demonio o se le conoce con el nombre de rpcbind).
Sin embargo, las listas de control de acceso son sólo una solución parcial para los servidores RPC, porque la exploración de puerto podría identificar los puertos abiertos de RPC de cualquier manera. Un hacker podría pasarse por alto tanto el demonio portmap como las listas de control de acceso.
Para obtener más información sobre el uso de hosts.allow y hosts.deny con el demonio portmap, consulte las páginas man, portmap(8).

Directorios legibles públicamente

Muy pocos directorios requieren permiso de escritura para todo el mundo. En principio, el directorio /tmp debería ser el único directorio legible públicamente del sistema. En realidad, existen unos cuantos directorios más con permiso de escritura global para los servicios LAN que están destinados para compartir archivos, como el servicio SAMBA. Es posible encontrar directorios legibles públicamente en el sistema ejecutando el programa find con los siguientes argumentos:

```
find / -perm -0002 -fstype ext2 -type d -print
```

Dependiendo de lo que se haya instalado de la versión de Linux, se verán unos cuantos directorios más legibles públicamente. Con la posible excepción de unos pocos directorios bajo el directorio /var, cualquier otro directorio no debería tener permiso de escritura global.

Programas setuid y setgid

Los programas setuid y setgid son programas que se ejecutan con el Id. de usuario o el Id. de grupo de alguna otra cuenta. El programa accede, o modifica, recursos del sistema como un proceso privilegiado en nombre del usuario que ejecuta el programa setuid.

Por ejemplo, los programas de usuario, como login, passwd y su, necesitan acceder a servicios disponibles para cuentas de sistema privilegiadas que sólo están disponibles para la cuenta root. El programa sendmail necesita escribir en los archivos de buzón de correo de los usuarios particulares. Los buzones de correo suelen tener permiso de escritura y lectura sólo para el usuario al que pertenecen y para el grupo de sistema de correo. Los programas remotos Berkeley rcp, rsh y rlogin necesitan establecer conexiones usando puertos de servicio privilegiados que sólo están accesibles desde la cuenta root.

Los programas setuid tienen una larga historia en cuanto a explosiones de seguridad. Una de las primeras cosas que hacen los hacker que consiguen entrar en un sistema es instalar caballos de Troya setuid binarios, para concederse a sí mismos los privilegios de sistema del root.

Uno de los errores más peligrosos es crear secuencias de comandos de shell de setuid. Una secuencia de comandos de shell es un archivo ejecutable legible por el usuario. Con un poco de constancia, un usuario no privilegiado puede encontrar una forma de modificar una secuencia de comandos de shell del sistema, o duplicar una secuencia de comandos de shell de setuid y luego volver a escribir los comandos. Los permisos de directorio estándar actuales dificultan mucho esta tarea, que era sencilla en los albores del sistema operativo UNIX.

El software que realiza la comprobación de integridad del sistema, que se explica en el Capítulo 8, "Detección de intrusos e informe de incidentes", comprueba regularmente el sistema en busca de programas setuid y setgid inesperados. Se pueden encontrar los programas setuid y setguid en un sistema ejecutando el programa find con los siguientes argumentos:

```
find / \ ( -perm -4000 -o -perm -2000 \ ) -fstype ext2 -type f -print
```

Servicios chroot

Los servicios accesibles públicamente que leen o escriben en una parte del sistema de archivos, pueden tener restringido su acceso al sistema de archivos a un árbol de directorio específico. chroot cambia la visión de un proceso del sistema definiendo un directorio particular como el sistema de archivos root para ese proceso. El proceso no tiene permiso para ver nada por encima de ese punto. La definición de una vista virtual restringida del sistema de archivos es una protección de seguridad adicional que se sitúa por encima de cualquier restricción de acceso específica definida en el servidor.

Los servicios que se ejecutan en entornos chroot pueden necesitar que se dupliquen otras piezas del sistema de archivos que ahora están fuera del entorno chroot. Por ejemplo, ftp usa el programa ls para permitir a los usuarios ver un listado de directorios del directorio actual de ftp. En un entorno chroot, estos programas binarios están fuera del directorio chroot, de forma que dichos programas binarios deben duplicarse dentro de la estructura del directorio chroot. Es posible que los archivos de configuración del sistema, las bibliotecas compartidas y los archivos de registro del servidor necesiten también duplicarse dentro de la estructura del directorio chroot. Si se desea más información sobre el programa chroot, consulte la ayuda en línea de la página man, chroot(1).

Configuración específica del servidor

Una de las mejores formas de protegerse, con o sin un firewall, consiste en ejecutar simplemente los servicios necesarios. Si un servicio no es nece-

sario, desactívelo. Si un servicio concreto es necesario, actívelo. Si un servicio concreto es necesario, se debe prestar atención a los valores de configuración del servidor.

En esta sección se examinan las opciones de configuración de servidores individuales desde la perspectiva de la seguridad.

Cuestiones relacionadas con la configuración de Telnet

No debería ofrecerse el servicio Telnet a sitios remotos, ni debería usarse para acceder a sitios remotos, a menos que no tenga otra opción. SSH es mucho más recomendable que Telnet. telnetd se habilita desde el archivo /etc/inetd.conf. Según se incluye en la versión de Red Hat, telnetd es un servicio tcp_wrappered. Debe restringirse el acceso exterior al servidor telnetd a host externos específicos mediante la configuración del firewall, así como mediante la configuración del archivo /etc/hosts.allow.

Cuando se inicia una sesión telnet, el contenido del archivo /etc/issue.net se muestra en el terminal del cliente antes de enviar la línea de comandos de inicio de sesión. El archivo /etc/issue.net contiene información de identificación del sistema que puede que no se quiera publicar a nadie que sondee el puerto telnet abierto. Los detalles del archivo /etc/issue.net se explicarán posteriormente en este capítulo.

En concreto, en términos del programa telnet, se puede configurar el servidor para que no muestre toda la información del sistema, ignorando el archivo /etc/issue.net y mostrando simplemente una línea de comandos de login. Si se inicia el servidor con la opción -h, se ignora el archivo /etc/issue.net. Se puede configurar el servidor telnet para que haga esto, modificando el archivo /etc/inetd.conf y cambiando la entrada de telnet para que diga:

```
telnet  stream  tcp   nowait  root   /usr/sbin/tcpd  in.telnetd -h
```

Cuestiones relacionadas con la configuración de SSH

SSH es un sustituto de otros fabricantes para programas como telnet y rlogin. SSH no sólo cifra las sesiones, sino que el cifrado empieza con el establecimiento de la conexión inicial, antes de que se solicite al usuario una contraseña. SSH está disponible tanto como un paquete de software gratuito para uso no comercial, como un producto comercial propio. Se puede encontrar información sobre SSH en SSH Communications Security (Seguridad de las comunicaciones SSH) en Finlandia. El sitio web está en la dirección http://www.ssh.fi/. La versión no comercial de SSH se distribuye como código fuente que es necesario compilar en el sistema. El almacén del código fuente oficial se encuentra en la dirección ftp://ftp.cs.hut.fi/pub/ssh. Allí se puede encontrar tanto la versión 1 original, como la nueva versión 2, disponible actualmente.

El demonio sshd se suele iniciar como un proceso en segundo plano independiente, en vez de iniciarse cuando lo necesita inetd. SSH Communications

Security explica que para el cifrado de claves de host mayores de 512 bits (768 bits es el tamaño de la clave de host predeterminada), el intercambio de claves cifradas de host RSA puede sobrepasar el tiempo de respuesta de inetd. Como los servidores están normalmente tcp_wrapped desde el archivo /etc/inetd.conf, SSH se ha escrito para ser compatible con tcp_wrappers.

A continuación se muestra la forma de compilar e instalar tanto el código fuente de la versión 1.2.26 de SSH1 original, como el código fuente de la versión nueva, la versión 2.0.13 SSH2:

```
./configure —with-libwrap
make all
make install
```

Para SSH1, se modifica el archivo de configuración del servidor, /etc/sshd_config, y el archivo de configuración del cliente, /etc/ssh_config, para ajustarlo a las necesidades del sitio particular. Se crean claves RSA para los dominios que contienen host con los que se supone que se van a compartir conexiones:

```
make-ssh-known-hosts <domainname>
```

Luego, para iniciar el demonio sshd de forma automática en tiempo de inicio, se modifica el archivo /etc/rc.d/rc.local y se agrega la siguiente línea:

```
/usr/local/sbin/sshd
```

Para SSH2, se modifica el archivo de configuración del servidor, /etc/ssh2/sshd2_config, y el archivo de configuración del cliente, /etc/ssh2/ssh2_config, para que coincidan con las necesidades del sitio. Las claves RSA se han creado automáticamente durante el proceso de instalación.

Luego, para iniciar el demonio sshd2 de forma automática en tiempo de inicio, se modifica el archivo /etc/rc.d/rc.local y se agrega la siguiente línea:

```
/usr/local/sbin/sshd2
```

Si desea más información sobre el uso de ssh, consulte las páginas man de ssh(1), sshd(8), ssh-keygen(1) y make-ssh-known-hosts(1) o ssh2(1), sshd2(8) y ssh-keygen2(1).

Cuestiones relacionadas con la configuración de SMTP

Los servidores de correo tienen una larga historia en cuanto a explosiones de vulnerabilidad, tanto smtp como sendmail, al igual que los agentes de entrega de correo pop e imap. Un control cuidadoso del acceso remoto a estos servidores puede ser un largo camino para asegurar estos servicios. Se ha realizado un gran esfuerzo para hacer tan seguro como sea posible al actual servidor de correo, sendmail. La versión Linux 6.0 de Red Hat incluye la versión 8.9.3 de sendmail, una de las versiones más recientes de este programa.

Sin embargo, gran parte de los posibles problemas de sendmail residen en su configuración local y en los host remotos que tienen permiso para transmitir el correo a través del servidor local. Afortunadamente para nosotros, la configuración predeterminada de Linux para sendmail es segura. Un sitio con una única máquina no necesitará modificar la configuración de sendmail.

Un sistema que proporcione servicio de correo para una LAN necesitará habilitar la transmisión para los host de la LAN. En otras palabras, el servidor de correo no aceptará el correo saliente procedente de las máquinas locales. La función de transmisión se habilita para los host específicos en uno de los dos archivos siguientes: /etc/mail/access o /etc/mail/relay-domains. Un ejemplo del archivo /etc/mail/access para una pequeña LAN puede contener:

```
localhost.localdomain          RELAY
localhost                      RELAY
windows.private.lan            RELAY
linux2.private.lan             RELAY
macintosh.private.lan          RELAY
```

Los archivos de configuración en /etc/mail son pares de archivos, un archivo ASCII y un archivo de base de datos hash creado a partir del archivo ASCII. Después de modificar uno de los archivos ASCII, debe actualizarse su correspondiente archivo hash. Los siguientes comandos actualizan la tabla hash del archivo /etc/mail/access, el archivo /etc/mail/access.db:

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

Se puede encontrar información adicional sobre la configuración de sendmail en las siguientes páginas man: sendmail(8), aliases(5) y newaliases(1). Se puede encontrar información importante en el directorio /usr/doc/sendmail. La información más actual se puede encontrar en la dirección http://www.sendmail.org/.

Es necesario asegurarse que se dispone de la última versión de sendmail y que se utiliza smrsh, el shell seguro diseñado específicamente para funcionar con sendmail.

smrsh

Los usuarios pueden hacer uso de sendmail para que ejecute comandos en su nombre. El programa vacation es un ejemplo de esto, que genera automáticamente respuestas de correo para el correo entrante. El resultado es que los usuarios pueden ejecutar cualquier programa en el sistema, pero con privilegios del nivel de sistema en vez de con los suyos propios. smrsh es un shell restringido para sendmail. Si se reemplaza el shell /bin/sh con el shell /usr/sbin/smrsh en el archivo de configuración de sendmail, /etc/sendmail.cf, sólo se permite que los usuarios utilicen los programas que puede ejecutar sendmail. smrsh ejecuta sólo los programas que ha instalado el usuario o a los que se han creado vínculos en el directorio /etc/smrsh/. El directorio está va-

cío de forma predeterminada. En concreto, no ponga ninguno de los siguientes shell: sed, awk, perl, ni ningún otro programa de interpretación en el directorio /etc/smrsh/. La mayoría de los sitios pequeños no necesitarán poner nada en el directorio /etc/smrsh/.

Para reemplazar el shell estándar con el shell smrsh, se modifica el archivo /etc/sendmail.cf y se cambia la línea que empieza por:

```
Mlocal,      P=/bin/sh,
```

a

```
Mlocal,      P=/usr/sbin/smrsh,
```

Cómo probar la capacidad de transmisión del servidor de correo

Un programa de otro fabricante, rlytest, prueba la capacidad de transmisión remota de un servidor de correo. Se puede encontrar un servicio público para probar el servidor de correo de forma remota en http://maps.vix.com/tsi/ar-test.html.
El programa rlytest está disponible en http://www.unicom.com/sw/#rlytest.

Cuestiones relacionadas con la configuración de DNS

DNS, cuando se implementa como el servicio dominio de nombres de Berkeley Internet (BIND, Berkeley Internet Name Domain) en Linux, no es un servicio inherentemente inseguro. Además de los posibles problemas que aparecen con cualquier servicio basado en UDP, otros posibles problemas se centran en los aspectos relacionados con TCP del servicio DNS. Las cuestiones de seguridad se centran en los parámetros de configuración que afectan a la relación del servidor con otros servidores, y la información que el servidor puede proporcionar a los clientes. DNS se hace potencialmente inseguro en términos de privacidad y explosiones por denegación de servicio que implican usurpamiento de DNS o corrupción de la caché DNS. La privacidad es una cuestión de qué información se almacena en la base de datos del servidor de nombres y a quién se permite hacer peticiones a la base de datos. Las explosiones por denegación de servicio son una cuestión de a quién se permite copiar la base de datos del servidor de nombres local, desde dónde se copia la información de zona y a quién se permite actualizar la base de datos.

Si desea más información sobre los servicios DNS y BIND, consulte las siguientes páginas man: named(8), resolver(5) y hostname(7); o la documentación oficial de Bind 8, que se encuentra en el directorio /usr/doc/bind-8.2/html/index.html, el libro *DNS & BIND* de Albitz y Liu (O'Reilly) y la DNS-HOWTO.

/etc/resolv.conf

El solucionador es el componente cliente de DNS. En lugar de ser un programa cliente específico, el solucionador es la parte de las bibliotecas de C que se compila en cualquier programa que necesita tener acceso a la red. El código del solucionador es, de hecho, lo que envía la petición DNS a algunos

servidores de nombres. /etc/resolv.conf es el archivo de configuración del solucionador.

Para un sistema sin un named local en ejecución, son importantes dos directivas: domain y nameserver. domain es el nombre del dominio local. Se pueden listar hasta tres directivas de nameserver, cada una apuntando a un servidor de nombres específico. En este caso, se puede usar una tercera directiva, option rotate, para hacer peticiones a servidores de nombre con el estilo de ronda recíproca, en vez de probar primero cada vez el servidor de nombres principal. Un archivo resolv.conf de ejemplo puede contener el nombre del dominio del ISP y punteros a tres de los servidores de nombres del ISP:

```
domain my.isp.net
nameserver 192.168.47.81
nameserver 192.168.60.7
nameserver 192.168.60.8
option rotate
```

Para un sistema que ejecuta un servidor de nombres named local, una directiva nameserver sencilla apunta a la máquina local. Un ejemplo del archivo resolv.conf puede contener:

```
domain my.isp.net
nameserver 127.0.0.1
```

Para un sistema que ejecuta un named local para una LAN, se puede utilizar una tercera directiva, search, en lugar de la directiva domain. La directiva search toma una lista de dominios desde los cuales realizar peticiones a los servidores de nombres. Un ejemplo del archivo resolv.conf puede contener:

```
search my.local.lan my.isp.net
nameserver 127.0.0.1
```

Registros de recurso de los archivos maestro BIND

Si se está ejecutando named, éste puede tener autorización para una zona o parte del espacio del dominio. Un archivo de zona maestra define las características de una zona para la que el servidor de nombres está autorizado. El archivo contiene información de control para la zona y los registros de recurso que describen la traducción de dirección a nombre para los host dentro de la zona. En el caso de un sistema independiente o de un sistema que ejecute named y esté configurado como un servidor de nombres de reenvío, named debería tener autorización para el localhost.

Los archivos de zona se almacenan en el directorio /var/named/. Los archivos pueden tener nombres arbitrarios. El paquete caching-nameserver de Red Hat de Linux proporciona un archivo named.local para la zona localhost obligatoria, que es casi idéntico al siguiente ejemplo. A continuación se muestra un ejemplo de un archivo de zona para localhost, /var/named/na-

med.127.0.0, que todos los servidores de nombres deben tener para servir búsquedas locales por sí mismos:

```
1. 0.0.127.in-addr.arpa.   IN      SOA  localhost.  root.localhost. (
2.                         1           ; serial
3.                         28800       ; refresh
4.                         14400       ; retry
5.                         3600000     ; expire
6.                         604800      ; default_ttl
7.                 )
8.         IN      NS      localhost.
9. 1       IN      PTR     localhost.
```

Las líneas 1 a 7 contienen la información de control de la zona. Las líneas 8 y 9 son registros de recurso:

- La línea 1 comienza un registro de control SOA (Start of Authority, Inicio de autoridad) para la zona. 0.0.127.in-addr.arpa. es el origen de la zona. Como el origen y el dominio son los mismos, debería reemplazarse el origen con el símbolo @. IN indica que los datos de este registro pertenecen a la clase de datos de Internet (al contrario que, por ejemplo, que la clase de datos hesiod). SOA indica que este registro de recurso es un registro Inicio de autoridad. localhost. es el nombre del dominio. root.localhost. indica la dirección de correo electrónico de la persona de contacto que es el responsable de la información de zona. Los paréntesis abiertos indican el principio de un registro multilínea.
- La línea 2 es un número de serie. Si se ofrece servicio a servidores de nombres secundarios, un cambio en el número de serie indicará que los datos de la zona han cambiado y que éstos necesitarán actualizar sus copias locales de la base de datos de la zona.
- La línea 3 es la frecuencia de actualización, en segundos, que indica la frecuencia con la que los servidores secundarios deben comprobar el número de serie de los datos de la zona. En este caso, todos los servidores secundarios comprobarán el número de serie cada ocho horas.
- La línea 4 es la frecuencia de reintento, en segundos, que indica el tiempo que debe esperar un servidor secundario si falló a la hora de intentar contactar con el servidor primario durante el tiempo de actualización. En este caso, un servidor secundario seguirá intentando contactar con el servidor cada cuatro horas.
- La línea 5 es el tiempo de caducidad, en segundos, que indica el momento en que un servidor secundario eliminará la información de la zona de la caché si falla a la hora de contactar el servidor primario durante este tiempo. En este caso, el servidor secundario caduca los datos de la zona si no fue capaz de contactar con el servidor primario durante los últimos 41 días.
- La línea 6 es el intervalo de tiempo de vida, en segundos, que indica el tiempo que los servidores remotos pueden mantener en la caché la

información que se devuelve como respuesta a una petición. Estas peticiones no serán cualquier petición al localhost procedentes de servidores remotos, pero si lo fueran, el intervalo de tiempo de vida es, en este caso, de una semana.

- La línea 7 contiene el paréntesis cerrado del registro SOA multilínea.
- La línea 8 es un registro de recurso del nombre de servidor (NS), que indica que el localhost es el servidor de nombres para este dominio. Para los registros siguientes al registro SOA, el origen, @, 0.0.127.in-addr.arpa. se considera como el valor del primer campo.
- La línea 9 es registro de recurso puntero (PTR), que indica que la traducción de dirección a nombre para la dirección 127.0.0.1 es el nombre de host localhost. El 1 del principio del registro indica que este registro define la traducción de dirección a nombre para la dirección 127.0.0.1. Como el origen se expresa en el antiguo estilo de dominio IP de la red arpanet, empezando con la dirección del dominio IP en formato inverso de tuplas separadas por puntos, el 1 del principio es una abreviatura de 1.0.0.127.in-addr.arpa.

Actualización de named.boot a named.conf

Las versiones anteriores de BIND usaban /etc/named.boot como archivo de configuración de servidor. Una secuencia de comandos de Perl, /usr/doc/bind-8.2/named-bootconf/Grot/named-bootconf.pl convertirá un archivo existente named.boot al nuevo formato:

```
cd /usr/doc/bind-8.2/named-bootconf/Grot
perl ./named-bootconf.pl < /etc/named.boot > /tmp/named.conf
```

/etc/named.conf

named es el componente de servidor DNS. El servidor encuentra la información solicitada en su caché local o envía la petición DNS a algún otro servidor de nombres. /etc/named.conf es el archivo de configuración del demonio de named.

El nombre del archivo /etc/named.conf y su formato son nuevos en BIND 8.2, que es la versión de BIND que se incluye con Red Hat Linux 6.0.

named.conf se puede configurar de muchas formas diferentes. Consulte al archivo /usr/doc/bind-8.2/html/config.html para ver una lista completa y las descripciones. Aquí sólo se describirán las sentencias que se usan en las siguientes secciones para configurar servidores específicos.

Configuración de un servidor de nombres de reenvío local que sólo haga uso de la caché

En el Capítulo 3, "Creación e instalación de un firewall", se describe una configuración de servidor DNS en la que la máquina firewall alberga un servidor de nombres de reenvío para uso local. El servidor no ofrece el servicio

DNS a host remotos en Internet. El servidor de reenvío simplemente introduce en la caché la información de las búsquedas de forma local después de reenviar inicialmente las peticiones no resueltas a uno de los servidores de nombres del ISP. Esta sección muestra ejemplos del archivo /etc/resolv.conf, la base de datos /var/named/ y los archivos de configuración /etc/named.conf para un servidor de nombres de reenvío local.

Configuración del servidor de nombres de reenvío local

Los clientes DNS locales apuntan al host local como el servidor de nombres al que realizar la petición. El archivo /etc/resolv.conf contiene:

```
domain my.isp.net
nameserver 127.0.0.1
```

El archivo de la base de datos de la zona del localhost, /var/named/named.127.0.0, es el único archivo necesario de información de zona.

El archivo de configuración del servidor named local, /etc/named.conf, contiene lo siguiente:

```
 1. options {
 2.         directory "/var/named";
 3.         forward only;
 4. //      forward first;
 5.         forwarders {
 6.                  my.name.server.1;
 7.                  my.name.server.2;
 8.                  my.name.server.3;
 9.          } ;

10.          query-source address * port 53;

11.          allow-query {
12.                 127.0.0.1;
13. //              192.168.1/24;
14.          } ;
15.          listen-on port 53 {
16.                 127.0.0.1;
17. //              192.168.1.1;
18.          } ;
19. } ;

20. zone "0.0.127.in-addr.arpa" {
21.          type master;
22.          notify no;
23.          file "named.127.0.0";
24. } ;

25. zone "." {
26.          type hint;
27.          file "root.cache";
28. } ;
```

El archivo /etc/name.conf contiene dos clases de registros en este ejemplo: el registro options (opciones) y el registro zone (zona). Se pueden definir las opciones globales del servidor en el registro options y los valores de opción individuales en los registros zone individuales para que se apliquen sólo a dicha zona. Las líneas 1 a 19 definen un registro options:

- La línea 1 declara que el tipo de registro es options y abre el registro multilínea con una llave.
- La línea 2 define el directorio de trabajo del servidor de nombres, /var/named/, en el que se guardan los archivos maestros de la zona.
- La línea 3 indica al servidor que funcione como un servidor de nombres forward only (sólo de reenvío). Es decir, el servidor dirigirá las peticiones no resueltas sólo a los host listados en el registro forwarders.
- La línea 4 es una forma alternativa y comentada de la opción forward. Indica al servidor que funcione como un servidor de nombres forward first (reenvío primero). Es decir, el servidor dirigirá en primer lugar las peticiones no resueltas a los host listados en el registro forwarders. Si estos host no son capaces de resolver la petición, o no responden, el servidor intentará resolver la petición por sí mismo, funcionando con un servidor de nombres normal.
- La línea 5 abre un registro forwarders multilínea, indicado por una llave izquierda. El registro forwarders contiene una lista de servidores a los que reenviar las peticiones.
- Las líneas 6 a 8 listan los servidores de nombre individuales a los que reenviar las peticiones. Se pueden listar hasta tres servidores.
- La línea 9 cierra el registro forwarders, indicado por una llave derecha.
- La línea 10 es necesaria cuando un firewall permanece entre el servidor local e Internet. La línea define que el puerto origen del servidor es el puerto 53 cuando envía peticiones de búsqueda de igual a igual a otros servidores. Es decir, el servidor usa el puerto UDP 53 tanto como puerto origen como puerto destino, para peticiones servidor a servidor.
- La línea 11 abre un registro allow-query multilínea, indicado por una llave izquierda. El registro allow-query contiene una lista de direcciones IP de red de las que se aceptan peticiones.
- La línea 12 indica al servidor que acepte peticiones procedentes del localhost.
- La línea 13 es una red adicional desde la que aceptar peticiones. Si una LAN está conectada a una interfaz de red interna, puede que se quieran aceptar peticiones procedentes de los host de la LAN.
- La línea 14 cierra el registro allow-query, indicado por una llave derecha.
- La línea 15 abre un registro listen-on multilínea, indicado por una llave izquierda. El registro listen-on define el puerto en el que el servidor escucha esperando peticiones procedentes de clientes. El registro contiene una lista de interfaces de red locales en las que escuchar.
- La línea 16 indica al servidor que escuche las peticiones que llegan a la interfaz loopback (de bucle invertido).

- La línea 17 es una interfaz de red adicional en la que escuchar, en espera de peticiones. Si existe una LAN conectada a una interfaz de red interna, se debe especificar la dirección de la dirección IP de la interfaz interna.
- La línea 18 cierra el registro listen-on, indicado por una llave derecha.
- La línea 19 cierra el registro options, indicado por una llave derecha.
- La línea 20 abre un registro zone multilínea para la red de bucle invertido. Los nombres de dominio de la zona se especifican como dominios de ARPANET, como la dirección de red loopback, 127.0.0, se especifica en orden inverso de tuplas separada por puntos, como el dominio ARPANET 0.0.127.in-addr.arpa.
- La línea 21 declara los datos de la zona que se describen en este registro como la copia maestra. Este servidor de nombres es el servidor autorizado para esta zona.
- La línea 22 indica que no es necesario notificar a otros servidores si cambian los datos de zona.
- La línea 23 proporciona el nombre del archivo de la base de datos de la zona. Como named.127.0.0 es un nombre de trayectoria relativo, se supone que puede ubicarse de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 24 cierra el registro zone, indicado por una llave derecha.
- La línea 25 abre un registro zone multilínea para la caché de dominio del root, ofreciendo la ubicación de los servidores de dominio del root de Internet. Un servidor de nombres forward only no necesita la zona root, ya que todas las peticiones se reenvían al host específico declarado en el registro forwarders. Un servidor de nombres forward first, al igual que un servidor de nombres normal, necesita este registro zone como una sugerencia de donde empezar a buscar servidores de nombres autorizados.
- La línea 26 declara los datos de la zona que describe este registro como una sugerencia. Es decir, es sólo un sitio donde empezar.
- La línea 27 proporciona el nombre del archivo de la base de datos de la zona. Como root.cache es un nombre de trayectoria relativo, se supone que se ubica de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 28 cierra el registro zone, indicado por una llave derecha.

Convenciones de nombre y orígenes autorizados para root.cache

Red Hat distribuye una copia del archivo root de caché del servidor de nombres como /var/named/named.ca, Ese archivo se ha renombrado aquí como root.cache, el nombre que se usa en la DNS-HOWTO. También es posible descargar copias desde ftp.rs.internic.net como /domain/named.root.

Configuración DNS del cliente LAN

Los clientes de una LAN interna pequeña no necesitan ejecutar sus propios servidores de nombres. Por el contrario, pueden simplemente apuntar

al host del servidor de nombres local como el servidor de nombres donde realizar peticiones. El archivo /etc/resolv.conf contiene:

```
domain my.private.lan
nameserver 192.168.1.1
```

Esto es todo lo que necesitan las máquinas cliente para realizar peticiones de búsqueda DNS.

Si es necesario ejecutar servidores de nombres en máquinas LAN, también es posible realizar peticiones de reenvío locales sólo al servidor de nombres maestro, peticiones de reenvío locales a los servidores de nombres del ISP (igual que hace el maestro) o peticiones de reenvío a los demás servidores de nombres locales.

La siguiente sección describe otra posibilidad más de configuración cuando se ejecuta un servidor de nombres local en una máquina interna. En este caso, el servidor de nombres público será un servidor tonto que dice ser el autorizado. El servidor interno será realmente el servidor autorizado para la LAN privada interna, invisible para Internet.

Configuración de un conjunto de servidores de nombres clásicos privados y públicos

El Capítulo 4, "Redes de perímetro, firewalls múltiples y problemas con la LAN", describe una configuración de servidor DNS en la que la máquina firewall alberga un servidor DNS público y una máquina interna alberga un servidor privado, local. El servidor público dice ser el autorizado para el dominio, pero, de hecho, no sabe nada sobre las máquinas LAN. Su verdadero propósito es servir como el conducto firewall para los servidores DNS externos y proporcionar un mecanismo para ocultar la información del DNS local. El servidor privado es, en realidad, el autorizado para el dominio LAN privado y realiza servicios de búsqueda local para máquinas de la LAN.

Esta sección muestra ejemplos de los archivos de configuración /etc/resolv.conf, la base de datos /var/named/ y /etc/named.conf tanto para los servidores de nombres públicos como para los privados.

Configuración del servidor de nombres público

El servidor de nombres público dice ser el autorizado para el dominio local, cuando, de hecho, tiene poca o ninguna información sobre la LAN interna. El hecho de que el servidor diga ser el autorizado no tiene sentido en una configuración particular en la que la máquina firewall está conectada a una red de clientes del ISP. Existen pocas razones para configurar el servidor de forma que conteste las peticiones procedentes de host remotos. Una configuración de pequeña empresa con múltiples direcciones IP es probable que tenga razones para ofrecer cierta información local a los host remotos. Como ejemplo, se puede suponer que el servidor público acepta peticiones remotas.

Como el servidor público no sabe nada sobre la LAN, los clientes DNS locales de la máquina no usan el servidor de nombres que se ejecuta en su propia máquina. Por el contrario, los clientes locales usan el servidor de nombres privado que se ejecuta en una máquina interna. Como ejemplo, la máquina interna es el firewall de contención que se describió en el Capítulo 4. El archivo /etc/resolv.conf contiene:

```
search my.local.lan my.isp.net
nameserver 192.168.11.2
```

El servidor de nombres siempre necesita el archivo de la base de datos de la zona del localhost,

```
/var/named/named.127.0.0.
```

Como ejemplo, supongamos que el sitio pertenece al bloque de direcciones entre la red 192.168.10.0 y la red 192.168.11.0, que la dirección IP externa de la máquina firewall es 192.168.10.30, y que la red DMZ pertenece al espacio de direcciones 192.168.11.0.

Además del archivo local de host, se almacenan dos archivos más de zona en el directorio /var/named/. El primero es el archivo de traducción de dirección a nombre del servidor público, named.public:

```
1. 10.168.192.in-addr.arpa.  IN  SOA  my.domain.com.  postmaster.my.domain.com. (
2.                               1999072701     ; Serial
3.                               28800          ; Refresh after 8 hours
4.                               14400          ; Retry 4 hours
5.                               3600000        ; Expire after 41 days
6.                               86400 )        ; Minimum

7.          IN      NS       bastion.my.domain.com.
8.          IN      MX       10 bastion.my.domain.com.

9. 30       IN      PTR      bastion.my.domain.com.
```

Las líneas 1 a 6 contienen la información de control de la zona. Las líneas 7, 8 y 9 son registros de recurso:

- La línea 1 comienza un registro de control SOA para la zona. 10.168.192.in-addr.arpa. es el origen de la zona. IN indica que los datos de este registro pertenecen a la clase de datos de Internet. SOA indica que este registro de recurso es el registro inicio de autoridad. my.domain.com. es el nombre del dominio. postmaster.my.domain.com. indica la dirección de correo electrónico de la persona de contacto responsable de la información de la zona. Los paréntesis abiertos indican el principio de un registro multilínea.
- La línea 2 es un número de serie. Si se ofrece servicio a servidores de nombres secundarios, un cambio en el número de serie indicará que los datos de la zona han cambiado y que necesitarán actualizar sus copias

locales de la base de datos de la zona. En este caso, el número de serie se expresa en la convención común de año, mes, día y número de veces que la base de datos fue cambiada en dicha fecha: YYYYMMDDNN.

- La línea 3 es la frecuencia de actualización, en segundos, que indica la frecuencia con la que los servidores secundarios deben comprobar el número de serie de los datos de la zona. En este caso, todos los servidores secundarios comprobarán el número de serie cada ocho horas.
- La línea 4 es la frecuencia de reintento, en segundos, que indica el tiempo que debe esperar un servidor secundario si falló a la hora de intentar contactar con el servidor primario durante el tiempo de actualización. En este caso, un servidor secundario seguirá intentando contactar con el servidor cada cuatro horas.
- La línea 5 es el tiempo de caducidad, en segundos, que indica el momento en que un servidor secundario eliminará la información de la zona de la caché si falla a la hora de contactar el servidor primario durante este tiempo. En este caso, el servidor secundario caduca los datos de la zona si no fue capaz de contactar con el servidor primario durante los últimos Tiempo de vida (TTL, time-to-live) segundos. En este caso, el servidor secundario caduca los datos de la zona si no fue capaz de contactar con el servidor primario durante los últimos 41 días.
- La línea 6 es el intervalo de tiempo de vida, en segundos, que indica el tiempo que los servidores remotos pueden mantener en la caché la información que se devuelve como respuesta a una petición. El intervalo de tiempo de vida para la información que se proporciona a las peticiones remotas es, en este caso, de una semana.
- La línea 7 es un registro de recurso NS, servidor de nombres, que indica que bastion.my.domain.com., la máquina firewall pública, es el servidor de nombres para este dominio.
- La línea 8 es un registro de recurso MX, intercambiador de correo, que indica que bastion.my.domain.com. es el host de correo o el reenviador de correo para este dominio. El valor 10, el intervalo de preferencia del intercambiador, no tiene sentido en este ejemplo. Para los sitios con múltiples intercambiadores de correo, cada host puede recibir un intervalo de prioridad preferida entre 0 y 65535. Se intenta transmitir el correo a cada host de uno en uno, desde el intervalo inferior al superior, hasta que el envío finaliza con éxito.
- La línea 9 es un registro de recurso PTR, puntero, que indica la traducción de dirección a nombre para el nombre de host bastion.my.domain.com.. El 30 que aparece al principio en el registro indica que este registro define la traducción de dirección a nombre para la dirección 192.168.10.30.

Es necesario un archivo de zona secundario para las búsquedas inversas de nombre a dirección. El archivo es /var/named/named.public.reverse. Las primeras ocho líneas son idénticas a las del archivo de zona de dirección a nom-

bre, incluyendo el registro SOA, el registro NS y el registro MX. El registro PTR se reemplaza con un nuevo tipo de registro, el registro A, de address (dirección):

```
1. 10.168.192.in-addr.arpa.   IN  SOA   my.domain.com.
    postmaster.my.domain.com. (
2.                                  1999072701     ; Serial
3.                                       28800     ; Refresh after 8 hours
4.                                       14400     ; Retry 4 hours
5.                                     3600000     ; Expire after 41 days
6.                                       86400 )   ; Minimum

7.          IN       NS       bastion.my.domain.com.
8.          IN       MX       10 bastion.my.domain.com.

9. bastion.my.domain.com.   IN     A      192.168.10.30
```

La línea 9 es un registro de recurso A, dirección, que indica que la traducción de nombre a dirección para el nombres de host bastion.my.domain.com. es la dirección 192.168.10.30.

El archivo de configuración del servidor named local, /etc/named.conf contiene las siguientes líneas:

```
 1. options {
 2.         directory "/var/named";
 3.         forward first;
 4.         forwarders {
 5.                  my.name.server.1;
 6.                  my.name.server.2;
 7.                  my.name.server.3;
 8.          } ;
 9.          query-source address * port 53;
10.          allow-query {
11.                 ! 127/8;
12.                 192.168.11.2;
13.                 ! 192.168.11/24;
14.                 *;
15.          } ;
16.          allow-transfer {  ! *; } ;
17.          allow-update {  ! *; } ;
18.          listen-on port 53 {
19.                 192.168.10.30;
20.                 192.168.11.1;
21.          } ;
22. } ;
23. zone "0.0.127.in-addr.arpa" {
24.          type master;
25.          notify no;
26.          file "named.127.0.0";
27. } ;
```

```
28. zone "my.domain.com" {
29.          type master;
30.          notify no;
31.          file "named.public.reverse";
32. } ;

33. zone "10.168.192.in-addr.arpa" {
34.          type master;
35.          notify no;
36.          file "named.public";
37. } ;

38. zone "." {
39.          type hint;
40.          file "root.cache";
41. } ;
```

El archivo /etc/name.conf contiene dos clases de registros de este ejemplo: el registro options (opciones) y el registro zone (zona). Se pueden definir las opciones de forma global para el servidor en el registro options, y los valores de opción individuales se pueden definir en los registros zone individuales que se aplican sólo a dicha zona. Las líneas 1 a 22 definen el registro options:

- La línea 1 declara el tipo de registro como un registro options y abre un registro multilínea con una llave izquierda.
- La línea 2 define el directorio de trabajo del servidor de nombres, /var/named/, donde se guardan los archivos maestro de zona.
- La línea 3 indica al servidor que funcione como un servidor de nombres forward only (sólo de reenvío). Es decir, el servidor dirigirá las peticiones no resueltas sólo a los host listados en el registro forwarders. Si estos host son incapaces de resolver la petición, o no responden, el servidor intentará resolver la petición por sí mismo, funcionando como un servidor de nombres normal.
- La línea 4 abre un registro forwarders multilínea, indicado por la llave izquierda. El registro forwarders contiene una lista de servidores a los que reenviar las peticiones.
- Las líneas 5 a 7 listan los servidores de nombres individuales a los que reenviar las peticiones. Se pueden listar hasta tres servidores.
- La línea 8 cierra el registro forwarders, indicado por la llave derecha.
- La línea 9 es necesaria cuando se mantiene un firewall entre el servidor local e Internet. Esta línea define el puerto origen del servidor para que sea el puerto 53 cuando este envía peticiones de búsqueda de igual a igual a otros servidores. Es decir, el servidor usa el puerto UDP 53 tanto como puerto origen como puerto destino, para peticiones servidor a servidor.
- La línea 10 abre un registro allow-query multilínea, indicado por una llave izquierda. El registro allow-query contiene una lista de direcciones IP de red de las que se aceptan y deniegan peticiones. El orden de

las direcciones es importante. La lista de direcciones se revisa para ver si coincide con el orden en que se define la lista. La primera regla que coincide gana.

- La línea 11 indica al servidor que deniegue las peticiones procedentes desde el localhost. Las peticiones locales se envían al servidor de nombres interno privado.
- La línea 12 indica al servidor que acepte peticiones desde el servidor de nombres interno en la dirección 192.168.11.2. Si el servidor público no tiene la información en la caché local, éste reenviará la petición a los servidores de nombre remotos.
- La línea 13 indica al servidor que deniegue las peticiones procedentes de cualquier otro host de la LAN interna. Las peticiones internas se dirigen al servidor de nombres interno privado.
- La línea 14 indica al servidor que acepte las peticiones procedentes de cualquier lugar. Como la regla de aceptación del comodín se dirige a las direcciones denegadas, se aceptan las peticiones entrantes procedentes de cualquier dirección diferente de aquellas de las listas denegadas.
- La línea 15 cierra el registro allow-query, indicado por una llave derecha.
- La línea 16 contiene un registro allow-transfer de una sola línea. El registro allow-transfer usa el operador de negación, !, para denegar las peticiones de transferencia de zona procedentes de cualquier lugar. De forma predeterminada, cualquier lugar tiene permiso para las transferencias de zona.
- La línea 17 contiene un registro allow-update de una sola línea. El registro allow-update usa el operador de negación , !, para denegar las instrucciones de actualización de base de datos de zona procedentes de cualquier lugar. De forma predeterminada, se deniegan las actualizaciones de zona. Si se incluye la opción de forma explícita, proporciona una forma de documentar la línea, así como una regla de copia de seguridad en caso de error.
- La línea 18 abre un registro listen-on multilínea, indicado por una llave izquierda. El registro listen-on define el puerto en el que el servidor escucha esperando peticiones procedentes de clientes. El registro contiene una lista de interfaces de red locales en las que escuchar.
- La línea 19 indica al servidor que escuche las peticiones que llegan a la interfaz externa de Internet, 192.168.10.30.
- La línea 20 indica al servidor que escuche las peticiones que llegan a la interfaz LAN interna, 192.168.11.1.
- La línea 21 cierra el registro listen-on, indicado por una llave derecha.
- La línea 22 cierra el registro options, indicado por una llave derecha.
- La línea 23 abre un registro zone multilínea para la red de bucle invertido. Los nombres de dominio de zona se especifican como dominios ARPANET, de forma que la dirección de red de loopback, 127.0.0, se especifica en orden inverso de tuplas separadas por puntos, como un dominio ARPANET 0.0.127.in-addr.arpa.

- La línea 24 declara los datos de zona que describe este registro, para que sean la copia maestra. Este servidor de nombres es el servidor autorizado para esta zona.
- La línea 25 indica que no se notificará a ningún otro servidor si se cambian los datos de la zona.
- La línea 26 proporciona el nombre del archivo de la base de datos de la zona. Como named.127.0.0 es un nombre de trayectoria relativo, se supone que se localiza de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 27 cierra el registro zone, indicado por una llave derecha.
- La línea 28 abre un registro zone multilínea para el dominio público, my.domain.com. La información de zona de este fichero se usa para búsquedas inversas de nombre a dirección.
- La línea 29 declara los datos de zona descritos por este registro como la copia maestra. Este servidor de nombres es el servidor autorizado para esta zona.
- La línea 30 indica que no se notificará a ningún otro servidor si los datos de la zona cambian.
- La línea 31 proporciona el nombre del archivo de la base de datos de zona. Como named.public.reverse es un nombre de trayectoria relativo, se supone que se ubica de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 32 cierra el registro zone, indicado por una llave derecha.
- La línea 33 abre un registro zone multilínea para el dominio público, 10.168.192.in-addr.arpa, utilizado para búsquedas de dirección a nombre.
- La línea 34 declara los datos de la zona descritos por este registro como la copia maestra. Este nombre de servidor es el servidor autorizado para esta zona.
- La línea 35 indica que no se notificará a ningún otro servidor si cambian los datos de la zona.
- La línea 36 proporciona el nombre del archivo de la base de datos de la zona. Como named.public es un nombre de trayectoria relativo, se supone que se ubica de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 37 cierra el registro zone, indicado por una llave derecha.
- La línea 38 abre un registro zone multilínea para la caché del dominio de root, dando la ubicación de los servidores de dominio del root de Internet. El servidor de nombres forward first, así como un servidor de nombres normal, necesita este registro zone como una sugerencia de por donde empezar a buscar servidores de nombres autorizados.
- La línea 39 declara los datos de la zona que describe este registro como una sugerencia. Es decir, es sólo un lugar desde donde empezar.
- La línea 40 proporciona el nombre del archivo de base de datos de la zona. Como root.cache es un nombre de trayectoria relativo, se supone que se ubica de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 41 cierra el registro zone, indicado por una llave derecha.

Configuración del servidor de nombres privado

El servidor de nombres privado está autorizado para dos dominios locales: el interno, LAN privada, my.local.lan, y la LAN DMZ entre el bastión y el firewall de contención, my.firewall.lan. Los clientes locales, tanto los clientes LAN como los clientes locales en la máquina firewall bastión pública, usan el servidor de nombres privado que se ejecuta en una máquina interna. Como ejemplo, la máquina de contención interna es el firewall de contención descrito en el Capítulo 4. El archivo /etc/resolv.conf contiene:

```
search my.local.lan my.firewall.lan
nameserver 127.0.0.1
```

El servidor de nombres siempre necesita el archivo de la base de datos del localhost, /var/named/named.127.0.0.

Como ejemplo, supongamos que el sitio pertenece al bloque de direcciones entre la red 192.168.10.0 y la red 192.168.11.0, y que la dirección IP externa del firewall bastión es 192.168.10.30. La red 192.168.11.0 se usa para la DMZ. La interfaz interna del bastión usa la dirección IP 192.168.11.1. La interfaz externa de la máquina de contención usa la dirección IP 192.168.11.2.

En el directorio /var/named/ se almacenan otros dos pares de archivos de zona. El primer par son los archivos de traducción del DMZ dirección a nombre y nombre a dirección. El segundo par son los archivos de la LAN privada dirección a nombre y nombre a dirección.

El archivo de zona de la base de datos dirección a nombre de la DMZ es /var/named/named.dmz. Contiene:

```
 1. 11.168.192.in-addr.arpa.   IN  SOA  my.dmz.lan.
    postmaster.my.dmz.lan. (
 2.                                  1999072701 ; Serial
 3.                                  28800      ; Refresh after 8 hours
 4.                                  14400      ; Retry 4 hours
 5.                                  3600000    ; Expire after 41 days
 6.                                  86400 )    ; Minimum

 7.          IN       NS       choke.my.dmz.lan.
 8.          IN       MX       10 bastion.my.dmz.lan.

 9. 1        IN       PTR      bastion.my.dmz.lan.
10. 2        IN       PTR      choke.my.dmz.lan.
```

Las líneas 1 a 6 contienen la información de control de la zona. Las líneas 7 a 10 son registros de recurso:

- La línea 1 comienza un registro de control SOA para la zona. 10.168.192.in-addr.arpa. es el origen de la zona. IN indica que los datos de este registro pertenecen a la clase de datos de Internet. SOA indica que este registro de recurso es el registro Inicio de autoridad. my.domain.com. es el nombre del dominio. postmaster.my.domain.com. indica la dirección de correo electrónico de la persona de contacto

responsable de la información de la zona. Los paréntesis abiertos indican el principio de un registro multilínea.

- La línea 2 es un número de serie. Si se ofrece servicio a servidores de nombres secundarios, un cambio en el número de serie indicará que los datos de la zona han cambiado y que necesitarán actualizar sus copias locales de la base de datos de la zona. En este caso, el número de serie se expresa en la convención común de año, mes, día y número de veces que la base de datos fue cambiada en dicha fecha: YYYYMMDDNN.
- La línea 3 es la frecuencia de actualización, en segundos, que indica la frecuencia con la que los servidores secundarios deben comprobar el número de serie de los datos de la zona. En este caso, todos los servidores secundarios comprobarán el número de serie cada ocho horas.
- La línea 4 es la frecuencia de reintento, en segundos, que indica el tiempo que debe esperar un servidor secundario si falló a la hora de intentar contactar con el servidor primario durante el tiempo de actualización. En este caso, un servidor secundario seguirá intentando contactar con el servidor cada cuatro horas.
- La línea 5 es el tiempo de caducidad, en segundos, que indica el momento en que un servidor secundario eliminará la información de la zona de la caché si falla a la hora de contactar el servidor primario durante este tiempo. En este caso, el servidor secundario caduca los datos de la zona si no fue capaz de contactar con el servidor primario durante los últimos Tiempo de vida (TTL, time-to-live) segundos. En este caso, el servidor secundario caduca los datos de la zona si no fue capaz de contactar con el servidor primario durante los últimos 41 días.
- La línea 6 es el intervalo de tiempo de vida, en segundos, que indica el tiempo que los servidores remotos pueden mantener en la caché la información que se devuelve como respuesta a una petición. El intervalo de tiempo de vida para la información que se proporciona a las peticiones remotas es, en este caso, de una semana.
- La línea 7 es un registro de recurso NS, servidor de nombres, que indica que choke.my.dmz.lan., la máquina firewall pública, es el servidor de nombres para este dominio..
- La línea 8 es un registro de recurso MX, intercambiador de correo, que indica que bastion.my.domain.com. es el host de correo o el reenviador de correo para este dominio. El valor 10, el intervalo de preferencia del intercambiador, no tiene sentido en este ejemplo. Para los sitios con múltiples intercambiadores de correo, cada host puede recibir un intervalo de prioridad preferida entre 0 y 65535. Se intenta transmitir el correo a cada host de uno en uno, desde el intervalo inferior al superior, hasta que el envío finaliza con éxito.
- La línea 9 es un registro de recurso PTR, puntero, que indica la traducción de dirección a nombre para el nombre de host bastion.my.dmz.lan. El 1 que aparece al principio en el registro indica que este registro define la traducción de nombre a dirección para la dirección 192.168.11.1.

- La línea 10 es un registro de recurso PTR, puntero, que indica la traducción de dirección a nombre para el nombre de host choke.my.dmz.lan. El 2 que aparece al principio en el registro indica que este registro define la traducción de dirección a nombre para la dirección 192.168.11.2.

El archivo de la base de datos de la zona nombra dirección de la DMZ es el archivo /var/named/named.dmz.reverse. Las primeras ocho líneas son idénticas a las del archivo de zona dirección a nombre, incluyendo el registro SOA, el registro NS y el registro MX. Los registros PTR se reemplazan con el registro A, de address (dirección):

```
 1. 11.168.192.in-addr.arpa.   IN  SOA  my.dmz.lan.
    postmaster.my.dmz.lan. (
 2.                                  1999072701 ; Serial
 3.                                  28800      ; Refresh after 8 hours
 4.                                  14400      ; Retry 4 hours
 5.                                  3600000    ; Expire after 41 days
 6.                                  86400 )    ; Minimum

 7.          IN       NS       choke.my.dmz.lan.
 8.          IN       MX       10 bastion.my.dmz.lan.

 9. bastion.my.dmz.lan.      IN    A     192.168.11.1
10. choke.my.dmz.lan.        IN    A     192.168.11.2
```

La línea 9 es un registro recurso A, dirección, que indica que la traducción de nombre a dirección para el nombre de host bastion.my.dmz.lan. es la dirección 192.168.11.1.

La línea 10 es un registro recurso A, dirección, que indica que la traducción de nombre a dirección para el nombre de host choke.my.dmz.lan. es la dirección 192.168.11.2.

El archivo de la zona de la base de datos dirección a nombre de la LAN privada es el archivo /var/named/named.local.lan. Contiene:

```
 1. 1.168.192.in-addr.arpa.   IN   SOA  my.local.lan.
    postmaster.my.local.lan. (
 2.                                  1999072701 ; Serial
 3.                                  28800      ; Refresh after 8 hours
 4.                                  14400      ; Retry 4 hours
 5.                                  3600000    ; Expire after 41 days
 6.                                  86400 )    ; Minimum

 7.          IN       NS       choke.my.local.lan.
 8.          IN       MX       10 bastion.my.dmz.lan.

 9. 1        IN       PTR      choke.my.local.lan.
10. 2        IN       PTR      macintosh.my.local.lan.
11. 3        IN       PTR      bsd.my.local.lan.
```

Las líneas 1 a 6 contienen la información de control de la zona. Las líneas 7 a 11 son registros de recurso:

- La línea 1 comienza un registro de control SOA para la zona. 1.168.192.in-addr.arpa. es el origen de la zona. IN indica que los datos de este registro pertenecen a la clase de datos de Internet. SOA indica que este registro de recurso es el registro Inicio de autoridad. my.local.lan. es el nombre del dominio. postmaster.my.local.lan. indica la dirección de correo electrónico de la persona de contacto responsable de la información de la zona. Los paréntesis abiertos indican el principio de un registro multilínea.
- La línea 2 es un número de serie. Si se ofrece servicio a servidores de nombres secundarios, un cambio en el número de serie indicará que los datos de la zona han cambiado y que necesitarán actualizar sus copias locales de la base de datos de la zona. En este caso, el número de serie se expresa en la convención común de año, mes, día y número de veces que la base de datos fue cambiada en dicha fecha: YYYYMMDDNN.
- La línea 3 es la frecuencia de actualización, en segundos, que indica la frecuencia con la que los servidores secundarios deben comprobar el número de serie de los datos de la zona. En este caso, todos los servidores secundarios comprobarán el número de serie cada ocho horas.
- La línea 4 es la frecuencia de reintento, en segundos, que indica el tiempo que debe esperar un servidor secundario si falló a la hora de intentar contactar con el servidor primario durante el tiempo de actualización. En este caso, un servidor secundario seguirá intentando contactar con el servidor cada cuatro horas.
- La línea 5 es el tiempo de caducidad, en segundos, que indica el momento en que un servidor secundario eliminará la información de la zona de la caché si falla a la hora de contactar el servidor primario durante este tiempo. En este caso, el servidor secundario caduca los datos de la zona si no fue capaz de contactar con el servidor primario durante los últimos 3600000 segundos. En este caso, el servidor secundario caduca los datos de la zona si no fue capaz de contactar con el servidor primario durante los últimos 41 días.
- La línea 6 es el intervalo de tiempo de vida, en segundos, que indica el tiempo que los servidores remotos pueden mantener en la caché la información que se devuelve como respuesta a una petición. El intervalo de tiempo de vida para la información que se proporciona a las peticiones remotas es, en este caso, de una semana.
- La línea 7 es un registro de recurso NS, servidor de nombres, que indica que choke.my.local.com., la máquina firewall interna, es el servidor de nombres para este dominio.
- La línea 8 es un registro de recurso MX, intercambiador de correo, que indica que bastion.my.dmz.lan. es el host de correo o el reenviador de correo para este dominio. El valor 10, el intervalo de preferencia del intercambiador, no tiene sentido en este ejemplo. Para los sitios con múltiples intercambiadores de correo, cada host puede recibir un intervalo de prioridad preferida entre 0 y 65535. Se intenta transmitir

el correo a cada host de uno en uno, desde el intervalo inferior al superior, hasta que el envío finaliza con éxito.

- La línea 9 es un registro de recurso PTR, puntero, que indica la traducción de dirección a nombre para el nombre de host bastion.my.local.lan. El 1 que aparece al principio en el registro indica que este registro define la traducción de dirección a nombre para la dirección 192.168.1.1.
- La línea 10 es un registro de recurso PTR, puntero, que indica la traducción de dirección a nombre para el nombre de host macintosh.my.local.lan. El 2 que aparece al principio en el registro indica que este registro define la traducción de dirección a nombre para la dirección 192.168.1.2.
- La línea 11 es un registro de recurso PTR, puntero, que indica la traducción de dirección a nombre para el nombre de host bsd.my.local.lan. El 3 que aparece al principio en el registro indica que este registro define la traducción de dirección a nombre para la dirección 192.168.1.3.

El archivo de la zona de la base de datos nombre a dirección de la LAN privada es el archivo /var/named/named.local.lan.reverse. Las primeras ocho líneas son idénticas a las del archivo de la zona dirección a nombre, incluyendo el registro SOA, el registro NS y el registro MX. Los registros PTR se reemplazan con el registro A, de address (dirección):

```
1. 1.168.192.in-addr.arpa.   IN  SOA  my.local.lan.
   postmaster.my.local.lan. (
2.                                1999072701 ; Serial
3.                                28800      ; Refresh after 8 hours
4.                                14400      ; Retry 4 hours
5.                                3600000    ; Expire after 41 days
6.                                86400 )    ; Minimum

7.          IN       NS       choke.my.local.lan.
8.          IN       MX       10 bastion.my.dmz.lan.

9. choke.my.local.lan.      IN    A     192.168.1.1
10. macintosh.my.local.lan. IN    A     192.168.1.2
11. bsd.my.local.lan.       IN    A     192.168.1.3
```

La línea 9 es un registro recurso A, dirección, que indica que la traducción de nombre a dirección para el nombre de host choke.my.local.lan. es la dirección 192.168.1.1.

La línea 10 es un registro recurso A, dirección, que indica que la traducción de nombre a dirección para el nombre de host macintosh.my.local.lan. es la dirección 192.168.1.2.

La línea 11 es un registro recurso A, dirección, que indica que la traducción de nombre a dirección para el nombre de host bsd.my.local.lan. es la dirección 192.168.1.3.

El archivo de configuración del servidor named local, /etc/named.conf, contiene lo siguiente:

```
 1. options {
 2.         directory "/var/named";
 3.         forward only;
 4.         forwarders {
 5.                  192.168.11.1;
 6.         } ;
 7.          query-source address * port 53;
 8.          allow-query {
 9.                 127/8;
10.                 192.168.11.1;
11.                 ! 192.168.11/24;
12.                 192.168.1/24;
13.          } ;
14.          allow-transfer {  ! *; } ;
15.          allow-update {  ! *; } ;
16.          listen-on port 53 {  *; } ;
17. } ;
18. zone "0.0.127.in-addr.arpa" {
19.          type master;
20.          notify no;
21.          file "named.127.0.0";
22. } ;
23. zone "my.dmz.lan" {
24.          type master;
25.          notify no;
26.          file "named.dmz.reverse";
27. } ;
28. zone "11.168.192.in-addr.arpa" {
29.          type master;
30.          notify no;
31.          file "named.dmz";
32. } ;
33. zone "my.local.lan" {
34.          type master;
35.          notify no;
36.          file "named.local.lan.reverse";
37. } ;
38. zone "1.168.192.in-addr.arpa" {
39.          type master;
40.          notify no;
41.          file "named.local.lan";
42. } ;
43. zone "." {
44.          type hint;
45.          file "root.cache";
46. } ;
```

El archivo /etc/name.conf contiene dos clases de registros en este ejemplo: el registro options (opciones) y el registro zone (zona). Se pueden definir las

opciones globales del servidor en el registro options, y definir los valores de opción individuales en los registros zone individuales para que se apliquen sólo a dicha zona. Las líneas 1 a 17 definen un registro options:

- La línea 1 declara el tipo de registro como un registro options y abre un registro multilínea con una llave izquierda.
- La línea 2 define el directorio de trabajo del servidor de nombres, /var/named/, donde se guardan los archivos maestro de zona.
- La línea 3 indica al servidor que funcione como un servidor de nombres forward only (sólo de reenvío). Es decir, el servidor dirigirá las peticiones no resueltas sólo a los host listados en el registro forwarders. El único servidor al que reenviará el servidor de nombres de la máquina de contención será el servidor que se ejecuta en la máquina firewall bastión.
- La línea 4 abre un registro forwarders multilínea, indicado por una llave izquierda. El registro forwarders contiene una lista de servidores a los que reenviar las peticiones.
- La línea 5 lista la dirección IP de la interfaz interna del firewall bastión.
- La línea 6 cierra el registro forwarders, indicado por una llave derecha.
- La línea 7 es necesaria cuando un firewall se sitúa entre el servidor local y la red externa. La línea define que el puerto origen del servidor sea el puerto 53 a la hora de enviar peticiones de búsqueda de igual a igual a otros servidores. Es decir, el servidor usa el puerto 53 tanto como puerto origen como puerto destino para peticiones de servidor a servidor.
- La línea 8 abre un registro allow-query multilínea, indicado por una llave izquierda. El registro allow-query contiene una lista de direcciones IP de red de las que se aceptan y deniegan peticiones. El orden de las direcciones es importante. La lista de direcciones se revisa para ver si coincide con el orden en que se define la lista. La primera regla que coincide gana.
- La línea 9 indica al servidor para que acepte las peticiones procedentes desde el localhost.
- La línea 10 indica al servidor que acepte las peticiones procedentes desde clientes del servidor de nombres público en la dirección 192.168.11.1. Si el servidor privado no tiene la información en la caché local, reenviará la petición al servidor de nombres público, el cual puede o no reenviar la petición a servidores de nombres remotos.
- La línea 11 indica al servidor que deniegue las peticiones procedentes de cualquier host de la LAN DMZ. En este caso, se espera que todos los *host* de la red DMZ dirijan sus peticiones al servidor de nombres externo público.
- La línea 12 indica al servidor que acepte peticiones procedentes de cualesquiera máquinas en la LAN privada interna.
- La línea 13 cierra el registro allow-query, indicado por una llave derecha.
- La línea 14 contiene un registro allow-transfer de una sola línea. El registro allow-transfer usa el operador de negación, !, para denegar las

peticiones de transferencia de zona procedentes de cualquier lugar. De forma predeterminada, cualquier lugar tiene permiso para las transferencias de zona.

- La línea 15 contiene un registro allow-update de una sola línea. El registro allow-update usa el operador de negación, !, para denegar las instrucciones de actualización de la base de datos de la zona procedentes de cualquier lugar. De forma predeterminada, se deniegan las actualizaciones de zona. Si se incluye la opción de forma explícita, proporciona una forma de documentar la línea, así como una regla de copia de seguridad en caso de error.
- La línea 16 abre un registro listen-on multilínea, indicado por una llave izquierda. El registro listen-on define el puerto en el que el servidor escucha esperando peticiones procedentes de clientes. El registro contiene una lista de interfaces de red locales en las que escuchar. El servidor de nombres interno escucha en todas las interfaces.
- La línea 17 cierra el registro options, indicado por una llave derecha.
- La línea 18 abre un registro zone multilínea para la red de bucle invertido. Los nombres de dominio de zona se especifican como dominios ARPANET, de forma que la dirección de red de loopback, 127.0.0, se especifica en orden inverso de tuplas separadas por puntos, como un dominio ARPANET 0.0.127.in-addr.arpa.
- La línea 19 declara los datos de zona que describe este registro, para que sean la copia maestra. Este servidor de nombres es el servidor autorizado para esta zona.
- La línea 20 indica que no se notificará a ningún otro servidor si se cambian los datos de la zona.
- La línea 21 proporciona el nombre del archivo de la base de datos de la zona. Como named.127.0.0 es un nombre de trayectoria relativo, se supone que se localiza de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 22 cierra el registro options, indicado por una llave derecha.
- La línea 23 abre un registro zone multilínea para el dominio público, my.dmz.lan. La información de zona de este fichero se usa para búsquedas inversas de nombre a dirección.
- La línea 24 declara los datos de zona descritos por este registro como la copia maestra. Este servidor de nombres es el servidor autorizado para esta zona.
- La línea 25 indica que no se notificará a ningún otro servidor si los datos de la zona cambian.
- La línea 26 proporciona el nombre del archivo de la base de datos de la zona. Como named.dmz.reverse es un nombre de trayectoria relativo, se supone que se ubica de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 27 cierra el registro zone, indicado por una llave derecha.
- La línea 28 abre un registro zone multilínea para el dominio DMZ interno, 11.168.192.in-addr.arpa, utilizado para búsquedas de dirección a nombre.

- La línea 29 declara los datos de la zona descritos por este registro como la copia maestra. Este nombre de servidor es el servidor autorizado para esta zona.
- La línea 30 indica que no se notificará a ningún otro servidor si cambian los datos de la zona.
- La línea 31 proporciona el nombre del archivo de la base de datos de la zona. Como named.dmz es un nombre de trayectoria relativo, se supone que se ubica de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 32 cierra el registro zone, indicado por una llave derecha.
- La línea 33 abre un registro zone multilínea para el dominio local interno, my.local.lan. La información de zona de este archivo se utiliza para búsquedas inversas nombre a dirección.
- La línea 34 declara los datos de la zona que describe este registro como la copia maestra. Este servidor de nombres es el servidor autorizado para esta zona.
- La línea 35 indica que no se notificará a ningún otro servidor si cambian los datos de la zona.
- La línea 36 proporciona el nombre del archivo de la base de datos de zona. Como named.local.lan.reverse es un nombre de trayectoria relativo, se supone que se ubica de forma relativa al directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 37 cierra el registro zone, indicado por una llave derecha.
- La línea 38 abre un registro zone multilínea para el dominio local interno, 1.168.192.in-addr.arpa, utilizado para búsquedas dirección a nombre.
- La línea 39 declara los datos de la zona que describe este registro como la copia maestra. Este servidor de nombres será el servidor autorizado para esta zona.
- La línea 40 indica que no se notificará a otros servidores que los datos de la zona han cambiado.
- La línea 41 proporciona el nombre del archivo de la base de datos de la zona. Como named.local.lan es un nombre de trayectoria relativo, se supone que se ubica de forma relativa al directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 42 cierra el registro zone, indicado por una llave derecha.
- La línea 43 abre un registro zone multilínea para la caché del dominio de root, dando la ubicación de los servidores de dominio del root de Internet. El servidor de nombres forward first, así como un servidor de nombres normal, necesita este registro zone como una sugerencia de por dónde empezar a buscar servidores de nombre autorizados.
- La línea 44 declara los datos de la zona que describe este registro como una sugerencia. Es decir, es sólo un lugar desde donde empezar.
- La línea 45 proporciona el nombre del archivo de la base de datos de la zona. Como root.cache es un nombre de trayectoria relativo, se supone que se ubica de forma relativa en el directorio /var/named/, especificado como el valor directory en el registro options.
- La línea 46 cierra el registro zone, indicado por una llave derecha.

Cómo controlar el acceso a datos de la zona

Con la excepción de la corrupción de la caché DNS basada en el usurpamiento de dirección IP, permitir las peticiones de cliente DNS entrantes procedentes de host remotos no supone un gran riesgo de seguridad, suponiendo que las bases de datos de la zona maestra local no contienen información que no se quiere publica. (las grandes organizaciones a veces usan la base de datos DNS como un almacén central para información privada, como los nombres, números de teléfono y direcciones de sus empleados). Como ya se ha mostrado, se puede usar la opción allow-query del servidor de nombres para restringir quién puede realizar peticiones al servidor de nombres.

El mayor riesgo potencial a la hora de ejecutar un servidor de nombres, está en otorgar el permiso de lectura a toda la base de datos de la zona sobre una conexión TCP. Como se muestra en los ejemplos anteriores, se permiten las transferencias de zona salientes hacia host remotos, de forma predeterminada. El acceso a toda la base de datos de la zona debe restringirse sólo a los servidores de nombres secundarios oficiales, si existe alguno. La opción allow-transfer es necesaria para restringir qué host tienen autoridad para la transferencia de zona. Permitir el acceso a toda la base de datos implica permitir el acceso remoto a toda la topología de la LAN.

El mayor riesgo potencial está en otorgar el permiso de escritura a la base de datos de la zona. Como se muestra en los ejemplos anteriores, las actualizaciones de base de datos entrantes procedentes de host remotos se deniegan de forma predeterminada. El permiso de escritura para las bases de datos de los servidores secundarios debe restringirse sólo al servidor de nombres primario oficial. La opción allow-update es necesaria para permitir a los host tener la autoridad en las actualizaciones de zona. La concesión del permiso de escritura para la base de datos permite a los sitios remotos usurpar o volver a escribir toda la idea que se tiene del propio dominio.

Cuestiones relacionadas con la configuración FTP

FTP tiene una larga historia en cuanto a explosiones de seguridad. Aunque ya se han corregido los problemas conocidos actualmente, seguro que aparecen nuevos problemas con el tiempo. Si nos olvidamos de cualquier posible debilidad en la seguridad del propio software del servidor, el mayor problema potencial de seguridad, la configuración local, sigue en pie. Las dos áreas de interés son los valores en los archivos de configuración de ftp en el directorio /etc, así como los permisos de directorio y de archivo y el contenido en el área pública del ftp anónimo, /home/ftp.

El acceso autenticado al servidor ftp se habilita en el archivo /etc/inetd.conf de forma predeterminada. Si se instala el paquete anonymous ftp, también se habilita el acceso anónimo a ftp. ftpd está protegido por tcp_wrappers. El servidor ftpd se inicia mediante inetd y se le pasan dos opciones de línea de comando, -l y -a. La opción -l activa el registro. Todas las sesiones ftp las registra syslogd. La opción -a indica a ftpd que use su archivo de configuración de

acceso, /etc/ftpaccess. Si se alberga un servicio público de ftp anónimo, se recomiendan otras dos opciones más, -i y -o. La opción -i registra todas las transferencias de archivo entrantes en el archivo /var/log/xferlog. La opción -o registra todas las copias de archivo salientes en el archivo /var/log/xferlog.

Para obtener más información: Configuración de FTP

Para obtener más información sobre la configuración de ftp, consulte las páginas man ftpd(8), ftpaccess(5), ftpconversion(5) y xfer-log(5). Para obtener más información sobre la configuración de ftp anónimo, consulte los documentos "Anonymous FTP Configuration Guidelines" (Directrices generales sobre la configuración de FTP anónimo), que se puede obtener en ftp://ftp.cert.org/pub/tech_tips/anonymous_ftp_config, "UNIX Configuration Guidelines" (Directrices generales sobre UNIX), que se puede conseguir en http://www.cert.org/ftp/tech_tips/UNIX_configuration_guidelines y "UNIX Computer Security Checklist" (Lista de comprobación de seguridad de UNIX), que se puede conseguir en http://www.cert.org/ftp/tech_tips/AUSCERTchecklist.1. Para obtener más información sobre los problemas de la seguridad de ftp, consulte los documentos "Anonymous FTP Abuses" (Abusos de FTP anónimo), que puede obtener en http://www.cert.org/ftp/tech_tips/anonymous_ftp_abuses y "Problems With The FTP PORT Command" (Problemas con el comando PORT de FTP), que puede obtener en http://www.cert.org/ftp/tech_tips/FTP_PORT_attacks.

FTP autenticado y archivos de configuración general de /etc

FTP tiene cinco archivos de configuración en el directorio /etc:

- /etc/ftpaccess es el archivo de configuración principal para el servidor ftpd.
- /etc/ftpconversions contiene una lista de especificaciones de conversión para los archivos comprimidos de los tipos compress, gzip y tar.
- /etc/ftpgroups contiene una lista de grupos de usuarios y el área de ftp en la que el servidor debería usar chroot para cada grupo. Por ejemplo, debe definir grupos independientes para grupos de desarrollo de software diferentes.
- /etc/ftphosts contiene una lista de usuarios que tienen permiso para acceder al sistema como usuarios ftp autenticados, y los host desde los que pueden conectarse.
- /etc/ftpusers contiene una lista de cuentas locales válidas a las que no se concede acceso ftp autenticado. El root y otras cuentas de sistema deberían aparecer en el listado de ftpusers.

Es probable que se quieran personalizar los archivos de configuración ftphosts y ftpaccess. Quizá nunca tengan que modificarse los archivos ftpconversions, ftpgroups o ftpusers.

Un ejemplo de un archivo /etc/ftphosts es:

```
allow bob 192.168.1.*
allow jake 10.10.47.112
```

La cuenta bob tiene permiso para acceso autenticado desde cualquier host de la LAN privada. La cuenta jake tiene permiso para acceso autenticado sólo desde el host remoto en la dirección 10.10.47.112.

Como tcp_wrappers protege a ftpd, la siguiente entrada en el archivo /etc/hosts.allow habilitará estas conexiones entrantes al servidor ftp:

```
in.ftpd: LOCAL 192.168.1. 10.10.47.112
```

El acceso ftp autenticado requiere que los usuarios tengan cuentas locales en el sistema, contraseñas válidas en el archivo /etc/passwd y un shell estándar definido en /etc/passwd.

/etc/ftpaccess es el archivo de configuración de servidor principal. ftpd puede tomar muchas opciones de configuración. A continuación se muestra un ejemplo de un archivo de configuración:

```
 1. class   friends   real  192.168.1.*
 2. class   friends   real  10.10.47.112
 3. # class   other     anonymous  *

 4. deny !nameserved .no_name_server

 5. greeting brief
 6. # banner /home/ftp/banner

 7. message /welcome.msg             login
 8. message .message                 cwd=*

 9. email root@localhost

10. loginfails 5
11. limit-time anonymous 30
12. anonymous-root /home/ftp
13. defaultserver private

14. compress        yes     friends
15. tar             yes     friends
16. chmod           no      friends
17. delete          yes     friends
18. overwrite       yes     friends
19. rename          yes     friends

20. noretrieve passwd shadow group gshadow core

21. upload  /home/ftp *              no
22. upload  /home/ftp /incoming yes  bob   bob  0622 dirs
23. upload  /home/ftp /incoming yes  jake  jake 0622 nodirs

24. log transfers   anonymous        inbound,outbound
25. log security    anonymous

26. passwd-check rfc822 enforce
```

El archivo /etc/ftpaccess contiene definiciones y permisos para grupos de acceso, punteros a archivos de información de titular de inicio de sesión y la ubicación de los archivos de registro:

- La línea 1 define una clase de usuarios autenticados, friends, que tienen permiso para acceder desde cualquier host en la red local 192.168.1.

- La línea 2 agrega otro miembro a la clase friends para incluir el acceso desde el host remoto en la dirección 10.10.47.112.
- La línea 3 define una clase de usuarios anónimos, other, que tienen permiso para acceder desde cualquier host. La línea se comenta.
- La línea 4 deniega el acceso a los sitios cuyas direcciones no se resuelven. Se muestra al usuario el contenido del archivo, /home/ftp/.no_name_server.
- La línea 5 define la cantidad de información que se muestra antes de que el usuario remoto inicie la sesión. brief sólo muestra el nombre del host. De forma predeterminada, se muestran tanto el nombre de host como la versión de ftpd.
- La línea 6 define un archivo que contiene un mensaje que se mostrará antes de que el usuario inicie la sesión. La línea se comenta.
- La línea 7 define un archivo que contiene un mensaje que se mostrará después de que el usuario inicie la sesión.
- La línea 8 define un archivo que contiene un mensaje que se mostrará cada vez que el usuario cambie de directorio.
- La línea 9 especifica la dirección de correo electrónico del propietario del área ftp.
- La línea 10 especifica el número de intentos de inicio de sesión fallidos que un usuario puede realizar antes de terminar la conexión.
- La línea 11 define la cantidad de tiempo, en minutos, que puede durar una sesión anónima. En este caso, si el usuario permanece conectado y deja de usar el equipo, la sesión ftp terminará después de 30 minutos de inactividad.
- La línea 12 especifica la cima del árbol del directorio de ftp anónimo.
- La línea 13 es otra forma de denegar el acceso anónimo.
- Las líneas de la 14 a la 19 define para qué están autorizados los miembros de funciones de una clase. En este ejemplo, los usuarios autenticados no tienen permiso para cambiar los permisos de lectura o escritura de un archivo.
- La línea 20 especifica los archivos que nadie puede transferir.
- La línea 21 deniega las transferencias de archivo entrantes en el área de ftp anónimo.
- La línea 22 permite a la cuenta autenticada bob cargar archivos en el directorio incoming y crear directorios adicionales bajo el directorio /home/ftp/incoming.
- La línea 23 permite a la cuenta autenticada jake cargar archivos en el directorio incoming, pero no permite crear ningún subdirectorio más.
- La línea 24 registra todas las trasferencias anónimas de archivos, tanto entrantes como salientes.
- La línea 25 registra cualquier intento que realizan los usuarios anónimos de acceder a archivos asegurados o de realizar funciones ilegales.
- La línea 26 habilita la comprobación estricta de contraseña y termina la sesión si el usuario no proporciona una contraseña compatible.

FTP anónimo y /home/ftp

Si se instala el paquete anonymous ftp, el ftp anónimo estará disponible tan pronto como se inicie el sistema y habilite la conexión en red. El acceso ftp anónimo exige que se haya creado una cuenta ftp en el archivo /etc/passwd. La entrada de la cuenta ftp en el archivo /etc/passwd debería tener una contraseña deshabilitada, indicada por un símbolo * en el campo contraseña. Su directorio raíz debería ser /home/ftp. Su shell de inicio de sesión debería ser /bin/false.

Los usuarios anónimos se conectan como la cuenta de usuario anónimo, ftp. El servidor ftpd realiza automáticamente un chroot en el área de ftp anónimo, /home/ftp, cuando inetd inicia el servidor para la conexión actual y el usuario solicita una sesión anónima.

Además de las características generales de configuración del servidor, debe configurarse con mucho cuidado el contenido, la propiedad y los permisos de acceso en el árbol del directorio /home/ftp. La configuración errónea de los permisos de acceso del área anónima son una de las principales razones por las que los sitios ftp pueden verse comprometidos.

/home/ftp y sus subdirectorios no deberían ser propiedad del usuario anónimo, ftp, ni deberían estar en el grupo de usuarios ftp. Los directorios deberían ser propiedad del root o de alguna otra cuenta. Sólo el propietario del directorio, root, debería tener permiso de escritura en el directorio /home/ftp o en cualquiera de sus subdirectorios.

El área de ftp anónimo contiene cuatro subdirectorios predeterminados: bin, etc, lib y pub. Como el ftp anónimo se ejecuta en un entorno chroot, los archivos que usa ftpd y que suelen residir en los directorios /bin, /etc y /lib, deben duplicarse en directorios con los mismos nombres y trayectorias, relativas al directorio del root de chroot, /home/ftp. Cada uno de estos directorios requiere pertenencia a grupo y permisos de acceso ligeramente diferentes:

- /home/ftp/bin contiene copias de los programas ejecutables que usa ftp para ofrecer la función ls, la función cd y la compresión de archivos. bin y su contenido deberían tener activados los bits de acceso al directorio o de ejecución para todos, y tener denegados para todos los permisos de lectura y escritura. Es decir, bin y todos los archivos bajo el directorio bin deberían ser chmod 0111.
- /home/ftp/etc contiene una copia local del archivo de caché que contiene información sobre bibliotecas cargadas dinámicamente, ld.so.cache. Este archivo sólo debe poder escribirlo el root y ser legible para todos. El directorio etc también contiene copias tontas de los archivos /etc/passwd y /etc/group. Estos dos archivos deberían contener información mínima de cuentas para propietarios de archivos en el área de ftp anónimo. Normalmente, los propietarios serán las cuentas root y ftp. El archivo /home/ftp/etc/passwd no debería, bajo ningún concepto, contener ninguna contraseña. Las entradas de contraseña deberían deshabilitarse escribiendo un * en el campo contraseña. Los archivos group y passwd deben poder leerlos todos los usuarios, pero no deberían tener permiso de escritura para nadie. El único propósito de los

archivos group y passwd es ofrecer al propietario y a los nombres de los miembros del grupo el comando ls. El directorio etc debería tener activados los bits de acceso de ejecución o directorio para todos y permisos de lectura y escritura denegados para todos. Es decir, el directorio etc debería ser chmod 0111.

- /home/ftp/lib contiene copias de las pocas bibliotecas dinámicas que se pueden cargar y que ftp necesita. El directorio lib y su contenido deberían tener activados los bits de acceso de ejecución y direcotio, así como los bits de permiso de lectura, para todos. El permiso de escritura debe denegarse a todos. Es decir, el directorio lib y todos los archivos debajo de el deberían ser chmod 0555.
- /home/ftp/pub es la cima del árbol de directorio del ftp público. El directorio pub y cualquier subdirectorio contienen los archivos que el servidor ftp anónimo hace disponibles. pub y sus subdirectorios deberían tener activados los bits de acceso de ejecución y directorio, así como los bits de acceso de lectura, para todos. El permiso de escritura debería denegarse a todos. Es decir, pub y todos los directorios bajo pub deberían ser chmod 0555. Todos los archivos bajo pub deberían ser legibles por todos. Es decir, los archivos bajo pub deberían ser chmod 0444.

Precaución: Contenido del archivo de contraseña FTP

Es especialmente importante que /home/ftp/etc/passwd no contenga contraseñas. Un error habitual de configuración suele ser usar una copia del archivo de contraseña del sistema que contenga contraseñas cifradas. Resulta sencillo para los usuarios anónimos copiar el archivo passwd, obtener los nombres de las cuentas de usuario y acceder de forma regular mediante rlogin remoto.

Precaución: Contenido del archivo de contraseña FTP

En la instalación predeterminada de Red Hat 6.0, el directorio pub es propiedad del root y es uno de los miembros del grupo ftp. pub y todos los directorios bajo pub están setgid al grupo ftp. Este no es un problema de seguridad inmediato, ya que ninguno de los directorios tiene permiso de escritura. Sin embargo, si se ha permitido el acceso de escritura a uno de estos directorios y se ofrece el servicio de ftp anónimo, esta configuración predeterminada abriría el sitio a una de las explosiones de seguridad más habituales de ftp. Un usuario remoto anónimo podría escribir archivos en el sistema y modificar los archivos existentes. Es decir, el usuario remoto podría establecer un WAREZ oculto e ilegal en el sistema en su área de ftp, montar un ataque de denegación de servicio llenando el sistema de archivos o, dependiendo de los servicios locales que se hayan habilitado para los usuarios anónimos en /etc/ftpaccess, el usuario también podría modificar los permisos de archivos y directorios

---

Cuestiones relacionadas con la configuración del servidor POP

El servicio de recuperación de correo POP, si se usa, generalmente se configurará en un sitio pequeño como un servicio local privado. Los usuarios lo-

cales usarán POP para recuperar su correo en estaciones de trabajo desde un servidor de correo central. Si se necesita ofrecer servicio POP a sitios remotos, es necesario hacerlo con mucho cuidado para asegurar que es posible una configuración segura, como exigir que las conexiones se hagan sobre SSH.

Como servicio local y privado, el servidor popd se habilita en el archivo /etc/inetd.conf. Si se dispone de todos los servicios tcp_wrappered disponibles para las máquinas locales en el archivo /etc/hosts.allow, no es necesaria una entrada ipop3d específica, y todos los accesos externos al servicio se deniegan, de forma predeterminada, en el archivo /etc/hosts.deny. La seguridad se incrementa fuertemente mediante el firewall de filtrado de paquetes, ya que las conexiones remotas entrantes al servidor POP se deniegan de forma predeterminada. No existen reglas de firewall que permitan el acceso al servidor local.

Para configurar el servidor ipop3d, es necesario agregar cada cuenta de usuario de correo al grupo popusers en el archivo /etc/group. El programa de configuración del sistema, linuxconf, proporciona una interfaz GUI para modificar el archivo /etc/group.

Los archivos de configuración del servidor POP se guardan en el directorio /etc/ppp. Dos archivos, options y pap-secrets, son de especial interés (como la mayoría de los sitios usan autenticación PAP, en vez de autenticación CHAP, no se modificara el tercer archivo, chap-secrets).

A continuación se muestra un ejemplo del archivo de configuración /etc/ppp/options, junto con una explicación:

```
1. usehostname
2. noipdefault
3. auth
4. login
5. require-pap
```

La línea 1, usehostname, exige que se especifique el nombre de host del servidor como parte del proceso de autenticación.

La línea 2, noipdefault, exige que el cliente especifique la dirección IP del servidor como parte del proceso de autenticación.

La línea 3, auth, exige que el cliente se autentifique a sí mismo antes de que se establezca la conexión real de recuperación de correo.

La línea 4, login, indica al servidor POP que use la contraseña que se encuentra en /etc/passwd para autenticar al usuario, en vez de tener que especificar una contraseña ASCII en texto sin cifrar en el archivo pap-secrets (no debería usarse la opción login para las cuentas POP de acceso telefónico).

La línea 5, require-pap, exige autenticación PAP para que se pueda usar.

A continuación se muestra un ejemplo del archivo de configuración /etc/ppp/pap-secrets, junto con una explicación que contiene una entrada para cada usuario individual con autorización de acceso al correo POP:

```
1. # user  server        secret  IP addresses
2. name    192.168.1.1   ""      192.168.1.3 host.local.lan
```

La línea 1 es un comentario que etiqueta los campos necesarios en el archivo de configuración.

La línea 2 contiene una entrada de ejemplo de cuenta de usuario:

- name es el nombre de inicio de sesión de la cuenta de usuario.
- 192.168.1.1 es la dirección IP del servidor, que en este caso es la dirección IP de la interfaz de red interna.
- "" es el campo contraseña de usuario. El campo contiene una cadena en blanco porque el archivo /etc/ppp/options habilita la opción login para que compruebe la contraseña de la cuenta POP mediante el archivo de contraseñas del sistema.
- 192.168.1.3 host.local.lan es una lista de direcciones IP y de nombres de host, desde la que el usuario tiene permiso a conectarse.

De igual modo, a continuación se muestra un ejemplo de un archivo de configuración /etc/ppp/pap-secrets, junto con explicaciones que contienen una entrada global que permite a todas las cuentas de usuario normales autorización para acceder al correo POP:

```
1. # user   server        secret   IP addresses
2. *        192.168.1.1     ""
3. root     192.168.1.1     "*"       -
4. bin      192.168.1.1     "*"       -
5. mail     192.168.1.1     "*"       -
6. games    192.168.1.1     "*"       -
7. nobody   192.168.1.1     "*"       -
```

La línea 1 es un comentario que etiqueta los campos necesarios en el archivo de configuración.

La línea 2 contiene una entrada de cuenta de usuario global:

- * indica cualquier nombre de inicio de sesión de cuenta de usuario válida. No se olvide de incluir todos los usuarios en el grupo popusers en el archivo /etc/group.
- 192.168.1.1 es la dirección IP del servidor, que en este caso es la dirección IP de la interfaz de red interna.
- "" es el campo contraseña de usuario. El campo contiene una cadena de texto en blanco, porque el archivo /etc/ppp/options habilita la opción login para comprobar la contraseña de la cuenta POP a partir del archivo de contraseñas del sistema.
- El campo IP addresses (direciones IP) está vacío, lo que indica que las conexiones del usuario pueden proceder de cualquier lugar.

Las líneas 3 a 7 contienen ejemplos de entradas de cuentas de usuario que no tienen autorización de acceder al servidor POP. Como este ejemplo de configuración permite el acceso de todos los usuarios definidos en el archivo /etc/passwd, las cuentas de usuario sin permiso deben listarse de forma individual:

- el campo user contiene el nombre de cuenta a la que denegar el acceso.
- 192.168.1.1 es la dirección IP del servidor, que en este caso es la dirección IP de la interfaz de red interna.

- "*" es el campo contraseña de usuario. El campo contiene una cadena de un asterisco que indica que la contraseña de usuario no es válida.
- El campo IP addresses (direcciones IP) contiene un guión, que indica que no se permite ninguna dirección origen IP.

Al ser un servicio público restringido, es necesario realizar un par de cuestiones más relacionadas con la configuración, con el fin de mejorar la seguridad. Primero, toda dirección IP remota, o todo nombre de host que tiene permisos, debe listare como un cliente del servidor ipop3d en el archivo /etc/hosts.allow. En segundo lugar, deben definirse las reglas de firewall individuales con el fin de permitir que cada uno de estos host remotos acceda al servidor POP local.

Cuestiones relacionadas con la configuración del servidor DHCP

Resulta peligroso ejecutar un servidor DHCP local, dhcpd, en la máquina firewall. Si los vecinos de la subred del ISP son clientes DHCP, pueden confundir al servidor con el servidor DHCP del ISP. Si se ejecuta un servidor DHCP mal configurado, es muy probable que se rescinda su cuenta con el ISP hasta que se corrija el problema.

Sin embargo, en algunas situaciones resulta adecuado ejecutar un servidor local para uso interno de la LAN. Una situación de este tipo, podría ser un sitio particular pequeño, quizá un sitio compuesto por una sola máquina Linux de doble tarjeta y un equipo portátil que se use tanto en el trabajo como en casa. Otra situación podría ser un sitio de una pequeña empresa con más equipos de las que se pueden administrar de forma adecuada de forma manual.

Para un sitio pequeño con una única LAN interna, la configuración del servidor dhcpd en la máquina firewall requiere especial cuidado antes de llamar al servidor para asegurar que se ofrece servicio a la LAN y no a Internet. Las reglas de firewall deben asegurar meticulosamente que se permite pasar a los mensajes del servidor local sólo a través de la interfaz de red interna a la LAN. Igualmente, si alguna de las máquinas de la LAN es otra máquina Linux, la máquina interna podría albergar el servicio DHCP para otras máquinas de la LAN. Las reglas de firewall deben asegurar que no se reenvían los mensajes DHCP desde la LAN a través de la interfaz de red externa.

Para sitos grandes, con una red DMZ y un segundo firewall de contención interno, el servidor dhcpd podría albergarse en la máquina firewall interna. Tanto el firewall externo como el firewall interno deberían asegurar que el trafico DHCP local permanece dentro de la LAN.

No resulta difícil restringir el tráfico del servidor DHCP a una interfaz de red específica en una máquina multitarjeta. La cuestión es simplemente que la interfaz de red específica debe definirse para dhcpd antes de que el servidor se ejecute por primera vez. Para asignar el servidor dhcpd a una interfaz

en particular, es necesario modificar la secuencia de comandos de inicio, /etc/rc.d/init.d/dhcpd. De forma predeterminada, el servidor se llama como:

```
daemon /usr/sbin/dhcpd
```

Suponiendo que la interfaz externa es eth1, se debe modificar la línea como sigue:

```
daemon /usr/sbin/dhcpd eth1
```

Si el demonio se ejecuta en la máquina firewall, en vez de en una máquina interna, puede que sea necesario agregar una entrada de tabla de enrutamiento para la interfaz de red interna. Si es así, se debe agregar la siguiente línea al archivo /etc/rc.d/rc.local:

```
route add -host 255.255.255.0 dev eth1
```

El archivo de configuración del servidor dhcpd es /etc/dhcpd.conf; el archivo puede contener parámetros globales. El archivo debe contener un registro de declaración para cada subred conectada al servidor. Un ejemplo sencillo para una LAN privada, que cambia dinámicamente, sería:

```
 1. option domain-name "local.lan";
 2. option domain-name-servers 192.168.1.1;
 3. option subnet-mask 255.255.255.0;
 4. subnet 192.168.1.0 netmask 255.255.255.0 {
 5.         range 192.168.1.2 192.168.1.254;
 6.         default-lease-time 86400;
 7.         max-lease-time 2592000;
 8.         option broadcast-address 192.168.1.255;
 9.         option routers 192.168.1.1;
10. }
```

La línea 1 es un parámetro global que especifica el domain-name (nombre de dominio).

La línea 2 es un parámetro global que especifica el domain-name-servers.

La línea 3 es un parámetro global que especifica el subnet-mask. Cualquier dirección IP que coincida con los primeros 24 bits, la dirección de red, es un miembro del espacio de direcciones de red de la subred.

La línea 4 comienza el registro subnet para la interfaz de red particular. La dirección de red de la subred es 192.168.1.0 y su máscara de red es 255.255.255.0. Cualquier dirección IP que coincida con los primeros 24 bits, la dirección de red, es un miembro del espacio de direcciones de la red local.

La línea 5 define el range del grupo de direcciones IP dinámicas por la primera y la última dirección del intervalo.

La línea 6 define el default-lease-time en segundos. Un día tiene 86.400, por lo que el tiempo predeterminado de concesión es de un día.

La línea 7 define el max-lease-time en segundos. 2.592.000 segundos equivalen a 30 días, por lo que el máximo tiempo de concesión es de un mes.

La línea 8 muestra la broadcast-address de la LAN.

La línea 9 especifica el enrutador pasarela de la LAN.

La línea 10 cierra el registro de subred para la interfaz de red particular.

Si se desea más información sobre la configuración de dhcpd, consulte las siguientes páginas man: dhcpd(8) y dhcpd.conf(5).

Cuestiones relacionadas con la configuración de NTP

El protocolo de tiempo de red (NTP, *Network Time Protocol*) se usa para sincronizar los relojes del sistema de los equipos locales con el de un servidor de tiempo remoto autorizado. Normalmente, los sitios ejecutan el cliente NTP, ntpdate, en una sola máquina para obtener la hora actual desde varios servidores remotos. (NTP funciona mejor si se realiza peticiones, como mínimo, a tres servidores remotos). ntpdate hace la mejor estimación de la hora actual, basándose en el nivel y precisión de autoridad de informe del servidor y en el intervalo de diferencias entre las respuestas que recibe de los diferentes servidores, y actualiza la hora del sistema local de acuerdo con estos datos. Para reducir el tráfico de Internet redundante, esta máquina también ejecuta un servidor NTP local, xntpd, para distribuir la hora actual a otras máquinas de la LAN. Si el sitio es grande, este servidor local puede configurarse como un servidor autorizado y maestro para el sitio. Las demás máquinas internas se pueden configurar para que ejecuten sus propios servidores xntpd como servidores secundarios, con el fin de distribuir la hora a múltiples máquinas internas, reduciendo así la carga del servidor maestro.

Para una LAN pequeña, se muestra un ejemplo que consiste en un solo *host* que ejecuta el programa cliente de configuración de reloj, ntpdate, en tiempo de inicio, para inicializar la hora del sistema desde varios servidores remotos. Luego, el host inicia xntpd para ofrecer el servicio horario local a las demás máquinas de la LAN. Las demás máquinas ejecutan xntpd como un cliente para hacer peticiones periódicamente al servidor local.

El servidor se configura para ejecutar la secuencia de comandos de inicio /etc/rc.d/init.d/xntpd en tiempo de inicio. Si existe el archivo de configuración, /etc/ntp/step-tickers, ntpdate lee el nombre de los servidores horarios remotos a los que realizar peticiones desde este archivo. Luego, la secuencia de comandos de inicio ejecuta el demonio xntpd.

El archivo de configuración de xntpd es el archivo /etc/ntp.conf. A continuación se muestra un ejemplo de archivo de configuración ntp.conf, junto con una explicación para el servidor principal de la LAN:

```
1. restrict default nomodify
2. server  127.0.0.1
3. restrict 127.0.0.1
```

La línea 1 define el nivel de confianza en los host remotos. De forma predeterminada, los host remotos son seguros para la hora, pero no tienen per-

miso para realizar modificaciones en la configuración del servidor local. Sin embargo, en esta configuración no se contacta con los servidores remotos mediante el servidor xntpd local.

La línea 2 declara la máquina local como un servidor horario.

La línea 3 elimina las restricciones predeterminadas del servidor local para que se pueda modificar la configuración del servidor localmente en tiempo de ejecución.

La máquina cliente también está configurada para ejecutar la secuencia de comandos de inicio /etc/rc.d/init.d/xntpd durante el inicio. Si existe el archivo de configuración, /etc/ntp/step-tickers, entonces ntpdate lee el nombre de los servidores horarios remotos a los que realizar peticiones desde este archivo. Luego, la secuencia de comandos de inicio ejecuta el demonio xntpd.

El archivo de configuración de xntpd es el archivo /etc/ntp.conf. A continuación se muestra un ejemplo de archivo de configuración ntp.conf, junto con una explicación para las máquinas LAN internas:

```
1. restrict  192.168.11.1 nomodify
2. server 192.168.11.1
3. server  127.0.0.1
4. restrict 127.0.0.1
```

La línea 1 define el nivel de confianza predeterminado en los host remotos. De forma predeterminada, los host remotos son confiables en cuanto a la hora, pero no tienen permiso para realizar modificaciones a la configuración del servidor local.

La línea 2 indica que el servidor remoto será la interfaz de red interna de la máquina firewall. En este caso, no se especifican los intervalos de sondeo mínimo y máximo. El valor minpoll predeterminado es 6, que equivale a aproximadamente a un minuto. El valor predeterminado de maxpoll es 10, que equivale aproximadamente a 15 minutos.

La línea 3 declara la máquina local como un servidor horario.

La línea 4 elimina las restricciones predeterminadas del servidor local para que se pueda modificar la configuración del servidor localmente en tiempo de ejecución.

Advertencia: Interacciones entre ntpdate y xntpd

ntpdate no consultará el tiempo si se ejecuta actualmente xntpd.

Para obtener más información: Servidores oficiales de NTP

Para funcionar como servidor principal es necesario que reciba la hora mediante un modem o receptor de satélite o radio. Existen actualmente alrededor de cincuenta servidores principales públicos en todo el mundo, soportados por unos cien servidores secundarios públicos, que a su vez están soportados por miles de servidores públicos de capas superiores. Los sitios servidores NTP suelen preferir ofrecer sus servicios a determinadas áreas geográficas o bloques de direcciones de red. Puede encontrar información sobre los servidores públicos secundarios que se encuentren en su área

geográfica en la dirección http://www.eecis.udel.edu/~mills/ntp/servers.html. Se puede conseguir más información sobre documentación, software y el protocolo NTP, en el mismo sitio http://www.eecis.udel.edu/~mills/ntp/.

Cuestiones relacionadas con la secuencia de comandos CGI de HTTP

El servidor web ejecuta secuencias de comandos y programas CGI para realizar funciones especiales que sobrepasan las capacidades del servidor web. Las secuencias de comandos suelen ejecutar programas de sistema locales para realizar sus funciones. Disponen de tanta autorización para acceder a los recursos del sistema como la propia cuenta para la cual la secuencia de comandos dispone de privilegios setuid o el propio servidor web. Las acciones exactas que realiza la secuencia de comandos suelen basarse en datos que el usuario proporciona de forma arbitraria.

Las secuencias de comandos, o guiones CGI, son especialmente vulnerables a las explosiones de seguridad, a menos que se tomen precauciones especiales. Si la secuencia de comandos, o el programa, requiere privilegios especiales de cuenta de sistema para realizar su función, el usuario puede definir los bits setuid o setgid de la secuencia de comandos para ejecutar programas como un usuario privilegiado. En este caso, la secuencia de comandos se ejecuta con privilegios especiales de sistema. Los procesos del servidor web deberían ejecutarse bajo una cuenta no privilegiada, como usuario nadie. Si se configura de forma incorrecta el demonio principal del servidor web, que se ejecuta con privilegios de root, también se pueden crear procesos secundarios de servidor sin los privilegios de root, igual que con los privilegios mínimos recomendados del usuario nadie. En este caso, la secuencia de comandos CGI se ejecuta con privilegios de root. Estos posibles aspectos del nivel de autorización de la secuencia de comandos han provocado serias explosiones de seguridad, cuando el usuario remoto ha sido capaz de acceder como root al sistema proporcionando a la secuencia de comandos datos inesperados.

Además de las precauciones de seguridad recomendadas en la documentación del servidor web Apache, que está disponible en la dirección www.apache.org, es necesario poner en práctica dos precauciones especiales. La primera consiste en ejecutar un servidor intermedio, cgiwrap, que protege las secuencias de comandos CGI, de la misma forma que tcp_wrappers protege los servicios que se administran con inetd. La segunda consiste en comprobar con mucho cuidado la entrada del usuario antes de acceder a recursos del sistema en nombre del usuario.

cgiwrap no restringe el acceso remoto a las secuencias de comandos CGI. Por el contrario, cgiwrap asegura que la secuencia de comandos se ejecuta con los permisos que se pretende y no con los del servidor, incluso si el servidor se ejecuta erróneamente con privilegios de root. El paquete de código fuente de cgiwrap, y la documentación, se pueden conseguir en la dirección ftp://ftp.cc.umr.edu/pub/cgi/cgiwrap.

Controlar de forma cuidadosa el contenido de la entrada de usuario en las secuencias de comandos es algo más difícil, ya que la entrada que espera un programa dado es única. Un documento que puede conseguir del CERT, "*How To Remove Meta-characters From User-Supplied Data In CGI Scripts*" (Cómo quitar metacaracteres de las secuencias de comandos CGI proporcionadas por el usuario), en la dirección http://www.cert.org/tech_tips/cgi_metacharacters.html, describe una aproximación general al código de comprobación de errores y ofrece ejemplos de secuencias de comandos Perl y código fuente C. La idea general es que se debe comprobar toda la entrada de usuario. Es necesario descartar el flujo de datos que proporciona el usuario por encima de un máximo esperado. Este paso sólo protege la secuencia de comandos de las explosiones por desbordamiento del búfer. En lugar de intentar escribir código para anticipar cada entrada de usuario indefinida e inesperada, los documentos recomiendan definir exactamente los caracteres ASCII que son legales y descartar cualquier entrada que incluya caracteres ilegales.

SOCKS: un firewall proxy de nivel de aplicación

SOCKS es un paquete proxy de pasarela de circuito (SOCKS era el nombre interno del proyecto de desarrollo y es sinónimo de socket. Después de su lanzamiento, el nombre de desarrollo no cambió y el software siguió llamándose SOCKS). En un entorno LAN, el propósito de SOCK es funcionar como un firewall proxy de nivel de aplicación. Los programas cliente de la LAN sólo tienen permiso de acceso al servidor SOCKS local. Sólo el servidor SOCKS tiene permiso de acceso a Internet. SOCKS es un proxy transparente. Para el usuario, el programa cliente aparenta conectar directamente con el servicio remoto. Como los programas cliente SOCKS se modifican para que sólo se comuniquen con el servidor SOCKS, en su lugar, los programas cliente se conectan con el servidor local. El servidor autentica al usuario y establece una conexión al servicio remoto en nombre del cliente. Cuando se establece la conexión, el servidor SOCKS hace de transmisor de datos entre el cliente local y el servidor remoto. Como el servidor es un proxy para los clientes LAN, las máquinas de la LAN son invisibles a Internet, igual que sucede con el enmascaramiento de dirección IP de nivel de paquete.

La versión 4 de SOCKS es compatible con los servicios proxy de TCP. La actual versión 5 también es compatible con los servicios proxy de UDP y de multidifusión, proporciona unos servicios de búsqueda DNS y autenticación de usuario más estricta para el cliente e incluye el código cliente como bibliotecas compartidas, eliminando la necesidad de tener que volver a compilar los programas cliente en algunos casos.

La nueva transmisión UDP es una característica especialmente atractiva para aprovecharla en un esquema de seguridad multicapa. El filtrado a nivel de paquete no puede realizar correctamente los intercambios UDP. La característica de transmisión UDP crea una asociación de puerto y host entre los procesos cliente y servidor, asegurando que el cliente recibe datagramas sólo

desde el servidor remoto que desea. Para servicios multiprotocolo, como RealAudio, que usa una conexión TCP como canal de control y UDP para el flujo de datos, SOCKS asocia el extremo del host remoto con el origen de donde se esperan los datagramas UDP entrantes.

La posibilidad que tiene SOCKS de realizar un uso correcto de los protocolos del nivel de aplicación también resulta atractiva para los servicios de devolución de llamada de TCP, como FTP, en el modo de puerto de canal de datos. SOCKS asegura que la conexión del canal de datos entrante procedente del puerto TCP 20 se originó desde el host remoto al que el cliente está conectado cuando se estableció el canal de comandos en el puerto TCP 21.

En resumen, SOCKS es una alternativa atractiva a los módulos de enmascaramiento IP que incluye Linux, ya que todos sus programas cliente son compatibles con SOCKS. Aunque los módulos de enmascaramiento incluidos con Linux son compatibles con varios servicios conocidos, SOCKS es compatible con cualquier servicio de red.

En un entorno proxy SOCKS, se deberá bloquear la LAN del acceso a Internet, en vez de permitir el acceso LAN sólo a la interfaz de red interna. Es especialmente importante bloquear los intentos de conexión entrantes procedentes de Internet al servidor SOCKS en el puerto TCP 1080. El servidor nunca debería aceptar conexiones procedentes de sitios remotos.

Como servicio proxy, en realidad SOCKS realiza enmascaramiento IP para los clientes LAN. Cuando SOCKS proporciona el único conducto entre la LAN e Internet, no es necesario el enmascaramiento IP a nivel de paquete.

Se puede obtener SOCKS tanto gratuitamente como en forma de producto comercial. El mejor sitio web de SOCKS se encuentra en la dirección http://www.socks.nec.com/. La versión de referencia no comercial de la versión 5 de SOCKS se puede conseguir en http://www.socks.nec.com/socks5.html.

Cuentas varias del sistema en /etc/passwd y /etc/group

Los archivos de autenticación de usuario, /etc/passwd y /etc/group, contienen unas cuantas entradas especiales de cuentas del sistema. Las entradas permiten a los servicios a los que están asociados ejecutarse sin privilegios de root y controlar de forma más estricta el acceso a los programas y a las áreas del sistema de archivos reservadas para dichos servicios. Dependiendo de la instalación, no se usarán la mayoría de las cuentas del sistema porque el usuario no ejecutará los servicios a los que están asociados dichas cuentas.

En la distribución de Red Hat, las contraseñas de las entradas de cuentas del sistema están deshabilitadas en el archivo /etc/passwd. Éstas contienen un * en el campo contraseña, lo cual no significa que estos usuarios no tengan permiso para iniciar una sesión. Por el contrario, los procesos del sistema se inician con niveles de autorización restringidos en vez de iniciarse con privilegios de root.

Aunque, de forma predeterminada, no se permite iniciar una sesión a estas cuentas del sistema debido a los campos de contraseña deshabilitados, es necesario tener en cuenta dos precauciones de seguridad más profundas: definir /bin/false como el shell de inicio de sesión para las cuentas necesarias que no tienen un shell o un programa y eliminar las cuentas que quedan de los archivos /etc/passwd y /etc/group. Un conjunto mínimo de cuentas del sistema en el archivo /etc/passwd consta de las siguientes entradas:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/false
adm:x:3:4:adm:/var/adm:/bin/false
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/bin/false
nobody:x:99:99:Nobody:/:/bin/false
```

Las restantes cuentas del sistema predeterminadas son cuentas con un propósito especial. Es posible eliminar cualquiera de las cuentas de la Tabla 7.1 que no necesite tanto del archivo de contraseñas como del archivo de grupos.

No recorte en exceso los grupos del sistema

/etc/group contiene otros grupos de sistemas además de los grupos asociados con cuentas del sistema de propósito especial. No quite más nombres del grupo de sistema de /etc/group. Estos grupos adicionales sirven como mecanismo de control de acceso para los subsistemas de software, como los directorios de las páginas man y para ciertos dispositivos, como la consola, el núcleo, la memoria física, las unidades de disco y las terminales físicas.

Tabla 7.1. Cuentas de sistema de propósito especial

Cuenta	Subsistema o servidor asociado
daemon	Servidores PPP y POP.
lp	Subsistema de impresión.
news	Subsistema servidor de noticias.
uucp	Servidores UUCP.
operator	Alias de administración del sistema.
games	Programas de juego variados.
gopher	Servidor Gopher.
ftp	Servidor FTP.
xfs	Servidor de fuentes de X Window.
gdm	Administrador de visualización de Gnome.
postgres	Servidor de bases de datos SQL.
squid	Servidor proxy web squid.

Configuración de la variable PATH

La variable de entorno PATH del shell define los directorios en los que el shell buscará programas ejecutables cuando se ejecute un programa. El

shell busca en los directorios en el orden en que se haya definido en la variable PATH.

El orden de búsqueda de directorios introduce un posible problema de seguridad, especialmente en sistemas multiusuario. A menudo, es conveniente agregar el directorio actual, ., a la variable PATH. Si se usa el punto en las variables PATH de las cuentas de usuario, el punto debería ser el último directorio que se liste. Nunca debería preceder a los directorios binarios del sistema. De lo contrario, alguien podría colocar un programa de caballo de Troya con el mismo nombre que ls, por ejemplo, en un directorio que se quiera listar. En ningún caso debe usarse el punto en la variable PATH del root. La búsqueda de directorios del shell para el root debe restringirse a los directorios binarios del sistema estándar. La cuenta root debería ejecutar los programas en otros directorios proporcionando de forma explícita el nombre completo de ruta en la línea de comandos.

Las variables PATH individuales se definen en el archivo de inicialización del shell del usuario. Dependiendo del shell que se use, el archivo se llama .profile, .login, .cshrc, .kshrc, .bashrc, etc. Una variable estándar PATH del estilo de sh para el root podría ser: PATH=/bin:/sbin:/usr/bin:/usr/sbin.

Una variable PATH estándar para un usuario normal que ejecuta X Window podría ser:

```
PATH=/bin:/usr/bin:/usr/X11R6/bin:/usr/local/bin:.
```

/etc/issue.net

/etc/issue.net contiene el titular de login que se muestra cuando alguien intenta un login remoto. Aparece el titular, seguido de las solicitudes de nombre de usuario y de contraseña. /etc/issue contiene la misma información para los logins locales.

Uno de los problemas a la hora de ofrecer el servicio telnet es que el titular de login aparece antes de que comience el proceso de autenticación de usuario. Cualquiera puede abrir una conexión telnet a la máquina y leer la información del titular. De forma predeterminada, la mayoría de los sistemas UNIX, Linux incluido, muestran el nombre del sistema operativo y la versión, así como el tipo de CPU del equipo. Por ejemplo, el archivo issue.net de la versión Linux 6.0 de Red Hat muestra lo siguiente:

```
Red Hat Linux release 6.0 (Hedwig)
Kernel 2.2.5-15 on an i686
```

Este es exactamente el tipo de información que los programas de exploración de puerto intentan determinar. El titular de login remoto ofrece la información gratis si el servicio telnet está abierto a Internet.

Tanto /etc/issue como /etc/issue.net se crean cada vez que el sistema se inicia. Los archivos los crea la secuencia de comandos del shell /etc/rc.d/rc.local. Si se necesita ofrecer servicio de telnet remoto, será necesario modificar el

archivo /etc/rc.d/rc.local para escribir información más restringida en /etc/issue.net o para eliminar el código de rc.local y crear su propio /etc/issue.net a mano.

Por ejemplo, se puede crear un titular de login remoto sencillo escribiendo las siguientes líneas en /etc/issue.net:

```
Welcome to Bastion
%d
```

Aparecerá el siguiente mensaje cuando alguien intente una conexión remota telnet a la máquina:

```
Welcome to Bastion
13:45 on Saturday, 24 July 1999
```

Inicio de sesión remoto

Se puede configurar el demonio del registro del sistema, syslogd, para que escriba los registros del sistema en una máquina remota. Es probable que un usuario particular no necesite este nivel de complejidad añadido ni la sobrecarga de administración del sistema que implica. Un sitio que use una configuración de servidor de red parecida a la del ejemplo del Capítulo 4, donde los servicios se ofrecen desde máquinas internas en la DMZ, podría querer almacenar una copia remota de los registros del sistema. Mantener una copia remota ofrece dos ventajas: los archivos de registro se consolidan en una única máquina, facilitando al administrador del sistema la supervisión de registros, y que la información esté protegida si una de las máquinas de servidor se ve comprometida.

El Capítulo 8 explica la importancia que juegan los registros del sistema durante una recuperación si un sistema se ve comprometido. Una de las primeras cosas que hace un hacker después acceder como root a una máquina comprometida es borrar los registros del sistema o instalar programas de caballo de Troya que no registrarán sus actividades. Los archivos de registro del sistema han desaparecido o son poco fiables justo cuando son más necesarios. Mantener una copia remota de los registros ayudará a proteger esta información, al menos hasta el momento en que el hacker reemplace los demonios que escriben la información del archivo de registro.

Para registrar la información del sistema de forma remota, es necesario realizar ligeras modificaciones tanto en la configuración local del registro como en la configuración remota.

En la máquina remota que recopila los registros del sistema, se modifica la secuencia de comandos de configuración en tiempo de inicio, /etc/rc.d/init.d/syslog, y se agrega la opción -r a la llamada de syslogd. La opción -r indica a syslogd que escuche en el puerto 514 del servicio UDP esperando información entrante de registros procedente de sistemas remotos.

En la máquina local que crea los registros del sistema, se modifica el archivo de configuración de syslogd, /etc/syslog.conf, y se agregan líneas que especifican las facilidades de registro y prioridades que se desean escribir en el host remoto. Por ejemplo, la siguiente línea copia toda la información de los registros al *hostname*:

```
*.*                    @hostname
```

Mantenerse al día con actualizaciones de software

Una de las precauciones de seguridad más importantes que se pueden tomar consiste en mantener actualizado el software del sistema, especialmente los servicios de red y el propio núcleo. Esta precaución es especialmente importante para los sistemas de código abierto como Linux, donde los programadores y los hackers tienen el mismo acceso al código fuente.

Cómo obtener actualizaciones de software de Red Hat

Se pueden conseguir los parches de seguridad y las actualizaciones para la versión Red Hat de Linux en la dirección www.redhat.com, en la sección Errata de la página Support. Los vínculos web cambian a medida que se reorganizan los sitios web. El URL actual del área Errata de Red Hat es http://www.redhat.com/corp/support/errata/. Se recomienda que se visite el área Errata semanalmente para obtener actualizaciones recientes de seguridad.

Un ejemplo: explosión de mountd

En el invierno de 1998 se descubrió una vulnerabilidad de desbordamiento del búfer en el demonio mountd de NFS, que permitía a un hacker tener acceso de root al sistema. Esta versión de mountd se distribuyó con la versión 5.1 de Red Hat un poco antes de descubrir el fallo. Cuando la comunidad hacker descubrió el agujero de seguridad, se distribuyeron rápidamente en Internet programas que se aprovechaban del fallo. Red Hat, junto con otros fabricantes de Linux, publicó un parche de seguridad casi inmediatamente. Si se desea más información sobre la explosión de mountd, consulte el documento del CERT http://www.cert.org/advisories/CA-98.12.mountd.html.

Casi todo el mundo con el que he hablado y que ejecutaba mountd sin protegerlo mediante un firewall fueron objeto de los ataques de los hackers antes de que se actualizaran a la versión parcheada.

La explosión de mountd es un ejemplo de:

- La necesidad de mantenerse al día en cuanto a actualizaciones y parches de seguridad.
- La vulnerabilidad de desbordamiento del búfer.

- La ejecución de servidores públicos no destinados a Internet. Incluso si no se montaran o no se exportaran de forma pública los sistemas de archivos NFS, la debilidad residiría en el propio mountd.
- La necesidad de bloquear el acceso externo a servicios y demonios LAN.

Resumen

Aunque se instale un firewall de filtrado de paquetes, un sistema UNIX puede verse fácilmente comprometido si no se toman consideraciones de seguridad y precauciones a nivel de administración del sistema. Ningún nivel de seguridad ni ningún mecanismo ofrecen protección completa contra los sistemas comprometidos. En este capítulo se explican varias opciones de administración del sistema y de configuración del servidor que deberían usarse para ayudar a asegurar el sistema.

Unas cuantas áreas requieren la atención del administrador, incluyendo la autenticación, así como las ventajas que ofrecen las contraseñas secundarias y la contraseña MD5 de hash frente al mecanismo de contraseñas predeterminado de UNIX. Después de la autenticación, se muestran las tareas de administración del sistema centradas en la autorización. Se explica el mecanismo de empaquetamiento TCP. Se discute el peligro de los archivos y los directorios legibles públicamente, además de las herramientas para ayudar a localizar los objetos legibles públicamente en el sistema de archivos. Se explican las cuestiones de seguridad y los mecanismos específicos a unos cuantos servicios habituales de Internet, incluyendo servicios como telnet, smtp, ftp, pop y DNS.

8

Informes de incidentes y detección de intrusos

Un sistema unix correctamente configurado puede llegar a ser bastante seguro. De todos los peligros inherentes asociados con un sistema tan complicado como UNIX, las cuestiones sobre la seguridad suelen comprenderse mejor que las de otros sistemas operativos, ya que UNIX se ha usado y ha evolucionado en el mundo real durante veinte años. De hecho, ha sido la base de Internet.

Sin embargo, es posible cometer errores en la configuración. Siempre aparecerán fallos en el software. En la vida real, es necesario llegar a compromisos de seguridad. Los programadores de seguridad y los hackers están en una carrera continua para mantenerse por delante unos de los otros. Lo que es seguro hoy en día no será seguro mañana. Es prácticamente un hecho que alguna pieza del software que se ejecuta de forma segura hoy, mañana no será tan segura.

El Capítulo 7, "Problemas a nivel de administración del sistema UNIX", explica la importancia de mantener el software del sistema actualizado. En él se describe cómo justo después de que Red Hat 5.1 saliera a la calle, se encontró y se reventó un fallo en el desbordamiento del búfer del demonio mountd de Linux. Casi inmediatamente, se lanzó una actualización de seguridad. Esta no fue, ni con mucho, la primera vez que sucede algo del estilo, y no será la última.

A medida que la tecnología de los hackers se hace más sofisticada y se encuentra una debilidad en una parte del software, no se suele catalogar de forma rotunda a ese software como inseguro, ni se deja de utilizar. La vulnerabilidad se corrige y el software sobrevive. Vuelve a ser seguro, por ahora. Los

hackers buscan continuamente nuevos fallos y nuevas aproximaciones. Así que cualquier día, puede descubrirse un agujero de seguridad en la misma pieza de software. El proceso empieza de nuevo.

Me gustaría poder decir que, si se siguen todos los consejos y procedimientos de este libro, nadie se introducirá ilegalmente en un sistema, pero no es cierto. No existen garantías. La seguridad del sistema es un proceso en marcha, vivo, un sistema de vigilancia y mantenimiento de actualizaciones, que debe mantenerse un paso por delante de los hackers. El peligro siempre está presente.

Una última precaución que se debe tomar, y de forma regular, es ejecutar comprobadores de integridad del sistema (este capítulo explica tres de ellos). Son revisores de la integridad del sistema, de otros fabricantes o herramientas de detección de intrusos. Este capítulo termina con algunos síntomas habituales de ataques a la seguridad de un sistema, realiza algunas sugerencias sobre qué hacer si ocurre lo impensable y alguien se introduce en el sistema, y explica brevemente los factores que se deben tener en cuenta para tomar la decisión de informar acerca de un incidente.

Comprobadores de integridad del sistema

Los comprobadores de integridad del sistema se especializan en análisis, auditorias y comprobaciones de vulnerabilidad. Algunos de ellos sirven como herramientas de análisis y seguridad para ayudar a ajustar la configuración del sistema. Otros sondean activamente en busca de posibles fallos conocidos. Otros comparan el sistema actual con un estado anterior conocido, comprobando la existencia de cambios no autorizados o dudosos. Estas herramientas se pueden conseguir sin ningún problema en almacenes de software Linux, en sitios sobre seguridad web y ftp, proyectos de seguridad de universidades fundados por el DARPA y sitios relacionados con la seguridad del gobierno.

COPS

El Computer Oracle and Password System (COPS) de Dan Farmer es una familia de programas que revisan de forma conjunta un gran conjunto de posibles áreas de debilidad de seguridad en un sistema UNIX. cops se ha pensado para ejecutarlo de forma periódica como una tarea cron. Los resultados se escriben en un archivo informe que, opcionalmente, se puede enviar por correo a un usuario específico, normalmente a la cuenta de root.

Entre las comprobaciones que realiza cops se encuentra la comprobación de permisos de acceso a archivos y directorios, calidad de la contraseña, configuración ftp, suma de comprobaciones de archivos ejecutables clave, acceso tftp, configuración sendmail, configuración del archivo inetd.conf y el estado de los diferentes archivos de configuración del directorio /etc.

El paquete COPS se puede conseguir en muchos sitios. Dos fuentes son: ftp://info.cert.org/pub/tools/cops/cops.1.04.tar.gz y http://metalab.unc.edu/pub/Linux/system/security/cops_104_linux.tgz.

Como COPS se pensó como una herramienta de análisis de seguridad para los sistemas UNIX generales, los archivos de configuración y ejecutables de COPS deben personalizarse para cada tipo y versión de Linux.

Crack

Crack es un programa de adivinación de contraseñas independiente. También incluye parte de las funciones que se encuentran en otro software de comprobación de integridad. La herramienta está diseñada para ayudar a los administradores del sistema a identificar las cuentas de usuario que tienen contraseñas débiles, que se pueden adivinar fácilmente. Usa una combinación de diccionarios generales y especializados y varios algoritmos heurísticos para probar diferentes combinaciones de modelos. El programa Crack ofrece compatibilidad incorporada para mezclar los archivos passwd y shadow, si se usan las contraseñas de sombra. Crack usa el esquema de cifrado DES en el que se han basado históricamente las contraseñas UNIX. Durante la instalación, la versión Linux 6.0 de Red Hat ofrece la opción de usar contraseñas basadas en MD5. Crack no soporta actualmente MD5 sin modificaciones.

Se puede conseguir el programa Crack en la dirección ftp://info.cert.org/pub/tools/crack/crack5.0.tar.gz.

ifstatus

ifstatus comprueba las configuraciones de la interfaz de red del sistema. Se informa de cualquier interfaz que se encuentre en modo depuración o con excesiva actividad. El que las interfaces de red se encuentren en estos estados puede ser un signo de que un intruso ha accedido al sistema y está usando un rastreador de paquetes de forma local, normalmente para leer las contraseñas que están en texto sin cifrar, como las que pasa el programa telnet sobre una red.

Se puede conseguir la herramienta ifstatus en la dirección ftp://coast.cs.purdue.edu/pub/tools/unix/ifstatus.

También se puede encontrar información adicional sobre problemas relacionados con el rastreo de paquetes en la sección "Ongoing Network Monitoring Attacks" (Supervisión de ataques de red en curso) en la dirección http://www.cert.org/advisories/CA-94.01.ongoing.network.monitoring.attacks.html.

MD5

MD5 es un algoritmo de cifrado de suma de comprobación que se usa para asegurar la integridad de los datos. MD5 lee una cadena o un archivo y genera una suma de comprobación de 128 bits. La versión Red Hat de Linux incluye ahora un programa de suma de comprobación MD5 y bibliotecas compatibles para programar en C. Algunos de los paquetes para integridad

del sistema que se presentan aquí incluyen sus propias bibliotecas MD5 para crear bases de datos de sumas de comprobación de los archivos de sistema seleccionados.

SATAN

La Security Administrator Tool for Analyzing Networks (SATAN), de Wietse Venema y Dan Farmer, es una herramienta que ayuda a identificar fallos de seguridad en las configuraciones de servicios de red. SATAN comprueba problemas comunes relacionados con NFS, NIS, tftp, ftp, y los comandos remotos BSD. Además, realiza un examen de puerto para identificar puertos abiertos. Produce informes muy largos y tutoriales que describen los problemas encontrados y las posibles soluciones a éstos. SATAN necesita algo de ayuda por parte del usuario para conseguir que funcione en un sistema Linux, pero la calidad y la amplitud de los informes hace que el esfuerzo valga la pena. La interfaz de usuario de SATAN se ha mejorado para ofrecer una interfaz de navegador web.

Si desea información más detallada sobre SATAN, visite la dirección ftp://ftp.porcupine.org/pub/security/index.html. Se puede conseguir la herramienta SATAN en muchos sitios, incluyendo los siguientes: ftp://ftp.porcupine.org/pub/security/satan-1.1.1.tar.Z, ftp://ftp.net.ohio-state.edu/pub/security/satan/ y ftp://sunsite.unc.edu/pub/packages/security/Satan-for-Linux/.

tiger

tiger es una colección de secuencias de comandos y programas en C diseñados para comprobar posibles fallos de seguridad que pueden permitir el acceso no autorizado de root. tiger comprueba los valores de la configuración del sistema para variables PATH, inetd.conf, sistemas de archivos exportados NFS, nombres de archivo no usuales, permisos de archivos y directorios y archivos .rhost. tiger mantiene una base de datos de firmas digitales para archivos de sistema claves, con el fin de comprobar si se han modificado fraudulentamente. Se puede conseguir en la siguiente dirección: http://metalab.unc.edu/pub/Linux/system/security/tiger-2.2.4.tgz y http://www.net.tamu.edu/ftp/security/TAMU/tiger-2.2.4p1.tar.gz, así como en muchas otras.

tripwire

tripwire crea y mantiene una base de datos de firmas digitales MD5 para todos o un conjunto de archivos y directorios del sistema. Su propósito es detectar adiciones, eliminaciones o cambios en archivos no autorizados. Se puede conseguir muy fácilmente la última versión de tripwire, la 1.2, en las siguientes direcciones: http://www.cert.org/ftp/tools/tripwire/tripwire-1.2.tar.Z y ftp://ftp.auscert.org.au/pub/coast/COAST/Tripwire/tripwire-1.2.tar.Z.

tripwire es ahora un producto comercial que se puede conseguir de la mano de Tripwire Security Systems, Inc. en la dirección http://www.tripwiresecurity.com/. La versión comercial 2.0 de Tripwire es gratuita para la versión Red Hat de Linux.

Síntomas que sugieren que el sistema puede estar comprometido

Los cambios inesperados que aparecen en los informes que producen las herramientas de auditoría mencionados en la última sección, especialmente las firmas digitales que no coinciden y los cambios en los permisos de archivos y directorios, son indicaciones claras de que el sistema puede estar comprometido. A menudo, el hacker que consiga introducirse en nuestro sistema intentará ocultar sus huellas. El hacker no quiere que se detecte su presencia. Su equipo es la nueva base de operaciones del hacker. Afortunadamente, el hacker está en un sistema nuevo, y los hacker también cometen errores. Sin embargo, la vigilancia continua es lo que se debe hacer. El hacker puede tener más nivel a la hora de ocultar sus huellas que el usuario a la hora de hacer un seguimiento de los estados anómalos del sistema.

Los sistemas UNIX son muy diversos, personalizables y complicados como para definir una lista detallada, con los síntomas definitivos que prueben que el sistema está comprometido. Como con cualquier clase de trabajo relacionado con la detección o diagnostico, se deben buscar las pistas donde se pueda y de forma tan sistemática como sea posible. La RFC 2196, "*Site Security Handbook*", ofrece una lista de síntomas que se deberían revisar. La "*Intruder Detection Checklist*", que se puede conseguir del CERT en la dirección http://www.cert.org/ftp/tech_tips/intruder_detection_checklist, ofrece otra lista de anomalías que se deberían revisar. *"Steps for Recovering from a UNIX Root Compromise"*, que también se puede conseguir en el CERT en la dirección http://www.cert.org/tech_tips/root_compromise.html, ofrece una tercera lista de elementos a revisar.

Las siguientes secciones explican las tres listas, incluyendo todos o la mayoría de sus puntos de una forma u otra. Se ha realizado una sencilla ordenación de las anomalías del sistema en las siguientes categorías: indicaciones relacionadas con el registro del sistema; cambios en la configuración del sistema; cambios relacionados con el sistema de archivos; contenido de los archivos; permisos de acceso a archivos y tamaño de los archivos; cambios en las cuentas de usuario, contraseñas y permisos de usuario; problemas que aparecen en los informes de las auditorias, y degradación inesperada del rendimiento. Las indicaciones anómalas suelen cruzar las fronteras entre varias categorías.

Indicaciones del registro del sistema

Las indicaciones de los registros del sistema incluyen mensajes no usuales de error y de estado en los registros del sistema, archivos de registro trunca-

dos, archivos de registro eliminados e informes de estado enviados por correo electrónico:

- Archivos de registro del sistema. Las entradas sin explicación que aparecen en los archivos de registro del sistema, la reducción de los archivos de registro y la pérdida de éstos, indican que algo anda mal. /var/log/auth contiene un registro de todos los accesos con cuenta. /var/log/secure contiene los accesos con cuenta privilegiada. /var/log/maillog registra todas las conexiones de correo. /var/log/xferlog registra las transferencias de archivos ftp y uucp, si se ha configurado. /var/log/messages contiene la mayoría de la información histórica del sistema.
- Informes de estado de los demonios del sistema. En lugar, o además, de escribir los archivos de registro, algunos demonios, como crond, envían informes de estado por correo electrónico. Los informes inusuales o desaparecidos sugieren que algo no va bien.
- Mensajes de consola y de terminal anómalos. Los mensajes sin explicación durante un inicio de sesión, que anuncian la posible presencia del hacker, son claramente sospechosos.
- Intentos de acceso repetidos. Los intentos de acceso activos o los intentos de acceder de forma ilegal a los archivos a través de ftp o de un servidor web, particularmente los intentos para modificar los guiones CGI, son sospechosos si son persistentes, incluso cuando parezcan terminar en errores repetidos.

Los programas de supervisión de archivos de registro que se mostraron en el Capítulo 5, "Depuración de las reglas del firewall", pueden ayudar a alertar o realizar alguna otra acción en tiempo real.

Indicaciones de la configuración del sistema

Las indicaciones de la configuración del sistema incluyen archivos de configuración y secuencias de comandos del sistema modificadas, procesos no planeados que se ejecutan de forma inexplicable, uso y asignaciones de los puertos de servicio no planeados y cambios en el funcionamiento de los dispositivos de red:

- Tareas cron. Comprueba posibles modificaciones en las secuencias de comandos de configuración de cron y los ejecutables en busca de modificaciones.
- Archivos de configuración del sistema modificados. Una comprobación de una firma digital puede indicar archivos de configuración cambiados en el directorio /etc. Estos archivos son muy importantes para que el sistema funcione de forma correcta. Es importante revisar cualquier cambio que se produzca en un archivo (como /etc/inetd.conf, /etc/named.conf y en sus archivos de bases de datos DNS en el directorio

/var/named, /etc/passwd, /etc/group, /etc/hosts.equiv, o cualquier archivo de configuración de exportación del sistema de archivos de red).

- Servicios y procesos sin explicación según los muestra ps. Los programas en ejecución inesperados son un una mala señal.
- Conexiones y uso de puertos inesperados según los muestra netstat. El tráfico inesperado de red es una muy mala señal.
- Caídas del sistema y procesos desaparecidos. Las caídas de sistema, así como las caídas inesperadas del servidor, pueden ser sospechosas.
- Cambios en la configuración de los dispositivos. Volver a configurar una interfaz de red que se encuentre en modo depuración, o con demasiada actividad, es un signo de que se ha instalado un rastreador de paquetes.

Indicaciones del sistema de archivos

Las indicaciones del sistema de archivos incluyen nuevos archivos y directorios, archivos y directorios desaparecidos, contenidos de archivo modificados, firmas digitales MD5 no coincidentes, nuevos programas setuid y sistemas de archivos que crecen o se desbordan rápidamente:

- Nuevos archivos y directorios. Además de las firmas digitales erróneas que aparecen de repente, pueden aparecer nuevos archivos y directorios. Son especialmente sospechosos los nombres de archivo que empiezan por uno o más puntos y que suenan como legales, pero aparecen en sitios incorrectos.
- Programas setuid y setgid. Los nuevos archivos setuid y los archivos setuid que se acaban de configurar, son un buen sitio donde empezar a buscar problemas.
- Archivos desaparecidos. Los archivos desaparecidos, particularmente los archivos de registro, indican que existe algún problema.
- Los sistemas de archivos que cambian de tamaño rápidamente, según los muestra df. Si la máquina está comprometida, los sistemas de archivos que crecen rápidamente pueden ser signo de la existencia de un programa de supervisión de un hacker que produce grandes archivos históricos.
- Ficheros de archivos públicos modificados. Se debe comprobar el contenido de las áreas Web y ftp en busca de archivos nuevos o modificados.
- Archivos nuevos en el directorio /dev. El CERT alerta especialmente para que se revise la presencia de nuevos archivos ASCII en el directorio /dev, que suelen ser archivos de configuración de programas Troyanos.

Indicaciones de las cuentas de usuario

Las indicaciones de las cuentas de usuario incluyen nuevas cuentas de usuario, cambios en el archivo passwd, actividad poco usual en el proceso de

usuario de informes de cuentas o pérdida del proceso de informes de cuentas, cambios en los archivos de usuario, especialmente archivos de entorno, y pérdida de acceso de cuenta:

- Cuentas de usuario nuevas y modificadas. La aparición de nuevas cuentas en /etc/passwd y los procesos que se ejecutan con identidades de usuario nuevas o inesperadas, según muestra ps, son claras indicaciones de la existencia de nuevas cuentas. Las cuentas con contraseñas que se pierden repentinamente indican que se ha abierto una cuenta.
- Registros de cuentas de usuarios. Los informes con cuentas de usuario no usuales, inicios de sesión inexplicables, archivos de registros perdidos o modificados (como /var/log/lastlog, /var/log/pacct o /var/log/usracct) y la actividad irregular de usuario son claros indicios de problemas.
- Cambios en las cuentas de root o de usuarios. Una clara señal de problemas es si se modifica un entorno de inicio de sesión de usuario, o se daña hasta el punto que la cuenta se hace inaccesible. Son especialmente importantes los cambios en los archivos de usuario .rhost y .forward y los cambios en la variable de entorno PATH.
- Pérdida de acceso de cuenta. La denegación de acceso de forma intencionada se parece a los cambios en el entorno de inicio de sesión de un usuario, ya sea cambiando la contraseña de la cuenta o eliminando la cuenta o, para usuarios normales, cambiando el nivel de ejecución a modo monousuario.

Indicaciones de las herramientas de auditoría de seguridad

Las indicaciones de las herramientas de auditoría de seguridad incluyen firmas digitales que no coinciden, cambios en el tamaño de los archivos, cambios en los bit de modo de permiso de archivo y nuevos programas setuid y setgid.

Los archivos con firmas digitales no coincidentes pueden ser archivos nuevos, archivos cuya fecha de creación o modificación haya cambiado y archivos cuyos modos de acceso se han modificado. Son especialmente interesantes los programas de caballo de Troya instalados recientemente. Es frecuente que sea necesario reemplazar los programas que se ejecutan desde /etc/inetd.conf, ls, ps, netstat, ifconfig, telnet, login, su, ftp, inetd, syslogd, du, df, sync y libc.

Indicaciones del rendimiento del sistema

Las indicaciones del rendimiento del sistema incluyen medias de carga inusualmente altas y un intenso acceso a disco.

Un rendimiento pobre del sistema, sin explicación aparente, puede deberse a una actividad poco usual de los procesos, promedios de carga actuales inusualmente altos, tráfico de red excesivo o un intenso acceso al sistema de archivos.

Si el sistema muestra signos evidentes de estar comprometido, no debe cundir el pánico. No reinicie el sistema. Podría perder información importante. Simplemente desconecte físicamente el sistema de Internet.

¿Qué hacer si el sistema está comprometido?

Tanto las instrucciones "*Steps for Recovering from a UNIX Root Compromise*", (Pasos para recuperarse de una raíz UNIX comprometida), que puede conseguir en el CERT en la dirección http://www.cert.org/tech_tips/root_compromise.html, como la RFC 2196, "*Site Security Handbook*", (Manual de seguridad de sitio), explican procedimientos a seguir en el caso de una violación de la seguridad del sistema. Estos documentos muestran procedimientos más formales que los que pueden seguir una empresa, una oficina del gobierno o las universidades. Los procedimientos suponen que se dispone de cierta cantidad de espacio de almacenamiento, además de tomar instantáneas del sistema. También suponen que se dispone de plantilla suficiente para analizar y diagnosticar el problema de seguridad y explican situaciones en las que la víctima puede que quiera iniciar alguna acción legal.

Para los sitios pequeños que acceden a Internet a través de un ISP público orientado al consumidor, lo normal es lo siguiente: un sitio se ve comprometido sin que el propietario sea consciente, el sitio se usa como una nueva base de operaciones para lanzar ataques contra otros sitios, alguien se queja al ISP y el cliente se da cuenta de la intrusión cuando el ISP le llama y desconecta el servicio de Internet del usuario. Cuando el cliente convence al ISP de que el sistema ha sido comprometido y que ha arreglado el problema, el ISP normalmente restaura el servicio del usuario. Pueden pasar varias semanas.

Si sucede lo peor, ¿qué hacer si se da cuenta de que alguien ha entrado en el sistema? De nuevo, lo primero que se debe hacer es desconectar la máquina de Internet.

En ningún caso reinicie el sistema. Si el hacker instaló o inició programas manualmente, reiniciar el sistema destruirá la información de estado del sistema.

Si se dispone de espacio de almacenamiento, debe realizarse una instantánea de todo el sistema en su estado actual para analizarla posteriormente. Si esto no es posible, al menos debe realizarse una instantánea de los registros del sistema del directorio /var/log y de los archivos de configuración del directorio /etc.

Se debe conservar un registro. Se debe anotar todo. Documentar todo lo que se hace y todo lo que se encuentra no es sólo para informar del incidente a un equipo de respuesta, al ISP o a un abogado. La documentación también ayuda a mantener un registro de lo que ha examinado y lo que queda por hacer.

Los pasos a seguir para determinar si el sistema está comprometido son los mismos pasos que se siguen para analizar el sistema comprometido:

1. Compruebe los registros del sistema, qué procesos se están ejecutando y a qué puertos están conectados. Revise el contenido de los

archivos de configuración del sistema. Verifique el contenido de los modos de acceso de los archivos y directorios comprobando las firmas digitales. Busque nuevos programas setuid. Compare los archivos de configuración y de usuario con las copias de seguridad limpias. Es muy probable que el hacker instale programas de caballo de Troya en lugar de las propias herramientas del sistema que se usan para analizar el sistema.

2. Almacene cualquier información volátil, como los procesos que se están ejecutando y los puertos que están en uso.
3. Arranque desde un disco de inicio o una copia de seguridad del sistema. Examine el sistema usando las herramientas de limpieza de un sistema no afectado.
4. Determine cómo entró en el sistema el hacker y qué ha hecho en el sistema.
5. Vuelva a instalar completamente el sistema desde el medio de distribución original de Linux.
6. Corrija el fallo de seguridad, haciendo una selección más cuidadosa de los servicios a ejecutar, volviendo a configurar los servicios de forma más segura, definiendo listas de acceso en el nivel tcp_wrapper y a un nivel de servidor individual, instalando un firewall de filtrado de paquetes o instalando servidores proxy de aplicación.
7. Habilite todos las entradas en el registro.
8. Restaure los archivos de usuario y de configuración especiales que sepa que no están contaminados.
9. Instale cualquier actualización de seguridad nueva que ofrezca el proveedor de Linux. Instale y configure los paquetes de integridad del sistema. Cree sumas de comprobación MD5 para los binarios instalados recientemente y almacene la base de datos de suma de comprobación en un disquete o en algún otro sistema.
10. Supervise el sistema para los intentos de acceso repetitivos del hacker.

Casi todas las personas con las que he hablado se sienten culpables y como unos tontos después de que se haya comprometido el sistema. Recuerde que la seguridad es una batalla diaria y que requiere gran agudeza entre los hacker y los administradores de sistemas. No se ha invitado al hacker a entrar. Realizan un esfuerzo consciente, probablemente un esfuerzo muy constante, para descubrir los fallos del sistema y aprovecharse de ellos. No se haga la víctima. No está solo. Hay muchos sistemas comprometidos. Simplemente intente estar un poco más alerta ante lo que hacen los chicos malos.

Información de incidentes

Un incidente puede ser varias cosas y debe definirlo el propio usuario. En términos más generales, es cualquier acceso anómalo desde Internet. En tér-

minos más estrictos, un incidente es un acceso de inicio de sesión anómalo con éxito, un ataque por denegación de servicio con éxito, explotar la debilidad del sistema con éxito y usurpar con éxito los servicios y los recursos del sistema.

Como administrador de una maquina conectada a Internet, debería supervisar los archivos de registro del sistema, los informes de integridad del sistema y los informes de cuentas del sistema como algo habitual. Incluso con un mínimo registro habilitado, antes o después se verá algo que se considera que es importante denunciar. Si se han habilitado todas las entradas del registro, habrá tantas entradas como para estar reflexionando las veinticuatro horas del día.

Algunos intentos de acceso son más serios que otros, y algunos molestarán personalmente más que otros.

Las siguientes secciones comienzan explicando las razones por las que se podría querer denunciar un incidente, y las consideraciones sobre los tipos de incidentes que se pueden denunciar. Estas decisiones son individuales. Si se decide denunciar algo, las secciones restantes se centran en los diferentes grupos de denuncia que existen para que se elija entre ellos y el tipo de información que se debe proporcionar.

¿Por qué informar de un incidente?

Puede que se quiera denunciar un incidente incluso si el intento del hacker no tuvo éxito. Estas son algunas de las razones:

- Para finalizar los sondeos. El firewall asegura que la mayoría de los sondeos son inofensivos. Incluso los sondeos inofensivos son molestos si tienen lugar periódicamente. Los constantes exámenes activos y repetidos llenan los archivos de registro. Dependiendo de cómo se hayan definido los disparadores de notificación en cualquier software de supervisión de registros que se ejecute, los sondeos repetidos pueden ser molestos, originando notificaciones continuas por correo electrónico.
- Para ayudar a proteger otros sitios. Los sondeos y los exámenes automatizados suelen generar una base de datos de todos los equipos vulnerables en un gran bloque de direcciones IP. Cuando se identifican como potencialmente vulnerables a explosiones específicas, estos equipos se convierten en el objetivo de ataques selectivos. Las actuales herramientas sofisticadas de piratería pueden comprometer un sistema vulnerable y ocultar su rastro en cuestión de segundos. La denuncia de un incidente puede frenar las exploraciones, antes de que alguien salga herido en algún lugar.
- Para informar al administrador del sistema o de la red. Los sitios atacantes suelen ser sistemas comprometidos, albergan una cuenta de usuario comprometida, tienen software mal configurado, están siendo sometidos a usurpamiento de direcciones o tienen una persona que

genera problemas. Los administradores de sistemas suelen ser receptivos a la denuncia de un incidente. Los ISP tienden a detener a los clientes que generan problemas antes de que otros clientes empiecen a quejarse de que los sitios remotos han bloqueado el acceso desde su bloque de direcciones y no puedan intercambiar correo con un amigo o con la familia en un sitio remoto.

- Para recibir una confirmación del ataque. A veces simplemente se querrá una confirmación de si lo que aparece en los registros es problemático o no. Quizá se quiera una confirmación de si un sitio remoto está realmente filtrando paquetes de forma no intencionada debido a una configuración errónea. El sitio remoto también agradece que se le informe de que la red no funciona como se esperaba.
- Para aumentar la conciencia y supervisión por todas las partes involucradas. Si se denuncia el incidente al sitio atacante, el sitio supervisará con más cuidado la configuración y las actividades de los usuarios. Si se denuncia el incidente a un centro de informes de ataques, la plantilla del centro puede contactar con el sitio remoto con más fuerza que la que tiene una sola persona, estar pendiente de la actividad de forma continuada y ayudar mejor a los clientes que han sido comprometidos. Si se denuncia el incidente a un grupo de noticias sobre seguridad, otras personas pueden tener una idea mejor sobre lo que se debe vigilar.

¿De qué tipo de incidentes debería informar?

El tipo de incidentes sobre los que se debe informar depende completamente de la tolerancia del usuario, de cómo de serios considera los diferentes sondeos y de cuánto tiempo se dedique a lo que puede ser una infección global y que crece de forma exponencial. Depende de la definición del término "incidente". Para diferentes personas, los incidentes varían desde simples exploraciones de puerto hasta intentos de acceder a sus archivos privados o recursos del sistema, de ataques por denegación de servicio hasta hacer caer los servidores o todo el sistema y poder acceder al sistema con privilegios de root:

- Ataques por denegación de servicio. Cualquier clase de ataque por denegación de servicio es intencionadamente hostil. No resulta difícil realizar personalmente un ataque como éste. Estos ataques son la forma electrónica de vandalismo, obstrucción y robo de servicio. Algunas formas de ataque por denegación de servicio son posibles debido a la naturaleza inherente de los dispositivos de red, por lo que poco o nada se puede hacer sobre algunas formas de ataque, aparte de denunciar los incidentes y bloquear todo el conjunto de direcciones del atacante.
- Intentos de volver a configurar el sistema. Un hacker no puede volver a configurar los servidores sin una cuenta de inicio de sesión de root en la máquina, pero sí puede modificar el sistema cuando las tablas re-

lacionadas con la red se encuentran en la memoria del sistema, o al menos intentarlo. Las explosiones que debe tener en cuenta son:

- Transferencias de zona DNS no autorizadas a o desde la máquina del usuario, sobre TCP. Si se desea más información sobre transferencias de zona, consúltese **DNS & BIND**, de Albitz y Liu (O'Reilly).
- Cambios en las tablas de enrutamiento dinámicas mediante ICMP Redirect o sondeos al puerto UDP 520 para routed o gated. (Recuerde, una máquina firewall no debería soportar enrutamiento dinámico). Si desea más información sobre las explosiones de las tablas de enrutamiento, consulte **Firewalls and Internet Security: Repelling the Wily Hacker,** de Cheswick y Bellovin (Addison-Wesley).
- Intentos de volver a configurar las interfaces de red o las tablas de enrutamiento mediante sondeos al puerto UDP 161 para snmpd.
- Intentos de conseguir información de configuración local y de la topología de la red. Las peticiones de información de red se dirigen principalmente al puerto UDP 161 por snmpd. Las peticiones DNS sobre el puerto TCP 53 ofrecen información de la topología de red, así como las peticiones de enrutamiento al puerto UDP 520 para routed o gated.
- Intentos de acceder al registro de las cuentas. Los sondeos a los puertos telnet TCP 23 y al puerto ssh TCP 22 son obvios. Menos obvios son los sondeos a puertos asociados con servidores bien conocidos como explotables, ya sea histórica o actualmente. Las explosiones por desbordamiento de búfer son generalmente intencionadas y, últimamente, para ejecutar comandos y acceder al shell. La explosión mountd es un ejemplo de esto.
- Intentos de acceder a archivos no públicos. Los intentos de acceder a archivos no públicos, como el archivo /etc/passwd, a los archivos de configuración o a archivos propietarios, se ponen de manifiesto en el registro de FTP (/var/log/xferlog o /var/log/messages) y en el registro de acceso del servidor web (/var/log/httpd/error_log).
- Intentos de usar servicios privados. Por definición, cualquier servicio que no se haya puesto a disposición en Internet es privado. Nos referimos a los servicios privados potencialmente disponibles a través de los servidores públicos, como intentar enviar correo a través de su servidor de correo. Lo más probable es que la gente esté tramando algo malo si quieren usar su máquina para enviar correo en lugar de hacerlo desde sus propios ISP. Los intentos de transmisión se ponen de manifiesto en el archivo de registro del correo (/var/log/maillog).
- Intentos de almacenar archivos en su disco. Si se alberga un sitio FTP anónimo configurado incorrectamente, es posible que alguien cree un almacén de software robado (WAREZ) en la máquina. Los intentos de cargar archivos se registran en el registro de FTP (/var/log/xferlog), si se configura ftpd para registrar las cargas de archivos.
- Intentos de hacer caer el sistema o servidores individuales. Los intentos de desbordamiento del búfer contra los guiones CGI disponibles a través del sitio web son posiblemente los más fáciles de identificar me-

diante los mensajes de error escritos en los archivos de registro del guión CGI. Otros informes de datos erróneos aparecerán en el archivo syslog general (/var/log/messages), en el registro del demonio general (/var/log/daemon), en el registro del correo (/var/log/maillog), en el registro de FTP (/var/log/xferlog) o en el registro de acceso seguro (/var/log/secure).

- Intentos de explosión de fallos concretos, explotables y conocidos actualmente. Los hacker encuentran nuevos fallos en cada versión de software que sale al mercado. Debe mantenerse actualizado con las últimas noticias que se pueden encontrar en www.cert.org y en el proveedor del software Linux. En los últimos años, los servicios que más se han explotado han sido sunrpc/mount, pop3, imap, socks, ftp, fallos en los guiones CGI, repetición de correo smtp y los comandos remotos BSD.

¿A quién informar del incidente?

Se ofrecen varias opciones a la hora de decidir a quién denunciar un incidente:

- root, postmaster o abuse en el sitio que comete la infracción. El sitio más normal donde interponer una reclamación es el administrador del sitio que comete la infracción. Denunciar ante el administrador del sistema puede que sea lo necesario para corregir el problema. Sin embargo, esto no es siempre posible. Muchos sondeos se originan desde direcciones IP usurpadas, no existentes. Un nslookup en la dirección origen devuelve un mensaje indicando que el equipo o el dominio no existen.
- Coordinador de red. Si la dirección IP no se resuelve, a menudo ayuda contactar con el coordinador de ese bloque de direcciones de red. El coordinador puede contactar con el administrador del sitio que comete la infracción y además, ponerle en contacto directo con él. Si la dirección IP no se resuelve a través de nslookup, casi siempre puede informar al coordinador de red proporcionando la dirección a las bases de datos whois. El comando whois es interno a la base de datos ARIN. Las cuatro bases de datos están disponibles a través de la Web:
- ARIN. El American Registry for Internet Numbers mantiene la base de datos de direcciones IP para América. El ARIN se encuentra en http://whois.arin.net/whois/arinwhois.html.
- APNIC. El Asia Pacific Network Information Centre mantiene la base de datos de direcciones IP para Asia. El APNIC se encuentra en http://www.apnic.net/reg.html.
- RIPE. El Réseaux IP Européens mantiene la base de datos de direcciones IP para Europa. El RIPE se encuentra en http://www.ripe.net/db/whois.html.

- NIRPNET. El Department of Defense Whois Database mantiene la base de datos de direcciones IP para redes militares. El NIRPNET se encuentra en http://nic.mil/cgi-bin/whois.
- El centro de ataques del ISP. Si las exploraciones se originan dentro del espacio de direcciones del ISP, se debe contactar con el centro de ataques. El ISP también puede ser de ayuda con los exámenes originados en cualquier otro lugar, contactando con el sitio que cometió la infracción en su nombre. Lo más probable es que su máquina no sea la única que esté siendo sondeada en al red del ISP.
- CERT. El CERT Coordination Center no es probable que tenga los recursos necesarios para responder a incidentes generales y corrientes. Las prioridades del CERT se centran más en los aspectos globales, en las grandes instituciones y en las emergencias de seguridad de Internet. Sin embargo, el CERT agradece los esfuerzos por ofrecer información de denuncias de incidentes, con el fin de registrarlas y usarlas con fines estadísticos. Se puede contactar con el CERT en la dirección http://www.cert.org/contact_cert/contactinfo.html, o mediante correo electrónico en la dirección cert@cert.org.
- La dirección Linux. Si el sistema se ve comprometido debido a un fallo software de la versión, el fabricante querrá saberlo para que se pueda preparar y distribuir una actualización de seguridad.

¿Qué información proporciona?

Una denuncia de incidente debe contener suficiente información como para ayudar al equipo de respuesta de incidentes a localizar el problema. Cuando se contacte con el sitio que originó el ataque, es necesario recordar que la persona de contacto puede ser el individuo que intencionadamente lanzó el ataque. Lo que se debe incluir, además de lo que aparece en la siguiente lista, depende de con quién contacte o de lo cómodo que se sienta incluyendo la información:

- La dirección de correo electrónico.
- El número de teléfono, si procede.
- La dirección IP, el nombre del equipo y el nombre de dominio.
- Las direcciones IP y los nombres de host, si es posible, implicados en el ataque.
- La fecha y la hora del incidente (debe incluirse la zona horaria de forma relativa al GMT).
- Una descripción del ataque.
- Cómo se detectó el ataque.
- Entradas del archivo de registro representativas que muestren el incidente.
- Una descripción del formato del archivo histórico.

- Referencias a avisos y noticias de seguridad que describan la naturaleza e importancia del ataque.
- Qué se quiere que haga la persona (por ejemplo, arreglarlo, confirmarlo, explicarlo, supervisarlo o informarle de ello).

¿Dónde se puede encontrar más información?

A medida que los usuarios dan cada vez más importancia a la seguridad de las redes, el número de sitios web relacionados con la seguridad crece rápidamente. CERT y COAST siguen siendo los sitios más antiguos de todos los almacenes de información y herramientas de seguridad. Varios departamentos del gobierno federal también ofrecen gran cantidad de información de seguridad sobre la Web. A continuación se muestra una lista de algunos de los sitios más conocidos, los más grandes, los más antiguos y los sitios web de seguridad oficiales que se pueden visitar:

- http://www.cert.org. CERT Coordination Center: Carnegie-Mellon Software Engineering Institute. El CERT mantiene consultas, noticias, listas de correo, documentos de seguridad, almacenes de software relacionados con la seguridad e informes anuales y de largos periodos acerca de la actividad de los hacker en Internet.
- http://www.first.org. FIRST: Forum of Incident Response and Security Teams. El FIRST mantiene listas de miembros de equipo de respuestas a incidentes para organizaciones, universidades y el gobierno.
- http://www.cerias.purdue.edu. CERIAS: Center for Education and Research in Information Assurance and Security. CERIAS incorpora el proyecto COAST (http://www.cs.purdue.edu/coast). COAST alberga herramientas relacionadas con la seguridad, papeles, guías, informes y estándares.
- http://ciac.llnl.gov. CIAC: Computer Incident Advisory Capability. CIAC proporciona asistencia técnica y servicios de respuesta a incidentes a los sitios del Departamento de energía (DOE, Department of Energy). Sin embargo, el CIAC también alberga boletines, tutoriales, herramientas de software de seguridad y enlaces a otros sitios relacionados con la seguridad.
- http://csrc.nist.gov. NIST: National Institute of Standards and Technology. NIST alberga un Computer Security Resource Clearinghouse, que incluye numerosas publicaciones relacionadas con la seguridad.
- http://www.fedcirc.gov. FedCIRC: Federal Computer Incident Response Capability. Proporciona asistencia técnica y servicios de respuesta a incidentes a los sitios de equipos del gobierno federal. Sin embargo, el FedCIRC también alberga consultas, noticias de vulnerabilidad, documentos de seguridad, vínculos a sitios relacionados y herramientas software de seguridad.

Resumen

Este capítulo se centra en la supervisión de la integridad del sistema y la detección de intrusos. Se presentan herramientas de análisis de otros fabricantes, que se pueden conseguir en muchos sitios de seguridad web y ftp, incluyendo herramientas como COPS, crack, SATAN, tiger y tripwire. Se muestra la importancia de las firmas digitales, como las que emiten MD5 y tripwire. Si se sospecha que el sistema puede estar comprometido, se debe presentar una lista de indicaciones de posibles problemas. Si aparece alguno de estos indicios y se concluye que el sistema está comprometido, se explica una lista de pasos de recuperación. Por último, se muestran las consideraciones sobre la denuncia de incidentes, junto con las direcciones donde se debe denunciar un incidente.

IV

Apéndices

A Recursos de seguridad.
B Ejemplos de firewalls y secuencias de comandos compatibles.
C Glosario.

A

Recursos de seguridad

Este apéndice lista las fuentes más habituales de noticias relacionadas con la seguridad, información, herramientas, actualizaciones y parches que se encuentran actualmente en Internet. Existen muchos otros sitios, y otros sitios surgen a diario, por lo que se debe considerar esta lista como punto de referencia, no como una lista completa.

Fuentes de información

Se puede encontrar información de seguridad de todos los tipos, noticias y alertas, libros blancos, tutoriales, etc., en los siguientes sitios:

- Centro de coordinación CERT:
  http://www.cert.org/
- CIAC (Computer Incident Advisory Capability):
  http://ciac.llnl.gov/ciac/
- Recursos de firewalls de Internet COAST:
  http://www.cs.purdue.edu/coast/firewalls/
- Archivo de seguridad de COAST:
  ftp://coast.cs.purdue.edu/pub/
- Página de seguridad de Dave Dittrich:
  http://www.washington.edu/People/dad/
- Archivo de lista de correo de asistentes para firewalls:
  http://www.nfr.net/firewall-wizards/fwsearch.html

- Internet Engineering Task Force (IETF):
  http://www.ietf.cnri.reston.va.us/home.html
- Sitio de seguridad y firewalls de Linux:
  http://linux-firewall-tools.com/linux
- Recursos de seguridad de Linux:
  http://www.linux-security.org/
- Seguridad para UNIX de Matt:
  http://www.deter.com/unix/
- Información de seguridad de equipos NIH:
  http://www.alw.nih.gov/Security/
- Página de seguridad de Red Hat:
  http://www.redhat.com/LinuxIndex/Administration/Security/
- Instituto SANS:
  http://www.sans.org/

Colecciones de software

Los siguientes sitios están especializados en mantener almacenes de software de todo tipo relacionado con la seguridad:

- Archivo CERT:
  http://www.cert.org/ftp/tools/
- Herramientas de seguridad de COAST:
  http://www.cs.purdue.edu/coast/coast-tools.html
- Herramientas de seguridad de FreeFire:
  http://sites.inka.de/sites/lina/freefire-l/tools.html
- Proyecto de enrutador de Linux:
  http://www.linuxrouter.org/
- Software Linuxberg:
  http://idirect.Linuxberg.com/software.html
- Masquerading Applications:
  http://www.tsmservices.com/masq
- Área de seguridad Sunsite:
  ftp://sunsite.unc.edu/pub/Linux/system/security/

Herramientas de seguridad

Las herramientas de software de seguridad están reflejadas en muchos sitios. Estas son algunas de las herramientas más habituales, con punteros a dónde encontrarlas:

- argus:
  ftp://ftp.sei.cmu.edu/pub/argus/
- COPS:
  ftp://sunsite.unc.edu/pub/Linux/system/security/cops_104_linux.tgz
  ftp://coast.cs.purdue.edu/pub/tools/unix/cops
  http://www.cert.org/ftp/tools/cops/
- crack:
  http://www.cert.org/ftp/tools/crack/
  http://www.cert.org/ftp/tools/cracklib/
  ftp://coast.cs.purdue.edu/pub/tools/unix/crack
  ftp://coast.cs.purdue.edu/pub/tools/unix/cracklib/
- dig:
  ftp://coast.cs.purdue.edu/pub/tools/unix/dig/
- icmpinfo:
  ftp://coast.cs.purdue.edu/pub/tools/unix/netmon/icmpinfo/
  http://www.deter.com/unix/software/icmpinfo-1.10.tar.gz
- ifstatus:
  ftp://coast.cs.purdue.edu/pub/tools/unix/ifstatus/
- IPSec:
  http://www.xs4all.nl/~freeswan/
- ISS:
  http://www.deter.com/unix/software/iss13.tar.gz
- Cadenas de firewalls IP de Linux:
  http://www.rustcorp.com/linux/ipchains/
- Herramienta de diseño de firewall IPFW de Linux IPFW:
  http://linux-firewall-tools.com/linux/firewall/
- logcheck:
  ftp://coast.cs.purdue.edu/pub/tools/unix/logcheck/
- lsof:
  ftp://coast.cs.purdue.edu/pub/tools/unix/lsof/
- md5:
  http://www.cert.org/ftp/tools/md5/
  ftp://coast.cs.purdue.edu/pub/tools/unix/md5/
- netcat:
  ftp://coast.cs.purdue.edu/pub/tools/unix/netcat/
- nmap:
  ftp://coast.cs.purdue.edu/pub/tools/unix/nmap/
- SATAN:
  ftp://coast.cs.purdue.edu/pub/tools/unix/satan/satan/satan-1.1.1.tar.Z
  http://www.fish.com/~zen/satan/satan.html

- sbscan:
  ftp://sunsite.unc.edu/pub/Linux/system/security/sbscan-0.04.tar.gz
- Página de firewalls de SINUS:
  http://www.ifi.unizh.ch/ikm/SINUS/firewall/
- SOCKS:
  http://www.socks.nec.com
- SSH:
  http://www.ssh.fi/sshprotocols2/
- SSL:
  http://psych.psy.uq.oz.au/~ftp/Crypto/
- strobe:
  ftp://coast.cs.purdue.edu/pub/tools/unix/strobe/
- tiger (Linux):
  ftp://sunsite.unc.edu/pub/Linux/system/security/tiger-2.2.4.tgz
  http://www.net.tamu.edu/ftp/security/TAMU/tiger-2.2.4.tgz
- Conjunto de herramientas de firewall TIS:
  http://www.tis.com/research/software/
- tripwire:
  http://www.tripwiresecurity.com/
  ftp://coast.cs.purdue.edu/pub/tools/unix/Tripwire/
  http://www.cert.org/ftp/tools/tripwire/

Herramientas de firewall

- Herramientas de diseño de firewalls:
  http://linux-firewall-tools.com/linux/firewall/
- FWCONFIG:
  http://www.mindstorm.com/~sparlin/fwconfig.shtml
- Filtro IP:
  http://cheops.anu.edu.au/~avalon/
- Módulo Dotfile ipfwadm:
  http://www.wolfenet.com/~jhardin/ipfwadm.html
- Firewall PPP Isinglass:
  http://www.tummy.com/isinglass/
- Mason:
  http://www.pobox.com/~wstearns/mason/
- Firewall SINUS:
  http://www.ifi.unizh.ch/ikm/SINUS/firewall/

- Conjunto de herramientas de firewalls de TIS:
  ftp://ftp.tis.com/pub/firewalls/toolkit/

Papeles de referencia y FAQ

La mayoría de los papeles de referencia que se listan en las siguientes secciones organizadas por temas las mantiene el CERT. Están reflejados en muchos sitios.

Seguridad de UNIX

- FTP anónimo Abuses:
  http://www.cert.org/ftp/tech_tips/anonymous_ftp_abuses
- Directrices de configuración de FTP anónimo:
  http://www.cert.org/ftp/tech_tips/anonymous_ftp_config
- Ataques de denegación de servicio:
  http://www.cert.org/ftp/tech_tips/denial_of_service
- Lista de comprobación de detección de intrusos:
  http://www.cert.org/ftp/tech_tips/intruder_detection_checklist
- Problemas con el comando PORT de FTP:
  http://www.cert.org/ftp/tech_tips/FTP_PORT_attacks
- Protegerse de ataques de archivos de contraseña:
  http://www.cert.org/ftp/tech_tips/passwd_file_protection
- Pasos para recuperarse de un compromiso raíz en UNIX:
  http://www.cert.org/tech_tips/root_compromise.html
- Lista de comprobación de seguridad de equipos UNIX:
  http://www.cert.org/ftp/tech_tips/AUSCERT_checklist1.1
- Directrices de configuración de UNIX:
  http://www.cert.org/ftp/tech_tips/UNIX_configuration_guidelines

Cuestiones de firewall

- Derrotas de ataques de denegación de servicio que emplean usurpamento de direcciones fuente IP:
  http://www.ietf.cnri.reston.va.us/rfc/rfc2267.txt
- Guía de directivas de firewalls (ICSA):
  http://www.icsa.net/services/consortia/firewalls/fwpg.shtml
- Preguntas comunes acerca de firewalls de Internet:
  http://www.clark.net/pub/mjr/pubs/fwfaq/

- Enmascaramiento IP para Linux:
  http://www.tor.shaw.wave.ca/~ambrose/
- Firewalls de Linux y FAQ de seguridad de LAN:
  http://linux-firewall-tools.com/linux/faq/
- Facilidades de seguridad de firewalls de Linux para revisión de paquetes a nivel de núcleo:
  http://www.xos.nl/linux/ipfwadm/paper/
- Sitio web de enmascaramiento IP de Linux:
  http://www.indyramp.com/masq/
- Filtrado de paquetes para sistemas de firewalls:
  http://www.cert.org/ftp/tech_tips/packet_filtering
- Reenvío de puerto:
  http://www.ox.compsoc.org.uk/~steve/portforwarding.html
- Tutorial de firewalls de PORTUS:
  http://www.lsli.com/tutorial.html
- Inundación síncrona TCP y ataques de usurpación IP (CERT Advisory CA-96.21):
  http://www.cert.org/advisories/CA-96.21.tcp_syn_flooding.html
- Números de puerto de servicio TCP/UDP (IANA):
  http://www.isi.edu/in-notes/iana/assignments/port-numbers

Cuestiones de servidor web

- Sugerencias de seguridad del servidor web Apache:
  http://www.apache.org/docs/misc/security_tips.html
- Cómo quitar metacaracteres de datos proporcionados por el usuario en secuencias de comandos CGI:
  http://www.cert.org/ftp/tech_tips/cgi_metacharacters
- Las FAQ de seguridad de la World Wide Web:
  http://www.w3.org/Security/Faq/www-security-faq.html

Documentación en línea

Linux se distribuye con un conjunto completo de documentación en varios formatos. Los documentos HTML que se listan aquí son especialmente importantes para problemas relacionados con la seguridad de redes:

- Cómo instalar firewalls y servidores proxy:
  /usr/doc/HOWTO/other-formats/html/Firewall-HOWTO.html
- Preguntas:
  /usr/doc/HOWTO/other-formats/html/HOWTO-INDEX.html

- Preguntas acerca del enmascaramiento IP de Linux:
  /usr/doc/HOWTO/other-formats/html/mini/IP-Masquerade.html
- Preguntas acerca de IPCHAINS de Linux:
  /usr/doc/HOWTO/other-formats/html/IPCHAINS-HOWTO.html
- Preguntas sobre la seguridad de Linux:
  /usr/doc/HOWTO/other-formats/html/Security-HOWTO.html
- Guía del administrador de red (NAG):
  /usr/doc/LDP/nag/nag.html
- Guía del administrador de sistemas (SAG):
  /usr/doc/LDP/sag/sag.html

Sitios web generales

Los sitios Linux que contienen software e información son abundantes; a continuación se listan algunos de las mejores referencias sobre estos dos temas:

- CableModem Info:
  http://www.cablemodeminfo.com/
- FreshMeat:
  http://freshmeat.net/
- Enlaces generales de Linux:
  http://www.emuse.net/
- Utilidades y aplicaciones de Linux:
  http://www.xnet.com/~blatura/linapps.shtml
- Aplicaciones de Linux:
  http://www.linuxapps.com/
- Proyecto de documentación de Linux:
  http://metalab.unc.edu/LDP/
- Linux a fondo:
  http://www.linuxpowered.com/html/linux_links/netw.html
- Inicio a Linux:
  http://linuxstart.com/
- Linux actual:
  http://linuxtoday.com/
- Mundo Linux:
  http://www.linuxworld.com/
- Red Hat:
  http://www.redhat.com/

- SlashDot:
  http://slashdot.org/
- SUNET (Swedish University Network):
  http://ftp.sunet.se/pub/os/Linux/
- Sunsite:
  http://metalab.unc.edu/pub/Linux/

Libros

Apache Server for Dummies, by Ken A. L. Coar. Foster City, CA: IDG Books Worldwide, Inc., 1998.

Building Internet Firewalls, by D. Brent Chapman and Elizabeth D. Zwicky. Sebastopol, CA: O'Reilly & Associates, Inc., 1995.

Firewalls and Internet Security: Repelling the Wily Hacker, by William R. Cheswick and Steven M. Bellovin. Reading, MA: Addison-Wesley, 1994.

Linux Network Toolkit, by Paul G. Sery. Foster City, CA: IDG Books Worldwide, Inc., 1998.

B

Ejemplos de firewalls y secuencias de comandos compatibles

Un firewall para un sistema independiente, o para proteger una LAN particular, se describe en el Capítulo 3, "Creación e instalación de un firewall". El ejemplo independiente se amplía en el Capítulo 4, "Redes de perímetro, firewalls múltiples y problemas con las LAN", para que éste funcione como un firewall bastión con un conjunto completo de reglas de firewall aplicadas tanto a la interfaz de Internet externa como a la interfaz LAN interna. El propio firewall independiente del Capítulo 3 se rehace para que funcione como un firewall de contención secundario. El bastión sirve como pasarela tanto para Internet como para una red DMZ que contiene servidores públicos. El firewall de contención sirve como pasarela entre una LAN privada y la DMZ o el firewall bastión, dependiendo del diseño de la DMZ.

En el Capítulo 3, el firewall de ejemplo se presenta de forma poco sistemática. Este apéndice muestra el mismo ejemplo de firewall como si apareciera en una secuencia de comandos de firewall, tanto con semántica ipchains como con semántica ipfwadm. De nuevo, el ejemplo no está optimizado. Las versiones ipchains e ipfwadm se presentan casi con una correspondencia de uno a uno.

En este libro no se explica la optimización de las reglas. Aquí se muestran algunas optimizaciones sencillas, tanto usando ipchains como ipfwadm, basándose en una secuencia de comandos de firewall sencillo para un solo equipo.

Por último, se muestran unas cuantas secuencias de comandos compatibles. No es fácil asignar direcciones dinámicas a los sistemas UNIX. DNS, en particular, espera una visión estable de las direcciones de equipo. Muchos sistemas dependen de DHCP para las direcciones IP asignadas dinámicamente, especialmente los sistemas particulares con conexiones a Internet 24/7. Tanto DNS como la secuencia de comandos de firewall requieren algo de ayuda, tanto cuando se asigna una dirección IP por primera vez, como posteriormente si se asigna una nueva dirección IP dinámicamente mientras la máquina está en línea.

ipchains rc.firewall para un sistema individual o para una LAN particular del Capítulo 3

El Capítulo 3 trata los protocolos de aplicación y las reglas de firewall para los tipos de servicio más usados en una máquina independiente e individual de Linux. Si se anexa una pequeña LAN de equipos cliente a una LAN interna, el firewall reenvía y enmascara todo el tráfico entre la LAN e Internet. Como ejemplo, el Capítulo 3 muestra muchos eventos de registro y de seguridad que no son estrictamente necesarios en un firewall totalmente funcional. Además, se presentan tanto las reglas de cliente como las de servidor para servicios que no todo el mundo usará. A continuación aparece la secuencia de comandos de firewall completo, como aparecería en el archivo /etc/rc.d/rc.firewall, y construido sobre ipchains:

```
#!/bin/sh

echo "Starting firewalling... "

# Algunas definiciones para mantenimiento sencillo.

# ------------------------------------------------------------------
# MODIFIQUE ESTE CÓDIGO PARA QUE SE AJUSTE
# A SUS NECESIDADES Y PSI.

EXTERNAL_INTERFACE="eth0"              # Interfaz conectada de Internet
LOOPBACK_INTERFACE="lo"                # o la especificación de nombres local
LAN_INTERFACE_1="eth1"                 # de la interfaz LAN interna

IPADDR="mi.dirección.ip"               # su dirección IP
LAN_1="192.168.1.0/24"                 # cualquier intervalo (privado) que use
LAN_IPADDR_1="192.168.1.1"             # dirección de interfaz interna

ANYWHERE="cualquiera/0"                # coincidir con cualquier dirección IP

DHCP_SERVER="mi.servidor.dhcp"         # si utiliza un intervalo de
MY_ISP="mi.intervalo.direcciones.psi"  # direcciones de PSI y NOC
NAMESERVER_1="mi.nombre.servidor.1"    # todos deben tener al menos uno

SMTP_SERVER="cualquiera/0"             # servidor de correo externo
SMTP_GATEWAY="mi.servidor.psi"         # envío de correo externo
POP_SERVER="mi.servidor.pop"           # servidor pop externo, si existe
IMAP_SERVER="mi.servidor.imap.psi"     # servidor imap externo, si existe
NEWS_SERVER="mi.servidor.noticias"     # servidor de noticias externo, si existe
WEB_PROXY_SERVER="mi.www.proxy"        # servidor proxy web del PSI, si existe
WEB_PROXY_PORT="www.puerto.proxy"      # puerto proxy web del PSI, si existe
                                       # normalmente 8008 o 8080
LOOPBACK="127.0.0.0/8"                 # intervalo reservado de direcciones de
bucle inverso
CLASS_A="10.0.0.0/8"                   # redes privadas de clase A
CLASS_B="172.16.0.0/12"                # redes privadas de clase B
```

```
CLASS_C="192.168.0.0/16"                    # redes privadas de clase C
CLASS_D_MULTICAST="224.0.0.0/4"             # direcciones de difusión múltiple de clase D
CLASS_E_RESERVED_NET="240.0.0.0/5"          # direcciones reservadas de clase E
BROADCAST_SRC="0.0.0.0"                     # dirección origen de difusión
BROADCAST_DEST="255.255.255.255"            # dirección destino de difusión
PRIVPORTS="0:1023"                          # intervalo de puerto privilegiado, bien conocido
UNPRIVPORTS="1024:65535"                    # intervalo de puerto no privilegiado

TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ------------------------------------------------------------------------

# Si la dirección IP se asigna de forma dinámica por un servidor DHCP,
# los nombres de servidor se encuentran en /etc/dhcpc/resolv.conf. Si se usa la
# secuencia de comandos de ejemplo ifdhcpc-done se actualiza automáticamente y
# se agrega a /etc/dhcpc/hostinfo-$EXTERNAL_INTERFACE o
# /etc/dhcpc/dhcpcd-$EXTERNAL_INTERFACE.info.

# Si se usa la secuencia de comandos de ejemplo ifdhcpc-done, las siguientes
# definiciones de NAMESERVER (una por servidor, hasta 3) se omitirán
# aquí correctamente.

#  La dirección IP, $IPADDR, se define por DHCP.

if [ -f /etc/dhcpc/hostinfo-$EXTERNAL_INTERFACE ]; then
    . /etc/dhcpc/hostinfo-$EXTERNAL_INTERFACE
elif [ -f /etc/dhcpc/dhcpcd-$EXTERNAL_INTERFACE.info ]; then
    . /etc/dhcpc/dhcpcd-$EXTERNAL_INTERFACE.info
    DHCP_SERVER=$DHCPSIADDR
else
    echo "rc.firewall:  DHCP no está configurado."
    ipchains -F
    ipchains -P input  DENY
    ipchains -P output DENY
    ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
    ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT
    ipchains -A input  -i $LAN_INTERFACE -j ACCEPT
    ipchains -A output -i $LAN_INTERFACE -j ACCEPT
    exit 1
fi

# Si se usa la secuencia de comandos de ejemplo ifdhcpc-done, cualquier versión
# anterior de IPADDR y NAMESERVER se omitirán aquí correctamente.
# ------------------------------------------------------------------------

# MODIFIQUE ESTO PARA QUE SE AJUSTE AL NÚMERO DE SERVIDORES
# CONEXIONES QUE SE SOPORTEN.

# La asignación del puerto X Window comienza en 6000 y se incrementa
# para cada servidor adicional que funcione entre 6000 y 6063.

XWINDOW_PORTS="6000"                    # (TCP) X Window

# SSH comienza en 1023 y sigue hasta 513 para
# cualquier otra conexión entrante simultánea adicional.
```

```
SSH_PORTS="1020:1023"                # conexiones simultáneas

# ------------------------------------------------------------------------

SOCKS_PORT="1080"                    # conectores (TCP)
OPENWINDOWS_PORT="2000"              # ventanas abiertas (TCP)
NFS_PORT="2049"                      # (TCP/UDP) NFS

# ------------------------------------------------------------------------

# Eliminar cualquier regla existente de todas las cadenas.
ipchains -F

# Establecer la directiva predeterminada a denegar.
ipchains -P input   DENY
ipchains -P output  REJECT
ipchains -P forward REJECT

# Establecer el tiempo de espera de enmascaramiento a 10 horas para las conexiones TCP.
ipchains -M -S 36000 0 0

# Deshabilitar los paquetes fragmentados
ipchains -A input -f -i LAN_INTERFACE_1 -j DENY

# Habilitar TCP SYN Cookie Protection
echo 1 >/proc/sys/net/ipv4/tcp_syncookies

# Permite activar la protección IP contra spoofing
# en Source Address Verification
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 > $f
done

# Deshabilitar la aceptación de direcciones ICMP
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    echo 0 > $f
done

# Deshabilitar paquetes de origen enrutado
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
    echo 0 > $f
done

# Estos módulos son necesarios para el enmascaramiento
# de sus respectivos servicios.
#/sbin/modprobe ip_masq_ftp.o
#/sbin/modprobe ip_masq_raudio.o
#/sbin/modprobe ip_masq_irc.o
#/sbin/modprobe/ip_masq_vdolive.o
#/sbin/modprobe/ip_masq_cuseeme.o
#/sbin/modprobe/ip_masq_quake.o

# ------------------------------------------------------------------------
# BUCLE INVERSO
```

```
# Tráfico ilimitado en la interfaz de bucle inverso
ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT

# ------------------------------------------------------------------------
# Rechazar cualquier conexión de sitios problemáticos.

# /etc/rc.d/rc.firewall.blocked contiene una lista de
# ipchains -A input -i $EXTERNAL_INTERFACE -s <dirección/máscara> -j DENY
# reglas para bloquear todos los accesos.

# Rechazar todos los paquetes que parezcan proceder de la lista prohibida.
if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    . /etc/rc.d/rc.firewall.blocked
fi

# ------------------------------------------------------------------------
# SPOOFING y DIRECCIONES ERRÓNEAS
# Rechazar los paquetes de spoof.
# Ignorar las direcciones origen de red que sean claramente ilegales.
# Protegerse de realizar envíos a direcciones erróneas.

# Rechazar los paquetes de spoof que pretendan entrar de
# direcciones IP de la interfaz externa.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $IPADDR -j DENY -l

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de clase A.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_A -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_A -j DENY -l

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de clase B.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_B -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_B -j DENY -l

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de clase C.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_C -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_C -j DENY -l

# Rechazar todos los paquetes que parezcan ir o proceder de la interfaz de bucle inverso.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $LOOPBACK -j DENY
ipchains -A output -i $EXTERNAL_INTERFACE -s $LOOPBACK -j DENY -l

# Rechazar paquetes de difusión mal formados.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $BROADCAST_DEST -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -d $BROADCAST_SRC  -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -s $BROADCAST_DEST -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $BROADCAST_SRC  -j DENY -l
```

```
# Rechazar direcciones de difusión múltiple de clase D.
# La difusión múltiple sólo es ilegal como dirección origen.
# La difusión múltiple usa UDP.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST \
         -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST \
         -j REJECT -l

# Rechazar direcciones IP reservadas de clase E.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_E_RESERVED_NET \
         -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_E_RESERVED_NET \
         -j REJECT

# Rechazar direcciones definidas como reservadas por el IANA.
# 0.*.*.*, 1.*.*.*, 2.*.*.*, 5.*.*.*, 7.*.*.*, 23.*.*.*, 27.*.*.*
# 31.*.*.*, 37.*.*.*, 39.*.*.*, 41.*.*.*, 42.*.*.*, 58-60.*.*.*

ipchains -A input -i $EXTERNAL_INTERFACE -s 1.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 2.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 5.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 7.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 23.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 27.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 31.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 37.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 39.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 41.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 42.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 58.0.0.0/7 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 60.0.0.0/8 -j DENY -l

# 65: 01000001   - /3 incluye 64 - es necesario escribir 65-79
ipchains -A input -i $EXTERNAL_INTERFACE -s 65.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 66.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 67.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 68.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 69.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 70.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 71.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 72.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 73.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 74.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 75.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 76.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 77.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 78.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 79.0.0.0/8 -j DENY -l

# 80: 01010000   - /4 enmascara 80-95
ipchains -A input -i $EXTERNAL_INTERFACE -s 80.0.0.0/4 -j DENY -l

# 96: 01100000  - /4 enmascara 96-111
ipchains -A input -i $EXTERNAL_INTERFACE -s 96.0.0.0/4 -j DENY -l
```

```
# 126: 01111110  - /3 incluye 127 - es necesario escribir 112-126
ipchains -A input -i $EXTERNAL_INTERFACE -s 112.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 113.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 114.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 115.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 116.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 117.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 118.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 119.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 120.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 121.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 122.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 123.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 124.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 125.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 126.0.0.0/8 -j DENY -l

# 217: 11011001  - /3 incluye 216 - es necesario escribir 217-219
ipchains -A input -i $EXTERNAL_INTERFACE -s 217.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 218.0.0.0/8 -j DENY -l
ipchains -A input -i $EXTERNAL_INTERFACE -s 219.0.0.0/8 -j DENY -l

# 223: 11011111  - /6 enmascara 220-223
ipchains -A input -i $EXTERNAL_INTERFACE -s 220.0.0.0/6 -j DENY -l
# ----------------------------------------
# ICMP

# (4)  Source_Quench
#      Solicitudes entrantes y salientes para ralentizar (control de flujo)

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 4 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 4 -d $ANYWHERE -j ACCEPT

# (12) Parameter_Problem
#      Mensajes de error entrantes y salientes

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 12 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 12 -d $ANYWHERE -j ACCEPT

# (3)  Dest_Unreachable, Service_Unavailable
#      Negociación de tamaño entrantes y salientes, no disponibilidad
#      del servicio o destino, respuesta final de traceroute

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 3 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 3 -d $MY_ISP -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR fragmentation-needed -d $ANYWHERE -j ACCEPT

# (11) Time_Exceeded
#       Condiciones de tiempo de espera entrantes y salientes,
#       además de respuestas intermedias TTL para trazar rutas

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 11 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 11 -d $MY_ISP -j ACCEPT

# Permitir pings salientes a cualquier sitio.

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 8 -d $ANYWHERE -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 0 -d $IPADDR -j ACCEPT

# Permitir pings entrantes de equipos de confianza.

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $MY_ISP 8 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 0 -d $MY_ISP -j ACCEPT

# ------------------------------------------
# PUERTOS NO PRIVILEGIADOS
# Evitar puertos sujetos a problemas de administración de sistema y protocolo.

# OpenWindows: estableciendo una conexión
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
         -s $IPADDR \
         -d $ANYWHERE $OPENWINDOWS_PORT -j REJECT

# Conexión entrante OpenWindows
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $OPENWINDOWS_PORT -j DENY

# X Window: establecer una conexión remota
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
         -s $IPADDR \
         -d $ANYWHERE $XWINDOW_PORTS -j REJECT

# X Window: intento de conexión entrante
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $XWINDOW_PORTS -j DENY -l

# SOCKS: establecer una conexión
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
         -s $IPADDR \
         -d $ANYWHERE $SOCKS_PORT -j REJECT -l
```

```
# SOCKS conexión entrante
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $SOCKS_PORT -j DENY

# NFS: conexiones TCP
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $NFS_PORT -j DENY -l

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $ANYWHERE $NFS_PORT -j REJECT -l

# NFS: conexiones UDP
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR $NFS_PORT -j DENY -l

# Solicitud entrante NFS (modo UDP normal)
ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -d $ANYWHERE $NFS_PORT -j REJECT -l

# ----------------------------------------------------------
# NOTA:
#       Los nombres simbólicos que se usan en /etc/services para los nombres de puerto
#       varían según el proveedor. Usarlos es menos propenso a error y tiene más
#       sentido.

# ----------------------------------------------------------------------------
# Servicios necesarios

# modos de cliente DNS  (53)
# ------------.

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NAMESERVER_1 53 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER_1 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# El protocolo permite las solicitudes TCP de cliente a servidor
# si fallan las solicitudes UDP. Esto raramente se ve. Normalmente, los
# clientes usan TCP como nombre de servidor secundario para transferencias
# de zona desde su servidor de nombres primario, igual que los hacker.

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d <mi.dns.primario> 53 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s <mi.dns.primario> 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT


# modos de servidor DNS (53)
# --------------------
```

```
# nombre de servidor de reenvío y caché DNS
# ----------------------------------------

# Solicitud o respuesta servidor a servidor
# Guardar en caché sólo el nombre de servidor usa UDP, no TCP

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d $NAMESERVER_1 53 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER_1 53 \
         -d $IPADDR 53 -j ACCEPT

# nombre completo del servidor DNS
# --------------------

# Transacción DNS cliente a servidor
ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s <mis.clientes.dns> $UNPRIVPORTS \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d <mis.clientes.dns> $UNPRIVPORTS -j ACCEPT

# Transacción DNS de servidor principal a principal
ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s <mis.clientes.dns> 53 \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d <mis.clientes.dns> 53 -j ACCEPT

# Transferencias de zona
# debido al posible daño de transferencias de zona,
# permitir el tráfico TCP sólo a secundarios específicos.

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <mis.dns.secundarios> $UNPRIVPORTS \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 53 \
         -d <mis.dns.secundarios> $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------

# AUTH (113) - Aceptar sus solicitudes AUTH salientes como cliente
# ------------------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 113 -j ACCEPT
```

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 113 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# Servidor AUTH (113)
# -----------------
# Aceptar solicitudes AUTH entrantes

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 113 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 113 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# OR

# Rechazar solicitudes AUTH entrantes

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -d $IPADDR 113 -j REJECT

# ------------------------------------------------------------------------
# Servicios TCP en puertos seleccionados

# Enviar correo a través de una pasarela SMTP remota (25)
# ------------------------------------------------

# Cliente SMTP a una cuenta PSI sin un servidor local

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ SMTP_GATEWAY 25 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ SMTP_GATEWAY 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# OR
# Enviar correo a través de un servidor SMTP local

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 25 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# Recibir correo a través de un servidor SMTP local (25)
# ----------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 25 -j ACCEPT
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 25 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# POP (110) - Obtener correo como cliente POP
# ------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $POP_SERVER 110 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $POP_SERVER 110 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# POP (110) - Albergar un servidor POP para clientes remotos
# ----------------------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <mis.clientes.pop> $UNPRIVPORTS \
         -d $IPADDR 110 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 110 \
         -d <mis.clientes.pop> $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# IMAP (143) - Obtener correo como cliente IMAP
# --------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d <mi.servidor.imap> 143 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s <mi.servidor.imap> 143 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# IMAP (143) - Albergar un servidor IMAP para clientes remotos
# -----------------------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <mis.clientes.imap> $UNPRIVPORTS \
         -d $IPADDR 143 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 143 \
         -d <mis.clientes.imap> $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# NNTP (119) - Recibir y enviar noticias como un cliente de Usenet
# ---------------------------------------------------------
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NEWS_SERVER 119 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $NEWS_SERVER 119 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT


# NNTP (119) - Albergar un servidor de noticias OP para clientes remotos
# ---------------------------------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <mis.clientes.noticias> $UNPRIVPORTS \
         -d $IPADDR 119 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 119 \
         -d <mis.clientes.noticias> $UNPRIVPORTS -j ACCEPT

# NNTP (119) - Permitir la recepción de noticias de servidores principales para
# un servidor local de Usenet
# ---------------------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d <mi.origen.noticias> 119 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s <mi.origen.noticias> 119 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ---------------------------------------------------------------

# TELNET (23) - Permitir el acceso cliente saliente a sitios remotos
# ---------------------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 23 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 23 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# TELNET (23) - Permitir el acceso de entrada al servidor local
# ---------------------------------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 23 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 23 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

```
# ------------------------------------------------------------------

# Cliente SSH (22) - Permitir el acceso cliente a servidores remotos SSH
# ------------------------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $SSH_PORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $IPADDR $SSH_PORTS -j ACCEPT

# SSH (22) - Permitir el acceso remoto de cliente al servidor local SSH
# ------------------------------------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 22 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $SSH_PORTS \
         -d $IPADDR 22 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE $SSH_PORTS -j ACCEPT

# ------------------------------------------------------------------

# FTP (20, 21) - Permitir el acceso cliente saliente a servidores FTP remotos
# ------------------------------------------------------------------

# Solicitud saliente

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 21 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 21 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

```
# Canales de datos FTP de modo de puerto normal

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE 20 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 20 -j ACCEPT

# Canales de datos FTP de modo pasivo

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR $UNPIRVPORTS -j ACCEPT

# FTP (20, 21) - Permitir el acceso entrante al servidor FTP local
# ----------------------------------------------------------------

# Solicitud entrante

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 21 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 21 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# Respuestas de canales de datos FTP de modo de puerto normal

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR 20 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 20 -j ACCEPT

# Respuestas de canales de datos FTP de modo pasivo

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# ----------------------------------------------------------------
```

```
# HTTP (80) - Acceder a sitios web remotos como cliente
# ----------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 80 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 80 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# HTTP (80) - Permitir el acceso remoto al servidor web local
# -----------------------------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 80 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 80 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT


# HTTPS (443) - Acceder a sitios web remotos como cliente sobre SSL
# -----------------------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 443 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 443 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# HTTPS (443) - Permitir el acceso remoto al servidor web SSL local
# -----------------------------------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 443 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 443 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------------

# Cliente proxy HTTP (8008/8080)
# ------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $WEB_PROXY_SERVER $WEB_PROXY_PORT -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $WEB_PROXY_SERVER $WEB_PROXY_PORT \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

```
# ------------------------------------------------------------------

# FINGER (79) - Acceder a servidores finger remotos como cliente
# ----------------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 79 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 79 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# FINGER (79) - Permitir el acceso remoto cliente al servidor finger local
# ----------------------------------------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s <mis.clientes.finger> $UNPRIVPORTS \
         -d $IPADDR 79 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 79 \
         -d <mis.clientes.finger> $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# Cliente WHOIS (43)
# ------------------
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 43 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 43 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# Cliente Gopher (70)
# -------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 70 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 70 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# Cliente WAIS (210)
# ------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 210 -j ACCEPT
```

```
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 210 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------
# Aceptar UDP sólo en puertos seleccionados

# TRACEROUTE
# traceroute suele usar -S 32769:65535 -D 33434:33523
# ------------------------------------------------------

# Habilitar solicitudes traceroute salientes
# ------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $TRACEROUTE_SRC_PORTS \
         -d $ANYWHERE $TRACEROUTE_DEST_PORTS -j ACCEPT

# Solicitud entrante del PSI.
# Todas las demás se deniegan de forma predeterminada.
# ------------------.

ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -s $MY_ISP 32769:65535 \
         -d $IPADDR 33434:33523 -j ACCEPT

# ------------------------------------------------------------------

# Cliente DHCP (67, 68)
# --------------------

# INIT o REBINDING: No alquilar o tiempo de alquiler sobrepasado.

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $BROADCAST_0 68 \
         -d $BROADCAST_1 67 -j ACCEPT

# Cambiando de número

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $BROADCAST_0 67 \
         -d $BROADCAST_1 68 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $DHCP_SERVER 67 \
         -d $BROADCAST_1 68 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $BROADCAST_0 68 \
         -d $DHCP_SERVER 67 -j ACCEPT

# Como resultado de lo anterior, se supone que hemos cambiado
# la dirección IP con este mensaje, que se envía a nuestra nueva dirección
# antes de que el cliente dhcp haya recibido la actualización.

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $DHCP_SERVER 67 \
         -d $MY_ISP 68 -j ACCEPT
```

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $DHCP_SERVER 67 \
         -d $IPADDR 68 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 68 \
         -d $DHCP_SERVER 67 -j ACCEPT

# ------------------------------------------------------------------

# NTP (123) - Acceder a servidores remotos externos de hora
# ------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $UNPRIVPORTS \
         -d <mi.proveedor.hora> 123 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s <mi.proveedor.hora> 123 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------
# Tráfico ilimitado dentro de la red local.

# Todas las máquinas internas tienen acceso a la máquina firewall.

ipchains -A input  -i $LAN_INTERFACE_1 \
         -s $LAN_1 -j ACCEPT

ipchains -A output -i $LAN_INTERFACE_1 \
         -d $LAN_1 -j ACCEPT

# ------------------------------------------------------------------
# Enmascarar el tráfico interno.

# Todo el tráfico interno se enmascara de forma externa.

ipchains -A forward -i $EXTERNAL_INTERFACE -s $LAN_1 -j MASQ

# ------------------------------------------------------------------

echo "done"

exit 0
```

ipfwadm rc.firewall para un sistema individual o para una LAN particular del Capítulo 3

Todos los ejemplos de reglas de firewall de este libro se basan en la versión Linux 6.0 de Red Hat usando ipchains. Sin embargo, muchos sistemas ejecutan versiones anteriores de Linux que construyen firewalls basados en ipfwadm. Esta sección lista la misma secuencia de comandos de firewall com-

pleta, como debería aparecer en el archivo /etc/rc.d/rc.firewall, creado sobre ipfwadm:

```
#!/bin/sh

echo "Starting firewalling... "

# Algunas definiciones para mantenimiento sencillo.

ANYWHERE="cualquiera/0"

# ------------------------------------------------------------------
# MODIFIQUE ESTE CÓDIGO PARA QUE SE AJUSTE
# A SUS NECESIDADES Y PSI.

EXTERNAL_INTERFACE="eth0"              # Interfaz conectada de Internet
LOOPBACK_INTERFACE="lo"                # o la especificación de nombres local
LAN_INTERFACE_1="eth1"                 # de la interfaz LAN interna

IPADDR="mi.dirección.ip"               # su dirección IP
LAN_1="192.168.1.0/24"                 # cualquier intervalo (privado) que use
LAN_IPADDR_1="192.168.1.1"             # dirección de interfaz interna

ANYWHERE="cualquiera/0"                # coincidir con cualquier dirección IP

DHCP_SERVER="mi.servidor.dhcp"         # si utiliza un intervalo de
MY_ISP="mi.intervalo.direcciones.psi"  # direcciones de PSI y NOC
NAMESERVER_1="mi.nombre.servidor.1"    # todos deben tener al menos uno

SMTP_SERVER="cualquiera/0"             # servidor de correo externo
SMTP_GATEWAY="mi.servidor.psi"         # envío de correo externo
POP_SERVER="mi.servidor.pop"           # servidor pop externo, si existe
IMAP_SERVER="mi.servidor.imap.psi"     # servidor imap externo, si existe
NEWS_SERVER="mi.servidor.noticias"     # servidor de noticias externo, si existe
WEB_PROXY_SERVER="mi.www.proxy"        # servidor proxy web del PSI, si existe
WEB_PROXY_PORT="www.puerto.proxy"      # puerto proxy web del PSI, si existe
                                       # normalmente 8008 o 8080

LOOPBACK="127.0.0.0/8"                 # intervalo reservado de direcciones de
                                       # bucle inverso
CLASS_A="10.0.0.0/8"                   # redes privadas de clase A
CLASS_B="172.16.0.0/12"                # redes privadas de clase B
CLASS_C="192.168.0.0/16"               # redes privadas de clase C
CLASS_D_MULTICAST="224.0.0.0/4"        # direcciones de difusión múltiple de clase D
CLASS_E_RESERVED_NET="240.0.0.0/5"     # direcciones reservadas de clase E
BROADCAST_SRC="0.0.0.0"                # dirección origen de difusión
BROADCAST_DEST="255.255.255.255"       # dirección destino de difusión
PRIVPORTS="0:1023"                     # intervalo de puerto privilegiado, bien
                                       # conocido
UNPRIVPORTS="1024:65535"               # intervalo de puerto no privilegiado

TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ..................................................................
```

```
# Si la dirección IP se asigna de forma dinámica por un servidor DHCP,
# los nombres de servidor se encuentran en /etc/dhcpc/resolv.conf. Si se usa la
# secuencia de comandos de ejemplo ifdhcpc-done se actualiza automáticamente y
# se agrega a /etc/dhcpc/hostinfo-$EXTERNAL_INTERFACE o
# /etc/dhcpc/dhcpcd-$EXTERNAL_INTERFACE.info.

# Si se usa la secuencia de comandos de ejemplo ifdhcpc-done, las siguientes
# definiciones de NAMESERVER (una por servidor, hasta 3) se omitirán
# aquí correctamente.

# En caso contrario, si dispone de una dirección IP estática,
# debe definir tanto la dirección IP estática como las direcciones
# IP externas del nombre del servidor.

if [ -f /etc/dhcpc/hostinfo-$EXTERNAL_INTERFACE ]; then
    . /etc/dhcpc/hostinfo-$EXTERNAL_INTERFACE
elif [ -f /etc/dhcpc/dhcpcd-$EXTERNAL_INTERFACE.info ]; then
    . /etc/dhcpc/dhcpcd-$EXTERNAL_INTERFACE.info
else
    NAMESERVER_1="mi.nombre.servidor.1"
    IPADDR="mi.dirección.ip"
fi

# ------------------------------------------------------------------
# MODIFIQUE ESTO PARA QUE SE AJUSTE AL NÚMERO DE SERVIDORES
# CONEXIONES QUE SE SOPORTEN.

# La asignación del puerto X Window comienza en 6000 y se incrementa
# para cada servidor adicional que funcione entre 6000 y 6063.

XWINDOW_PORTS="6000"                    # (TCP) X Window

# SSH comienza en 1023 y sigue hasta 513 para
# cualquier otra conexión entrante simultánea adicional.

SSH_PORTS="1020:1023"                   # conexiones simultáneas

# ------------------------------------------------------------------

SOCKS_PORT="1080"                       # conectores (TCP)
OPENWINDOWS_PORT="2000"                 # ventanas abiertas (TCP)
NFS_PORT="2049"                         # (TCP/UDP) NFS

# ------------------------------------------------------------------

# Habilitar TCP SYN Cookie Protection
echo 1 >/proc/sys/net/ipv4/tcp_syncookies

# Permite activar la protección IP contra spoofing
# en Source Address Verification
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 > $f
done

# Deshabilitar la aceptación de direcciones ICMP
```

```
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    echo 0 > $f
done

# Deshabilitar paquetes de origen enrutado
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
    echo 0 > $f
done

# Estos módulos son necesarios para el enmascaramiento
# de los servicios.
#/sbin/modprobe ip_masq_ftp.o
#/sbin/modprobe ip_masq_raudio.o
#/sbin/modprobe ip_masq_irc.o
#/sbin/modprobe/ip_masq_vdolive.o
#/sbin/modprobe/ip_masq_cuseeme.o
#/sbin/modprobe/ip_masq_quake.o

# ------------------------------------------------------------------

# Eliminar cualquier regla existente de todas las cadenas

ipfwadm -I -f
ipfwadm -O -f
ipfwadm -F -f

# Establecer la directiva predeterminada a denegar.
ipfwadm -I -p deny
ipfwadm -O -p reject
ipfwadm -F -p deny

# ------------------------------------------------------------------
# BUCLE INVERSO

# Tráfico ilimitado en la interfaz de bucle inverso
ipfwadm -I -a accept -W $LOOPBACK_INTERFACE
ipfwadm -O -a accept -W $LOOPBACK_INTERFACE

# ------------------------------------------------------------------
# Rechazar cualquier conexión de sitios problemáticos.

# /etc/rc.d/rc.firewall.blocked contiene una lista de
# ipchains -A input -i $EXTERNAL_INTERFACE -s <dirección/máscara> -j DENY
# reglas para bloquear todos los accesos.

# Rechazar todos los paquetes que parezcan proceder de la lista prohibida
if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    . /etc/rc.d/rc.firewall.blocked
fi

# ------------------------------------------------------------------
# SPOOFING y DIRECCIONES ERRÓNEAS
# Rechazar los paquetes de spoof.
# Ignorar las direcciones origen de red que sean claramente ilegales.
# Protegerse de realizar envíos a direcciones erróneas.
```

```
# Rechazar los paquetes de spoof que pretendan ser del usuario
# o de direcciones ilegales.
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S $IPADDR -o

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase A.
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S $CLASS_A -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -D $CLASS_A -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -S $CLASS_A -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -D $CLASS_A -o

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase B.
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S $CLASS_B -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -D $CLASS_B -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -S $CLASS_B -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -D $CLASS_B -o

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase C.
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S $CLASS_C -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -D $CLASS_C -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -S $CLASS_C -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -D $CLASS_C -o

# Rechazar todos los paquetes que parezcan ir o proceder de la interfaz de bucle
# inverso.
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S $LOOPBACK -o

# Rechazar paquetes de difusión mal formados.
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S $BROADCAST_DEST -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -D $BROADCAST_SRC -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -S $BROADCAST_DEST -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -D $BROADCAST_SRC -o

# Rechazar direcciones de difusión múltiple de clase D.
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S $MULTICAST -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -S $MULTICAST -o

# Rechazar direcciones IP reservadas de clase E.
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S $RESERVED_NET -o
ipfwadm -O -a reject -W $EXTERNAL_INTERFACE -D $RESERVED_NET -o


# dirección definida como reservada por el IANA.
# 0.*.*.*, 1.*.*.*, 2.*.*.*, 5.*.*.*, 7.*.*.*, 23.*.*.*, 27.*.*.*
# 31.*.*.*, 37.*.*.*, 39.*.*.*, 41.*.*.*, 42.*.*.*, 58-60.*.*.*

ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 1.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 2.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 5.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 7.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 23.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 27.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 31.0.0.0/8 -o
```

```
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 37.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 39.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 41.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 42.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 58.0.0.0/7 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 60.0.0.0/8 -o

# 65: 01000001    - /3 incluye 64 - es necesario escribir 65-79
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 65.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 66.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 67.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 68.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 69.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 70.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 71.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 72.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 73.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 74.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 75.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 76.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 77.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 78.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 79.0.0.0/8 -o

# 80: 01010000    - /4 enmascara 80-95
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 80.0.0.0/4 -o

# 96: 01100000  - /4 enmascara 96-111
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 96.0.0.0/4 -o

# 126: 01111110  - /3 incluye 127 - es necesario escribir 112-126
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 112.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 113.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 114.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 115.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 116.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 117.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 118.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 119.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 120.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 121.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 122.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 123.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 124.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 125.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 126.0.0.0/8 -o

# 217: 11011001  - /3 incluye 216 - es necesario escribir 217-219
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 217.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 218.0.0.0/8 -o
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 219.0.0.0/8 -o

# 223: 11011111  - /6 enmascara 220-223
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S 220.0.0.0/6 -o

# ----------------------------------------
# ICMP
```

```
# (4)  Source_Quench
#       Solicitudes entrantes y salientes para ralentizar (control de flujo)
ipfwadm -I -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 4 -D $IPADDR

ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 4 -D $ANYWHERE

# (12) Parameter_Problem
#       Mensajes de error entrantes y salientes
ipfwadm -I -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 12 -D $IPADDR

ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 12 -D $ANYWHERE

# (3)  Dest_Unreachable, Service_Unavailable
#       Negociación de tamaño entrantes y salientes, no disponibilidad
#       del servicio o destino, respuesta final de traceroute
ipfwadm -I -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 3 -D $IPADDR

ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 3 -D $ANYWHERE

# (11) Time_Exceeded
#       Condiciones de tiempo de espera entrantes y salientes,
#       además de respuestas intermedias TTL para trazar rutas
ipfwadm -I -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 11 -D $IPADDR

ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 11 -D $MY_ISP


# Permitir pings salientes a cualquier sitio.
ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 8 -D $ANYWHERE

ipfwadm -I -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 0 -D $IPADDR

# Permitir pings entrantes de equipos de confianza.
ipfwadm -I -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $MY_ISP 8 -D $IPADDR

ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 0 -D $MY_ISP

# ----------------------------------------------------
# No permitir cierto tráfico saliente para protegerle de errores.

# OpenWindows: estableciendo una conexión
ipfwadm -O -a reject -P tcp -y -W $EXTERNAL_INTERFACE \
        -S $IPADDR \
        -D $ANYWHERE $OPENWINDOWS_PORT
```

```
# X Window: estableciendo una conexión
ipfwadm -O -a reject -P tcp -y -W $EXTERNAL_INTERFACE \
        -S $IPADDR \
        -D $ANYWHERE $XWINDOW_PORTS

# SOCKS: estableciendo una conexión
ipfwadm -O -a reject -P tcp -y -W $EXTERNAL_INTERFACE \
        -S $IPADDR \
        -D $ANYWHERE $SOCKS_PORT

# ------------------------------------------------------------------
# PUERTOS TCP NO PRIVILEGIADOS
# Evitar puertos sujetos a problemas de administración de sistema y protocolo.

# Conexión entrante NFS (modo TCP atípico)
ipfwadm -I -a deny -P tcp -y -W $EXTERNAL_INTERFACE \
        -D $IPADDR $NFS_PORT

# Conexión entrante OpenWindows
ipfwadm -I -a deny -P tcp -y -W $EXTERNAL_INTERFACE \
        -D $IPADDR $OPENWINDOWS_PORT

# Conexión entrante X Window
ipfwadm -I -a deny -P tcp -y -W $EXTERNAL_INTERFACE \
        -D $IPADDR $XWINDOW_PORTS

# Conexión entrante SOCKS
ipfwadm -I -a deny -P tcp -y -W $EXTERNAL_INTERFACE \
        -D $IPADDR $SOCKS_PORT

# ------------------------------------------------------------------
# PUERTOS UDP NO PRIVILEGIADOS
# Evitar puertos sujetos a problemas de administración de sistema y protocolo.

# Solicitud entrante NFS (modo UDP normal)
ipfwadm -I -a deny -P udp -W $EXTERNAL_INTERFACE \
        -D $IPADDR $NFS_PORT

# ------------------------------------------------------------------
# NOTA:
#       Los nombres simbólicos que se usan en /etc/services para los nombres de puerto
#       varían según el proveedor. Usarlos es menos propenso a error  y tiene más
#       sentido.

# ------------------------------------------------------------------
# Servicios necesarios

# modos de cliente DNS  (53)
# --------------------

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $NAMESERVER_1 53

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $NAMESERVER_1 53 \
        -D $IPADDR $UNPRIVPORTS
```

```
# El protocolo permite las solicitudes TCP de cliente a servidor
# si fallan las solicitudes UDP. Esto raramente se ve. Normalmente, los
# clientes usan TCP como nombre de servidor secundario para transferencias
# de zona desde su servidor de nombres primario, igual que los hacker.

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $NAMESERVER_1 53

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $NAMESERVER_1 53 \
        -D $IPADDR $UNPRIVPORTS

# modos de servidor DNS (53)
# ----------------------

# nombre de servidor de reenvío y caché DNS
# -------------------------------------

# Solicitud o respuesta servidor a servidor
# Guardar en caché sólo el nombre de servidor usa UDP, no TCP

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 53 \
        -D $NAMESERVER_1 53

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $NAMESERVER_1 53 \
        -D $IPADDR 53

# nombre completo del servidor DNS
# --------------------

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S <mis.clientes.dns> 53 $UNPRIVPORTS \
        -D $IPADDR 53

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 53 \
        -D <mis.clientes.dns> 53 $UNPRIVPORTS

# Transferencias de zona
# debido al posible daño de transferencias de zona,
# permitir el tráfico TCP sólo a secundarios específicos.

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S <mis.dns.secundarios> $UNPRIVPORTS \
        -D $IPADDR 53

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 53 \
        -D <mis.dns.secundarios> $UNPRIVPORTS

# ------------------------------------------------------------------

# Cliente AUTH (113)
```

```
# ---------.

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 113

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 113 \
        -D $IPADDR $UNPRIVPORTS

# Servidor AUTH (113)
# ---------.

# Rechazar solicitudes AUTH entrantes

ipfwadm -I -a reject -P tcp -W $EXTERNAL_INTERFACE \
        -D $IPADDR 113

# O
# Rechazar solicitudes AUTH entrantes

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR 113

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 113 \
        -D $ANYWHERE $UNPRIVPORTS

# ------------------------------------------------------------------------
# Servicios TCP en puertos seleccionados

# Cliente SMTP (25)
# ----------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 25

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 25 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------------
# O cliente SMTP a una cuenta PSI sin un servidor local

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $SMTP_SERVER 25

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $SMTP_SERVER 25 \
        -D $IPADDR $UNPRIVPORTS

# Servidor SMTP (25)
```

```
# -----------------

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR 25

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 25 \
        -D $ANYWHERE $UNPRIVPORTS

# ------------------------------------------------------------------------

# Cliente POP (110)
# ----------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $POP_SERVER 110

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $POP_SERVER 110 \
        -D $IPADDR $UNPRIVPORTS

# Servidor POP (110)
# ---------

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S <mis.clientes.pop> $UNPRIVPORTS \
        -D $IPADDR 110

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 110 \
        -D <mis.clientes.pop> $UNPRIVPORTS

# ------------------------------------------------------------------------

# Cliente IMAP (143)
# -----------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D <mi.servidor.imap> 143

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S <mi.servidor.imap> 143 \
        -D $IPADDR $UNPRIVPORTS

# Servidor IMAP (143)
# -----------------

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S <mis.clientes.imap> $UNPRIVPORTS \
        -D $IPADDR 143

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
```

```
        -S $IPADDR 143 \
        -D <mis.clientes.imap> $UNPRIVPORTS

# ------------------------------------------------------------------

# Cliente NNTP NEWS (119)
# ----------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $NEWS_SERVER 119

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $NEWS_SERVER 119 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------

# Cliente TELNET (23)
# ------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 23

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 23 \
        -D $IPADDR $UNPRIVPORTS

# Servidor TELNET (23)
# ------------------

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR 23

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 23 \
        -D $ANYWHERE $UNPRIVPORTS

# ------------------------------------------------------------------

# Cliente SSH (22)
# ---------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 22

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 22 \
        -D $IPADDR $UNPRIVPORTS

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $SSH_PORTS \
        -D $ANYWHERE 22
```

```
ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 22 \
        -D $IPADDR $SSH_PORTS

# Servidor SSH (22)
# ---------------

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR 22

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 22 \
        -D $ANYWHERE $UNPRIVPORTS

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $SSH_PORTS \
        -D $IPADDR 22

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 22 \
        -D $ANYWHERE $SSH_PORTS

# ------------------------------------------------------------------------

# Cliente FTP (20, 21)
# --------------------

# Solicitud saliente

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 21

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 21 \
        -D $IPADDR $UNPRIVPORTS

# Creación del canal de datos del modo NORMAL

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 20 \
        -D $IPADDR $UNPRIVPORTS

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 20

# Creación del canal de datos del modo PASSIVE

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE $UNPRIVPORTS

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
```

```
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR $UNPRIVPORTS

# Servidor FTP (20, 21)
# --------------------

# Solicitud entrante

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR 21

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 21 \
        -D $ANYWHERE $UNPRIVPORTS

# Respuestas del canal de datos del modo PORT

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR 20 \
        -D $ANYWHERE $UNPRIVPORTS

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR 20

# Respuestas del canal de datos del modo PASSIVE

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR $UNPRIVPORTS

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE $UNPRIVPORTS

# ------------------------------------------------------------------------

# Cliente HTTP (80)
# -----------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 80

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 80 \
        -D $IPADDR $UNPRIVPORTS

# Servidor HTTP (80)
# -----------------

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR 80
```

```
ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 80 \
        -D $ANYWHERE $UNPRIVPORTS

# ------------------------------------------------------------------


# Cliente HTTPS (443)
# ------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 443

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 443 \
        -D $IPADDR $UNPRIVPORTS

# Servidor HTTPS (443)
# ------------------

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR 443

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 443 \
        -D $ANYWHERE $UNPRIVPORTS

# ------------------------------------------------------------------

# Cliente proxy HTTP (8008/8080)
# ----------------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $WEB_PROXY_SERVER $WEB_PROXY_PORT

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $WEB_PROXY_SERVER $WEB_PROXY_PORT \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------

# Cliente FINGER (79)
# ------------------

ipfwadm -O -a accept -P tcp   -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 79

ipfwadm -I -a accept -P tcp -k  -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 79 \
        -D $IPADDR $UNPRIVPORTS
```

```
# Servidor FINGER (79)
# ------------------

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S <mis.clientes.finger> $UNPRIVPORTS \
        -D $IPADDR 79

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 79 \
        -D <mis.clientes.finger> $UNPRIVPORTS

# ------------------------------------------------------------------------

# Cliente WHOIS (43)
# ------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 43

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 43 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------------

# Cliente Gopher (70)
# -------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 70

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 70 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------------

# Cliente WAIS (210)
# ------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 210

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 210 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------------

# TRACEROUTE
# traceroute suele usar -S 32769:65535 -D 33434:33523
# ------------------------------------------------------
```

```
# Habilitar solicitudes traceroute salientes
# ------------------------------------

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 32769:65535 \
        -D $ANYWHERE 33434:33523

# Solicitud entrante del PSI.
# Todas las demás se deniegan de forma predeterminada.
# --------------------------------

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $MY_ISP 32769:65535 \
        -D $IPADDR 33434:33523

# ------------------------------------------------------------------------
# Aceptar UDP sólo en puertos seleccionados


# Cliente DHCP (67, 68)
# --------------------

# INIT o REBINDING: No alquilar o tiempo de alquiler sobrepasado.

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $BROADCAST_SRC 68 \
        -D $BROADCAST_DEST 67

# Cambiando de número

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $BROADCAST_SRC 67 \
        -D $BROADCAST_DEST 68

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $DHCP_SERVER 67 \
        -D $BROADCAST_DEST 68

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $BROADCAST_SRC 68 \
        -D $DHCP_SERVER 67

# Como resultado de lo anterior, se supone que hemos cambiado
# la dirección IP con este mensaje, que se envía a nuestra nueva dirección
# antes de que el cliente dhcp haya recibido la actualización.

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $DHCP_SERVER 67 \
        -D $MY_ISP 68

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $DHCP_SERVER 67 \
        -D $IPADDR 68

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 68 \
        -D $DHCP_SERVER 67
```

```
# ------------------------------------------------------------------

# NTP (123) - Acceder a servidores remotos externos de hora
# ------------------------------------------------------

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D <mi.proveedor.hora> 123

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S <mi.proveedor.hora> 123 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------
# Tráfico ilimitado dentro de la red local.

# Todas las máquinas internas tienen acceso a la máquina firewall.

ipfwadm -I -a accept -W $LAN_INTERFACE_1 \
           -S $LAN_1

ipfwadm -O -a accept -W $LAN_INTERFACE_1 \
           -D $LAN_1

# ------------------------------------------------------------------
# Enmascarar el tráfico interno.

# Todo el tráfico interno se enmascara de forma externa.

ipfwadm -F -a masquerade -W $EXTERNAL_INTERFACE -S $LAN_1

# ------------------------------------------------------------------

echo "done"

exit 0
```

Optimización de las reglas de firewall

La optimización del firewall no es un tema excesivamente importante para un sistema particular. Es muy probable que el código de red de Linux pueda manejar los paquetes más rápidamente que la conexión de red. Esto se debe a que las reglas de firewall son dependientes del orden y difíciles de crear, por lo que optimizar su legibilidad es probablemente mucho mejor que optimizar su velocidad.

Es necesario ordenar las reglas de firewall de más específicas a más generales. Por lo que respecta a la dependencia del orden creado por las reglas particulares que se usan, el orden de las reglas puede optimizarse colocando las reglas para los paquetes que aparecen con mayor frecuencia antes en las listas de reglas de firewall. Por ejemplo, si rara vez se usa ftp, es muy probable que el tráfico relacionado con la Web se vea más a menudo que el tráfico

relacionado con ftp. El recorrido de la lista acaba tan pronto como el paquete coincide con una regla, así que colocando las reglas http antes de las reglas ftp se reduce la sobrecarga extra cuando se deben hacer coincidir los paquetes http que son más habituales.

También podemos optimizar las reglas de firewall colocando las reglas de entrada y salida de forma relativa a otras reglas en sus cadenas respectivas, en vez de agrupar juntas el par de reglas de E/S. Por ejemplo, el tráfico de un navegador web a un servidor web es pequeño en comparación con la cantidad de tráfico que se devuelve desde el servidor en respuesta a la petición URL del cliente. Lo mismo se puede decir para ftp. La cantidad de tráfico de control que pasa a través del puerto 21 es insignificante en comparación con la cantidad de tráfico que pasa a través del puerto 20.

Piense en un equipo concreto de Linux conectado a Internet. La máquina usa el firewall que se presentó en el Capítulo 3. La interfaz de red del equipo tiene una dirección IP estática asignada permanentemente por el ISP. El firewall enmascara las conexiones de Internet, para una LAN privada pequeña.

La máquina ejecuta un servidor DNS local que primero reenvía las peticiones al servidor de nombres del ISP, para continuar con las peticiones de cliente a servidor a cualquier servidor de nombres en el caso de que el servidor de nombres del ISP no responda. Todos los servicios de correo SMTP se ejecutan localmente, pero el firewall también soporta una conexión de cliente POP saliente para recuperar el correo procedente de la cuenta de correo del ISP. El sitio alberga su propio servidor web. No hay servidor FTP público local.

Como se explica en este libro, una secuencia de comandos del firewall para esta máquina debería parecerse a la siguiente:

```
#!/bin/sh

echo "Starting firewalling... "

# Algunas definiciones para mantenimiento sencillo.

# ------------------------------------------------------------------
# MODIFIQUE ESTE CÓDIGO PARA QUE SE AJUSTE
# A SUS NECESIDADES Y PSI.

EXTERNAL_INTERFACE="eth0"              # Interfaz conectada de Internet
INTERNAL_INTERFACE="eth1"              # o la especificación de nombres local
LOOPBACK_INTERFACE="lo"                # de la interfaz LAN interna

IPADDR="mi.dirección.ip"               # su dirección IP
LAN_IPADDR="192.168.1.1"               # cualquier intervalo (privado) que use
LAN_ADDRRESSES="192.168.1.0/24"        # dirección de interfaz interna

ANYWHERE="cualquiera/0"                # coincidir con cualquier dirección IP

MY_ISP="mi.intervalo.direcciones.psi" # intervalo direcciones de PSI y NOC
NAMESERVER_1="mi.nombre.servidor.1"    # todos deben tener al menos uno
```

```
SMTP_SERVER="cualquiera/0"              # servidor de correo externo
SMTP_GATEWAY="mi.servidor.psi"          # envío de correo externo
POP_SERVER="mi.servidor.pop"            # servidor pop externo, si existe
NEWS_SERVER="mi.servidor.noticias"      # servidor de noticias externo, si existe
WEB_PROXY_SERVER="mi.www.proxy"         # servidor proxy web del PSI, si existe
WEB_PROXY_PORT="www.puerto.proxy"       # puerto proxy web del PSI, si existe
                                        # normalmente 8008 o 8080

BUCLE_INVERSO="127.0.0.0/8"             # intervalo reservado de direcciones de
                                        # bucle inverso
CLASS_A="10.0.0.0/8"                    # redes privadas de clase A
CLASS_B="172.16.0.0/12"                 # redes privadas de clase B
CLASS_C="192.168.0.0/16"                # redes privadas de clase C
CLASS_D_MULTICAST="224.0.0.0/4"         # direcciones de difusión múltiple de clase D
BROADCAST_SRC="0.0.0.0"                 # dirección origen de difusión
BROADCAST_DEST="255.255.255.255"        # dirección destino de difusión
PRIVPORTS="0:1023"                      # intervalo de puerto privilegiado, bien
                                        # conocido
UNPRIVPORTS="1024:65535"                # intervalo de puerto no privilegiado
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"
SOCKS_PORT="1080"                       # conectores (TCP)
OPENWINDOWS_PORT="2000"                 # (TCP) openwindows
NFS_PORT="2049"                         # (TCP/UDP) NFS

# ....................................................................
# MODIFIQUE ESTO PARA QUE SE AJUSTE AL NÚMERO DE SERVIDORES
# CONEXIONES QUE SE SOPORTEN.

XWINDOW_PORTS="6000"                    # (TCP) X Window

SSH_PORTS="1020:1023"                   # conexiones simultáneas

# ....................................................................

# Habilitar TCP SYN Cookie Protection
echo 1 >/proc/sys/net/ipv4/tcp_syncookies

# Permite activar la protección IP contra spoofing
# en Source Address Verification
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 > $f
done

# Deshabilitar la aceptación de direcciones ICMP
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    echo 0 > $f
done

# Deshabilitar paquetes de origen enrutado
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
    echo 0 > $f
done

# ....................................................................
```

```
# Eliminar cualquier regla existente de todas las cadenas.
ipchains -F

# Establecer la directiva predeterminada a denegar
ipchains -P input   DENY
ipchains -P output  REJECT
ipchains -P forward REJECT

# ------------------------------------------------------------------
# BUCLE INVERSO

# Tráfico ilimitado en la interfaz de bucle inverso
ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT

# ------------------------------------------------------------------
# Rechazar cualquier conexión de sitios problemáticos.

# /etc/rc.d/rc.firewall.blocked contiene una lista de
# ipchains -A input -i $EXTERNAL_INTERFACE -s <dirección/máscara> -j DENY
# reglas para bloquear todos los accesos.

# Rechazar todos los paquetes que parezcan proceder de la lista prohibida
if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    . /etc/rc.d/rc.firewall.blocked
fi

# ------------------------------------------------------------------
# SPOOFING y DIRECCIONES ERRÓNEAS
# Rechazar los paquetes de spoof.
# Ignorar las direcciones origen de red que sean claramente ilegales.
# Protegerse de realizar envíos a direcciones erróneas.

# Rechazar los paquetes de spoof que pretendan entrar de
# direcciones IP de la interfaz externa.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $IPADDR -j DENY -l

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase A.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase B.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase C.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY

# Rechazar todos los paquetes que parezcan ir o proceder de la interfaz de bucle
# inverso.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $LOOPBACK -j DENY

# Rechazar paquetes de difusión mal formados.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $BROADCAST_DEST -j DENY -l
```

```
ipchains -A input  -i $EXTERNAL_INTERFACE -d $BROADCAST_SRC  -j DENY -l

# Rechazar todos los paquetes que parezcan ir o proceder de direcciones de
# difusión múltiple de clase D.
# La difusión múltiple usa UDP.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST \
         -j DENY -l

# ------------------------------------------------------------------
# ICMP

# (4)  Source_Quench
#      Solicitudes entrantes y salientes para ralentizar (control de flujo)
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 4 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 4 -d $ANYWHERE -j ACCEPT

# (12) Parameter_Problem
#      Mensajes de error entrantes y salientes
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 12 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 12 -d $ANYWHERE -j ACCEPT

# (3)  Dest_Unreachable, Service_Unavailable
#      Incoming & outgoing size negotiation, service or
#      destination unavailability, respuesta final de traceroute
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 3 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 3 -d $MY_ISP -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR fragmentation-needed -d $ANYWHERE -j ACCEPT

# (11) Time_Exceeded
#      Condiciones de tiempo de espera entrantes y salientes,
#      además de respuestas intermedias TTL para trazar rutas
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 11 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 11 -d $MY_ISP -j ACCEPT

# Permitir pings salientes a cualquier sitio.
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 8 -d $ANYWHERE -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $ANYWHERE 0 -d $IPADDR -j ACCEPT
```

```
# Permitir pings entrantes de equipos de confianza.
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -s $MY_ISP 8 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 0 -d $MY_ISP -j ACCEPT

# ------------------------------------------------------------------

# PUERTOS NO PRIVILEGIADOS
# Evitar puertos sujetos a problemas de administración de sistema y protocolo.

# Conexión entrante OpenWindows
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $OPENWINDOWS_PORT -j DENY

# X Window: intento de conexión entrante
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $XWINDOW_PORTS -j DENY -l

# Conexión entrante SOCKS
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $SOCKS_PORT -j DENY

# NFS: Conexiones TCP
ipchains -A input -i $EXTERNAL_INTERFACE -p tcp -y \
         -d $IPADDR $NFS_PORT -j DENY -l

# NFS: Conexiones UDP
ipchains -A input -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR $NFS_PORT -j DENY -l

# ------------------------------------------------------------------

# Modos de cliente DNS (53)
# ----------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d $NAMESERVER_1 53 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER_1 53 \
         -d $IPADDR 53 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NAMESERVER_1 53 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER_1 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# El protocolo permite las solicitudes TCP de cliente a servidor
# si fallan las solicitudes UDP. Esto raramente se ve.
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d <mi.dns.primario> 53 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s <mi.dns.primario> 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# AUTH (113)
# ----------

# Aceptar sus solicitudes AUTH entrantes

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 113 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 113 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# Aceptar sus solicitudes AUTH salientes

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 113 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y  \
         -s $ANYWHERE 113 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# Cliente HTTP (80)
# -------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 80 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 80 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# Servidor HTTP (80)
# ------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 80 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 80 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT
```

```
# ------------------------------------------------------------

# Cliente SSL (443)
# ----------------.

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 443 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 443 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# NNTP (119) - Recibir y enviar noticias como un cliente de Usenet
# ----------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NEWS_SERVER 119 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $NEWS_SERVER 119 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# POP (110) - Obtener correo como cliente POP
# ------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $POP_SERVER 110 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $POP_SERVER 110 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# Cliente SMTP (25)
# ----------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 25 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 25 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# Servidor SMTP (25)
# ----------------------------------
```

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 25 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 25 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# TELNET (23) - Permitir el acceso cliente a sitios remotos
# ------------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 23 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 23 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# Cliente SSH (22) - Permitir el acceso cliente a servidores remotos SSH
# ------------------------------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $SSH_PORTS \
         -d $ANYWHERE 22 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 22 \
         -d $IPADDR $SSH_PORTS -j ACCEPT

# ------------------------------------------------------------------

# FTP (20, 21) - Permitir el acceso cliente saliente a servidores FTP remotos
# ------------------------------------------------------------------

# Solicitud saliente

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 21 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 21 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

```
# Canales de datos FTP de modo de puerto normal

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE 20 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 20 -j ACCEPT

# Canales de datos FTP de modo pasivo

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR $UNPIRVPORTS -j ACCEPT

# ------------------------------------------------------------------

# Cliente WHOIS (43)
# ------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $ANYWHERE 43 -j ACCEPT

ipchains -A input -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $ANYWHERE 43 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# TRACEROUTE
# ----------

# Habilitar solicitudes traceroute salientes
# ------------------------------------

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $TRACEROUTE_SRC_PORTS \
         -d $ANYWHERE $TRACEROUTE_DEST_PORTS -j ACCEPT

# Habilitar solicitudes traceroute entrantes de PSI
# -------------------------------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $MY_ISP $TRACEROUTE_SRC_PORTS \
         -d $IPADDR $TRACEROUTE_DEST_PORTS -j ACCEPT

# ------------------------------------------------------------------

# NTP (123) - Acceder a servidores remotos externos de hora
# ---------------------------------------------------
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
        -s $IPADDR $UNPRIVPORTS \
        -d <mi.proveedor.hora> 123 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s <mi.proveedor.hora> 123 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

echo "done"

exit 0
```

Optimización del orden de las reglas ipfwadm

Además de ordenar las reglas generales, ipfwadm es compatible con un mecanismo que se puede usar para optimizar las reglas de firewall. Los puertos y los tipos de mensajes ICMP pueden expresarse como una lista de hasta diez valores. Un intervalo de valores se trata como dos valores. Un intervalo puede definirse dentro de cualquier lista de puertos destino o puertos origen. Las reglas múltiples pueden contraerse en una sola regla agrupando los puertos o los tipos de mensaje ICMP en una sola lista. Un paquete se comprobará con una sola regla, en vez de con reglas individuales para cada valor de la lista.

A continuación se muestra la secuencia de comandos del firewall del Capítulo 3, optimizado para ipfwadm:

```
#!/bin/sh

echo "Starting firewalling... "

# Algunas definiciones para mantenimiento sencillo.

# ------------------------------------------------------------------
# MODIFIQUE ESTE CÓDIGO PARA QUE SE AJUSTE
# A SUS NECESIDADES Y PSI.

EXTERNAL_INTERFACE="eth0"             # Interfaz conectada de Internet
INTERNAL_INTERFACE="eth1"             # de la interfaz LAN interna
LOOPBACK_INTERFACE="lo"               # o la especificación de nombres local

IPADDR="mi.dirección.ip"              # su dirección IP
LAN_IPADDR="192.168.1.1"              # dirección de interfaz interna
LAN_ADDRRESSES="192.168.1.0/24"       # dirección de red LAN

ANYWHERE="cualquiera/0"               # coincidir con cualquier dirección IP

MY_ISP="mi.intervalo.direcciones.psi" # direcciones de PSI y NOC
NAMESERVER_1="mi.nombre.servidor.1"   # todos deben tener al menos uno

SMTP_SERVER="cualquiera/0"            # servidor de correo externo
SMTP_GATEWAY="mi.servidor.psi"        # envío de correo externo
```

```
POP_SERVER="mi.servidor.pop"          # servidor pop externo, si existe
NEWS_SERVER="mi.servidor.noticias"    # servidor de noticias externo, si existe
WEB_PROXY_SERVER="mi.www.proxy"       # servidor proxy web del PSI, si existe
WEB_PROXY_PORT="www.puerto.proxy"     # puerto proxy web del PSI, si existe
                                      # normalmente 8008 o 8080

LOOPBACK="127.0.0.0/8"                # intervalo reservado de direcciones de
bucle inverso
CLASS_A="10.0.0.0/8"                  # redes privadas de clase A
CLASS_B="172.16.0.0/12"               # redes privadas de clase B
CLASS_C="192.168.0.0/16"              # redes privadas de clase C
CLASS_D_MULTICAST="224.0.0.0/4"       # direcciones de difusión múltiple de clase D
BROADCAST_SRC="0.0.0.0"               # dirección origen de difusión
BROADCAST_DEST="255.255.255.255"      # dirección destino de difusión
PRIVPORTS="0:1023"                    # intervalo de puerto privilegiado, bien
                                      # conocido
UNPRIVPORTS="1024:65535"              # intervalo de puerto no privilegiado
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"
SOCKS_PORT="1080"                     # (TCP) SOCKS
OPENWINDOWS_PORT="2000"               # (TCP) OpenWindows
NFS_PORT="2049"                       # (TCP/UDP) NFS

# ..................................................................
# MODIFIQUE ESTO PARA QUE SE AJUSTE AL NÚMERO DE SERVIDORES
# CONEXIONES QUE SE SOPORTEN.

XWINDOW_PORTS="6000"                  # (TCP) X Window

SSH_PORTS="1020:1023"                 # conexiones simultáneas

# ..................................................................

# Habilitar TCP SYN Cookie Protection
echo 1 >/proc/sys/net/ipv4/tcp_syncookies

# Permite activar la protección IP contra spoofing
# en Source Address Verification
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 > $f
done

# Deshabilitar la aceptación de direcciones ICMP
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    echo 0 > $f
done

# Deshabilitar paquetes de origen enrutado
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
    echo 0 > $f
done

# ..................................................................

# Eliminar cualquier regla existente de todas las cadenas.
```

```
ipfwadm -I -f
ipfwadm -O -f
ipfwadm -F -f

# Establecer la directiva predeterminada a denegar
ipfwadm -I -p deny
ipfwadm -O -p reject
ipfwadm -F -p reject

# ------------------------------------------------------------------
# BUCLE INVERSO

# Tráfico ilimitado en la interfaz de bucle inverso
ipfwadm -I -a accept -W $LOOPBACK_INTERFACE
ipfwadm -O -a accept -W $LOOPBACK_INTERFACE

# ------------------------------------------------------------------
# Rechazar cualquier conexión de sitios problemáticos.

# /etc/rc.d/rc.firewall.blocked contiene una lista de
# ipfwadm -I -a deny -W $EXTERNAL_INTERFACE -S <dirección/mácara>
# reglas para bloquear todos los accesos.

# Rechazar todos los paquetes que parezcan proceder de la lista prohibida
if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    . /etc/rc.d/rc.firewall.blocked
fi

# ------------------------------------------------------------------
# SPOOFING y DIRECCIONES ERRÓNEAS
# Rechazar los paquetes de spoof.
# Ignorar las direcciones origen de red que sean claramente ilegales.

# Rechazar los paquetes de spoof que pretendan entrar de
# direcciones IP de la Interfaz externa.
ipfwadm -I -a deny   -W $EXTERNAL_INTERFACE -S $IPADDR -o

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase A.
ipfwadm -I -a deny   -W $EXTERNAL_INTERFACE -S $CLASS_A -o

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase B.
ipfwadm -I -a deny   -W $EXTERNAL_INTERFACE -S $CLASS_B -o

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase C.
ipfwadm -I -a deny   -W $EXTERNAL_INTERFACE -S $CLASS_C -o

# Rechazar todos los paquetes que parezcan ir o proceder de la interfaz de bucle
# inverso
ipfwadm -I -a deny   -W $EXTERNAL_INTERFACE -S $LOOPBACK -o

# Rechazar paquetes de difusión mal formados.
ipfwadm -I -a deny   -W $EXTERNAL_INTERFACE -S $BROADCAST_DEST -o
```

```
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -D $BROADCAST_SRC -o

# Rechazar todos los paquetes que parezcan proceder de direcciones de difusión
# múltiple de clase D.
# La difusión múltiple usa UDP.
ipfwadm -I -a deny    -W $EXTERNAL_INTERFACE -S $MULTICAST -o

# ------------------------------------------------------------------
# ICMP

# (0)  Echo Reply (pong)
# (3)  Dest_Unreachable, Service_Unavailable
#      Negociación de tamaño entrantes y salientes, no disponibilidad
#      del servicio o destino, respuesta final de traceroute
# (4)  Source_Quench
#      Solicitudes entrantes y salientes para ralentizar (control de flujo)
# (8)  Solicitud de Echo (ping)
# (12) Parameter_Problem

ipfwadm -I -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 0 3 4 11 12 -D $IPADDR

ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 3 4 8 12 -D $ANYWHERE

ipfwadm -I -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $MY_ISP 8 -D $IPADDR

ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 0 11 -D $MY_ISP

# ------------------------------------------------------------------

# Modo de reenviador UDP DNS (53)
# ------------------------------

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 53 \
        -D $NAMESERVER_1 53

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $NAMESERVER_1 53 \
        -D $IPADDR 53

# Modo de cliente UDP DNS (53)
# ------------------------

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 53

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 53 \
        -D $IPADDR $UNPRIVPORTS
```

```
# ------------------------------------------------------------------

# Cliente HTTP (80)
# Cliente SSL (443)
# SMTP - enviando correo (25)
# Cliente AUTH (113)
# Cliente WHOIS (43)
# Respuestas entrantes del servidor
# ----------------------------

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 80 443 25 113 43 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------

# NNTP (119) - Leer y enviar noticias como un cliente de Usenet
# ----------------------------------------------------

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $NEWS_SERVER 119 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------

# POP (110) - Obtener correo como cliente POP
# ------------------------------------------

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $POP_SERVER 110 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------

# Cliente HTTP (80)
# Cliente SSL (443)
# SMTP - enviando correo (25)
# Cliente AUTH (113)
# Cliente WHOIS (43)
# Solicitudes salientes del cliente
# ----------------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 80 443 25 113 43

# ------------------------------------------------------------------

# NNTP (119) - Leer y enviar noticias como un cliente de Usenet
# ----------------------------------------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $NEWS_SERVER 119
```

```
# ------------------------------------------------------------------

# POP (110) - Obtener correo como cliente POP
# ------------------------------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $POP_SERVER 110

# ------------------------------------------------------------------

# Servidor web (80)
# Servidor AUTH (113)
# SMTP - recibiendo correo (25)
# Manejar solicitudes entrantes de cliente

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR 80 113 25 \
        -D $ANYWHERE $UNPRIVPORTS

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR 80 113 25

# ------------------------------------------------------------------

# TELNET (23) - Permitir el acceso cliente saliente a sitios remotos
# Cliente SSH (22) - Permitir el acceso inicial al cliente a servidores remotos SSH
# Cliewnte FTP (21) - Permitir acceso al cliente al canal de comandos a
# servidores remotos FTP
# ------------------------------------------------------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 23 22 21

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 23 22 21 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------

# Cliente SSH (22) - Permitir el acceso inicial al cliente a servidores remotos SSH
# ---------------------------------------------------------------

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $SSH_PORTS \
        -D $ANYWHERE 22

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 22 \
        -D $IPADDR $SSH_PORTS

# ------------------------------------------------------------------
```

```
# PUERTOS NO PRIVILEGIADOS
# Evitar puertos sujetos a problemas de administración de sistema y protocolo.

# Conexión entrante de X Window
# Conexión entrante de SOCKS
# Conexión entrante NFS (modo TCP atípico)

ipfwadm -I -a deny -P tcp -y -W $EXTERNAL_INTERFACE \
        -D $IPADDR $XWINDOW_PORTS $SOCKS_PORT $NFS_PORT

# Solicitud entrante NFS (modo UDP normal)
ipfwadm -I -a deny -P udp -W $EXTERNAL_INTERFACE \
        -D $IPADDR $NFS_PORT

# ------------------------------------------------------------------

# FTP (20, 21) - Permitir el acceso cliente saliente a servidores FTP remotos
# ------------------------------------------------------------------

# Canales de datos FTP de modo pasivo

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE $UNPRIVPORTS \
        -D $IPADDR $UNPRIVPORTS

# Canales de datos FTP de modo de puerto normal

ipfwadm -I -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 20 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------

# FTP (20, 21) - Permitir el acceso cliente saliente a servidores FTP remotos
# ------------------------------------------------------------------

# Canales de datos FTP de modo pasivo

ipfwadm -O -a accept -P tcp     -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE $UNPRIVPORTS

# Canales de datos FTP de modo de puerto normal

ipfwadm -O -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 20

# ------------------------------------------------------------------

# NTP (123) - Acceder a servidores remotos externos de hora
# ------------------------------------------------

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D <mi.proveedor.hora> 123
```

```
ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S <mi.proveedor.hora> 123 \
        -D $IPADDR $UNPRIVPORTS

# ------------------------------------------------------------------

# TRACEROUTE
# ----------

# Habilitar solicitudes traceroute salientes
# ------------------------------------

ipfwadm -O -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $IPADDR $TRACEROUTE_SRC_PORTS \
        -D $ANYWHERE $TRACEROUTE_DEST_PORTS

# Habilitar solicitudes traceroute entrantes del PSI
# --------------------------------------------------

ipfwadm -I -a accept -P udp -W $EXTERNAL_INTERFACE \
        -S $MY_ISP $TRACEROUTE_SRC_PORTS \
        -D $IPADDR $TRACEROUTE_DEST_PORTS

# (11) Time_Exceeded
#      Condiciones de tiempo de espera de salida,

ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 11 -D $MY_ISP

# ------------------------------------------------------------------

# PING
# ----

# permitir pings entrantes desde equipos de confianza
ipfwadm -I -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $MY_ISP 8 -D $IPADDR

ipfwadm -O -a accept -P icmp -W $EXTERNAL_INTERFACE \
        -S $IPADDR 0 -D $MY_ISP

# ------------------------------------------------------------------

# Modo cliente TCP DNS (53)
# -------------------------

# El protocolo permite solicitudes TCP cliente/servidor
# si fallan las solicitudes UDP. Esto raramente se ve.

ipfwadm -I -a accept -P tcp -k -W $EXTERNAL_INTERFACE \
        -S $ANYWHERE 53 \
        -D $IPADDR $UNPRIVPORTS

ipfwadm -O -a accept -P tcp    -W $EXTERNAL_INTERFACE \
        -S $IPADDR $UNPRIVPORTS \
        -D $ANYWHERE 53
```

```
# ------------------------------------------------------------------

echo "done"

exit 0
```

Optimización de las reglas de ipchains con cadenas definidas por el usuario

Además de ordenar las reglas generales, ipchains es compatible con las listas de reglas definidas por el usuario, o cadenas, que se pueden usar para optimizar las reglas de firewall. El paso de un paquete de una cadena a otra basándose en valores del encabezado del paquete ofrece un medio para probar el paquete de manera selectiva con un subconjunto de las reglas input, output o forward, en vez de probar el paquete con todas las reglas de la lista hasta que se encuentra una coincidencia.

Por ejemplo, si se usa la secuencia de comandos del firewall sin inicializar, un paquete entrante procedente de un servidor de tiempo NTP puede encontrarse con aproximadamente 40 reglas antes de que el paquete coincida con su regla ACCEPT. Usando cadenas definidas por el usuario para optimizar el firewall, el mismo paquete entrante se compara con unas 15 reglas antes de que coincida con su regla ACCEPT.

En ipchains, las reglas se usan para pasar paquetes entre cadenas, así como para definir bajo qué condiciones se acepta o se deniega el paquete. Si un paquete no coincide con ninguna regla de la cadena definida por el usuario, se devuelve el control a la cadena que llama. Si el paquete no coincide con una regla de selección de cadena, el paquete no se pasa a dicha cadena para compararlo con las reglas de la cadena. El paquete simplemente se compara con la regla de selección de la siguiente cadena.

A continuación se muestra la secuencia de comandos del firewall del Capítulo 3, optimizada para ipchains:

```
#!/bin/sh

echo "Starting firewalling... "

# Algunas definiciones para mantenimiento sencillo.

# ------------------------------------------------------------------
# MODIFIQUE ESTE CÓDIGO PARA QUE SE AJUSTE
# A SUS NECESIDADES Y PSI.

EXTERNAL_INTERFACE="eth0"              # Interfaz conectada de Internet
LAN_INTERFACE="eth1"                   # interfaz conectada a la LAN
LOOPBACK_INTERFACE="lo"                # o la especificación de nombres local

IPADDR="mi.dirección.ip"               # su dirección IP pública
LAN_IPADDR="192.168.1.1"               # dirección de interfaz interna
LAN_ADDRESSES="192.168.1.0/24"         # dirección de red LAN
```

```
ANYWHERE="cualquiera/0"                 # coincidir con cualquier dirección IP

MY_ISP="mi.intervalo.direcciones.psi"   # direcciones de PSI y NOC
NAMESERVER_1="mi.nombre.servidor.1"     # todos deben tener al menos uno

SMTP_SERVER="cualquiera/0"              # servidor de correo externo
SMTP_GATEWAY="mi.servidor.psi"          # envío de correo externo
POP_SERVER="mi.servidor.pop"            # servidor pop externo, si existe
NEWS_SERVER="mi.servidor.noticias"      # servidor de noticias externo, si existe
WEB_PROXY_SERVER="mi.www.proxy"         # servidor proxy web del PSI, si existe
WEB_PROXY_PORT="www.puerto.proxy"       # puerto proxy web del PSI, si existe
                                        # normalmente 8008 o 8080

LOOPBACK="127.0.0.0/8"                  # intervalo reservado de direcciones de
                                          bucle inverso
CLASS_A="10.0.0.0/8"                    # redes privadas de clase A
CLASS_B="172.16.0.0/12"                 # redes privadas de clase B
CLASS_C="192.168.0.0/16"                # redes privadas de clase
CLASS_D_MULTICAST="224.0.0.0/4"         # direcciones de difusión múltiple de clase D
BROADCAST_SRC="0.0.0.0"                 # dirección origen de difusión
BROADCAST_DEST="255.255.255.255"        # dirección destino de difusión
PRIVPORTS="0:1023"                      # intervalo de puerto privilegiado, bien
                                        # conocido
UNPRIVPORTS="1024:65535"                # intervalo de puerto no privilegiado
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"
SOCKS_PORT="1080"                       # conectores (TCP)
OPENWINDOWS_PORT="2000"                 # (TCP) openwindows
NFS_PORT="2049"                         # (TCP/UDP) NFS

# ------------------------------------------------------------------
# MODIFIQUE ESTO PARA QUE SE AJUSTE AL NÚMERO DE SERVIDORES
# CONEXIONES QUE SE SOPORTEN.

XWINDOW_PORTS="6000"                    # (TCP) X Window

SSH_PORTS="1020:1023"                   # conexiones simultáneas

# ------------------------------------------------------------------

# Habilitar TCP SYN Cookie Protection
echo 1 >/proc/sys/net/ipv4/tcp_syncookies

# Permite activar la protección IP contra spoofing
# en Source Address Verification
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
    echo 1 > $f
done

# Deshabilitar la aceptación de direcciones ICMP
for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
    echo 0 > $f
done

# Deshabilitar paquetes de origen enrutado
```

```
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
    echo 0 > $f
done

# ------------------------------------------------------------------

# Eliminar cualquier regla existente de todas las cadenas
ipchains -F
ipchains -F spoofed
ipchains -F tcp-c-o
ipchains -F tcp-s-i
ipchains -F udp-c-o
ipchains -F udp-s-i
ipchains -F tcp-c-i
ipchains -F tcp-s-o
ipchains -F misc-out
ipchains -F misc-in
ipchains -F icmp-in
ipchains -F icmp-out
ipchains -F log-in
ipchains -F log-out

ipchains -X spoofed
ipchains -X tcp-c-o
ipchains -X tcp-s-i
ipchains -X udp-c-o
ipchains -X udp-s-i
ipchains -X tcp-c-i
ipchains -X tcp-s-o
ipchains -X misc-out
ipchains -X misc-in
ipchains -X icmp-in
ipchains -X icmp-out
ipchains -X log-in
ipchains -X log-out

# Establecer la directiva predeterminada a denegar
ipchains -P input   DENY
ipchains -P output  REJECT
ipchains -P forward REJECT

# ------------------------------------------------------------------
# INTERFAZ DE BUCLE INVERSO

# Tráfico ilimitado en la interfaz de bucle inverso
ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT

# ------------------------------------------------------------------
# INTERFAZ LAN

ipchains -A input  -i $LAN_INTERFACE -j ACCEPT
ipchains -A output -i $LAN_INTERFACE -j ACCEPT

# ------------------------------------------------------------------
```

```
# Cadenas definidas por el usuario

ipchains -N spoofed
ipchains -N tcp-c-o
ipchains -N tcp-s-i
ipchains -N udp-c-o
ipchains -N udp-s-i
ipchains -N tcp-c-i
ipchains -N tcp-s-o
ipchains -N misc-out
ipchains -N misc-in
ipchains -N icmp-in
ipchains -N icmp-out
ipchains -N log-in
ipchains -N log-out

# ------------------------------------------------------------------
# INTERFAZ EXTERNA Spoofed Source Address CHAIN  (spoofed)

# Rechazar cualquier conexión de sitios problemáticos.

# /etc/rc.d/rc.firewall.blocked contiene una lista de
# ipchains -A spoofed -s <dirección/mácara> -j DENY
# reglas para bloquear todos los accesos.

# Rechazar todos los paquetes que parezcan proceder de la lista prohibida
if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    . /etc/rc.d/rc.firewall.blocked
fi

# ------------------------------------------------------------------
# SPOOFING y DIRECCIONES ERRÓNEAS
# Rechazar los paquetes de spoof.
# Ignorar las direcciones origen de red que sean claramente ilegales.
# Protegerse de enviar a direcciones erróneas.

# Rechazar los paquetes de spoof que pretendan proceder de
# direcciones IP de la interfaz externa.
ipchains -A spoofed -s $IPADDR -j DENY -l

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase A.
ipchains -A spoofed -s $CLASS_A -j DENY

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase B.
ipchains -A spoofed -s $CLASS_B -j DENY

# Rechazar todos los paquetes que parezcan ir o proceder de redes privadas de
# clase C.
ipchains -A spoofed -s $CLASS_C -j DENY

# Rechazar todos los paquetes que parezcan ir o proceder de la interfaz de bucle
# inverso
ipchains -A spoofed -s $LOOPBACK -j DENY
```

```
# Rechazar todos los paquetes que parezcan proceder de direcciones de difusión
# múltiple de clase D.
# La difusión múltiple usa UDP.
ipchains -A spoofed -s $CLASS_D_MULTICAST -j DENY -l

# Rechazar paquetes de difusión mal formados.
ipchains -A spoofed -s $BROADCAST_DEST -j DENY -l

# ------------------------------------------------------------------
# INTERFAZ EXTERNA - TCP Client Output CHAIN (tcp-c-o -p tcp)


# Cliente HTTP (80)

ipchains -A tcp-c-o -p tcp \
         -d $ANYWHERE http -j ACCEPT

# ------------------------------------------------------------------

# NNTP (119) - Leer y enviar noticias como un cliente de Usenet

ipchains -A tcp-c-o -p tcp \
         -d $NEWS_SERVER nntp -j ACCEPT

# ------------------------------------------------------------------

# POP (110) - Obtener correo como cliente POP

ipchains -A tcp-c-o -p tcp \
         -d $POP_SERVER pop-3 -j ACCEPT

# ------------------------------------------------------------------

# Cliente SMTP enviando correo (25)

ipchains -A tcp-c-o -p tcp \
         -d $ANYWHERE smtp -j ACCEPT

# ------------------------------------------------------------------

# Cliente AUTH (113)

ipchains -A tcp-c-o -p tcp \
         -d $ANYWHERE auth -j ACCEPT

# ------------------------------------------------------------------

# TELNET (23) - Permitir el acceso cliente saliente a sitios remotos

ipchains -A tcp-c-o -p tcp \
         --destination-port telnet -j ACCEPT

# ------------------------------------------------------------------

# Cliente SSH (22) - Permitir el acceso inicial al cliente a servidores remotos SSH
```

```
ipchains -A tcp-c-o -p tcp \
         --destination-port ssh -j ACCEPT

# ------------------------------------------------------------------

# FTP (20, 21) - Permitir el acceso cliente saliente a servidores FTP remotos

# Canales de datos FTP de modo pasivo

ipchains -A tcp-c-o -p tcp \
         --destination-port $UNPRIVPORTS -j ACCEPT

# Canales de datos FTP de modo de puerto normal

ipchains -A tcp-c-o -p tcp ! -y \
         --destination-port ftp-data -j ACCEPT

# solicitud saliente

ipchains -A tcp-c-o -p tcp \
         --destination-port ftp -j ACCEPT

# ------------------------------------------------------------------

# Cliente SSL (443)

ipchains -A tcp-c-o -p tcp \
         -d $ANYWHERE https -j ACCEPT

# ------------------------------------------------------------------

# Cliente WHOIS (43)

ipchains -A tcp-c-o -p tcp \
         --destination-port whois -j ACCEPT

# ------------------------------------------------------------------

# Modo cliente TCP DNS (53)

# El protocolo permite solicitudes TCP cliente/servidor
# si fallan las solicitudes UDP. Esto raramente se ve.

ipchains -A tcp-c-o -p tcp \
         -d $ANYWHERE domain -j ACCEPT


# ------------------------------------------------------------------
# INTERFAZ EXTERNA - TCP Server Input CHAIN (tcp-s-i -p tcp)

# Cliente HTTP (80)

ipchains -A tcp-s-i -p tcp \
         -s $ANYWHERE http -j ACCEPT
```

```
# ------------------------------------------------------------------

# NNTP (119) - Leer y enviar noticias como un cliente de Usenet

ipchains -A tcp-s-i -p tcp \
        -s $NEWS_SERVER nntp -j ACCEPT

# ------------------------------------------------------------------

# POP (110) - Obtener correo como cliente POP

ipchains -A tcp-s-i -p tcp \
        -s $POP_SERVER pop-3 -j ACCEPT

# ------------------------------------------------------------------

# Cliente SMTP enviando correo (25)

ipchains -A tcp-s-i -p tcp \
        -s $SMTP_GATEWAY smtp -j ACCEPT

# ------------------------------------------------------------------

# Cliente AUTH (113)

ipchains -A tcp-s-i -p tcp \
        -s $ANYWHERE auth -j ACCEPT

# ------------------------------------------------------------------

# TELNET (23) - Permitir el acceso cliente saliente a sitios remotos

ipchains -A tcp-s-i -p tcp \
        —source-port telnet -j ACCEPT

# ------------------------------------------------------------------

# Cliente SSH (22) - Permitir el acceso inicial al cliente a servidores remotos
# SSH

ipchains -A tcp-s-i -p tcp \
        —source-port ssh -j ACCEPT

# ------------------------------------------------------------------

# FTP (20, 21) - Permitir el acceso cliente saliente a servidores FTP remotos

# Canales de datos FTP de modo pasivo

ipchains -A tcp-s-i -p tcp \
        —source-port $UNPRIVPORTS -j ACCEPT

# Solicitud saliente

ipchains -A tcp-s-i -p tcp \
        —source-port ftp -j ACCEPT
```

```
# ------------------------------------------------------------------

# Cliente SSL (443)

ipchains -A tcp-s-i -p tcp \
         -s $ANYWHERE https -j ACCEPT

# ------------------------------------------------------------------

# Cliente WHOIS (43)

ipchains -A tcp-s-i -p tcp \
         —source-port whois -j ACCEPT

# ------------------------------------------------------------------

# Modo cliente TCP DNS (53)

# El protocolo permite solicitudes TCP cliente/servidor
# si fallan las solicitudes UDP. Esto raramente se ve.

ipchains -A tcp-s-i -p tcp \
         -s $ANYWHERE domain -j ACCEPT


# ------------------------------------------------------------------
# INTERFAZ EXTERNA - UDP Client Output CHAIN (udp-c-o)

# Modo de reenviador principal UDP (53)
# -----------------------------

ipchains -A udp-c-o -p udp \
         --source-port domain \
         -d $NAMESERVER_1 domain -j ACCEPT

# Modo cliente UDP (53)
# --------------------

ipchains -A udp-c-o -p udp \
         --source-port $UNPRIVPORTS \
         -d $ANYWHERE domain -j ACCEPT

# ------------------------------------------------------------------

# NTP (123) - Acceder a servidores remotos externos de hora

ipchains -A udp-c-o -p udp \
         --source-port $UNPRIVPORTS \
         -d <mi.proveedor.hora> ntp -j ACCEPT

# ------------------------------------------------------------------
# INTERFAZ EXTERNA - UDP Server Input CHAIN (udp-s-i)

# Modo de reenviador principal UDP (53)
# -----------------------------
```

```
ipchains -A udp-c-o -p udp \
         -s $NAMESERVER_1 domain \
         --destination-port domain -j ACCEPT

# Modo cliente UDP (53)

ipchains -A udp-s-i -p udp \
         -s $ANYWHERE domain \
         --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# NTP (123) - Acceder a servidores remotos externos de hora

ipchains -A udp-s-i -p udp \
         -s <mi.proveedor.hora> ntp \
         --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------
# INTERFAZ EXTERNA - Client Input CHAIN (tcp-c-i -p tcp)

# Servidor HTTP (80)

# Aceptar solicitudes HTTP entrantes

ipchains -A tcp-c-i -p tcp \
         --destination-port http -j ACCEPT

# ------------------------------------------------------------------

# Servidor AUTH (113)

# Aceptar solicitudes AUTH entrantes

ipchains -A tcp-c-i -p tcp \
         --destination-port auth -j ACCEPT

# ------------------------------------------------------------------

# PUERTOS NO PRIVILEGIADOS
# Evitar puertos sujetos a problemas de administración de sistema y protocolo.

# Conexión entrante OpenWindows
ipchains -A tcp-c-i -p tcp -y \
         --destination-port $OPENWINDOWS_PORT -j DENY

# X Window: intento de conexión entrante
ipchains -A tcp-c-i -p tcp -y \
         --destination-port $XWINDOW_PORTS -j DENY -l

# Conexión entrante de SOCKS
ipchains -A tcp-c-i -p tcp -y \
         --destination-port $SOCKS_PORT -j DENY

# NFS: Conexiones TCP
```

```
ipchains -A tcp-c-i -p tcp -y \
        --destination-port $NFS_PORT -j DENY -l


# ------------------------------------------------------------------
# INTERFAZ EXTERNA - Server Output CHAIN (tcp-s-o -p tcp)

# HTTP (80)

# Aceptar solicitudes AUTH entrantes

ipchains -A tcp-s-o -p tcp \
        --source-port http -j ACCEPT

# ------------------------------------------------------------------

# AUTH (113)

# Aceptar solicitudes AUTH entrantes

ipchains -A tcp-s-o -p tcp \
        --source-port auth -j ACCEPT


# ------------------------------------------------------------------
# INTERFAZ EXTERNA - Misc Output CHAIN (misc-out)


# Cliente SSH (22) - Permitir el acceso inicial al cliente a servidores remotos SSH

ipchains -A misc-out -p tcp \
        --source-port $SSH_PORTS \
        --destination-port ssh -j ACCEPT

# ------------------------------------------------------------------

# TRACEROUTE

# Habilitar solicitudes traceroute salientes a cualquier sitio

ipchains -A misc-out -p udp \
        --source-port $TRACEROUTE_SRC_PORTS \
        --destination-port $TRACEROUTE_DEST_PORTS -j ACCEPT

# Habilitar solicitudes traceroute entrantes del PSI

# (3)  Dest_Unreachable, respuesta final de traceroute
# (11) Time_Exceeded - intermediate traceroute response

ipchains -A misc-out -p icmp \
        --icmp-type port-unreachable -d $MY_ISP -j ACCEPT

ipchains -A misc-out -p icmp \
        --icmp-type time-exceeded -d $MY_ISP -j ACCEPT
```

```
# ------------------------------------------------------------------

# PING

# Permitir pings de salida a equipos de confianza
ipchains -A misc-out -p icmp \
         --icmp-type echo-reply -d $MY_ISP -j ACCEPT

# ------------------------------------------------------------------
# INTERFAZ EXTERNA - Misc Input CHAIN (misc-in)


# Cliente SSH (22) - Permitir el acceso inicial al cliente a servidores remotos SSH

ipchains -A misc-in -p tcp ! -y \
         --source-port ssh \
         --destination-port $SSH_PORTS -j ACCEPT

# ------------------------------------------------------------------

# FTP (20, 21) - Permitir el acceso cliente saliente a servidores FTP remotos

# Canales de datos FTP de modo de puerto normal

ipchains -A misc-in -p tcp \
         --source-port ftp-data \
         --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# NFS: Conexiones UDP
ipchains -A misc-in -p udp \
         --destination-port $NFS_PORT -j DENY -l

# ------------------------------------------------------------------

# TRACEROUTE

# Habilitar solicitudes entrantes traceroute del PSI

ipchains -A misc-in -p udp \
         -s $MY_ISP $TRACEROUTE_SRC_PORTS \
         --destination-port $TRACEROUTE_DEST_PORTS -j ACCEPT

# ------------------------------------------------------------------

# PING

# Permitir pings de entrada a equipos de confianza
ipchains -A misc-in -p icmp \
         -s $MY_ISP echo-request -j ACCEPT


# ------------------------------------------------------------------
# CADENA ICMP ENTRANTE
```

```
ipchains -A icmp-in -p icmp --icmp-type echo-reply -j ACCEPT
ipchains -A icmp-in -p icmp --icmp-type destination-unreachable -j ACCEPT
ipchains -A icmp-in -p icmp --icmp-type source-quench -j ACCEPT
ipchains -A icmp-in -p icmp --icmp-type time-exceeded -j ACCEPT
ipchains -A icmp-in -p icmp --icmp-type parameter-problem -j ACCEPT


# ------------------------------------------------------------------
# CADENA ICMP SALIENTE

ipchains -A icmp-out -p icmp --icmp-type fragmentation-needed -j ACCEPT
ipchains -A icmp-out -p icmp --icmp-type source-quench -j ACCEPT
ipchains -A icmp-out -p icmp --icmp-type echo-request -j ACCEPT
ipchains -A icmp-out -p icmp --icmp-type parameter-problem -j ACCEPT


# ------------------------------------------------------------------
# Filtrado de paquetes

ipchains -A input  -i $EXTERNAL_INTERFACE -j spoofed
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp  ! -y \
         -d $IPADDR $UNPRIVPORTS -j tcp-s-i
ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -d $IPADDR -j udp-s-i
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
         -d $IPADDR -j icmp-in
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR -j tcp-c-i
ipchains -A input  -i $EXTERNAL_INTERFACE \
         -d $IPADDR -j misc-in
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR \
         -d $ANYWHERE $UNPRIVPORTS -j tcp-s-o
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS -j tcp-c-o
ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR -j udp-c-o
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR -j icmp-out
ipchains -A output -i $EXTERNAL_INTERFACE \
         -s $IPADDR -j misc-out

# ------------------------------------------------------------------
# CADENA DE REENVÍO Y ENMASCARAMIENTO

ipchains -A forward -i $EXTERNAL_INTERFACE -s $LAN_ADDRESSES -j MASQ

# ------------------------------------------------------------------
# paquetes de acceso denegado

ipchains -A input  -i $EXTERNAL_INTERFACE -j log-in

ipchains -A output -i $EXTERNAL_INTERFACE -j log-out
```

```
# ------------------------------------------------------------------

echo "done"

exit 0
```

Secuencias de comandos compatibles de propósito especial

Es posible que alguien me escriba periódicamente y pregunte si dispongo de una secuencia de comandos para realizar alguna función especial, como puede ser permitir todo o denegar (casi) todo. A continuación se muestran las cuatro secuencias de comandos que más se suelen solicitar.

Permitir todo

La secuencia de comandos accept-all (permitir todo) es una de depuración. De vez en cuando, puede que sea necesario comprobar si un servicio en particular está fuera de servicio debido a algo que están haciendo las reglas.

Una descripción de la secuencia de comandos accept-all, con el formato de una página man es:

NOMBRE:

accept-all Permitir todo el tráfico de red.

SINOPSIS:

accept-all

DESCRIPCIÓN:

Elimina cualquier regla actual de firewall y establece el valor predeterminado a aceptar.

Se debe ejecutar como raíz.

La secuencia de comandos del shell accept-all contiene:

```
#!/bin/sh

# Eliminar todas las reglas que pertenezcan a esta cadena
ipchains -F input
ipchains -F output
ipchains -F forward

# Establecer la directiva predeterminada de la cadena para aceptar.
ipchains -P input   ACCEPT
ipchains -P output  ACCEPT
ipchains -P forward ACCEPT
```

Denegar todo

La secuencia de comandos deny-all (permitir todo) es en parte una herramienta de depuración, en parte un error de detención cuando una secuencia de comandos del nuevo firewall contiene errores sintácticos y no está completo, y en parte una medida contra el pánico. De vez en cuando, puede que sea necesario bloquear el acceso desde Internet y volver a empezar.

Una descripción de la secuencia de comandos deny-all, formateada como una página man sería:

NOMBRE:
deny-all Bloquear el tráfico de red de Internet.

SINOPSIS:
deny-all

DESCRIPCIÓN:
Eliminar cualquier regla de firewall actual y establecer las directivas predeterminadas para la interfaz de red externa a denegar.
Se debe ejecutar como raíz.
La secuencia de comandos del shell deny-all contiene:

```
#!/bin/sh

LOOPBACK_INTERFACE="lo"
LAN_INTERFACE="eth1"

# Eliminar todas las reglas que pertenezcan a esta cadena
ipchains -F input
ipchains -F output
ipchains -F forward

# Establecer la directiva predeterminada de la cadena para denegar
ipchains -P input   DENY
ipchains -P output  REJECT
ipchains -P forward REJECT

ipchains -A input  -i $LOOPBACK_INTERFACE  -j ACCEPT
ipchains -A output -i $LOOPBACK_INTERFACE  -j ACCEPT

# Habilitar el acceso a la máquina firewall desde la LAN
ipchains -A input  -i $LAN_INTERFACE  -j ACCEPT
ipchains -A output -i $LAN_INTERFACE  -j ACCEPT
```

Cómo bloquear una dirección IP

block-ip es útil si un sitio remoto no responde correctamente y se quieren bloquear los paquetes procedentes de este sitio de forma inmediata, sin tener que perder tiempo en modificar la secuencia de comandos del firewall y reinicializar el firewall. block-ip inserta una regla de firewall en la cadena input y además agrega la regla al archivo que contiene cualquier otro sitio que se está bloqueando.

Una descripción de block-ip, formateada como una página man es:

NOMBRE:

block-ip Deniega paquetes de entrada de una dirección.

SINOPSIS:

block-ip <*dirección*>[/máscara]

DESCRIPCIÓN:

Inserta una regla de denegación de firewall en el encabezado de la cadena input. *dirección* puede ser un nombre de equipo completo, una dirección IP o un intervalo de direcciones IP especificadas como una dirección y una máscara.

Si /etc/rc.d/rc.firewall.blocked no existe, se crea. La regla de firewall también se anexa al final de /etc/rc.d/rc.firewall.blocked por lo que la regla está permanentemente incluida en la lista bloqueada.

Se debe ejecutar como root.

La secuencia de comandos del shell block-ip contiene:

```
#!/bin/sh

EXTERNAL_INTERFACE="eth0"

# Si el archivo bloqueado no existe, crearlo.
if [ ! -f /etc/rc.d/rc.firewall.blocked ]; then
    touch /etc/rc.d/rc.firewall.blocked
fi

# insertar la regla en el encabezado de la cadena de entrada existente
ipchains -i input -i $EXTERNAL_INTERFACE -s $1 -j DENY

# anexar la nueva regla al final del archivo bloqueado
echo ipchains -i input -i $EXTERNAL_INTERFACE -s $1 -j DENY \
     >> /etc/rc.d/rc.firewall.blocked
```

Cómo desbloquear una dirección IP

Si quiere eliminar una regla deny (denegar) de la lista de sitios remotos denegados globalmente, sin perder tiempo modificando la secuencia de comandos del firewall y reinicializando el firewall, unblock-ip es útil para eliminar una regla de firewall existente de la cadena input. Se debe eliminar manualmente la regla del archivo /etc/rc.d/rc.firewall.blocked.

Una descripción de unblock-ip, con el formato de una página man es:

NOMBRE:

unblock-ip Elimina una regla DENY de firewall.

SINOPSIS:

unblock-ip <*dirección*>[/máscara]

DESCRIPCIÓN:

Elimina una regla existente de denegación de firewall de la cadena input.*dirección* puede ser un nombre de equipo completo, una dirección IP o un intervalo de direcciones IP especificadas como una dirección y una máscara. *dirección* debe coincidir exactamente con el argumento que se usa para crear la regla de denegación.

Debe ejecutarse como root.

La secuencia de comandos del shell unblock-ip contiene:

```
#!/bin/sh

EXTERNAL_INTERFACE="eth0"

ipchains -D input -i $EXTERNAL_INTERFACE -s $1 -j DENY
```

DHCP: compatibilidad del firewall con una dirección IP dinámica y servidores de nombres

El programa cliente DHCP de Linux ha sido siempre dhcpcd. Con la versión 6.0 de Red Hat, se ha reemplazado dhcpcd con un nuevo programa cliente, pump. Sin embargo, dhcpcd todavía se incluye con las versiones actuales de Linux. Aunque el nuevo programa pump funciona muy bien para una conexión telefónica ppp, pump no es muy compatible con una conexión de red permanente. Los sitios con una conexión de modem o DSL, por ejemplo, deben seguir usando dhcpcd.

DHCP ofrece la dirección IP pública, las direcciones del servidor de nombres, la máscara de red, la dirección de difusión y probablemente la dirección del servidor DHCP. Esta información puede cambiar cada vez que se inicie la máquina, así como al final de cada periodo de concesión DHCP. Cuando se inicializa el firewall, la secuencia de comandos debe incluir las direcciones IPADDR y NAMESERVER. La información que proporciona DHCP se almacena en lugares diferentes, dependiendo de la versión de Linux que se utilice. Puede insertar los siguientes comandos shell en la secuencia de comandos del firewall en algún punto después de cualquier definición introducida en el código de IPADDR, NAMESERVER y DHCP_SERVER:

```
#  La dirección IP, $IPADDR, se define mediante dhcp

if [ -f /etc/dhcpc/hostinfo-$EXTERNAL_INTERFACE ]; then
    # Red Hat Release < 6.0 - dhcpcd
    . /etc/dhcpc/hostinfo-$EXTERNAL_INTERFACE
elif [ -f /etc/dhcpc/dhcpcd-$EXTERNAL_INTERFACE.info ]; then
    # Red Hat Release 6.0 - dhcpcd
    . /etc/dhcpc/dhcpcd-$EXTERNAL_INTERFACE.info
    DHCP_SERVER=$DHCPSIADDR
```

```
    else
        echo "rc.firewall:  dhcp no está configurado."
        ipchains -F
        ipchains -P input  DENY
        ipchains -P output DENY
        ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
        ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT
        ipchains -A input  -i $LAN_INTERFACE -j ACCEPT
        ipchains -A output -i $LAN_INTERFACE -j ACCEPT
        exit 1
    fi

    # Si se usa la secuencia de comandos de ejemplo ifdhcpc-done,
# cualquier definición anterior de
    # IPADDR, NAMESERVER y DHCP_SERVER se omitirá aquí.
```

dhcpcd-${DISPOSITIVO}.exe (para versiones de Red Hat posteriores a la 6.0)

En la versión 6.0 de Red Hat, se ha reconstruido el programa cliente /sbin/dhcpcd. Hay unas cuantas diferencias sutiles entre el dhcpcd actual y las versiones anteriores.

El mecanismo para especificar el programa que se quiere ejecutar después de que dhcpcd consiga una nueva dirección IP es crear un archivo ejecutable llamado /etc/dhcpc/dhcpcd-${DISPOSITIVO}.exe, donde ${DISPOSITIVO} es el nombre de la interfaz de red conectada a Internet, por ejemplo, eth0 o eth1. No se recomienda despreciar la opción -c de dhcpcd.

Al contrario que en las versiones anteriores, el dhcpcd actual no ejecuta el archivo dhcpcd-eth0.exe al iniciarse si la asignación de la nueva dirección IP es la misma dirección que ejecutó la última vez dhcpcd. dhcpcd-eth0.exe se ejecuta sólo cuando la dirección IP cambia. Esto es una importante, pero sutil, diferencia con las versiones anteriores de dhcpcd.

Las secuencias de comandos de instalación de Linux no crean el directorio /etc/dhcpc. El usuario debe crear el directorio:

```
mkdir /etc/dhcpc
```

/etc/resolv.conf se modifica directamente. dhcpcd crea una copia de seguridad del archivo, /etc/resolv.conf.sv, antes de escribir en el archivo /etc/resolv.conf. dhcpcd restaura el archivo resolv.conf a partir de la copia de seguridad cuando el programa termina. La forma de tratar el archivo en las versiones anteriores variaba según la versión. Algunas escribían directamente en el archivo resolv.conf. Otras escribían en un archivo, /etc/dhcpc/resolv.conf, y esperaban que /etc/resolv.conf fuese un vínculo simbólico al archivo /etc/dhcpc/resolv.conf.

La información que recibe dhcpcd desde el servidor DHCP se almacena en /etc/dhcpc/dhcpcd-${DISPOSITIVO}.info, donde ${DISPOSITIVO} es el nom-

bre de la interfaz de red, por ejemplo, eth0 o eth1. La información se almacena como asignaciones de variables de shell, de forma que se puede incluir el archivo en otras secuencias de comandos. Por ejemplo, el archivo contiene definiciones para IPADDR, NETMASK, NETWORK, BROADCAST, GATEWAY, DOMAIN y para la dirección del servidor DHCP, DHCPSIADDR.

Actualización de direcciones dinámicas e instalación del firewall desde el archivo /etc/dhcpc/dhcpcd-${DISPOSITIVO}.exe

Como el archivo dhcpcd-eth0.exe se ejecuta cada vez que cambia la dirección IP, el archivo de secuencia de comandos es un sitio práctico desde el que volver a ejecutar la secuencia de comandos del firewall. Si las reglas de firewall no se reinicializan con la nueva dirección IP, el firewall propio se bloqueará y no permitirá enviar tráfico de red a Internet.

La secuencia de comandos es también un sitio práctico para actualizar cualquier dato dinámico que se use en otros archivos de configuración. La siguiente secuencia de comandos de ejemplo actualiza la dirección IP en el archivo /etc/hosts. Éste copia las direcciones del servidor de nombres del archivo /etc/resolv.conf al archivo /etc/dhcpc/dhcpcd-eth0.info como asignaciones de variable de shell que puede usar la secuencia de comandos del firewall. Si ejecuta un servidor de nombres de reenvío, la secuencia de comandos actualiza el archivo /etc/named.conf con las direcciones del servidor de nombres y actualiza el vínculo simbólico /etc/resolv.conf para que apunte al servidor de nombres local:

```
#!/bin/bash


# Obtener la información del equipo
. /etc/dhcpc/dhcpcd-eth0.info

# Actualizar el nombre de dominio
domainname $DOMAIN

# Actualizar /etc/hosts
# Reemplazar <you> con su nombre de equipo.
sed -e "s/^.*<you>/$IPADDR          <you>.$DOMAIN      <you>/" /etc/hosts \
        > /var/tmp/hosts
cp /var/tmp/hosts /etc/hosts
rm /var/tmp/hosts

------------------------------------------------------------
# Actualizar /etc/rc.d/rc.firewall con los servidores de nombre actuales
# de /etc/resolv.conf.

# Las dos siguientes secciones son compatibles con un servidor de nombres
# guardado únicamentre en caché.
# /etc/named.conf, /etc/dhcpc/resolv.conf y /etc/resolv.conf are modified.
```

```
    # Actualizar los reenviadores /etc/named.conf para que sean los nombres
    # originales de servidores de /etc/resolv.conf.

    let cnt=1
    fgrep nameserver /etc/resolv.conf ¦ sed -e "s/nameserver //" ¦
        while read naddr
        do
           echo "NAMESERVER_$cnt=\ "$naddr\ "" >> /etc/dhcpc/dhcpcd-eth0.info
            let cnt=$cnt+1
        done

        . /etc/dhcpc/dhcpcd-eth0.info

        awk '/forwarders { / {  active = 1; print }            \
                        /} ;/ {                                \
                            if ( active == 1 ) {               \
                                active = 0;                    \
                                print "\ t\ t" f1 ";";         \
                                print "\ t\ t" f2 ";";         \
                                print "\ t\ t" f3 ";";         \
                            }                                  \
                        }                                      \
                                                               \
                        {                                      \
                            if ( active == 0 )                 \
                                print;                         \
                        }                                      \
        f1="$NAMESERVER_1" f2="$NAMESERVER_2" f3="$NAMESERVER_3" /etc/named.conf \
           > /var/tmp/named.conf

    cp /var/tmp/named.conf /etc/named.conf
    rm /var/tmp/named.conf

    ------------------------------------------------------------------------
    # Actualizar /etc/resolv.conf para guardar el servidor de nombres en la
    # caché local.
    # Reemplazar los servidores de nombres con una entrada para el servidor
    # local.

    # Guardar el original
    cp /etc/resolv.conf /etc/dhcpc/resolv.conf.bak

    echo "domain $DOMAIN" > /var/tmp/resolv.conf
    echo "nameserver 127.0.0.1" >> /var/tmp/resolv.conf

    cp /var/tmp/resolv.conf /etc/dhcpc/resolv.conf
    rm /var/tmp/resolv.conf

    ------------------------------------------------------------------------

    sh /etc/rc.d/rc.firewall
    echo "Firewalling enabled." > /dev/console

    cp /etc/dhcpc/resolv.conf /etc

    exit 0
```

Compatibilidad DHCP con /sbin/ifup

En la versión 6.0 de Red Hat, pump es el cliente DHCP predeterminado, no dhcpcd. Para usar dhcpcd, se debe modificar la secuencia de comandos de inicialización de la interfaz de red, `/sbin/ifup`. Se deben reemplazar las líneas en el bloque if $PUMP con las siguientes:

```
if [ -n "$PUMP" ]; then
     echo -n "Determinando la información IP de $DEVICE..."
     if /sbin/dhcpcd; then
          /etc/rc.d/rc.firewall
     else
        echo "failed."
        exit 1
     fi
else
```

Si dhcpcd detecta una nueva dirección IP al iniciarse, ejecutará el archivo /etc/dhcpc/dhcpcd-eth0.exe. Si se activa dhcpcd al iniciarse, el firewall se inicializa desde /sbin/ifup. Si dhcpcd se activa pero la dirección IP no ha cambiado, el firewall se instala desde /sbin/ifup, pero no se ejecuta el archivo /etc/dhcpc/dhcpcd-eth0.exe. Si dhcpcd se activa y la dirección IP ha cambiado, la secuencia de comandos del firewall se ejecuta dos veces, primero desde /etc/dhcpc/dhcpcd-eth0.exe y luego desde /sbin/ifup.

ifdhcpc-done (anterior a la versión 6.0 de Red Hat)

Antes de la versión 6.0 de Red Hat, el mecanismo cliente de DHCP era algo diferente de lo que es ahora. Cuando se iniciaba desde /sbin/ifup, dhcpcd se ejecutaba con la opción -c /etc/sysconfig/network-scripts/ifdhcpc-done. dhcpcd ejecuta ifdhcpc-done cada vez que se inicia la máquina y siempre que la dirección IP cambia dinámicamente.

Las secuencias de comandos pueden crear o no el directorio /etc/dhcpc de la instalación de Linux. Si el directorio no existe, el propio usuario debe crearlo:

```
mkdir /etc/dhcpc
```

/etc/resolv.conf no se modifica directamente. La información del servidor de nombres se escribe en un archivo, /etc/dhcpc/resolv.conf. Se espera a que /etc/resolv.conf sea un vínculo simbólico al archivo /etc/dhcpc/resolv.conf, o la secuencia de comandos ifdhcpc-done copie el archivo /etc/dhcpc/resolv.conf a /etc/resolv.conf.

La información que recibe dhcpcd desde el servidor DHCP se almacena en /etc/dhcpc/hostinfo-${DEVICE}, donde ${DEVICE} es el nombre de la interfaz de red, como, por ejemplo, eth0 o eth1. La información se almacena como asignaciones de variable de shell, de forma que el archivo puede incluirse en

otras secuencias de comandos. Por ejemplo, el archivo contiene definiciones para IPADDR, NETMASK, BROADCAST y GATEWAY.

Actualización de direcciones dinámicas e instalación del firewall a partir de la secuencia de comandos /etc/sysconfig/network-scripts/ifdhcpc-done

El propósito de la secuencia de comandos ifdhcpc-done es ejecutarse en la inicialización y cada vez que la dirección IP cambia. Espera hasta que dhcpcd finaliza su inicialización y luego copia la información del servidor de nombres a /etc/resolv.conf.

La secuencia de comandos es un sitio práctico para actualizar cualquier dato dinámico que se use en otros archivos de configuración. La siguiente secuencia de comandos de ejemplo actualiza la dirección IP en el archivo /etc/hosts. Éste copia las direcciones del servidor de nombres desde /etc/dhcpc/resolv.conf a /etc/dhcpc/hostinfo-eth0 como asignaciones de variables del shell que puede usar la secuencia de comandos del firewall. Si se ejecuta un servidor de nombres de reenvío, la secuencia de comandos actualiza /etc/named.conf o /etc/named.boot con las direcciones del servidor de nombres y actualiza /etc/dhcpc/resolv.conf para que apunte al servidor de nombres local. /etc/dhcpc/resolv.conf se copia a /etc/resolv.conf. Por último, se ejecuta la secuencia de comandos del firewall, /etc/rc.d/rc.firewall:

```
#!/bin/sh

SLEEPPIDFILE=/var/run/dhcp-wait-${ IFNAME} .pid

# Si no existe el archivo de espera, se sobrepasó el tiempo de espera
# del primario o el servidor dhcp dispone de una nueva dirección IP.

if [ -f $SLEEPPIDFILE ]; then
    SLEEPPID=`cat $SLEEPPIDFILE`
    rm -f $SLEEPPIDFILE
    kill $SLEEPPID
fi

# Obtener el pid del proceso, que está esperando a esto para finalizar.

----------------------------------------------------------------
# Actualizar hostinfo
. /etc/dhcpc/hostinfo-eth0

# Actualizar domainname
domain=`fgrep domain /etc/dhcpc/resolv.conf ¦ sed -e "s/domain //"`
domainname $domain

# Actualizar /etc/hosts
# Algunos servicios dejarán de funcionar con esto, a menos que use localhost

# Reemplazar <you> con su nombre de equipo.
```

```
sed -e "s/^.*<you>/$IPADDR          <you>.$domain      <you>/" /etc/hosts \
      > /var/tmp/hosts
cp /var/tmp/hosts /etc/hosts
rm /var/tmp/hosts

------------------------------------------------------------
# Actualizar $hostinfo con los servidores de nombres actuales de /etc/resolv.conf.
# Gracias a Roger Goun por la idea de anexar estos nombres a $hostinfo y
# quitar temporalmente de en medio el archivo.

let cnt=1
fgrep nameserver /etc/dhcpc/resolv.conf ¦ sed -e "s/nameserver //" ¦
    while read naddr
    do
        echo "NAMESERVER_$cnt=\ "$naddr\ "" >> /etc/dhcpc/hostinfo-eth0
        let cnt=$cnt+1
    done

   ------------------------------------------------------------
# Las dos siguientes secciones son compatibles con un servidor de nombres
# guardado únicamentre en caché.
# La primera sección es compatible con BIND incluido en Red Hat 5.1 y anterior.
# La segunda sección es compatible con BIND incluido en Red Hat 5.2 y posterior.

# Actualizar los reenviadores /etc/named.boot o /etc/named.conf para que
# sean los servidores de
# nombres originales de /etc/resolv.conf.

# ------------------------------------------------------------

# Nota: Esto se aplica a la versión Red Hat 5.1 y anteriores.

#forwarders=""
fgrep nameserver /etc/dhcpc/resolv.conf ¦ sed -e "s/nameserver //" ¦
    while read naddr
    do
       forwarders=$forwarders" "$naddr
    done

sed -e "s/^forwarders.*$/forwarders    $forwarders/" /etc/named.boot \
    > /var/tmp/named.boot
#p /var/tmp/named.boot /etc/named.boot
rm /var/tmp/named.boot

# ------------------------------------------------------------

# Nota: Esto se aplica a la versión Red Hat 5.2 y posteriores.

let cnt=1
fgrep nameserver /etc/dhcpc/resolv.conf ¦ sed -e "s/nameserver //" ¦
    while read naddr
    do
        case $cnt in
            1 ) forwarder1=$naddr ;;
            2 ) forwarder2=$naddr ;;
            3 ) forwarder3=$naddr ;;
        esac
```

```
        let cnt=$cnt+1
    done

awk '/forwarders { / {  active = 1; print }                    \
                     /} ;/ {                                   \
                         if ( active == 1 ) {                  \
                             active = 0;                       \
                             print "              " f1 ";"; \
                             print "              " f2 ";"; \
                             print "              " f3 ";"; \
                         }                                     \
                     }                                         \
                                                               \
                     {                                         \
                         if ( active == 0 )                    \
                             print;                            \
                     }  '                                      \
    f1=$forwarder1 f2=$forwarder2 f3=$forwarder3 \
    /etc/named.conf > /tmp/named.conf

cp /var/tmp/named.conf /etc/named.conf
rm /var/tmp/named.conf

# ------------------------------------------------------------------
# Actualizar /etc/resolv.conf para guardar el servidor de nombres en la caché local.
# Reemplazar los servidores de nombres con una entrada para el equipo local.

# Guardar el original
cp /etc/dhcpc/resolv.conf /etc/dhcpc/resolv.conf.bak

echo "domain $domain" > /var/tmp/resolv.conf
echo "search $domain firewall.lan" > /var/tmp/resolv.conf
echo "nameserver 127.0.0.1" >> /var/tmp/resolv.conf

cp /var/tmp/resolv.conf /etc/dhcpc/resolv.conf
rm /var/tmp/resolv.conf

# ------------------------------------------------------------------

cp /etc/dhcpc/resolv.conf /etc

sh /etc/rc.d/rc.firewall
echo "Firewalling enabled." > /dev/console
```

C

Glosario

Este glosario define términos y acrónimos que se usan en el libro. Los términos con varias palabras están ordenados alfabéticamente por el nombre principal del término, seguido de una coma y del resto del término.

ACCEPT Una regla de decisión de filtrado de firewall para pasar el paquete a través del firewall hasta el siguiente destino.

ACK El indicador de TCP que confirma la recepción de un segmento TCP recibido anteriormente.

ataque por denegación de servicio Un ataque basado en la idea del envío de datos inesperados o de inundar un sistema con paquetes para interrumpir o ralentizar seriamente su conexión a Internet, disminuyendo el rendimiento de los servidores hasta el extremo de que no es posible atender las peticiones legítimas, o en el peor de los casos, bloqueando todo el sistema.

autenticación El proceso de determinar que una entidad es quien o lo que dice ser.

AUTH Puerto de servicio TCP 113, asociado con el servidor de autenticación de usuario identd.

autorización El proceso de determinar los recursos y servicios que puede usar una entidad.

bastión Véase firewall, bastión.

BIND *Berkeley Internet Name Domain*, (Dominio de nombres Internet de Berkeley), la implementación del UNIX de Berkeley del protocolo DNS.

BOOTP *Bootstrap Protocol* (Protocolo de secuencia de inicio); que usan las estaciones sin disco para averiguar su dirección IP y la ubicación del servidor de inicio, así como para iniciar la descarga del sistema sobre tftp antes de iniciar el sistema. BOOTP se ha desarrollado para reemplazar a RARP.

BOOTPC Puerto de servicio UDP 68, asociado con los clientes BOOTP y DHCP.

bootpd El programa servidor BOOTP.

BOOTPS Puerto de servicio UDP 67, asociado con los servidores BOOTP y DHCP.

cadena Lista de reglas que define los paquetes que entran y salen a través de una interfaz de red.

capa de red En el modelo de referencia OSI, la tercera capa, que representa la comunicación punto a punto entre dos equipos, como el enrutamiento y la entrega de un datagrama IP desde un equipo origen hacia algunos equipos destino externos. En el modelo de referencia TCP/IP, se conoce como la segunda capa, la capa de Internet.

capa de subred En el modelo de referencia TCP/IP, la primera capa, que representa el medio físico que se usa para transportar las señales eléctricas entre dos dispositivos de red adyacentes, como cable de cobre, fibra óptica, tasa de paquetes o infrarrojos. También incluye el envío de señales de datos punto a punto entre dos dispositivos de red adyacentes, como el envío de una trama Ethernet de un equipo al enrutador externo.

capa de transporte En el modelo de referencia OSI, la cuarta capa, que representa la comunicación punto a punto entre dos programas, como el envío de un paquete de un programa cliente a un programa servidor. En el modelo de referencia TCP/IP, esto se conoce como la tercera capa, también la capa de transporte. Sin embargo, el nivel de abstracción de la capa de transporte 3 de TCP/IP incluye el concepto de la quinta capa de sesión de la capa OSI, que incluye los conceptos de un intercambio de mensajes ordenado y sincronizado.

capa de vínculo de datos En el modelo de referencia OSI, la segunda capa, que representa la entrega de señal de datos punto a punto entre dos dispositivos de red adyacentes, como la entrega de una trama Ethernet desde un equipo a nuestro enrutador externo (en el modelo de referencia TCP/IP, se incluye esta funcionalidad como parte de la primera capa, la capa de subred).

capa física En el modelo de referencia OSI, la primera capa, que representa el medio físico que se usa para transportar las señales eléctricas entre dos dispositivos de red adyacentes, como un cable de cobre, fibra óptica, paquete de radio o infrarrojos. En el modelo de referencia TCP/IP, se incluye como parte de la primera capa, la capa subred.

CERT *Computer Emergency Response Team*, (Equipo informático de respuestas de emergencia), un centro de coordinación de información de seguridad de Internet creado en el Instituto de ingeniería de software de la Universidad Carnegie Mellon después del incidente Internet Worm, en 1988.

CGI *Common gateway interface*, (Interfaz de pasarela común), los programas CGI son programas locales que ejecuta el servidor web en beneficio del cliente remoto. Los programas CGI suelen ser secuencias de comandos Perl, por lo que estos programas se llaman secuencias de comandos, o guiones, CGI.

circuito de pasarela Véase proxy, pasarela de circuito.

clase, dirección de red Una de las cinco clases de direcciones de red. Una dirección IPv4 es un valor de 32 bit. El espacio de direcciones se divide en direcciones de clase A hasta E, en función del valor del los 4 bits más significativos del valor de 32 bits. El espacio de direcciones de red de clase A asigna 126 redes independientes, donde cada una direcciona más de 16 millones de host. El espacio de direcciones de red de clase B asigna redes de 16 K, donde cada una direcciona hasta 64 K host. El espacio de direcciones de red de clase C asigna 2 millones de redes, donde cada una direcciona hasta 254 host. La clase D se usa para direcciones multidifusión. La clase E está reservada para propósitos experimentales no especificados.

Concentrador Un hardware repetidor de señales que se usa para conectar físicamente múltiples segmentos de red, extender la distancia de una red física o conectar segmentos de red de distintos tipos físicos.

contención Véase firewall, de contención.

COPS *Computer Oracle and Password System*, un conjunto de programas que comprueban un gran conjunto de áreas de seguridad, potencialmente vulnerables, en un sistema UNIX.

Crack Un programa de adivinación de contraseñas.

cron Un sistema demonio llamado crond y unos archivos de configuración y secuencias de comandos que ponen en marcha tareas del sistema programadas.

chroot Tanto un programa como una llamada de sistema que define que un directorio es la raíz del sistema de archivos y, a continuación, ejecuta un programa de forma restringida a ese sistema de archivos virtual.

DARPA *Defense Advanced Research Projects Agency*. (Agencia de proyectos de investigación avanzada de defensa).

datagrama IP Un paquete de la capa de red IP.

demonio Un servidor de servicios de sistema básicos que se ejecuta en segundo plano.

DENY Una decisión de regla de filtrado de un firewall para eliminar simplemente un paquete sin devolver una notificación al emisor.

DHCP *Dynamic Host Configuration Protocol*, Protocolo de configuración de host dinámico, se usa para asignar direcciones IP de forma dinámica y proporcionar información de servidor y de enrutador a clientes sin direcciones IP registradas. DHCP se desarrollo para sustituir a BOOTP.

dhcpcd Uno de los programas cliente DHCP que localiza un servidor DHCP, solicita una dirección IP y renueva su concesión en la dirección que se ha asignado.

dhcpd El programa servidor de DHCP, que atiende las solicitudes de los clientes sobre un servidor disponible y solicita una asignación de dirección IP, además de renovar periódicamente la concesión de las direcciones del cliente.

difusión múltiple Un paquete IP direccionado especialmente para una dirección IP multidifusión de clase D. Los clientes multidifusión se registran con los enrutadores intermedios para recibir paquetes direccionados a una dirección multidifusión particular.

difusión Un paquete IP que se dirige y envía a todas las interfaces conectadas a la misma red.

dirección asignada de forma dinámica Direcciones IP asignadas temporalmente a una interfaz de red de cliente por un servidor central, como un servidor DHCP. Las direcciones asignadas dinámicamente se suelen asignar a máquinas de clientes o empleados de un grupo de direcciones IP registradas, propiedad de un ISP o de un negocio.

dirección asignada de forma estática Direcciones IP impresas, asignadas de forma permanente, tanto si son direcciones registradas públicamente como direcciones privadas.

dirección IP Un identificador numérico único que se asigna a una red específica o a una interfaz de red de un dispositivo específico en una red. Es una dirección software que se puede traducir directamente a un host o nombre de red comprensible por el usuario. Las direcciones IP de interfaz de red de host también se asocian con una o más direcciones de interfaz de red de hardware.

directiva aceptar todo de forma predeterminada Una directiva que acepta todos los paquetes que no coinciden con una regla de firewall de la cadena. Por tanto, casi todas las reglas de firewall son reglas DENY que definen las excepciones a la directiva aceptar predeterminada.

Directiva denegar todo de forma predeterminada Una directiva que elimina todos los paquetes que no cumplen una regla del firewall en la cadena. Por tanto, la mayoría de reglas de firewall son reglas ACCEPT que definen las excepciones a la directiva denegar predeterminada.

directiva predeterminada Una directiva para un conjunto de reglas, ya sea para una cadena input, o para una cadena forward, que define una disposición del paquete cuando éste no coincide con ninguna regla del conjunto. Véase también la directiva aceptar todo de forma predeterminada.

DMZ La zona desmilitarizada, un perímetro de red que contiene maquinas que atienden servicios públicos, separadas de la red privada local. Los servidores públicos menos seguros están aislados de la LAN privada.

DNS *Domain Name System*, (Sistema de nombres de dominio); un servicio global de base de datos de Internet que permite inicialmente a los clientes la búsqueda de direcciones IP de *host*, dado el *host* completamente cualificado y el nombre de dominio, así como para buscar nombres de equipo completamente cualificados, dadas sus direcciones IP.

doble tarjeta Un equipo con dos o más interfaces de red. Véase también multitarjeta.

enmascaramiento Proceso de sustituir una dirección origen local de un paquete saliente con la del firewall o máquina que hace de pasarela, de forma que permanezcan ocultas las direcciones IP de la LAN. El paquete parece proceder de la máquina de pasarela en lugar de una máquina interna de la LAN. El proceso se invierte para paquetes de respuesta entrantes desde servidores remotos. La dirección de destino del paquete, la dirección IP de la máquina firewall, se sustituye con la dirección de la máquina cliente dentro de la LAN interna. El enmascaramiento IP se suele llamar traducción de direcciones red (NAT, network address translation).

escribibles por el mundo Objetos del sistema de archivos, archivos, directorios o sistemas de archivos completos, que los puede escribir cualquier cuenta o programa del sistema.

exploración de puerto Una exploración de todos o de parte de los puertos de servicio de un host, normalmente los puertos de servicio que se suelen asociar con debilidades de seguridad.

filtrado de paquetes Véase firewall, filtrado de paquetes.

Filtro, firewall Una regla de filtrado de paquetes de un firewall, o paquete de exploración, que define las características del paquete, que si coinciden, determina si se permite que el paquete pase a través de la interfaz de red o se elimina. Los filtros se definen en términos de un origen de paquetes y direcciones destino,

puertos destino y origen, tipo de protocolo, estado de la conexión TCP y tipo de mensaje ICMP.

finger Un programa de búsqueda de información de usuario.

firewall Un equipo, enrutador o aplicación, que impone un conjunto de directivas de seguridad que restringen de forma severa el acceso al sistema y a los recursos de red.

firewall, bastión Un firewall que tiene dos o más interfaces de red y es la pasarela o punto de conexión entre dos redes, la mayoría entre una LAN e Internet. Como un firewall bastión es un único punto de conexión entre redes, el bastión se asegura con todas las medidas posibles.

firewall, contención Un firewall de LAN que tiene dos o más interfaces de red y que es la pasarela o punto de conexión entre esas redes. Una parte conecta con una DMZ, la red de perímetro entre el firewall de contención y un firewall bastión. Las otras interfaces de red conectan con las LAN internas y privadas.

firewall, filtrado de paquetes Un firewall que se implementa en la red y en las capas de transporte que filtran el tráfico de red en función de los paquetes, tomando decisiones de enrutamiento basándose en la información del encabezado del paquete IP.

firewall, host explorado Un firewall de host único que requiere que los usuarios locales conecten específicamente con la máquina firewall para tener acceso a Internet desde la máquina firewall. Un firewall de exploración de host no enruta el tráfico entre redes. Además de filtrar paquetes, la exploración se realiza mediante NAT o los servidores proxy de aplicación. Como se muestra en la Figura C.1, en un sistema firewall de host único, la funcionalidad de filtrado de paquetes de un firewall bastión se combina con las funciones de servidor público asociadas con máquinas de servidor público en una red DMZ.

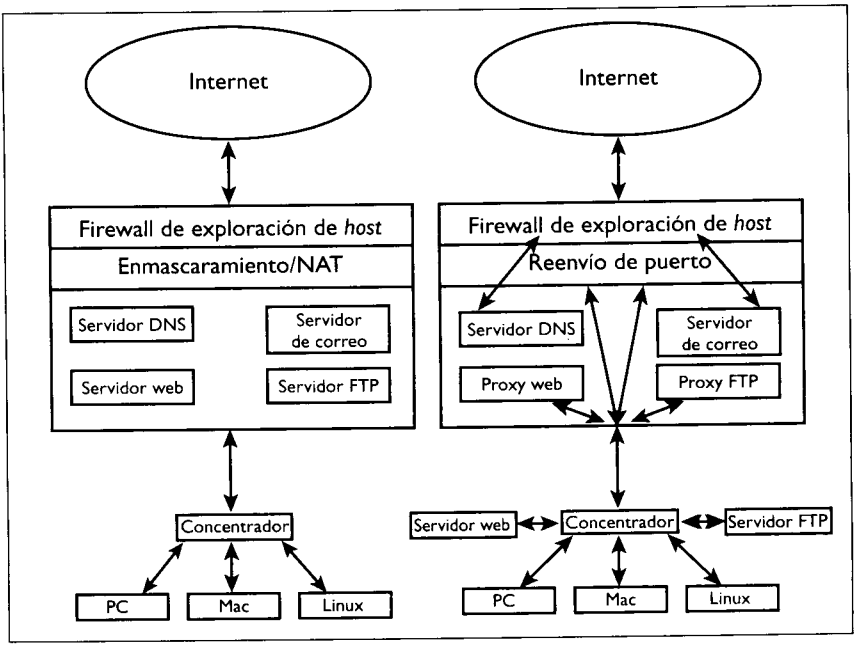

Figura C.1. Dos arquitecturas de firewall de exploración de *host*.

firewall, subred explorada Un sistema firewall que incorpora un firewall bastión y una red DMZ para explorar una LAN del acceso directo a Internet. La red DMZ, que contiene servidores públicos, es una red independiente o subred de una LAN privada. La Figura C.2. muestra las dos configuraciones de firewall básicas de subred explorada.

fragmento Un paquete IP que contiene una parte de un segmento TCP.

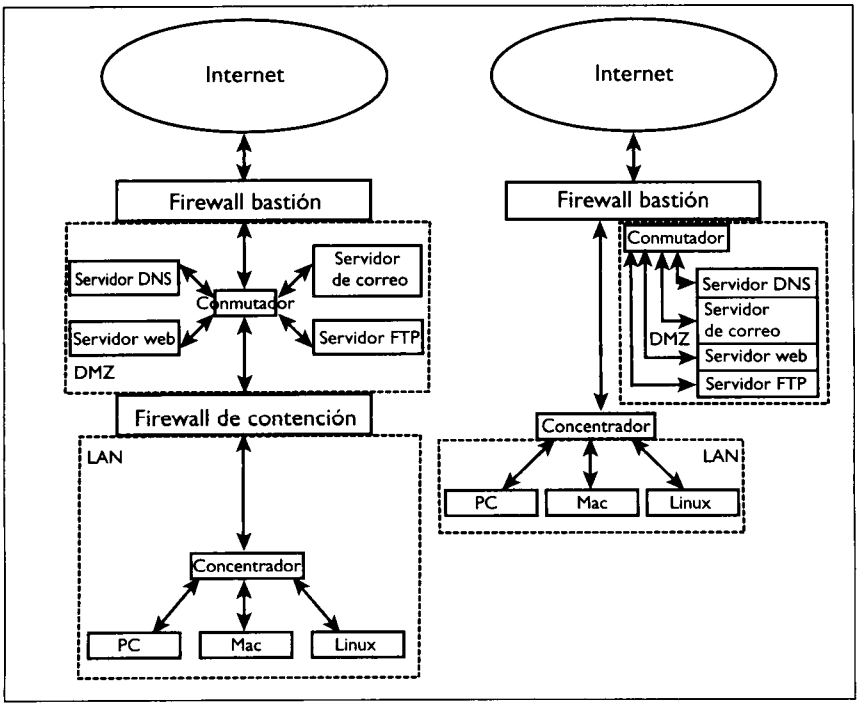

Figura C.2. Dos arquitecturas de firewall de exploración de subred.

FTP *File Transfer Protocol* (Protocolo de transferencia de archivos), el protocolo y los programas que se utilizan para copiar archivos entre equipos de una red.

FTP, anónimo Servicio FTP accesible por cualquier cliente que solicita el servicio.

FTP, autenticado Servicio FTP accesible a cuentas previamente definidas y que deben autenticarse antes de utilizar el servicio.

host explorado Véase firewall, *host* explorado.

hosts.allow, hosts.deny Los archivos de configuración de los empaquetadores de TCP son /etc/hosts.allow y /etc/hosts.deny.

HOWTO Además de las páginas man estándar, Linux incluye documentación interactiva sobre muchos temas, en muchos idiomas y en varios formatos. Los documentos HOWTO se coordinan y mantienen por el Proyecto de documentación de Linux.

HTTP *Hypertext Transfer Protocol* (Protocolo de transferencia de hipertexto); se usa en servidores y navegadores web.

IANA *Internet Assigned Numbers Authority* (Autoridad de números asignados de Internet).

ICMP *Internet Control Message Protocol*, (Protocolo de control de mensajes internet); un estado IP de la capa de red y un mensaje de control.

identd El servidor de autenticación (AUTH) de usuarios.

ifstatus Un programa del sistema que comprueba la configuración de la interfaz de red del sistema y que informa de cualquier interfaz en modo de depuración o promiscuo.

IMAP *Internet Message Access Protocol* (Protocolo de Acceso de Mensajes Internet); se usa para recuperar correo de servidores de correo que ejecutan un servidor IMAP.

inetd Un superservidor de red que escucha conexiones entrantes a puertos de servicio que utilizan servidores administrados por él. Cuando llega una solicitud, inetd inicia una copia del servidor solicitado para controlar la conexión.

inetd.conf Archivo de configuración de inetd.

innd El servidor de noticias de red Usenet de UNIX.

interfaz de bucle invertido Una interfaz de red software especial que utiliza el sistema para entregar mensajes de red locales destinados a la máquina local, pasando por alto la interfaz de red hardware y el controlador de red asociado.

inundación, paquete Un ataque por denegación de servicio en el que el host víctima, o la red, recibe más paquetes de un cierto tipo de los que puede aceptar la víctima.

IP *Internet Protocol* (Protocolo de Internet).

ipchains Con la presentación de la nueva implementación en Linux del mecanismo de firewall IPFW, es el nuevo programa de administración de firewall.

IPFW Mecanismo de firewall IP.

ipfwadm Antes de la presentación de ipchains, era el programa de administración de firewall IPFW de Linux.

IRC *Internet Relay Chat*, se utiliza para comunicación de escritura electrónica entre usuarios y grupos en red.

ISP *Internet service provider* (Proveedor de servicios de Internet).

klogd El demonio de registro del núcleo que recopila errores del sistema operativo y mensajes de estado desde los búfers de mensaje del núcleo y, junto con syslogd, escribe los mensajes en el archivo de registro del sistema.

LAN *Local Area Network* (Red de área local).

legibles por el mundo Objetos del sistema de archivos, archivos, directorios o sistemas de archivos completos, que son legibles por cualquier cuenta o programa del sistema.

localhost Nombre simbólico que se suele utilizar para la interfaz de bucle invertido de una máquina en /etc/hosts.

MD5 Un algoritmo de suma de comprobación cifrado que se usa para asegurar la integridad de los datos creando firmas digitales de objetos, llamadas síntesis del mensaje.

modelo cliente/servidor Modelo para los servicios de red distribuidos, en el que un programa centralizado, un servidor, proporciona un servicio a programas cliente remotos que solicitan ese servicio, tanto si el servicio escribe una copia de una página web, descarga un archivo desde un almacén central, realiza una búsqueda en una base de datos, envía o recibe correo electrónico, realiza algún tipo de cálculo sobre datos proporcionados por un cliente o establece conexiones de comunicación entre dos o más personas.

modelo de referencia OSI (*Open Systems Interconnection*, Interconexión de sistemas abiertos) Un modelo de siete capas de la Organización internacional para la estandarización (OSI, *International Organization for Standardization*) que marca una línea de trabajo o directrices para los estándares de interconexión de redes.

modelo de referencia TCP/IP Un modelo de comunicación de red informal desarrollado cuando TCP/IP se convirtió en el estándar de facto para las comunicaciones de Internet entre máquinas UNIX a finales de los años 70 y comienzos de los años 80. En lugar de ser un ideal académico y formal, el modelo de referencia TCP/IP se basa en lo que los fabricantes y programadores se pusieron de acuerdo que debería ser la comunicación en Internet.

MTU *Maximum Transmission Unit* (Unidad de transmisión máxima): el máximo tamaño de paquete en función de la red subyacente.

multitarjeta Un equipo que tiene dos o más interfaces de red. Véase también de doble tarjeta.

named El servidor de nombres DNS.

NAT *Network address translation* (Traducción de direcciones de red o enmascaramiento IP); el proceso de sustituir una dirección de origen local de un paquete con la de un firewall o máquina de pasarela, de forma que permanezcan ocultas las direcciones IP de la LAN. El paquete parece proceder de una máquina de pasarela en lugar de una máquina interna de la LAN. El proceso se invierte para paquetes de respuesta entrantes desde servidores remotos. La dirección destino del paquete, la dirección IP de la máquina firewall, se sustituye con la dirección de la máquina cliente en la LAN interna.

netstat Un programa que registra distintos tipos de estados de red en función de varias tablas de núcleo relacionadas con la red.

NFS *Network File System* (Sistema de archivos de red), se usa para compartir sistemas de archivos entre equipos que funcionan en red.

NIS *Network Information Service* (Servicio de información en red), se usa para administrar de forma centralizada y proporcionar cuentas de seguridad e información de host.

nivel de ejecución Un concepto de inicio y estado del sistema original de System V UNIX. Un sistema funciona normalmente en uno de los niveles de ejecución 2, 3 ó 5. El nivel de ejecución 3 es el estado de sistema multiusuario normal y predeterminado. El nivel de ejecución 2 es el mismo que el 3 sin la ejecución de servicios NFS. El nivel de ejecución 5 es el mismo que el 3, con el añadido del ad-

ministrador de pantalla de X Windows, que representa un inicio de sesión de usuario basado en X y una pantalla de selección de host.

nmap Una nueva herramienta de auditoría de seguridad en red (por ejemplo, exploración de puertos) que incluye muchas de las nuevas técnicas de exploración que se utilizan actualmente.

NNTP *Network News Transfer Protocol*; el protocolo de transferencia de noticias de red que utiliza Usenet.

NTP *Network Time Protocol*, el protocolo de hora de red que utilizan xntpd y ntpdate.

ntpdate Un programa cliente que se pone en contacto con uno o más servidores de tiempo NTP para solicitar la hora actual.

OSPF El protocolo de enrutamiento Abrir primero la ruta de acceso más corta (*Open Shortest Path First*) para TCP/IP, que es actualmente el protocolo de enrutamiento que más se utiliza. El demonio de enrutamiento gated utiliza OSPF.

página man El formato estándar de documentación interactiva de UNIX. Existen páginas del manual para la mayoría de programas de usuario y de administración del sistema, así como llamadas del sistema, llamadas de biblioteca, tipos de dispositivos y formatos de archivos del sistema.

panel de control Un conjunto de herramientas básicas de administración del sistema presentadas en una interfaz GUI. linuxconf es un programa nuevo de pleno derecho.

paquete Un datagrama de red IP.

pasarela de aplicación Véase proxy, pasarela de aplicación.

pasarela Un equipo o programa que sirve como conducto, o transmisión, entre dos redes.

PATH La variable de entorno del shell que define los directorios que debería examinar el shell en busca de comandos ejecutables no cualificados y el orden en que el shell debería buscar en esos directorios.

PID Process ID (Id. de proceso) es un identificador numérico único para el proceso del sistema, que suele estar asociado con la ranura del proceso en la tabla de procesos del sistema.

ping Una herramienta sencilla de análisis de red que se usa para determinar si es posible contactar con un *host* remoto y está preparado para responder. ping envía un mensaje de solicitud de eco ICMP. El host receptor devuelve un mensaje de respuesta de eco ICMP.

POP *Post Office Protocol* (Protocolo de oficina de correos); se usa para recuperar el correo de los *host* que ejecutan un servidor POP.

portmap Un demonio que administra RPC y se usa para asignar entre un número de servicio RPC particular al que un cliente solicita acceso y el puerto de servicio al que está enlazado el servidor asociado.

proxy Un programa que crea y mantiene una conexión de red en beneficio de otro programa, proporcionando un conducto en el nivel aplicación entre un cliente y un servidor. El cliente y el servidor, en realidad, no tienen comunicación directa. El proxy aparenta ser el servidor para el programa cliente y parece ser el cliente para el programa servidor. Como se muestra en la Figura C.3, los servidores proxy

de aplicación se clasifican generalmente en pasarelas de aplicación y pasarelas de circuito.

proxy, pasarela de aplicación Parecida a un firewall de exploración de host que se implementa en la configuración del sistema y en los niveles de aplicación. Sólo el equipo que ejecuta la pasarela de aplicación tiene acceso directo a Internet. El tráfico de red nunca se envía automáticamente a través de la pasarela de aplicación. Todo el acceso a Internet se realiza a través del programa de pasarela. Sólo se permite el acceso externo hacia la máquina de pasarela. El acceso interno sólo se permite hacia la máquina de pasarela. Los usuarios locales deben iniciar una sesión de usuario en la máquina de pasarela y tener acceso a Internet desde ahí, o conectar con la pasarela de aplicación y autenticarse primero a sí mismos. Un servidor proxy se suele implementar como una aplicación independiente para cada servicio que hace uso del proxy. El servidor proxy del nivel aplicación entiende el protocolo de comunicación específico de la aplicación. Cada aplicación proxy aparenta ser el servidor para el programa cliente y aparenta ser el cliente para el servidor real. Los programas cliente especiales, o los programas cliente configurados especialmente, se conectan a un servidor proxy en lugar de a un servidor remoto. El proxy establece la conexión con el servidor remoto en beneficio de la aplicación cliente, después de sustituir la dirección de origen del cliente con la suya. Se incluyen algunos ejemplos de pasarelas en el nivel aplicación como el servidor proxy web de Apache y servidores proxy específicos de aplicación en el TIS Firewall Toolkit.

proxy, pasarela de circuito Un servidor proxy que se puede implementar como aplicaciones separadas para que cada servicio pueda hacer uso del proxy, como una transmisión generalizada de la conexión que no tiene un conocimiento específico sobre protocolos de aplicación. La transmisión a nivel de circuito crea un circuito de conexiones virtuales administradas por software entre un cliente y un programa servidor. Al contrario que los servidores proxy del nivel aplicación, los pasos intermedios de la conexión se realizan de forma transparente hacia el usuario. SOCKS es un sistema proxy de nivel de circuito.

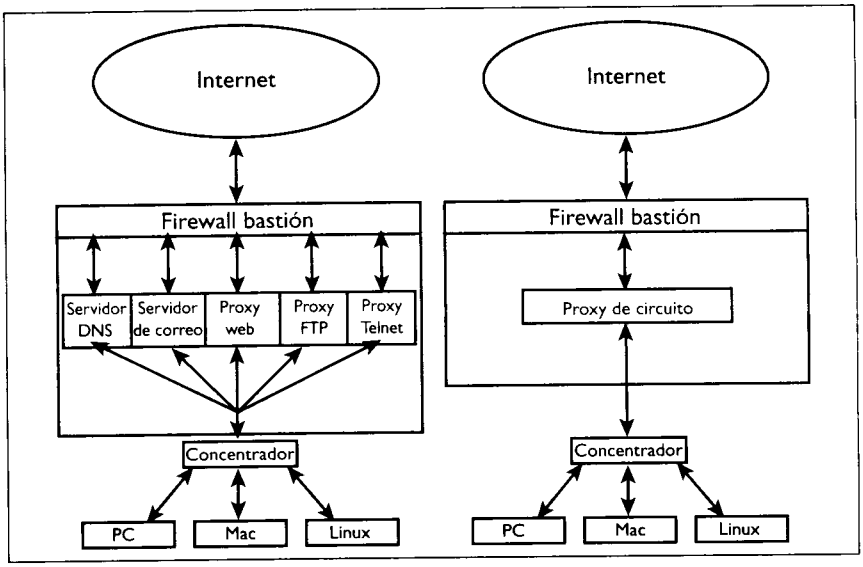

Figura C.3. Pasarelas proxy del nivel aplicación y del nivel de circuito.

puerto En TCP o UDP, el designador numérico de un canal particular de comunicación de red. Las asignaciones de puerto las administra el IANA. Algunos puertos se asignan a protocolos de comunicación de una aplicación particular como parte del estándar del protocolo. Algunos puertos se registran como asociados con un servicio particular por convención. Algunos puertos tienen libertad de asignación para que los usen los clientes y los programas de usuario:

- **privilegiado** Un puerto del intervalo que va desde 0 hasta 1023. Muchos de estos puertos los asigna el estándar internacional a protocolos de aplicación. El acceso a puertos privilegiados requiere privilegios del nivel de sistema.
- **no privilegiado** Un puerto del intervalo que va desde 1024 hasta 65535. Algunos de estos puertos se registran y se usan según una convención para ciertos programas. Cualquier puerto que pertenezca a este intervalo lo puede utilizar un programa cliente para establecer una conexión con un servidor de la red.

punto a punto Un modo de comunicación que se usa para la comunicación entre dos programas servidor. Un protocolo de comunicación de igual a igual suele ser, pero no siempre, distinto de los protocolos que se utilizan para la comunicación entre el servidor y el cliente.

QoS Calidad de servicio.

RARP *Reverse Address Resolution Protocol* (Protocolo de resolución de direcciones inversas); desarrollado para permitir a las máquinas sin disco solicitar su dirección IP basándose en la dirección hardware MAC.

reenviar Para enrutar paquetes desde una red a otra en el proceso de envío de un paquete desde un equipo a otro.

regla REJECT Una decisión de regla de filtrado firewall para eliminar un paquete y devolver al emisor un mensaje de error ICMP.

regla Véase firewall y filtro.

RFC *Request for Comment* (Peticiones de comentarios); una nota o recordatorio publicada a través de la Internet Society (Sociedad de Internet) o del Internet Engineering Task Force (Grupo de trabajo de ingeniería de Internet). Algunos RFC se convierten en estándares. Los RFC suelen estar relacionados con temas de Internet o el conjunto de protocolos TCP/IP.

RIP *Routing Information Protocol* (Protocolo de información de enrutamiento); un antiguo protocolo de enrutamiento que todavía se usa actualmente, especialmente dentro de una LAN grande. El demonio routed utiliza RIP.

RPC *Remote-procedure call* (Llamada a procedimiento remoto).

saludo de tres vías El protocolo de establecimiento de la conexión TCP. Cuando un programa cliente envía el primer mensaje al servidor, el mensaje de solicitud de conexión, se activa el indicador SYN y se acompaña de un número de secuencia de sincronización que usará el cliente como punto de partida para numerar los demás mensajes que envíe el cliente. El servidor responde con un reconocimiento (ACK) al mensaje SYN, junto con su propia solicitud de sincronización (SYN). El servidor incluye el número de secuencia del cliente incrementado en uno. El propósito es reconocer el mensaje al que hace referencia el cliente por su número de secuencia. Al igual que con el primer mensaje del cliente, el indicador SYN se acompaña de un número de secuencia de sincronización. El servidor sigue pasando su propio número de secuencia inicial hasta la mitad de la conexión. El cliente responde con un ACK del SYN-ACK del servidor,

incrementando el número de secuencia del servidor en uno para indicar la recepción del mensaje. La conexión se ha establecido.

SATAN *Security Administrator Tool for Analyzing Networks* (Herramienta del administrador de seguridad para análisis de redes); una herramienta que ayuda a identificar posibles debilidades de seguridad en configuraciones de servicios de red.

secuencia de comandos Una secuencia de comandos ejecutable ASCII del shell, como un archivo que contiene comandos sh, csh, bash, ksh o perl.

segmento, TCP Un mensaje TCP.

servidor de nombres, maestro Un servidor con autoridad para un dominio o para una zona del espacio del dominio. El servidor mantiene una base de datos completa de nombres de equipos y direcciones IP para esta zona.

Servidor de nombres, secundario Una copia de seguridad o principal para un servidor de nombres maestro.

setgid Un programa que, cuando se ejecuta, asume el Id. del grupo del propietario del programa, en lugar de la Id. del grupo del proceso que ejecuta el programa.

setuid Un programa que cuando se ejecuta asume la Id. del propietario del programa, en lugar de la Id. del usuario del proceso que ejecuta el programa.

shell Un interprete de comandos UNIX, como sh, ksh, bash y csh.

smrsh *Sendmail Restricted Shell* (Shell de envío de correo restringida).

SMTP *Simple Mail Transport Protocol* (Protocolo simple de transferencia de correo); se usa para intercambiar correo entre servidores de correo y entre programas de correo y servidores de correo.

SNMP *Simple Network Management Protocol* (Protocolo simple de administración de redes); se usa para administrar la configuración de dispositivos de red desde una estación de trabajo.

socket El único punto de conexión de red definido por el par de una dirección IP con un puerto de servicio UDP o TCP particular.

SOCKS Un conocido paquete proxy de pasarela de circuito, que puede conseguirse desde NEC.

solucionador Programas de cliente DNS. Los solucionadores se implementa como un código de biblioteca que se vincula a programas que necesitan tener acceso a la red. El archivo de configuración del cliente DNS es /etc/resolv.conf.

sondeo Para enviar algún tipo de paquete al puerto de servicio del host de alguien. El propósito de un sondeo es determinar si una respuesta la ha generado el host destino.

SSH *Protocolo Secure shell* (Shell seguro); se usa para conexiones de red cifradas y autenticadas de forma más segura.

SSL *Protocolo Secure Socket Layer* (Nivel de socket seguro); se usa para comunicación cifrada. SSL se suele usar habitualmente en servidores y navegadores web para intercambiar información personal de comercio electrónico.

strobe Una exploración básica del puerto TCP.

subred explorada Véase firewall, subred explorada.

subred Un esquema de direccionamiento de subred, que es una forma de usar las máscaras de dirección IP para dividir internamente un único espacio de direcciones de red de Internet en varios espacios de direcciones de red internos. El espacio de direcciones aparece como una única red para Internet, pero de forma interna aparece como varias redes o LAN. Esto se hace usando parte de los bits de la Id. del host de la dirección IP como si fueran una extensión de la sección Id. de red de la dirección IP. Por ejemplo, el espacio de direcciones de red 192.168.30.0, enmascarado como 255.255.255.0/24, se podría dividir en dos subredes internas usando el bit de orden superior de la cuarta tupla, enmascarado como 255.255.255.128/25. El resultado son dos espacios de direcciones de red internas, cada una con 126 direcciones disponibles.

suma de comprobación Un número que resulta de ciertos cálculos aritméticos sobre el valor numérico de cada byte de un archivo o paquete. Si se cambia el archivo, o se corrompe el paquete, una segunda suma de comprobación para el mismo objeto no coincidirá con la suma de comprobación original.

SUNRPC Puerto de servicio 111, que lo usa el demonio portmap para asignar solicitudes entrantes a servicios RPC al puerto de servicio al que está vinculado el servidor asociado.

SYN El indicador de solicitud de sincronización TCP. Un mensaje SYN es el primer mensaje que envía un programa para abrir una conexión con otros programas presentes en la red.

syslog.conf El archivo de configuración del demonio de inicio de sesión.

syslogd El demonio de inicio de sesión del sistema, que recopila mensajes de estado y error generados por los programas que envían mensajes mediante la llamada al sistema syslog().

TCP *Transmission Control Protocol* (Protocolo de control de la transmisión); se usa para conexiones de red activas y fiables entre dos programas.

tcp_wrapper Un esquema de autorización que se usa para controlar los servicios locales que están disponibles para los equipos remotos de la red.

tcpd Servicio de empaquetamiento TCP, que se proporciona para servicios administrados por inetd o por el programa tcpd.

TFTP *Trivial File Transfer Protocol* (Protocolo de transferencia de archivos trivial); es el protocolo que se usa para descargar una imagen de inicio a un enrutador o a una estación de trabajo sin disco. El protocolo es una versión simplificada de FTP basada en UDP.

tiger Una colección de secuencias de comandos y programas C diseñados para comprobar la vulnerabilidad de seguridad que podría permitir que alguien accediera al equipo de forma no autorizada con permisos de root.

TOS Tipo de servicio, campo de la cabecera del paquete IP que se ha intentado que proporcione una sugerencia como directiva de enrutamiento preferida o enrutamiento de paquetes preferida.

traceroute Una herramienta de análisis de red que se usa para determinar la ruta de un equipo a otro a través de la red.

Trama Ethernet En una red Ethernet, los datagramas IP se encapsulan en tramas Ethernet.

transferencia de zona El proceso de copiar la base de datos de nombres de equipo del servidor de nombres autorizado DNS y las direcciones IP que pertenecen a una sección contigua de un dominio a un servidor de nombres secundario.

tripwire Un programa que genera y mantiene una base de datos de firmas digitales MD5 para todo un conjunto de archivos o directorios del sistema. Su propósito es detectar agregaciones no autorizadas o cambios en los archivos.

TTL Time-to-live (Tiempo de vida); campo de la cabecera de un paquete IP que es el valor máximo del número de enrutadores a través de los que puede pasar el enrutador antes de alcanzar su destino.

UDP *User Datagram Protocol* (Protocolo de datagrama de usuario); se usa para enviar mensajes de red individuales entre programas, sin ninguna garantía de envío u orden de envío.

unidifusión Un paquete IP enviado punto a punto, de la interfaz de red de un equipo a la de otro.

usurpamiento, dirección origen Falsificación de la dirección origen en un encabezado de paquete IP para ser el de otras direcciones.

UUCP Protocolo UNIX-to-UNIX Copy (Copia UNIX a UNIX).

WAIS *Wide Area Information Service* (Servicio de información de área extensa); conocido ahora como un motor de búsqueda de Internet.

WAREZ Un almacén de software pirata.

WWW World Wide Web.

X Window El sistema que muestra la ventana de la interfaz gráfica de usuario de UNIX.

xntpd El servidor de hora de red NTP, que proporciona a los clientes de red tanto la fecha actual como información de intercambio de hora con otros servicios.