

Nos centraremos en cómo aumentar la seguridad en sistemas basados en el sistema operativo Linux. Su gratuidad, flexibilidad, potencia, apertura, facilidad para obtención de herramientas y otras muchas virtudes hacen de Linux la elección cada vez más frecuente entre los administradores de sistemas a la hora de decidirse por una u otra plataforma.

Aunque Linux es un sistema muy robusto e incorpora las características de seguridad comunes a todos los sistemas tipo Unix, a pesar de todo resulta fundamental dedicar cierto tiempo y recursos para conocer cuáles son sus debilidades y vías frecuentes de ataque y adoptar posteriormente las medidas más eficaces para contrarrestarlas. A menudo un sistema operativo es tan seguro como la astucia y habilidad de su administrador.

Se aprenderá todo lo necesario para configurar su red de manera segura, así como indicaciones y referencias donde profundizar en algunos aspectos de la administración, auditorías de seguridad, planes de contingencia, etc.

Lo primero que tenemos que tener en mente es que no existe nada como un sistema completamente seguro. Todo lo que puede hacer es aumentar la dificultad para que alguien pueda comprometer su sistema. En el caso medio del usuario de Linux en casa, no se requiere demasiado para mantener alejado al cracker. Para usuarios con grandes requisitos (bancos, compañías de telecomunicaciones, etc.) se requiere mucho más trabajo.

Otro factor a tener en cuenta es que cuanto más incrementa la seguridad de su sistema, más intrusiva se vuelve la seguridad, en otras palabras, su sistema puede perder funcionalidad y resentirse la comodidad. Necesita decidir en qué medida su sistema es utilizable y en qué medida es seguro para sus propósitos. Por ejemplo, puede necesitar que cualquiera marque a su modem para que éste devuelva la llamada a su casa. Esto es más seguro, pero si alguien no está en casa hace más difícil que se pueda conectar. También puede configurar su sistema Linux sin conexión a Internet, pero esto dificulta que pueda navegar por las webs. Si tiene un sitio medio-grande, debería establecer una "Política de Seguridad" que indique qué niveles requiere su sitio y qué medidas de evaluación se realizan.

Los dos puntos principales de los que se tiene que dar cuenta cuando lea estas páginas son:

- \* Tenga cuidado con su sistema. Verifique los registros (logs) del sistema, tales como /var/log/messages y no pierda de vista su sistema.

- \* Tenga su sistema actualizado, estando seguro de que ha instalado las versiones actuales de los programas y esté al tanto de las nuevas alertas de seguridad. Hacer esto le ayudará a conseguir que su sistema sea mucho más seguro.

## **Por qué es necesaria la seguridad**

Ante todo Linux es un sistema multiusuario real. Puede haber varios usuarios distintos trabajando a la vez cada uno desde su terminal. El sistema tiene la obligación de proteger a unos usuarios frente a otros y protegerse a sí mismo.

Linux es una excelente estación de trabajo aislada, pero lo habitual es que cada máquina linux esté conectada a una red y además esté prestando servicios de red. El sistema tiene la obligación de garantizar los servicios que presta.

Además, con la generalización de las conexiones con Internet y el rápido desarrollo del software, la seguridad se está convirtiendo en una cuestión cada vez más importante. Ahora, la seguridad es un

requisito básico, ya que la red global es insegura por definición. Mientras sus datos estén almacenados en un soporte informático, mientras sus datos vayan desde un sistema X a otro sistema Y a través de un medio físico, Internet, por ejemplo, puede pasar por ciertos puntos durante el camino proporcionando a otros usuarios la posibilidad de interceptarlos, e incluso alterar la información contenida. Incluso algún usuario de su sistema puede modificar datos de forma maliciosa para hacer algo que nos pueda resultar perjudicial. Con el acceso masivo y barato a Internet se han reducido notablemente los costes de un atacante para asaltar un sistema en red, a la vez que ha aumentado paralelamente el número de potenciales atacantes. Resumiendo, a nadie le gustaría que desconocidos abran su correo privado, que miren en sus cajones, que se hagan copias de las llaves de su escritorio o de la tarjeta de crédito. Pues todo esto es aplicable en la misma medida a las redes telemáticas.

También queremos remarcar el carácter dinámico de la seguridad de los sistemas en red. Continuamente aparecen nuevos métodos para conseguir accesos indebidos o comprometer el correcto funcionamiento de la red. Esto obliga a estar actualizado permanentemente, y consultar las publicaciones electrónicas que informan de los últimos sucesos detectados. Evidentemente, estas publicaciones informan principalmente sobre actividades que ya se han llevado a cabo, por lo que esperamos que no sea el primero en sufrirlas. Pero también se puede encontrar información sobre debilidades detectadas antes de que se lleven a cabo.

Por todo esto, este curso no pretende proporcionar una lista actualizada de programas o servicios potencialmente inseguros o de programas que afectan a la seguridad (denominados exploits). Como continuamente aparecen nuevos programas para comprometer la seguridad de las redes y de las comunicaciones, lo que sí haremos será indicar los lugares más habituales donde buscar una información actualizada de ese tipo, y algunos métodos generales para prevenir que esos programas tengan éxito en su sistema.

¿Cómo de Seguro es Seguro?

Primero, tenga en cuenta que ningún sistema es "completamente" seguro. El único sistema seguro es aquel que no está conectado en red, que está apagado y encerrado bajo llave.

Desde esta perspectiva, partimos de que todo lo único que puede hacer es aumentar la dificultad para que alguien pueda comprometer la seguridad de su sistema. Tampoco todos los usuarios tienen las mismas necesidades de seguridad en sus entornos. Por ejemplo, los usuarios domésticos de Linux, no necesitan demasiado trabajo para mantener alejados a los crackers ocasionales, mientras que para los usuarios muy especializados de Linux, como por ejemplo servidores de Internet, bancos, compañías de telecomunicaciones, etc., se necesita un trabajo mucho más detallado para garantizar la seguridad en los términos previstos.

También tenemos que tener en cuenta que existe una relación inversa entre seguridad y funcionalidad. Tiene que decidir dónde está el equilibrio entre la facilidad de uso de su sistema y su seguridad. Por ejemplo, puede necesitar que cualquiera que llame por el módem a su sistema, y nuestro módem tenga que devolver la llamada a su número de casa. Este mecanismo garantiza un mayor nivel de seguridad, pero si alguien no está en casa, le hace más difícil conectarse. Otra forma de aumentar la seguridad sería no tener conexión con Internet, pero seguro que no es eso lo que queremos.

También existe una relación inversa entre el nivel de seguridad y el número de servicios distintos que presta un sistema. Cada servicio prestado por un sistema puede ser susceptible de ser utilizado contra el propio equipo servidor, como puede ser el caso de bloqueos intencionados, conocidos como ataque de denegación de servicio (DoS). Pero un sistema servidor que no presta servicios es menos servidor.

En el caso de administrar un instalación mediana o grande conviene establecer una "Política de

Seguridad" que fije el nivel de seguridad que requiere ese sitio y que sistema de comprobación se realiza. Puede encontrar un ejemplo muy conocido de política de seguridad en el RFC 2196.

## **Seguridad en el desarrollo de Linux**

Como probablemente sabrá, las fuentes del núcleo de linux son abiertas. Cualquiera puede obtenerlas, analizarlas y modificarlas. Este modelo de desarrollo abierto, que siguen tanto Linux como la mayoría de las aplicaciones que se ejecutan sobre él, conduce a altos niveles de seguridad. Es cierto que cualquiera puede acceder al código fuente para encontrar las debilidades, pero no es menos cierto que el tiempo que tarda en aparecer la solución para cualquier debilidad se mide más fácilmente en horas que en días.

Gracias a esto, Linux es conocido por su alto nivel de estabilidad que parte del propio núcleo del sistema operativo (lo que propiamente es Linux).

Linux tiene disponible todos los servicios habituales en una red:

- \* Bases de datos.
- \* Servicios de internet.
- \* Servicio de ficheros e impresión.
- \* Utilidades necesarias para mantener el nivel de seguridad requerido.

Pero además hay que reseñar que cada servicio funciona sin afectar al resto de los servicios. Vd. puede modificar la dirección IP de su equipo, las rutas, añadir, parar o reiniciar un servicio concreto sin que el resto de los servicios se vean afectados. Sólo es necesario detener el equipo para realizar operaciones con el hardware, como añadir un disco duro, o utilizar un nuevo núcleo. No tendrá, pues, la necesidad de tener que ser Vd. mismo el atacante de su propio sistema, a diferencia de lo que ocurre en otros sistemas operativos.

## **¿Qué Quiere Proteger?**

Normalmente querrá garantizar que el sistema permanece en funcionamiento de forma adecuada. Querrá garantizar que nadie pueda obtener o modificar una información a la que no tiene derecho legítimo. Y también querrá garantizar unas comunicaciones seguras. Una buena planificación ayuda bastante a conseguir los niveles de seguridad que pretendemos. Antes de intentar asegurar su sistema debería determinar contra qué nivel de amenaza quiere protegerse, qué riesgo acepta o no y, como resultado, cómo de vulnerable es su sistema. Debería analizar su sistema para saber qué está protegiendo, por qué lo está protegiendo, qué valor tiene y quiénes tienen responsabilidad sobre sus datos y otros elementos.

\* Riesgo es la posibilidad de que un intruso pueda intentar acceder con éxito a su equipo. ¿Puede un intruso leer y escribir en ficheros o ejecutar programas que puedan causar daño? ¿Pueden borrar datos críticos? ¿Prevenir a su compañía de la pérdida de un trabajo importante realizado? Recuerde también que alguien que obtenga acceso a su cuenta, o su sistema, también puede hacerse pasar por Vd.

Además, basta tener una sola cuenta insegura en su sistema para comprometer toda su red. Un simple usuario al que se le permita presentarse al sistema usando un fichero rhost, o permitir el uso de un servicio inseguro como tftp, origina el riesgo de que los intrusos puedan usarlo para "meter un puerta". Una vez que el intruso tiene una cuenta de usuario en su sistema, o en el sistema de cualquier otro, puede usarla para conseguir acceso a otros sistemas o a otras cuentas.

La amenaza proviene de alguien que tiene motivos para obtener acceso sin autorización a su red o

equipo. Debe decidir en quién confía para dotar de acceso a su sistema y qué amenaza puede representar.

Las amenazas proceden de varios tipos de intrusos, y es útil tener en mente sus diferentes características cuando esté asegurando sus sistemas.

- El Curioso - Este tipo de intruso está interesado básicamente en qué tipo de sistema y datos posee.
- El Malicioso - Este tipo de intruso pretenderá, bien hacerle caer su sistema, modificar su página web o cualquier otra cosa que le cueste tiempo y dinero recuperar.
- El Intruso muy personalizado - Este tipo de intruso trata de usar su sistema para ganar popularidad o mala fama. Puede usar su sistema para promocionar sus habilidades.
- La Competencia - Este tipo de intruso está interesado en los datos que tiene en su sistema. Puede ser alguien que piense que tiene algo que le puede interesar financieramente o de otra forma.

La vulnerabilidad describe el nivel de protección de su equipo frente a otra red, y la posibilidad potencial para alguien que pueda obtener acceso no autorizado.

¿Qué está en juego si alguien entra en su sistema? Desde luego será distinto para un usuario en casa con un enlace PPP dinámico que para las compañías que conectan su máquina a Internet u otra gran red.

¿Cuanto tiempo me llevaría recuperar/recrear cualquier dato que se ha perdido? Una inversión en tiempo ahora puede ahorrar diez veces más de tiempo con posterioridad si tiene que recrear los datos que se perdieron. ¿Ha verificado su estrategia de copias de respaldo, y ha verificado sus datos últimamente?

La seguridad pretende que los sistemas informáticos que utiliza una entidad se mantengan en funcionamiento según los requisitos de la política establecida por la propia entidad. Cada entidad define una serie de servicios que pretende obtener de una red de ordenadores para prestarlos a unos usuarios legítimos. Básicamente los requisitos de seguridad se puede resumir en una serie de puntos ilustrativos:

o Disponibilidad: consistente en mantener la información y los recursos de acuerdo con los requisitos de utilización que pretende la entidad que utiliza la red informática. La disponibilidad de la información pretende garantizar que no se limite el acceso autorizado a la información y el correcto funcionamiento de los recursos.

o Integridad: la información que se almacena en los sistemas o que circula por las líneas de comunicación debe estar protegida contra la modificación no autorizada. Requiere que la información sólo pueda ser modificada por las entidades autorizadas.

o Autenticidad: la información que se almacena o circula por una red debe permanecer protegida ante falsificaciones. La autenticidad requiere mecanismos de identificación correctos, asegurando que las comunicaciones se realizan entre entidades legítimas.

o Confidencialidad: pretende evitar la difusión no autorizada de la información. Requiere que la información sea accesible únicamente por las entidades autorizadas.

o No rechazo: establecer los mecanismos para que nadie pueda negar que ha realizado una determinada comunicación.

En particular, en Linux tendremos que proteger ciertos ficheros que contienen información sobre los usuarios (/etc/passwd, /etc/shadow), los ficheros de configuración del sistema (los contenidos en

/etc), el acceso al sistema para que se realice según marca la política prevista y la correcta utilización de los recursos para evitar abusos (p.e. si un usuario tiene sólo una cuenta de correo, que no pueda abrir una shell en el sistema). A todo esto habrá que sumar la protección de los distintos servicios que presta.

## **Seguridad física**

Las primeras medidas de seguridad que necesita tener en cuenta son las de seguridad física de sus sistemas. Hay que tomar en consideración quiénes tienen acceso físico a las máquinas y si realmente deberían acceder.

El nivel de seguridad física que necesita en su sistema depende de su situación concreta. Un usuario doméstico no necesita preocuparse demasiado por la protección física, salvo proteger su máquina de un niño o algo así. En una oficina puede ser diferente.

Linux proporciona los niveles exigibles de seguridad física para un sistema operativo:

- \* Un arranque seguro
- \* Posibilidad de bloquear las terminales
- \* Por supuesto, las capacidades de un sistema multiusuario real.

## **Seguridad en el arranque**

Cuando alguien inicia el sistema operativo Linux se encuentra con una pantalla de login: el sistema está pidiendo que se identifique. Si es un usuario conocido, podrá iniciar una sesión y trabajar con el sistema. Si no lo es, no tendrá opción de hacer absolutamente nada. Además, el sistema registra todos los intentos de acceso (fallidos o no), por lo que no pasarán desapercibidos intentos repetidos de acceso no autorizado.

LILO (Linux Loader) es el encargado de cargar el sistema operativo en memoria y pasarle información para su inicio. A su vez, Vd. puede pasarle parámetros a LILO para modificar su comportamiento.

Por ejemplo, si alguien en el indicador de LILO añade `init single`, el sistema se inicia en modo monousuario y proporciona una shell de root sin contraseña. Si en su entorno de trabajo cree necesario evitar que alguien pueda iniciar el sistema de esta forma, debería utilizar el parámetro `restricted` en el fichero de configuración de LILO (habitualmente `/etc/lilo.conf`). Este parámetro le permite iniciar normalmente el sistema, salvo en el caso de que se hayan incluido argumentos en la llamada a LILO, que solicita una clave. Esto proporciona un nivel de seguridad razonable: permite iniciar el sistema, pero no manipular el arranque. Si tiene mayores necesidades de seguridad puede incluir la opción `password`. De esta forma necesitará una clave para iniciar el sistema. En estas condiciones, sólo podrá iniciar el sistema quien conozca la clave.

Puede encontrar más detalles en las páginas del manual `lilo` y `lilo.conf`. Para ello, introduzca en la línea de comandos las siguientes órdenes:

```
# man lilo
# man lilo.conf
```

## **Otros detalles que le podrían resultar útiles.**

\* Prepare un disco de arranque del sistema. Es muy fácil, simplemente tiene que copiar el núcleo del sistema operativo en el disco, sin sistema de ficheros, e indicarle cual es la partición raíz del

sistema.

```
# dd if=/boot/vmlinuz of=/dev/fd0  
# rdev /dev/fd0 /dev/hdXY
```

Suponiendo que estemos usando un disco duro IDE, X indica el disco (a,b,c, o d), Y indica la partición (1,2,...).

\* Si tiene más de un sistema operativo en su máquina, le puede interesar hacer una copia de salvaguardia del MBR:

```
# dd if=/dev/hda of=/boot/arranque.mbr count=1
```

Y ahora algunas consideraciones generales:

\* Tenga en cuenta que ningún sistema es realmente seguro si alguien, con los conocimientos necesarios, puede usar su propio disco para arrancar.

\* Si tiene una máquina servidora, y tiene una clave para el arranque, la máquina no se reiniciará sin suministrar la clave y tendrá que acudir a introducirla en el caso de una parada no prevista.

### *Bloqueo de la consola*

En los entornos Unix es conocido el truco de ejecutar en una terminal, que alguien ha dejado inocentemente abierto, un guion que simule la pantalla de presentación al sistema. Entonces un usuario incauto introducirá su nombre y clave, que quedarán a merced del autor del engaño.

Si se aleja de su máquina de vez en cuando, estaría bien poder bloquear su consola para que nadie pueda manipularla o mirar su trabajo. Dos programas que hacen esto son xlock y vlock.

xlock bloquea la pantalla cuando nos encontramos en modo gráfico. Está incluido en la mayoría de las distribuciones Linux que soportan X. En general puede ejecutar xlock desde cualquier xterm de su consola y bloqueará la pantalla de forma que necesitará su clave para desbloquearla. Mire la página de manual para ver más detalles, algunos curiosos.

vlock es un simple programa que le permite cerrar alguna o todas las consolas virtuales de su máquina Linux. Puede bloquear sólo aquella en la que está trabajando o todas. Si sólo cierra una, las otras se pueden abrir y utilizar la consola, pero no se podrá usar su vty hasta que no la desbloquee.

Desde luego, bloquear una consola prevendrá que alguien manipule su trabajo, pero no previene de reinicios de la máquina u otras formas de deteriorar su trabajo.

Linux es un sistema operativo multiusuario real: puede haber varios usuarios trabajando simultáneamente con él, cada uno en su terminal. Esto obliga a tener en cuenta medidas de seguridad adicionales. Además, según hablan las estadísticas, el mayor porcentaje de violaciones de un sistema son realizadas por usuarios locales. Pero no sólo hay que protegerse de las violaciones intencionadas, sino que el sistema debe protegernos de operaciones accidentales debidas a descuidos o ignorancia de los usuarios.

En este aspecto de la seguridad, Linux dispone de todas las características de los sistemas Unix: un control de acceso a los usuarios verificando una pareja de usuario y clave; cada fichero y directorio tienen sus propietario y permisos.

Por otro lado, la meta de la mayoría de los ataques es conseguir acceso como root, lo que garantiza un control total sobre el sistema. Primero se intentará conseguir acceso como usuario "normal" para

posteriormente ir incrementando sus niveles de privilegio utilizando las posibles debilidades del sistema: programas con errores, configuraciones deficientes de los servicios o el descifrado de claves cifradas. Incluso se utilizan técnicas denominadas "ingeniería social", consistentes en convencer a ciertos usuarios para que suministren una información que debiera ser mantenida en secreto, como sus nombres de usuario y contraseñas.

En este apartado de seguridad local pretendemos dar unas ideas generales de los riesgos existentes, mecanismos para su solución y unas directrices de actuación que deberían convertirse en hábitos cotidianos.

### *Cuentas de usuario, grupos*

Cada usuario del sistema está definido por una línea en el fichero `/etc/passwd` y cada grupo por otra línea en el fichero `/etc/group`. Cada usuario pertenece a uno o varios grupos y cada recurso pertenece a un usuario y un grupo. Los permisos para un recurso se pueden asignar al propietario, al grupo y a otros (resto de los usuarios). Ahora bien, para mantener un sistema seguro, pero funcional, tendremos que realizar las combinaciones necesarias entre el propietario y grupo de un recurso con los permisos de los propietarios, grupos y otros. Por ejemplo, la unidad de disco flexible tiene las siguientes características:

```
brw-rw-r-- 1 root floppy 2,0 may 5 1998 /dev/fd0
```

- \* Propietario: root con permiso de lectura y escritura.
- \* Grupo: floppy con permiso de lectura y escritura.
- \* Otros: resto de los usuario con permiso de lectura.

Cuando queramos que un usuario pueda escribir en la unidad de disco, sólo tendremos que incluirlo en el grupo floppy. Cualquier otro usuario que no pertenezca al grupo floppy (salvo root) sólo podrá leer el disquete.

El administrador tiene que conocer las necesidades de cada uno de sus usuarios y asignarle los mínimos privilegios para que pueda realizar su trabajo sin resultar un peligro para otros usuarios o el sistema. Más abajo veremos otro mecanismo para poder utilizar un recurso sobre el cual no tenemos privilegios. No se asuste. Los valores que traen por defecto las distribuciones de Linux son suficiente para mantener el sistema seguro. Otro peligro potencial para el sistema es mantener cuentas abiertas que se han dejado de utilizar. Estas cuentas pueden constituir un buen refugio para un potencial atacante y pasar desapercibidas sus acciones. La seguridad de una sola cuenta puede comprometer la seguridad de todo el sistema. Esto es una realidad ante la que debemos protegernos.

Por un lado tenemos que asegurarnos de que nuestros usuarios utilizan claves sólidas:

- \* No deben ser una palabra conocida.
- \* Deberían contener letras, números y caracteres especiales.
- \* Deben ser fáciles de recordar y difíciles de adivinar.

Para comprobar que este requisito se verifica en nuestro sistema, podemos usar los mismos mecanismos que utilizan los atacantes. Existen varios programas que van probando varias palabras de diccionario, claves habituales y combinaciones de caracteres, que son cifrados con el mismo algoritmo que usa el sistema para mantener sus claves; después son comparadas con el valor de la clave cifrada que queremos averiguar hasta que el valor obtenido de un cifrado coincide con una clave cifrada. Posteriormente notificaremos al usuario que su clave es débil y le solicitaremos que la modifique. Usando este mecanismo, al menos podemos garantizar que no estaremos en inferioridad de condiciones con respecto a los atacantes locales.

Un conocido programa para realizar el descifrado de claves es *John the Ripper*.

Por otro lado, las claves cifradas se almacenan en el fichero `/etc/passwd`. Cualquier usuario del sistema tiene permiso de lectura sobre este fichero. Lo que es peor, agujeros en los navegadores permiten que se puedan leer ficheros arbitrarios de una máquina (evidentemente, que el usuario de navegador tenga permiso para leer), de manera que lleguen hasta un hacker que cree páginas web que exploten estos agujeros. Entonces puede parecer a primera vista que nos encontramos con un grave agujero de seguridad. El atacante, una vez obtenido el fichero `/etc/passwd` no tiene más que ejecutar su programa revienta claves favorito y sentarse a esperar hasta que empiecen a aparecer nombres de usuario con sus respectivas contraseñas, algo que suele pasar muy rápidamente. Con suerte, si el administrador es ingenuo o dejado, incluso dará con la clave del root, abriéndosele así las puertas a la máquina objetivo. Para solucionar esta vulnerabilidad, podemos recurrir a contraseñas en la sombra (shadow passwords), un mecanismo consistente en extraer las claves cifradas del fichero `/etc/passwd` y situarlas en otro fichero llamado `/etc/shadow`, que sólo puede leer el root y dejar el resto de la información en el original `/etc/passwd`. El fichero `/etc/shadow` sólo contiene el nombre de usuario y su clave, e información administrativa, como cuándo expira la cuenta, etc. El formato del fichero `/etc/shadow` es similar al siguiente:

usuario : clave : ultimo : puede : debe : aviso : expira : desactiva : reservado

- \* usuario: El nombre del usuario.
- \* clave: La clave cifrada
- \* ultimo: Días transcurridos del último cambio de clave desde el día 1/1/70
- \* puede: Días transcurridos antes de que la clave se pueda modificar.
- \* tiene: Días transcurridos antes de que la clave tenga que ser modificada.
- \* aviso: Días de aviso al usuario antes de que expire la clave.
- \* expira: Días que se desactiva la cuenta tras expirar la clave.
- \* desactiva: Días de duración de la cuenta desde el 1/1/70.
- \* reservado: sin comentarios.

Un ejemplo podría ser:

julia : gEvm4sbKnGRlg : 10627 : 0 : 99999 : 7 : -1 : -1 : 134529868

El paquete de Shadow Passwords se puede descargar desde cualquiera de los siguientes sitios, con instrucciones para su instalación:

- \* <ftp://i17linuxb.ists.pwr.wroc.pl/pub/linux/shadow/shadow-current.tar.gz>
- \* <ftp://ftp.icm.edu.pl/pub/Linux/shadow/shadow-current.tar.gz>
- \* <ftp://iguana.hut.fi/pub/linux/shadow/shadow-current.tar.gz>
- \* <ftp://ftp.cin.net/usr/ggallag/shadow/shadow-current.tar.gz>
- \* <ftp://ftp.netural.com/pub/linux/shadow/shadow-current.tar.gz>

Para activar contraseñas en la sombra, tiene que ejecutar `pwconv` como root; acción que creará su fichero `/etc/shadow`. Si su distribución de Linux no incluye contraseñas en la sombra o encuentra alguna dificultad para incorporar esta característica, existe un documento HOWTO (CÓMO) titulado `Shadow-Password-HOWTO` que le puede resultar de gran utilidad. Aquí podrá encontrar también información adicional que le puede ayudar a mantener su seguridad local. Hasta ahora hemos visto diversas situaciones en las que podemos aumentar la seguridad usando una clave. Piense que tiene que recordar cada una de las claves que utiliza, piense que **NO debe anotar NUNCA su clave en un papel (y menos pegarla a la pantalla)**. En alguna situación olvidar una clave puede ser un serio problema.

## El bit SUID/SGID

En muchas ocasiones un proceso necesita ejecutarse con unos privilegios mayores (o



menores) que el usuario que lo lanzó. Por ejemplo, un usuario puede modificar su propia clave con el mandato passwd, pero esto implica modificar el fichero /etc/passwd, y para esto un usuario "de a pie" no tiene permiso. ¿Cómo se soluciona? Pues activando el bit SUID del comando passwd (nótese que cuando esto sucede, la x de ejecutable pasa a ser una s):

```
ls -la /usr/bin/passw*
```

```
-r-sr-xr-x 1 root bin 15613 abr 27 1998 /usr/bin/passwd
```

Esto quiere decir que cuando se ejecute, el proceso correspondiente va a tener los privilegios del propietario del comando (es decir, el root), no del usuario que lo lanzó. En otras palabras, el proceso generado por passwd pertenece a root. A primera vista, esto puede parecer una seria brecha de seguridad. Y lo es. Si el programa funciona correctamente, no tiene por qué dar problemas; pero pequeños defectos en el programa pueden ser utilizados por alguien para tratar de ejecutar otro código distinto con los privilegios de este proceso. El método suele ser el desbordamiento de la pila (buffer overflow).

Cualquier atacante que haya entrado en un sistema de forma ilegítima intentará dejar una shell con el bit SUID para mantener ese nivel de privilegio cuando vuelva a entrar en el sistema.

SGID es lo mismo que SUID, pero aplicado al grupo.

Así pues, tenga cuidado con los programas con el bit SUID/SGID. Puede encontrarlos con

```
root# find / -type f \( -perm -04000 -o -perm -02000 \) -print
```

Tenga en cuenta que algunos programas (como passwd) tienen que tener el bit SUID. Compruebe en los lugares habituales, (que indicamos en la sección correspondiente) que ninguno de los programas propiedad del root o SUID que utiliza en su sistema, tiene un fallo de seguridad conocido que pueda ser explotado. Nunca debe permitir que quede una shell SUID corriendo en el sistema. Si el root deja desatendida la consola durante unos instantes (recuerde, debe utilizar siempre xlock), un intruso podría escribir lo siguiente:

```
# cp /bin/sh /tmp/cuenta-root  
# chmod 4755 /tmp/cuenta-root
```

creándose una versión SUID de la shell sh. En el futuro, cuando el atacante ejecute ese programa, cuenta-root, ¡se convertirá en root! Si lo escondiera en un directorio oculto, la única forma de encontrarlo sería escaneando el sistema completo como se ha explicado antes. Y recuerde, nunca escriba guiones de shell SUID.

## Seguridad del root

Los peores destrozos de un sistema es probable que no vengan de ningún cracker, o de un malévolo intruso. En muchísimas más ocasiones ha sido el propio administrador el que ha destrozado el sistema. Sí, el root. ¿Por qué? Por descuido, por exceso de confianza, por ignorancia. Evitar este problema no es difícil. Siguiendo unas fáciles normas, podrá protegerse de Vd. mismo:

\* No use la cuenta de root por norma. Evítela. Intente primero cualquier acción como un usuario normal, y si ve que no tiene permiso, piense porqué y use el comando su si es necesario.

\* Ejecute los comandos de forma segura verificando previamente la acción que va a realizar. Por ejemplo si quiere ejecutar rm borrar.\*, ejecute primero ls borrar.\* y si es lo que pretende modifique

el mandato y ejecútelo.

\* Ciertos mandatos admiten una opción (-i) para actuar de forma interactiva. Actívela, si no lo está ya añadiendo estas líneas a su fichero de recursos por la shell:

```
o alias rm='rm -i'  
o alias cp='cp -i'  
o alias mv='mv -i'
```

Siempre puede evitar estas preguntas, a veces incordiosas, con el mandato yes, cuando esté seguro de la operación que está realizando:

```
$ yes s|rm borrar.*
```

\* Como ya deben saber, el directorio actual no está, por defecto, en la ruta de búsqueda de ejecutables (PATH). Esto garantiza que no lanzaremos, sin darnos cuenta, un ejecutable que esté en el directorio actual llamado, por ejemplo ls.

\* Evite que la clave del root viaje por una red sin cifrar. Utilice ssh u otro canal seguro.

\* Limite los terminales desde los que se puede conectar root. Es preferible limitarlo a la consola del sistema. Esto se puede decidir en /etc/securetty. Si necesita una sesión remota como root, entre como usuario normal y luego use su.

\* Actúe de forma lenta y meditada cuando sea root. Sus acciones podrían afectar a un montón de cosas. ¡Piénselo antes de teclear!

Hay herramientas como sudo que permiten a ciertos usuarios utilizar comandos privilegiados sin necesidad de ser root, como montar o desmontar dispositivos. Además registra las actividades que se realizan, lo que ayuda a determinar qué hace realmente este usuario.

Linux tiene la gran ventaja de tener disponible el código fuente del núcleo; en realidad Linux propiamente dicho es sólo el núcleo. Esto nos permite la posibilidad de crear núcleos a medida de nuestras necesidades. Y parte de nuestras necesidades será la mejora de la seguridad.

Para compilar el núcleo primero tendremos que configurar las opciones que nos interesen. Los fuentes del núcleo se guadan habitualmente en el directorio /usr/src/linux, y una vez situados en él, tendremos que ejecutar «make menuconfig» (o «make xconfig» si estamos en modo gráfico). Así nos aparecen todas las opciones de configuración. Dentro de ellas nos vamos a fijar en las que están relacionadas con la seguridad, viendo una breve explicación de lo que hacen y cómo se usan.

Como el núcleo controla las características de red de su sistema, es importante que el núcleo tenga las opciones que garanticen la seguridad y que el propio núcleo no pueda ser comprometido. Para prevenir algunos de los últimos ataques de red, debe intentar mantener una versión del núcleo actualizada. *Puede encontrar las nuevas versiones del núcleo en The Linux Kernel Archives.*

Una de las características más interesantes del núcleo Linux es la posibilidad de realizar enmascaramiento de direcciones. Con esta técnica podremos dar acceso a Internet a una red local con direcciones privadas de forma transparente, es decir, sin ningún tipo de modificación en la configuración de las aplicaciones clientes, a diferencia de los proxies clásicos. Consiste en que el sistema Linux que posee la conexión hacia el exterior recibe las peticiones de conexión desde los equipos de la red local que tienen direcciones no válidas para Internet. El equipo Linux rehace la petición poniendo su propia dirección IP y modificando el puerto al que tiene que responder el

equipo remoto. Cuando Linux recibe la respuesta del equipo remoto, mira el puerto al que va destinado y vuelve a rehacer el paquete para enviarlo al equipo concreto de la red local que solicitó la conexión. De esta forma podemos mantener un nivel aceptable de protección para los equipos de la red local, ya que sólo podrán recibir respuestas a peticiones que ellos mismos originaron.

## Opciones de compilación del núcleo

### IP: Drop source routed frames (CONFIG\_IP\_NOSR)

Esta opción debería estar activada. Source routed frames contienen la ruta completa de sus destinos dentro del paquete. Esto significa que los enrutadores a través de los que circula el paquete no necesitan inspeccionarlo, y sólo lo reenvían. Esto podría ocasionar que los datos que entran a su sistema puedan ser un exploit potencial.

### IP: Firewalling (CONFIG\_IP\_FIREWALL)

Esta opción es necesaria si va a configurar su máquina como un cortafuegos, hacer enmascaramiento o desea proteger su estación de trabajo con línea telefónica de que alguien entre a través de su interfaz PPP. Con esta opción activa podremos usar el filtrado de paquetes en el propio núcleo del sistema, decidiendo el tráfico que llega o sale de nuestro equipo.

### IP: forwarding/gatewaying (CONFIG\_IP\_FORWARD)

Si activa reenvío IP (IP forwarding), su Linux esencialmente se convierte en un encaminador (router). Si su máquina está en una red, podría estar enviando datos de una red a otra, y quizás saltándose un cortafuegos que esté puesto allí para evitar que esto suceda. A los usuarios con un puesto aislado y conexión telefónica les interesará desactivar esta característica. Otros usuarios deberían pensar en las implicaciones de seguridad de hacer esto en su caso concreto. Las máquinas que actúen como cortafuegos tendrán que activar esta característica y usarla junto al software cortafuegos.

Puede activar y desactivar el reenvío IP (IP forwarding) dinámicamente usando el siguiente comando:

```
root# echo 1 > /proc/sys/net/ipv4/ip_forward
```

y desactivarlo con el comando:

```
root# echo 0 > /proc/sys/net/ipv4/ip_forward
```

Ese fichero (y muchos otros ficheros de /proc) aparecerá con longitud cero, pero en realidad no es un fichero en el sentido clásico, sino que son datos guardados en memoria.

### IP: firewall packet logging (CONFIG\_IP\_FIREWALL\_VERBOSE)

Esta opción le suministra información sobre los paquetes que su cortafuegos recibe, como remitente, destinatario, puerto, etc. Así podremos rastrear los orígenes de los posibles intentos de ataque.

### IP: always defragment (CONFIG\_IP\_ALWAYS\_DEFRAG)

Generalmente esta opción está desactivada, pero si está construyendo un host cortafuegos o para enmascaramiento, deberá activarla. Cuando se envían paquetes de un host a otro, no siempre se envían como simples paquetes de datos, sino que se fragmentan en varios trozos. El problema es que los números de puerto sólo se almacenan en el primer fragmento. Esto significa que alguien puede insertar información en el resto de los paquetes para su conexión que se supone que no deberían estar allí.

### IP: syn cookies (CONFIG\_SYN\_COOKIES)

El ataque SYN es un ataque de denegación de servicio (denial of service, DoS) que consume todos los recursos de su máquina forzando un reinicio. No podemos encontrar ninguna razón por la que no debiera activar esto.

## **Seguridad del núcleo**

### Dispositivos del núcleo

Hay algunos dispositivos de bloque y carácter disponibles en Linux que también le resultarán útiles

para mantener la seguridad de sus sistema.

Los dos dispositivos `/dev/random` y `/dev/urandom` los proporciona el núcleo para generar datos aleatorios en cualquier instante. Por ejemplo, se utilizan para iniciar un número de secuencia para conexiones TCP/IP.

Ambos, `/dev/random` y `/dev/urandom`, deberían ser suficientemente seguros como para generar claves PGP, SSH y otras aplicaciones donde son un requisito números aleatorios seguros para generar claves válidas para una sesión. Los atacantes no deberían ser capaces de determinar el siguiente número dada cualquier secuencia de números con este origen. Se han dedicado muchos esfuerzos para garantizar que los números que se obtienen de esta fuente son pseudoaleatorios en todos los sentidos de la palabra pseudoaleatorio.

La única diferencia es que `/dev/random` suministra bytes aleatorios y le hace esperar para que se acumulen más. Observe que en algunos sistemas puede bloquear durante un rato a la espera de que se genere una nueva entrada de usuario al sistema. Por tanto debe tener cuidado al usar `/dev/random`. (Quizás lo mejor que puede hacer es usarlo cuando esté generando información sensible de claves e indicarle al usuario que pulse una tecla repetidas veces hasta que indique por la pantalla "Ya es suficiente").

`/dev/random` tiene gran calidad de entropía, midiendo tiempos entre interrupciones, etc. Bloquea hasta que hay disponibles suficientes bits de datos aleatorios.

`/dev/urandom` es parecido, no es tan seguro, pero suficiente para la mayoría de las aplicaciones. Puede leer los dispositivos usando algo parecido a lo siguiente:

```
root# head -c 6 /dev/urandom | uuencode -
```

Esto imprimirá seis caracteres aleatorios en la consola, válidos para la generación de una clave.

La seguridad de las conexiones en red merecen en la actualidad una atención especial, incluso por medios de comunicación no especializados, por el impacto que representan los fallos ante la opinión pública. El propio desarrollo tanto de Linux, como de la mayoría del software que lo acompaña, es de fuentes abiertas. Podemos ver y estudiar el código. Esto tiene la ventaja de que la seguridad en Linux no sea una mera apariencia, sino que el código está siendo escrutado por muchas personas distintas que rápidamente detectan los fallos y los corrigen con una velocidad asombrosa. Si además comprendemos los mecanismos que se siguen en las conexiones en red, y mantenemos actualizados nuestros programas, podemos tener un nivel de seguridad y una funcionalidad aceptables. Tampoco tienen las mismas necesidades de seguridad un equipo doméstico, con conexiones esporádicas a Internet, que un servidor conectado permanentemente y que actúe como pasarela entre una intranet e Internet.

Para describir las pautas de actuación seguras iremos examinando cómo actúan las conexiones y cómo podemos protegerlas.

## **xinetd**

Para atender las solicitudes de conexión que llegan a nuestro equipo existe un demonio llamado `inetd` que está a la escucha de todos los intentos de conexión que se realicen a su máquina. Cuando le llega una solicitud de conexión irá dirigida a un puerto (número de servicio, quizás sea más claro que puerto), por ejemplo, el 80 sería una solicitud al servidor de páginas web, 23 para telnet, 25 para smtp, etc. Los servicios de red que presta su máquina están descritos en `/etc/xinetd.conf` (y en `/etc/services` los números de puertos). Esto quiere decir que, cuando llegue una solicitud de

conexión al puerto 110 (pop3) se ejecutará el programa `/usr/sbin/tcpd ipop3d`. Sin embargo, el servicio `imap` está deshabilitado (está comentado con un `#` delante), por lo que el sistema no le responde. El siguiente paso es `/usr/sbin/tcpd`, que es el `tcp_wrapper`: un servicio que verifica el origen de las conexiones con su base de datos `/etc/hosts.allow` (equipos autorizados) y `/etc/hosts.deny` (equipos a los que se les deniega la conexión). `tcpd` anota todos los intentos de conexión que le llegan en `/var/log/secure` para que tenga la posibilidad de saber quién intenta conectarse a su máquina y si lo consigue. Si `tcpd` autoriza la conexión, ejecuta `ipop3d` que es el programa que realmente atiende la conexión, ante el cual se tiene que validar el usuario mediante una clave. Observe que ya llevamos tres niveles de seguridad: prestar un servicio, autorizar un conexión y validar un usuario.

Un consejo que es conveniente seguir: No tenga abiertos los servicios que no necesita; esto supone asumir un riesgo a cambio de nada. Tampoco limite la funcionalidad del sistema, si tiene que usar un servicio, hágalo sabiendo lo que hace.

También hay que asegurarse de que el programa `ipop3d` no tenga ninguna vulnerabilidad, es decir, que esté actualizado. Existen numerosos medios para estar al día de las vulnerabilidades que aparecen. Una buena lista de correo o una revista electrónica tal vez sean la forma más rápida de conocer los incidentes, las causas y las soluciones. Posiblemente la mejor lista de correo para el mundo Unix sea `Bugtraq` (busque en forums).

Pero esto no es todo, además puede filtrar las conexiones que le lleguen desde el exterior para que ni siquiera alcancen a los `tcp_wrappers`. Por ejemplo, en el caso de conexiones a Internet por módem:

```
ipchains -A input -j DENY -s 0/0 -d $4/32 23 -p tcp -i ppp0 -l
```

poniendo la anterior línea en el fichero `/etc/ppp/ip-up` (y `ipchains -F input` en `ip-down`) estaríamos añadiendo (`-A`) un filtro de entrada (`input`), que deniega (`-j DENY`) desde cualquier sitio de internet (`-s 0/0`) dirigidas a nuestro equipo (`-d $4/32`) al puerto telnet (23) por `tcp` (`-p tcp`) que lleguen desde internet (en este caso `-i ppp0`) y que además las anote en el registro de incidencias (`-l`) (`$4` es la dirección IP que obtenemos dinámicamente).

El mecanismo de filtrado de conexiones se realiza en el núcleo del sistema operativo y si ha sido compilado con estas opciones. Normalmente lo está. Este filtrado se realiza a nivel de red y transporte: cuando llega un paquete por un interfaz de red se analiza siguiendo los filtros de entrada. Este paquete puede ser aceptado, denegado o rechazado, en este último caso se avisa al remitente. Si los filtros de entrada aceptan el paquete, pasa al sistema si era su destino final o pasa por los filtros de reenvío o enmascaramiento, donde se vuelve a repetir una de las acciones. Por último, los paquetes que proceden del propio sistema o los que han sido aceptados por los filtros de reenvío o enmascaramiento pasan al filtro de salida. En cualesquiera de estos filtros se puede indicar que lo anote en el registro de incidencias.

## **Registro y conocimiento de incidencias**

A parte de todo esto, puede conocer las incidencias que ocurren durante el funcionamiento del sistema. Por un lado conviene familiarizarse con los procesos que se ejecutan habitualmente en una máquina. Es una buena costumbre ejecutar periódicamente `ps axu`. Cualquier cosa extraña debiéramos aclararla. Puede matar cualquier proceso con la orden `kill -9 pid` (o `killall -9 nombre_proceso`). Pero en caso de ataque activo, lo mejor es desconectar de la red inmediatamente, si es posible, claro está.

Después tenemos los registros de incidencias (las ubicaciones pueden ser diferentes dependiendo del sistema, pero no radicalmente distintas) de `/var/log`.

Trabajando en modo texto se puede hacer en una consola virtual (como root)  
tail -f /var/log/messages y tail -f /var/log/secure

y de esta forma podemos ir viendo las incidencias del sistema. Conviene también familiarizarse con las anotaciones que aparecen habitualmente para diferenciarlas de las que puedan presentar un problema. En modo gráfico hay un programa llamado ktail que le muestra las incidencias de una forma similar a la anterior.

## **Comunicaciones seguras**

Por último, nos interesará mantener unas comunicaciones seguras para garantizar la privacidad e integridad de la información. Actualmente existen las herramientas necesarias para cada necesidad. Podemos usar cifrados simétricos como pgp y gpg para documentos, correo electrónico y comunicaciones sobre canales inseguros

Podemos crear canales de comunicación seguros de distinto tipo:

- \* SSH (Secure Shell), stelnets: SSH y stelnets son programas que le permiten efectuar conexiones con sistemas remotos y mantener una conexión cifrada. Con esto evitamos, entre otras cosas, que las claves circulen por la red sin cifrar.

- \* Cryptographic IP Encapsulation (CIPE): CIPE cifra los datos a nivel de red. El viaje de los paquetes entre hosts se hace cifrado. A diferencia de SSH que cifra los datos por conexión, lo hace a nivel de socket. Así una conexión lógica entre programas que se ejecutan en hosts diferentes está cifrada. CIPE se puede usar en tunnelling para crear una Red Virtual Privada. El cifrado a bajo nivel tiene la ventaja de poder hacer trabajar la red de forma transparente entre las dos redes conectadas en la RVP sin ningún cambio en el software de aplicación.

- \* SSL: o Secure Sockets Layer, fue diseñado y propuesto en 1994 por Netscape Communications Corporation junto con su primera versión del Navigator como un protocolo para dotar de seguridad a las sesiones de navegación a través de Internet. Sin embargo, no fue hasta su tercera versión, conocida como SSL v3.0 que alcanzó su madurez, superando los problemas de seguridad y las limitaciones de sus predecesores. En su estado actual proporciona los siguientes servicios:

o Cifrado de datos: la información transferida, aunque caiga en manos de un atacante, será indescifrable, garantizando así la confidencialidad.

o Autenticación de servidores: el usuario puede asegurarse de la identidad del servidor al que se conecta y al que posiblemente envíe información personal confidencial.

o Integridad de mensajes: se impide que modificaciones intencionadas o accidentales en la información mientras viaja por Internet pasen inadvertidas.

o Opcionalmente, autenticación de cliente: permite al servidor conocer la identidad del usuario, con el fin de decidir si puede acceder a ciertas áreas protegidas.

## **Consejos finales**

Limite las acciones que realice como root al mínimo imprescindible, y sobre todo no ejecute programas desconocidos. Por ejemplo, en un juego (el quake) que distribuía una revista había un programa llamado runme que enviaba por mail las características de la máquina a una dirección de correo. En este caso se trataba de un troyano inofensivo, pero ofrece una idea de lo que puede hacer un programa ejecutado sin saberse lo que hace. Linux también tiene la posibilidad de proporcionar acceso transparente a Internet a una red local mediante IP masquerade. En este caso, si usamos direcciones de red privadas, nos aseguramos que los equipos de la red interna no son accesibles desde Internet si no es a través del equipo Linux. También podemos instalar un servidor proxy con caché, que a la vez que actúa de filtro de conexiones a nivel de aplicación, puede acelerar el acceso a servicios desde la red local.

A menudo, el mayor enemigo del sistema es el propio administrador del sistema, sí, tiene todos los privilegios y cualquier acción puede ser irreversible y hacerle perder posteriormente mucho más tiempo que el que hubiera perdido por realizar las tareas de forma segura. Puede borrar cualquier fichero e incluso destruir el propio sistema, mientras que un usuario «normal» sólo puede perjudicarse a sí mismo. Por estos motivos, conseguir privilegios de root es la meta de cualquier ataque.

Tampoco hay que alarmarse. Piense que en un sistema operativo monousuario cualquiera podría darle formato al disco duro y perder toda la información almacenada o borrar cualquier fichero necesario para el funcionamiento del sistema. En un sistema estilo Unix, como Linux, esto sólo lo podría hacer el usuario root.

## **Hábitos seguros**

La seguridad del administrador es simple, en mayor medida consiste en tener unos hábitos seguros y también en utilizar herramientas seguras.

Una primera norma que siempre debería tener presente es usar la cuenta de root sólo para realizar tareas concretas y breves y el resto hacerlo como usuario normal. Es una costumbre muy peligrosa usar todo el tiempo la cuenta de root. Al principio, a los usuarios de Linux les gusta sentir todo el poder de la cuenta de root, les molesta que su propio sistema les deniegue el permiso para hacer algo, pero van cambiando de opinión poco a poco, conforme se van familiarizando con el sistema o cuando han realizado un destrozo de esos que nos hacen proferir insultos contra nosotros mismos acompañados de un desesperado puñetazo en la mesa (o en el teclado). Piense que cuando el sistema le deniega alguna operación es porque puede conllevar algún riesgo. El sistema le avisa para que piense dos veces lo que está haciendo.

En los casos de tareas que necesiten privilegios de administrador para realizar una operación concreta, podemos usar la orden «su» (Super Usuario) o también «sudo». De esta forma podremos acceder a los privilegios de root sólo cuando nos interese.



\* Asegúrese de que en los borrados de ficheros por parte del root se pide confirmación. Esto lo puede hacer poniendo «alias rm="rm -i"». Esto es lo habitual para el root. Si en alguna ocasión tiene que borrar muchos ficheros y no quiere confirmar cada uno de ellos, puede usar la opción «-f» para no pedir confirmación, deshacer el alias con «alias rm=rm» o bien usando la orden «yes», poniendo «yes|rm ficheros» para ir confirmando automáticamente cada una de las preguntas de la orden.

\* Procure prever los resultados de una orden, sobre todo si usa comodines, intentándolo antes una forma no irreversible. Por ejemplo, si quiere borrar todos los ficheros terminados en «~» ejecute primero «ls -la \*~» y una vez que haya verificado a qué va a afectar la orden, ejecute «rm \*~».

\* Vigile la variable de entorno «PATH». Limite la búsqueda automática de ejecutables a las ubicaciones estándar del sistema. De forma particular evite incluir el directorio actual, es decir «.», en esta búsqueda. Bastaría incluir un ejecutable llamado «ls» en un directorio para que usted mismo ejecute un código desconocido con privilegios de root, y cuando se dé cuenta de lo que ha hecho sea demasiado tarde.

Además, no tenga directorios con permiso de escritura en su ruta de búsquedas, ya que esto puede permitir a un posible atacante modificar o poner nuevos binarios que se puedan ejecutar como root la próxima vez que ejecute una determinada orden.

\* No utilice las herramientas rlogin/rsh/rexec como root. Pueden ser objeto de diversos tipos de ataques y es peligroso ejecutarlas como root. Nunca cree un fichero .rhosts para root. Evite que la clave del root circule por la red sin cifrar. Si tiene la necesidad de ofrecer servicios de shell o ejecución remotas sobre un canal inseguro utilice «ssh» en lugar de «telnet» u otra herramienta que no cifre los datos de las conexiones. Los servicios remotos como «rlogin», «rsh» y «rexec», como dijimos antes, no suelen ser seguros si se utilizan canales no seguros. Es mejor deshabilitarlos.

\* Limite las ubicaciones desde donde alguien se puede conectar como root al sistema. En el fichero /etc/securetty puede especificar la lista de terminales desde las cuales se puede conectar el root. Las terminales predeterminadas para conectarse como root sólo incluyen las consolas virtuales (vtys). Si tuviera que conectarse como root desde una ubicación distinta a la consola, hágalo como usuario normal primero y luego utilice «su» para acceder a los privilegios de root. De esta forma un posible atacante tendría que conocer el nombre de un usuario del sistema, conocer su clave y también conocer la clave del root. Esto pone más dificultades para obtener privilegios remotos en su sistema.

\* Evite siempre dar la clave de root, no lo haga bajo ningún concepto por mucha confianza que tenga con esa persona. Si tiene que otorgar privilegios a algún usuario para realizar alguna tarea de administración, como montar un CD u otro dispositivo similar, utilice «sudo» para permitirlo con la propia clave del usuario. Así puede decidir qué usuario tiene acceso a una determinada orden.

\* No modifique los permisos de un fichero o directorio si no sabe realmente qué está haciendo. Los valores que trae la instalación de las distintas distribuciones suelen ser adecuados.

\* Jamás, insistimos, jamás se conecte a un servicio IRC como usuario root.

La seguridad es un proceso continuo, que requiere tener previsto hasta lo imprevisible. Tener unos buenos hábitos y tomar unas pequeñas precauciones nos ayudarán mucho.

### **Determinar los servicios activos**

Desactive todos los servicios que no vaya a prestar, en particular revise los ficheros /etc/inittab, /etc/inetd.conf y los demonios que se lanzan durante el arranque. Si no está realmente seguro de lo que hace, mejor no haga nada; las distribuciones más modernas incorporan unos mínimos de seguridad aceptables para un usuario medio.

No tiene sentido tener abierto un servicio del que no va a hacer uso ningún usuario legal. Puede que esté consumiendo recursos de su sistema para ofrecer a algún atacante la posibilidad de violarlo.

Puede usar un analizador de puertos para ver qué parte de su sistema es visible desde el exterior. Existen utilidades como SATAN, Nessus o nmap que realizan esta labor.

Trinux es una minidistribución de Linux totalmente portable que se puede llevar en 2 ó 3 disquetes y se ejecuta por completo en RAM, pudiéndose usar desde cualquier equipo para la red. Se arranca desde el disquete y no utiliza el disco duro para nada. Contiene las últimas versiones de algunas herramientas muy prácticas enfocadas a la seguridad en redes. Nos permitirá analizar el tráfico de la red, analizar puertos e incluso ver el contenido de los paquetes que circulan por la red.

### **Proteger los ficheros importantes**

Existe un parche para el núcleo Linux que impide que ciertos ficheros puedan ser modificados, incluso por el propio root. El núcleo parcheado de esta forma puede garantizarnos la integridad de la información almacenada incluso en el caso de que alguien consiguiera privilegios de root en nuestro sistema. Este parche se puede obtener, junto con la información necesaria para su instalación, en LIDS.

Si no queremos aplicar el parche, sí que deberíamos vigilar los permisos de ficheros y directorios.  
Software actualizado

La gran mayoría del software que acompaña a Linux es de código fuente público, como el propio núcleo. Esto es una garantía de seguridad en sí. Cientos de expertos analizan minuciosamente el código para detectar alguna pega que poder publicar en las listas de correo sobre seguridad más prestigiosas, y se corrigen con gran rapidez. De esta forma nos garantizamos un software de calidad y no una mera seguridad aparente. Esto por otro lado nos obliga a ir sustituyendo las versiones defectuosas por otras corregidas y que mejoran las prestaciones. En cualquier sistema operativo, mantener un software que ha demostrado tener fallos supone asumir un riesgo innecesario. Para estar actualizado consulte los recursos de información sobre seguridad en Linux.

### **Prevenir pérdidas de información**

Existen acontecimientos de los que nos puede resultar muy difícil protegernos como son los desastres naturales, únicamente podremos seguir una serie de pasos para evitar que su incidencia sea lo menor posible. La mejor solución es mantener un buen conjunto de copias de seguridad sobre toda la información necesaria del sistema. Hay que pensar que las copias de seguridad no sólo nos protegen de desastres naturales, también de los desastres que pueda ocasionar algún intruso en nuestro sistema, de cualquier ataque a la disponibilidad o integridad de la información del sistema.

Si los datos tienen un tamaño inferior a 650Mb, puede ser una buena opción grabarlos en un CD, bien permanente (ya que es más difícil de falsificar con posterioridad, y si están almacenados de forma adecuada pueden durar mucho tiempo) o regrabable. Las cintas y otros medios sobre los que se puede escribir deberían protegerse tan pronto como se completa la copia y se verifica para evitar la falsificación. Tenga cuidado y almacene su copia de seguridad en un sitio seguro. Una buena copia de seguridad le asegura que tiene un buen punto desde el que restaurar su sistema.

Hay que insistir en la seguridad de las copias de seguridad. Si las copias de seguridad no están almacenadas en un sitio seguro, puede que el posible intruso no tenga necesidad de idear métodos sofisticados para obtenerla, si le basta con copiar o sustraer un CD.

### **Características de las copias de seguridad**

Cuando se realice una copia de seguridad es conveniente seleccionar un método que garantice la conservación de las características de la información como son derechos y permisos. Si realizamos

una copia de seguridad de una forma o sobre un soporte que no contemple esta posibilidad, si tenemos que restaurar los datos sobre el sistema el resultado sobre la seguridad y funcionalidad globales puede ser impredecible.

### *Secuencia de Copias*

Es necesario tener una política de copias de seguridad adecuada a las características de la entidad que estamos gestionando. Quizás el mejor método es el de rotación de cintas. Pasamos a verlo con un ejemplo.

Un ciclo de seis cintas es fácil de mantener. Esto incluye cuatro cintas para la semana, una cinta para cada Viernes y una cinta para los Viernes impares. Se realiza una copia incremental cada día, y una copia completa en la cinta adecuada de cada Viernes. Si hace algún cambio importante o añade datos importantes a su sistema también sería adecuado efectuar una copia.

### *Copiar las Bases de Datos del Sistema*

Existe cierta información del sistema que es imprescindible para su correcto funcionamiento. Es conveniente tener una copia de estos ficheros en una ubicación segura. En particular resulta conveniente tener una copia del contenido del directorio /etc. También hay que mantenerla en lugar seguro, ya que tiene copias de los ficheros /etc/passwd y /etc/shadows, entre otros que puedan contener claves de usuarios que no están cifradas.

También en cada sistema se puede tener una base de datos de las aplicaciones que hay instaladas en el servidor. Cada distribución dispone de alguna herramienta que nos realiza el mantenimiento de la base de datos a la misma vez que instala o desinstala aplicaciones. La pérdida de esta base de datos nos haría perder el control sobre qué aplicaciones tenemos instaladas.

En muchas situaciones también será necesario tener copia de seguridad de los ficheros de registro de incidencias, para tener constancia de las distintas actividades del sistema.

## Consejos

- \* Suscribirse a las listas de correo de alertas de seguridad para estar actualizado.
- \* Prestar atención a los ficheros de registro.
- \* Actualizar el software inseguro.
- \* Verificar regularmente la integridad de los ficheros con algún software como tripwire.
- \* Tener unas copias de seguridad adecuadas.
- \* Utilizar PGP o GnuPG para garantizar la autenticidad y la privacidad.
- \* Verificar con periodicidad los puertos de los equipos.
- \* Revisar periódicamente las cuentas de usuario.
- \* Asignar cuotas de uso de recursos del sistema.
- \* Mantener los terminales seguros.
- \* Asegurarse de tener claves sólidas.
- \* Mantener el sistema de ficheros con propietarios y permisos adecuados.
- \* Instalar cortafuegos.

## En resumen

Ahora, una vez vistas las características generales de seguridad, lo que queda es aplicar el sentido común. Tenemos que ver nuestra situación y respondernos a una serie de preguntas:

- \* ¿Qué queremos proteger?
- \* ¿Qué valor tiene lo que queremos proteger?
- \* ¿Qué coste tiene la seguridad?
- \* ¿De quién nos queremos proteger?
- \* ¿Cuáles son los puntos débiles de nuestro sistema?

Las posibles respuestas a estas preguntas nos propocionan un abanico de posibilidades demasiado amplio como para poderlo tratar todo.

Lo primero que tenemos que determinar es lo que queremos proteger. No será lo mismo una estación de trabajo personal aislada con conexiones a Internet esporádicas que un servidor web con conexión permanente o un cortafuegos.

También tendremos que considerar el coste de lo que queremos proteger: posible coste económico, tiempo de restauración o instalación, prestigio, perdida de clientes, etc. También el coste de la seguridad en unos términos parecidos a los anteriores. Sería absurdo que invirtiéramos más en protección que el coste de lo protegido.

También hay que considerar que existe una relación inversa entre seguridad y funcionalidad. Cuanto más seguro hacemos un sistema, menos funcional resulta, ofreciendo menos servicios y más limitaciones de acceso. Esto también constituye otro coste adicional: facilidad de uso.

Después de saber qué y de qué tenemos que protegernos, de quiénes y cuáles son sus posibles objetivos, y viendo los servicios que necesariamente hay que prestar o usar, obtendremos un esquema elemental de nuestra situación y de las medidas que tenemos que tomar.

Ahora vamos a ver qué se puede hacer en caso de haber sufrido o estar sufriendo un ataque.

No es una situación agradable, y aunque siempre sería preferible que no hubiera sucedido, conviene tener en mente una serie de normas que nos permitan una actuación rápida y certera que disminuya las consecuencias del incidente. Como norma general hay que conservar la calma. No conviene tomar medidas apresuradas que puedan aumentar el impacto del ataque. Vamos a distinguir una serie de situaciones posibles y cómo se debe actuar. Una vez visto esto nos queda aplicar el sentido común.

## Detección de un ataque activo

Nos ponemos en situación: acabamos de detectar un ataque que está actualmente en curso.

El ataque puede ser de diversa naturaleza. Dejaremos aparte los casos genéricos como detectar alguien manipulando físicamente el ordenador.

### Ataque local

Cuando detectamos un ataque local tendremos que verificar la identidad del atacante. No conviene sacar conclusiones precipitadas y culpar a alguien de atacar el sistema cuando sólo puede que sea una negligencia a la hora de seleccionar una clave o abandonar abierta una consola.

Hay que verificar el origen de la conexión, los registros del sistema y los procesos que tiene activos. Tendremos que comprobar si son los habituales y qué es lo que se sale de lo normal. Después nos dirigiremos a esa persona, por teléfono o personalmente, para preguntar qué está haciendo y pedir que cese en la actividad. Si no tiene una conexión activa y no tiene idea de lo que le estamos diciendo, habrá que profundizar en la investigación porque cabe la posibilidad de que alguien haya utilizado esa cuenta de forma ilegítima. Si reconoce el incidente, que le informe de los mecanismos que ha utilizado, las acciones que ha realizado y actúe en consecuencia.

Nunca se precipite para hacer acusaciones. Recopile todas las pruebas que haya detectado en los registros, procesos, modificaciones de información, etc. Sea rápido, pero seguro. Está en juego su sistema y su prestigio.

### Ataque en red

Si el ataque se produce a través de la red podemos tener distintas situaciones. En general puede ser conveniente espiar un poco al intruso para obtener más pruebas y después desconectar el interfaz de red si es posible. Si no fuera posible desconectar el interfaz, deberíamos usar algún filtro para las conexiones procedentes de la dirección del atacante. Programas como ipchains (o ipfwadm en su caso) pueden realizar esta labor. Si desconectamos el interfaz o denegamos (no rechazar) los paquetes procedentes de esa dirección el intruso lo podría interpretar como un error de red, más que una detección del ataque. Si no se pudiera limitar el acceso a las direcciones que usa el intruso, intente cerrar la cuenta del usuario. Observe que cerrar una cuenta no es una cosa simple. Tiene que tener en cuenta los ficheros .rhosts, el acceso FTP y otras posibles puertas traseras.

En general no es aconsejable apagar el sistema. Por supuesto, nunca apagarlo en caliente; esto podría hacernos perder la información que tenemos en memoria. En Linux podemos ver la lista de procesos que hay en ejecución y matar aquellos que puedan estar dañando al sistema.

¿Somos el destino del ataque o somos un punto intermedio?

Se puede dar la situación que nuestra máquina no sea el destino final del ataque. Puede que el intruso la haya utilizado como punto intermedio para atacar a otros sistemas e intentar dificultar el seguimiento de las pistas. En este caso, además de limitar las acciones del atacante deberíamos notificarlo al administrador del destino del ataque y conservar todas las pruebas existentes por si se pudieran reclamar judicialmente.

En cualquier caso, si queremos dar validez legal a las pruebas obtenidas, sería conveniente la intervención judicial.

Es habitual que durante los próximos minutos el atacante vuelva a intentar continuar con sus acciones, tal vez usando una cuenta diferente y/o una dirección de red distinta.

### El ataque ha concluido

Ha detectado un compromiso que ya ha ocurrido o bien lo ha detectado mientras ocurría y ha echado al atacante fuera de su sistema.

Ahora viene la parte más dura del incidente: tratar de dejar el sistema mejor que estaba antes de que ocurriera.

Tapar el agujero

Determine los medios que usó el atacante para acceder a su sistema. Deberá analizar cuidadosamente los ficheros de registro del sistema. En ellos debería haber una información valiosa para seguir la pista de las actividades del intruso en nuestra máquina. Las causas más habituales son una mala configuración de algún servicio, un programa defectuoso o la negligencia de algún usuario con respecto a su clave de acceso.

Compruebe por los cauces más conocidos, que se pueden consultar en la página sobre recursos de seguridad bajo Linux, la existencia de algún nuevo «exploit» que pueda ser la causa u otros fallos que tenga que corregir.

Si no elimina al atacante, probablemente volverá. No sólo a su máquina, sino a cualquiera otra de la red. Durante sus incursiones ha podido utilizar algún «sniffer», y disponer de información suficiente para tener acceso a otras máquinas locales.

Si sospecha que el atacante ha obtenido copias de los ficheros `/etc/passwd`, `/etc/shadow`, `/etc/ppp/pap-secrets`, `/etc/ppp/chap-secrets` o cualquier otro fichero que contenga datos de usuarios y claves, sería conveniente modificarlas todas. Si tiene distintos usuarios en su máquina, oblíguelos a cambiar su clave. En general es preferible cambiar siempre las claves después de un incidente, una vez que sepamos que lo hacemos de una forma segura.

Verifique si se han modificado las limitaciones al acceso a distintas herramientas de administración remota como `linuxconf`. Puede que el atacante trate de abrir alguna puerta trasera para continuar aprovechándose de nuestras máquinas.

En algunos casos puede interesar antes de nada, hacer alguna copia de todo el disco duro para seguir investigando el incidente en otra máquina distinta que no esté conectada a la red y no perder una información que puede ser valiosa.

Evaluación de los efectos del ataque

El siguiente paso que hay que realizar es la evaluación de los efectos que ha tenido el ataque. Tiene que tener en mente la naturaleza del ataque para evaluar los efectos. Si ha sido una denegación de servicio es probable que el atacante no haya tenido acceso a la información local. Si tenía instalado algún programa, estilo `Tripwire`, que verifica la integridad, su trabajo ahora sería más cómodo. En caso contrario tendrá que verificar todos sus datos importantes. Verifique las fechas de creación de los ficheros binarios y si detecta alguna incongruencia con la fecha de instalación puede empezar a sospechar. Si tiene la posibilidad, compare los tamaños de los ficheros con otro sistema «limpio» y por supuesto, no trate de verificar los programas ejecutándolos como `root`.

Una buena alternativa es volver a instalar el sistema. Guarde los ficheros de configuración para tener una funcionalidad idéntica a la previa al ataque. En Linux, los ficheros de configuración se almacenan en modo texto por lo que no son susceptibles de contener caballos de Troya. Eso sí, debería verificar que las configuraciones son las originales y no han sido manipuladas por el atacante. Reinstale el sistema y utilice las copias de seguridad para reponer los datos de los usuarios.

Hay que tener especial cuidado con las copias de seguridad. Tiene que estar seguro de que las copias de seguridad que está utilizando son previas a cualquier ataque. No se arriesgue a restaurar unas copias de seguridad que pudieran tener algún caballo de Troya; tenga un cuidado especial con los ficheros binarios que restaure.

Avisar

Si cree que ha sido objeto de un ataque que no está documentado, debería notificarlo a alguna organización de seguridad como CERT o similar para que se pueda solucionar lo antes posible y evitar que otros sistemas lo puedan padecer.

Y aunque sea un hecho documentado con anterioridad, no dude en pedir consulta a alguna de la múltiples lista de correo que tratan temas de seguridad en general y de Linux en particular.

Si ha conseguido información sobre el atacante, se lo debería notificar al administrador del dominio del intruso. Puede buscar este contacto con whois, con la base de datos del Internic. Podría enviarles un mensaje de correo con todos los registros relacionados con el incidente, fechas y horas. Si conoce alguna otra información sobre su intruso, podría mencionarla también. En ciertas situaciones, tras enviar el correo podría llamar por teléfono al administrador del sistema que originó el incidente. Si el administrador localiza a su atacante, podría hacerle las cosas mucho más fáciles.

Los buenos hackers con frecuencia usan sistemas intermedios. Algunos (o muchos) puede que ni sepan que han sido comprometidos. Intentar seguir la pista de un cracker hasta su origen puede ser difícil. Siendo educado con los administradores, le puede facilitar la obtención de la ayuda necesaria.

De todas formas, esperamos que la lectura de este capítulo sea totalmente innecesaria, si ha seguido unas normas adecuadas de seguridad.

Con esta lista de recursos perseguimos dos objetivos fundamentales, tener una buena guía para saber dónde consultar y obtener información, y por otro lado, obtener la información sobre las novedades que van ocurriendo, nuevos fallos descubiertos, los exploits que aparecen y los mecanismos para poderlos solucionar.

En SeguriNet se puede encontrar una traducción en castellano de la Guía de Seguridad del Administrador de Linux. En ella tenemos una documentación valiosa donde completar aspectos más concretos, así como una buena lista de recursos.

Por otro lado en CICA tenemos la mejor lista de correo sobre seguridad en castellano que hay sobre linux. No es un foro de discusión, no está pensada para realizar consultas, sino para recibir de forma rápida y clara todos los avisos sobre nuevas vulnerabilidades que se detectan en Linux, los efectos que pueden tener y cómo se pueden solucionar.

En las páginas del GLUB hay una buena recopilación de software y documentación de seguridad en Linux en castellano.

RedIris es otro lugar que también dispone de información valiosa y listas sobre seguridad y Linux.

Recursos Web y ftp

Distribuciones

- \* Caldera OpenLinux
- \* RedHat
- \* Guía de errores de RedHat
- \* Lista de archivos de seguridad de RedHat
- \* SuSE
- \* Listas de correo de SuSE
- \* Debian
- \* Slackware
- \* Fórum Slackware

- \* TurboLinux
- \* Stampede GNU/Linux
- \* Mandrake
- \* LinuxPPC
- \* Linux Pro
- \* LinuxWare
- \* MKLinux
- \* Yggdrasil
- \* Conectiva
- \* DLD
- \* Pacific HiTech TurboLinux Errata Guide
- \* Eagle Linux M68K
- \* Eurielec
- \* Kheops Linux
- \* MNIS Linux
- \* Trinux (una minidistribución enfocada hacia la seguridad)

## Red / Análisis de Host / Intrusion

- \* nmap
- \* ftpcheck, relaycheck
- \* cheops
- \* nessus
- \* saint
- \* SBScan
- \* Samba scanner (funciona con windows)
- \* HUNT
- \* SATAN y otras herramientas

## Administración

- \* rpm
- \* Guiones de actualización RedHat RPM
- \* Logcheck
- \* bgcheck
- \* COAS
- \* Webmin
- \* Linuxconf
- \* super
- \* YaST

## Cortafuegos

- \* ipchains configuración más fácil
- \* Configuración de un cortafuegos mediante un CGI
- \* ipchains y otros

## Servicios de Red

- \* sendmail
- \* apache
- \* BIND/DHCPD/DHCPD
- \* secure syslog
- \* openssl



- \* ncftp, ncdftp
- \* qmail
- \* postfix
- \* ProFTPD
- \* rsync
- \* PPP
- \* Servicios de internet
- \* versión libre de LDAP
- \* Lenguaje de programación para generar HTML dinámico

## Soluciones para Redes Virtuales Privadas

- \* CIPE
- \* ECLiPt
- \* Productos hardware RedCreek VPN (IPSec)
- \* Controladores de tarjetas RedCreek VPN para Linux
- \* Desarrollo PPTP para Linux
- \* FreeS/WAN, un desarrollo IPSec para Linux

## SSL

- \* SSLeay
- \* Desarrollo open SSL
- \* Aplicaciones para SSL
- \* Ftp oficial
- \* Desarrollo Apache SSL libre

## Servicios/Cifrado de datos

- \* Desarrollo libre de SSH
- \* Sustituto libre de PGP
- \* CFS

## Herramientas de Seguridad

- \* audit
- \* Varios paquetes de software de seguridad

## Kernel

- \* stackguard
- \* rule based access control

## Detección de intrusos

- \* port sentry
- \* host sentry
- \* Network Flight Recorder

## Grupos de respuestas sobre seguridad

- \* USA - CERT
- \* Información sobre seguridad en castellano
- \* INTERNATIONAL - HERT

- \* AUSTRALIA - AUSCERT
- \* SINGAPORE - SINGCERT
- \* Winn Schwartau's site
- \* L0pht
- \* Root Shell
- \* Infowar UK
- \* buenos documentos

#### Programas comerciales para copias de seguridad

- \* BRU
- \* Quickstart
- \* Arkeia
- \* CTAR
- \* CTAR:NET
- \* Backup Professional

#### Aplicaciones

- \* Página de actualización diaria y búsqueda
- \* Prácticamente todos los paquete RPM
- \* Un gran listado de aplicaciones

#### Varios

\* Security and Encryption: Probablemente la mejor lista de recursos sobre seguridad en internet. Aquí podrá encontrar casi todo y algo más.

- \* Replay tiene archivos de muchos programas de seguridad.
- \* Wietse's
- \* Ftp de ES-CERT
- \* The Hacker FA
- \* Bugtraq
- \* First
- \* Hacking
- \* CPSR
- \* Underground
- \* Defcon
- \* El archivo COAST tiene un gran número de programas de seguridad Unix e información.
- \* Reptile tiene montones de buena información sobre seguridad en sus páginas.
- \* Infilsec tiene un motor de vulnerabilidad que puede decirle cuales afectan a una plataforma específica.
- \* CIAC envíos periódicos de seguridad sobre exploits comunes.
- \* Un buen punto de inicio para Linux Pluggable Authentication Modules (PAM).

#### Listas de correo

- \* Bugtraq: Para suscribirse a bugtraq, envíe un mail a [listserv@netspace.org](mailto:listserv@netspace.org) que contenga en el cuerpo del mensaje `subscribe bugtraq`. (ver el enlace anterior para los archivos).
- \* CIAC: Envíe un e-mail a: [majordomo@tholia.llnl.gov](mailto:majordomo@tholia.llnl.gov)  
En el cuerpo (no en el subject) del mensaje ponga: `subscribe ciac-bulletin`.