Red Hat Enterprise Linux 6

Guía de seguridad

Guía para proteger a Red Hat Enterprise Linux

Red Hat Enterprise Linux 6 Guía de seguridad
Guía para proteger a Red Hat Enterprise Linux
Edición 1.5

Autor

1801 Varsity Drive
Raleigh, NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701

Este libro ayuda a los usuarios y administradores en el aprendizaje de los procesos y prácticas en la protección de estaciones de trabajo contra intrusos remotos o locales, explotación o actividad maliciosa.

Enfocada en Red Hat Enterprise Linux pero describiendo conceptos y técnicas válidas para todos los sistemas de Linux, esta guía describe la planeación y herramientas que tienen que ver en la creación de un entorno informático seguro para el centro de datos, el sitio de trabajo y el hogar.

Con el conocimiento administrativo adecuado, vigilancia y herramientas, los sistemas que ejecutan Linux pueden ser completamente funcionales y estar protegidos de los métodos más comunes de intrusión y de explotación.

Prefacio

1. Convenciones del Documento

Este manual utiliza varias convenciones para resaltar algunas palabras y frases y llamar la atención sobre ciertas partes específicas de información.

En ediciones PDF y de papel, este manual utiliza tipos de letra procedentes de *Liberation Fonts*[1]. Liberation Fonts también se utilizan en ediciones de HTML si están instalados en su sistema. Si no, se muestran tipografías alternativas pero equivalentes. Nota: Red Hat Enterprise Linux 5 y siguientes incluyen Liberation Fonts predeterminadas.

1.1. Convenciones Tipográficas

Se utilizan cuatro convenciones tipográficas para llamar la atención sobre palabras o frases específicas. Dichas convenciones y las circunstancias en que se aplican son las siguientes:

`Negrita monoespaciado`

Utilizada para resaltar la entrada del sistema, incluyendo comandos de shell, nombres de archivo y rutas. También se utiliza para resaltar teclas claves y combinaciones de teclas. Por ejemplo:

> Para ver el contenido del archivo **`my_next_bestselling_novel`** en su directorio actual de trabajo, escriba el comando **`cat my_next_bestselling_novel`** en el intérprete de comandos de shell y pulse **`Enter`** para ejecutar el comando.

El ejemplo anterior incluye un nombre de archivo, un comando de shell y una tecla clave. Todo se presenta en negrita-monoespaciado y distinguible gracias al contexto.

Las combinaciones de teclas se pueden distinguir de las teclas claves mediante el guión que conecta cada parte de una combinación de tecla. Por ejemplo:

> Pulse **`Enter`** para ejecutar el comando.
>
> Pulse **`Control`**+**`Alt`**+**`F2`** para cambiar a la primera terminal virtual. Pulse **`Control`**+**`Alt`**+**`F1`** para volver a su sesión de Ventanas-X.

La primera oración resalta la tecla clave determinada que se debe pulsar. La segunda resalta dos conjuntos de tres teclas claves que deben ser presionadas simultáneamente.

Si se discute el código fuente, los nombres de las clase, los métodos, las funciones, los nombres de variables y valores de retorno mencionados dentro de un párrafo serán presentados en **`Negrita-monoespaciado`**. Por ejemplo:

> Las clases de archivo relacionadas incluyen **`filename`** para sistema de archivos, **`file`** para archivos y **`dir`** para directorios. Cada clase tiene su propio conjunto asociado de permisos.

Negrita proporcional

Esta denota palabras o frases encontradas en un sistema, incluyendo nombres de aplicación, texto de cuadro de diálogo, botones etiquetados, etiquetas de cajilla de verificación y botón de radio; títulos de menú y títulos del sub-menú. Por ejemplo:

[1] https://fedorahosted.org/liberation-fonts/

> Seleccionar **Sistema** → **Preferencias** → **Ratón** desde la barra del menú principal para lanzar **Preferencias de Ratón**. En la pestaña de **Botones**, haga clic en la cajilla **ratón de mano izquierda** y luego haga clic en **Cerrar** para cambiar el botón principal del ratón de la izquierda a la derecha (adecuando el ratón para la mano izquierda).
>
> Para insertar un caracter especial en un archivo de **gedit**, seleccione desde la barra del menú principal **Aplicaciones** → **Accessories** → **Mapa de caracteres**. Luego, desde la barra de menúes de **mapa de caracteres** elija **Búsqueda** → **Hallar…**, teclee el nombre del caracter en el campo **Búsqueda** y haga clic en **Siguiente**. El caracter buscado se resaltará en la **Tabla de caracteres**. Haga doble clic en este caracter resaltado para colocarlo en el campo de **Texto para copiar** y luego haga clic en el botón de **Copiar**. Ahora regrese a su documento y elija **Editar** → **Pegar** desde la barra de menú de **gedit**.

El texto anterior incluye nombres de aplicación; nombres y elementos del menú de todo el sistema; nombres de menú de aplicaciones específicas y botones y texto hallados dentro de una interfaz gráfica de usuario, todos presentados en negrita proporcional y distinguibles por contexto.

Itálicas-negrita monoespaciado o ***Itálicas-negrita proporcional***

Ya sea negrita monoespaciado o negrita proporcional, la adición de itálicas indica texto reemplazable o variable. Las itálicas denotan texto que usted no escribe literalmente o texto mostrado que cambia dependiendo de la circunstancia. Por ejemplo:

> Para conectar a una máquina remota utilizando ssh, teclee **ssh *nombredeusuario@dominio.nombre*** en un intérprete de comandos de shell. Si la máquina remota es **example.com** y su nombre de usuario en esa máquina es john, teclee **ssh john@example.com**.
>
> El comando **mount -o remount *file-system*** remonta el sistema de archivo llamado. Por ejemplo, para volver a montar el sistema de archivo **/home**, el comando es **mount -o remount /home**.
>
> Para ver la versión de un paquete actualmente instalado, utilice el comando **rpm -q *paquete***. Éste entregará el resultado siguiente: ***paquete-versión-lanzamiento***.

Observe las palabras en itálicas y negrita sobre — nombre de usuario, domain.name, sistema de archivo, paquete, versión y lanzamiento. Cada palabra es un marcador de posición, tanto para el texto que usted escriba al ejecutar un comando como para el texto mostrado por el sistema.

Aparte del uso estándar para presentar el título de un trabajo, las itálicas denotan el primer uso de un término nuevo e importante. Por ejemplo:

> Publican es un sistema de publicación de *DocBook*.

1.2. Convenciones del documento

Los mensajes de salida de la terminal o fragmentos de código fuente se distinguen visualmente del texto circundante.

Los mensajes de salida enviados a una terminal se muestran en **romano monoespaciado** y se presentan así:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Los listados de código fuente también se muestran en **`romano monoespaciado`**, pero se presentan y resaltan de la siguiente manera:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notas y Advertencias

Finalmente, utilizamos tres estilos visuales para llamar la atención sobre la información que de otro modo se podría pasar por alto.

Nota

Una nota es una sugerencia, atajo o enfoque alternativo para una tarea determinada. Ignorar una nota no debería tener consecuencias negativas, pero podría perderse de algunos trucos que pueden facilitarle las cosas.

Importante

Los cuadros con el título de importante dan detalles de cosas que se pueden pasar por alto fácilmente: cambios de configuración únicamente aplicables a la sesión actual, o servicios que necesitan reiniciarse antes de que se aplique una actualización. Ignorar estos cuadros no ocasionará pérdida de datos, pero puede causar enfado y frustración.

Advertencia

Las advertencias no deben ignorarse. Ignorarlas muy probablemente ocasionará pérdida de datos.

2. ¡Necesitamos sus comentarios!

Si encuentra algún error tipográfico en este manual o si ha pensado en alguna forma de mejorarlo, nos gustaría saberlo. Por favor envío un reporte en Bugzilla:*http://bugzilla.redhat.com/* a nombre del producto **Red Hat Enterprise Linux.**

Al enviar un informe de error, asegúrese de mencionar el identificador del manual: doc-*Security_Guide* y el número de la versión: **6**.

Si tiene alguna sugerencia para mejorar la documentación, trate de ser lo más específico posible al describirla. Si ha encontrado algún error, por favor incluya el número de la sección y parte del contexto para que sea más fácil encontrarlo.

Visión general de seguridad

Debido a la creciente confianza en equipos de red poderosos para ayudar a manejar empresas y mantener el seguimiento de la información personal, se han formado industrias enteras en torno a la práctica de la seguridad de redes y equipos. Las empresas solicitan el conocimiento y las habilidades de los expertos para auditar los sistemas y dar soluciones que se ajusten a los requerimientos operativos de la empresa. Puesto que la mayoría de las organizaciones son de por sí cada vez más dinámicas, sus trabajadores tienen acceso local y remoto a los recursos informáticos, por lo tanto, la necesidad de proteger entornos informáticos se ha acentuado más.

Lamentablemente, muchas organizaciones (así como los usuarios individuales) consideran la seguridad más como una ocurrencia tardía, un proceso que se pasa por alto en favor del aumento en la energía, la productividad, la comodidad, la facilidad de uso y los problemas presupuestarios. La aplicación adecuada de la seguridad suele realizarse postmortem — *después* de que ha ocurrido una intrusión. Si se toman las medidas correctas antes de conectar un sitio a una red insegura, como la Internet, es una forma efectiva de frustrar muchos intentos de intrusión.

Nota

Este documento hace varias referencias a los archivos en el directorio **`/lib`**. Cuando se usan sistemas de 64 bits, algunos de los archivos mencionados pueden localizarse en **`/lib64`**.

1.1. Introducción a seguridad

1.1.1. ¿Qué es la seguridad informática?

La seguridad informática es un término genérico que cubre una vasta área de la informática y del procesamiento de información. Las industrias que dependen de sistemas computarizados y redes para realizar a diario transacciones de negocios y acceder a información confidencial, consideran los datos como una parte importante de sus activos totales. Varios términos e indicadores han entrado a su vocabulario diario, tales como el Costo total de propiedad (TCO), Retorno de la inversión (ROI) y Calidad de servicio (QoS). Mediante estos indicadores, las industrias pueden calcular aspectos tales como integridad de datos y Alta disponibilidad (HA) como parte de los costos administrativos de procesos y planeación. En algunas industrias, como por ejemplo, el comercio electrónico, la habilidad y confiabilidad de datos pueden significar la diferencia entre éxito y fracaso.

1.1.1.1. ¿Cómo surgió la seguridad informática?

La seguridad informática ha evolucionado con los años debido a la creciente confianza en las redes públicas de no revelar información personal, financiera u otra información restringida. Hay varias instancias tales como los casos de Mitnick [1]y los casos de Vladimir Levin [2]que solicitaban organizaciones a través de todas las industrias repensar la forma como manejaban la información, incluyendo su transmisión y revelación. La popularidad de la Internet era uno de los desarrollos más importantes que solicitaban un esfuerzo intensificado en seguridad de datos.

[1] http://law.jrank.org/pages/3791/Kevin-Mitnick-Case-1999.html
[2] http://www.livinginternet.com/i/ia_hackers_levin.htm

Un número creciente de personas utiliza los computadores personales para obtener acceso a recursos que ofrece la Internet. Desde la investigación y la recuperación de información hasta correo electrónico y transacciones comerciales, la Internet es considerada como uno de los desarrollos más importantes del siglo veinte.

Sin embargo, la Internet y los primeros protocolos, se desarrollaron como un sistema *basado en la confiabilidad*. Es decir, el protocolo de Internet (IP) no estaba diseñado para protegerse en sí mismo. No hay estándares de seguridad aprobados que estén incorporados en el paquete de comunicaciones TCP/IP, lo cual deja las puertas abiertas a usuarios malintencionados y procesos a través de la red. Desarrollos modernos han hecho la comunicación por Internet más segura, pero aún se presentan incidentes que llaman la atención nacional y nos alertan con el hecho de que nada es completamente seguro.

1.1.1.2. La seguridad actual

En febrero de 2000, se lanzó un ataque de DDos(Denegación de Servicio Distribuido) en varios sitios de alto tráfico en la Internet. El ataque dejó a yahoo.com, cnn.com, amazon.com, fbi.gov, y otros sitios completamente inaccesibles para los usuarios normales, puesto que ataba los enrutadores por varias horas con grandes transferencias de paquetes ICMP, también conocido como un *flujo de pings*. El ataque era enviado por atacantes desconocidos mediante programas especialmente creados, extensamente disponibles que escaneaban servidores de red vulnerables, aplicaciones de clientes instaladas llamadas *Troyanos* en los servidores y programaban un ataque con cada servidor infectado inundando los sitios de las víctimas y dejándolos no disponibles. Muchos acusan el ataque a defectos en la forma como los enrutadores y protocolos utilizados se estructuran para aceptar todos los datos entrantes, sin importar dónde o con qué propósito se envían los paquetes.

En 2007, una violación de datos aprovechándose de las debilidades ampliamente conocidas del Wired Equivalent Privacy (WEP) el protocolo de cifrado inalámbrico, resultó en el robo de una institución financiera global de más de 45 millones de números de tarjetas de crédito.[3]

En un incidente independiente, los registros almacenados de facturación de más de 2,2 millones de pacientes en una cinta de copia de seguridad fueron robados desde el asiento delantero del transportador de mensajería.[4]

Actualmente, se estima que 1.4 mil millones de personas usan o han usado la Internet en el mundo. [5] Al mismo tiempo:

- En un determinado día, hay aproximadamente 225 incidencias de violación de seguridad reportadas al CERT: Coordination Center at Carnegie Mellon University.[6]
- El número de incidencias reportadas al CERT de 52.658 en 2001, 82.094 en 2002 y hasta 137.529 en 2003.[7]
- Según el FBI, los delitos relacionados con la informática costaron a los negocios estadounidenses $67.2 mil millones de dólares en 2006.[8]

Según una encuesta de seguridad global de 2009 y profesionales de la tecnología informática, "¿Por qué es importante la seguridad ahora?"[9], realizada por *CIO Magazine*, algunos resultados importantes son:

[3] http://www.theregister.co.uk/2007/05/04/txj_nonfeasance/
[4] http://www.fudzilla.com/content/view/7847/1/
[5] http://www.internetworldstats.com/stats.htm
[9] http://www.cio.com/article/504837/Why_Security_Matters_Now

- Apenas el 23% de participantes tienen políticas para usar tecnologías de Web 2.0. Dichas tecnologías, tales como Twitter, Facebook y LinkedIn pueden proporcionar a compañías e individuos una forma conveniente para comunicarse y colaborar, sin embargo pueden abrir espacio para nuevas vulnerabilidades, principalmente en filtraje de información confidencial.

- Incluso durante la reciente crisis financiera de 2009, se halló en la encuesta que los presupuestos para seguridad o el aumento eran en su mayoría igual que en los años anteriores (cerca de 2 a 3 participantes esperan que el gasto en seguridad aumente o permanezca igual). Son buenas noticias y reflejan la importancia que las empresas están prestando a la seguridad hoy en día.

Estos resultados imponen la realidad de que la seguridad informática se ha convertido en un gasto cuantificable y justificable para presupuestos de TI. Las organizaciones que requieren la integridad de datos y alta disponibilidad obtienen las habilidades de los administradores de sistemas, desarrolladores e ingenieros para asegurar la confiabilidad 24x7 de sus sistemas, servicios e información. Ser víctimas de usuarios malintencionados, procesos, o ataques coordinados, es una amenaza directa para el éxito de la organización.

Infortunadamente, la seguridad de sistemas y redes pueden dificultar la proposición, que requieren un conocimiento intrincado de cómo una empresa, ve, usa, manipula y transmite la información. Entender la forma como una organización lleva a cabo un negocio es de suma importancia para implementar un propio plan de seguridad.

1.1.1.3. Estándares de seguridad

Las empresas en cada industria confían en lineamientos y reglas establecidas por instituciones que diseñan estándares como la American Medical Association (AMA) o el Institute of Electrical and Electronics Engineers (IEEE). Los mismos ideales se mantienen para la seguridad de la información. Muchos consultores de seguridad y proveedores están de acuerdo con el modelo de seguridad estándar conocido como CIA o *Confidencialidad, Integridad y Disponibilidad*. Este modelo que consta de tres niveles es un componente generalmente aceptado para evaluar los riesgos de la información y establecer políticas de seguridad. A continuación se describe el modelo de CIA en mayor detalle:

- Confidencialidad — La información confidencial debe estar disponible únicamente para un grupo de individuos pre-establecido. La transmisión y uso de información no autorizada debe restringirse. Por ejemplo, la confidencialidad de información garantiza que la información personal o financiera no esté al alcance de individuos no autorizados con propósitos malintencionados tales como robo de identidad o fraude de crédito.

- Integridad — La información no se debe alterar en formas que la reproduzcan incompleta o incorrecta. Se debe restringir a los usuarios no autorizados de la capacidad de modificar o destruir información confidencial.

- Disponibilidad — La información debe estar accesible a usuarios autorizados, en cualquier momento. Es decir, cuando se necesite. La disponibilidad garantiza que la información pueda obtenerse con una frecuencia y puntualidad acordadas. Suele medirse en términos de porcentajes y se acepta de manera formal en los Acuerdos de Nivel de Servicio (SLA) usados por los proveedores de servicios de red y los clientes corporativos.

1.1.2. SELinux

Red Hat Enterprise Linux incluye una mejora al kernel de Linux llamada SELinux, la cual implementa una arquitectura de Control de acceso obligatorio (MAC) que proporciona un nivel de grano fino de control sobre archivos, procesos, usuarios y aplicaciones en el sistema. Una discusión detallada sobre SELinux está fuera del alcance de este documento; sin embargo, para obtener mayor información sobre SELinux y su uso en Red Hat Enterprise Linux, consulte la Guía del usuario de SELinux de Red Hat Enterprise Linux. Para obtener información sobre los servicios de

configuración y ejecución que están protegidos por SELinux, consulte la Guía de administración de servicios confinados de SELinux. Otros recursos disponibles para SELinux se listan en el *Capítulo 8, Referencias*.

1.1.3. Controles de seguridad

La seguridad informática suele dividirse en tres categorías importantes, comúnmente conocidas como *controles*:

- Físicos\n\n\n
- Técnicos
- Administrativos

Estas tres categorías definen los objetivos principales de una implementación correcta de seguridad. Dentro de dichos controles existen las sub-categorías que más adelante describen los controles y su implementación.

1.1.3.1. Controles físicos

Control físico es la implementación de medidas de seguridad en una estructura definida utilizada para impedir el acceso no autorizado a material confidencial. A continuación, ejemplos de controles físicos:

- Vigilancia de cámaras de circuito cerrado
- Sistemas de alarma térmica o de movimiento
- Guardias de seguridad
- ID de retratos
- Puertas de acero cerradas y cerrojos de seguridad con punto muerto
- Biometría (incluye huellas digitales, voz, cara, iris, tipo de letra y otros métodos usados de identificación)

1.1.3.2. Controles técnnicos

Los controles técnicos usan tecnología como base para controlar el acceso y uso de datos confidenciales a través de la estructura física y la red. Los controles técnicos son de gran alcance y abarcan tecnologías tales como:

- Cifrado
- Tarjetas inteligentes
- Autenticidad de redes
- Listas de control de acceso (ACL)
- Software de auditoría de integridad de archivos

1.1.3.3. Controles administrativos

Los controles administrativos definen los factores de seguridad humanos. Dichos controles involucran a todos los niveles de personal de una organización y determinan qué usuarios tienen acceso a los recursos e información por medios tales como:

- Formación y reconocimiento
- Preparación para desastres y planes de recuperación
- Reclutamiento de personal y estrategias de separación
- Registro de personal y contabilidad

1.1.4. Conclusión

Ahora que ha aprendido acerca de los orígenes, motivos y aspectos de la seguridad, le será más fácil determinar el curso de acción apropiado con respecto a Red Hat Enterprise Linux. Es importante saber cuáles son los factores y condiciones que conforman la seguridad con el fin de planificar e implementar una estrategia adecuada. Con esta información en mente, el proceso se puede formalizar y la forma se aclara a medida que se profundiza en los detalles del proceso de seguridad.

1.2. Evaluación de vulnerabilidad

Con tiempo, recursos y motivación, un intruso puede violar casi cualquier sistema. Todos los procedimientos y tecnologías de seguridad disponibles en la actualidad no puede garantizar que los sistemas estén completamente a salvo de intrusos. Los enrutadores ayudan a proteger las puertas de enlace a Internet. Los cortafuegos ayudan a proteger el perímetro de la red. Las redes privadas virtuales pasan los datos en un flujo cifrado. Los sistemas de detección de intrusos advierten sobre actividades maliciosas. Sin embargo, el éxito de cada una de estas tecnologías depende de una serie de variables, entre ellas:\n

- La habilidad de la persona responsable de la configuración, monitorización y mantenimiento de tecnologías.
- La habilidad de corregir y actualizar servicios y kernel en forma rápida y efectiva.
- La habilidad de la persona responsable para mantener constante vigilancia en la red.

Dado el estado dinámico de los sistemas de información y tecnologías, la protección de los recursos corporativos puede ser bastante compleja. Debido a esta complejidad, suele ser difícil encontrar recursos humanos expertos para todos los sistemas. Aunque es posible tener personal con conocimientos en muchas áreas de seguridad informática en un nivel alto, es difícil retener al personal experto en más de una pocas áreas temáticas. Esto se debe principalmente a que cada materia de seguridad informática requiere constante atención y enfoque. La seguridad informática no se detiene.

1.2.1. Pensar como el enemigo

Suponga que usted quiere administrar una red empresarial. Dichas redes comúnmente comprenden sistemas operativos, aplicaciones, servidores, monitores de redes, cortafuegos, sistemas de detección de intrusos y mucho más. Ahora imagine tratar de estar al día con cada uno de ellos. Dada la complejidad del software actual, y de los entornos de redes, las vulnerabilidades y los errores son una certeza. El mantenerse al corriente con parches y actualizaciones para toda una red puede ser una tarea de enormes proporciones en una empresa grande con sistemas heterogéneos.

Combine los requerimientos de experiencia con la tarea de mantenerse al día, es inevitable que incidentes adversos se presenten, tales como, violación de sistemas, corrupción de datos e interrupción del servicio.

Para aumentar las tecnologías de seguridad y ayudar a proteger sistemas, redes y datos, debemos pensar como el cracker y y evaluar la seguridad de los sistemas revisando las debilidades. Las evaluaciones preventivas de vulnerabilidades contra sus propios recursos de sistemas y

redes pueden revelar problemas que no se han abordado antes de que el cracker aproveche la vulnerabilidad.

La evaluación de una vulnerabilidad es una auditoría interna de la red y sistema de seguridad; el resultado del cual indica la confidencialidad, integridad y disponibilidad, de su red (como se explica en la *Sección 1.1.1.3, "Estándares de seguridad"*). Por lo general, la evaluación de la vulnerabilidad empieza con la fase de reconocimiento, durante la cual se reúnen los datos sobre sistemas de destino y recursos. Esta fase conduce a la fase de preparación del sistema, en la cual el destino es esencialmente revisado sobre todas las vulnerabilidades conocidas. La fase de preparación culmina en la fase del informe, en la que los hallazgos se clasifican en las categorías de alto, medio y bajo riesgo y se tratan los métodos para mejorar la seguridad (o mitigar el riesgo de vulnerabilidad).

Si fuera a realizar la evaluación de una vulnerabilidad en su hogar, probablemente revisaría la puerta de su casa para ver si está cerrada y con seguro. También revisaría las ventanas, para asegurarse de que están completamente cerradas. Este mismo concepto se aplica a los sistemas, redes y datos electrónicos. Los usuarios malintencionados son los ladrones y vándalos de sus datos. Céntrese en sus herramientas, la mentalidad y motivaciones y podrá reaccionar rápidamente a sus acciones.

1.2.2. Definición de evaluación y prueba

Las evaluaciones de vulnerabilidad se dividen en dos tipos: *desde afuera mirando hacia adentro* y *desde adentro mirando alrededor*.

Al realizar una evaluación de vulnerabilidades desde afuera mirando hacia adentro, usted está tratando de comprometer sus sistemas desde el exterior. Estando afuera de la compañía puede ver el punto de vista del cracker. Usted ve lo que el cracker ve — las direcciones IP enrutables públicamente, los sistemas en su *DMZ*, las interfaces externas de su cortafuegos, y mucho más. DMZ quiere decir "Zona desmilitarizada" la cual corresponde a un equipo o a una subred pequeña que se establece entre una red interna de confianza, tal como la Internet pública. Por lo general, la DMZ contiene dispositivos accesibles a tráfico de Internet, tal como servidores de red (HTTP), servidores FTP, servidores SMTP (correo-e) y servidores DNS.

Cuando realiza una evaluación de vulnerabilidad desde adentro mirando alrededor, usted tiene la ventaja de que es interno y su estatus se eleva a confiable. Este es el punto de vista que usted y sus cotrabajadores han registrado en sus sistemas. Usted verá servidores de impresión, servidores de archivos y otros recursos.

Hay distinciones impactantes entre los dos tipos de evaluación de vulnerabilidades. Al ser interno su compañía le otorgará más privilegios que a un externo. En la mayoría de las organizaciones, la seguridad es configurada para mantener a los intrusos fuera. Muy poco se hace para asegurar a los internos de la organización (tal como cortafuegos departamentales, controles de acceso de usuario y procedimientos de autenticación para recursos internos). Por lo general, hay muchos más recursos cuando se mira dentro alrededor ya que la mayoría de sistemas son internos para una compañía. Una vez que usted está fuera de la compañía su estatus pasa a no confiable. Los sistemas y recursos disponibles para usted de modo externo suelen ser limitados.

Considere la diferencia entre las evaluaciones de vulnerabilidad y las *pruebas de penetración*. Piense en la evaluación de vulnerabilidad como el primer paso para una prueba de penetración. La información obtenida de la evaluación se utiliza para pruebas. Mientras que la evaluación se lleva a cabo para comprobar si hay agujeros y vulnerabilidades potenciales, las pruebas de penetración en realidad intentan explotar los resultados.

La evaluación de la infraestructura de red es un proceso dinámico. La seguridad, tanto de la información como física, es dinámica. La realización de una evaluación muestra una visión general, la cual puede arrojar falsos positivos y falsos negativos.

Los administradores de seguridad son solamente tan buenos como las herramientas que utilizan y el conocimiento que posean. Elija cualquiera de las herramientas de evaluación disponibles en la actualidad, ejecútelas en el sistema, y es casi una garantía de que hay algunos falsos positivos. Ya sea por error del programa o del usuario, el resultado es el mismo. La herramienta puede encontrar vulnerabilidades que en realidad no existen (falsos positivos), o, peor aún, la herramienta no puede encontrar vulnerabilidades que en realidad sí existen (falsos negativos).

Ahora que la diferencia entre una evaluación de vulnerabilidad y una prueba de penetración está definida, tome los resultados de la evaluación y revíselos cuidadosamente antes de conducir una prueba de penetración como parte del nuevo enfoque de mejores prácticas.

Advertencia

El intento por explorar vulnerabilidades en recursos de producción puede tener efectos adversos de productividad y eficiencia de sus sistemas y redes.

La lista a continuación examina algunos de los beneficios para realizar evaluaciones de vulnerabilidad.

- Crea un enfoque proactivo en la seguridad de la información.
- Busca vulnerabilidades potenciales antes de que los agresores las encuentren
- Resulta en sistemas que se mantienen actualizados y corregidos.
- Promueve crecimiento y ayuda en el desarrollo de conocimientos del personal.
- Abate pérdidas financieras y publicidad negativa

1.2.2.1. Establece una metodología

Para ayudar en la selección de herramientas para una evaluación de vulnerabilidad, es útil establecer una metodología de evaluación de la vulnerabilidad. Infortunadamente, no existe una metodología predefinida o aprobada por la industria en este momento, sin embargo, el sentido común y los buenos hábitos pueden actuar como una guía completa.

¿Qué es un destino? Estamos buscando en un servidor, o estamos buscando en la red total y en todo lo que hay dentro de la red? ¿Somos externos o internos para la compañía? Las respuestas a estas preguntas son importantes ya que ayudan a determinar no solamente cuáles herramientas se deben seleccionar sino también la forma como se utilizan.

Para obtener mayor información sobre el establecimiento de metodologías, consulte las siguientes páginas web:

- *http://www.isecom.org/osstmm/ The Open Source Security Testing Methodology Manual* (OSSTMM)
- *http://www.owasp.org/ The Open Web Application Security Project*

1.2.3. Evaluación de herramientas

La evaluación puede comenzar con el uso de alguna forma de herramienta de recolección de información. Cuando se evalúa toda la red, trace primero un mapa para identificar las máquinas que se están ejecutando. Una vez localizadas, examine cada máquina individualmente. Para enfocarse

en estas máquinas requiere otro conjunto de herramientas. Conocer la herramienta que debe utilizar puede ser el paso más importante para hallar vulnerabilidades.

Al igual que en cualquier aspecto de la vida cotidiana, hay muchas herramientas diferentes que pueden hacer el mismo trabajo. Este concepto se aplica a la realización de evaluaciones de vulnerabilidad. Hay herramientas específicas para los sistemas operativos, aplicaciones y hasta redes (basadas en los protocolos utilizados). Algunas herramientas son gratuitas, mientras que otros no lo son. Algunas herramientas son intuitivas y fáciles de usar, mientras que otras son enigmáticas y están muy mal documentadas pero tienen características que otras no tienen.

Encontrar la herramienta adecuada puede ser una tarea de enormes proporciones y, al final, la experiencia cuenta. Si es posible, establezca un laboratorio de pruebas y evalúe tantas herramientas como pueda, anotando las fortalezas y debilidades de cada una. Revise el archivo README o la página de manual de la herramienta. Además, revise la Internet para obtener más información, como por ejemplo, artículos, guías paso a paso o incluso, listas de correo específicas para la herramienta.

Las herramientas que se abordan a continuación son una muestra de las herramientas disponibles.

1.2.3.1. Cómo escanear hosts con Mmap

Nmap es una herramienta popular que se puede utilizar para determinar el diseño de una red. Nmap ha estado disponible durante muchos años y es probablemente la herramienta más utilizada para recopilar información. Incluye una página excelente de manual que proporciona la descripción detallada de sus opciones y uso. Los administradores pueden usar Nmap en una red para encontrar sistemas host y puertos abiertos en esos sistemas.

Nmap es un primer paso en la evaluación de la vulnerabilidad. Puede asignar todos los hosts dentro de la red y hasta puede pasar una opción que permita a Nmap tratar de identificar el sistema operativo que se ejecuta en un host determinado. Nmap es un buen fundamento para establecer una política de uso de servicios seguros y restringir los servicios no utilizados.

1.2.3.1.1. Uso de Nmap

Nmap puede ejecutarse desde un indicador de shell mediante el comando **nmap** seguido por el nombre de host o la dirección IP de la máquina que va a ser escaneada.

```
nmap foo.example.com
```

Los resultados del escaneo básico (el cual puede tomar más de unos minutos en donde se localiza el host y otras condiciones) deben parecerse a los siguientes:

```
Interesting ports on foo.example.com:
Not shown: 1710 filtered ports
PORT    STATE  SERVICE
22/tcp  open   ssh
53/tcp  open   domain
80/tcp  open   http
113/tcp closed auth
```

Nmap prueba los puertos de comunicación de red más comunes para servicios de escucha o espera. Este conocimiento puede ser útil para un administrador que desee cerrar servicios innecesarios o no utilizados.

Para obtener mayor información sobre el uso de Nmap, consulte la página principal oficial en la siguiente URL:

http://www.insecure.org/

1.2.3.2. Nessus

Nessus es un escáner de seguridad de servicio completo. La arquitectura del complemento de Nessus permite a los usuarios ajustarlo a sus sistemas y redes. Al igual que con cualquier escáner, Nessus es tan bueno como la base de datos de la que depende. Afortunadamente, Nessus es actualizado con frecuencia y pone en relieve un reporte completo, escaneo de host y búsquedas de vulnerabilidades en tiempo real. Recuerde que pueden haber falsos positivos y falsos negativos, incluso en una herramienta actualizada tan frecuentemente como Nessus.

Nota

El cliente de Nessus y el software de servidor necesita una suscripción. Puede incluirse en este documento como referencia para los usuarios que no están interesados en usar esta aplicación.

Para obtener mayor información sobre Nessus, consulte la página web oficial en la siguiente URL:

http://www.nessus.org/

1.2.3.3. Nikto

Nikto es un escáner de script (CGI) de una interfaz de puerta de enlace común excelente pero lo hace en una forma evasiva, para eludir sistemas de detección de intrusos. Viene con una documentación detallada que debe ser revisada antes de ejecutar el programa. Si tiene servidores web que sirven los scripts CGI, Nikto puede ser un recurso excelente para revisar la seguridad de estos servidores.

Para obtener mayor información sobre Nikto, consulte la siguiente URL:

http://cirt.net/nikto2

1.2.3.4. Anticipación a sus necesidades futuras

Según sus objetivos y recursos, hay muchas herramientas disponibles. Hay herramientas para redes inalámbricas, redes Novell, sistemas de Windows, sistemas de Linux y muchas más. Otra parte esencial de realizar evaluaciones puede incluir la revisión de la seguridad física, la inspección de personal, o la evaluación de redes de Voice o PBX. Nuevos conceptos tales como *war walking* y *wardriving*, los cuales tienen que ver con el escaneo del perímetro de las estructuras físicas de la empresa para vulnerabilidades de redes inalámbricas, son algunos conceptos que debe investigar y, si es necesario, incorporar dentro de sus evaluaciones. La imaginación y exposición son únicamente límites de planeación y manejo de evaluación de vulnerabilidades.

1.3. Atacantes y vulnerabilidades

Para planificar e implementar una buena estrategia de seguridad, tenga en cuenta primero los aspectos que determinaron a los agresores a explotar los sistemas comprometidos. Sin embargo, antes de describir estos aspectos, se debe definir la terminología utilizada para identificar un agresor.

1.3.1. Breve historia de los hackers

El significado moderno del término *hacker* se remonta a la década de 1960 en el Tech Model Railroad Club (Club de modelo técnico de trenes) del Instituto de Tecnología de Massachusetts (MIT), el cual diseñaba trenes en gran escala y detalle. Hacker era el nombre utiilizado por los miembros del club que descubrían un truco o solución para un problema.

El término hacker desde entonces se utiliza para describir todo desde aficionados a los computadores hasta programadores talentosos, Un rasgo común entre los hackers es el interés de explorar en detalle cómo funcionan los sistemas informáticos y las redes con muy poca o ninguna motivación externa. Los desarrolladores de software de código abierto suelen considerar a sus colegas y a sí mismos como hackers y usar ese término como un término de respeto.

Normalmente, los hackers siguen la *ética del hacker*, la cual dicta que la búsqueda de información y experiencia es esencial y que es deber de los hackers compartir a la comunidad ese conocimiento. Durante esa búsqueda de conocimiento, algunos hackers disfrutan los retos académicos de burlar los controles de seguridad en sistemas informáticos. Por esta razón, la prensa suele usar el término hacker para describir a quienes acceden ilícitamente a los sistemas y redes con intención inescrupulosa, maliciosa o delictiva. El término más adecuado para este tipo de hacker es *cracker* - un término creado por los hackers a mediados de la década de 1980 para diferenciar las dos comunidades.

1.3.1.1. Tonos de gris

Dentro de la comunidad de individuos que buscan y explotan vulnerabilidades en sistemas y redes hay varios grupos distintivos. Dichos grupos suelen describirse por tonos de sombreros que 'llevan puestos' cuando realizan sus investigaciones de seguridad y este tono es indicativo de la intención.

El *hacker de sombrero blanco* es el que prueba las redes y sistemas para examinar su rendimiento y determinar qué tan vulnerables son para un intruso. Por lo general, los hackers de sombrero blanco descifran sus propios sistemas o sistemas de un cliente que los ha empleado específicamente con propósitos de auditoría de seguridad. Los investigadores académicos y consultores profesionales son dos ejemplos de hackers de sombreros blancos.

El *hacker de sombrero negro* es sinónimo de cracker. En general, los crackers se enfocan menos en la programación y la parte académica para entrar en los sistemas. No dependen de programas para descifrar. Suelen depender de programas para descifrar y explotar las vulnerabilidades en sistemas par descubrir información confidencial para ganancia personal o infligir daño en el sistema o red.

Por otra parte, el *hacker de sombrero gris*, tiene las destrezas e intensiones del hacker de sombrero blanco en la mayoría de los casos pero utiliza su conocimiento para propósitos menos nobles. Un hacker de sombrero gris puede pensarse como un hacker de sombrero blanco que a veces porta el sombrero negro para cumplir con su propia agenda.

Los hackers de sombrero gris normalmente se adhieren a otra forma de ética de hacker, la cual dice que se acepta ingresar ilegalmente en los sistemas siempre y cuando el hacker no cometa ningún robo de confidencialidad. Algunas personas pueden argumentar que el acto de ingresar ilegalmente a un sistema lo hace de por sí no ético.

Independiente de la intención del intruso, es importante saber las debilidades que un cracker probablemente puede tratar de aprovechar. La parte restante de este capítulo se enfoca en esos aspectos.

1.3.2. Amenazas a la seguridad de la red

Las prácticas erradas al configurar los siguientes aspectos de una red pueden aumentar el riesgo de un ataque.

1.3.2.1. Arquitecturas inseguras

Una red mal configurada es un punto de entrada a usuarios no autorizados. Abandonar un sitio confiable y abrir una red local vulnerable a la Internet que es altamente insegura, es como si dejara la

puerta entreabierta en un vecindario de alta delincuencia organizada — no ocurre nada por un tiempo, pero *al final* alguien aprovecha la oportunidad.

1.3.2.1.1. Redes de difusión

Los administradores de sistemas suelen no dar importancia al hardware de red en sus esquemas de seguridad. El hardware sencillo tal como concentradores y enrutadores depende del principio de difusión o del principio de no-conmutado; es decir, cada vez que un nodo transmite datos a través de la red a un nodo receptor, el concentrador o enrutador envía una transmisión de paquetes de datos hasta que el nodo recipiente reciba y procese los datos. Este método es el más vulnerable para el protocolo de resolución de direcciones (*ARP*) o enmascaramiento de direcciones de control de acceso de medios (*MAC*) tanto por intrusos externos como por usuarios no autorizados en hosts locales.

1.3.2.1.2. Servidores centralizados

Otra falla potencial de redes es el uso de computación centralizada. Una forma común de reducción de costos para muchos negocios, es la de consolidar todos los servicios para una sola máquina poderosa. Esto puede ser conveniente, ya que es más fácil de manejar y cuesta considerablemente menos que la configuración de múltiples servidores. Sin embargo, un servidor centralizado introduce un punto único de fallo en la red. Si el servidor central está comprometido, puede dejar la red totalmente inútil o peor aún, con tendencia a la manipulación de datos o robo. En estos casos, el servidor central se convierte en una puerta abierta a toda la red.

1.3.3. Amenazas a la seguridad del servidor

La seguridad del servidor es tan importante como la seguridad de la red porque los servidores suelen contener una gran cantidad de información vital de una organización. Si un servidor está comprometido, todos sus contenidos pueden estar disponibles para que el cracker robe o manipule a su antojo. Las siguientes secciones describen algunos de los temas principales.

1.3.3.1. Servicios no utilizados y puertos abiertos

Una instalación completa de Red Hat Enterprise Linux 6 contiene 1.000+ aplicaciones y paquetes de la biblioteca. Sin embargo, la mayoría de los administradores de servidores optan por no instalar todos los paquetes de la distribución, y prefieren una instalación base de paquetes que incluya varias aplicaciones de servidor.

Es muy común entre los administradores de sistemas instalar el sistema operativo sin prestar atención a qué programas están siendo realmente instalados. Esto puede ser problemático porque se pueden instalar servicios innecesarios con la configuración predeterminada, y posiblemente se activen. Esto puede hacer que servicios indeseados, tales como Telnet, DHCP o DNS, se ejecuten en un servidor o estación de trabajo sin que el administrador se de cuenta, lo cual a su vez, puede ocasionar tráfico indeseado para el servidor o incluso, establecer una ruta potencial para crackers. Consulte *Sección 2.2, “Seguridad del servidor”* para obtener información sobre cómo cerrar puertos y desactivar los servicios no utilizados.

1.3.3.2. Servicios sin parches

La mayoría de las aplicaciones de servidor que se incluyen en una instalación predeterminada son partes sólidas de software probadas a fondo. Después de haber estado en uso en entornos de producción desde hace muchos años, su código ha sido refinado en detalle y muchos de los errores han sido encontrados y corregidos.

Sin embargo, no hay un software perfecto y siempre habrá espacio para un mayor refinamiento. Por otra parte, el nuevo software no suele ser probado tan rigurosamente como es de esperar, debido a

su reciente llegada a los entornos de producción o porque puede que no sea tan popular como otras aplicaciones de servidores.

Los desarrolladores y administradores de sistemas suelen encontrar errores en las aplicaciones de servidor y publicar la información sobre el seguimiento de errores y sitios web relacionados con la seguridad como la lista de correo Bugtraq (*http://www.securityfocus.com*) o the Computer Emergency Response Team (CERT) website (*http://www.cert.org*). Aunque estos mecanismos son una forma efectiva de alertar a la comunidad sobre las vulnerabilidades de seguridad, le corresponde a los administradores del sistema corregir los sistemas a tiempo. Esto es particularmente cierto porque los crackers tienen acceso a estos mismos servicios de seguimiento de vulnerabilidades y utilizarán la información para violar sistemas no actualizados siempre que puedan. La buena administración del sistema requiere vigilancia, seguimiento constante de errores y mantenimiento de sistemas apropiado para asegurar un entorno informático más seguro.

Consulte *Sección 1.5, "Actualizaciones de seguridad"* para obtener mayor información sobre cómo mantener actualizado el sistema.

1.3.3.3. Administración inatenta

Los administradores que no corrigen los sistemas son los más propensos a las mayores amenazas a la seguridad del servidor. Según el *SysAdmin, Audit, Network, Security Institute* (*SANS*), la principal causa de vulnerabilidad de seguridad informática es "designar personas no entrenadas para mantener la seguridad y no proporcionar ni la formación ni el tiempo para hacer posible este trabajo."[10] Esto se aplica tanto a los administradores sin experiencia como a los administradores desmotivados o demasiado seguros.

Algunos administradores fallan en corregir los servidores y estaciones de trabajo, mientras que otros no pueden ver mensajes de registro desde el kernel del sistema o el tráfico de red. Otro error común es cuando no se cambian las contraseñas o llaves a los servicios. Por ejemplo, algunas bases de datos tienen contraseñas administrativas predeterminadas porque los desarrolladores asumen que el administrador del sistema cambia las contraseñas inmediatamente después de la instalación. Si el administrador de base de datos no cambia la contraseña, incluso un cracker sin experiencia puede utilizar la contraseña predeterminada ampliamente conocida para obtener privilegios de administrador para la base de datos. Estos son solamente algunos ejemplos de cómo la falta de atención del administrador puede llevar a servidores comprometidos.

1.3.3.4. Servicios intrínsecamente inseguros

Incluso la organización más atenta y vigilante puede ser victima de vulnerabilidades si los servicios de red que seleccionan son intrínsecamente inseguros. Por ejemplo, hay muchos servicios desarrollados bajo el supuesto de que se utilizan en redes de confianza, sin embargo, este supuesto falla tan pronto como el servicio está disponible a través de la Internet - la cual es en sí misma insegura.

Los servicios de red inseguros son los servicios que requieren nombres de usuario y contraseñas sin cifrar. Por ejemplo, Telnet y FTP son dos de esos servicios. Si el software de paquetes de husmeo monitoriza el tráfico entre el usuario remoto y un servicio tal los nombres de usuario y contraseñas pueden ser interceptadas fácilmente.

Naturalmente, estos servicios también pueden ser presa fácil de lo que el sector de seguridad industrial llama ataque de *hombre en el medio*. En este tipo de ataque, el cracker redirige el tráfico de red engañando a un servidor de nombres descifrado en la red para que apunte a su máquina en vez de al servidor en cuestión. Cuando alguien abra una sesión remota en el servidor, la máquina

[10] http://www.sans.org/resources/errors.php

del atacante actúa como un conductor invisible, instalada en silencio capturando información entre el servicio remoto y los usuarios desprevenidos. De este modo, un cracker puede reunir contraseñas administrativas y datos sin que el servidor o el usuario se den cuenta.

Otra categoría de servicios inseguros incluyen los sistemas de archivos de red y los servicios de información tales como NFS o NIS, los cuales son desarrollados específicamente para uso de LAN pero, infortunadamente, se extienden para incluir las WAN (para los usuarios remotos). NFS no lo hace, no tiene ningún tipo de mecanismo de autenticación y de seguridad configurados para evitar que un cracker monte el recurso compartido de NFS y acceda a todo el contenido. NIS también tiene información vital que debe ser conocida por cada equipo en una red, entre ella contraseñas y permisos de archivos, dentro de una base de datos de texto plano de ASCII o DBM (derivado de ASCII). El cracker que obtiene acceso a esta base de datos puede acceder a cada cuenta de usuario en la red, entre ellas la cuenta del administrador.

Red Hat Enterprise Linux se lanza de forma predeterminada con todos los servicios desactivados. Sin embargo, dado que a menudo los administradores se ven obligados a utilizar estos servicios, la configuración cuidadosa es fundamental. Consulte la *Sección 2.2, “Seguridad del servidor”* para obtener más información sobre la configuración segura de servicios.

1.3.4. Amenazas a la estación de trabajo y a la seguridad del computador personal

Las estaciones de trabajo y PC del hogar pueden no ser tan susceptibles a ataques como las redes o servidores, pero pueden contener datos confidenciales, como información de tarjetas de crédito, que son el blanco de los crackers del sistema. Las estaciones de trabajo también pueden ser elegidas sin conocimiento del usuario y utilizadas por los agresores como máquinas "esclavas" en ataques coordinados. Por estas razones, si conoce las vulnerabilidades de una estación de trabajo puede ahorrar a los usuarios el dolor de cabeza de reinstalar el sistema operativo, o peor aún, recuperarse del robo de datos.

1.3.4.1. Malas contraseñas

Las malas contraseñas son una de las formas más fáciles para que un agresor obtenga acceso a un sistema. Para obtener mayor información sobre cómo evitar errores comunes al crear una contraseña, consulte la *Sección 2.1.3, “Seguridad de contraseña”*.

1.3.4.2. Aplicaciones de clientes vulnerables

El hecho de que el administrador tenga un servidor completamente seguro y corregido, no significa que los usuarios remotos estén protegidos cuando acceden a él. Por ejemplo, si el servidor ofrece servicios Telnet o FTP sobre una red pública, un agresor puede capturar los nombres de usuario y contraseñas de texto plano a medida que pasan por la red, y luego usar la información de la cuenta para acceder a la estación de trabajo del usuario remoto.

Incluso cuando se utilizan protocolos seguros, tales como SSH, un usuario remoto puede ser vulnerable a ciertos ataques si no mantiene sus aplicaciones de cliente actualizadas. Por ejemplo, los clientes v.1 SSH son vulnerables a un ataque de reenvío de X desde servidores SSH maliciosos. Una vez conectado al servidor, el agresor puede capturar en silencio cualquiera de las pulsaciones de teclado y los clics del ratón hechos por el cliente en la red. Este problema se corrigió con el protocolo v.2 SSH, pero es responsabilidad del usuario hacer un seguimiento de las aplicaciones que tienen tales vulnerabilidades y actualizarlas si es necesario.

Sección 2.1, “Seguridad de estación de trabajo” aborda en más detalle los pasos que los administradores y usuarios domésticos deben seguir para limitar la vulnerabilidad de las estaciones de trabajo de computadores.

1.4. Vulnerabilidades y ataques comunes

Tabla 1.1, "Vulnerabilidades comunes" describe algunas de las vulnerabilidades más comunes y puntos de entrada utilizados por intrusos para acceder a los recursos de redes empresariales. Lo clave para estas vulnerabilidades comunes son las explicaciones de cómo se realizan y cómo los administradores pueden proteger su red contra dichos ataques.

Tabla 1.1. Vulnerabilidades comunes

Vulnerabilidades	Descripción	Notas
Contraseñas predeterminadas o ninguna	El hecho de dejar las contraseñas en blanco o de usar una contraseña predeterminada por el fabricante. Es lo más común en hardware tales como enrutadores y cortafuegos, aunque algunos servicios que se ejecutan en Linux pueden contener contraseñas de administrador predeterminadas (aunque Red Hat Enterprise Linux no se distribuye con ellas).	Comúnmente asociadas con hardware de redes tales como enrutadores, cortafuegos, VPN y dispositivos de almacenamiento conectados a redes (NAS). Comunes en muchos sistemas operativos existentes, especialmente aquellos que empaquetan servicios (tales como UNIX y Windows). Algunas veces los administradores crean cuentas de usuario privilegiadas de afán y dejan la contraseña en blanco, creando el punto perfecto para que usuarios maliciosos descubran la cuenta.
Llaves compartidas predeterminadas	Los servicios seguros algunas veces empaquetan las llaves de seguridad predeterminadas para propósitos de desarrollo o de evaluación. Si estas llaves no se cambian y se sitúan en un entorno de producción en la Internet, *todos* los usuarios con las mismas llaves predeterminadas tendrán acceso a ese recurso de llave compartida y a cualquier información confidencial que la contenga.	Los puntos de acceso inalámbricos y dispositivos de servidor seguro preconfigurados más comunes.
Suplantación de IP	Una máquina remota actúa como un nodo en su red local, encuentra vulnerabilidades con sus servidores e instala un programa trasero o Caballo de Troya para obtener control sobre los recursos de la red.	La suplantación es bastante difícil ya que el agresor debe predecir los números de secuencia TCP/IP para coordinar la conexión a sistemas de destino, aunque hay varias herramientas disponibles para que los atacantes realicen dicha agresión. Depende de los servicios que se ejecuten en el sistema de destino (tales como **`rsh`**, **`telnet`**, FTP y otros) que usan técnicas de autenticación *source-based*, las cuales no se recomiendan cuando se comparan a PKI u otras formas de autenticación de cifrado utilizadas en **`ssh`** o SSL/TLS.
Interceptación pasiva	Recolectar los datos que pasan entre dos nodos activos en la red mediante	El tipo de ataque funciona principalmente con protocolos de

Vulnerabilidades	Descripción	Notas
	interceptación pasiva en la conexión entre los dos nodos.	transmisión de texto plano tales como transferencias de Telnet, FTP y HTTP. El agresor remoto debe tener acceso al sistema comprometidos en un LAN para poder realizar dicho ataque. Por lo general, el ciberpirata ha empleado un ataque activo (tal como suplantación de IP o tercero interpuesto) para comprometer un sistema en el LAN. Medidas preventivas incluyen servicios con intercambio de llave criptográfica, contraseñas de una sola vez o autenticación cifrada para evitar suplantación de contraseñas; también se recomienda el cifrado fuerte durante la transmisión.
Vulnerabilidades de servicios	El atacante busca una falla o debilidad en un servicio en la red; a través de esta vulnerabilidad, el agresor compromete todo el sistema y los datos que pueda contener y posiblemente comprometa otros sistemas en la red.	Los servicios basados en HTTP tales como CGI son vulnerables a la ejecución de comandos remotos e incluso al acceso de shell interactivo. Incluso el servicio HTTP se ejecuta como usuario sin privilegios tales como información de "nadie", información tal como archivos de configuración y mapas de redes que pueden leer o el atacante puede iniciar o negar el ataque del servicio que drena los servicios del sistema o los convierte en no disponibles para otros usuarios. Algunas veces los servicios pueden tener vulnerabilidades que no se notan durante el desarrollo y evaluación: estas vulnerabilidades (tales como *sobreflujos de buffer*, donde los agresores atacan un servicio con valores arbitrarios que llenan el buffer de la memoria de una aplicación, dando a los agresores un indicador de comandos interactivo desde el cual pueden ejecutar comandos arbitrarios) pueden dar control administrativo total a un agresor. Los administradores deben asegurarse de no operar como usuarios de root, y deben estar alertas a los parches y actualizaciones de erratas para aplicaciones de los proveedores u organizaciones de seguridad tales como CERT y CVE.

Vulnerabilidades	Descripción	Notas
Vulnerabilidades de aplicaciones	Los atacantes buscan fallas en el escritorio y aplicaciones de trabajo (tales como clientes de correo-e) y ejecutan código arbitrario, implantan caballos de Troya para compromiso futuro o para dañar sistemas. Otras vulnerabilidades se pueden presentar si la estación de trabajo tiene privilegios administrativos en la parte restante de la red.	Las estaciones de trabajo y escritorios son más propensas a vulnerabilidades ya que los trabajadores no tienen la experiencia para evitar o detectar un compromiso; es imperativo informar a los individuos de los riesgos que corren cuando instalan software no autorizado o abren anexos de correo no solicitado. La protección puede implementarse para que ese software de clientes de correo-e no se abra automáticamente o ejecute archivos adjuntos. Además, la actualización automática del software de estación de trabajo vía Red Hat Network u otros servicios administrativos de sistemas pueden aliviar las cargas de seguridad de implementaciones de seguridad de varios puestos.
Ataques de denegación de servicio (DoS)	El atacante o grupo de atacantes coordina contra una red de organización o recursos de servidor al enviar paquetes no autorizados al host de destino (ya sea servidor, enrutador o estación de trabajo). De esta manera se fuerza al recurso a convertirse en disponible para usuarios legítimos.	El caso de DoS más reportado en los Estados Unidos ocurrió en 2000. Varios sitios comerciales y gubernamentales quedaron no disponibles por un ataque coordinado de ping mediante varios sistemas comprometidos con conexiones de ancho de banda actuando como *zombies*, o nodos de transmisión redirigidos . Los paquetes de fuente generalmente se falsifican (como también se retransmiten), lo que dificulta la investigación de la fuente verdadera del ataque. Avances en filtrado de ingreso (IETF rfc2267) mediante **`iptables`** y Sistemas de detección de Intrusos de Redes tales como **`snort`** ayudan a los administradores a rastrear y a evitar ataques DoS distribuidos.

1.5. Actualizaciones de seguridad

Cuando se descubren vulnerabilidades de seguridad, se debe actualizar el software afectado para limitar los riesgos potenciales de seguridad. Si el software es parte de un paquete dentro de una distribución soportada de Red Hat Enterprise Linux, Red Hat se compromete a producir los paquetes de actualización que reparan las vulnerabilidades tan pronto como sea posible. A menudo, el anuncio de una falla de seguridad viene acompañado de un parche (o el código de origen que soluciona el problema). Este parche se aplica al paquete de Red Hat Enterprise Linux, se prueba y distribuye como una actualización de erratas. Sin embargo, si el anuncio no incluye el parche, el programador

primero trabaja con el mantenedor del software para solucionar el problema. Una vez solucionado el problema, el paquete es probado y distribuido como una actualización de erratas.

Si se lanza una actualización de erratas para software utilizado en su sistema, se recomienda encarecidamente actualizar los paquetes tan pronto como sea posible para minimizar el tiempo en que el sistema está potencialmente vulnerable.

1.5.1. Actualización de paquetes

Al actualizar el software en un sistema, es importante descargar la actualización desde una fuente confiable. Un intruso puede fácilmente reconstruir un paquete con el mismo número de versión como el que se supone que debe solucionar el problema, pero con una vulnerabilidad de seguridad diferente y distribuirlo en Internet. Si esto sucede, el uso de medidas de seguridad tales como la verificación de archivos con el RPM original no detectará la agresión. Por lo tanto, es muy importante solamente descargar los RPM desde fuentes confiables, tales como Red Hat y comprobar la firma del paquete para verificar su integridad.

Nota

Red Hat Enterprise Linux incluye un icono de panel muy conveniente que muestra señales visibles cuando hay una actualización disponible.

1.5.2. Cómo verificar paquetes firmados

Todos los paquetes de Red Hat Enterprise Linux son firmados con la llave *GPG* de Red Hat. GPG viene de GNU Privacy Guard, o GnuPG, un paquete de software gratuito utilizado para asegurar la autenticidad de los archivos distribuidos. Por ejemplo, una clave privada (clave secreta) bloquea el paquete mientras que la clave pública desbloquea y verifica el paquete. Si la clave pública distribuida por Red Hat Enterprise Linux no coincide con la clave privada durante la verificación de RPM, el paquete puede haber sido alterado y por lo tanto no es confiable.

La herramienta de RPM dentro de Red Hat Enterprise Linux 6 automáticamente trata de verificar la firma de GPG de un paquete RPM antes de instalarlo. Si la llave GPG de Red Hat no está instalada, instálela desde un sitio seguro, estático, tal como una instalación un CD o DVD de instalación de Red Hat.

Suponiendo que el disco está montado en **`/mnt/cdrom`**, use el siguiente comando para importarlo en el archivo de claves *archivo de llaves* (una base de datos de llaves confiables en el sistema):

```
rpm --import /mnt/cdrom/RPM-GPG-KEY
```

Para desplegar una lista de todas las llaves instaladas para verificación de RPM, ejecute el siguiente comando:

```
rpm -qa gpg-pubkey*
```

La salida será similar a la siguiente:

```
gpg-pubkey-db42a60e-37ea5438
```

Para desplegar información sobre la llave específica, use el comando **`rpm -qi`** seguido por la salida del comando anterior, como el siguiente ejemplo:

```
rpm -qi gpg-pubkey-db42a60e-37ea5438
```

Es muy importante verificar la firma de los archivos RPM antes de instalarlos para garantizar que no han sido alterados de su fuente original de paquetes. Para verificar todos los paquetes descargados, emita el siguiente comando:

```
rpm -K /tmp/updates/*.rpm
```

Por cada paquete, si la llave de GPG es correcta, el comando retorna **gpg OK**. Si no lo es, verifique si está utilizando la llave pública apropiada de Red Hat y verifique también el origen del contenido. Los paquetes que no pasan las verificaciones de GPG no se deben instalar, ya que pueden haber sido alterados por un tercero.

Tras verificar la llave GPG y descargar todos los paquetes asociados al reporte de erratas, instale los paquetes como root en el indicador de shell.

1.5.3. Cómo instalar paquetes firmados

La instalación de la mayoría de paquetes (excepto los paquetes de kernel) puede hacerse mediante el siguiente comando:

```
rpm -Uvh /tmp/updates/*.rpm
```

Para paquetes de kernel utilice el siguiente comando:

```
rpm -ivh /tmp/updates/<kernel-package>
```

Remplace <*kernel-package*> en el ejemplo anterior por el nombre del RPM de kernel.

Una vez que se ha vuelto a arrancar la máquina mediante el nuevo kernel, se puede retirar el kernel anterior mediante el siguiente comando:

```
rpm -e <old-kernel-package>
```

Remplace <*old-kernel-package*> en el ejemplo anterior por el nombre del RPM de kernel anterior.

Nota

No se requiere retirar el kernel anterior. El gestor de arranque predeterminado, GRUB, permite que haya varios kernel instalados, que se eligen desde un menú en el momento de arranque.

Importante

Antes de instalar erratas de seguridad, asegúrese de leer las instrucciones especiales contenidas en el reporte de erratas y ejecútelas como corresponde. Consulte la *Sección 1.5.4, "Cómo aplicar los cambios"* para obtener instrucciones generales sobre los cambios hechos por una actualización de erratas.

1.5.4. Cómo aplicar los cambios

Tras descargar e instalar las erratas y actualizaciones de seguridad, es importante detener el uso del software anterior y comenzar a usar el nuevo software. La forma como se hace esto depende del tipo de software que se ha actualizado. La lista siguiente muestra las categorías generales de software y proporciona instrucciones para utilizar las versiones actualizadas después de una actualización del paquete.

Nota

En general, la mejor manera para asegurarse de que se utiliza la última versión de un paquete de software, es reiniciar el sistema. Sin embargo, esta opción no siempre se necesita o está disponible para el administrador del sistema.

Aplicaciones

Las aplicaciones de espacio de usuario son los programas que pueden ser iniciados por un usuario del sistema. Normalmente, tales aplicaciones se utilizan únicamente cuando un usuario, script o tarea automática las abre y no persisten por largos períodos de tiempo.

Cuando una aplicación tal de espacio de usuario se actualiza, detenga cualquier instancia de la aplicación en el sistema y lance el programa nuevamente para así utilizar la versión actualizada.

Kernel

El kernel es el componente de software de núcleo para el sistema operativo de Red Hat Enterprise Linux. Administra el acceso a memoria, al procesador y periféricos como también programa todas las tareas.

Debido a su papel central, el kernel no puede reiniciar el equipo sin que se detenga. Por lo tanto, la versión actualizada del kernel no se puede utilizar hasta que el sistema no sea reiniciado.

Bibliotecas compartidas

Las bibliotecas compartidas son unidades de código, tales como **glibc**, las cuales son utilizadas por un número de aplicaciones y servicios. Las aplicaciones que utilizan una biblioteca compartida por lo general cargan el código compartido cuando se inicia la aplicación, por lo tanto las aplicaciones que utilizan la biblioteca actualizada se deben detener y volver a abrir.

Para determinar cuáles aplicaciones en ejecución se enlazan con una biblioteca determinada, use el comando **lsof** como en el siguiente ejemplo:

```
lsof /lib/libwrap.so*
```

Este comando retorna una lista de todos los programas en ejecución que utilizan envolturas TCP para controlar el acceso de host. Por lo tanto, cualquier programa en la lista debe ser detenido y relanzado al actualizar el paquete **tcp_wrappers**.

Servicios de SysV
Los servicios de SysV son programas de servidor persistente ejecutados durante el proceso de arranque. Los ejemplos de servicios de SysV incluyen **sshd**, **vsftpd** y **xinetd**.

Debido a que estos programas suelen persistir en la memoria, siempre y cuando la máquina sea reiniciada, cada servicio de SysV actualizado debe detenerse y relanzarse después de actualizar el paquete. Esto se puede hacer con la **Herramienta de configuración de servicios** , o ingresando en un indicador de comandos de shell de root y emitiendo el comando **/sbin/service** como en el ejemplo siguiente:

```
/sbin/service <service-name> restart
```

En el ejemplo anterior, remplace <*service-name*> con el nombre del servicio, tal como **sshd**.

xinetd Services
Los servicios controlados por el súper servicio de **xinetd** solamente se ejecutan cuando hay una conexión activa. Entre los ejemplos de servicios controlados por **xinetd** se incluyen Telnet, IMAP y POP3.

Puesto que **xinetd** lanza nuevas instancias de estos servicios cada vez que se recibe una solicitud, las conexiones que suceden después de una actualización son manejadas por el software actualizado. Sin embargo, si hay conexiones activas en el momento en que el servicio controlado de **xinetd** es actualizado, son manejadas por la versión anterior del software.

Para matar instancias anteriores de un determinado servicio controlado de **xinetd**, actualice el paquete para el servicio y luego detenga todos los procesos que se estén ejecutando en el momento. Para determinar si el proceso se está ejecutando, use el comando **ps** y luego el comando **kill** o **killall** para detener las instancias actuales del servicio.

Por ejemplo, si se lanzan los paquetes de erratas de seguridad **imap**, actualice los paquetes y luego escriba el siguiente comando como root en el indicador de comanodos de shell:

```
ps aux | grep imap
```

Este comando retorna todas las sesiones IMAP activas. Para finalizar las sesiones individuales utilice el siguiente comando:

```
kill <PID>
```

Si este comando no puede terminar la sesión, use le siguiente comando en su lugar:

```
kill -9 <PID>
```

En los ejemplos anteriores, remplace <*PID*> por el número de identificación del proceso (que se encuentra en la segunda columna del comando **ps**) para una sesión de IMAP.

Para matar todas las sesiones IMAP activas, emita el siguiente comando:

```
killall imapd
```

Cómo proteger la red

2.1. Seguridad de estación de trabajo

La protección de un entorno de Linux comienza por la estación de trabajo. Ya sea al bloquear una máquina personal o protegiendo un sistema empresarial, la política de seguridad sólidacomienza por el computador personal. La red de computadores es tan segura como su nodo más débil.

2.1.1. Evaluación de la seguridad de la estación de trabajo

Al evaluar la seguridad de una estación de trabajo de Red Hat Enterprise Linux, considere lo siguiente:

- *BIOS y Seguridad del gestor de arranque* — ¿Puede un usuario no autorizada acceder físicamente a la máquina y arrancar como usuario único o modo de rescate sin una contraseña?
- *Seguridad de contraseña* — ¿Qué tan seguras están sus contraseñas de cuenta de usuario en la máquina?
- *Controles administrativos* — ¿Quíen tiene una cuenta en el sistema y cuánto control administrativo tiene?
- *Servicios de redes disponibles* — ¿Qué servicios escuchan las solicitudes desde la red y deben estarse ejecutando?
- *Cortafuegos personales* — ¿Qué tipo de cortafuegos , si lo hay, es necesario?
- *Herramientas de protección de comunicación mejoradas* — ¿Qué herramientas deben utilizarse para comunicarse entre estaciones de trabajo y cuáles deberían evitarse?

2.1.2. BIOS y seguridad del gestor de arranque

La protección de contraseñas para BIOS (o equivalente de BIOS) y el gestor de arranque pueden evitar que usuarios no autorizados tengan acceso físico a sistemas desde el arranque mediante medios extraíbles u obteniendo privilegios de root a través del modo de usuario único. Las medidas de seguridad que usted debe tomar para protegerse de dichos ataques dependen de la confidencialidad de la información y del sitio de la máquina.

Por ejemplo, si se usa una máquina en una feria de exposición y no contiene información confidencial, puede no ser fundamental prevenir esos ataques. Sin embargo, si se descuida el portátil de un empleado con llaves privadas SSH sin cifrar para la red corporativa en la misma exposición, podría conducir a un fallo de seguridad importante con consecuencias para toda la empresa.

Si la estación de trabajo se localiza en un lugar donde solamente personas autorizadas y de confianza tienen acceso, entonces el BIOS o gestor de arranque puede no ser necesario.

2.1.2.1. Contraseñas de BIOS

Las dos razones principales por las cuales se debe proteger el BIOS de un equipo[1]:

[1] Puesto que el BIOS del sistema se diferencia entre los fabricantes, algunos no aceptan la protección de contraseña de cualquier tipo, mientras que otros pueden aceptar un tipo de protección pero no el otro.

1. *Evitar cambios a la configuración del BIOS* — Si los intrusos tienen acceso al BIOS, pueden configurarlo para que arranque desde un disquete o un CD-ROM. Esto les permite entrar en modo de rescate y a su vez, iniciar procesos arbitrarios en el sistema o copiar datos confidenciales.

2. *Evitar arrancar el sistema* — Algunos BIOS permiten la protección de contraseña del proceso de arranque. Cuando están activados, el agresor es obligado a ingresar una contraseña antes de que el BIOS lance el gestor de arranque.

Puesto que los métodos para establecer una contraseña de BIOS varían entre los fabricantes de computadores, consulte las instrucciones específicas del manual del equipo.

Si olvida la contraseña de BIOS, se puede restablecer con puentes en la placa madre o desconectar la pila CMOS. Por esta razón, es una buena práctica bloquear la caja del computador si es posible. Sin embargo, consulte el manual para el equipo o la placa madre antes de intentar desconectar la pila de CMOS.

2.1.2.1.1. Cómo proteger las plataformas que no son x-86

Otras arquitecturas usan diferentes programas para realizar tareas de bajo nivel casi equivalentes a aquellos BIOS en sistemas x86. Por ejemplo, los computadores Intel® Itanium™ usan la shell de la *Interfaz extensible de Firmware* (*EFI*).

Para obtener instrucciones sobre programas para proteger BIOS con contraseñas en otras arquitecturas, consulte las instrucciones del fabricante.

2.1.2.2. Contraseñas de gestor de arranque

Las razones primarias para proteger con contraseñas un gestor de arranque de Linux son las siguientes:

1. *Evitar el acceso en modo monousuario* — Si los agresores pueden arrancar el sistema en modo monousuario, se pueden registrar automáticamente como root sin que se les pida la contraseña de root.

2. *Evitar el acceso a la consola de GRUB* — Si la máquina utiliza a GRUB como el gestor de arranque, el agresor puede utilizar la interfaz del editor de GRUB para cambiar la configuración o para reunir información mediante el comando **`cat`**.

3. *Evitar el acceso a sistemas operativos inseguros* — Si se trata de un sistema de doble arranque, el agresor puede seleccionar un sistema operativo en tiempo de arranque (por ejemplo, DOS), el cual ignora los controles de acceso y los permisos de archivos.

Red Hat Enterprise Linux 6 se distribuye con el gestor de arranque GRUB en la plataforma x86. Para obtener una visión detallada de GRUB, consulte la Guía de Instalación de Red Hat.

2.1.2.2.1. Contraseña de protección de GRUB

Puede configurar GRUB para solucionar los primeros problemas que se listan en la *Sección 2.1.2.2, "Contraseñas de gestor de arranque"* al añadir una directiva de contraseña a su archivo de configuración. Para hacer esto, elija primero una contraseña fuerte, abra un shell, ingrese como root y luego escriba el siguiente comando:

```
/sbin/grub-md5-crypt
```

Cuando se le indique, escriba la contraseña de GRUB y pulse **`Enter`**. De esta manera retorna un MD5 hash de contraseña.

Luego, modifique el archivo de configuración de GRUB **/boot/grub/grub.conf**. Abra el archivo y debajo de la línea **timeout** en la sección principal del documento, añada la siguiente línea:

```
password --md5 <password-hash>
```

Remplace <*password-hash*> por el valor retornado por **/sbin/grub-md5-crypt**[2].

La próxima vez que el sistema reinicie, el menú de GRUB evitará el acceso al editor o interfaz de comandos sin presionar primero la **p** seguida de la contraseña de GRUB.

Lamentablemente, esta solución no evita que el agresor ingrese en un sistema operativo inseguro en el entorno de doble arranque. Para esto, se debe editar una parte del archivo **/boot/grub/grub.conf**.

Busque la línea de **title** del sistema operativo que desea proteger y añada una línea con la directiva **lock** inmediatamente.

Para un sistema de DOS, la estanza debe comenzar así:

```
title DOS lock
```

Advertencia

Una línea de **password** debe estar presente en la sección principal del archivo **/boot/grub/grub.conf** para que este método funcione correctamente. De lo contrario puede acceder a la interfaz del editor de GRUB y retirar la línea de bloqueo.

Para crear una contraseña diferente para un kernel o sistema operativo determinado, añada la línea **lock** a la estanza, seguida de una línea de contrasaña.

Cada estanza protegida por una contraseña única debe comenzar por líneas similares a las del ejemplo a continuación:

```
title DOS lock password --md5 <password-hash>
```

2.1.3. Seguridad de contraseña

Las contraseñas son un método primario que Red Hat Enterprise Linux utiliza para verificar la identidad de usuario. Esta el la razón por la cual la seguridad de la contraseña es tan importante para proteger al usuario, la estación de trabajo y la red.

Por razones de seguridad, el programa de instalación configura el sistema para usar *Secure Hash Algorithm 512* (*SHA512*) y contraseñas ocultas. Se recomienda no alterar esta configuración.

Si las contraseñas ocultas se desactivan durante la instalación, todas las contraseñas se almacenan como un hash de una vía en el archivo **/etc/passwd**, lo que hace al sistema vulnerable a ataques de piratas de contraseñas fuera de línea. Si el intruso puede obtener acceso a la máquina como un

[2] GRUB también acepta contraseñas sin cifrar, pero se recomienda utilizar un MD5 hash para añadir seguridad.

usuario normal, puede copiar el archivo **/etc/passwd** en su propia máquina y ejecutar cualquier cantidad de programas para descifrar las contraseñas. Si hay una contraseña insegura en el archivo, es sólo cuestión de tiempo antes de que el pirata la descubra.\n

Las contraseñas ocultas eliminan este tipo de ataque al almacenar hash de contraseñas en el archivo **/etc/shadow**, el cual únicamente puede ser leído por el usuario root.

Esto obliga al agresor potencial a intentar descubrir la contraseña de forma remota al ingresar a servicios de redes tales como SSH o FTP. Este tipo de ataque de fuerza bruta es mucho más lento y deja un rastro evidente, pues los intentos fallidos de conexión son registrados en los archivos del sistema. Por supuesto, si el cracker comienza un ataque en medio de la noche en un sistema con contraseñas débiles, el atacante podrá obtener acceso antes del amanecer y editar los archivos de registro para cubrir sus huellas.

Además del formato y las consideraciones de almacenamiento está el problema del contenido. Lo más importante que el usuario debe hacer para proteger la cuenta de un ataque de violación de contraseñas es crear una contraseña fuerte.

2.1.3.1. Cómo crear contraseñas fuertes

Al crear una contraseña segura, es una buena idea seguir estos lineamientos:

- *No utilice solo palabras o números* — Nunca use únicamente números o palabras en la contraseña.

  Algunos ejemplos inseguros se incluyen a continuación:

  - 8675309
  - juan
  - hackme

- *No utilice palabras reconocibles* — Palabras tales como nombres propios, palabras de diccionario, o incluso términos de los programas de televisión o novelas deben evitarse, incluso si terminan en números.

  Algunos ejemplos inseguros se incluyen a continuación:

  - john1
  - DS-9
  - mentat123

- *No utilice palabras en otro idioma* — Los programas para descubrir contraseñas suelen comparar todas las listas de los diccionarios en muchos idiomas. Depender de idiomas extranjeros para proteger sus contraseñas no es seguro.

  Algunos ejemplos inseguros se incluyen a continuación:

  - cheguevara
  - bienvenido1
  - 1dumbKopf

- *No utilice terminología de hacker* — Si piensa que pertenece a una élite porque utiliza en su contraseña terminología de hacker — conocida también como l337 o LEET — ¡piénselo dos veces!. Muchas listas de palabras incluyen LEET.

Algunos ejemplos inseguros se incluyen a continuación:

- H4X0R
- 1337

- *No use información personal* — Evite el uso de información personal en sus contraseñas. Si el agresor conoce su identidad, la tarea de deducir su contraseña será más fácil. A continuación se enumeran los tipos de información que se deben evitar al crear una contraseña:

  Algunos ejemplos inseguros se incluyen a continuación:

  - Su nombre
  - Los nombres de mascotas
  - Los nombres de miembros de familia
  - Las fechas de cumpleaños
  - Su número telefónico o código postal

- *No ponga al revés palabras reconocibles* — Los revisores de contraseñas siempre reversan las palabras comunes, por lo tanto al invertir una mala contraseña no la hace más segura.

  Algunos ejemplos inseguros se incluyen a continuación:

  - R0X4H
  - nauj
  - 9-DS

- *No anote su contraseña* — Nunca escriba una contraseña en papel. Es más seguro memorizarla.
- *No use la misma contraseña para todas las máquinas* — Es importante crear contraseñas independientes para cada máquina. De esta forma si un sistema pierde su caracter confidencial, no todas sus máquinas estarán en riesgo.

Los lineamientos a continuación le ayudarán a crear una contraseña fuerte:

- *Cree una contraseña de por lo menos ocho caracteres* — Entre más larga su contraseña, mejor. Si utiliza contraseñas MD5, debe ser por lo menos de 15 caracteres. Con contraseñas DES, use la longitud máxima (ocho caracteres).
- *Mezcle letras mayúsculas y minúsculas* — Red Hat Enterprise Linux es sensible a mayúsculas y minúsculas, por lo tanto, mezcle las mayúsculas y minúsculas para fortalecer la contraseña.
- *Mezcle letras y números* — Si añade números a contraseñas, especialmente en el medio (no solo al comienzo o al final), puede mejorar la fortaleza de la contraseña.
- *Incluya caracteres no alfanuméricos* — Caracteres especiales tales como &, $, y > pueden mejorar ampliamente la fortaleza de la contraseña (no es posible si se utilizan contraseñas DES).
- *Elija una contraseña que usted recuerde* — La mejor contraseña del mundo no sirve de nada si usted no la recuerda; use acrónimos u otros dispositivos nemotécnicos para ayudar a memorizar las contraseñas.

Con todas estas reglas, puede parecer difícil crear una contraseña que cumpla todos los criterios de buenas contraseñas evitando al mismo tiempo las características de una mala. Afortunadamente, hay algunos pasos que puede seguir para generar una fácil de recordar, contraseña segura.

2.1.3.1.1. Metodología para la creación de una contraseña segura

Hay varios métodos que se pueden utilizar para crear contraseñas seguras. Uno de los métodos más conocidos tiene que ver con los acrónimos. Por ejemplo:

- Piense en una frase fácil de recordar, tal como esta en Inglés:

  "over the river and through the woods, to grandmother's house we go."

- Luego conviértala en un acrónimo (incluyendo la puntuación)

  otrattw,tghwg.

- Añada complejidad al sustituir por números y símbolos las letras en el acrónimo. Por ejemplo, substituya **7** para **t** y el símbolo de arroba (@) para **a**:

  o7r@77w,7ghwg.

- Añada más complejidad al poner en mayúsculas al menos una letra, tal como **H**.

  o7r@77w,7gHwg.

- *Por último, no utilice nunca la contraseña del ejemplo anterior para ningún sistema*.

Aunque la creación de contraseñas seguras es imperativa, manejarlas adecuadamente es también importante, especialmente para los administradores de sistemas dentro de grandes organizaciones. La siguiente sección describe buenas prácticas para crear y administrar contraseñas de usuarios dentro de una organización.

2.1.3.2. Cómo crear contraseñas de usuario dentro de una organización

Si una organización tiene un gran número de usuarios, los administradores de sistemas tienen dos opciones básicas disponibles para forzar el uso de buenas contraseñas. Pueden crear contraseñas para el usuario o dejar que los usuarios creen sus propias contraseñas, verificando que las contraseñas sean de una calidad aceptable.

La creación de contraseñas para los usuarios garantiza que las contraseñas sean buenas, pero se convierte en una tarea de enormes proporciones cuando la organización crece. También aumenta el riesgo de que los usuarios anoten las contraseñas.

Por estas razones, la mayoría de los administradores de sistemas prefieren dejar que los usuarios creen sus propias contraseñas, pero activamente verificar que las contraseñas sean buenas y, en algunos casos, obligarlos a cambiar sus contraseñas periódicamente a través de la caducidad de contraseñas.

2.1.3.2.1. Cómo forzar contraseñas fuertes

Para proteger la red de intrusos es una buena idea que los administradores de sistemas verifiquen si las contraseñas utilizadas dentro de la organización son fuertes. Cuando se les solicita a los usuarios crear o cambiar contraseñas, pueden utilizar la aplicación de línea de comandos **passwd**, la cual reconoce los *Módulos de autenticación conectables* (*PAM*) y por lo tanto verifica si la contraseña es demasiado corta o fácil de averiguar. Esta revisión se realiza mediante el módulo PAM de **pam_cracklib.so**. Puesto que PAM puede personalizarse, es posible añadir más revisores de integridad de contraseñas, tales como **pam_passwdqc** (disponible en *http://www.openwall.com/*

passwdqc/) o escribir un nuevo módulo. Para obtener una lista de los módulos PAM disponibles, consulte *http://www.kernel.org/pub/linux/libs/pam/modules.html*. Para mayor información sobre PAM, consulte *Managing Single Sign-On and Smart Cards*.

La revisión de contraseñas que se realiza en el momento de creación no descubre las contraseñas malas de forma tan efectiva como al ejecutar un programa de violación de contraseñas que compara contraseñas.

Muchos programas de contraseñas que están disponibles se pueden ejecutar bajo Red Hat Enterprise Linux, aunque ninguna se envía con el sistema operativo. A continuación una breve lista de algunos de los programas más conocidos para violar contraseñas:

- ***John The Ripper*** — Un programa rápido y flexible de violación de contraseñas. Permite el uso de múltiples listas de palabras y puede usar la fuerza bruta para violar las contraseñas. Está disponible en *http://www.openwall.com/john/*.
- ***Crack*** — Quizás el software de violación de contraseñas más conocido, **Crack** también es rápido, aunque no fácil de usar como **John The Ripper**. Puede encontrarlo en *http://www.crypticide.com/alecm/security/crack/c50-faq.html*.
- ***Slurpie*** — **Slurpie** es una aplicación similar a **John The Ripper** y **Crack**, pero está diseñada para ser ejecutada en varios equipos al mismo tiempo, creando un ataque de violación de contraseñas distribuido. Se encuentra junto con otro número de herramientas de evaluación de seguridad de ataques distribuidos en *http://www.ussrback.com/distributed.htm*.

Advertencia

Siempre obtenga autorización escrita antes de intentar violar contraseñas dentro de una organización.

2.1.3.2.2. Frases de paso

Las frases de paso y contraseñas son la piedra angular de la seguridad en la mayoría de los sistemas actuales. Infortunadamente, las técnicas tales como la biometría y la autenticación de dos factores aún no han convertido aún no se han establecido en muchos sistemas. Si se van a utilizar contraseñas para asegurar un sistema, entonces se debe considerar el uso de una frase de paso. Las frases de paso son más largas que las contraseñas y proporcionan una mejor protección de una contraseña, incluso cuando se implementan con caracteres no estándar, tales como números y símbolos.

2.1.3.2.3. Caducidad de las contraseñas

La caducidad de las contraseñas es una técnica utilizada por los administradores de sistemas para protegerse de las malas contraseñas dentro de una organización. La caducidad de la contraseña significa que después de un período determinado (generalmente 90 días), se pide al usuario crear una nueva contraseña. La teoría detrás de esto es que si un usuario se ve obligado a cambiar la contraseña periódicamente, una contraseña que ha sido descubierta solamente será útil para el intruso por un periodo limitado de tiempo. No obstante, la desventaja de la caducidad de las contraseñas, es que los usuarios tienden más a anotar sus contraseñas.

Hay dos programas principales utilizados para especificar la caducidad de las contraseñas bajo Red Hat Enterprise Linux: el comando **chage** o la aplicación gráfica **Administrador de usuario** (**`system-config-users`**).

La opción **-M** del comando **chage** especifica el número de días máximo que la contraseña es válida. Por ejemplo, para establecer una contraseña de usuario que expire en 90 días, use el siguiente comando:

```
chage -M 90 <username>
```

En el comando anterior, remplace *<username>* por el nombre del usuario. Para desactivar la expiración de la contraseña, se acostumbra a usar el valor de **99999** después de **-M** (esto equivale a un poco más de 273 años).

También puede usar el comando **chage** en modo interactivo para modificar la caducidad de varias contraseñas e información de cuentas. Use el siguiente comando para ingresar en modo interactivo:

```
chage <username>
```

El siguiente es un ejemplo de una sesión interactiva con este comando:

```
[root@myServer ~]# chage davido
Changing the aging information for davido
Enter the new value, or press ENTER for the default
Minimum Password Age [0]: 10
Maximum Password Age [99999]: 90
Last Password Change (YYYY-MM-DD) [2006-08-18]:
Password Expiration Warning [7]:
Password Inactive [-1]:
Account Expiration Date (YYYY-MM-DD) [1969-12-31]:
[root@myServer ~]#
```

Consulte la página de manual de chage para obtener mayor información en las opciones disponibles.

También puede usar la aplicación gráfica **Administrador de usuario** para crear políticas de caducidad de contraseñas, como sigue. Nota: necesitará privilegios de administrador para realizar este procedimiento.

1. Haga clic en el menú **Sistema** en el panel, señale **Administración** y luego haga clic en **Usuarios y grupos** para desplegar el administrador de usuario. De modo alternativo, escriba el comando **system-config-users** en el indicador de shell.

2. Haga clic en la pestaña **Usuarios** y seleccione el usuario requerido en la lista de usuarios.

3. Haga clic en **Propiedades** en la barra de herramientas para desplegar el cuadro de diálogo del usuario (o elija **Propiedades** en el menú **Archivo**).

4. Haga clic en **Información de contraseña**, y selecciones la cajilla de verificación para **Habilitar expiración de contraseña**.

5. Ingrese el valor requerido en el campo **Días antes del cambio requerido** y haga clic en **Aceptar**.

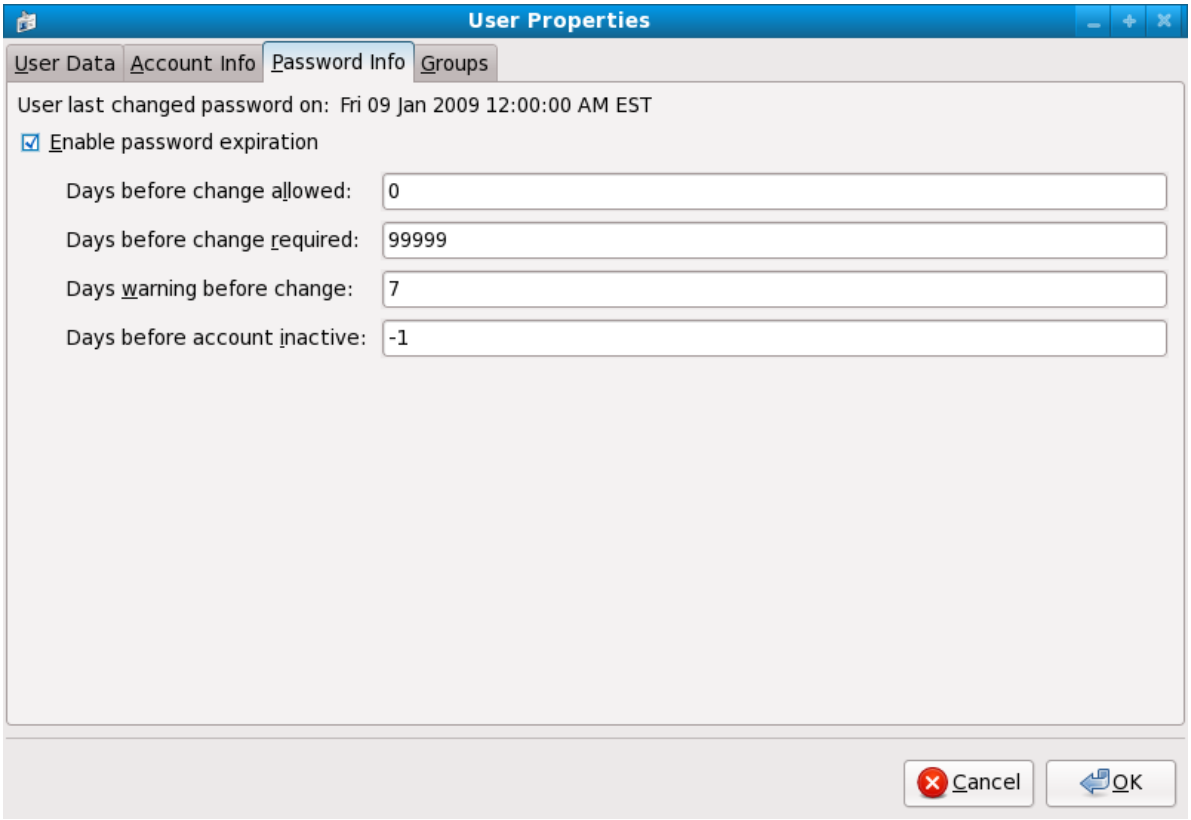


Figura 2.1. Cómo especificar opciones de caducidad de contraseñas

2.1.4. Controles administrativos

Al administrar un equipo del hogar, el usuario debe realizar algunas tareas como usurio de root o adquirir privilegios de root efectivos a través de un programa de *setuid*, tal como **sudo** o **su**. Un programa de setuid es el que funciona con el ID de usuario (*UID*) del propietaio del programa en lugar del usuario que opera el programa. Dichos programas se indican con una **s** en la sección de listado de formato largo, como en el siguiente ejemplo:

```
-rwsr-xr-x 1 root root 47324 May 1 08:09 /bin/su
```

Nota

La **s** puede estar en minúscula o mayúscula. Si aparece en mayúscula, significa que el bit de permiso subyacente aún no se ha establecido.

Sin embargo, para los administradores de sistemas de una organización, la elección debe hacerse en cuánto acceso administrativo deben tener los usuarios deben tener en su máquina. A través de un módulo PAM llamado **pam_console.so**, algunas actividades normalmente reservadas para el usuario root, tales como reinicio o el montaje de medios extraíbles, para el primer usuario que ingrese en la consola física (consulte la *Managing Single Sign-On and Smart Cards* para obtener más información acerca del módulo **pam_console.so**) Sin embargo, otras tareas administrativas del sistema, tales como alterar la configuración de red, configurar un nuevo ratón o montar dispositivos de red, no son posibles sin privilegios de administrador. Como resultado, los administradores de sistemas deben decidir cuánto acceso pueden tener los usuarios en la red.

2.1.4.1. Permitir acceso de root

Si los usuarios dentro de una organización son de confianza y tienen conocimientos de computación, entonces el darles acceso root no puede ser un problema. Permitir el acceso de root a los usuarios significa que las actividades de menor importancia, tales como añadir dispositivos o configurar interfaces de red, pueden ser manejadas por los usuarios individuales, dejando así a los administradores de sistemas libres para manejar la seguridad de la red y otros temas importantes.

Por otra parte, el dar acceso de root a usuarios individuales puede conducir a los siguientes problemas:

- *Error en la configuración de la máquina* — Los usuarios con acceso de root pueden desconfigurar las máquinas y necesitarán ayuda para resolver los problemas. Peor aún, podrían abrir agujeros de seguridad sin saberlo.
- *Ejecutar servicios inseguros* — Los usuarios con acceso de root pueden ejecutar servicios inseguros en sus máquinas, tales como FTP o Telnet, poniendo en riesgo los nombres de usuario y contraseñas. Estos servicios transmiten la información a través de la red en texto plano.
- *Ejecutar anexos de correo-e como root* — Aunque escasos, los virus de correo-e que afectan a Linux existen. No obstante, el único momento en que son una amenaza, es cuando son ejecutados por el usuario root.

2.1.4.2. Desactivación del acceso root

Si un administrador no se siente bien al permitir a los usuarios iniciar una sesión como root por estas u otras razones, la contraseña de root debe mantenerse en secreto, y el acceso en nivel de ejecución uno o en modo de monousuario debe desactivarse a través de la protección de contraseña del gestor de arranque (consulte la *Sección 2.1.2.2, "Contraseñas de gestor de arranque"* para obtener mayor información sobre este tema.)

Tabla 2.1, "Métodos para desactivar la cuenta de root" describe las formas en que un administrador puede garantizar que los nombres de usuario de root están desactivados:

Tabla 2.1. Métodos para desactivar la cuenta de root

Método	Descripción	Efectos	No afecta
Cambio de shell de root.	Edite el archivo **/etc/passwd** y cambie el shell de **/bin/bash** a **/sbin/nologin**.	Evita el acceso al shell de root y registra los intentos. Los siguientes programas no pueden acceden desde la cuenta de root: · **login** · **gdm** · **kdm** · **xdm** · **su** · **ssh** · **scp** · **sftp**	Los programas no requieren una shell, tal como clientes FTP, clientes de correo y muchos programas de setuid. Los siguientes programas *no* impiden el acceso a la cuenta de root: · **sudo** · Clientes FTP · Clientes de correo-e
Desactivación del acceso de root a través de un	Un archivo vacío **/etc/securetty** impide el ingreso de root a los dispositivos conectados al equipo.	Impide el acceso a la cuenta de root a través de la consola o la red. Los siguientes programas no tienen acceso a la cuenta root:	Did you mean: The following programs are *and* prevented from accessing the root account: \nType text or a website address or translate

Método	Descripción	Efectos	No afecta
dispositivo de consola (tty).		· **login** · **gdm** · **kdm** · **xdm** · Otros servicios de red que abren una tty	a document.\nCancel \nEnglish - detected to Spanish translation\nLos programas que no inician una sesión como root, pero que realizan tareas administrativas a través de setuid u otros mecanismos. Los siguientes programas *no* impiden el acceso a la cuenta de root: · **su** · **sudo** · **ssh** · **scp** · **sftp**
Desactivación de nombres de usuario de root SSH.	Edite el archivo **/etc/ssh/sshd_config** y establezca el parámetro **PermitRootLogin** a **no**.	Impide el acceso de root a través del conjunto de herramientas OpenSSH. A los siguientes programas se les impide acceder a la cuenta de root: · **ssh** · **scp** · **sftp**	Únicamente impide el acceso de root al paquete de herramientas OpenSSH.
Use PAM para limitar el acceso de root a servicios.	Edite el archivo para el servicio de destino en el directorio **/etc/pam.d/**. Asegúrese de que **pam_listfile.so** se requiera para la autenticación.[1]	Impide el acceso de root a los servicios de red que PAM reconoce. Los servicios a continuación no tienen acceso a la cuenta de root: · Clientes FTP · Clientes de correo-e · **login** · **gdm** · **kdm** · **xdm** · **ssh** · **scp** · **sftp** · Todos los servicios que PAM reconoce	Los programas y servicios que PAM no reconoce.

[1] Consulte la *Sección 2.1.4.2.4, "Desactivación de root mediante PAM"* for details.

2.1.4.2.1. Desactivación del shell de root

Para evitar que los usuarios ingresen directamente como root, el administrador de sistemas puede establecer el shell de la cuenta de root a **/sbin/nologin** en el archivo **/etc/passwd**. Así impide el acceso a la cuenta de root a través de los comandos que requieren un shell, tal como los comandos **su** y **ssh**.

Importante

Los programas que no requieren acceso al shell, tales como clientes de correo-e o el comando **sudo**, aún pueden acceder a la cuenta de root.

2.1.4.2.2. Desactivación de los nombres de usuario de root

Para limitar aún más el acceso a la cuenta de root, los administradores pueden inhabilitar las conexiones de root en la consola mediante la edición del archivo **/etc/securetty**. Este archivo muestra todos los dispositivos a los que el usuario de root puede conectarse. Si el archivo no existe en absoluto, el usuario puede conectarse a través de cualquier dispositivo de comunicación en el sistema, ya sea a través de la consola o una interfaz de red cruda. Esto es peligroso, ya que un usuario puede conectarse a su máquina como root a través de Telnet, la cual transmite por la red a contraseña en texto plano. De forma predeterminada, el archivo **/etc/securetty** de Red Hat Enterprise Linux solamente permite que el usuario de root inicie sesión en la consola conectada físicamente a la máquina. Para impedir que root ingrese, retire el contenido de este archivo mediante el siguiente comando:

```
echo > /etc/securetty
```

Advertencia

Un archivo en blanco **/etc/securetty** *no* impide que el usuario de root ingrese de forma remota mediante el paquete de herramientas de OpenSSH porque la consola no se abre sino después de la autenticación.

2.1.4.2.3. Desactivación de los nombres de usuario de root SSH

Los registros de root a través del protocolo SSH se desactivan por defecto en Red Hat Enterprise Linux 6; si esta opción ha sido habilitada, puede inhabilitarse de nuevo al editar el archivo de configuración del demonio (**/etc/ssh/sshd_config**). Cambie la línea que se lee:

```
PermitRootLogin yes
```

Para que se lea:

```
PermitRootLogin no
```

Para que estos cambios se efectúen, deberá reiniciar el demonio. Esto puede hacerse con el siguiente comando:

```
kill -HUP `cat /var/run/sshd.pid`
```

2.1.4.2.4. Desactivación de root mediante PAM

PAM, a través del módulo **/lib/security/pam_listfile.so**, permite gran flexibilidad en la negación de cuentas específicas. El administrador puede usar este módulo para hacer referencia

a una lista de usuarios que no tienen permiso de ingreso. A continuación verá un ejemplo de cómo se utiliza el módulo para el servidor FTP de **vsftpd** en el archivo de configuración **/etc/pam.d/vsftpd** (el caracter **** al final de la primera línea en el siguiente ejemplo *no* es necesario si la directiva está un una línea):

```
auth required /lib/security/pam_listfile.so item=user \
sense=deny file=/etc/vsftpd.ftpusers onerr=succeed
```

Así le pide a PAM que consulte el archivo **/etc/vsftpd.ftpusers** y niegue el acceso al servicio para cualquier usuario en la lista. El administrador puede cambiar el nombre de este archivo y mantener listas independientes para cada servicio o usar una lista central para negar acceso a múltiples servicios.

Si el administrador desea negar acceso a múltiples servicios, se puede añadir una línea similar a los archivos de configuración PAM, tales como **/etc/pam.d/pop** y **/etc/pam.d/imap** para clientes de correo o **/etc/pam.d/ssh** para clientes SSH.

Para obtener mayor información sobre PAM, consulte *Managing Single Sign-On and Smart Cards*.

2.1.4.3. Limitación del acceso de root

En vez de negar completamente el acceso al usuario de root, el administrador puede desear permitir el acceso a través de programas de setuid, tales como **su** o **sudo**.

2.1.4.3.1. El comando su

Cuando un usuario ejecuta el comando **su** se le solicitará la contraseña de root y después de la autenticación, se le dará un indicador de shell de root.

Cuando el usuario ingresa a través del comando **su**, el usuario *se convierte* en el usuario de root y tiene acceso administrativo total en el sistema[3]. Además, una vez que el usuario se convierta en root, le será posible utilizar el comando **su** para cambiar a cualquier otro usuario en el sistema sin que se le solicite una contraseña.

Ya que este programa es tan poderoso, los administradores dentro de la organización pueden limitar el acceso al comando.

Una de las formas más sencillas de hacerlo es añadir usuarios al grupo administrativo especial llamado *wheel*. Para hacerlo, escriba el siguiente comando como root:

```
usermod -G wheel <username>
```

En el comando anterior, remplace <*username*> por el nombre de usuario que desea añadir al grupo **wheel**.

También puede usar el **Administrador de usuarios** para modificar membresías de grupos, como se presenta a continuación. Nota: necesita privilegios administrativos para realizar este procedimiento.

1. Haga clic en el menú **Sistema** en el panel, señale **Administración** y luego haga clic en **Usuarios y grupos** para desplegar el administrador de usuario. De modo alternativo, escriba el comando **system-config-users** en el indicador de shell.

2. Haga clic en la pestaña **Usuarios** y seleccione el usuario requerido en la lista de usuarios.

[3] Este acceso está aún sujeto a restricciones impuestas por SELinux, si SELinux está habilitado.

3. Haga clic en **Propiedades** en la barra de herramientas para desplegar el cuadro de diálogo del usuario (o elija **Propiedades** en el menú **Archivo**).

4. Haga clic en la pestaña **Groups**, seleccione la cajilla de verificación para el grupo wheel y luego haga clic en el botón **Aceptar**. Consulte la *Figura 2.2, “Adición de usuarios al grupo "wheel".”*.

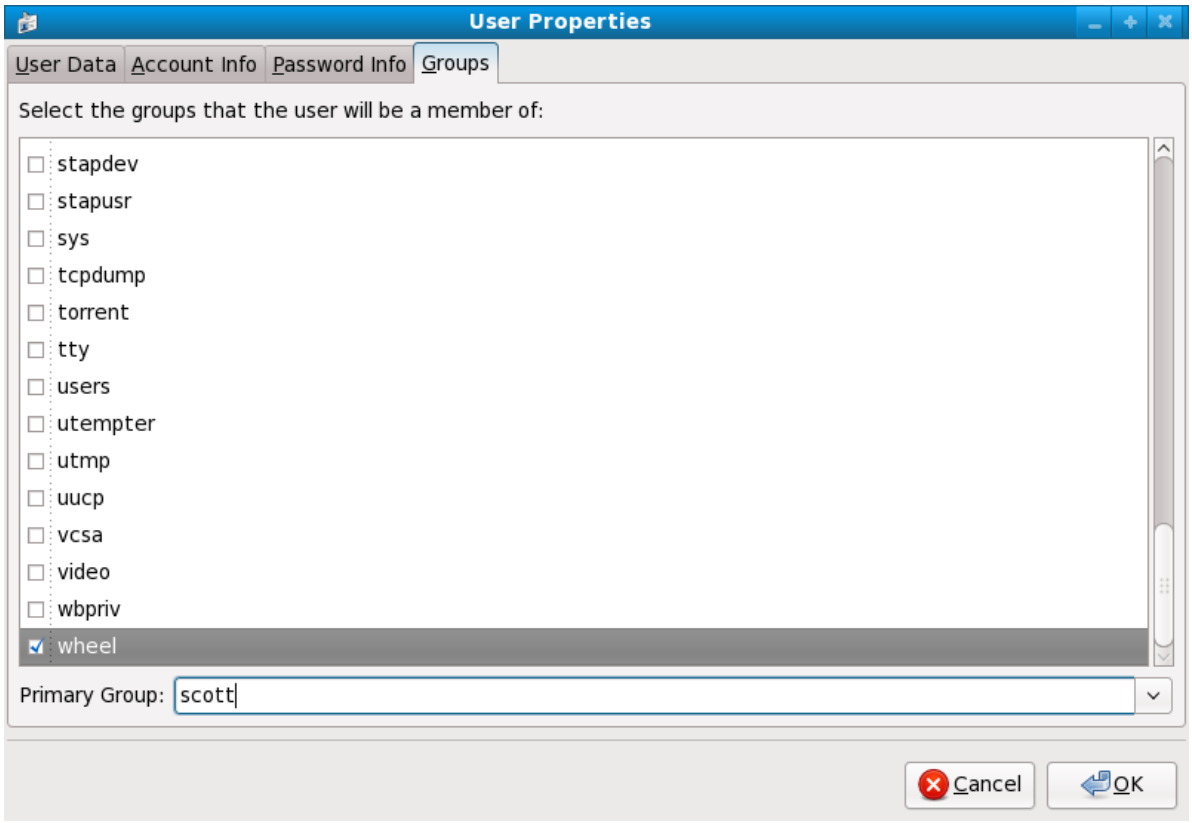


Figura 2.2. Adición de usuarios al grupo "wheel".

Abra el archivo de configuración PAM para **su** (**/etc/pam.d/su**) en un editor de texto y retire el comentario # de la siguiente línea:

```
auth  required /lib/security/$ISA/pam_wheel.so use_uid
```

Este cambio significa que únicamente los miembros del grupo administrativo **wheel** pueden usar este programa.

Nota

El usuario de root es parte del grupo **wheel** de forma predeterminada.

2.1.4.3.2. El comando sudo

El comando **sudo** ofrece otro método para proporcionar acceso administrativo a los usuarios. Cuando los usuarios fiables preceden un comando administrativo con **sudo**, se les solicitará su *propia* contraseña. Luego, cuando el usuario ha sido autenticado y se supone que el comando es permitido, el comando administrativo se ejecutará como si el usuario fuera root.

El formato básico del comando **sudo** es el siguiente:

```
sudo <command>
```

En el ejemplo anterior, <*command*> se remplazaría por el comando reservado generalmente para el usuario root, tal como **mount**.

Importante

Los usuarios del comando **sudo** deben tener extremo cuidado al salir antes de abandonar las máquinas ya que los usuarios de sudo pueden usar el comando otra vez en un lapso de 5 minutos sin que se les pregunte la contraseña. Esta configuración puede modificarse en el archivo de configuración, **/etc/sudoers**.

El comando **sudo** permite un alto grado de flexibilidad. Por ejemplo, solamente los usuarios listados en el archivo de configuración **/etc/sudoers** tienen permiso para usar el comando **sudo** y el comando se ejecuta en el shell del *usuario*, no en el shell de root. Esto significa que el shell de root puede estar completamente desactivado, como se muestra en la *Sección 2.1.4.2.1, "Desactivación del shell de root"*.

El comando **sudo** también proporciona un rastro de auditoría integral. Cada autenticación correcta se registra en el archivo **/var/log/messages** y el comando expedido junto con el nombre de usuario de quien lo expide se registran en el archivo **/var/log/secure**.

Otra ventaja del comando **sudo** es que el administrador puede permitir a diferentes usuarios el acceso a comandos específicos con base en sus necesidades.

Los administradores que desean modificar el archivo de configuración **sudo**, deben usar el comando **visudo**.

Para otorgar privilegios administrativos totales, escriba **visudo** y añada una línea similar a la siguiente en la sección de especificación de privilegios del usuario.

```
juan ALL=(ALL) ALL
```

Este ejemplo establece que el usuario, **juan**, puede usar **sudo** desde cualquier host y ejecutar cualquier comando.

El ejemplo a continuación ilustra la forma como se configura **sudo**:

```
%users localhost=/sbin/shutdown -h now
```

Este ejemplo establece que cualquier usuario puede emitir el comando **/sbin/shutdown -h now** siempre y cuando se ejecute desde la consola.

La página de manual para **sudoers** tiene un listado detallado de las opciones para este archivo.

2.1.5. Servicios de red disponibles

Aunque el acceso de usuario a controles administrativos es un aspecto importante para administradores de sistemas dentro de una organización, la monitorización de los servicios de red que están activos es de gran importancia para cualquier persona que opere un sistema operativo.

Muchos servicios bajo Red Hat Enterprise Linux 6 se comportan como servidores de red. Si un servicio de red se ejecuta en una máquina, entonces la aplicación de servidor (llamado *demonio*), escucha conexiones en uno o más puertos. Cada uno de estos servidores debe ser tratado como una avenida potencial de ataque.

2.1.5.1. Riesgos para los servicios

Los servicios de red pueden presentar muchos riesgos para los sistemas de Linux. A continuación, un lista de algunos de los principales problemas:

- *Ataques de denegación de servicio (DoS)* Al inundar un servicio con solicitudes, un ataque de denegación de servicio puede volver inservible a un sistema, ya que trata de registrar y responder a cada solicitud.
- *Ataque de denegación de servicio distribuido (DDoS)* — Un tipo de ataque DoS que utiliza varias máquinas que han perdido su caracter confidencial (a menudo enumerándolas en miles o más) para dirigir un ataque coordinado en un servicio, inundarlo con solicitudes y convertirlo en inservible.
- *Ataques de vulnerabilidad de Scripts* — Si un servidor utiliza scripts para ejecutar acciones relacionadas con el servidor, como comúnmente lo hacen los servidores de Web, un cracker puede atacar incorrectamente los scripts escritos. Estos ataques de vulnerabilidades de script pueden conducir a una condición de desbordamiento de buffer o permitir que el agresor altere los archivos en el sistema.
- *Ataques de desbordamientos de buffer* — Los servicios que se conectan a puertos enumerados del 0 al 1023 deben ser ejecutados como usuario administrativo. Si la aplicación sufre un desbordamiento de buffer, el atacante podría obtener acceso al sistema como el usuario al ejecutar el demonio. Puesto que los desbordamientos de buffer existen, los agresores utilizan herramientas automatizadas para identificar vulnerabilidades en los sistemas, y una vez que han obtenido acceso, utilizarán rootkits para mantener el acceso al sistema.\n

Nota

La amenaza de un desbordamiento de buffer es mitigada en Red Hat Enterprise Linux por *ExecShield*, una segmentación de memoria ejecutable y tecnología de protección soportada por kernels de procesadores (uni y multi) compatibles - x86. ExecShield reduce el riesgo de desbordamiento de buffer mediante la separación de la memoria virtual en segmentos ejecutables y no ejecutables. Cualquier código de programa que intenta ejecutarse en el segmento ejecutable (como el código malicioso inyectado desde un ataque de desbordamiento de buffer) provoca un fallo de segmentación y termina.\n

Execshield también incluye soporte para tecnología *No eXecute* (NX) en plataformas AMD64 y tecnología *eXecute Disable* (XD) en Itanium y 64 sistemas de Intel®. Estas tecnologías junto con ExecShield evitan que código malicioso se ejecute en la parte ejecutable de la memoria virtual con una granularidad de 4KB de código ejecutable, reduciendo el riesgo de un ataque de vulnerabilidades de desbordamiento de buffer invisible.

Importante

Para limitar la exposición a ataques en la red, todos los servicios que no están en uso son apagados.

2.1.5.2. Identificación y configuración de servicios

Para mejorar la seguridad, la mayoría de los servicios instalados con Red Hat Enterprise Linux están apagados de forma predeterminada. Hay, sin embargo, algunas excepciones:

- **cupsd** — El servidor de impresión predeterminado para Red Hat Enterprise Linux.
- **lpd** — Un servidor de impresión alternativo.
- **xinetd** — Un súper servidor que controla conexiones para un rango de servidores subordinados, tales como **gssftp** y **telnet**.
- **sendmail** — El *Mail Transport Agent* (MTA) de Sendmail está habilitado de forma predeterminada, pero no solamente escucha conexiones desde el host local.
- **sshd** — El servidor de OpenSSH, el cual es un remplazo seguro para Telnet.

Para determinar si deja estos servicios en ejecución, es mejor usar el sentido común y errar por precaución. Por ejemplo, si la impresora no está disponible, no deje a **cupsd** ejecutándose. Lo mismo se cierto para **portmap**. Si usted no monta volúmenes NFSv3 o usa NIS (el servicio **ypbind**), entonces **portmap** debe estar desactivado.

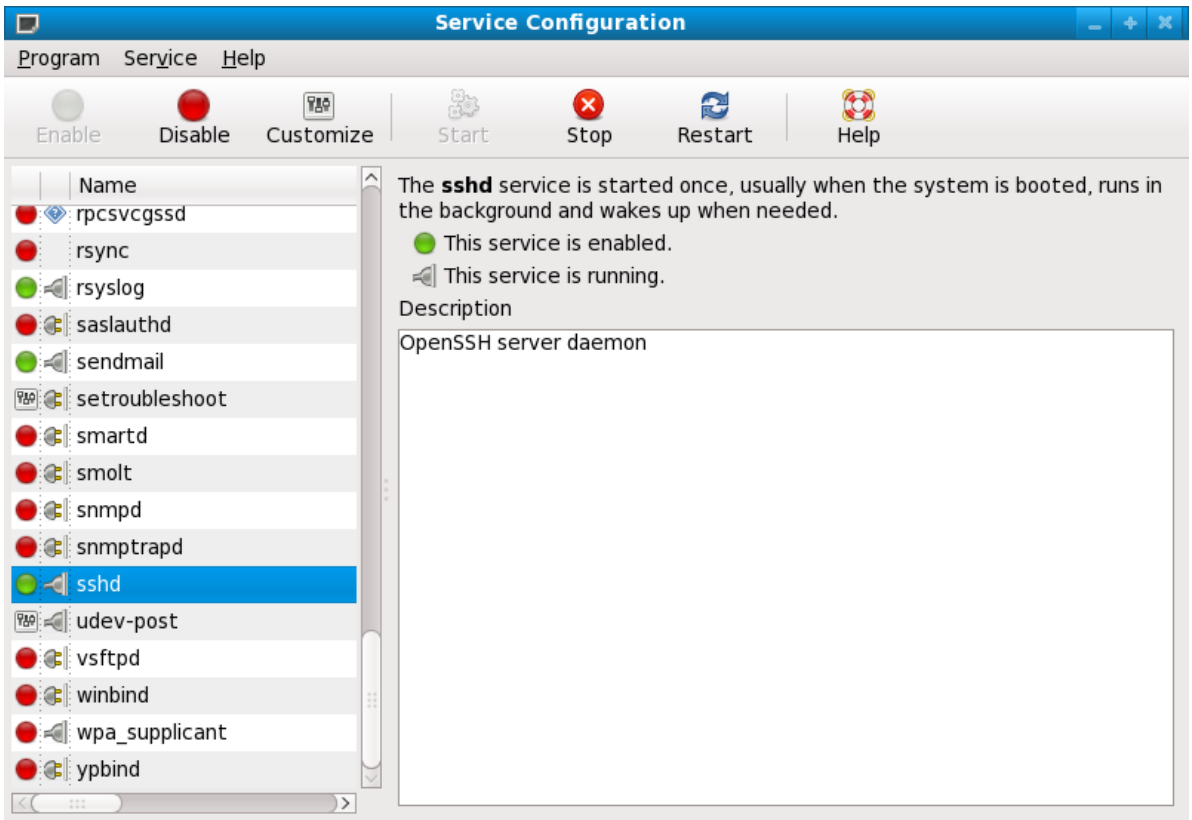


Figura 2.3. **Services Configuration Tool**

Si no está seguro del propósito de un determinado servicio, la **herramienta de configuración de servicios** tiene un campo de descripción, ilustrado en la *Figura 2.3, "**Services Configuration Tool**"*, la cual proporciona información adicional.

Revisar cuáles servicios de red están disponibles para iniciar en tiempo de arranque es tan solo una parte. También debe revisar los puertos que están abiertos y escuchando. Consulte la *Sección 2.2.8, "Cómo verificar los puertos que están escuchando"* para obtener mayor información.

2.1.5.3. Servicios inseguros

En potencia, cualquier servicio de red es inseguro. Esta es la razón por la cual es importante apagar los servicios que no se estén utilizando. Las vulnerabilidades para servicios se revelan y se corrigen, de forma rutinaria, lo cual hace importante actualizar regularmente los paquetes asociados con cualquier servicio de redes. Consulte la *Sección 1.5, "Actualizaciones de seguridad"* para obtener mayor información.

Algunos protocolos de red son más inseguros que otros. Entre ellos se incluyen los servicos que:

- *Transmitir por la red nombres de usuarios y contraseñas sin cifrar* — Muchos protocolos anteriores, tales como Telnet y FTP, no cifran la sesión de autenticación y deben evitarse cuando sea posible.
- *Transmitir por la red datos confidenciales sin cifrar* — Muchos protocolos transmiten datos sin cifrar por la red. Estos protocolos incluyen Telnet, FTP, HTTP, y SMTP. Muchos sistemas de archivos de red, tales como NFS y SMB, también transmiten información sin cifrar por la red. Es responsabilidad del usuario cuando use estos protocolos, limitar el tipo de datos que se transmite.

  Servicios de volcado de memoria, tales como **`netdump`**, transmiten el contenido de memoria por la red sin cifrar. Los volcados de memoria pueden contener contraseñas o incluso peor, entradas de base de datos y otra información confidencial.

  Otros servicios como **`finger`** y **`rwhod`** revelan información sobre usuarios del sistema.

Entre los ejemplos de servicios inseguros heredados se incluyen **`rlogin`**, **`rsh`**, **`telnet`**, y **`vsftpd`**.

Todos los programas remotos y de shell (**`rlogin`**, **`rsh`**, and **`telnet`**) deben evitarse en favor de SSH. Consulte la *Sección 2.1.7, "Herramientas de comunicación de Seguridad Mejorada "* para obtener mayor información sobre **`sshd`**.

FTP no es de por sí tan peligrosa para la seguridad del sistema como los shell remotos, pero los servidores FTP deben ser configurados cuidadosamente y monitorizados para evitar problemas. Consulte la *Sección 2.2.6, "Cómo proteger FTP"* para obtener mayor información sobre cómo proteger servidores FTP.

Entre los servicios que deben ser cuidadosamente implementados y detrás de un cortafuegos, están:

- **`finger`**
- **`authd`** (conocido como **`identd`** en lanzamientos anteriores de Red Hat Enterprise Linux.)
- **`netdump`**
- **`netdump-server`**
- **`nfs`**
- **`rwhod`**
- **`sendmail`**

- **smb** (Samba)
- **yppasswdd**
- **ypserv**
- **ypxfrd**

Puede encontrar más información sobre protección de servicios de red en *Sección 2.2, "Seguridad del servidor"*.

La siguiente sección describe las herramientas disponibles para configurar un cortafuegos sencillos.

2.1.6. Cortafuegos personales

Después de que los servicios *necesarios* de red son configurados, es importante implementar un cortafuegos.

Importante

Debe configurar los servicios necesarios e implementar un cortafuegos *antes* de conectarse a la Internet o a otra red en la que usted no confía.

Los cortafuegos impiden que los paquetes de red accedan a la interfaz de red del sistema. Si se hace una solicitud a un puerto que está bloqueado por un cortafuegos, la petición es ignorada. Si un servicio está escuchando en uno de estos puertos bloqueados, no recibe los paquetes y se desactiva. Por esta razón, se debe tener cuidado al configurar un cortafuegos para bloquear el acceso a los puertos que no están en uso y no bloquear el acceso a los puertos usados por servicios configurados.

Para la mayoría de usuarios, la mejor herramienta para configurar un cortafuegos sencillo es la herramienta de configuración que se distribuye con Red Hat Enterprise Linux: la **Firewall Configuration Tool** (**system-config-firewall**). Esta herramiena crea amplias reglas **iptables** para un cortafuegos de propósitos generales mediante una interfaz de panel de control.

Consulte la *Sección 2.5.2, "Configuración básica de cortafuegos"* para obtener mayor información sobre el uso de esta aplicación y sus opciones disponibles.

Para usuarios avanzados y administradores, configurar manualmente un cortafuegos con **iptables** probablemente es una mejor opción. Consulte la *Sección 2.5, "Cortafuegos"* para obtener más información. Consulte la *Sección 2.6, "IPTables"* para obtener una guía integral de **iptables**.

2.1.7. Herramientas de comunicación de Seguridad Mejorada

A medida que el tamaño y la popularidad de la Internet crece, también crece la amenaza de la interceptación de las comunicaciones. Con los años, se han desarrollado herramientas para cifrar las comunicaciones, ya que se transfieren a través de la red.

Red Hat Enterprise Linux 6 se distribuye con dos herramientas básicas que usan un alto nivel de algoritmos cifrados basados en criptografía de llave pública para proteger información cuando viaja sobre la red.

- *OpenSSH* — Una implementación libre de protocolo SSH para cifrar la comunicación de redes.

- *Gnu Privacy Guard (GPG)* — Una aplicación de cifrado PGP (Pretty Good Privacy) para cifrado de datos.

OpenSSH es una forma más segura para acceder a una máquina remota y remplaza los anteriores, los servicios no cifrados como **telnet** y **rsh**. OpenSSH incluye un servicio de red llamado **sshd** y tres aplicaciones de cliente de línea de comandos:

- **ssh** — Un cliente de consola de acceso remoto seguro.
- **scp** — Un comando de copia remoto seguro.
- **sftp** — Un seudo cliente ftp que permite sesiones de transferencia de archivos.

Consulte la *Sección 3.6, "Shell segura"* para obtener mayor información sobre OpenSSH.

Importante

Aunque el servicio **sshd** es esencialmente seguro, el servicio *debe* mantenerse actualizado para prevenir amenazas de seguridad. Consulte la *Sección 1.5, "Actualizaciones de seguridad"* para obtener mayor información.

GPG es una forma de comunicación privada por correo-e. Puede servir tanto para datos confidenciales de correo-e como pare redes públicas y para proteger datos confidenciales en discos duros.

2.2. Seguridad del servidor

Cuando se utiliza un sistema como un servidor en una red pública, se convierte en objetivo para los agresores. Por lo tanto, es de suma importancia para el administrador de sistemas fortalecer el sistema y bloquear los servicios.

Antes de profundizar en aspectos específicos, revise los siguientes consejos generales para mejorar la seguridad del servidor:

- Mantenga todos los servicios actualizados, para proteger contra últimas amenazas.
- Use protocolos seguros cuando sea posible.
- Sirva solamente un tipo de servicio por máquina cuando sea posible.
- Monitorice cuidadosamente todos los servicios para actividades sospechosas.

2.2.1. Cómo proteger servicios con envolturas TCP y xinetd

Las *envolturas TCP* proporcionan control de acceso a una variedad de servicios. La mayoría de servicios de red modernos, tales como SSH, Telnet y FTP, hacen uso de envolturas TCP, las cuales montan guardia entre las peticiones entrantes y el servicio solicitado.

Los beneficios que ofrecen las envolturas TCP se mejoran cuando se utilizan junto con **xinetd**, un súper servidor que proporciona acceso adicional, ingreso, enlace, redirección y control de uso de recursos.

Nota

Es una buena idea usar reglas de cortafuegos de iptables junto con envolturas TCP y **xinetd** para crear redundancia dentro de controles de acceso de servicios. Consulte, la *Sección 2.5, "Cortafuegos"* para obtener información sobre implementación de cortafuegos con comandos de iptables.

Las siguientes sub-secciones suponen un conocimiento básico de cada tema y enfoque sobre las opciones de seguridad específicas.

2.2.1.1. Cómo mejorar la seguridad con envolturas TCP

Las envolturas TCP pueden hacer mucho más que negar el acceso a los servicios. Esta sección ilustra cómo se pueden utilizar para enviar pancartas de conexión, advertir de ataques de algún determinado host y mejorar la funcionalidad de registro. Consulte la página de manual **hosts_options** para obtener mayor información sobre la funcionalidad de envolturas TCP y el lenguaje de control. También puede consultar la página de manual **xinetd.conf** disponible en *http://linux.die.net/man/5/xinetd.conf* para obtener información sobre indicadores disponibles que actúan como opciones que se pueden aplicar a un servicio.\n

2.2.1.1.1. Envolturas TCP y pancartas de conexión

Desplegar una pancarta apropiada cuando los usuarios se conectan al servicio es una buena forma de hacer saber a los atacantes de que hay un administrador de sistemas en alerta. Puede también controlar la información sobre el sistema que se presenta a los usuarios. Para implementar una pancarta de envolturas TCP para un servicio, use la opción **banner**.\n\t\n

Este ejemplo implementa una pancarta para **vsftpd**. Para comenzar, cree un archivo de pancartas. El archivo puede estar en cualquier parte del sistema, pero debe tener el mismo nombre del demonio. En este ejemplo, el archivo se denomina **/etc/banners/vsftpd** y contiene la siguiente línea:

```
220-Hello, %c
220-All activity on ftp.example.com is logged.
220-Inappropriate use will result in your access privileges being removed.
```

El símbolo **%c** ofrece una variedad de información del cliente, tal como el nombre de usuario y el nombre de host o el nombre de usuario y la dirección IP para hacer la conexión mucho más intimidante.

Para desplegar esta pancarta para conexiones entrantes, añada la siguiente línea al archivo **/etc/hosts.allow**:

```
vsftpd : ALL : banners /etc/banners/
```

2.2.1.1.2. Envolturas TCP y Advertencias de ataques

Si se detecta un determinado host o red atacando al servidor, pueden utilizarse las envolturas TCP para advertir al administrador de ataques posteriores de ese host o red mediante la directiva **spawn**.

En este ejemplo, se asume que el cracker de la red 206.182.68.0/24 ha sido detectado tratando de atacar al servidor. Añada esta línea en el archivo para negar cualquier intento de conexión desde la red y para registrar los intentos en un archivo especial.

```
ALL : 206.182.68.0 : spawn /bin/echo `date` %c %d >> /var/log/intruder_alert
```

El símbolo **%d** provee el nombre del servicio al que el atacante está tratando de acceder.

Para permitir la conexión y el ingreso, sitúe la directiva de **spawn** en el archivo **/etc/hosts.allow**.

Nota

Puesto que la directiva **spawn** ejecuta cualquier comando de shell, es una buena idea crear un script especial para notificar al administrador o ejecutar una cadena de comandos en caso de que un cliente particular intente conectarse al servidor.

2.2.1.1.3. Envolturas TCP e ingreso mejorado

Si algunos tipos de conexiones preocupan más que otras, el nivel de registro se puede elevar para ese servicio mediante la opción **severity**.

Para este ejemplo, se supone que la persona que está intentando conectarse al puerto 23 (el puerto de Telnet) en un servidor FTP es un cracker. Para resaltar esto, sitúe el indicador **emerg** en los archivos de registro en lugar del indicador predeterminado, **info**, y niegue la conexión.

Para hacer esto, escriba la siguiente línea en **/etc/hosts.deny**:

```
in.telnetd : ALL : severity emerg
```

De esta manera, se utiliza la herramienta predeterminada **authpriv**, pero se eleva la prioridad del valor predeterminado de **info** a **emerg**, la cual envía directamente mensajes de registro a la consola.

2.2.1.2. Cómo mejorar la seguridad con xinetd

Esta sección se centra en el uso de **xinetd** para establecer el servicio de trampa y al usarlo controlar los niveles de recursos disponibles para un determinado servicio de **xinetd**. Si establece límites de recurso para servicios puede ayudar a frustrar ataques de *Denegación del servicio* (DoS). Consulte las páginas de manual para **xinetd** y **xinetd.conf** para obtener una lista de las opciones disponibles.

2.2.1.2.1. Cómo establecer una trampa

Un rasgo importante de **xinetd** es la capacidad de añadir hosts a una lista global de **no_access**. A los hosts en esta lista se les niegan las conexiones posteriores a servicios administrados por **xinetd** por un periodo específico de tiempo o hasta que se reinicie **xinetd**. Puede hacerlo mediante el atributo **SENSOR**. Esta es una forma fácil de bloquear los hosts que intentan escanear los puertos en el servidor.

El primer paso para configurar el **SENSOR** es seleccionar el servicio que no piensa utilizar. Para este ejemplo, se utiliza Telnet.

Edite el archivo **/etc/xinetd.d/telnet** y cambie la línea **flags** para que se lea:

```
flags           = SENSOR
```

Añada la siguiente línea:

```
deny_time       = 30
```

Así, niega otros intentos de conexión al puerto por ese host por 30 minutos. Otros valores aceptables para el atributo **deny_time** son FOREVER, el cual mantiene el bloqueo hasta que se inicie **xinetd** y NEVER, el cual permite la conexión y la conecta.

Por último, la línea se lee:

```
disable         = no
```

Así se habilita la trampa misma.

El uso de **SENSOR** es una buena forma de detectar y detener las conexiones de hosts no deseables, hay dos inconvenientes:

- No funciona con escáner invisible
- Un agresor que sepa que **SENSOR** se está ejecutando, puede montar un ataque de Denegación del servicio contra determinados hosts falsificando sus direcciones IP y conectándose al puerto prohibido.

2.2.1.2.2. Cómo controlar recursos de servidor

Otra funcionalidad importante de **xinetd** es su habilidad de establecer límites de recursos para servicios bajo su control.

Lo hace mediante las siguientes directivas:

- **cps = <number_of_connections> <wait_period>** — Limita la tasa de conexiones entrantes. Esta directiva toma dos argumentos:
  - **<number_of_connections>** — El número de conexiones a manejar por segundo. Si la tasa de conexiones entrantes es mayor que ésta, el servicio se inhabilitará temporalmente. El valor predeterminado es cincuenta (50).
  - **<wait_period>** — El número de segundos de espera antes de volver a habilitar el servicio después de que ha sido desactivado. El intervalo predeterminado es de diez (10) segundos.
- **instances = <number_of_connections>** — Especifica el número total de conexiones permitidas para el servicio. Esta directiva acepta ya sea un valor entero o un valor **UNLIMITED** (ILIMITADO).
- **per_source = <number_of_connections>** — Especifica el número de conexiones permitidas a un servicio por cada host. Esta directiva acepta un valor entero o un valor **UNLIMITED**.
- **rlimit_as = <number[K|M]>** — Especifica la cantidad de espacio de dirección de memoria que el servicio puede ocupar en kilobytes o megabytes. Esta directiva acepta un valor entero o un valor **UNLIMITED**.
- **rlimit_cpu = <number_of_seconds>** — Especifica la cantidad de tiempo en segundos que un servicio puede ocupar la CPU. Esta directiva acepta el valor entero y el valor **UNLIMITED**.

Con el uso de estas directivas se puede ayudar a evitar que cualquier servicio **xinetd** sobrecargue el sistema y que resulte en una negación del servicio.

2.2.2. Cómo proteger a Portmap

El servicio **portmap** es un demonio de asignación de puerto dinámico para servicios de RPC tales como NIS y NFS. Tiene mecanismos de autenticación débiles y la capacidad de asignar un amplio rango de puertos para los servicios que controla. Por estos motivos es difícil proteger dicho servicio.

Nota

La protección de **portmap** solamente afecta las implementaciones NFSv2 y NFSv3, puesto que NFSv4 ya no lo requiere. Si piensa implementar un servidor NFSv2 o NFSv3, entonces necesitará **portmap** y la siguiente sección se aplicará.

Si ejecuta servicios de RPC, siga las siguientes reglas básicas.

2.2.2.1. Proteja a portmap con envolturas TCP

Es importante usar envolturas TCP para las limitar a las redes o hosts que tienen acceso al servicio **portmap**, ya que éste no tiene una forma de autenticación incorporada.

Además, use *únicamente* direcciones IP para limitar el acceso al servicio. Evite el uso de nombres de hosts, puesto que pueden ser falsificados al envenenar el DNS y otros métodos.

2.2.2.2. Proteja a portmap con iptables

Para restringir más el acceso al servicio **portmap**, se recomienda añadir reglas de iptables al servidor y restringir el acceso a las redes específicias.

A continuación aparecen dos ejemplos de comandos de iptables. El primero acepta conexiones TCP al puerto 111 (usado por el servicio **portmap**) desde la red 192.168.0.0/24. El segundo permite conexiones TCP al mismo puerto desde el host local. Esto es necesario para el servicio **sgi_fam** usado por **Nautilus**. Los demás paquetes se descartan.

```
iptables -A INPUT -p tcp ! -s 192.168.0.0/24 --dport 111 -j DROP
iptables -A INPUT -p tcp -s 127.0.0.1  --dport 111 -j ACCEPT
```

Para limitar el tráfico UDP, use el siguiente comando.

```
iptables -A INPUT -p udp ! -s 192.168.0.0/24  --dport 111 -j DROP
```

Nota

Consulte la *Sección 2.5, “Cortafuegos”* para obtener información sobre cómo implementar cortafuegos con comandos de iptables.

2.2.3. Cómo proteger a NIS

El *Servicio de información de red* (NIS) es un servicio RPC, llamado **ypserv**, el cual se utiliza junto con **portmap** y otros servicios relacionados para distribuir mapas de nombres de usuarios, contraseñas y otra información confidencial para cualquier equipo que diga que está dentro de su dominio.

Un servidor NIS está comprometido con varias aplicaciones. Entre ellas las siguientes:

- **/usr/sbin/rpc.yppasswdd** — Conocido también como el servicio **yppasswdd**, este demonio permite a los usuarios cambiar sus contraseñas de NIS.
- **/usr/sbin/rpc.ypxfrd** — Conocido también como el servicio **ypxfrd**, es el demonio responsable de las transferencias de mapas de NIS en la red.
- **/usr/sbin/yppush** — Esta aplicación propaga las bases de datos cambiadas de NIS a múltiples servidores de NIS.
- **/usr/sbin/ypserv** — Este es el demonio del servidor de NIS.

NIS es algún tanto insegura, según los estándares actuales. No tiene mecanismos de autenticación de host y transmite toda la información sin cifrar por la red, incluyendo los hash de contraseñas. Como resultado, se debe tener extremo cuidado al configurar una red que utiliza NIS. Esto es aún más complicado por el hecho de que la configuración predeterminada de NIS es de por sí insegura.

Si desea implementar un servidor de NIS, se recomienda primero proteger el servicio **portmap** como se describe en la *Sección 2.2.2, “Cómo proteger a Portmap”*, luego aborde los siguientes problemas, tal como el planeamiento de redes.

2.2.3.1. Planee cuidadosamente la red

Puesto que NIS transmite información confidencial en la red, es importante que el servicio se ejecute detrás de un cortafuegos y en una red segmentada y segura. Siempre que la información de NIS sea transmitida sobre una red insegura, se correrá el riesgo de ser interceptada. El diseño cuidadoso de redes puede ayudar a prevenir infracciones graves de seguridad

2.2.3.2. Utilice un nombre de dominio NIS como contraseña y el nombre de host.

Cualquier máquina dentro de un dominio NIS puede usar comandos para extraer información desde el servidor sin necesidad de autenticación, siempre y cuando el usuario conozca el nombre de host DNS del servidor NIS y el nombre del dominio de NIS.

Por ejemplo, si alguien conecta un portátil en la red o ingresa sin permiso desde afuera (y logra husmear una dirección IP interna), el siguiente comando revelará el mapa **/etc/passwd**:

```
ypcat -d <NIS_domain> -h <DNS_hostname> passwd
```

Si el agresor es un usuario de root, puede obtener el archivo **/etc/shadow** mediante el siguiente comando:

```
ypcat -d <dominio de NIS> -h <nombre de host de DNS> shadow
```

Nota

Si se utiliza Kerberos, el archivo **/etc/shadow** no es almacenado dentro de un mapa de NIS.

Para hacer mucho más difícil el acceso a los mapas NIS para un agresor, cree una cadena aleatoria para el nombre de host de DNS, tal como **o7hfawtgmhwg.domain.com**. Igualmente, cree un nombre de dominio NIS aleatorio *diferente*.

2.2.3.3. Edite el archivo /var/yp/securenets

Si el archivo **/var/yp/securenets** está en blanco o no existe (como sucede después de una instalación predeterminada), NIS escucha a todas las redes. Una de las primeras cosas que se deben hacer es poner una máscara de red o pares de red en el archivo para que **ypserv** responda únicamente a la red apropiada.

La siguiente es una muestra de una entrada del archivo **/var/yp/securenets**:

```
255.255.255.0     192.168.0.0
```

Advertencia

Nunca inicie un servidor NIS por primera vez sin crear el archivo **/var/yp/securenets**.

Esta técnica no proporciona protección de un ataque de engaño de direcciones IP, pero al menos crea límites sobre las redes en que opera el servidor de NIS.

2.2.3.4. Asigne puertos estáticos y utilice reglas de iptables

Todos los servidores relacionados con NIS se pueden asignar a puertos específicos a excepción de **rpc.yppasswdd** — el demonio que permite a los usuarios cambiar sus contraseñas de ingreso. La asignación de puertos a otros dos demonios de servidor NIS, **rpc.ypxfrd** y **ypserv**, permite la creación de reglas de cortafuegos para proteger de intrusos los demonios del servidor NIS.

Para hacerlo, añada las siguientes líneas a **/etc/sysconfig/network**:

```
YPSERV_ARGS="-p 834" YPXFRD_ARGS="-p 835"
```

Las reglas iptables a continuación sirven para imponer la red que el servidor escuchará para estos puertos:

```
iptables -A INPUT -p ALL ! -s 192.168.0.0/24  --dport 834 -j DROP
iptables -A INPUT -p ALL ! -s 192.168.0.0/24  --dport 835 -j DROP
```

Esto significa que el servidor solamente permite conexiones a puertos 834 y 835 si las solicitudes vienen de la red 192.168.0.0/24, independientes del protocolo.

Nota

Consulte la *Sección 2.5, "Cortafuegos"* para obtener información sobre cómo implementar cortafuegos con comandos de iptables.

2.2.3.5. Utilice autenticación de kerberos

Uno de los aspectos a tener en cuenta cuando se utiliza NIS para autenticación, es que cada vez que un usuario se conecta a una máquina, un hash de la contraseña del mapa **/etc/shadow** se envía por la red. Si un intruso obtiene acceso a un dominio NIS y husmea el tráfico de red, puede recoletar nombres de usuario y contraseñas. Con tiempo suficiente, el programa de descifrado de contraseñas puede adivinar las contraseñas débiles, y el agresor puede obtener acceso a una cuenta válida en la red.

Puesto que Kerberos usa la criptografía de llave secreta, no se envían nunca contraseñas hash por la red, lo cual protege mucho más el sistema. Consulte, *Managing Single Sign-On and Smart Cards* para obtener mayor información sobre Kerberos.

2.2.4. Cómo proteger a NFS

Importante

La versión de NFS que se incluye en Red Hat Enterprise Linux 6, NFSv4, ya no requiere el servicio **portmap** como se establece en la *Sección 2.2.2, "Cómo proteger a Portmap"*. Ahora, el tráfico NFS utiliza TCP en todas las versiones, en lugar de UDP y requiere TCP cuando utiliza NFSv4. NFSv4 incluye ahora la autenticación de usuario Kerberos y de grupo, como parte del módulo de kernel **RPCSEC_GSS**. La información sobre **portmap** aún se incluye, porque Red Hat Enterprise Linux 6 soporta NFSv2 y NFSv3, y ambos utilizan **portmap**.

2.2.4.1. Planee cuidadosamente la red

Ahora NFSv4 tiene la capacidad de pasar por la red toda la información cifrada mediante kerberos, es importante que el servicio sea configurado correctamente si está detrás de un cortafuegos o de una red segmentada. NFSv2 y NFSv3 aún pasan datos de forma insegura y esto debería tenerse en cuenta. El diseño cuidadoso de redes en todos estos aspectos puede evitar a prevenir infracciones de seguridad.

2.2.4.2. Tenga cuidado con los errores de sintaxis

El servidor NFS determina los sistemas de archivos que se van a exportar y los hosts que se van a exportar a estos directorios si consulta el archivo **/etc/exports**. Tenga cuidado de no añadir espacios extraños al editar este archivo.

Por ejemplo, la siguiente línea en el archivo **/etc/exports** comparte el directorio **/tmp/nfs/** para el host **bob.example.com** con permisos de lectura o escritura.

```
/tmp/nfs/     bob.example.com(rw)
```

Por otra parte, la siguiente línea en el archivo **/etc/exports**, comparte el mismo directorio al del host **bob.example.com** con permisos de solo-lectura y lo comparte con el *mundo* con permisos de lectura o escritura debido al caracter de un espacio después del nombre de host.

```
/tmp/nfs/     bob.example.com (rw)
```

Es una buena práctica revisar los recursos compartidos NFS mediante el comando **showmount** para verificar si se están compartiendo.

```
showmount -e <hostname>
```

2.2.4.3. No utilice la opción no_root_squash

De forma predeterminada, los recursos compartidos NFS cambian el usuario de root por el usuario **nfsnobody**, una cuenta de usuario sin privilegios. Esto cambia el propietario de todos los archivos creados de root a **nfsnobody**, lo cual evita la carga de programas con un conjunto de bits de setuid.

Si se utiliza **no_root_squash**, los usuarios de root remotos pueden cambiar cualquier archivo en el sistema de archivos compartido y dejar las aplicaciones infectadas por troyanos para otros usuarios que las ejecuten sin darse cuenta.

2.2.4.4. Configuración de cortafuegos NFS

Los puertos utilizados por NFS son asignados de forma dinámica por rpcbind, lo cual puede ocasionar problemas al crear reglas de cortafuegos. Para simplificar este proceso, utilice el archivo */etc/sysconfig/nfs* para especificar los puertos que se deben usar:

- **MOUNTD_PORT** — puertos TCP y UDP para mountd (rpc.mountd)
- **STATD_PORT** — puertos TCP y UDP para estatus (rpc.statd)
- **LOCKD_TCPPORT** — puerto TCP para nlockmgr (rpc.lockd)
- **LOCKD_UDPPORT** — puerto UDP nlockmgr (rpc.lockd)

Los números de puertos especificados no deben ser utilizados por otro servicio. Configure su cortafuegos para autorizar los números de puertos especificados, al igual que el puerto TCP y UDP 2049 (NFS).

Ejecute el comando **rpcinfo -p** en el servidor NFS para ver los puertos y programas de RPC se están utilizando.

2.2.5. Cómo proteger el servidor HTTP de Apache

El servidor HTTP Apache es uno de los servicios más estables y seguros que se distribuye con Red Hat Enterprise Linux. Un gran número de opciones y técnicas están disponibles para proteger al servidor HTTP Apache — muy numerosas para profundizar aquí. La sección a continuación explica brevemente las buenas prácticas cuando se ejecuta el servidor HTTP Apache.

Siempre verifique si los scripts que se ejecutan en el sistema funcionan como se espera *antes* de colocarlos en producción. También asegúrese de que el usuario de root escriba permisos a cualquier directorio que contenga scripts o CGI. Para hacerlo, ejecute los siguientes comandos como usuario de root:

1.
```
chown root <directory_name>
```

2.
```
chmod 755 <directory_name>
```

Los administradores de sistemas deben tener cuidado al usar las siguientes opciones de configuración (configuradas en **/etc/httpd/conf/httpd.conf**):

FollowSymLinks
Esta directiva se habilita de forma predeterderminada, por lo tanto sea cauteloso al crear enlaces simbólicos en la raíz del documento del servidor de Web. Por ejemplo, es una mala idea proporcionar el enlace simbólico a **/**.\n

Indexes
Esta directiva está habilitada de forma predeterminada, pero puede no ser deseable. Para evitar que los visitantes naveguen los archivos en el servidor, retire esta directiva.

UserDir
La directiva **UserDir** se inhabilita de forma predeterminada porque puede confirmar la presencia de la cuenta de un usuario en el sistema. Para habilitar el directorio de usuario en el servidor, use las siguientes directivas:

```
UserDir enabled
UserDir disabled root
```

Estas directivas activan la navegación del directorio de usuario para todos los directorios de usuarios diferentes a **/root/**. Para añadir usuarios a la lista de cuentas inhabilitadas, añada una lista delimitada por espacios en la línea **UserDir disabled**.

Importante

No retire la directiva **IncludesNoExec** directive. Por defecto, el módulo *Server-Side Includes* (SSI) no puede ejecutar comandos. Se recomienda no cambiar esta configuración a menos que sea absolutamente necesario, ya que podría permitir que un agresor ejecute comandos en el sistema.

2.2.6. Cómo proteger FTP

El *Protocolo de transferencia de archivos* (FTP) es un protocolo TCP viejo diseñado para transferir archivos en la red. Puesto que todas las transacciones con el servidor, entre ellas la autenticación de usuarios, no están cifradas, se considera un protocolo inseguro y debe configurarse con cuidado.

Red Hat Enterprise Linux proporciona tres servidores FTP.

- **gssftpd** — Un kerberos que reconoce a **xinetd**-demonio basado en FTP no transmite información de autenticación por la red.
- **Red Hat Content Accelerator** (**tux**) — Un servidor de espacio de web con capacidades de FTP.
- **vsftpd** — Una implementación de seguridad autónoma del servicio FTP.

Las siguientes guías de seguridad son par establecer el servicio FTP **vsftpd**.

2.2.6.1. Pancarta de saludo de FTP

Antes de enviar el nombre de usuario y contraseña, se presenta de forma predeterminada a todos los usuarios una pancarta de saludo. Esta pancarta incluye información sobre la versión útil para crackers que tratan de identificar las debilidades del sistema.

Para cambiar la pancarta de saludo **vsftpd**, añada la siguiente directiva al archivo **/etc/vsftpd/vsftpd.conf**:

```
ftpd_banner=<inserte_saludo_aquí>
```

Remplace *<insert_greeting_here>* en la directiva de arriba por el texto del mensaje de saludo.

Para pancartas de varias líneas, es mejor utilizar un archivo de pancartas. A fin de simplificar la administración de varias pancartas, coloque todas las pancartas en el nuevo directorio llamado **/etc/banners/**. El archivo de pancartas para conexiones FTP en este ejemplo es **/etc/banners/ftp.msg**. El siguiente es un ejemplo de cómo se vería un archivo tal:\n

```
######### # Hello, all activity on ftp.example.com is logged. #########
```

Nota

No se necesita comenzar cada línea del archivo por **220** como se especifica en la *Sección 2.2.1.1.1, “Envolturas TCP y pancartas de conexión ”*.

Para hacer referencia a esta pancarta de saludo para **vsftpd**, añada la siguiente directiva al archivo **/etc/vsftpd/vsftpd.conf**:

```
banner_file=/etc/banners/ftp.msg
```

También se pueden enviar pancartas adicionales a conexiones entrantes mediante envolturas TCP como se describe en la *Sección 2.2.1.1.1, “Envolturas TCP y pancartas de conexión ”*.

2.2.6.2. Acceso de anónimo

La presencia del directorio **/var/ftp/** activa la cuenta anónima.

La forma más fácil de crear este directorio es instalar el paquete **vsftpd**. Este paquete establece un árbol de directorios para usuarios anónimos y configura los permisos en directorios de solo lectura para usuarios anónimos.

De forma predeterminada el usuario no puede escribir en ninguno de los directorios.

Advertencia

Si se habilita el acceso anónimo para un servidor FTP, tenga en cuenta en dónde se instala la información confidencial.se

2.2.6.2.1. Descarga de anónimo

Para autorizar a los usuarios anónimos a cargar archivos, se recomienda crear el directorio de solo escritura dentro de **/var/ftp/pub/**.\n

Para hacer esto, escriba el siguiente comando:

```
mkdir /var/ftp/pub/upload
```

Luego, cambie los permisos para que los usuarios anónimos no puedan ver el contenido del directorio:

```
chmod 730 /var/ftp/pub/upload
```

Un largo listado de directorio se debe ver así:

```
drwx-wx---    2 root     ftp          4096 Feb 13 20:05 upload
```

Advertencia

Los administradores que permiten a los usuarios anónimos leer y escribir en directorios suelen hallar que sus servicios se convierten en repositorio de software robado.

Además, bajo **vsftpd**, añada la siguiente línea al archivo **/etc/vsftpd/vsftpd.conf**:

```
anon_upload_enable=YES
```

2.2.6.3. Cuentas de usuarios

Puesto que FTP transmite nombres de usuarios y contraseñas sin cifrar en redes inseguras para autenticación, es una buena idea negar el acceso de usuarios del sistema al servidor desde sus cuentas de usuario.

Para inhabilitar todas las cuentas de usuario en **vsftpd**, añada la siguiente directiva a **/etc/vsftpd/vsftpd.conf**:

```
local_enable=NO
```

2.2.6.3.1. Cómo restringir las cuentas de usuarios

Para desactivar el acceso FTP para cuentas o grupos específicos de cuentas, tales como usuario de root y las personas con privilegios de **sudo**, la forma más fácil es usar la lista de PAM como se describe en la *Sección 2.1.4.2.4, "Desactivación de root mediante PAM"*. El archivo de configuración PAM para **vsftpd** es **/etc/pam.d/vsftpd**.

También es posible inhabilitar directamente\n las cuentas de usuario dentro de cada servicio.

Para inhabilitar las cuentas de usuario específico en **vsftpd**, añada el nombre de usuario a **/etc/vsftpd/ftpusers**

2.2.6.4. Use las envolturas TCP para controlar acceso

Use las envolturas TCP para controlar el acceso a cualquier demonio de FTP como se resume en la *Sección 2.2.1.1, "Cómo mejorar la seguridad con envolturas TCP"*.

2.2.7. Cómo proteger a Sendmail

Sendmail es el Agente de transferencia de correo (MTA) que utiliza el Protocolo simple de transferencia de correo (SMTP) para enviar mensajes electrónicos entre otros MTA y los clientes de correo-e o agentes de transmisión. Aunque muchos MTA pueden cifrar tráfico entre sí, la mayoría no, por lo tanto el envío de correo-e por redes públicas se considera de por sí una forma insegura de comunicación.

Si alguien está planeando implementar un servidor Sendmail, se recomienda abordar los siguientes aspectos:

2.2.7.1. Limitar el ataque de Denegación del servicio

Debido a la naturaleza del correo-e, un agresor puede con relativa facilidad inundar de correo al servidor y producir un ataque de Denegación del servicio. Al establecer límites para las siguientes directivas en **`/etc/mail/sendmail.mc`**, se limitará la eficacia de dichos ataques

- **`confCONNECTION_RATE_THROTTLE`** — El número de conexiones que el servidor puede recibir por segundo. Por defecto, Sendmail no limita el número de conexiones. Si se establece y alcanza el límite, otras conexiones se retrasarán.
- **`confMAX_DAEMON_CHILDREN`** — El número máximo de procesos hijos que pueden ser generados por el servidor. De forma predeterminada, Sendmail no asigna un límite al número de procesos. Si se establece y alcanza un límite, se retrasarán las siguientes conexiones.
- **`confMIN_FREE_BLOCKS`** — El número mínimo de bloques libres que deben estar disponibles para que el servidor acepte el correo. El número predeterminado de bloques es 100.
- **`confMAX_HEADERS_LENGTH`** — El tamaño máximo aceptable (en bytes) para un encabezado de mensaje.
- **`confMAX_MESSAGE_SIZE`** — El tamaño aceptable máximo (en bytes) para un mensaje.

2.2.7.2. NFS y Sendmail

Nunca ponga el directorio de cola de impresión de correo **`/var/spool/mail/`** en un volumen compartido NFS.

Puesto que NFSv2 y NFSv3 no mantienen el control sobre los ID de usuario y grupo, dos o más usuarios pueden tener el mismo UID, y recibir y leer el correo del otro.

Nota

Esto no sucede con NFSv4 que usa Kerberos, ya que el módulo de kernel **`SECRPC_GSS`** no utiliza autenticación basada en UID. Sin embargo, aún se considera una buena práctica *no* poner el directorio de cola de impresión de correo en volúmenes compartidos de NFS.

2.2.7.3. Usuarios de solo-correo

Para ayudar al usuario local a evitar vulnerabilidades en el servidor Sendmail, es mejor para los usuarios de correo accedan únicamente al servidor Sendmail mediante el programa de correo-e. Las cuentas de shell en el servidor de correo no se deben permitir y todas las shell de usuario en el archivo **/etc/passwd** se deben configurar para **/sbin/nologin** (con la posible excepción del usuario de root).

2.2.8. Cómo verificar los puertos que están escuchando

Después de configurar los servicios de red, es importante prestar atención a los puertos que están escuchando en las interfaces de red del sistema. Cualquier puerto abierto puede ser evidencia de una intrusión.

Existen dos métodos básicos para listar los puertos que escuchan en la red. El método menos fiable es la solicitud de la pila de redes mediante comandos tales como **netstat -an** o **lsof -i**. Este método es menos confiable, ya que los programas no se conectan a la máquina desde la red, sino que chequean cuál está en ejecución en el sistema. Por esta razón, estas aplicaciones son objetos frecuentes de remplazo para los agresores. Los crackers intentan cubrir sus rastros al abrir puertos de red no autorizadas remplazando a **netstat** y **lsof** por sus propias versiones modificadas.

La forma más confiable para revisar los puertos que están escuchando en la red es mediante el uso de un escáner tal como **nmap**.

El comando a continuación emitido desde la consola determina cuáles puertos están escuchando para conexiones TCP desde la red:

```
nmap -sT -O localhost
```

La salida de este comando aparece así:

```
Starting Nmap 4.68 ( http://nmap.org ) at 2009-03-06 12:08 EST
Interesting ports on localhost.localdomain (127.0.0.1):
Not shown: 1711 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
25/tcp    open  smtp
111/tcp   open  rpcbind
113/tcp   open  auth
631/tcp   open  ipp
834/tcp   open  unknown
2601/tcp  open  zebra
32774/tcp open  sometimes-rpc11
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.17 - 2.6.24
Uptime: 4.122 days (since Mon Mar  2 09:12:31 2009)
Network Distance: 0 hops
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.420 seconds
```

Esta salida muestra que el sistema está ejecutando **portmap** debido a la presencia del servicio **sunrpc**. Sin embargo, también hay un servicio misterioso en el puerto 834. Para verificar si el puerto está asociado con la lista de servicios conocidos, escriba:

```
cat /etc/services | grep 834
```

Este comando no retorna salida para el puerto 834. Debido al formato del comando, la salida para los puertos (1834, 2834 y 3834) se mostrará. Esto indica que mientras el puerto 834 está en el rango

reservado (es decir entre 0 y 1023) y requiere acceso de root para abrir, no se asocia con un servicio conocido.

Luego, verifique la información sobre el puerto mediante **netstat** o **lsof**. Para revisar el puerto 834 mediante **netstat**, use el siguiente comando:

```
netstat -anp | grep 834
```

El comando retorna la siguiente salida:

```
tcp   0    0 0.0.0.0:834    0.0.0.0:*   LISTEN   653/ypbind
```

La presencia del puerto abierto en **netstat** es tranquilizante puesto que un cracker que abre un puerto subrepticiamente en un sistema pirateado no es probable que le permita ser revelado a través de este comando. También, la opción **[p]** revela el ID del proceso (PID) del servicio que abrío el puerto. En este caso, el puerto abierto pertenece a **ypbind** (NIS), el cual es un servicio RPC manejado junto con el servicio **portmap**.

El comando **lsof** revela informaición simitar **netstat** puesto que también puede conectar los puertos abiertos a los servicios:

```
lsof -i | grep 834
```

La porción importante de la salida de este comando es la siguiente:

```
ypbind      653        0    7u  IPv4       1319                TCP *:834 (LISTEN)
ypbind      655        0    7u  IPv4       1319                TCP *:834 (LISTEN)
ypbind      656        0    7u  IPv4       1319                TCP *:834 (LISTEN)
ypbind      657        0    7u  IPv4       1319                TCP *:834 (LISTEN)
```

Estas herramientas revelan sobre el estatus de los servicios que se ejecutan en una máquina. Son herramients flexibles y pueden proporcionar una gran cantidad de información sobre servicios de red y configuración. Consulte las páginas de manual de **lsof**, **netstat**, **nmap** y **services** para obtener más información.

2.3. Envolturas TCP y xinetd

El control de acceso a los servicios de redes es una de las tareas de seguridad más importantes que enfrentan los administradores de servidores. Red Hat Enterprise Linux pone a su disposición varias herramientas para este propósito. Por ejemplo, los cortafuegos de **iptables** filtran, dentro de una pila de redes de kernel, los paquetes de red no esperados. Para los servicios de redes que los utilizan las *envolturas TCP* añaden una capa adicional de protección al definir los hosts que tienen o no permiso para conectarse a los servicios de redes "*wrapped*". Un servidor de redes con envolturas es el *súper servidor* `xinetd`. Este servicio se denomina súper servidor puesto que controla las conexiones para un subconjunto de servicios de redes y refina un mayor control de acceso.

Figura 2.4, “Control de acceso a servicios de redes” es una ilustración básica de cómo funcionan las herramientas para proteger los servicios de redes.

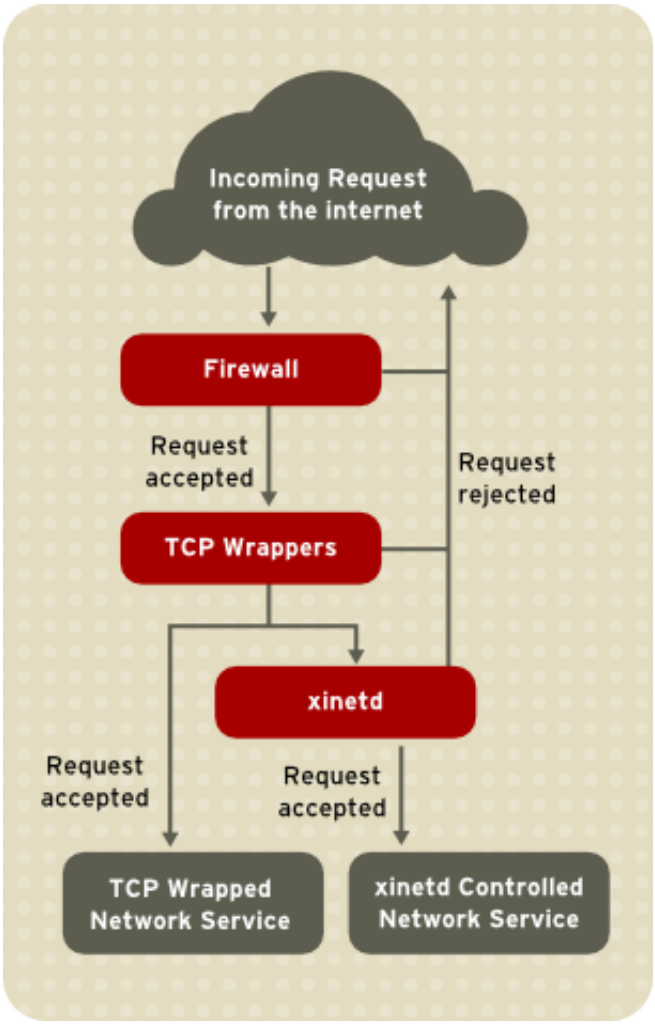


Figura 2.4. Control de acceso a servicios de redes

Esta capítulo se enfoca en el rol de las envolturas TCP y `xinetd` en el control de acceso a los servicios de redes y revistas de cómo pueden utilizarse estas herramientas para mejorar tanto el ingreso como la administración de uso. Consulte *Sección 2.6, "IPTables"* para obtener mayor información acerca de los cortafuegos con **`iptables`**.

2.3.1. Envolturas TCP

Los paquetes de envolturas TCP (**`tcp_wrappers`** y **`tcp_wrappers-libs`**) están instalados de forma predeterminada y proporcionan un control de acceso de host a los servicios de redes. El componente más importante dentro del paquete es la biblioteca **`/lib/libwrap.a`** or **`/lib64/libwrap.a`**. En términos generales, un servicio de envolturas TCP es uno que ha sido recopilado a nombre de la biblioteca **`libwrap.a`**.

Cuando se hace un intento de conexión a un servicio de envolturas TCP, el servicio primero hace referencia a los archivos de acceso de host (**`/etc/hosts.allow`** y **`/etc/hosts.deny`**) para determinar si el cliente está autorizado para conectarse. En la mayoría de los casos, luego usa el demonio de syslog (`syslogd`) para escribir el nombre del cliente que lo solicita y el servicio solicitado para **`/var/log/secure`** o **`/var/log/messages`**.

Si un cliente tiene permiso para conectarse, las envolturas TCP liberan el control de la conexión al servicio solicitado y no hacen parte de la comunicación entre el cliente y el servidor.

Además de acceder al control y registro, las envolturas TCP pueden ejecutar comandos para interactuar con el cliente antes de negar o liberar el control de la conexión al servicio de red solicitado.

Puesto que las envolturas TCP son una adición valiosa para el arsenal de herramientas de seguridad de cualquier administrador del servidor, la mayoría de los servicios dentro de Red Hat Enterprise Linux se conectan a la biblioteca **libwrap.a**. Algunas de estas aplicaciones incluyen `/usr/sbin/sshd`, **/usr/sbin/sendmail**, y `/usr/sbin/xinetd`.

Nota

Para determinar si el binario de servicio de redes está conectado a **libwrap.a**, escriba el siguiente comando como usuario de root:

```
ldd <binary-name> | grep libwrap
```

Remplace *<binary-name>* por el nombre del binario de servicio de redes.

Si el comando retorna directamente al indicador de comandos sin ninguna salida, entonces el servicio de redes *no* se vinculará a **libwrap.a**.

El ejemplo a continuación indica que `/usr/sbin/sshd` está vinculado a **libwrap.a**:

```
[root@myServer ~]# ldd /usr/sbin/sshd | grep libwrap
        libwrap.so.0 => /lib/libwrap.so.0 (0x00655000)
[root@myServer ~]#
```

2.3.1.1. Ventajas de las envolturas TCP

Las envolturas de TCP proporcionan las siguientes ventajas sobre las técnicas de control de servicios de red:

- *La transparecia tanto del cliente como del servicio de red protegido* — desconocen que las envolturas TCP están en uso. Los usuarios legítimos se registran y conectan al servicio solicitado mientras que las conexiones de los clientes no autorizados fallan.
- *Administración centralizada de múltiples protocolos* — Las envolturas TCP operan independientemente de los servicios de red que ellas protegen, permitiendo así que muchas aplicaciones de servidor compartan un conjunto común de archivos de configuración de control de acceso, para una administración más sencilla.

2.3.2. Archivos de configuración de envolturas TCP

Para determinar si un cliente tiene permiso de conectarse al servicio, las envolturas TCP hacen referencia a los dos archivos siguientes, los cuales se conocen como archivos de *acceso de hosts*:

- **/etc/hosts.allow**
- **/etc/hosts.deny**

Cuando un servicio de envolturas TCP recibe una solicitud de cliente, realiza los siguientes pasos:

1. *Hacer referencia a **/etc/hosts.allow*** — El servicio de envolturas TCP lee en secuencia el archivo **/etc/hosts.allow** y aplica la primera regla especificada para ese servicio. Si encuentra una regla que coincida, permitirá la conexión. Si no, irá al siguiente paso.

2. *Hacer referencia a* ***/etc/hosts.deny*** — El servicio de envolturas TCP lee en secuencia el archivo **/etc/hosts.deny**. Si encuentra la regla coincidente, negará la conexión. Si no, otorgará acceso al servicio.

Cuando utilice las envolturas TCP para proteger servicios de red, es importante tener en cuenta lo siguiente:

- Puesto que las reglas de acceso en **hosts.allow** se aplican primero, pueden tener prioridad sobre las reglas especificadas en **hosts.deny**. Por lo tanto, si se permite el acceso al servicio en **hosts.allow**, se omitirá la regla que niega el acceso al mismo servicio en **hosts.deny**.
- Las reglas en cada archivo se leen de arriba a abajo y la primera que concuerde para el servicio determinado será la única que se aplica. El orden de las reglas es extremadamente importante.
- Si no se encuentran reglas para el servicio en cada archivo o si no existe ningún archivo, se otorgará el acceso al servicio.
- Los servicios de envolturas TCP no guardan en cache las reglas de los archivos de acceso de hosts, por lo tanto los cambios a **hosts.allow** o **hosts.deny** se efectuarán inmediatamente, sin reiniciar los servicios de redes.

Advertencia

Si la última línea de un archivo de acceso de hosts no es un caracter de salto de línea (creado al presionar la tecla **Enter** key), la última regla en el archivo fallará y se registrará un error ya sea en **/var/log/messages** o en **/var/log/secure**. También es el caso para la regla que extiende varias líneas sin necesidad de usar el caracter de barra invertida. El ejemplo a continuación ilustra la parte importante de un mensaje de registro de un error de la directiva debido a alguna de estas circunstancias:

```
warning: /etc/hosts.allow, line 20: missing newline or line too long
```

2.3.2.1. Cómo dar formato a las reglas de acceso

El formato para **/etc/hosts.allow** y **/etc/hosts.deny** es idéntico. Cada regla debe tener su propia línea. Las líneas en blanco o líneas que inician con el símbolo de numeral (#) se omiten.

Cada regla usa el siguiente formato básico para controlar el acceso a servicios de redes:

```
<daemon list>: <client list> [: <option>: <option>: ...]
```

- *<Lista de demonios>* — Una lista de los nombres de procesos separados por coma (*no* los nombres del servicio) o el comodín **ALL**. La lista de demonios también acepta operadores (consulte la *Sección 2.3.2.1.4, "Operadores"*) para permitir mayor flexibilidad.
- *<Lista de clientes>* — Una lista de nombres de hosts separados por coma, direcciones IP de host, patrones especiales o comodines que identifican a los hosts afectados por la regla. El cliente también acepta los operadores listados en la *Sección 2.3.2.1.4, "Operadores"* para permitir mayor flexibilidad.
- *<opción>* — Una acción óptima o lista separada por comas de las acciones realizadas cuando se activa la regla. Los campos de opciones soportan expansiones, lanzan los comandos de shell, permiten o niegan el acceso y alteran la conducta de registro.

Nota

Puede obtener mayor información sobre los términos arriba mencionados en esta guía:

- *Sección 2.3.2.1.1, “Comodines”*
- *Sección 2.3.2.1.2, “Patrones”*
- *Sección 2.3.2.2.4, “Extensiones”*
- *Sección 2.3.2.2, “Campos de opciones\n”*

A continuación, una muestra básica de una regla de acceso de host:

```
vsftpd : .example.com
```

Esta regla le solicita a las envolturas TCP observar las conexiones para el demonio FTP (`vsftpd`) desde cualquier host en el dominio `example.com`. Si la regla aparece en **`hosts.allow`**, se acepta la conexión. Si la regla aparece en **`hosts.deny`**, se rechaza la conexión.

La siguiente muestra de reglas de acceso de hosts es más compleja y utiliza dos campos de opciones:

```
sshd : .example.com  \ : spawn /bin/echo `/bin/date` access denied>>/var/log/sshd.log \ :
 deny
```

Observe que cada campo de opción va precedido por la barra invertida (\). Use la barra invertida para evitar fallas de la regla debido a la longitud.

Esta regla de muestra establece que si se intenta una conexión al demonio SSH (`sshd`) desde un host en el dominio `example.com`, se ejecuta el comando **`echo`** para anexar el intento en un archivo de registro especial y se niega la conexión. Puesto que se utiliza la directiva opcional **`deny`**, esta línea niega el acceso incluso si aparece en el archivo **`hosts.allow`**. Consulte la *Sección 2.3.2.2, “Campos de opciones\n”* para obtener una visión más detallada en opciones disponibles.

2.3.2.1.1. Comodines

Los comodines le permiten a las envolturas TCP encontrar más fácilmente los grupos de demonios o hosts. Los comodines se utilizan en el campo de la lista del cliente de reglas de acceso.

Los siguientes comodines están disponibles:

- **`ALL`** — Concuerda con todo. Puede servir tanto para la lista de demonios como para la lista de clientes.
- **`LOCAL`** — Concuerda con cualquier host que no contiene un punto (.), tal como localhost.
- **`KNOWN`** — Concuerda con cualquier host cuyo nombre y dirección se conocen o cuando el usuario es conocido.
- **`UNKNOWN`** — Concuerda con cualquier host cuyo nombre o dirección se desconozcan o cuando el usuario es desconocido.

- **PARANOID** — Concuerda con cualquier host cuyo nombre no corresponda a la dirección.

Importante

Los comodines **KNOWN**, **UNKNOWN** y **PARANOID** se deben usar con cuidado, puesto que dependen de un servidor de DNS para corregir la operación. Cualquier interrupción de la resolución de nombre puede impedir que usuarios legítimos puedan acceder a un servicio.

2.3.2.1.2. Patrones

Los patrones se pueden usar en el campo de cliente de reglas de acceso para más grupos específicos de hosts de clientes.

A continuación, una lista de los patrones comunes para entradas en el campo de cliente:

- *Nombre de host que comienza por un punto (.)* — Al poner un punto en el comienzo de un nombre de host buscará todos los hosts que comparten los componentes listados del nombre. El ejemplo a continuación se aplica a cualquier host dentro del dominio `example.com`:

```
ALL : .example.com
```

- *Dirección IP que termina en un punto (.)* — Al poner un punto al final de una dirección IP encontrará todos los hosts que compartan los grupos numéricos de una dirección IP. El ejemplo a continuación se aplica a cualquier host dentro de la red `192.168.x.x`:

```
ALL : 192.168.
```

- *Par de dirección IP y máscara de red* — Las expresiones de máscara de red también sirven de patrón para controlar el acceso a un grupo de direcciones IP determinado. Los ejemplos a continuación se aplican a cualquier host con un rango de direcciones de `192.168.0.0` a `192.168.1.255`:

```
ALL : 192.168.0.0/255.255.254.0
```

Importante

Cuando se opera en un espacio de direcciones IPv4, el par de longitud del prefijo y dirección (*prefixlen*) las declaraciones de pares (notación CIDR) are no son compatibles. Solamente las reglas IPv6 pueden usar este formato.

- *[IPv6 address]/prefixlen pair* — los pares [net]/prefixlen también sirven de patrón para controlar el acceso a un grupo determinado de direcciones IPv6. El siguiente ejemplo se aplicaría a cualquier host con un rango de direcciones de `3ffe:505:2:1::` through `3ffe:505:2:1:ffff:ffff:ffff:ffff`:

```
ALL : [3ffe:505:2:1::]/64
```

- *Asterisco (*)* — Los asteriscos pueden usarse para buscar todos los grupos o nombres de hosts o direcciones IP siempre y cuando no estén mezclados en una lista de cliente que contenga otros tipos de patrones. El ejemplo a continuación se aplicaría a cualquier host dentro del dominio `example.com`:

```
ALL : *.example.com
```

- *Barra invertida (/)* — Si la lista de un cliente comienza por una barra invertida, se considerará como un nombre de archivo. Esto es útil si se necesitan las reglas que especifican grandes cantidades de hosts. El ejemplo a continuación, se refiere a las envolturas TCP para el archivo **`/etc/telnet.hosts`** para todas las conexiones de Telnet:

```
in.telnetd : /etc/telnet.hosts
```

Las envolturas TCP también aceptan otros patrones menos utilizados. Consulte la página de manual 5 **`hosts_access`** para obtener mayor información.

Advertencia

Sea cuidadoso al utilizar nombres de hosts y nombres de dominio. Los agresores pueden realizar una variedad de trucos para sortear la resolución de nombre. Además, la ruptura del servicio DNS evita incluso a usuarios autorizados el uso de servicios de red. Por lo tanto, es mejor usar las direcciones IP cuando sea posible.

2.3.2.1.3. Portmap y envolturas TCP

La implementación de **`Portmap`** de envolturas TCP no soporta la búsqueda de hosts, lo cual significa que **`portmap`** no puede usar nombres de host para identificar hosts. Como consecuencia, las reglas de control de acceso para portmap en **`hosts.allow`** o **`hosts.deny`** deben usar direcciones IP o la palabra clave **`ALL`**, para especificar hosts.

Los cambios a las reglas de control de acceso **`portmap`** no se efectúan inmediatamente. Necesitará reiniciar el servicio de **`portmap`**.

Los servicios ampliamente utilizados, tales como NIS y NFS, dependen de **`portmap`** para operar, por lo tanto tenga en cuenta estas limitaciones.

2.3.2.1.4. Operadores

En el momento, las reglas de control de acceso solamente aceptan el operador, **`EXCEPT`**. Se puede utilizar tanto para la lista de demonios como para la lista de clientes de una regla.

El operador **`EXCEPT`** permite a excepciones específicas ampliar las coincidencias dentro de la misma regla.

En el siguiente ejemplo del archivo **`hosts.allow`**, todos los hosts `example.com` tienen permiso para conectarse a los servicios, a excepción de `cracker.example.com`:

```
ALL: .example.com EXCEPT cracker.example.com
```

En otro ejemplo del archivo **`hosts.allow`**, los clientes de la red `192.168.0.`*x* pueden usar todos los servicios a excepción de FTP:

```
ALL EXCEPT vsftpd: 192.168.0.
```

Nota

Para ordenar, suele ser más fácil evitar el uso de los operadores **EXCEPT**. De esta manera se permite que otros administradores puedan escanear rápidamente los archivos para ver qué hosts tienen permiso o no de acceder a los servicios sin tener que ordenar a través de los operadores **EXCEPT**.

2.3.2.2. Campos de opciones\n

Además de las reglas básicas que permiten la negación o la autorización del acceso, la implementación de envolturas TCP de Red Hat Enterprise Linux soportan las extensiones para el lenguaje de control de acceso a través de *campos de opciones*. Al usar los campos de opciones en las reglas de acceso de hosts, los administradores pueden realizar una variedad de tareas, tales como alterar la conducta del registro, consolidar el control de acceso y lanzar comandos de shell.

2.3.2.2.1. Registro

Los campos de opciones le permiten a los operadores cambiar el recurso de registro y el nivel de prioridad para una regla mediante la directiva **severity**.

En el siguiente ejemplo, las conexiones al demonio SSH desde cualquier host en el dominio `example.com` se registran de forma predeterminada a la instalación **authpriv** **syslog** (porque no se especifica el valor de la instalación) con la prioridad de **emerg**:

```
sshd : .example.com : severity emerg
```

También se puede especificar un recurso mediante la opción **severity**. El ejemplo a continuación registra todos los intentos de conexión SSH por hosts desde el dominio `example.com` al recurso **local0** con una prioridad de **alert**:

```
sshd : .example.com : severity local0.alert
```

Nota

En práctica, este ejemplo no funciona hasta que el demonio syslog (`syslogd`) sea configurado para la instalación **local0**. Consulte la página de manual **syslog.conf** para obtener información sobre los recursos de registro personalizados.

2.3.2.2.2. Control de acceso

Los campos de opciones también permiten a los administradores autorizar o negar explícitamente hosts en una sola regla al añadir la directiva **allow** o **deny** respectivamente como opción final.

Por ejemplo, las dos reglas siguientes permiten conexiones SSH desde `client-1.example.com`, pero niegan conexiones desde `client-2.example.com`:

```
sshd : client-1.example.com : allow
```

```
sshd : client-2.example.com : deny
```

Al permitir el control de acceso basado en reglas, el campo de opciones permite a los administradores consolidar todas las reglas de acceso en un solo archivo: ya sea **hosts.allow** o **hosts.deny**. Algunos administradores consideran que esta es la forma más fácil de organizar las reglas de acceso.

2.3.2.2.3. Comandos de shell

Los campos de opciones autorizan a las reglas de acceso lanzar comandos de shell a través de las siguientes dos directivas:

- **spawn** — Lanza un comando de shell como un proceso hijo. Esta directiva puede realizar tareas tales como la de usar **/usr/sbin/safe_finger** para obtener mayor información sobre el cliente solicitado o para crear archivos de registro especiales mediante el comando **echo**.

  En el siguiente ejemplo, los clientes que intentan acceder a servicios de Telnet desde el dominio `example.com` se registran silenciosamente en un archivo especial:

  ```
  in.telnetd : .example.com \
   : spawn /bin/echo `/bin/date` from %h>>/var/log/telnet.log \
   : allow
  ```

- **twist** — Remplaza el servicio que se solicita por el comando especificado. Esta directiva suele utilizarse para establecer trampas a intrusos conocidas también como "honey pots" (ollas de miel). También pueden utilizarse para enviar mensajes a los clientes que se conectan. La directiva **twist** debe ir al final de la línea de la regla.

  En el ejemplo a continuación, los clientes que intentan acceder a servicios FTP desde el dominio `example.com` reciben un mensaje mediante el comando **echo**: command:

  ```
  vsftpd : .example.com \
   : twist /bin/echo "421 This domain has been black-listed. Access denied!"
  ```

Para obtener mayor información sobre opciones de comandos de shell, consulte la página de manual **hosts_options**.

2.3.2.2.4. Extensiones

Cuando se utilizan las extensiones con las directivas **spawn** y **twist**, proporcionan información sobre el cliente, servidor y procesos involucrados.

A continuación, una lista de extensiones admitidas:

- **%a** — Retorna la dirección IP del cliente.
- **%A** — Retorna la dirección IP del servidor.
- **%c** — Retorna una variedad de información de cliente, tal como nombre de usuario y nombre de host o nombre de usuario y dirección IP.
- **%d** — Retorna el nombre del proceso del demonio.
- **%h** — Retorna el nombre de host del cliente (o dirección IP, si el nombre de host no está disponible).
- **%H** — Retorna el nombre de host del servidor (o dirección IP, si el nombre de host no está disponible).

- **%n** — Retorna el nombre de host del cliente. Si no está disponible, aparecerá en pantalla: **unknown**. Si el nombre de host del cliente y la dirección del host no coinciden, aparecerá en pantalla: **paranoid**.
- **%N** — Retorna el nombre de host del servidor. Si no está disponible, aparecerá en pantalla: **unknown**. Si el nombre de host del servidor y la dirección del host no coinciden, aparecerá en pantalla: **paranoid**.
- **%p** — Retorna el ID del proceso del demonio.
- **%s** —Retorna varios tipos de información del servidor, tal como el proceso del demonio y la dirección IP del servidor.
- **%u** — Retorna el nombre de usuario del cliente. Si no está disponible, **unknown** aparece en pantalla.

La siguiente muestra de regla usa una extensión junto con el comando **spawn** para identificar el host del cliente en un archivo de registro personalizado.

Cuando se intentan conexiones al demonio SSH (`sshd`) desde un host en el dominio `example.com`, ejecute el comando **echo** para registrar el intento, el cual incluye el nombre de host del cliente (al usar la extensión **%h**), para un archivo especial:

```
sshd : .example.com  \
 : spawn /bin/echo `/bin/date` access denied to %h>>/var/log/sshd.log \
 : deny
```

Igualmente, las extensiones pueden servir para personalizar mensajes de regreso al cliente. En el ejemplo a continuación, a los clientes que intentan acceder a servicios FTP desde el dominio `example.com` se les informan que han sido rechazados del servidor:

```
vsftpd : .example.com \
: twist /bin/echo "421 %h has been banned from this server!"
```

Para obtener una explicación completa de las extensiones disponibles, como también de las opciones de control de acceso adicionales, consulte la sección 5 de las páginas de manual para **hosts_access** (**man 5 hosts_access**) y la página de manual para **hosts_options**.

Consulte la *Sección 2.3.5, “Recursos adicionales”* para obtener mayor información sobre envolturas TCP.

2.3.3. xinetd

El demonio `xinetd` es un *súper servicio* de envolturas TCP que controla acceso a un subconjunto de servicios de red populares, entre ellos FTP, IMAP y Telnet. También ofrece opciones de configuración para control de acceso, registro mejorado, vinculación, redirección y control de utilización de recursos.

Cuando un cliente intenta conectarse a un servicio de red controlado por `xinetd`, el súper servicio recibe la solicitud y revisa si hay reglas de control de acceso de envolturas TCP.

Al autorizar el acceso, `xinetd` verificará si la conexión está permitida bajo sus propias reglas de acceso para ese servicio. También verificará si el servicio puede tener más recursos asignados y si no infringe ninguna de las reglas definidas.

Si todas estas condiciones se cumplen (es decir, si se permite el acceso al servicio; si el servicio no ha alcanzado el límite de recursos y si el servicio no infringe ninguna regla definida), `xinetd`

iniciará una instancia del servicio solicitado y pasará control de la conexión. Cuando la conexión se establezca, `xinetd` no hará más parte de la comunicación entre el cliente y el servidor.

2.3.4. Archivos de configuración xinetd

Los archivos de configuración para `xinetd` son los siguientes:

- **`/etc/xinetd.conf`** — El archivo de configuración global `xinetd`.
- **`/etc/xinetd.d/`** — El directorio que contiene todos los archivos específicos del servicio.

2.3.4.1. El archivo /etc/xinetd.conf

El archivo **`/etc/xinetd.conf`** contiene los parámetros de configuración generales que afectan a cada servicio bajo el control de `xinetd`. Se lee cuando el servicio `xinetd` inicia, para que los cambios de configuración pueden efectuarse, necesitará reiniciar el servicio `xinetd`. A continuación, se presenta un ejemplo del archivo **`/etc/xinetd.conf`**:

```
defaults
{
  instances               = 60
  log_type                = SYSLOG authpriv
  log_on_success          = HOST PID
  log_on_failure          = HOST
  cps                     = 25 30
}
includedir /etc/xinetd.d
```

Estas líneas controlas los siguientes aspectos de `xinetd`:

- **`instances`** — Especifica el número máximo de solicitudes simultáneas que `xinetd` puede procesar.
- **`log_type`** — Configura `xinetd` para usar el recurso de registro **`authpriv`**, el cual registra las entradas al archivo **`/var/log/secure`**. Al añadir una directiva tal como **`FILE /var/log/xinetdlog`** se crearía un archivo de registro personalizado llamado **`xinetdlog`** en el directorio **`/var/log/`**.
- **`log_on_success`** — Configura a `xinetd` para que registre los intentos de conexión correctos. La dirección IP de host remoto predeterminada y el ID del proceso del servidor que procesa la solicitud serán registrados.
- **`log_on_failure`** — Configura a `xinetd` par registrar los intentos de conexiones fallidas o que han sido denegadas.
- **`cps`** — Configura a `xinetd` para que no permita más de 25 conexiones por segundo para ningún servicio. Si se sobrepasa este límite, el servicio será retirado por 30 segundos.
- **`includedir /etc/xinetd.d/`** — Incluye las opciones declaradas en los archivos de configuración del servicio localizados en el directorio **`/etc/xinetd.d/`**. Consulte la *Sección 2.3.4.2, “El directorio /etc/xinetd.d/”* para obtener mayor información.

Nota

Tanto los parámetros de **log_on_success** como de **log_on_failure** en el directorio **/etc/xinetd.conf** suelen ser modificados en los archivos de configuración específica del servicio. Por lo tanto, puede aparecer más información en el archivo de registro de un determinado archivo que en el archivo **/etc/xinetd.conf** puede indicar. Consulte la *Sección 2.3.4.3.1, "Opciones de registro"* para obtener mayor información.

2.3.4.2. El directorio /etc/xinetd.d/

El directorio **/etc/xinetd.d/** contiene los archivos de configuración para cada servicio administrado por `xinetd` y los nombres de los archivos se relacionan con el servicio. Como con **xinetd.conf**, este directorio se lee únicamente cuando se inicia el servicio `xinetd`. Para que los cambios se efectúen, el administrador debe reiniciar el servicio `xinetd`.

El formato de archivos en el directorio **/etc/xinetd.d/** usa las mismas convenciones que **/etc/xinetd.conf**. La razón principal por la cual la configuración para cada servicio se almacena en un servicio independiente es la de facilitar la personalización y para que otros servicios se afecten menos.

Para entender cómo se estructuran estos archivos, considere el archivo **/etc/xinetd.d/krb5-telnet**:

```
service telnet
{
  flags           = REUSE
  socket_type     = stream
  wait            = no
  user            = root
  server          = /usr/kerberos/sbin/telnetd
  log_on_failure  += USERID
  disable         = yes
}
```

Estas líneas controlan varios aspectos de control del servicio **telnet**:

- **service** — Especifica el nombre del servicio, por lo general uno de los que aparece en la lista del archivo **/etc/services**.
- **flags** — Establece el número de atributos para la conexión. **REUSE** le pide a `xinetd` que reutilice el socket para una conexión de Telnet.

Nota

El indicador **REUSE** está depreciado. Ahora, todos los servicios implícitamente usan el indicador **REUSE** .

- **socket_type** — Establece el tipo de conexión de red para **stream**.
- **wait** — Especifica si el servicio es de un solo hilo (**yes**) o de varios hilos (**no**).

- **user** — Especifica el ID de usuario bajo el cual se ejecuta el proceso.
- **server** — Especifica el binario ejecutable que se va a lanzar.
- **log_on_failure** — Especifica los parámetros para **log_on_failure** además de los que ya están definidos en **xinetd.conf**.
- **disable** — Especifica si el servicio está inhabilitado (**yes**) o habilitado (**no**).

Consulte la página de manual **xinetd.conf** para obtener mayor información sobre estas opciones y su uso.

2.3.4.3. Cómo alterar los archivos de configuración xinetd

Existe un rango de directivas disponible para los servicios protegidos por `xinetd`. Esta sección resalta algunas de las opciones más comunes.

2.3.4.3.1. Opciones de registro

Las opciones de registro a continuación están disponibles tanto para **/etc/xinetd.conf** como para los archivos de configuración de servicios específicos dentro del directorio **/etc/xinetd.d/**.

A continuación, una lista de algunas de las opciones de registro más utilizadas:

- **ATTEMPT** — Registra el hecho de que se hizo un intento fallido (**log_on_failure**).
- **DURATION** — Registra el tiempo de servicio utilizado por un sistema remoto (**log_on_success**).
- **EXIT** — Registra el estatus de salida o señal de terminación del servicio (**log_on_success**).
- **HOST** — Registra la dirección IP de host (**log_on_failure** y **log_on_success**).
- **PID** — Registra el ID de proceso del servidor que recibe la solicitud (**log_on_success**).
- **USERID** — Registra al usuario remoto que utiliza el método definido en RFC 1413 para todos los servicios de flujo de multihilos (**log_on_failure** y**log_on_success**).

Para obtener una lista de las opciones de registro, consulte la página de manual **xinetd.conf**.

2.3.4.3.2. Opciones de control de acceso

Los usuarios de servicios `xinetd` pueden elegir el uso de las reglas de acceso de hosts de envolturas TCP, proporcionar control de acceso a través de archivos de configuración de `xinetd` o una combinación de los dos. Consulte la *Sección 2.3.2, “Archivos de configuración de envolturas TCP ”* para obtener mayor información sobre archivos de control de acceso de hosts de envolturas TCPor more information about TCP Wrappers hosts access control files.

Esta sección aborda el uso de `xinetd` para controlar el acceso a servicios.

Nota

A diferencia de las envolturas TCP, los cambios al control de acceso se efectúan si el administrador de `xinetd` reinicia el servicio `xinetd`.

También, a diferencia de las envolturas TCP, el control de acceso a través de `xinetd` solamente afecta los servicios controlados por `xinetd`.

El control de acceso de hosts `xinetd` difiere del método utilizado por envolturas TCP. Cuando las envolturas TCP sitúan toda la configuración de acceso en dos archivos, **`/etc/hosts.allow`** y **`/etc/hosts.deny`**, el control de acceso de `xinetd` se encuentra en cada archivo de configuración en el directorio **`/etc/xinetd.d/`**.

Las opciones de acceso de hosts a continuación están soportadas por `xinetd`:

- **`only_from`** — Únicamente acepta hosts especificados para usar el servicio.
- **`no_access`** — Bloquea a los hosts listados para usar el servicio.
- **`access_times`** — Especifica el rango de tiempo que un servicio determinado puede utilizarse. El rango debe establecerse en un formato de 24 horas, HH:MM-HH:MM.

Las opciones **`only_from`** y **`no_access`** pueden utilizar una lista de todas las direcciones IP o nombres de hosts, o pueden especificar una red completa. Igual que las envolturas TCP, al combinar el control de acceso `xinetd` con la configuración de registro mejorada se puede aumentar la seguridad al bloquear solicitudes de los hosts rechazados mientras se registra detalladamente cada intento de conexión.

Por ejemplo, el siguiente archivo **`/etc/xinetd.d/telnet`** se utiliza para bloquear el acceso Telnet desde un grupo de red determinado y restringir todo el intervalo de tiempo que incluso los usuarios autorizados pueden ingresar:

```
service telnet
{
  disable         = no
  flags           = REUSE
  socket_type     = stream
  wait            = no
  user            = root
  server          = /usr/kerberos/sbin/telnetd
  log_on_failure  += USERID
  no_access       = 172.16.45.0/24
  log_on_success  += PID HOST EXIT
  access_times    = 09:45-16:15
}
```

En este ejemplo, cuando el sistema de cliente de una red `172.16.45.0/24` tal como `172.16.45.2`, intenta acceder al servicio Telnet, recibe el siguiente mensaje:

```
Conexión cerrada por un host externo
```

Además sus intentos de ingreso se registran en **`/var/log/messages`**, así:

```
Sep  7 14:58:33 localhost xinetd[5285]: FAIL: telnet address from=172.16.45.107
```

```
Sep  7 14:58:33 localhost xinetd[5283]: START: telnet pid=5285 from=172.16.45.107
Sep  7 14:58:33 localhost xinetd[5283]: EXIT: telnet status=0 pid=5285 duration=0(sec)
```

Al usar las envolturas TCP junto con los controles de acceso de `xinetd`, es importante entender la relación entre los dos mecanismos de control de acceso.

La siguiente es la secuencia de eventos seguidos por `xinetd` cuando el cliente solicita una conexión:

1. El demonio `xinetd` accede a las reglas de acceso de las envolturas TCP mediante una llamada de biblioteca **`libwrap.a`**. Si una regla de negación coincide con el cliente, la conexión se descartará. Si una regla de permiso coincide con el cliente, la conexión pasará a `xinetd`.

2. El demonio `xinetd` verifica sus propias reglas de acceso tanto para el servicio `xinetd` como para el servicio solicitado. Si una regla de negación coincide con el cliente, la conexión se descartará. De lo contrario, `xinetd` iniciará una instancia del servicio solicitado y pasará control de la conexión de ese servicio.

Importante

Se debe tener cuidado al usar los controles de acceso de las envolturas TCP junto con los controles de acceso `xinetd`. Si no se configuran correctamente pueden ocasionar efectos indeseables.

2.3.4.3.3. Opciones de vinculación y redirección

Los archivos de configuración de servicios para `xinetd` soportan la vinculación del servicio a una dirección IP y las solicitudes de entrada de redirección para ese servicio a otra dirección IP, nombre de host o puerto.

La vinculación se controla con la opción **`bind`** en los archivos de configuración específicos del servicio y enlaza al servicio a una dirección IP en el sistema. Cuando se configura, la opción **`bind`** únicamente permite solicitudes para acceder al servicio a la dirección IP correcta. Puede usar este método para vincular diferentes servicios a diversas interfaces de red con base en los requisitos.

Esto es bastante útil en los sistemas con múltiples adaptadores de red o con múltiples direcciones IP. En dichos sistemas, los servicios que no son seguros (como por ejemplo, Telnet), se pueden configurar para escuchar únicamente a la interfaz que está conectada a una red privada y no a la interfaz conectada a la Internet.

La opción **`redirect`** acepta una dirección IP o nombre de host seguido de un número de puerto. Esta opción configura el servicio para redirigir las solicitudes para este servicio a un host especificado o número de puerto. Esta funcionalidad puede utilizarse para señalar las diferentes direcciones IP en la misma máquina, cambiar la solicitud a un sistema y número de puerto totalmente diferentes o alguna combinación de estas opciones. Un usuario que se conecte a un servicio en un sistema puede por lo tanto ser redirigido a otro sistema sin interrupción.

El demonio `xinetd` puede realizar esta redirección al generar un proceso que permanezca vivo para la duración de la conexión entre el cliente que solicita la máquina y el host que proporciona el servicio, al transferir los datos entre los dos sistemas.

Las ventajas de las opciones de **`bind`** y **`redirect`** son evidentes cuando se utilizan juntas. Al vincular un servicio a una dirección IP particular en un sistema y luego redirigir las solicitudes para este servicio a una segunda máquina que únicamente la primera máquina puede ver, se puede utilizar un sistema interno para proporcionar servicios a una red totalmente diferente. También estas

opciones sirven para limitar la exposición de un determinado servicio en un equipo de host múltiple a una dirección IP conocida, como también para redirigir las solicitudes para ese servicio a otra máquina especialmente configurada con ese propósito.

Por ejemplo, considere un sistema que se utilice como cortafuegos con esta configuración para el servicio de Telnet:

```
service telnet
{
  socket_type  = stream
  wait   = no
  server   = /usr/kerberos/sbin/telnetd
  log_on_success  += DURATION USERID
  log_on_failure  += USERID
  bind                    = 123.123.123.123
  redirect                = 10.0.1.13 23
}
```

Las opciones **bind** y **redirect** en este archivo garantizan que el servicio de Telnet en la máquina esté asociado a una dirección IP externa (`123.123.123.123`), la que se encarga de la Internet. Además, las solicitudes al servicio de Telnet enviadas a `123.123.123.123` se redirigen a través de un segundo adaptador de red a una dirección IP interna (`10.0.1.13`) que únicamente el cortafuegos puede acceder. Luego, el cortafuegos envía la comunicación entre los dos sistemas y el sistema que se conecta piensa que está conectado a `123.123.123.123` cuando en realidad está conectado a una máquina diferente.

Esta funcionalidad es bastante útil para usuarios con conexiones de banda ancha y una sola dirección IP fija. Cuando se utiliza la Traducción de direcciones de red (NAT), los sistemas detrás de la máquina de puerta de enlace, que solamente utilizan las direcciones IP internas, no están disponibles desde fuera del sistema de puerta de enlace. Sin embargo, cuando algunos sistemas están controlados por las opciones **bind** y **redirect**, la máquina de puerta de enlace puede actuar como un proxy entre los sistemas externos y la máquina interna configurada para proporcionar el servicio. Además, las diversas opciones de registro y control de acceso `xinetd` están disponibles como protección adicional.\n

2.3.4.3.4. Opciones de administración de recursos

El demonio `xinetd` puede añadir un nivel básico de protección contra ataques de \tDenegación de Servicio (DoS). La siguiente es una lista de las directivas que ayudan a limitar la eficacia de dichos ataques:

- **per_source** — Define el número máximo de instancias para un servicio por dirección IP. Acepta únicamente enteros como argumento y puede utilizarse en **xinetd.conf** y en los archivos de configuración específicos del servicio en el directorio **xinetd.d/**.

- **cps** — Define el número máximo de conexiones por segundo. Esta directiva toma dos argumentos de enteros separados por un espacio en blanco. El primer argumento es el número máximo de conexiones por segundo permitidas para el servicio. El segundo argumento es el número de segundos que `xinetd` debe esperar antes de rehabilitar el servicio. Aparecen únicamente los enteros como argumentos y pueden ser utilizados en el archivo **xinetd.conf** o los archivos de configuración específicos del servicio en el directorio **xinetd.d/**.

- **max_load** — Define el uso de la CPU o umbral promedio de carga para un servicio. Acepta un argumento de número de punto flotante.

  El promedio de carga es una medida aproximada del número de procesos activos en un tiempo determinado. Vea los comandos **uptime**, **who** y **procinfo** para obtener mayor información sobre el promedio de carga.

Hay más opciones de administración de recursos disponibles para `xinetd`. Consulte la página de manual **`xinetd.conf`** para obtener mayor información.

2.3.5. Recursos adicionales

Mayor información sobre las envolturas TCP y `xinetd` se puede obtener en la documentación del sistema y la Internet.

2.3.5.1. Documentación sobre envolturas TCP instaladas

La documentación en su sistema es un buen lugar para empezar a buscar opciones de configuración adicionales para envolturas TCP, `xinetd` y control de acceso.

- **`/usr/share/doc/tcp_wrappers-<version>/`** — Este directorio contiene un archivo **`README`** que aborda cómo funcionan las envolturas TCP y los varios riesgos de nombres y direcciones de hosts falsos que existen.
- **`/usr/share/doc/xinetd-<version>/`** — Este directorio contiene un archivo **`README`** que aborda los aspectos de control de acceso y un archivo **`sample.conf`** con varias ideas para modificar los archivos de configuración de servicios en el directorio **`/etc/xinetd.d/`**.
- TCP Wrappers and `xinetd`-related man pages — Una cantidad de páginas de manual para varias aplicaciones y archivos de configuración relacionados con envolturas TCP y `xinetd`. A continuación presentamos las páginas de manual más importantes:

  Aplicaciones de servidor
  - **`man xinetd`** — La página de manual para `xinetd`.

  Archivos de configuración
  - **`man 5 hosts_access`** — La página de manual para los archivos de control de acceso de hosts de envolturas TCP.
  - **`man hosts_options`** — La página de manual para campos de envolturas TCP.
  - **`man xinetd.conf`** — El listado de la página de manual de opciones de configuración de `xinetd`.

2.3.5.2. Sitios web útiles sobre envolturas TCP

- *http://www.docstoc.com/docs/2133633/An-Unofficial-Xinetd-Tutorial* — Un guía que aborda varias formas de optimizar los archivos de configuración predeterminados de `xinetd` para cumplir con las metas de seguridad específicas.

2.3.5.3. Bibliografía relacionada

- *Hacking Linux Exposed* por Brian Hatch, James Lee y George Kurtz; Osbourne/McGraw-Hill — Un recurso de seguridad excelente con información sobre envolturas TCP y `xinetd`.

2.4. Redes privadas virtuales (VPN)

Las organizaciones con varias oficinas satelitales suelen conectarse entre sí con líneas dedicadas para eficiencia y protección de datos confidenciales en tránsito. Por ejemplo, muchos negocios usan la técnica de 'Frame relay' o líneas del *Modo de transferencia asíncrono* (ATM) como una solución de red de extremo a extremo para enlazar a una oficina con otras personas. Puede ser una propuesta costosa, especialmente para pequeñas y medianas empresas (SMB) que desea ampliar sin tener que pagar los altos costos asociados con el nivel empresarial de circuitos digitales dedicados.\n

Para hacer frente a esta necesidad, se desarrollaron las *Redes privadas virtuales* (VPN). Siguiendo los mismos principios funcionales de los circuitos dedicados, VPN permite la comunicación segura entre dos partes (o redes), al crear una *Red de área amplia* (WAN) desde las *Redes de área locales* (LAN). Difiere del Frame relay o ATM en el medio de transporte. VPN transmite en IP mediante datagramas como capa de transporte, haciéndolo un conducto seguro a través de la Internet al destino planeado. La mayoría de las implementaciones de software libre de VPN incorporan los métodos de cifrado estándar para enmascarar aún más los datos en tránsito.

Algunas organizaciones emplean hardware de soluciones VPN para aumentar la seguridad, mientras que otros utilizan software o implementaciones de protocolo. Varios proveedores ofrecen harware de soluciones VPN, tales como Cisco, Nortel, IBM y Checkpoint. Hay un software libre basado en soluciones VPN para Linux llamado FreeS/Wan que utiliza la implementación de *Seguridad de protocolo de Internet* (IPsec). Estas soluciones de VPN, sin importar si se basan en hardware o software, actúan como enrutadores especializados que existen entre la conexión IP desde una ofician a otra.

2.4.1. ¿Cómo funciona la VPN?

Cuando se transmite un paquete desde un cliente, el cliente lo envía a través del enrutador o puerta de enlace de VPN, el cual añade un *Encabezado de autenticación* (AH) para enrutado y autenticación. Los datos se cifran y por último, se encierran en una *Carga de seguridad encapsuladora* (ESP). Más adelante constituirá las instrucciones de descifrado y manejo.

En enrutador receptor de VPN quita el encabezado de información, descifra los datos y los dirige a su destino (ya sea la estación de trabajo o el nodo en una red). Al usar una conexión de red a red, el nodo receptor en la red local recibe los paquetes descrifrados y listos para ser procesados. El proceso de cifrado y descifrado en una conexión de red a red de VPN es transparente para el nodo local.

Con ese nivel de seguridad, el agresor no solamente deberá interceptar el paquete, sino también descifrarlo. Los intrusos que emplean un ataque de hombre en el medio entre un servidor y un cliente deben también acceder al menos a una de las llaves privadas para autenticación de sesiones. Puesto que emplean varias capas de autenticación y cifrado, las VPN son un medio seguro y efectivo para actuar como una intranet unificada.

2.4.2. Openswan

2.4.2.1. Visión general

Visión general

Openswan es una implementación IPsec a nivel de kernel de código abierto en Red Hat Enterprise Linux. Emplea protocolos de establecimiento de llaves IKE (Internet Key Exchange) v1 y v2, implementadas como demonios de usuario. También se pueden establecer las llaves de forma manual a través de los comandos **`ip xfrm`**, sin embargo no se recomienda.

Soporte criptográfico

Openswan tiene una biblioteca criptográfica incorporada, no obstante también soporta la biblioteca de NSS (Servicios de seguridad de red), la cual está completamente soportada y se requiere para cumplimiento de seguridad FIPS. Para mayor información sobre FIPS (Federal Information Processing Standard) puede buscar en la *Sección 7.2, “Estándar de procesamiento de información federal (FIPS)”*.

Instalación

Ejecute el comando **yum install openswan** para instalar Openswan.

2.4.2.2. Configuración

Sitios

Esta sección enumera y describe los directorios y archivos utilizados para configurar Openswan.

- Directorio principal - **/etc/ipsec.d**. Almacena archivos relacionados con Openswan.
- Archivo maestro de configuración - **/etc/ipsec.conf**. Otros archivos de configuración ***.conf** se pueden crear en **/etc/ipsec.d** para configuraciones individuales.
- Archivo de secretos maestro - **/etc/ipsec.secrets**. Otros archivos ***.secrets** se pueden crear en **/etc/ipsec.d** para configuraciones individuales.
- Certificado de archivo de base de datos - **/etc/ipsec.d/cert*.db**. El archivo anterior predeterminado de base de datos NSS es **cert8.db**. Desde Red Hat Enterprise Linux 6 en adelante, las bases de datos NSS sqlite se utilizan en el archivo **cert9.db**.
- Archivos de claves de base de datos - **/etc/ipsec.d/key*.db**. El archivo anterior de base de datos predeterminado NSS es **key3.db**. Desde Red Hat Enterprise Linux 6 en adelante, las bases de datos NSS sqlite se utilizan en el archivo **key4.db**.
- Sitio para certificados de autoridad certificadora (CA) **/etc/ipsec.d/cacerts**.
- Sitio para certificados de usuario **/etc/ipsec.d/certs**. No son necesarios al usar NSS.
- Políticas de grupos - **/etc/ipsec.d/policies**. Las políticas se pueden definir como *de bloque*, *claras*, *claras-o-privadas*, *privadas*, *privadas-o-claras*.
- Archivo de contraseñas NSS - **/etc/ipsec.d/nsspassword**. Este archivo no existe de forma predeterminada y se requiere si la base de datos de NSS en uso se crea con una contraseña.

Parámetros de configuración

Esta sección enumera algunas de las opciones de configuración disponibles, la mayoría escritas a **/etc/ipsec.conf**.

- **protostack** - define la pila de protocolo que se utiliza. La opción predeterminada en Red Hat Enterprise Linux 6 es *netkey*. Otros valores válidos *auto*, *klips* y *mast*.
- **nat_traversal** - define si la solución de NAT para conexiones es aceptada. Se predetermina como no.
- **dumpdir** - define el sitio para archivos de vaciado de núcleo.
- **nhelpers** - Al usar NSS, se define el número de hilos utillizados para operaciones criptográficas. Si no se utiliza NSS, se define el número de procesos utilizados para operaciones criptográficas.
- **virtual_private** - subredes permitidas para conexión de clientes. Los rangos que pueden existir detrás de un enrutador NAT a través de los cuales se conecta el cliente.
- **plutorestartoncrash** - se establece a sí de forma predeterminada.
- **plutostderr** - ruta para registro de error pluto. Señala el sitio syslog de forma predeterminada.

- **connaddrfamily** - puede establecerse como ipv4 o ipv6.

Mayor información sobre la configuración de Openswan se encuentra en la página de manual **ipsec.conf(5)**.

2.4.2.3. Comandos

Esta sección explica y da ejemplos de algunos de los comandos utilizados para Openswan.

Nota

Como se ilustra en el siguiente ejemplo, el uso de **service ipsec start/stop** es el método recomendado para cambiar el estado del servicio ipsec. También el la técnica que se recomienda para iniciar y detener todos los servicios en Red Hat Enterprise Linux 6.

- Cómo iniciar y detener a Openswan:
  - **ipsec setup start/stop**
  - **service ipsec start/stop**
- Adición o borrado de una conexión:
  - **ipsec auto --add/delete <connection name>**
- Establecer conexión o desconexión
  - **ipsec auto --up/down <connection-name>**
- Generación de llaves RSA:
  - **ipsec newhostkey --configdir /etc/ipsec.d --password password --output /etc/ipsec.d/<name-of-file>**
- Revisión de políticas ipsec en el kernel:
  - **ip xfrm policy**
  - **ip xfrm state**
- Creación de un certificado de auto-firmado:
  - **certutil -S -k rsa -n <ca-cert-nickname> -s "CN=ca-cert-common-name" -w 12 -t "C,C,C" -x -d /etc/ipsec.d**
- Creación de certificado de usuario firmado por la autoridad certificadora anterior:
  - **certutil -S -k rsa -c <ca-cert-nickname> -n <user-cert-nickname> -s "CN=user-cert-common-name" -w 12 -t "u,u,u" -d /etc/ipsec.d**

2.4.2.4. Recursos de Openswan\n

- *http://www.openswan.org*
- *http://lists.openswan.org/pipermail/users/*

- *http://lists.openswan.org/pipermail/dev/*
- *http://www.mozilla.org/projects/security/pki/nss/*
- El paquete *Openswan-doc*: HTML, examples, README.*
- README.nss

2.5. Cortafuegos

La protección de seguridad comúnmente se considera más un proceso que un producto. Sin embargo, las implementaciones de seguridad estándar suelen emplear alguna forma de mecanismo dedicado para controlar privilegios de acceso y restringir recursos de redes a usuarios autorizados, identificables y rastreables. Red Hat Enterprise Linux incluye varias herramientas para ayudar a administradores e ingenieros de seguridad en problemas de control de acceso a nivel de redes.

Los cortafuegos son uno de los componentes de implementación de seguridad de redes. Varios proveedores ponen a disposición soluciones de cortafuegos para todos los niveles del mercado: desde los usuarios de hogares para proteger un computador personal hasta soluciones de centros de datos que protegen información vital empresarial. Los cortafuegos pueden ser soluciones de hardware autónomas tales como dispositivos de cortafuegos de Cisco, Nokia, y Sonicwall. Los proveedores tales como Checkpoint, McAfee, y Symantec también han desarrollado cortafuegos de software de propietario para hogares y mercados comerciales.a

Aparte de las diferencias entre cortafuegos de hardware y de software, hay también diferencias en la forma como los cortafuegos funcionan que separan una solución de la otra. *Tabla 2.2, "Tipos de cortafuegos"* para obtener mayor información sobre los tres tipos más comunes de cortafuegos y de cómo funcionan:

Tabla 2.2. Tipos de cortafuegos

Método	Descripción	Ventajas	Desventajas
NAT	*Traducción de dirección de red* (NAT) sitúa subredes IP detrás de una o un grupo pequeño de direcciones IP públicas, las cuales enmascaran todas las solicitudes a una fuente en lugar de varias. El kernel de Linux tiene una funcionalidad NAT incorporada a través del subsistema de kernel Netfilter.	· No se puede configurar de modo transparente para máquinas en una LAN · La protección de varias máquinas y servicios detrás de una o más direcciones externas IP simplifica las labores administrativas · La restricción de acceso de usuario a y desde la LAN puede configurarse al abrir y cerrar puertos en la puerta de enlace o cortafuegos NAT	· No se puede evitar actividad maliciosa una vez que los usuarios se conecten al servicio fuera del cortafuegos
Filtro de paquetes	Un cortafuegos de filtraje de paquetes lee cada paquete de datos que pasa a través de una LAN. Puede leer y procesar paquete por información de encabezado y flitra el paquete basado en conjuntos de reglas progamables	· Personalizable a través de la herramienta de entorno **`iptables`** · No requiere ninguna personalización de parte del cliente, ya que toda la actividad de red se filtra en el nivel del enrutador en lugar del nivel de aplicación	· No puede filtrar paquetes para contenidos como cortafuegos de proxy · Procesa paquetes en la capa del protocolo, pero no puede filtrar paquetes en la capa de la aplicación · Las arquitecturas de redes complejas pueden dificultar

Método	Descripción	Ventajas	Desventajas
	implementadas por el administrador de cortafuegos. El kernel de Linux tiene una funcionalidad de filtraje incorporada a través del subsistema de kernel Netfilter.	· Puesto que los paquetes no se transmiten a través de un proxy, el rendimiento de redes es más rápido debido a la conexión directa de cliente a host remoto	el establecimiento de reglas de filtraje, especialmente si se emparejan con *enmascaramiento IP* o subredes locales y redes DMZ
Proxy	Los cortafuegos de proxy filtran todas las solicitudes de algún protocolo o tipo desde clientes LAN a una máquina proxy y luego hacen esas solicitudes a la Internet en nombre del cliente local. Una máquina proxy actúa como un buffer entre usuarios malintencionados remotos y las máquinas de cliente de redes internas.	· Provee control a los administradores sobre qué aplicaciones y protocolos funcionan fuera de la LAN · Algunos servidores proxy pueden almacenar en cache datos frecuentemente accedidos de forma local en lugar de tener que usar la conexión de Internet para solicitarlos. De esa manera se ayuda a reducir el consumo de ancho de banda. · Servicios proxy pueden registrarse y monitorizarse de cerca, lo que permite un control mayor en el uso de recursos en la red	· Los proxy suelen ser aplicaciones específicas (HTTP, Telnet, etc.), o de protocolo restringido (la mayoría de proxy funcionan con servicios conectados de TCP únicamente) · Los servicios de aplicaciones no se pueden ejecutar detrás de un proxy, por lo tanto sus servidores de aplicaciones deben usar una forma independiente de seguridad de redes · Los proxy pueden convertirse en un cuello de botella de redes, ya que todas las solicitudes y transmisiones se pasan directamente a través de una fuente en lugar de un cliente a un servicio remoto

2.5.1. Netfilter e IPTables

El kernel de Linux presenta un subsistema de redes poderoso llamado *Netfilter*. El subsistema de Netfilter provee un filtraje de paquetes con y sin estado como también servicios de NAT y enmascarmiento de IP. Netfilter también tiene la habilidad de *exprimir* información de encabezamiento de IP y de administrar un estado de conexión. Netfilter se controla mediante la herramienta **`iptables`**.

2.5.1.1. Vision general de IPTables

El poder y la flexibilidad de Netfilter se implementa mediante la herramienta de administración de **`iptables`**, una herramienta de línea de comandos similar en sintaxis a su predecesora, **`ipchains`**, la cual fue remplazada por Netfilter o iptables en el kernel de Linux 2.4 y versiones superiores.

`iptables` usa el subsistema Netfilter para mejorar la conexión de redes, la inspección y el procesamiento. **`iptables`** ofrece ingreso avanzado, acciones de pre y post-enrutamiento, traducción de dirección de redes y el reenvío de puertos, todo en una interfaz de línea de comandos.

Esta sección proporciona una visión general de **`iptables`**. Para obtener mayor información, consulte *Sección 2.6, “IPTables”*.

2.5.2. Configuración básica de cortafuegos

Justo como en un cortafuegos en un edificio intenta evitar que un incendio se extienda, el cortafuegos en informática intenta evitar que software malintencionado se extienda a su equipo. También ayuda a evitar que usuarios no autorizados accedan a su equipo.

En una instalación predeterminada de Red Hat Enterprise Linux un cortafuegos existe entre su equipo o red y cualquier red desconocida, por ejemplo la Internet. Este determina los servicios que los usuarios remotos pueden acceder. Un cortafuegos correctamente configurado puede aumentar en gran medida la seguridad de su sistema. Se recomienda configurar el cortafuegos para todo sistema de Red Hat Enterprise Linux con una conexión de Internet.

2.5.2.1. Firewall Configuration Tool

Durante la instalación de Red Hat Enterprise Linux de la pantalla **Configuración de cortafuegos**, se le dio la opción de activar el cortafuegos básico y los dispositivos específicos, servicios de entrada y puertos.

Tras la instalación, puede cambiar esta preferencia mediante **Firewall Configuration Tool**.

Para iniciar esta aplicación, utilice el siguiente comando:

```
[root@myServer ~] # system-config-firewall
```

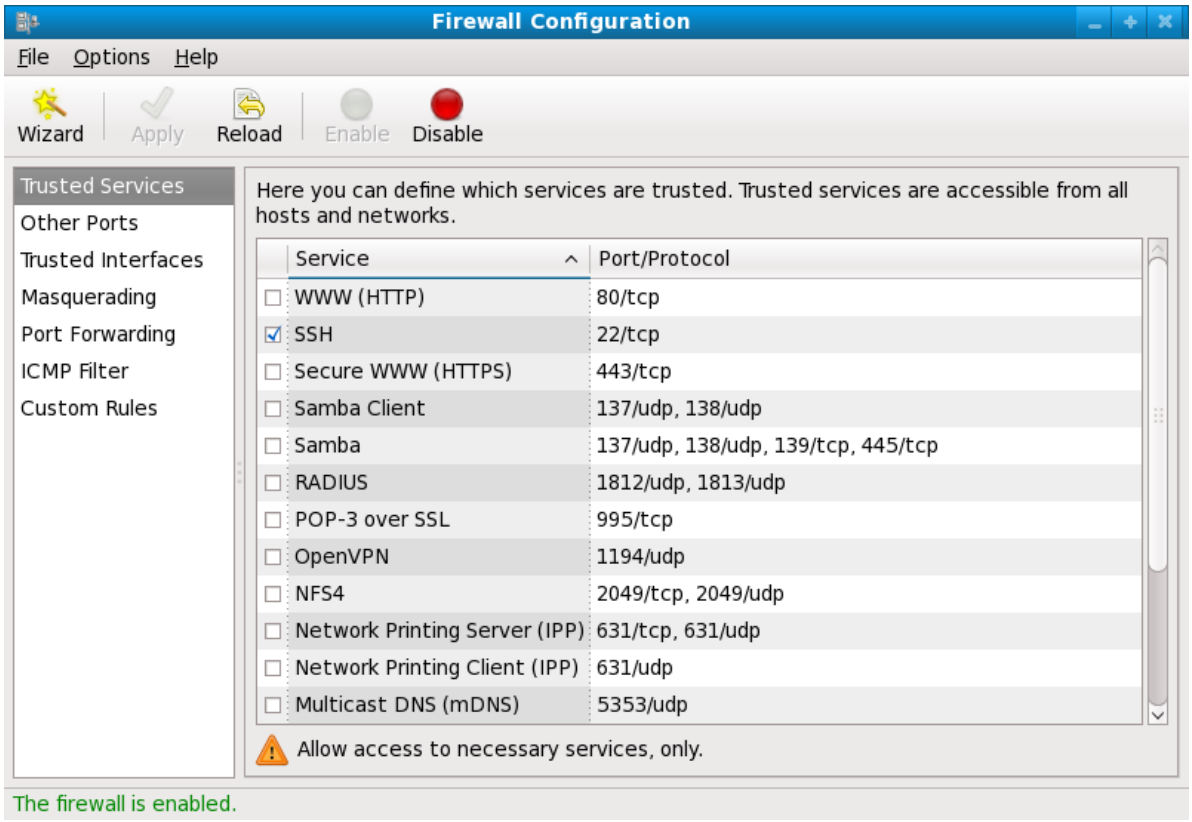


Figura 2.5. **Firewall Configuration Tool**

Nota

Firewall Configuration Tool solamente configura un cortafuegos básico. Si el sistema necesita reglas más complejas, consulte *Sección 2.6, "IPTables"* para obtener mayor información sobre cómo configurar reglas **`iptables`** específicas.

2.5.2.2. Habilitar o inhabilitar el cortafuegos

Seleccione una de las siguientes opciones para el cortafuegos:

- **Desactivado** — La desactivación del cortafuegos proporciona acceso completo al sistema y no verifica la seguridad. Solamente debe seleccionarse si se ejecuta en una red confiable (no la Internet) o si necesita configurar un cortafuegos personal mediante la línea de comandos de iptables.

Advertencia

Las configuraciones de cortafuegos y las reglas personalizadas de cortafuegos se almacenan en el archivo **`/etc/sysconfig/iptables`**. Si selecciona **Disactivado** y hace clic en **OK**, estas reglas de configuraciones y cortafuegos se perderán.

- **Activado** — Esta opción configura el sistema para que rechace las conexiones de entrada que no responden a solicitudes salientes, tales como respuestas de DNS o solicitudes de DHCP. Si es necesario el acceso a servicios que se están ejecutando en esta máquina, puede elegir servicios específicos a través del cortafuegos.

  Si conecta su sistema a la Internet, pero no planea ejecutar un servidor, esta es la opción más segura.

2.5.2.3. Servicios fiables

Si activa las opciones en la lista de **Servicios fiables** permitirá que el servicio especificado pase a través del cortafuegos.

WWW (HTTP)

El protocolo HTTP es utilizado por Apache (y por otros servidores de red) para servir páginas web. Si planea que su servidor esté disponible al público, seleccione esta cajilla de verificación. Esta opción no se requiere para ver localmente páginas o para desarrollar páginas web. Este servicio requiere que el paquete **`httpd`** esté instalado.

Al habilitar **WWW (HTTP)** no se abrirá el puerto para HTTPS, la versión sencilla de HTTP. Si requiere este servicio, seleccione la cajilla de verificación **Secure WWW (HTTPS)**.

FTP

El protocolo FTP sirve para transferir archivos entre máquinas en una red. Si planea que el servidor FTP esté disponible al público, seleccione esta cajilla de verificación. Este servicio requiere que el paquete **`vsftpd`** esté instalado.

SSH
Shell segura (SSH) es un paquete de herramientas para ingresar y ejecutar comandos en una máquina remota. Par permitir acceso remoto a una máquina a través de ssh, seleccione esta cajilla de verificación ahora. Este servicio requiere que el paquete **`openssh-server`** esté instalado.

Telnet
Telnet es un protocolo para ingresar a máquinas remotas. Las comunicaciones de Telnet están cifradas y no proporcionan seguridad de redes. Se recomienda permitir el acceso de Telnet. Para permitir el acceso remoto la máquina a través de Telnet, seleccione esta cajilla de verificación: Este servicio requiere que el paquete **`telnet-server`** esté instalado.

Mail (SMTP)
SMTP es un protocolo que permite a los hosts conectarse directamente al equipo para enviar correo. No necesita activar este servicio si recoge el correo desde su servidor de ISP mediante POP3 o IMAP, o si utiliza una herramienta tal como **`fetchmail`**. Para permitir la entrega de correo a su máquina, seleccione esta cajilla de verificación. Observe que un servicio SMTP mal configurado puede permitir que máquinas remotas usen su servidor para enviar correo no deseado.

NFS4
El Sistema de archivos de red (NFS) es un protocolo para compartir archivos comúnmente usados en sistemas *NIX. La versión 4 de este protocolo es más segura que sus predecesoras. Si desea compartir archivos o directorios en su sistema con otros usuarios de red, seleccione esta cajilla de verificación.

Samba
Samba es una implementación del protocolo de red SMB propietario de Microsoft. Si necesita archivos compartidos, directorios o impresoras conectadas localmente con equipos de Microsoft Windows, seleccione esta cajilla de verificación,

2.5.2.4. Otros puertos

Firewall Configuration Tool incluye la sección de **Otros puertos** para especificar los puertos IP personales confiables por **`iptables`**. Por ejemplo, para permitir a IRC y al protocolo de impresora (IPP) pasar a través del cortafuegos, añada la siguiente sección **Other ports**:

```
194:tcp,631:tcp
```

2.5.2.5. Cómo guardar la configuración

Haga clic en **OK** para guardar los cambios y habilitar o inhabilitar el cortafuegos. Si seleccionó **Activar cortafuegos**, las opciones seleccionadas se traducen en comandos de **`iptables`** y se escriben al archivo **`/etc/sysconfig/iptables`**. El servicio **`iptables`** también se inicia para que el cortafuegos se active inmediatamente después de guardar las opciones seleccionadas. Si seleccionó **Desactivar cortafuegos**, el archivo **`/etc/sysconfig/iptables`** se elimina y el servicio de **`iptables`** se detiene inmediatamente.

Las opciones seleccionadas también se escriben al archivo **`/etc/sysconfig/system-config-firewall`** para que la configuración se restaure la próxima vez que la aplicación inicie. No modifique a mano este archivo.

Incluso si el cortafuegos se activa inmediatamente, el servicio **`iptables`** no está configurado para iniciar de forma automática en el momento del inicio. Consulte *Sección 2.5.2.6, “Activación del servicio de IPTables”* para obtener mayor información.

2.5.2.6. Activación del servicio de IPTables

Las reglas de cortafuegos se activan únicamente si el servicio **iptables** está en ejecución. Para iniciar el servicio, use el siguiente comando:

```
[root@myServer ~] # service iptables restart
```

Para garantizar que **iptables** inicie en el arranque del sistema, use el siguiente comando:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

2.5.3. Uso de IPTables

El primer paso para usar **iptables** es iniciar el servicio de **iptables**. Use el comando a continuación para iniciar el servicio **iptables**:

```
[root@myServer ~] # service iptables start
```

Nota

El servicio de **ip6tables** puede apagarse si intenta usar únicamente el servicio de **iptables**. Si desactiva el servicio de **ip6tables**, recuerde desactivar también la red IPv6. Nunca deje el dispositivo de red activo sin el correspondiente cortafuegos.

Para forzar a **iptables** para que inicie de forma predeterminada al arrancar el sistema, use el siguiente comando:

```
[root@myServer ~] # chkconfig --level 345 iptables on
```

Así se fuerza a **iptables** a iniciar cada vez que el sistema se arranque en nivel de ejecución 3, 4, o 5.

2.5.3.1. Sintaxis del comando de IPTables

El siguiente comando **iptables** ilustra la sintaxis básica del comando:

```
[root@myServer ~ ] # iptables -A <cadena> -j <destino>
```

La opción **-A** especifica que la regla debe ser añadida a <*chain*>. Cada cadena comprende una o más *reglas* y se conoce también como *conjunto de reglas*.

Las tres cadenas incorporadas son ENTRADA, SALIDA, REENVÍO. Estas cadenas son permanentes y no se pueden borrar. La cadena especifica el punto en el cual se manipula el paquete.

La opción **-j <*target*>** especifica el destino de la regla; es decir, qué hacer si el paquete coincide con la regla. ACEPTAR, ENTREGAR Y RECHAZAR son ejemplos de destinos incorporados.

Consulte la página de manual **iptables** para obtener mayor información en las cadenas disponibles, opciones y destinos.

2.5.3.2. Políticas básicas de cortafuegos

El establecimiento de políticas básicas de cortafuegos crea una base para la construcción más detallada, de reglas de usuario.

Cada cadena de **iptables** consta de una política predeterminada y cero o más reglas que funcionan en concierto con la política predeterminada para definir todo el conjunto de reglas para el cortafuegos.

La política predeterminada para una cadena puede ser DROP o ACCEPT. Los administradores orientados a la seguridad implementan una política predeterminada de DROP y aceptan paquetes específicos en una base de caso por caso. Por ejemplo, las siguientes políticas bloquean todos los paquetes de salida en la puerta de enlace de red:

```
[root@myServer ~ ] # iptables -P INPUT DROP
[root@myServer ~ ] # iptables -P OUTPUT DROP
```

También se recomienda denegar el tráfico de redes de *paquetes reenviados* — es decir que el tráfico que sea dirigido desde el cortafuegos hasta su nodo de destino — también sea negado para restringir una exposición accidental a la Internet de clientes internos. Para ello, utilice la siguiente regla:

```
[root@myServer ~ ] # iptables -P FORWARD DROP
```

Cuando haya establecido las políticas predeterminadas para cada cadena, puede crear y guardar las nuevas reglas para su red y los requerimientos de seguridad.

Las secciones a continuación describen cómo almacenar las reglas de iptables y resume algunas de las reglas que podrían implementarse durante la construcción de su cortafuegos de iptables.

2.5.3.3. Guardado y restauración de reglas de IPTables

Los cambios a **iptables** son transitorios; si vuelve a arrancar el sistema o si reinicia el servicio de **iptables**, las reglas se pierden y se restablecen de forma automática. Para guardar las reglas para que se carguen al iniciar el servicio de **iptables**, use el siguiente comando:

```
[root@myServer ~ ] # service iptables save
```

Las reglas se guardan en el archivo **/etc/sysconfig/iptables** y se aplican cuando el servicio inicia o al arrancar la máquina.

2.5.4. Filtraje de IPTables comunes

Evitar que los agresores remotos accedan a una LAN es uno de los aspectos más importantes de la seguridad de red. La integridad de una LAN se debe proteger de usuarios malintencionados remotos a través del uso de reglas rigurosas de cortafuegos.

Sin embargo, con una política predeterminada para bloquear todos los paquetes entrantes, salientes y reenviados, es imposible para los usuarios de cortafuegos o puertas de enlace y para que los usuarios internos de LAN comunicarse entre sí o con los recursos externos.

Para permitir a los usuarios realizar funciones relacionadas con la red y usar aplicaciones de redes, los administradores deben abrir algunos puertos de comunicación.

Por ejemplo, para permitir el acceso al puerto 80 *en el cortafuegos*, añada la siguiente regla:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

De esta manera los usuarios pueden navegar sitios web que se comunican mediante el puerto estándar 80. Para permitir el acceso a sitios de web seguros (por ejemplo, https://www.example.com/), debe dar acceso al puerto 443, así:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

Importante

Al crear un conjunto de reglas de **iptables**, el orden es importante.

Si la regla especifica que los paquetes de la subred 192.168.100.0/24 sean eliminados, y esté acompañada por la regla que permite paquetes desde 192.168.100.13 (los cuales están dentro de la subred eliminada), entonces la segunda regla se omitirá.

La regla que permite paquetes desde 192.168.100.13 debe preceder a la regla que elimina las restantes de la subred.

Para insertar una regla en un sitio específico en una cadena existente, use la opción **-I**. Por ejemplo:

```
[root@myServer ~ ] # iptables -I INPUT 1 -i lo -p all -j ACCEPT
```

Esta regla se inserta como la primera regla en la cadena de ENTRADA para permitir el tráfico de dispositivo de bucle local.

A veces cuando se requiera el acceso remoto a la LAN, los servicios seguros, por ejemplo SSH, sirven para conexión remota cifrada con los servicios de LAN.

Para los administradores con recursos basados en PPP (tales como bancos o cuentas masivas ISP), el acceso telefónico se puede usar para burlar las barreras de cortafuegos de forma segura. Debido a que son conexiones directas, las conexiones de módem están típicamente detrás de un cortafuegos o puerta de enlace.

Sin embargo, para usuarios remotos con conexiones de banda ancha, se pueden crear casos especiales. Puede configurar **iptables** para aceptar conexiones desde clientes remotos SSH. Por ejemplo, las siguientes reglas permiten el acceso remoto SSH:

```
[root@myServer ~ ] # iptables -A INPUT -p tcp --dport 22 -j ACCEPT
[root@myServer ~ ] # iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

Estas reglas permiten el acceso de entrada y salida para un sistema individual, tal como un computador personal conectado directamente a la Internet o a un cortafuegos o puerta de enlace. No obstante, no permiten que nodos detrás del cortafuegos o puerta de enlace accedan estos servicios. Para permitir el acceso de LAN a estos servicios, puede usar *Traducción de dirección de red* (NAT) con reglas de filtraje de **iptables**.

2.5.5. Reglas FORWARD y NAT

La mayoría de los ISP proporcionan un número limitado de direcciones IP dirigibles públicamente a las organizaciones que sirven.

Los administradores deben, por lo tanto, buscar otras formas de acceder a los servicios de Internet sin dar direcciones IP públicas a cada nodo en la LAN. El uso de direcciones IP privadas es la forma

más común de permitir que todos los nodos en una LAN accedan correctamente a los servicios internos y externos de red.

Los enrutadores perimetrales (tales como los cortafuegos) pueden recibir transmisiones de entrada desde la Internet y dirigir los paquetes al nodo de LAN. Al mismo tiempo, los cortafuegos o puertas de enlace también pueden dirigir solicitudes de salida desde un nodo de LAN a un servicio de Internet remoto.

El reenvío de tráfico de redes puede ser peligroso a veces, especialmente con la disponibilidad de herramientas de pirateo modernas que pueden enmascarar direcciones IP *internas* y hacer que los agresores utilicen la máquina como un nodo en su LAN.

`iptables` proporciona políticas de enrutamiento y reenvío que se pueden implementar para prevenir el uso anormal de los recursos de la red.

La cadena de **`FORWARD`** permite al administrador controlar a dónde se pueden dirigir los paquetes dentro de una LAN. Por ejemplo, para permitir el reenvío de toda la LAN (asumiendo que al cortafuegos o puerta de enlade se le asigna una dirección IP interna en eth1), use las siguientes reglas:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth1 -j ACCEPT
[root@myServer ~ ] # iptables -A FORWARD -o eth1 -j ACCEPT
```

Esta regla provee acceso a la red interna detrás del cortafuegos o puerta de enlace. La puerta de enlace dirige los paquetes desde un nodo de LAN a su nodo de destino, pasando todos los paquetes a través de su dispositivo **`eth1`**.

Nota

Por defecto, la política IPv4 en kernel de Red Hat Enterprise Linux desactiva el soporte para reenvío IP. De esta manera, evita que las máquinas que ejecutan Red Hat Enterprise Linux funcionen como enrutadores perimetrales dedicados. Para activar el reenvío de IP, use el siguiente comando:

```
[root@myServer ~ ] # sysctl -w net.ipv4.ip_forward=1
```

Este cambio de configuración es únicamente válido para la sesión actual; no persiste en el re-arranque o reinicio del servicio de red. Para configurar de forma permanente el reenvío de IP, edite el archivo **`/etc/sysctl.conf`**, así:

Localice la siguiente línea:

```
net.ipv4.ip_forward = 0
```

Edítela para que se lea así:

```
net.ipv4.ip_forward = 1
```

Use el siguiente comando para permitir el cambio en el archivo **`sysctl.conf`**:

```
[root@myServer ~ ] # sysctl -p /etc/sysctl.conf
```

2.5.5.1. Post-enrutamiento y enmascarado de IP

La aceptación de paquetes reenviados a través del dispositivo IP interno de cortafuegos permite que nodos de LAN se comuniquen entre sí; no obstante aún no pueden comunicarse externamente a la Internet.

Para que los nodos LAN con direcciones IP privadas puedan comunicarse con redes públicas externas, configure el cortafuegos para *IP masquerading*, el cual enmascara solicitudes desde nodos LAN con la dirección IP del dispositivo externo de cortafuegos (en este caso, eth0):

```
[root@myServer ~ ] # iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Esta regla usa la tabla que concuerda con el paquete de NAT (**`-t nat`**) y especifica la cadena incorporada POSTROUTING para NAT (**`-A POSTROUTING`**) en el dispositivo de red externo de cortafuegos (**`-o eth0`**).

POST-Enrutamiento permite la alteración de paquetes cuando están saliendo del dispositivo externo de cortafuegos.

El destino **`-j MASQUERADE`** se especifica para enmascarar la dirección IP privada de un nodo con la dirección IP externa del cortafuegos o Puerta de enlace.

2.5.5.2. Pre-enrutamiento

Si tiene un servidor en la red interna que desee cambiar a externo, use la cadena de destino de PRE-ENRUTAMIENTO en NAT **`-j DNAT`** para especificar una dirección de destino IP y puerto en el que los paquetes que solicitan una conexión a su servicio interno pueden ser reenviados.

Por ejemplo, si desea reenviar solicitudes de entrada HTTP a su servidor dedicado Apache HTTP en 172.31.0.23, use el siguiente comando:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to
 172.31.0.23:80
```

Esta regla especifica que la tabla nat usa la cadena de PRE-ENRUTAMIENTO incorporada para reenviar las solicitudes entrantes de HTTP exclusivamente a la dirección IP de 172.31.0.23.

Nota

Si usted tiene una política predeterminada de DROP en su cadena FORWARD, debe añadir una regla para reenviar todas las solicitudes HTTP para que el destino de enrutamiento de NAT sea posible. Para hacer esto, use el siguiente comando:

```
[root@myServer ~ ] # iptables -A FORWARD -i eth0 -p tcp --dport 80 -d 172.31.0.23 -j
 ACCEPT
```

Esta regla reenvía todas las solicitudes de HTTP desde el cortafuegos al destino; el servidor APACHE HTTP detrás del cortafuegos.

2.5.5.3. DMZs e IPTables

Puede crear reglas **`iptables`** para enrutar el tráfico a algunas máquinas, tales como un servidor HTTP o FTP, en una *zona desmilitarizada* (DMZ). Un DMZ es una subred local especial dedicada a proporcionar servicios en un transportador público, tal como la Internet.

Por ejemplo, para establecer una regla para solicitudes HTTP de enrutamiento entrantes a un servidor dedicado HTTP en 10.0.4.2 (fuera del rango de LAN 192.168.1.0/24), NAT emplea la tabla **`PREROUTING`** para reenviar los paquetes al destino apropiado:

```
[root@myServer ~ ] # iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-
destination 10.0.4.2:80
```

Con este comando, todas las conexiones HTTP al puerto 80 desde fuera de LAN se enrutan al servidor HTTP en una red independiente del resto de la red interna. Esta forma de segmentación puede ser más segura que las conexiones que permiten conexiones HTTP a una máquina en la red.

Si el servidor HTTP está configurado para que acepte conexiones seguras, entonces el puerto 443 también se debe reenviar.

2.5.6. Software malintencionado y direcciones IP falsas

Se pueden crear reglas más elaboradas para controlar el acceso a subredes específicas o incluso nodos específico, dentro de una LAN. También puede restringir el contacto a su servidor de algunas aplicaciones dudosas o programas tales como troyanos, worms u otros virus de cliente y servidor.

Por ejemplo, algunos troyanos escanean redes de servicios en puertos de 31337 a 31340 (llamados los puertos *elite* en terminología de ciberpiratas).

Puesto que no hay servicios ilegítimos que se comuniquen a través de estos puertos no estándar, el bloquearlos puede disminuir efectivamente las posibilidades de que nodos infectados en su red se comuniquen de forma independiente con sus servidores maestros remotos.

Las siguientes reglas descargan todo el tráfico TCP para usar puerto 31337:

```
[root@myServer ~ ] # iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
[root@myServer ~ ] # iptables -A FORWARD -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
```

También puede bloquear conexiones externas que intenten suplantar rangos de direcciones IP privadas para infiltrar su LAN.

Por ejemplo, si su LAN usa el rango 192.168.1.0/24, usted puede diseñar una regla que instruya al dispositivo de red de Internet (por ejemplo, eth0) para que descargue los paquetes a ese dispositivo con una dirección en su rango IP de LAN.

Puesto que, como política predeterminada, se recomienda rechazar los paquetes reenviados, cualquier otra dirección IP engañosa para el dispositivo externo (eth0) es rechazada automáticamente.

```
[root@myServer ~ ] # iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -j DROP
```

Nota

Hay una diferencia entre los destinos **DROP** y **REJECT** cuando se emplean las reglas *appended*.

El destino **REJECT** niega acceso y retorna un error de **connection refused** para usuarios que intentan conectar el servicio. El destino **DROP**, como su nombre lo indica, lanza el paquete sin ninguna advertencia.

Está a discreción de los administradores el uso de estos destinos. Sin embargo, para evitar la confusión de usuario e intentos de continuar conectado, se recomienda el destino **REJECT**.

2.5.7. IPTables y trazado de conexiones

Para inspeccionar y restringir conexiones a servicios basados en su *estado de conexión.* El módulo dentro de **iptables** usa el método llamado *rastreo de conexión* para almacenar información sobre conexiones entrantes. Puede aceptar o negar acceso con base en los siguientes estados de conexión:

- **NEW** — Un paquete que solicita una nueva conexión, tal como una solicitud HTTP.
- **ESTABLISHED** — Un paquete que hace parte de una coexión existente.
- **RELATED** — Un paquete que solicita una nueva conexión pero que hace parte de una conexión existente. Por ejemplo, FTP usa el puerto 21 para establecer una conexión, pero los datos se transfieren en un puerto diferente (por lo general, el puerto 20).
- **INVALID** — Un paquete que no hace parte de ninguna conexión en la tabla de seguimiento de conexión.

Puede utilizar la funcionalidad de estado de la conexión de **iptables** que rastrea con cualquier protocolo de redes, incluso si el protocolo mismo no tiene estado (tal como UDP). El siguiente ejemplo muestra la regla que usa el seguimiento de conexión para reenviar únicamente los paquetes asociados a una conexión establecida:

```
[root@myServer ~ ] # iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

2.5.8. IPv6

La introducción del Protocolo de Internet de la siguiente generación, llamada IPv6, se extiende más allá del límite de direcciones de 32 bits de IPv4 (o IP). IPv6 soporta direcciones de 128 bits y las redes de transporte que reconocen a IPv6 pueden, por lo tanto, dirigirse a un número mayor de direcciones enrutables que IPv4.

Red Hat Enterprise Linux soporta el cortafuegos IPv6 mediante el sub-sistema Netfilter 6 y el comando de **ip6tables**. En Red Hat Enterprise Linux 6, tanto los servicio IPv4 como los servicios IPv6 se habilitan de forma predeterminada.

La sintaxis del comando de **ip6tables** es idéntica a **iptables** en cada aspecto excepto en el número de direcciones de 128 bits que soporta. Por ejemplo, use el siguiente comando para habilitar conexiones SSH en el servidor de redes que reconoce IPv6:

```
[root@myServer ~ ] # ip6tables -A INPUT -i eth0 -p tcp -s 3ffe:ffff:100::1/128 --dport 22 -j
 ACCEPT
```

Para mayor información sobre el uso redes IPv6, consulte la página de información sobre IPv6 en *http://www.ipv6.org/*.

2.5.9. Recursos adicionales

Hay varios aspectos para cortafuegos y sub-sistemas de Netfilter Linux que no se pueden abordar en este capítulo. Para obtener mayor información, consulte los siguientes recursos:

2.5.9.1. Documentación instalada de cortafuegos

- Consulte *Sección 2.6, "IPTables"* para obtener mayor información sobre el comando **iptables**, entre ellos definiciones para varias opciones de comandos.
- La página de manual **iptables** contiene un breve resumen de varias opciones.

2.5.9.2. Páginas web útiles sobre cortafuegos

- *http://www.netfilter.org/* — La página principal oficial de Netfilter y del proyecto de **iptables**.
- *http://www.tldp.org/* — El proyecto de documentación de Linux contiene varias guías relacionadas con la creación y administración de cortafuegos.
- *http://www.iana.org/assignments/port-numbers* — La lista oficial de puertos de servicios comunes y registrados como asignados por la Autoridad de números asignados de Internet.

2.5.9.3. Documentación relacionada

- *Red Hat Linux Firewalls*, por Bill McCarty; Red Hat Press — una referencia completa de la creación de cortafuegos de red y de servidor mediante la tecnología de filtrado de paquetes de código abierto tal como Netfilter e **iptables**. Incluye temas que abordan el análisis de registros de cortafuegos, el desarrollo de reglas de cortafuegos y la personalización de su cortafuegos mediante varias herramientas gráficas.
- *Linux Firewalls*, por Robert Ziegler; New Riders Press — contiene una información valiosa sobre la construcción de cortafuegos mediante **ipchains** de kernel 2.2 como también de Netfilter e **iptables**. Además aborda temas sobre aspectos de acceso remoto y sistemas de detección de intrusos.

2.6. IPTables

En Red Hat Enterprise Linux vienen incluidas las herramientas avanzadas para *filtraje de paquetes* — de redes, el proceso de control de paquetes de red en el ingreso, desplazamiento, control y salida de la pila de redes dentro del Kernel. Las versiones de kernel anteriores a 2.4 dependían de **ipchains** para filtraje de paquetes y utilizaban listas de reglas aplicadas a paquetes en cada paso del proceso del filtraje. El kernel 2.4 introdujo **iptables** (conocidas también como *netfilter*), el cual es similar a **ipchains** pero extiende ampliamente el alcance y control disponibles para filtrar paquetes de redes.

Este capítulo se enfoca en los fundamentos de filtraje de paquetes, explica varias opciones disponibles con comandos de **iptables** y cómo se pueden preservan las reglas de filtraje entre reinicios del sistema.

Consulte *Sección 2.6.6, "Recursos adicionales"* para obtener instrucciones sobre cómo crear reglas **iptables** y cómo establecer un cortafuegos en ellas.

Importante

El mecanismo predeterminado en el kernel 2.4 y posteriores es **iptables**, pero **iptables** no se puede utilizar si **ipchains** ya se está ejecutando. Si **ipchains** está presente en el momento de arranque, el kernel expide un error y falla al iniciar **iptables**.

La funcionalidad de **ipchains** no se afecta por esos errores.

2.6.1. Filtraje de paquetes

El kernel de Linux emplea la herramienta **Netfilter** para filtrar paquetes, lo que permite a algunos de ellos ser recibidos o pasados mediante el sistema mientras detiene a otros. Esta herramienta se incorpora en el kernel de Linux y tiene tres *tablas* o *listas de reglas* incorporadas, así:

- **filter** — La tabla predeterminada para manejar paquetes de redes.
- **nat** — Sirve para alterar paquetes que crean una nueva conexión y es utilizado por *Traducción de dirección de red* (*NAT*).
- **mangle** — Sirve para tipos específicos de alteración de paquetes.

Cada tabla tiene un grupo de *cadenas* incorporadas que corresponde a las acciones realizadas en el paquete por **netfilter**.

Las cadenas incorporadas para la tabla de **filtraje** son las siguientes:

- *INPUT* — Se aplica a los paquetes de redes destinados al host.
- *OUTPUT* — Se aplica a los paquetes de red generados localmente.
- *FORWARD* — Se aplica a los paquetes de redes enrutados a través del host.

Las cadenas incorporadas para la tabla **nat** son las siguientes:

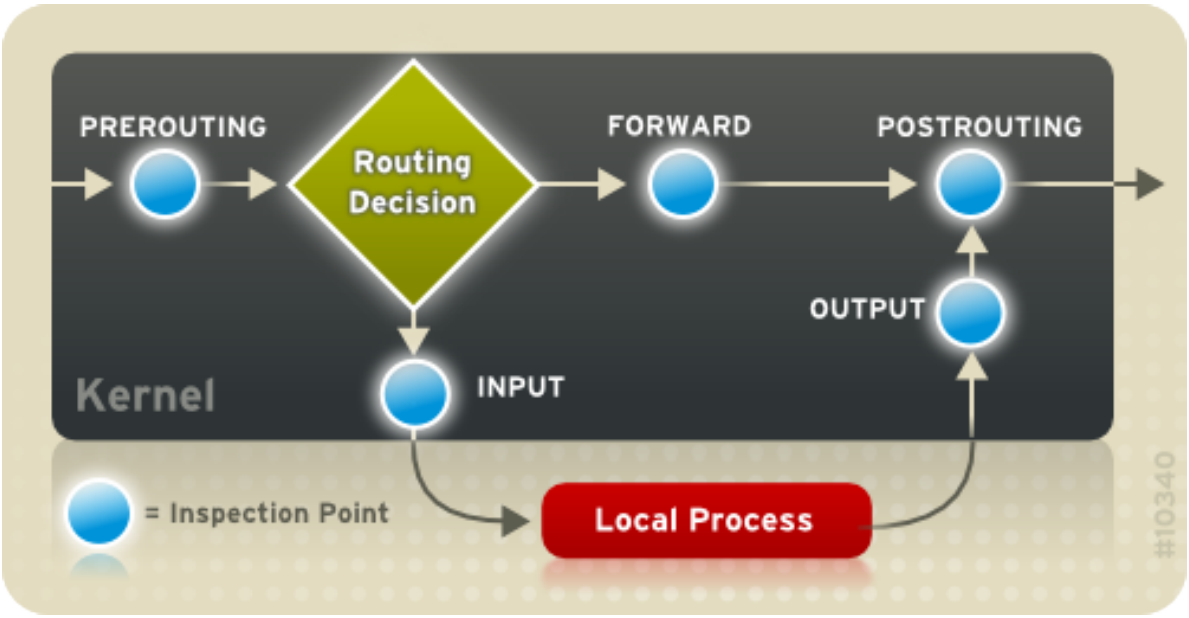
- *PREROUTING* — Altera los paquetes de redes a la llegada.
- *OUTPUT* — Altera los paquetes de redes generados localmente antes de ser enviados.
- *POSTROUTING* — Altera los paquetes de redes antes de ser enviados.

Las cadenas incorporadas para la tabla de **mangle** son las siguientes:

- *INPUT* — Altera los paquetes de redes destinados al host.
- *OUTPUT* — Altera los paquetes de redes generados localmente antes de ser enviados.
- *FORWARD* — Altera los paquetes de redes enrutados a través del host.
- *PREROUTING* — Altera los paquetes entrantes de red antes de ser enrutados.
- *POSTROUTING* — Altera los paquetes de redes antes de ser enviados.

Cada paquete de redes recibido por o enviado desde el sistema de Linux está sujeto al menos a una tabla. Sin embargo, el paquete puede estar sujeto a varias reglas dentro de cada tabla antes de emerger al final de la cadena. La estructura y propósito de dichas reglas pueden variar, pero suelen

tratar de identificar el paquete que ingresa o sale de una dirección IP determinada o de un grupo de direcciones, cuando usan un protocolo determinado y un servicio de redes. La siguiente imagen resume la forma como el subsistema de iptables examina el flujo de paquetes:



Nota

Las reglas de cortafuegos se guardan de forma predeterminada en los archivos **`/etc/sysconfig/iptables`** o **`/etc/sysconfig/ip6tables`**.

El servicio de **`iptables`** inicia antes de los servicios relacionados con DNS cuando se arranca el sistema de Linux. Esto significa que las reglas de cortafuegos solamente pueden hacer referencia a las direcciones IP numéricas (por ejemplo, 192.168.0.1). Nombres de dominio (por ejemplo, host.example.com) en dichas reglas se producen errores.

Independiente de su destino, cuando los paquetes coinciden con una regla determinada en una de las tablas, se les aplica un *destino* o acción. Si la regla especifica un destino **`ACCEPT`** para el paquete coincidente, el paquete omite la parte restante de las revisiones de la regla y puede continuar su destino. Si la regla especifica un destino **`DROP`**, ese paquete no podrá acceder al sistema y no se enviará nada al host que envío el paquete. Si la regla especifica un destino **`QUEUE`**, el paquete se pasa a espacio de usuario . Si la regla especifica el destino opcional de **`REJECT`**, el paquete es descargado, pero se envía al originador del paquete.

Cada cadena tiene una política predeterminada para **`ACCEPT`**, **`DROP`**, **`REJECT`**, o **`QUEUE`**. Si ninguna de estas reglas en la cadena se aplican al paquete, entonces el paquete es tratado de acuerdo con la política predeterminada.

El comando **`iptables`** configura estas tablas y establece las nuevas tablas si es necesario.

2.6.2. Opciones de comandos para IPTables

Las reglas para filtraje de paquetes se crean mediante el comando **`iptables`**. Los siguientes aspectos del paquete en su mayoría se utilizan como criterios:

- *Tipo de paquetes* — Especifica el tipo de filtros de comandos.
- *Destino u origen de paquetes* — Especifica los paquetes que filtra el comando basado en la fuente o el destino del paquete.
- *Destino* — Especifica la acción que se realiza en paquetes coincidente con los criterios mencionados.

Consulte la *Sección 2.6.2.4, "Opciones coincidentes de IPTables"* y la *Sección 2.6.2.5, "Opciones de destino"* para obtener mayor información sobre las opciones específicas que resuelven estos aspectos de un paquete.

Para que la regla sea válida, las opciones utilizadas con reglas específicas **iptables** deben agruparse localmente. La parte restante de esta sección explica las opciones utilizadas comúnmente para el comando **iptables**.

2.6.2.1. Estructura de opciones de comandos de IPTables

Muchos comandos **iptables** tienen la siguiente estructura:

```
iptables [-t <nombre-tabla>] <command> <nombre-cadena> \ <parámetro-1> <option-1>
\ <parámetro-n> <opción-n>
```

<*nombre-tabla*> — Especifica a qué tabla se aplica la regla. Si se omite, la tabla **filter** se utiliza.

<*comando*> — Especifica la acción a realizar, tal como añadir o borrar una regla.

<*nombre-cadena*> — Especifica la cadena a editar, crear o borrar.

Pares de <*parámetro*>-<*opción*> — Los parámetros y las opciones asociadas que especifican cómo procesar el paquete que coincide con la regla.

La longitud y complejidad de un comando **iptables** puede cambiar significativamente, según el propósito.

Por ejemplo, el comando para retirar una regla de una cadena puede ser muy corto:

iptables -D <*nombre-cadena*> <*númeroline-number*>

Por el contrario, un comando que agrega una regla que filtra los paquetes de una subred particular mediante un conjunto de parámetros específicos y opciones puede ser bastante largo. Al crear comandos de **iptables**, es importante recordar que algunos parámetros y opciones requieren otros parámetros y opciones para construir una regla válida. Esto puede producir un efecto en cascada, con los otros parámetros que requieren aún más parámetros. La regla no será válida, hasta que todos los parámetros y opciones que requieran otro conjunto de opciones hayan sido satisfechos.

Escriba **iptables -h** para ver una lista completa de estructuras de comandos **iptables**.

2.6.2.2. Opciones de comandos

Las opciones de comandos instruyen a los **iptables** para realizar una acción específica. Solamente una opción de comandos se permite por comando de **iptables**. A excepción del comando de ayuda, todos los comandos se escriben en mayúsculas.

Los comandos **iptables** son los siguientes:

- **-A** — Añade la regla al final de la cadena especificada. A diferencia de la opción **-I** descrita abajo, esta opción no lleva un argumento de entero. Siempre añade la regla al final de la cadena especificada.

- **-D <integer> | <rule>** — Borra una regla en una cadena determinada por número (tal como el número **5** para la quinta regla en la cadena) o por especificación de la regla. La especificación de la regla debe coincidir exactamente con una regla existente.

- **-E** — Renombra una cadena de usuario definido. Una cadena de usuario definido es una cadena en cualquier otra cadena diferente a la predeterminada, cadenas pre-existentes. (Consulte la opción **-N** a continuación, para obtener información sobre cómo crear cadenas de usuario definido.) Este es un cambio cosmético y no afecta la estructura de la tabla.

Nota

Si intenta renombrar una de las cadenas predeterminadas, el sistema reporta un error de **No se encuentra la coincidencia**. No pueden renombrar las cadenas predeterminadas.

- **-F** — Vacía la cadena seleccionada, la cual borra cada regla en la cadena. Si no se especifica, este comando vacía todas las reglas de todas las cadenas.

- **-h** — Proporciona una lista de estructuras de comandos, como también un breve resumen de parámetros de comandos y opciones.

- **-I [<integer>]** — Inserta la regla en una cadena especificada en el punto determinado por un argumento de entero de usuario definido. Si no se especifica ningún argumento, la regla se inserta en la parte superior de la cadena.

Importante

Como se anotó anteriormente, el orden de las reglas en una cadena determina las reglas que se aplican a cada paquete. Esto se debe tener presente al adicionar reglas mediante la opción **-A** o la opción **-I**.

Es de suma importancia añadir reglas mediante la opción **-I** con un argumento de entero. Si especifica un número existente al añadir una regla a una cadena, **iptables** añade la nueva regla *antes* (o por encima) de la regla existente.

- **-L** — Lista todas las reglas en una cadena especificada después del comando. Para listar todas las reglas en todas las cadenas en la tabla de **filter** predeterminada, no especifique la cadena o la tabla. De lo contrario, se deberá usar la siguiente sintaxis para listar las reglas en una cadena específica en una tabla particular:

```
iptables -L <nombre-cadena> -t <nombre-tabla>
```

Las opciones adicionales para la opción del comando **-L**, la cual proporciona números de regla y permite descripciones de regla más verbosas, se describen en la *Sección 2.6.2.6, “Listado de opciones”*.

- **-N** — Crea una nueva cadena con el nombre especificado por el usuario. El nombre de cadena debe ser único, de lo contrario se desplegará un mensaje de error.
- **-P** — Establece la política predeterminada para la cadena especificada, para que cuando los paquetes atraviesen toda la cadena sin coincidir con una regla, sean enviados al destino especificado, tal como ACCEPT o DROP.
- **-R** — Remplaza una regla en una cadena especificada. El número de la regla debe especificarse después del nombre de la cadena. La primer regla en una cadena corresponde a la regla número uno.
- **-X** — Borra una cadena especificada por el usuario. Usted no puede borrar una cadena incorporada.
- **-Z** — Establece los contadores de bytes y paquetes en todas las cadenas para una tabla a cero.

2.6.2.3. Opciones de parámetros de IPTables

Algunos comandos de **iptables**, entre ellos los utilizados para adicionar, adjuntar, borrar insertar, o remplazar reglas dentro de una cadena particular, requieren varios parámetros para construir una regla de filtraje de paquetes.

- **-c** — Restablece los contadores para una regla determinada. Este parámetro acepta las opciones **PKTS** y **BYTES** para especificar el contador que debe restablecerse.
- **-d** — Establece el nombre de host de destino, la dirección IP, o la red de un paquete que corresponda con la regla. Al coincidir con una red, los formatos de dirección IP o máscara de red son soportados:
  - ***N.N.N.N/M.M.M.M*** — Donde *N.N.N.N* es el rango de la dirección IP y *M.M.M.M* es la máscara de red.
  - ***N.N.N.N/M*** — Donde *N.N.N.N* es el rango de la dirección IP y *M* es la máscara de bits.
- **-f** — Aplica esta regla únicamente a paquetes fragmentados.

  Puede usar la opción del signo de exclamación (**!**) antes de este parámetro para especificar que únicamente paquetes des-fragmentados coinciden.

Nota

La distinción entre paquetes fragmentados y des-fragmentados es deseable, a pesar de que los paquetes fragmentados sean una parte estándar del protocolo IP.

En un principio fueron diseñados para viajar a través de las redes de tamaños diferentes, actualmente es más común usarlos para generar ataques DoS mediante paquetes mal construidos. Es también importante anotar que IPv6 no permite la fragmentación completa.

- **-i** — Establece la interfaz de redes, tal como **eth0** o **ppp0**. Con **iptables**, este parámetro opcional puede servir como cadenas de ENTRADA y REENVÍO cuando se usan con la tabla de **filter** y la cadena de PRE-ENRUTAMIENTO con las tablas **nat** y **mangle**.

  Este parámetro también soporta las siguientes opciones:

- Signo de exclamación (**!**) — Reversa la directiva, lo que significa que cualquier interfaz se excluirá de esta regla.
- El signo más (**+**) — Un comodín utilizado para hacer concordar todas las interfaces con la cadena especificada. Por ejemplo, el parámetro **-i eth+** aplicaría esta regla a todas las interfaces de Ethernet pero excluiría cualquier otra interfaz, tal como **ppp0**.

Si se utiliza el parámetro **-i** sin especificar la interfaz, entonces todas las interfaces se afectarán por la regla.

- **-j** — Salta al destino especificado cuando el paquete coincide con una regla determinada.

  Los destinos estándar son **ACCEPT**, **DROP**, **QUEUE** y **RETURN**.

  Las opciones extendidas también están disponibles a través de módulos cargados de forma predeterminada con el paquete RPM de **iptables** Red Hat Enterprise Linux. Los destinos válidos en estos módulos incluyen las opciones **LOG**, **MARK** y **REJECT**, entre otras. Consulte la página de manual **iptables** para obtener mayor información sobre otros destinos.

  Esta opción también sirve para dirigir el paquete coincidente con una regla determinada a una cadena definida por usuario fuera de la actual cadena para que otras reglas puedan aplicarse al paquete.

  Si no se especifica el destino, el paquete pasa la regla sin llevar a cabo ninguna acción. No obstante, el contador para esta regla aumenta en uno.

- **-o** — Establece la interfaz de red saliente para una regla. Esta opción es solamente válida para las cadenas de SALIDA Y REENVÍO en la tabla **filter**, y la cadena de POSTENRUTAMIENTO en las tablas **nat** y **mangle**. Este parámetro acepta las mismas opciones que las del parámetro de interfaz de redes (**-i**).
- **-p <protocol>** — Establece el protocolo IP afectado por la regla. Este protocolo puede ser **icmp**, **tcp**, **udp** o **all**, o también puede ser un valor numérico que represente alguno de estos protocolos o un protocolo diferente. También puede usar cualquiera de los protocolos listados en el archivo **/etc/protocols**.

  El protocolo "**all**" significa que la regla se aplica a cada protocolo soportado. Si no hay un protocolo listado con esta regla, se predeterminará el protocolo "**all**".

- **-s** — Establece el origen para un paquete determinado mediante la misma sintaxis del parámetro de destino (**-d**).

2.6.2.4. Opciones coincidentes de IPTables

Los protocolos de red diferentes proporcionan opciones coincidentes especializadas que pueden configurarse para coincidir con un determinado paquete mediante ese protocolo. Sin embargo, se debe especificar el protocolo en el comando de **iptables**. Por ejemplo, **-p *<nombre-protocolo>*** activa opciones para el protocolo especificado. Tenga en cuanta que puede usar el ID de protocolo, en lugar del nombre de protocolo. Consulte los siguientes ejemplos, cada uno de los cuales tiene el mismo efecto:

```
iptables -A INPUT -p icmp --icmp-type any -j ACCEPT
```

```
iptables -A INPUT -p 5813 --icmp-type any -j ACCEPT
```

Las definiciones de servicio se proporcionan en el archivo **/etc/services**. Para facilitar la lectura, se recomienda el uso de nombres de servicios en lugar de números de puertos.

Advertencia

Proteja el archivo **/etc/services** para evitar la modificación no autorizada. Si este archivo se puede editar, los ciberpiratas pueden usarlo para activar puertos en su máquina si no la ha cerrado. Para proteger este archivo, escriba los siguientes comandos como root:

```
[root@myServer ~]# chown root.root /etc/services
[root@myServer ~]# chmod 0644 /etc/services
[root@myServer ~]# chattr +i /etc/services
```

De esta manera se evita que se cambie de nombre, se borren o se creen enlaces al archivo.

2.6.2.4.1. Protocolo TCP

Estas opciones coincidentes están disponibles para el protocolo TCP (**-p tcp**):

- **--dport** — Establece el puerto de destino para el paquete.

  Para configurar esta opción, utilice el nombre del servicio de red (tal como www o smtp); un número de puerto; o un rango de números de puerto.

  Para especificar un rango de número de puertos, separe los dos números con dos puntos (**:**). Por ejemplo: **-p tcp --dport 3000:3200**. El rango más amplio que se acepta es **0:65535**.

  Use el signo de exclamación (**!**) antes de la opción **--dport** para hacer coincidir todos los paquetes que *no* usen ese servicio o puerto de red.

  Para explorar los nombres y alias de los servicios de red y los números de puerto que utilizan, consulte el archivo **/etc/services**.

  La opción coincidente **--destination-port** es sinónima de **--dport**.

- **--sport** — Establece el puerto de origen mediante las mismas opciones como **--dport**. La opción coincidente **--source-port** es sinónima de **--sport**.

- **--syn** — Se aplica a todos los paquetes TCP diseñados para iniciar la comunicación, conocidos comúnmente como *Paquetes SYN*. Los paquetes que llevan carga útil de datos no se tocan.

  Use un signo de exclamación (**!**) antes de la opción **--syn** para que coincida con todos los paquetes non-SYN.

- **--tcp-flags <tested flag list> <establece la lista de indicadores>** — Permite que los paquetes TCP que tienen establecidos bits específicos (indicadores), coincidan con una regla.

  La opción de coincidencia **--tcp-flags** acepta dos parámetros. El primer parámetro es la máscara; una lista de indicadores separados por coma que van a ser examinados en el paquete. El segundo paquete es una lista de indicadores separados por coma que se deben establecer para que la regla coincida.

  Los indicadores posibles son:

- **ACK**
- **FIN**
- **PSH**
- **RST**
- **SYN**
- **URG**
- **ALL**
- **NONE**

Por ejemplo, una regla de **iptables** que contiene la siguiente especificación solamente concuerda con los paquetes TCP que tienen el indicador SYN y los indicadores de ACK y FIN no configurados:

```
--tcp-flags ACK,FIN,SYN SYN
```

Use el signo de exclamación (**!**) antes de **--tcp-flags** para reversar el efecto de la opción de coincidencia.

- **--tcp-option** — Intenta coincidir con opciones específicas de TCP que puedan establecerse dentro de un paquete determinado. Esta opción de coincidencia también se puede reversar con el signo de exclamación (**!**).

2.6.2.4.2. Protocolo UDP

Estas opciones de coincidencia están disponibles para el protocolo UDP (**-p udp**):

- **--dport** — Especifica el puerto de destino del paquete UDP, mediante el nombre del servicio, el número de puerto o el rango de número de puertos. La opción de coincidencia **--destination-port** es sinónima de **--dport**.
- **--sport** — Especifica el puerto de origen del paquete UDP, mediante el nombre de servicio, el número de puerto o el rango de números de puerto. La opción **--source-port** es sinónima de **--sport**.

Respecto a las opciones **--dport** y **--sport**, para especificar el rango de números de puerto, separe los dos números con dos puntos (:). Por ejemplo: **-p tcp --dport 3000:3200**. El rango más amplio aceptable es 0:65535.

2.6.2.4.3. Protocolo ICMP

Las opciones a continuación están disponibles para el Protocolo de mensajes de Internet (ICMP) (**-p icmp**):

- **--icmp-type** — Establece el nombre o número de tipo ICMP para que coincida con la regla. Para obtener una lista de nombres de ICMP válidos, escriba el comando **iptables -p icmp -h**.

2.6.2.4.4. Módulos de opciones de coincidencia adicionales

Las opciones de coincidencia adicionales están disponibles a través de módulos cargados por el comando **iptables**.

Para usar un módulo de opción de coincidencia, cargue el módulo por nombre mediante **-m <*nombre-módulo*>**, donde <*nombre-módulo*> es el nombre del módulo.

Muchos módulos están disponibles de forma predeterminada. También puede crear módulos para proporcionar funcionalidades adicionales.

La siguiente es una lista parcial de los módulos más comúnmente usados:

- **limit** module — Coloca los límites en la cantidad de paquetes que coinciden con una regla determinada.

  Cuando se utiliza junto con el destino de **LOG**, el módulo **limit** puede evitar que el flujo de paquetes coincidentes llenen el registro del sistema con mensajes repetitivos o agoten todos los recursos del sistema.

  Consulte la *Sección 2.6.2.5, "Opciones de destino"* para obtener mayor información sobre el destino de **LOG**.

  El módulo **limit** permite las siguientes opciones:

  - **--limit** — Establece el número máximo para un periodo de tiempo determinado, especificado como un par de **<*valor*>/<*periodo*>**. Por ejemplo, el uso de **--limit 5/hour** permite cinco coincidencias de regla por hora.

    Los periodos pueden ser en segundos, minutos, horas o días.

    Si el modificador de número y tiempo no se utiliza, se toma el valor predeterminado **3/hour**.

  - **--limit-burst** —Establece un límite en el número de paquetes que pueden coincidir con una regla a la vez.

    Esta opción se especifica como un entero y debe utilizarse junto con la opción **--limit**.

    Si no se especifica el valor, se toma el valor predeterminado de cinco (5).

- Módulo de **state** — Permite coincidencias de estado.

  El módulo **state** permite las siguientes opciones:

  - **--state** — corresponde a un paquete con los siguientes estados de conexión:

    - **ESTABLISHED** — El paquete coincidente con otros paquetes en una conexión establecida. Necesita aceptar este estado para mantener una conexión entre el cliente y el servidor.

    - **INVALID** — El paquete coincidente no puede conectarse a una conexión conocida.

    - **NEW** — El paquete coincidente crea una nueva conexión o hace parte de una conexión de dos vías no vista anteriormente. Necesita aceptar este estado si desea permitir nuevas conexiones para el servicio.

    - **RELATED** — El paquete coincidente inicia una nueva conexión relacionada de alguna forma con una conexión existente. Un ejemplo de ella es el FTP, el cual usa una conexión para controlar el tráfico (puerto 21), y una conexión independiente para transferir los datos (puerto 20).

    Estos estados de conexión pueden usarse en combinación con otros al separarlos con comas, tal como **-m state --state INVALID,NEW**.

- **mac** module — Permite la concordancia entre direcciones MAC de hardware.

El módulo **mac** permite la siguiente opción:

- **--mac-source** — Coincide con una dirección MAC de la tarjeta de interfaz de red que envió el paquete. Para excluir una dirección MAC de una regla, coloque un signo de exclamación (**!**) antes de la opción coincidente **--mac-source**.

Consulte la página de manual **iptables** para obtener más opciones de coincidencia disponibles a través de los módulos.

2.6.2.5. Opciones de destino

Cuando un paquete ha coincidido con una determinada regla, la regla puede dirigir el paquete a un número de destinos diferentes los cuales determinan la acción apropiada. Cada cadena tiene un destino predeterminado, el cual se utiliza si ninguna de las reglas en esa cadena concuerda con un paquete o si ninguna de las reglas que coinciden con el paquete especifican el destino.

A siguientes son los estándares de destino:

- ***<user-defined-chain>*** — Una cadena definida por usuario dentro de la tabla. Los nombres de cadena definidos por usuario deben ser únicos. Este destino pasa al paquete de la cadena especificada.
- **ACCEPT** — Acepta que el paquete continúe a su destino o a otra cadena.
- **DROP** — elimina el paquete sin responder al solicitante. El sistema que envió el paquete no es notificado sobre la falla.
- **QUEUE** — El paquete está en la cola para ser manejado por una aplicación de espacio de usuario.
- **RETURN** — Para de comparar el paquete con las reglas en la cadena actual. Si el paquete con un destino de **RETURN** concuerda con una regla en una cadena llamada desde otra cadena, el paquete retorna a la primera cadena para continuar la revisión de la regla donde había quedado. Si la regla de **RETURN** se utiliza en una cadena incorporada y el paquete no se puede desplazar a la cadena anterior, se utilizará el destino predeterminado para la cadena actual.

Además, las extensiones que están disponibles permiten especificar otros destinos. Dichas extensiones se denominan módulos de destino o módulos de opciones coincidentes y la mayoría se aplican a tablas y situaciones específicas. Consulte *Sección 2.6.2.4.4, “Módulos de opciones de coincidencia adicionales”* para obtener mayor información sobre módulos de opciones coincidentes.

Existen muchos módulos de destino extendido, la mayoría de los cuales solamente se aplican a tablas o situaciones específicas. Algunos de los módulos de destino más populares incluidos de forma predeterminada en Red Hat Enterprise Linux son:

- **LOG** — Registra todos los paquetes coincidentes con esta regla. Puesto que el kernel registra los paquetes, el archivo **/etc/syslog.conf** determina el sitio en donde se escriben estos registros. Los registros se sitúan de forma predeterminada en el archivo **/var/log/messages**.

  Se pueden usar las opciones adicionales después del destino **LOG** para especificar la forma en la que el registro se presenta:

  - **--log-level** — Establece el nivel de prioridad de un evento de registro. Consulte la página de manual **syslog.conf** para obtener una lista de los niveles de prioridad.
  - **--log-ip-options** — Registra las opciones establecidas en el encabezamiento del paquete IP.

- **`--log-prefix`** — Sitúa una cadena de hasta 29 caracteres antes de la línea de registro cuando se escribe. Esto es útil para escritura de filtraje de syslog para usar junto con el registro de paquetes.

Nota

Debido a un problema con esta opción, debe añadir un espacio final al valor *`log-prefix`*.

- **`--log-tcp-options`** — Registra las opciones establecidas en el encabezamiento de un paquete TCP.
- **`--log-tcp-sequence`** — Escribe el número de secuencia TCP para el paquete en el registro.

- **`REJECT`** — Devuelve un paquete de errores al sistema remoto y elimina el paquete.

  El destino **`REJECT`** acepta **`--reject-with <tipo>`** (donde *<tipo>* es el tipo de rechazo) y permite una información más detallada que va a ser devuelta con el paquete de errores. El mensaje **`port-unreachable`** es el tipo de error predeterminado si no se utiliza otra opción. Consulte la página de manual de **`iptables`** para obtener una lista completa de opciones **`<type>`**.

Otras extensiones de destino, incluyen varias que son útiles para enmascaramiento de IP mediante la tabla **`nat`**o con la alteración de paquetes mediante la tabla **`mangle`**, se puede hallar en la página de manual de **`iptables`**.

2.6.2.6. Listado de opciones

El comando de la lista predeterminada, **`iptables -L [<chain-name>]`**, proporciona una visión general básica de las cadenas actuales de la tabla de filtros predeterminadas. Opciones adicionales proporcionan mayor información:

- **`-v`** — Despliega una salida verbosa, tal como el número de paquetes y bytes que cada cadena ha procesado, el número de paquetes que cada cadena ha coincidido y las interfaces que se aplican a una determinada regla.
- **`-x`** — Extiende los números a sus valores exactos. En un sistema ocupado, el número de paquetes y bytes procesados por una cadena determinada o regla puede abreviarse a **`Kilobytes`**, **`Megabytes`** (Megabytes) o **`Gigabytes`**. Esta opción fuerza a que el número completo sea desplegado.
- **`-n`** — Despliega las direcciones IP y números de puertos en formato numérico, en lugar del formato predeterminado nombre de host y servicio de red.
- **`--line-numbers`** — Lista las reglas en cada cadena cerca del orden numérico en la cadena. Esta opción sirve para cuando se intenta borrar la regla específica en al cadena o localizar dónde insertar la regla dentro de una cadena.
- **`-t <table-name>`** — Especifica un nombre de tabla. Si se omite, se predetermina a la tabla de filtro.

2.6.3. Cómo guardar reglas de IPTables

Las reglas creadas con el comando **iptables** se almacenan en la memoria. Si el sistema se inicia antes de guardar el conjunto de reglas **iptables**, todas las reglas se perderán. Para que las reglas de netfilter persistan a través del re-arranque del sistema, es necesario guardarlas. Para guardar las reglas de netfilter, escriba el siguiente comando como root:

```
/sbin/service iptables save
```

Este comando, ejecuta el script init de **iptables**, el cual ejecuta el programa **/sbin/iptables-save** y escribe la configuración actual de **iptables** a **/etc/sysconfig/iptables**. El archivo existente **/etc/sysconfig/iptables** se guarda como **/etc/sysconfig/iptables.save**.

La próxima vez que el sistema arranque , el script de init **iptables** reaplica las reglas guardas en **/etc/sysconfig/iptables** mediante el comando **/sbin/iptables-restore**.

Aunque es siempre una buena idea probar una nueva regla de **iptables** antes de enviarla al archivo **/etc/sysconfig/iptables**, es posible copiar las reglas de **iptables** dentro de este archivo desde otra versión del archivo. Así, se provee una forma rápida de distribuir los conjuntos de reglas de **iptables** a varias máquinas.

También puede guardar las reglas de IPTables en un archivo independiente para distribución, copia de seguridad u otros propósitos. Para almacenar las reglas de IPTables, escriba el siguiente comando como root:

```
[root@myServer ~]# iptables-save > <nombredearchivo>donde <filename> es el nombre definido por usuario para su conjunto de reglas.
```

Importante

Para distribuir el archivo **/etc/sysconfig/iptables** a otras máquinas, escriba **/sbin/service iptables restart** para que las nuevas reglas se efectúen.

Nota

Observe la diferencia entre **iptables** *command* (**/sbin/iptables**), el cual se utiliza para manipular las tablas y cadenas que constituyen la funcionalidad de **iptables** e **iptables** *service* (**/sbin/service iptables**), el cual se utiliza para habilitar o inhabilitar el servicio **iptables**.

2.6.4. Scripts de control de IPTables

Existen dos métodos básicos para controlar **iptables** en Red Hat Enterprise Linux:

- **Firewall Configuration Tool** (**system-config-firewall**) — Una interfaz gráfica para crear, activar y guardar reglas de cortafuegos básicas. Consulte la *Sección 2.5.2, "Configuración básica de cortafuegos"* para obtener mayor información.

- **/sbin/service iptables <option>** — Se utiliza para manipular varias funciones de **iptables** mediante el initscript. Las siguientes opciones están disponibles:

- **start** — Si un cortafuegos está configurado (es decir, **/etc/sysconfig/iptables** existe), todos los **iptables** que se estén ejecutando se detendrán completamente y luego se iniciarán mediante el comando **/sbin/iptables-restore**. Esta opción solamente funciona si el módulo de kernel **ipchains** se carga, escriba el siguiente comando como root:

```
[root@MyServer ~]# lsmod | grep ipchains
```

  Si este comando no retorna ninguna salida, significa que el módulo no está cargado. Si es necesario, utilice el comando **/sbin/rmmod** para retirar el módulo.

- **stop** — Si el cortafuegos está en ejecución, las reglas de cortafuegos en memoria se vacían y todos los módulos de iptables y asistentes se descargan.

  Si la directiva **IPTABLES_SAVE_ON_STOP** en el archivo de configuración **/etc/sysconfig/iptables-config** se cambia de su valor predeterminado **yes**, las reglas actuales se guardarán en **/etc/sysconfig/iptables** y las reglas existentes se desplazan al archivo **/etc/sysconfig/iptables.save**.

  Consulte la *Sección 2.6.4.1, “Archivo de configuración de scripts de control de IPTables”* para obtener mayor información sobre el archivo **iptables-config**.

- **restart** — Si un cortafuegos está en ejecución, las reglas de cortafuegos en memoria se eliminan, y el cortafuegos se reinicia si está configurado en **/etc/sysconfig/iptables**. Esta opción solamente funciona si el módulo de kernel **ipchains** no está cargado.

  Si la directiva **IPTABLES_SAVE_ON_RESTART** en el archivo de configuración **/etc/sysconfig/iptables-config** cambia de su valor predeterminado a **yes**, las reglas actuales se guardan en **/etc/sysconfig/iptables** y las reglas existentes se desplazan al archivo **/etc/sysconfig/iptables.save**.

  Consulte la *Sección 2.6.4.1, “Archivo de configuración de scripts de control de IPTables”* para obtener mayor información sobre el archivo **iptables-config**.

- **status** — Muestra el estatus del cortafuegos y lista todas las reglas activas.

  La configuración predeterminada para esta opción muestra las direcciones IP en cada regla. Para desplegar la información sobre el dominio y el nombre de host, modifique el archivo **/etc/sysconfig/iptables-config** y cambie el valor de **IPTABLES_STATUS_NUMERIC** a **no**. Consulte la *Sección 2.6.4.1, “Archivo de configuración de scripts de control de IPTables”* para obtener mayor información sobre el archivo de **iptables-config**.

- **panic** — Vacía todas las reglas. La política de todas las tablas configuradas se establece a **DROP**.

  Esta opción podría ser útil si se sabe que el servidor está comprometido. En lugar de desconectarlo físicamente desde la red o de apagar el sistema, puede usar esta opción para detener el tráfico de red posterior y dejar a la máquina lista para el análisis u otros exámenes forenses.

- **save** — Guarda las reglas de cortafuegos para **/etc/sysconfig/iptables** mediante **iptables-save**. Consulte la *Sección 2.6.3, “Cómo guardar reglas de IPTables”* para obtener mayor información.

Nota

Para usar los mismos comandos initscript para controlar netfilter para IPv6, substituya **ip6tables** por **iptables** en los comandos **/sbin/service** listados en esta sección. Para obtener mayor información sobre IPv6 y netfilter, consulte la *Sección 2.6.5, "IPTables e IPv6"*.

2.6.4.1. Archivo de configuración de scripts de control de IPTables

La conducta de los initscripts de **iptables** es controlada por los archivos de configuración **/etc/sysconfig/iptables-config**. La siguiente es una lista de directivas contenidas en este archivo:

- **IPTABLES_MODULES** — Especifica una lista de módulos de **iptables**, separada por espacios, para cargar cuando un cortafuegos esté activo. Estos módulos pueden incluir rastreo de conexiones y asistentes NAT.
- **IPTABLES_MODULES_UNLOAD** — Descarga módulos en el reinicio y se detiene. Esta directiva acepta los siguientes valores:
  - **yes** — Es el valor predeterminado. Esta opción debe configurarse para establecer un estado correcto para que un cortafuegos reinicie o se detenga.
  - **no** — Esta opción se debe establecer únicamente si hay problemas con la descarga de módulos de netfilter.
- **IPTABLES_SAVE_ON_STOP** — Guarda las reglas de cortafuegos actuales para **/etc/sysconfig/iptables** cuando el cortafuegos se detiene. Esta directiva acepta los siguientes valores:
  - **yes** — Guarda las reglas existentes para **/etc/sysconfig/iptables** cuando el cortafuegos se detiene, se desplaza la versión anterior a la del archivo **/etc/sysconfig/iptables.save**. \n
  - **no** — El valor predeterminado. No guarda las reglas existentes cuando se detiene el cortafuegos.
- **IPTABLES_SAVE_ON_RESTART** — Guarda las reglas de cortafuegos actuales cuando se reinicia el cortafuegos. Esta directiva acepta los siguientes valores:
  - **yes** — Guarda las reglas existentes en **/etc/sysconfig/iptables** cuando se reinicia el cortafuegos, desplazando la versión anterior del archivo **/etc/sysconfig/iptables.save**.
  - **no** — El valor predeterminado. No guarda reglas cuando se reinicia el cortafuegos.
- **IPTABLES_SAVE_COUNTER** — Guarda y restaura todos los paquetes y contadores de bytes en todas las cadenas y reglas. Esta directiva acepta los siguientes valores:
  - **yes** — Guarda los valores de contador.
  - **no** — Es el valor predeterminado. No guarda los valores de contador.
- **IPTABLES_STATUS_NUMERIC** — Direcciones IP de salida en forma numérica en lugar de dominio o nombres de host. Esta directiva acepta los siguientes valores:
  - **yes** — Es el valor predeterminado. Retorna únicamente direcciones IP dentro de una salida de estatus.

- **no** — Retorna el dominio o nombres de host dentro de una salida de estatus.

2.6.5. IPTables e IPv6

Si el paquete **iptables-ipv6** está instalado, netfilter en Red Hat Enterprise Linux puede filtrar la siguiente generación de protocolo de Internet IPv6. El comando utilizado para manipular el netfilter IPv6 es **ip6tables**.

La mayoría de directivas para este comando son idénticas a las utilizadas por **iptables**, excepto la tabla **nat** que aún no es compatible. Es decir que aún no es posible realizar tareas de traducción de direcciones de red IPv6, tales como enmascaramiento y reenvío de puertos.

La reglas para **ip6tables** se almacenan en el archivo **/etc/sysconfig/ip6tables**. Las reglas anteriores guardadas por initscripts **ip6tables** se guardan en el archivo **/etc/sysconfig/ip6tables.save**.

Las opciones de configuración para script init de **ip6tables** se almacenan en **/etc/sysconfig/ip6tables-config** y los nombres de cada directiva varían levemente de sus homólogos de **iptables**.

Por ejemplo, para la directiva de **iptables-config IPTABLES_MODULES**:la equivalente en el archivo **ip6tables-config** es **IP6TABLES_MODULES**.

2.6.6. Recursos adicionales

Consulte los siguientes recursos para obtener información adicional sobre el filtraje de paquetes con **iptables**.

- *Sección 2.5, “Cortafuegos”* — Contiene un capítulo sobre el rol de los cortafuegos dentro de una estrategia general de seguridad como también estrategias para construir reglas de cortafuegos.

2.6.6.1. Documentación de tablas IP instaladas

- **man iptables** — Contiene una descripción de **iptables** como también una lista completa de destinos, opciones y extensiones de coincidencia.

2.6.6.2. Sitios web útiles de tablas IP

- *http://www.netfilter.org/* — El hogar del proyecto netfilter o iptables. Contiene información variada sobre **iptables**, entre ellas Preguntas frecuentes que resuelven problemas específicos y varias guías útiles de Rusty Russell, el mantenedor de cortafuegos de Linux IP. Los documentos de CÓMO en el sitio cubren temas tales como conceptos básicos de redes, filtraje de paquetes de kernel y configuraciones de NAT.

- *http://www.linuxnewbie.org/nhf/Security/IPtables_Basics.html* — Una introducción a la forma como los paquetes se desplazan a través del kernel de Linux, más una introducción a la creación de comandos básicos de **iptables**.

Cifrado

Hay dos tipos principales de datos que deben estar protegidos: los datos quietos y los datos en movimiento. Estos dos tipos de datos están protegidos en formas similares mediante una tecnología similar, aunque la implementación puede ser completamente diferente. Ninguna implementación protectora puede evitar todos los métodos posibles de compromiso ya que la misma información puede estar quieta o en movimiento en diferentes momentos.

3.1. Datos quietos

Datos quietos son los datos almacenados en el disco duro, cinta, CD, DVD, disco, u otro medio. La mayor amenaza informática surge del robo físico. Portátiles en aeropuertos, CD que se envían por correo y cintas de seguridad que se dejan en lugares errados son todos los ejemplos de eventos en los que los datos pueden verse comprometidos por robo. Si los datos estaban cifrados en los medios entonces no habría por qué preocuparse tanto de que la información pueda comprometerse.

3.2. Cifrado total de disco

El disco completo o el cifrado de partición es una de las mejores formas de proteger sus datos. No solamente cada archivo está protegido sino también el almacenamiento temporal que pueda contener partes de estos archivos también lo está. El cifrado de todo el disco protegerá todos sus archivos, por lo tanto no tiene que preocuparse por seleccionar lo que desea proteger y posiblemente perder un archivo.

Red Hat Enterprise Linux 6 soporta el cifrado de LUKS. LUKS cifrará en volumen sus particiones de disco duro para que cuando su computador esté apagado sus datos estén protegidos. Así también protegerá a su computador de personas que intenten usar el modo único para ingresar a su computador o acceder de otra forma.

Las soluciones de cifrado de disco total como LUKS solamente protegen datos cuando su computador está apagado. Una vez que el computador esté encendido y LUKS haya descifrado el disco, los archivos en ese disco están disponibles para cualquier persona que desee tener acceso normal a ellos. Para proteger sus archivos cuando el computador esté encendido, use el cifrado de disco total en combinación con otra solución tal como cifrado basado en archivo. También recuerde bloquear su computador cada vez que esté lejos de él. Una pantalla protegida con frase de paso establecida para que se activa después de algunos minutos de inactividad es una buena forma de mantener lejos a los intrusos.

3.3. Cifrado basado en archivos

GnuPG (GPG) es una versión de fuente abierta de PGP que permite firmar y o cifrar un archivo o un mensaje-e. Esto es útil para mantener la integridad del mensaje o archivo y también para proteger la confidencialidad de la información contenida en el archivo o correo-e, GPG proporciona doble protección. No solamente puede proporcionar protección de Datos quietos, sino también protección de datos en movimiento una vez que el mensaje ha sido enviado a través de la red.

El archivo basado en cifrado protege un archivo después de que ha dejado su computador, tal como cuando envía un CD por correo. Algunas soluciones de cifrado de archivos dejan remanentes de los archivos cifrados que un atacante que tiene acceso físico a su computador puede recuperar bajo algunas circunstancias. Para proteger el contenido de esos archivos de los atacantes que pueden acceder a su computador, use el archivo basado en cifrado combinado con otra solución tal como un cifrado total de disco.

3.4. Datos en movimiento

Los datos en movimiento son los datos que se están transmitiendo en la red. La mayor amenaza para los datos en movimiento es la interceptación y su alteración. Su nombre de usuario y contraseña nunca se deben transmitir por la red sin protección, ya que puede ser interceptada y utilizada por alguien que se haga pasar por usted o pueda acceder a información confidencial. La información privada como la información de una cuenta bancaria se debe también proteger cuando se transmite por la red. Si la sesión de la red estaba cifrada entonces no tendrá que preocuparse tanto sobre que los datos se hayan visto comprometidos cuando se transmitían.

Los datos en movimiento son particularmente vulnerables porque el atacante no tiene que estar cerca del computador en el cual están los datos que se están almacenando, solamente necesita estar en alguna parte de la ruta. Los túneles de cifrado pueden proteger los datos junto con la ruta de comunicaciones.

3.5. Redes virtuales privadas

Las Redes virtuales privadas (VPN) proporcionan túneles cifrados entre computadores y redes de computadores a través de todos los puertos. Con VPN en su sitio, todo el tráfico de redes desde el cliente es reenviado al servidor a través del túnel cifrado. Es decir, que el cliente está lógicamente en la misma red a la que el servidor está conectado vía VPN. Las VPN son muy comunes, fáciles de usar y de configurar.

3.6. Shell segura

Shell segura (SSH) es un protocolo de redes poderoso que sirve para comunicarse con otro sistema en un canal seguro. Las transmisiones mediante SSH están cifradas y protegidas de interceptaciones. El registro criptográfico también puede utilizarse para proporcionar un mejor método de autenticación que los nombres de usuarios y contraseñas tradicionales.

SSH es muy fácil de activar. Simplemente inicie el servicio sshd y el sistema comenzará a aceptar conexiones cuando se proporcione un nombre de usuario y contraseña correctos durante el proceso de conexión. El puerto TCP estándar para el servicio SSH es 22, sin embargo puede cambiarse si modifica el archivo de configuración */etc/ssh/sshd_config* y reinicia el servicio. Este archivo contiene otras opciones de configuración para SSH.

La shell segura (SSH) también proporciona túneles cifrados entre computadores pero solo mediante un puerto único. *El reenvío de puerto puede hacerse en un túnel SSH*[1]y el tráfico será cifrado cuando pase por ese túnel aunque mediante reenvío de puerto no es tan fluido como un VPN.

3.7. Motor OpenSSL PadLock

El motor de VIA PadLock está disponible en los mismos procesadores VIA C3 (Nehemia), y permite hardware extremadamente rápido y cifrado y decifrado.

[1] http://www.redhatmagazine.com/2007/11/27/advanced-ssh-configuration-and-tunneling-we-dont-need-no-stinking-vpn-software

Nota

No hay soporte para VIA Padlock en sistemas de 64 bits.

Para activarlo, edite **`/etc/pki/tls/openssl.cnf`** y añada lo siguiente al comienzo del archivo:

```
openssl_conf = openssl_init
```

Luego añada lo siguiente al final del archivo:

```
[openssl_init]
engines = openssl_engines

[openssl_engines]
padlock = padlock_engine

[padlock_engine]
default_algorithms = ALL
dynamic_path = /usr/lib/openssl/engines/libpadlock.so
init = 1
```

Para verificar si el módulo está activado, ejecute el siguiente comando:

```
# openssl engine -c -tt
```

Para probar su velocidad, ejecute el siguiente comando:

```
# openssl speed aes-128-cbc
```

Para probar la velocidad de OpenSSH puede ejecutar un comando como el siguiente:

```
# dd if=/dev/zero count=100 bs=1M | ssh -c aes128-cbc
localhost "cat >/dev/null"
```

Encontrará más información sobre el motor VIA PadLock en las siguientes URL: *http://www.logix.cz/michal/devel/padlock/* y *http://www.via.com.tw/en/initiatives/padlock/*.

3.8. Cifrado de disco LUKS

La configuración de la llave unificada de Linux en formato de disco (o LUKS) le permite cifrar particiones en su equipo de Linux. Es principalmente importante cuando se trata de equipos móviles y soportes extraíbles. LUKS permite a varios usuarios descifrar una llave maestra utilizada para cifrado masivo de la partición.

3.8.1. Implementación de LUKS en Red Hat Enterprise Linux

Red Hat Enterprise Linux 6 utiliza a LUKS para realizar el cifrado de sistema de archivos. Por defecto, la opción para cifrar el sistema de archivos se desactiva durante la instalación. Si selecciona la opción para cifrar el disco duro, se le solicitará una frase de paso que se le pedirá cada vez que arranque el equipo. Esta frase de paso "desbloquea" la llave de cifrado utilizada para descifrar su partición. Si

elige modificar la tabla de partición predeterminada puede elegir las particiones que desea cifrar. Esto se establece en la configuración de la tabla de particiones.

La cifra predeterminada utilizada por LUKS (consulte **`cryptsetup --help`**) es aes-cbc-essiv:sha256 (ESSIV - Encrypted Salt-Sector Initialization Vector). Observe que el programa de instalación, **Anaconda**, emplea de forma predeterminada el modo XTS (aes-xts-plain64). El tamaño de la llave para LUKS es de 256 bits. El tamaño de llave predeterminada para LUKS con **Anaconda** (modo XTS) es de 512 bits. Las cifras que están disponibles son:

- AES - Estándar de cifrado avanzado - *FIPS PUB 197*[2]
- Twofish (un cifrado de bloque de 128 bits)
- Serpent
- cast5 - *RFC 2144*[3]
- cast6 - *RFC 2612*[4]

3.8.2. Directorios de cifrado manual

Advertencia

Al seguir este procedimiento, retirará todos los datos en la partición que está cifrando. ¡PERDERÁ toda la información! Asegúrese de hacer una copia de seguridad en una fuente externa antes de iniciar este procedimiento.

3.8.3. Instrucciones paso a paso

1. Ingrese el nivel de ejecución 1: **`telinit 1`**
2. Desmonte su /home actual: **`umount /home`**
3. Si esto falla utilice **`fuser`** para buscar y matar todos los procesos que acaparan a /home: **`fuser -mvk /home`**
4. Verifique si /home ya no está montado: **`cat /proc/mounts | grep home`**
5. Llene su partición con datos aleatorios: **`dd if=/dev/urandom of=/dev/VG00/LV_home`** Este proceso tarda varias horas en completar.

   Importante

   Sin embargo, el proceso es imperativo para tener una buena protección contra intentos de ingreso. Deje que se ejecute durante la noche.

6. Inicie la partición: **`cryptsetup --verbose --verify-passphrase luksFormat /dev/VG00/LV_home`**

7. Abra el dispositivo recién cifrado: **`cryptsetup luksOpen /dev/VG00/LV_home home`**
8. Verifique si está ahí: **`ls -l /dev/mapper | grep home`**
9. Cree un sistema de archivos: **`mkfs.ext3 /dev/mapper/home`**
10. Móntelo: **`mount /dev/mapper/home /home`**
11. Revise si está visible: **`df -h | grep home`**
12. Añada lo siguiente a /etc/crypttab: **`home /dev/VG00/LV_home none`**
13. Modifique su /etc/fstab, retire la entrada anterior para /home y añada **`/dev/mapper/home / home ext3 defaults 1 2`**
14. Restaure los contextos predeterminados de seguridad de SELinux: **`/sbin/restorecon -v -R /home`**
15. Reinicie: **`shutdown -r now`**
16. La entrada en /etc/crypttab hace que su equipo pregunte la contraseña de **`luks`** en el arranque
17. Ingrese como root y restaure su copia de seguridad

3.8.4. ¿Qué ha logrado?

¡Felicidades!, ahora tiene una partición cifrada para que todos sus datos estén protegidos cuando el equipo esté apagado.

3.8.5. Enlaces de interés

Para obtener información adicional sobre LUKS o cifrado de discos duros en Red Hat Enterprise Linux visite los siguientes enlaces:

- *LUKS home page*[5]
- *LUKS/cryptsetup FAQ*[6]
- *LUKS - Linux Unified Key Setup*[7]
- *HOWTO: Creating an encrypted Physical Volume (PV) using a second hard drive and pvmove*[8]

3.9. Uso del Guarda de privacidad GNU (GNUPG)

GPG le sirve para identificarse y autenticar sus comunicaciones, entre ellas aquellas que usted no conoce. GPG le permite a cualquier persona leer un correo-e firmado con GPG para verificar su autenticidad. En otras palabras, GPG le permite a alguien estar seguro de que las comunicaciones firmadas por usted han sido en realidad suyas. GPG es útil porque evita que terceros alteren su código o intercepten comunicaciones y alteren el mensaje.

3.9.1. Creación de llaves GPG in GNOME

Instale la herramienta Seahorse, la cual facilita la administración de la llave GPG. Desde el menú principal, seleccione **`Administración del sistema> Administración > Añadir/Eliminar software`** y espere el PackageKit para iniciar. Ingrese **`Seahorse`** en la cajilla de texto y seleccione buscar. Seleccione la cajilla de verificación cerca del paquete "seahorse" y seleccione "Aplicar" para

añadir el software. Puede instalar **Seahorse** en la línea de comandos con el comando **su -c "yum install seahorse"**

Para crear una llave, desde el menú "Aplicaciones > Accesorios" seleccione "Contraseñas y llaves cifradas", lo cual inicia la aplicación **Seahorse**. Desde el menú de "Archivo" seleccione "Nueva" luego "llave PGP". Después haga clic en "Continuar". Escriba su nombre completo, dirección de correo-e y un comentario opcional que describa quién es usted (por ejemplo.: John C. Smith, jsmith@example.com, The Man). Haga clic en "Crear". Se desplegará un diálogo que le preguntará su frase de paso para la llave. Elija una contraseña poderosa pero a la vez fácil de recordar. Haga clic en "Aceptar" y la llave se creará.

Advertencia

Si olvidó la frase de paso, no se puede usar la clave y los datos cifrados que la usan se perderán.

Para encontrar el ID de llave GPG, busque en la columna "ID de llave" cerca de la llave recién creada. En la mayoría de los casos, si se le pide el ID de la llave, usted debe anteponer "0x" al ID de la llave, como en "0x6789ABCD". Debe hacer una copia de respaldo de su llave privada y almacenarla en un lugar seguro.

3.9.2. Creación de llaves GPG en KDE

Inicie el programa KGpg desde el menú principal al seleccionar Aplicaciones > Herramientas > Herramienta de cifrado. Si nunca ha utilizado KGpg, el programa lo guiará a lo largo del proceso de creación de su propio par de llaves GPG. Un cuadro de diálogo aparecerá solicitándole la creación de un nuevo par de llaves. Ingrese su nombre, dirección de correo-e y un comentario adicional. También puede elegir la fecha de expiración para su llave, como también la fortaleza de la llave (número de bits) y algoritmos. El siguiente cuadro de diálogo le solicitará su contraseña. En este momento su llave aparecerá en la pantalla principal de **KGpg**.

Advertencia

Si olvidó la frase de paso, no se puede usar la clave y los datos cifrados que la usan se perderán.

Para encontrar el ID de llave GPG, busque en la columna "ID de llave" cerca de la llave recién creada. En la mayoría de los casos, si se le pide el ID de la llave, usted debe anteponer "0x" al ID de la llave, como en "0x6789ABCD". Debe hacer una copia de respaldo de su llave privada y almacenarla en un lugar seguro.

3.9.3. Creación de llaves GPG mediante la línea de comandos

Use el siguiente comando de shell: **gpg --gen-key**

Este comando genera un par de llaves que consta de una llave pública y una privada. Otras personas usan la llave pública para autenticar y descifrar sus comunicaciones. Distribuya su llave pública tanto

como sea posible, especialmente a las personas a quienes conoce y desean recibir comunicaciones autenticadas de su parte, como por ejemplo la lista de correos.

Una serie de indicadores se dirige a usted a través del proceso. Presione la tecla **Enter** para asignar el valor predeterminado si se desea. El primer indicador le pide que seleccione la clase de llave que usted prefiere.

Por favor ¿qué clase de llave desea: (1) DSA y ElGamal (predeterminada) (2) DSA (solo firma) (3) RSA (solo firma) ¿Su elección? En la mayoría de los casos, la predeterminada es la elección correcta. Una llave DSA/ElGamal le permite no solo firmar comunicaciones, sino cifrar archivos.

A continuación, elija el tamaño de la llave: el tamaño de llave mínimo es de 768 bits y el tamaño predeterminado es de 1024 bits el máximo sugerido es de 2048. ¿Qué tamaño de llave desea? (1024) De nuevo, el tamaño predeterminado es suficiente para la mayoría de usuarios y representa un nivel de seguridad "extremadamente " fuerte.

Luego, elija la fecha de expiración de la llave. Es una buena idea elegir una fecha en lugar de utilizar la predeterminada, la cual es "ninguna." Si por ejemplo, la dirección de correo-e en la llave se invalida, la fecha de expiración le recordará a otros parar mediante la llave pública.

Especifique cuánto tiempo debe ser válida la llave: 0 = la llave no expira, d = la llave expira en n días, w = la llave expira en n meses o, y = la llave expira en n años?

Al ingresar el valor de **1y**, por ejemplo, valida la llave por un año. (Puede cambiar esta fecha de expiración después de generar la llave, si cambia de parecer.)

Antes el programa **gpg** pide información sobre la firma, aparece el siguiente mensaje: **Is this correct (y/n)?** Ingrese **y** para finalizar el proceso.

Luego, ingrese le nombre y la dirección de correo-e. Recuerde que este proceso intenta autenticarlo a usted como un individuo real. Por esta razón, incluya su nombre verdadero. No utilice apodos ni identificadores, puesto que ocultan o confunden su identidad.

Ingrese su dirección de correo-e real para su llave GPG. Si elige una dirección de correo-e falsa, será más difícil para las demás encontrar su llave pública y dificultará la autenticación de sus comunicaciones. Si utiliza esta llave GPG para [[DocsProject/SelfIntroduction| self-introduction]] en su lista de correo, por ejemplo, ingrese la dirección de correo-e que usted utiliza en esa lista.

Use el campo del comentario para incluir los apodos u otra información adicional.. (Algunas personas usan las llaves para diferentes propósitos e identifican cada llave con un comentario, tal como "Oficina" o "Proyectos de código abierto.")

En el indicador de confirmación, ingrese la letra O para continuar si todas las entradas son correctas, o utilice las otras opciones para corregir los problemas. Por último, ingrese la frase de paso para su llave secreta. El programa **gpg** le pedirá que ingrese la frase de paso dos veces para asegurarse de que no se incurre en errores al escribirla.

Por último, **gpg** genera datos aleatorios para crear la llave tan única como sea posible. Desplace el ratón, escriba las llaves aleatorias o realice otras tareas en el sistema durante este paso para agilizar el proceso. Cuando haya terminado, sus llaves estarán completas y listas para ser usadas:

```
pub  1024D/1B2AFA1C 2005-03-31 John Q. Doe <jqdoe@example.com>
Key fingerprint = 117C FE83 22EA B843 3E86  6486 4320 545E 1B2A FA1C
sub  1024g/CEA4B22E 2005-03-31 [expires: 2006-03-31]
```

La huella digital de la llave es una "firma" abreviada para su clave. Le permite a usted confirmar que ha recibido la llave pública sin ninguna manipulación. Usted no necesita anotar esta huella digital.

Para desplegar la huella en cualquier momento, use este comando sustituyendo su dirección de correo-e **`gpg --fingerprint jqdoe@example.com`**

Su "ID de llave GPG" consta de 8 hexa dígitos que identifican la llave pública. En el ejemplo anterior, el ID de llave GPG es 1B2AFA1C. En la mayoría de los casos, si se les pregunta por el ID de llave, debe anteponer "0x" al ID de la llave, como en "0x1B2AFA1C".

Advertencia

Si olvidó la frase de paso, no se puede usar la clave y los datos cifrados que la usan se perderán.

3.9.4. Acerca del cifrado de llaves públicas

1. *Wikipedia - Public Key Cryptography*[9]
2. *HowStuffWorks - Encryption*[10]

Principios generales de protección de información

Los siguientes principios generales proporcionan una vision general de las buenas prácticas de seguridad:

- Cifra todos los datos transmitidos en redes para ayudar a prevenir los ataques de intermediarios e intrusos. Es importante cifrar la información de autenticación como por ejemplo las contraseña.
- Minimiza la cantidad de software instalado y de servicios en ejecución.
- Utiliza software de mejoramiento de seguridad y herramientas, por ejemplo, Seguridad mejorada de Linux (SELinux) para Control de acceso obligatorio (MAC), los iptables Netfilter para filtraje de paquetes (cortafuegos) y el Guardián de Privacidad GNU (GnuPG) para cifrar archivos .
- Si es posible, ejecute cada servicio de red en un sistema independiente para minimizar el riesgo de un servicio en peligro de ser utilizado para comprometer otros servicios.
- Mantenga cuentas de usuario: cree y aplique un política de contraseñas vigorosas; borre cuentas de usuarios no utilizadas.
- Revise de forma rutinaria el sistema y los registros de aplicación. Los registros de sistemas relevantes se escriben a **`/var/log/secure`** y **`/var/log/audit/audit.log`**. Nota: el envío de registros a un servidor de registros dedicado evita que los atacantes modifiquen fácilmente los registros locales para evitar detección.
- Nunca ingrese como usuario root a menos que sea absolutamente necesario. Se recomienda que los administradores usen **`sudo`** para ejecutar comandos como root cuando se requiera. Los usuarios que pueden ejecutar **`sudo`** se especifican en **`/etc/sudoers`**. Use la herramienta **`visudo`** para editar **`/etc/sudoers`**.

4.1. Consejos, guías y herramientas

La *National Security Agency (NSA)*[1] de los Estados Unidos pone a disposición guías de protección y consejos para muchos sistemas operativos diferentes, para ayudar a las agencias, negocios e individuos a proteger sus sistemas contra ataques. Las siguientes guías (en formato PDF) proporcionan instrucciones para Red Hat Enterprise Linux 6:

- *Hardening Tips for the Red Hat Enterprise Linux 5*[2]
- *Guide to the Secure Configuration of Red Hat Enterprise Linux 5*[3]

[1] http://www.nsa.gov/

Nota

Referencias de guías de protección de Red Hat Enterprise Linux 5 se proporcionan en este documento hasta que las guías de protección de Red Hat Enterprise Linux 6 estén disponibles. Mientras tanto, observe que las guías de protección de Red Hat Enterprise 5 no se pueden aplicar completamente a Red Hat Enterprise Linux 6.

La *Defense Information Systems Agency (DISA)*[4] proporciona documentación, listas de verificación y pruebas para ayudar a proteger el sistema (*Information Assurance Support Environment*[5]). La *UNIX SECURITY TECHNICAL IMPLEMENTATION GUIDE*[6] (PDF) es una guía específica para seguridad de UNIX - se recomienda tener un conocimiento avanzado de UNIX y Linux antes de leer esta guía.

El DISA *Unix Security Checklist*[7] proporciona una colección de documentos y listas de verificación que van desde propiedades y modos para sistemas de archivos hasta el control de parches.

[4] http://www.disa.mil/
[5] http://iase.disa.mil/index2.html
[6] http://iase.disa.mil/stigs/stig/unix-stig-v5r1.pdf
[7] http://iase.disa.mil/stigs/checklist/

Instalación segura

La seguridad comienza en el primer momento en que inserta un CD o DVD en el disco duro para instalar Red Hat Enterprise Linux. La configuración de su sistema desde el comienzo facilita la implementación de seguridad adicional más adelante.

5.1. Particiones de discos

La NSA recomienda la creación de particiones independientes para /boot, /, /home, /tmp y /var/tmp. Las razones son diferentes para cada una y se abordarán en la descripción de cada partición.

/boot - Esta es la primera partición leída por el sistema durante el arranque. El gestor de arranque y las imágenes de kernel utilizadas para arrancar el sistema dentro de Red Hat Enterprise Linux se almacenan en esta partición. Esta partición no se debe cifrar. Si la partición se incluye en / y dicha partición está cifrada no estará disponible y por lo tanto no podrá arrancar.

/home - Cuando los datos de usuario (/home) se almacenan en / en lugar de una partición independiente, la partición puede llenarse y ocasionar así que el sistema operativo se vuelva inestable. También, al actualizar el sistema a la versión siguiente de producto Red Hat Enterprise Linux es mucho más fácil mantener los datos en la partición /home ya que no se sobrescribirá durante la instalación. Si la partición de root (/) se corrompe sus datos se pueden perder para siempre. Si utiliza una partición independiente hay un poco más de protección ante la pérdida de datos. También puede destinar esta partición para copias de seguridad frecuentes.

/tmp y /var/tmp - Los directorios /tmp y /var/tmp se utilizan para almacenar datos que no necesitan ser almacenados por un largo periodo de tiempo. Sin embargo, si una gran cantidad de datos inunda uno de estos directorios, puede consumir todo el espacio de almacenamiento. Si esto sucede y los directorios están almacenados dentro de / entonces el sistema se puede volver inestable y colgar. Por esta razón, es una buena idea desplazar estos directorios a sus propias particiones.

5.2. Utilice el cifrado de particiones LUKS

Durante el proceso de instalación se le dará al usuario la opción para cifrar las particiones. El usuario debe proporcionar la frase de paso para desbloquear la llave de cifrado masivo que se utilizará para garantizar la seguridad de los datos de la partición.

Mantenimiento de software

El mantenimiento del software es extremadamente importante para proveer un sistema seguro. Es de vital importancia corregir el software tan pronto como esté disponible con el fin de evitar que los agresores utilicen los agujeros conocidos para infiltrarse en su sistema

6.1. Software mínimo de instalación

Se recomienda instalar únicamente los paquetes que va a utilizar, ya que cada parte de software en su computadora podría contener una vulnerabilidad. Si va a instalar desde el DVD aproveche la oportunidad de seleccionar exactamente los paquetes que desea instalar durante la instalación. Cuando necesite otro paquete, se podrá añadir al sistema más adelante.

6.2. Planeación y configuración de actualizaciones de seguridad

Todo software contiene errores. Por lo general, dichos errores pueden resultar en una vulnerabilidad que puede exponer su sistema a usuarios malintencionados. Una causa común de intrusión son los sistemas no parcheados. Se debe tener un plan para instalar parches de seguridad en una forma oportuna para que esas vulnerabilidades no sean aprovechadas.

Las actualizaciones de seguridad para usuarios domésticos deben instalarse tan pronto como sea posible. La configuración de instalación automática de actualizaciones de seguridad es una forma de evitar tener que recordar, pero se corre el leve riesgo de causar un conflicto con su configuración o con otro software en el sistema.

Para usuarios domésticos o negocios, las actualizaciones se deben probar y programar para instalación. Los controles adicionales necesitarán usarse para proteger el sistema durante el tiempo entre el lanzamiento del parche y su instalación en el sistema. Estos controles dependen de la vulnerabilidad exacta, pero podrían incluir reglas de cortafuegos adicionales, el uso de cortafuegos externos o cambios en la configuración de software.

6.3. Ajuste de actualizaciones automáticas

Red Hat Enterprise Linux está configurado para aplicar todas las actualizaciones en una programación diaria. Si desea cambiar su instalación de actualizaciones del sistema debe hacerlo a través de '"Preferencias de actualización de software"'. Puede cambiar la programación, el tipo de actualizaciones a aplicar o notificarle sobre actualizaciones disponibles.

En Gnome, puede encontrar controles para sus actualizaciones en: **`Sistema -> Preferencias -> Actualizaciones de software`**. En KDE se localiza en: **`Aplicaciones -> Configuración -> Actualizaciones de software`**.

6.4. Instalación de paquetes firmados desde repositorios bien conocidos

Los paquetes de software se publican a través de repositorios. Todos los repositorios conocidos admiten la firma de paquetes. El firmado de paquetes utiliza la tecnología de clave pública para comprobar que el paquete que fue publicado por el repositorio no se ha modificado desde que la firma fue aplicada. Así se proporciona una cierta protección contra las instalaciones de software que pueden haber sido alteradas maliciosamente después de la creación del paquete, pero antes de descargarlo.

El uso de demasiados repositorios poco fiables, o repositorios de paquetes sin firma tiene un mayor riesgo de introducción de código malintencionado o vulnerable en el sistema. Tenga cuidado al añadir repositorios para actualización de software o yum.

Normas y regulaciones federales

7.1. Introducción

Con el fin de mantener los niveles de seguridad, su empresa debe hacer esfuerzos para cumplir con las especificaciones de seguridad del gobierno y de la industria. Este capítulo describe algunas de las normas y regulaciones.

7.2. Estándar de procesamiento de información federal (FIPS)

La publicación 149-2 de la Federal Information Processing Standard (FIPS), es un estándar de seguridad, desarrollado por el grupo de trabajo del gobierno norteamericano y la industria para validar la calidad de módulos criptográficos. Las publicaciones FIPS (entre ellas la 140-2) se encuentran en la siguiente URL: *http://csrc.nist.gov/publications/PubsFIPS.html*. Observe que en el momento en que se escribió esta guía, la publicación 140-3 era un borrador y puede no representar el estándar completo. El estándar FIPS proporciona cuatro (4) *niveles* de seguridad para garantizar un cubrimiento adecuado de diferentes industrias, implementaciones de módulos criptográficos y tamaños y requerimientos corporativos. Estos niveles se describen a continuación:

- Nivel 1 - El nivel de seguridad 1 proporciona en nivel inferior de seguridad. Los requerimientos de seguridad básica se especifican para un módulo criptográfico (por ejemplo, se debe usar al menos un algoritmo aprobado o una función de seguridad aprobada). Los mecanismos de seguridad no específicos se requieren en un nivel 1 de seguridad criptográfico que va más allá de los requerimientos básicos para componentes de grado de producción. Un ejemplo de un módulo criptográfico de nivel 1 de seguridad es una panel de cifrado de un computador personal.

- Nivel 2 - El nivel de seguridad 2 mejora los mecanismos de seguridad física de un módulo de nivel de seguridad criptográfica 1 al añadir el requerimiento de manipulación de pruebas, el cual incluye el uso de revestimientos a prueba de manipulaciones o sellos o de cerraduras resistentes en cubiertas desplazables o puertas del módulo. Los revestimientos a prueba de manipulaciones o sellos se colocan en un módulo criptográfico para que el recubrimiento o el sello se rompan para acceder físicamente a las claves de cifrado de texto plano y los parámetros críticos de seguridad (CSP) en el módulo. Los sellos a prueba de manipulaciones o las cerraduras resistentes se colocan en cubiertas o puertas para protegerlas de accesos no autorizados.

- Nivel 3 - Además de los mecanismos de seguridad a prueba de manipulaciones físicas requeridas en el nivel de protección 2, el nivel de seguridad 3 intenta evitar que los intrusos puedan acceder a los CSP dentro del módulo criptográfico. Los mecanismos de seguridad físicos, necesarios en el nivel de seguridad 3 están destinados a tener una alta probabilidad de detección y respuesta tanto a los intentos de acceso físico, como al uso o modificación de los módulos criptográficos. Los mecanismos de seguridad física pueden incluir el uso de encerramientos fuertes y manipulación de detección o respuesta de circuitos que establecen a cero todos los CSP de texto cuando se abren las cubiertas o puertas desplazables del módulo criptográfico.

- Nivel 4 - El nivel de seguridad 4 proporciona el máximo nivel de protección definido en esta norma. En este nivel de seguridad, los mecanismos de seguridad física ofrecen una capa de protección completa en todo el módulo criptográfico para detectar y responder a todos los intentos no autorizados de acceso físico. La penetración de la caja del módulo criptográfico desde cualquier dirección tiene una muy alta probabilidad de ser detectada, lo que resulta en el establecimiento a cero de los CSP de texto. Los módulos criptográficos del nivel de seguridad 4 son útiles para el funcionamiento en entornos que no tienen físicamente protección.

Consulte todo el estándar FIPS 140-2 en *http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf* para obtener mayor información en estos niveles y las otras especificaciones del estándar de FIPS.

7.3. Manual de operación de programa de seguridad industrial Nacional (NISPON)

El NISPOM (conocido también como DoD 5220.22-M), como un componente del programa de Seguridad Industrial Nacional (NISP), establece una serie de procedimientos y requisitos para todos los contratistas respecto a la información clasificada. El actual NISPOM tiene fecha de febrero 28 de 2006. El documento de NISPOM también puede ser descargado desde la siguiente URL: *https://www.dss.mil/GW/ShowBinary/DSS/isp/fac_clear/download_nispom.html*.

7.4. Estándar de seguridad de datos de industria de tarjetas de pago (PCI DSS)

Desde la *https://www.pcisecuritystandards.org/about/index.shtml*: *El PCI ,Security Standards Council es un foro abierto, lanzado en 2006, el cual es responsable del desarrollo, administración, educación y reconocimiento de los estándares de seguridad de PCI, entre ellos el Estándar de seguridad de datos (DSS).*

Puede descargar el estándar PCI DSS desde *https://www.pcisecuritystandards.org/security_standards/pci_dss.shtml*.

7.5. Guía de implementación de seguridad técnica

La Guía de implementación técnica de seguridad o STIG es una metodología para instalación segura estandarizada y mantenimiento de software y hardware informáticos.

Consulte la siguiente URL para obtener un listado de las guías posibles: *http://iase.disa.mil/stigs/stig/index.html*.

Referencias

Las siguientes referencias son indicadores de información adicional que es importante para SELinux y Red Hat Enterprise Linux, pero que va más allá del contenido de esta guía. Observe que debido al rápido desarrollo de SELinux, parte de este material puede aplicarse a lanzamientos específicos de Red Hat Enterprise Linux.

Libros

SELinux por ejemplo
Mayer, MacMillan y Caplan

Prentice Hall, 2007

Guías y ayuda

Entendimiento y personalización de la política SELinux Apache HTTP
http://docs.fedoraproject.org/selinux-apache-fc3/

Guías y charlas de Russell Coker
http://www.coker.com.au/selinux/talks/ibmtu-2004/

CÓMO escribir la política de SELinux
http://www.lurking-grue.org/writingselinuxpolicyHOWTO.html

Base de conocimientos de Red Hat
http://kbase.redhat.com/

Información general

Sitio web de NSA SELinux
http://www.nsa.gov/selinux/[1]

Preguntas frecuentes sobre NSA SELinux
http://www.nsa.gov/selinux/info/faq.cfm[2]

Preguntas frecuentes sobre SELinux de Fedora
http://docs.fedoraproject.org/selinux-faq/

Seguridad mejorada Linux de código abierto de NSA SELinux
http://www.oreilly.com/catalog/selinux/

Tecnología

Una visión general sobre clases de objetos y permisos
http://www.tresys.com/selinux/obj_perms_help.html

Integración de soporte flexible para políticas de seguridad en el sistema operativo de Linux (Una historia de la implementación de Flask en Linux)
http://www.nsa.gov/research/_files/selinux/papers/selsymp2005.pdf

Implementación de SELinux como un módulo de seguridad de Linux
http://www.nsa.gov/research/_files/publications/implementing_selinux.pdf

[1] http://www.nsa.gov/research/selinux/index.shtml
[2] http://www.nsa.gov/research/selinux/faqs.shtml

Configuración de políticas para Seguridad Mejorada de Linux
http://www.nsa.gov/research/_files/selinux/papers/policy/policy.shtml

Comunidad

Guía de usuario SELinux Fedora
http://docs.fedoraproject.org/en-US/Fedora/13/html/Security-Enhanced_Linux/

Guía de servicios confinados de Administración de SELinux Fedora
http://docs.fedoraproject.org/en-US/Fedora/13/html/Managing_Confined_Services/

Página de la comunidad de SELinux
http://selinuxproject.org/

IRC
irc.freenode.net, #selinux, #fedora-selinux, #security

Historia

Breve historia de Flask
http://www.cs.utah.edu/flux/fluke/html/flask.html

Fondo total en Fluke
http://www.cs.utah.edu/flux/fluke/html/index.html

Apéndice A. Estándares de cifrado

A.1. Cifrado sincronizado

A.1.1. Estándar de cifrado avanzado - AES

En criptografía, el Estándar de cifrado avanzado (AES) es un estándar de cifrado adoptado por el gobierno de los Estados Unidos. El estándar consiste en tres cifras de bloques, AES-128, AES-192 y AES-256, adoptados de una gran colección originalmente publicada como Rijndael. Cada cifra de AES tiene un tamaño de bloque de 128, 192 y 256 bits, respectivamente. Las cifras de AES han sido analizada extensamente y ahora se utilizan a nivel mundial, como es el caso de su predecesor, el Estándar de cifrado de datos (DES).[1]

A.1.1.1. Usos de AES

A.1.1.2. Historia de AES

AES fue anunciado por el National Institute of Standards and Technology (NIST) como U.S. FIPS PUB 197 (FIPS 197) en noviembre 26 de 2001 después del proceso de normalización de 5 años en el cual quince diseños fueron presentados y evaluados antes de que Rijndael fuera seleccionado como el más apropiado (vea el proceso de Estándar de cifrado avanzado para obtener más información). Se convirtió en el estándar efectivo en Mayo 26 de 2002. Está disponible en varios paquetes de cifrado. AES es el primer estándar de cifra abierta y públicamente aprobado por el NSA para información altamente confidencial. (vea Seguridad de AES, abajo). [2]

La cifra de Rijndael fue desarrollada por dos criptógrafos belgas, Joan Daemen y Vincent Rijmen y enviada por ellos al proceso de selección. Rijndael (pronunciado [r□inda□l]) es la combinación de los nombres de los dos inventores. [3]

A.1.2. Estándar de cifrado de datos - DES

El Estándar de cifrado de datos (DES) es una cifra de bloque (una forma de cifrado secreto compartido) que fue seleccionada por el National Bureau of Standards (La oficina nacional de normas) como un Federal Information Processing Standard (Estándar de procesamiento de información federal) para los Estados Unidos en 1976 y el cual se ha extendido en uso a nivel mundial. Se basa en un algoritmo de clave simétrica que utiliza una llave de 56 bits. Al principio, el algoritmo fue controvertido con elementos de diseño clasificados, una longitud relativamente corta y sospechas de una puerta trasera de la Agencia de Seguridad Nacional (NSA). DES como consecuencia surgió ante este intenso escrutinio académico que motivó el entendimiento moderno de cifras de bloque y sus criptoanálisis.[4]

A.1.2.1. Usos de DES

[1] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Advanced_Encryption_Standard*
[2] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Advanced_Encryption_Standard*
[3] "Advanced Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Advanced_Encryption_Standard*
[4] "Data Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Data_Encryption_Standard*

A.1.2.2. Historia de DES

Ahora DES es considerada como insegura para muchas aplicaciones. Esto se debe principalmente al tamaño de la llave de 56 bits; en enero de 1999, distributed.net y the Electronic Frontier Foundation colaboraron para descifrar públicamente una llave DES en 22 horas y 15 minutos (ver cronología). Hay también algunos resultados analíticos que demuestran la debilidad en la cifra, aunque no es factible montarla en la práctica. Se cree que el algoritmo es prácticamente seguro en la forma de triple DES, aunque hay ataques teóricos. En años recientes, la cifra ha sido desbancada por el Estandar de cifrado avanzado (AES).[5]

En alguna documentación, se hace la distinción entre DES como el estándar y DES como el algoritmo conocido como DEA (Algoritmo de datos cifrados). Cuando se habla de "DES" se deletrea como una abreviatura en ingles (/□di□□i□□□s/) o como un acrónimo de una sílaba (/□d□z/).[6]

A.2. Cifrado de llave pública

La criptografía de llave pública es un método criptográfico empleado por muchos algoritmos criptográficos y cripto-sistemas cuyas características distintivas son el uso de algoritmos de llave asimétrica o de una adición de algoritmos de llave simétrica. Mediante las técnicas criptográficas de llave pública muchos métodos anteriormente desconocidos para proteger comunicaciones o autenticar mensajes se han vuelto prácticos. No requieren un intercambio inicial seguro de una o más llaves secretas como se requería al usar algoritmos de llave simétricos. También se puede usar para crear firmas digitales.[7]

La criptografía de llave pública es una tecnología fundamental y ampliamente utilizada a nivel internacional y es el enfoque que subyace en estándares de Internet tales como Seguridad de capa de transporte (TLS) (sucesor de SSL), PGP y GPG.[8]

La técnica distintiva utilizada en criptografía de llave pública es el uso de algoritmos de llave asimétrica, donde la llave utilizada para cifrar el mensaje no es la misma utilizada para descifrarla.Cada usuario tiene un par de llaves criptográficas una llave pública y una privada. La llave privada se mantiene en secreto, mientras que la llave pública puede ser distribuida ampliamente. Los mensajes se cifran con la llave pública del destinatario y solamente pueden descifrarse con la correspondiente clave privada. Las llaves se relacionan matemáticamente, pero la llave privada no puede ser factible (i.e. en la práctica o proyectada) derivada de la llave pública. El descubrimiento de dichos algoritmos fue el que revolucionó la práctica de criptografía que comenzó a mediados de los años setenta.[9]

Por el contrario, los algoritmos de llave simétrica, variaciones de los cuales se han hecho por miles de años, usan una sola llave secreta compartida por el remitente y el destinatario (el cual debe también mantenerla privada, lo que representa una ambigüedad de la terminología común, el remitente y destinatario deben compartir desde un principio una clave segura.[10]

Puesto que los algoritmos de llave secreta son casi siempre de menor computo intensivo, es común intercambiar una llave mediante un algoritmo de intercambio de llave. Por ejemplo, PDP, y la familia SSL/TLS de esquemas lo hacen y en consecuencia, se llaman cripto-sistemas híbridos.[11]

[5] "Data Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Data_Encryption_Standard*
[6] "Data Encryption Standard." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Data_Encryption_Standard*
[7] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*
[8] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*
[9] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*
[10] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*
[11] "Public-key Encryption." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Public-key_cryptography*

A.2.1. Diffie-Hellman

El intercambio de llave Diffie–Hellman (D–H) es un protocolo criptográfico que permite a dos partes que no tienen conocimiento previo el uno del otro, establecer en conjunto una llave secreta compartida en un canal de comunicaciones inseguras. Esta llave puede entonces usarse para cifrar comunicaciones subsiguientes mediante la cifra de llave simétrica.[12]

A.2.1.1. Historia de Diffie-Hellman

El esquema fue inicialmente publicado por Whitfield Diffie y Martin Hellman en 1976, aunque más tarde se supo que se había inventado individualmente unos años atrás dentro de la agencia de inteligencia de señales británica (GCHQ) por Malcolm J. Williamson pero se había mantenido en secreto. En 2002, Hellman sugirió el algoritmo de intercambio de llave llamado Diffie–Hellman–Merkle en reconocimiento de la contribución de Ralph Merkle al invento de la criptografía de llave pública (Hellman, 2002).[13]

Aunque el acuerdo de llave de Diffie–Hellman es un protocolo de acuerdo de llave anónimo (no-autenticado), proporciona las bases para una variedad de protocolos autenticados y se utiliza para proveer confidencialidad directa perfecta en los modos efímeros de Seguridad de capa de transporte (conocidos como EDH o DHE según el paquete de cifrado).[14]

La patente 4,200,770 de Estados Unidos, ya expirada, describe el algoritmo y los créditos de Hellman, Diffie, y Merkle como inventores.[15]

A.2.2. RSA

En criptografía, RSA (que significa Rivest, Shamir y Adleman quienes primero lo describieron públicamente; ver abajo) es un algoritmo para criptografía de llave pública. Se conoce como el primer algoritmo apropiado tanto para firmar como para cifrar, y fue el primer gran avance en criptografía de llave pública. RSA es utilizado extensamente en protocolos de comercio electrónico y se considera seguro dado a la longitud suficiente de las claves y del uso de implementaciones actualizadas.

A.2.3. DSA

DSA (Algoritmo de firma digital) es un estándar para firmas digitales, un estándar del gobierno federal de los Estados Unidos para firmas digitales. DSA es para firmas únicamente y no es un algoritmo de cifrado. [16]

A.2.4. SSL/TLS

Seguridad de capa de transporte (TLS) y su predecesor, Capa de conexión segura (SSL), son protocolos criptográficaos que ofrecen seguridad para comunicaciones en redes como la Internet. TLS y SSL cifran los segmentos de conexiones de redes en la capa de transporte de extremo a extremo.

Varias versiones de protocolos se utilizan ampliamente en aplicaciones tales como navegadores de red, correo electrónico, fax por Internet, mensajería instantánea y voz IP (VoIP).[17]

[12] "Diffie-Hellman." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Diffie-Hellman*
[13] "Diffie-Hellman." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Diffie-Hellman*
[14] "Diffie-Hellman." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Diffie-Hellman*
[15] "Diffie-Hellman." *Wikipedia.* 14 November 2009 *http://en.wikipedia.org/wiki/Diffie-Hellman*
[16] "DSA." *Wikipedia.* 24 February 2010 *http://en.wikipedia.org/wiki/Digital_Signature_Algorithm*
[17] "TLS/SSl." *Wikipedia.* 24 February 2010 *http://en.wikipedia.org/wiki/Transport_Layer_Security*

A.2.5. Cripto-sistema Cramer-Shoup

El sistema Cramer–Shoup es un algoritmo de llave asimétrica y fue el primer esquema eficiente y seguro demostrado contra el ataque de texto de cifrado mediante supuestos estándares criptográficos. Su seguridad se basa en la dificultad computacional (ampliamente asumida pero no demostrada) del supuesto decisivo de Diffie–Hellman. Desarrollado por Ronald Cramer y Victor Shoup in 1998, es una extensión del cripto-sistema Elgamal. A diferencia de Elgamal, el cual es extremadamente maleable, Cramer–Shoup añade elementos adicionales para garantizar la no maleabilidad incluso contra un atacante recursivo. Esta no maleabilidad se realiza a través del uso de una función hash de colisión resistente y cómputos adicionales, resultando en un texto cifrado que es dos veces más grande que un Elgamal.[18]

A.2.6. Cifrado ElGamal

En criptografía, el sistema de cifrado ElGamal es un algoritmo de cifrado de llave asimétrico para criptografía de llave pública basado en el acuerdo de llave Diffie-Hellman. Fue descrito por Taher Elgamal en 1985. El cifrado ElGamal se utiliza en software libre de Guardián de Privacidad, GNU, versiones recientes de PGP, y otros cripto-sistemas.[19]

[18] "Cramer-Shoup cryptosystem." *Wikipedia.* 24 February 2010 *http://en.wikipedia.org/wiki/Cramer–Shoup_cryptosystem*
[19] "ElGamal encryption" *Wikipedia.* 24 February 2010 *http://en.wikipedia.org/wiki/ElGamal_encryption*

Apéndice B. Historial de revisiones

Revisión 1.5-0 Apr 19 2010 **Scott Radvan** *sradvan@redhat.com*

Correcciones menores, creación final de Beta

Revisión 1.4.1-0 Mar 5 2010 **Scott Radvan** *sradvan@redhat.com*

Revisión de QE y actualizaciones

Revisión 1.3-0 Feb 19 2010 **Scott Radvan** *sradvan@redhat.com*

Enviar al área de prueba para revisión