
**Herramientas de seguridad en FreeBSD**
Este artículo pretende dar una idea general de las características de seguridad de FreeBSD, un sistema operativo Unix libre. FreeBSD contiene una serie de opciones que pueden ser configuradas para reducir el riesgo de compromiso del sistema. Un sistema FreeBSD bien configurado puede ofrecer muy buenos servicios a los usuarios y ser muy resistente a posibles ataques.

**Introducción**
Cualquier administrador de sistemas Unix o network manager en la necesidad de trabajar con servidores de red estables y de alto rendimiento debería considerar la posibilidad de integrar alguno de los sistemas libres y considerarlos como alternativa a los sistemas operativos comerciales convencionales. Mientras FreeBSD y sistemas similares ofrecen una alternativa de bajo coste en estaciones de trabajo y servidores, desde el punto de vista de la seguridad no está tan claro. Antes de introducirnos en la exploración de FreeBSD, quisiera comentar que la relación entre los sistemas Unix y la seguridad en Internet es ya un tópico. Existen numerosos libros y otros recursos que discuten los diferentes elementos implicados en este tema, algunos de los cuales están incluidos en las referencias citadas al final de este artículo. En particular, al lector interesado le recomiendo "Practical Unix and Internet Security" y "Firewalls and Internet Security".
OpenBSD, un sistema relativamente cercano a FreeBSD, está siendo desarrollado con el objetivo de reforzar la seguridad de sistemas basados en BSD. FreeBSD incorpora e incorporará las soluciones a problemas de seguridad identificados por OpenBSD. Más información de OpenBSD en:

http://www.openbsd.org/

**Sumario de conceptos de seguridad**
En general, un sistema debe ser tan seguro físicamente como sea necesario, y el sistema solo debe tener los servicios y programas privilegiados que sean estrictamente necesarios para el nivel de operaciones deseado. Las siguientes secciones tratan diferentes áreas de seguridad con más detalle.

**Sistema físico y cónsola**
Con acceso físico al sistema, un atacante puede seleccionar la opción -s en el prompt de arranque y obtener un shell con privilegios de root en modo monousuario. Para evitar esto, si el sistema no está físicamente en un lugar seguro, es una buena idea marcar la consola como insegura en /etc/ttys para que el sistema requiera el password de root al entrar en modo monousuario. Mira el man de init(8) para más detalles. Si, como decimos, el sistema no se encuentra en un lugar seguro, no debería tener disquetera. Un atacante puede usar el boot prompt para arrancar un kernel retocado o montar el floppy como la partición raíz, pasando por alto cualquier protección establecida en /etc/ttys.

**Programas setuid y Daemons**
Los problemas con demonios y programas setuid son explotados habitualmente por los crackers para obtener privilegios de root en los sistemas. Es aconsejable tener instalados el menor número posible de daemons y programas setuid en los sistemas para reducir los posibles problemas en caso de que se descubra algún tipo de vulnerabilidad en éstos.

**Daemons**
La idea con los daemons es desactivar todos y cada uno de ellos que no sean estrictamente necesarios. Edita los ficheros /etc/rc.conf y /etc/inetd.conf para desactivar los servicios no usados. Por ejemplo, si los servicios rlogin o rsh no son necesarios, pueden ser desactivados en /etc/inetd.conf comentándolos con el símbolo #. De la misma manera, si el daemon de impresión lpr no es necesario, pon la opción lpd_enable en el fichero /etc/rc.conf a "NO".
Otra estrategia es reemplazar los daemons que son conocidos por haber tenido problemas de seguridad por otros que nos provean de los mismos servicios pero que hayan sido desarrollados teniendo en cuenta la seguridad. Por ejemplo, mientras en FreeBSD se mantiene el sendmail relativamente actualizado, podríamos reemplazarlo por el programa qmail, diseñado y escrito enteramente pensando en la seguridad. Qmail está disponible en http://www.qmail.org/.

**Programas setuid**
Los programas setuid son programas especiales en el sistema que se ejecutan con los privilegios del propietario del programa, sea cual sea el usuario que lo esté ejecutando. Por ejemplo el programa PPP /usr/sbin/ppp realiza algunas operaciones que solo están permitidas al usuario root, por lo que tiene activado el bit de setuid, y su propietario es root. Como resultado, cuando cualquier usuario ejecuta ppp, éste se ejecuta con los permisos de seguridad del usuario root. Mira el man de chmod(1) para más información sobre el bit de setuid y otros permisos. Los programas setuid son un problema de seguridad, por que algunos programas tienen bugs que permiten a los crackers aprovecharse de ellos. Provocar problemas de overrun y sobreescribir el stack con código maligno es un sistema común de ataque en programas setuid. Desde el momento en que el programa se está ejecutando con privilegios de root, el código del cracker se ejecuta con los privilegios de root. La manera más sencilla de revisar los programas setuid y setgid en el sistema es ejecutar los comandos:

find / -perm -4000 -print
find / -perm -2000 -print

Usando la lista generada de ficheros setuid y setgid, puedes determinar los que no son necesarios para tu sistema. Usa el comando "chmod gu-s nombre_fichero" para desactivar los bits de setuid y setgid.

**Sistemas de ficheros**
Un sistema donde los sistemas de ficheros de los usuarios están separados de los sistemas de ficheros que contienen los programas ejecutables y ficheros de dispositivos pueden parar ataques potenciales o "back doors". Un método de obtener acceso de root o dejar una "back door" para obtener ese acceso posteriormente es crear un ejecutable setuid o crear un fichero especial de dispositivo que permita acceso de escritura a la memoria del sistema. Las opciones nosuid y nodev pueden ser usadas en /etc/fstab en sistemas de ficheros que no deban contener setuid o ficheros de dispositivos. Mira el man de mount(8) y fstab(5).
Estas opciones pueden ser usadas en sistemas de ficheros en los cuales los usuarios pueden crear ficheros. Es muy buena idea separar los directorios que contengan ficheros de usuarios, como /home, /tmp, /var/tmp en un sistema de ficheros separado del /usr. Las opciones nosuid y nodev son especialmente importantes en sistemas que monten unidades NFS de sistemas inseguros o no fiables. Un usuario avanzado del otro sistema podría crear un ejecutable setuid-root o un dispositivo /dev/kmem-equivalent en el servidor, y usar a continuación ese fichero en el cliente para obtener privilegios de root. La opción noexec puede ser muy útil en un sistema de ficheros montado desde un servidor no fiable.

**Monitorizando el sistema de ficheros**
Algunos ataques a sistemas se realizan mediante el sistema de ficheros Unix, normalmente abusando de daemons o programas setuid. Típicamente estos programas crean o cambian ficheros de configuración, como el fichero de usuario .rhosts, permitiendo el acceso al sistema. Estos ataques se previenen cerrando "agujeros" en los daemon y programas setuid, pero estos agujeros deben ser encontrados primero. Es recomendable reducir la posibilidad de estos ataques o su detección inmediata monitorizando el sistema.
FreeBSD envía diariamente un informe de seguridad al usuario root. Chequea los posibles nuevos o modificados programas setuid y ficheros de dispositivos. Si aparece un nuevo programa setuid, la hora o el tamaño cambia misteriosamente, o aparece un nuevo UID 0 (root), es probablemente el momento de investigar. Pero, si un sistema ha sido sabiamente crackeado, los informes no indicarán ningún problema, por que el script de seguridad y sus ficheros de datos son vulnerables a ser falseados. Solo tripwire (mira más adelante), cuando está correctamente instalado y usado, puede detectar de manera fiable que un sistema ha sido comprometido. Desafortunadamente, debido a la manera en la que el sistema es escaneado por programas setuid, es posible que el resultado del informe cambie, causando una falsa alarma ocasional en la sección de "setuid files and devices" del informe de seguridad.

**Monitorización avanzada con tripwire**
Tripwire es un package que monitoriza el sistema de ficheros para prevenir los cambios en ficheros y permisos. Las instrucciones de instalación del programa nos dan una excelente visión del package y del procedimiento correcto de instalación. Esencialmente, tripwire debería ser instalado e inicializado en un sistema totalmente seguro y no comprometido. El fichero ejecutable de tripwire, lincado como un ejecutable estático para evitar el uso de las librerías compartidas del sistema, y su fichero de base de datos, que contiene las firmas digitales que verifican la integridad de los ficheros del sistema, deberían ser escritos en un soporte de solo lectura. Después, cuando tripwire es usado para verificar la integridad de un sistema, debería ser invocado directamente desde su ejecutable en el soporte de solo lectura. Una posibilidad es instalar el ejecutable y la base de datos en un floppy de 3.5" y de solo lectura.

**rhosts y hosts.equiv**

Si los servicios de red de shell remota (rshell) o login remoto (login) están activados en /etc/inetd.conf, los ficheros /etc/hosts.equiv y .rhosts son importantes para la seguridad. El fichero hosts.equiv define los sistemas que tienen usuarios reconocidos por nuestro sistema para que no les sea requerido el password al realizar la conexión. Cada fichero $HOME/.rhosts define los sistemas remotos y usuarios que pueden usar esta cuenta sin tener que dar un password. Mira el man de hosts.equiv(5) para más información, pero esencialmente, ningún nombre de sistema debería figurar en estos ficheros a no ser que los hosts y redes sean de total y absoluta confianza.
Ten en cuenta que si la línea + + aparece en estos ficheros, el sistema es totalmente inseguro y cualquier usuario de cualquier sistema puede hacer un login a tu sistema sin necesidad de password.
Algunas formas de ataque se basan en crear un fichero .rhosts en el directorio de usuario o añadir datos a un fichero .rhosts ya existente. Parece que este tipo de ataque se lleva a cabo cada vez con menor frecuencia desde que se han mejorado las comprobaciones de propietarios y permisos que realizan rlogin y rsh antes de permitir el acceso. Aun así, este ataque es aun posible.
Un método que elimina completamente este problema es desactivar los servicios rlogin y rsh comentando o borrando las líneas referentes a login y shell en /etc/inetd.conf aunque a veces, la desactivación de estos servicios no es lo más apropiado. Por ejemplo, el protocolo shell es usado por otros servicios como rdump.
Un método más aceptable de limitar la exposición de servicios es limitar los sistemas a los cuales permitimos realizar rlogin y rsh. Esta limitación está implementada en el daemon TCP wrappers de Weitse Venema. Mira en la lista de packages de FreeBSD para más información. Otra solución es añadir la opción -l a los servicios rlogind y rshd en las entradas apropiadas de /etc/inetd.conf para forzar la autenticación usando cualquier fichero .rhosts diferente al de root.

**Ficheros de dispositivos especiales**
Los ficheros de dispositivos especiales dan acceso directo al hardware del sistema a través de nombres en el sistema de ficheros, como /dev/mem para acceso a la memoria del sistema. El acceso al hardware crítico debería ser fuertemente restringido. Por ejemplo, si /dev/mem fuese de libre lectura para todo el mundo, el password de un usuario podría ser observado leyendo el segmento de datos del programa de login mientras este está funcionando.
Un ataque a través de ficheros de dispositivos puede realizarse usando una vulnerabilidad en un daemon o en un programa setuid para crear un dispositivo /dev/kmem-equivalent, y a continuación escribir en el kernel a través del nuevo dispositivo creado y así poder incrementar el nivel de privilegio del shell del intruso. Si el fichero de dispositivo no está protegido correctamente, el trabajo del intruso para aumentar sus privilegios es realmente sencillo.
Los ficheros de dispositivos son correctamente protegidos durante la instalación de FreeBSD, pero deberíamos verificarlos.

**Protocolos de red**
Existen diferentes tareas a realizar en los protocolos de servicios de red y en el propio TCP/IP. Aquí tienes un breve sumario de soluciones referentes a la seguridad:
Desactivar los servicios innecesarios en los ficheros /etc/inetd.conf y /etc/rc.conf.
Filtrado de los paquetes que entran en nuestra red desde el exterior y que contengan direcciones internas en el campo origen. Estos paquetes son obviamente originados por un equipo mal configurado o por un intento de spoofing.
Usar el daemon TCP wrappers en los servicios TCP arrancados vía inetd para comprobar y filtrar el origen de las peticiones. Verificar la dirección IP de origen y el nombre del host remoto, e implementar restricciones de acceso basadas en las direcciones IP de origen.
El package "ssh secure shell" ofrece para los logins remotos y transferencia de ficheros la encriptación de todos los datos que viajan por la red. Solo por esta razón, ssh ya es una gran alternativa para telnet, rlogin y ftp, especialmente para la administración remota. La encriptación de todos los datos entre el origen y el destino nos previenen de los posibles sniffers instalados en la red. El protocolo "Netwok File System" (NFS), que permite compartir sistemas de ficheros entre sistema remotos, tiene diferentes vacíos de seguridad. Si es usado, es aconsejable usarlo a través de un firewall (ten en cuenta que FreeBSD puede funcionar como filtro de paquetes). Si NFS es usado en entorno hostil (por ejemplo, conectado a Internet sin filtrado de paquetes vía router o firewall), el comando fsirand(8) debería ser usado para hacer aleatoria la creación de números de inodes.
Como NFS, el "Network Information Service" (NIS, también conocido como "Paginas amarillas") tiene diferentes vacíos de seguridad. NIS es útil para la distribución de información de cuentas en una red de sistemas Unix. Al contrario que NFS, NIS no usa un rango particular de puertos, por lo que no puede ser ocultado tras un firewall. El NIS de FreeBSD usa el fichero de configuración /var/yp/securenets para limitar el acceso de clientes a la información.

**Características de FreeBSD**
FreeBSD ofrece diferentes características no habituales en el resto de sistemas Unix.

**Niveles de seguridad**
Los sistemas Unix sufren con el todo poderoso acceso concedido al usuario root. Una vez un cracker obtiene privilegios de root, el sistema completo se vuelve vulnerable al cracker. Los sistemas derivados de 4.4BSD ofrecen una nueva característica de seguridad llamada "niveles de seguridad del sistema" (más información en el man de init(8)). Usada correctamente, esta característica puede prevenir la introducción de caballos de Troya y back doors en los ejecutables del sistema y la modificación de los ficheros de configuración.
Los niveles de seguridad del sistema son: -1, modo permanentemente inseguro; 0, modo inseguro - no hay protecciones adicionales activadas; 1, modo seguro - ficheros protegidos contra la modificación y ficheros de dispositivos no abiertos para escritura; 2, modo altamente seguro - protección de nivel 1 más dispositivos de disco no abiertos para escritura; 3, modo de seguridad extendida - protección de nivel 2 más filtrado de paquetes IP. El nivel de seguridad por defecto -1 es llamado "modo de inseguridad permanente". Los niveles por encima de 0 causan la desactivación en el kernel de las siguientes operaciones:

- Activación de los flags de ficheros "immutable" y "append-only".
- Acceso directo de escritura a los dispositivos de disco montados en el nivel 1 de seguridad, o a los dispositivos de disco en nivel de seguridad 1 o superiores.
- Escritura a los ficheros de dispositivo /dev/mem y /dev/kmem
- Modificaciones a cualquier fichero con el flag "immutable" activado.
- Cargar cualquier modulo de kernel.
- Cambios a las listas de filtrado de paquetes IP (en nivel 2 o superiores).

Cuando un sistema FreeBSD es instalado usando la opción make world en el código fuente, una serie de ficheros son instalados con el flag "immutable" activado. En general, los ficheros ejecutables setuid de root son instalados con el flag "immutable". En cualquier momento, el flag schg puede ser activado con el comando chflags, pero el flag solo puede ser eliminado cuando el sistema está en el nivel de seguridad \0001 o 0. Durante el funcionamiento en modo multiusuario, el nivel de seguridad debería ser aumentado. El comando para aumentar el nivel de seguridad a 1 es:

sysctl -w kern.securelevel 1

El nivel de seguridad del sistema puede ser aumentado automáticamente en el arranque del sistema incluyendo el correspondiente comando sysctl en el fichero /etc/rc.local. El aumento de los niveles de seguridad comporta una limitación en las actividades de gestión de los ficheros, por lo que no es aconsejable usar esta opción en sistemas frecuentemente cambiantes. Una vez el nivel de seguridad ha sido aumentado, la única manera de disminuirlo es rearrancar el sistema. Si necesitas realizar tareas de mantenimiento importantes en el sistema, éste puede ser arrancado en modo monousuario, donde el nivel de seguridad por defecto es de -1.

**IP Firewall**
Muchos de los protocolos TCP/IP incluidos en FreeBSD, como NFS y NIS, son mucho más seguros si no pueden ser accedidos desde el exterior de la parte segura de la red. Para protocolos que no puedan ser restringidos de ninguna otra manera, un filtro de paquetes que solo permita pasar los protocolos especificados puede ser una buena idea para bloquear el acceso externo.
La opción del kernel de FreeBSD IPFIREWALL activa la opción de IP firewalling, la cual aplica reglas de filtrado sobre los paquetes entrantes o salientes de la máquina. Cuando un kernel compilado con IPFIREWALL arranca, por defecto descarta todos los paquetes IP. Reglas de filtrado adicionales deben ser establecidas para aceptar el tráfico deseado en los dos sentidos (entrada y salida). El script de arranque /etc/rc.firewall contiene diferentes ejemplos de filtrado. Los tipos de filtrado son seleccionados con la opción firewall en /etc/rc.conf, la cual incluye:

- Open - Sin limitación de ningún tipo. Cualquier paquete IP puede entrar y salir por cualquiera de las interfaces del sistema.
- Client - Protege un sistema que sea cliente en la red. Dado el nombre de la máquina, dirección IP y máscara, un filtro básico se establece para permitir todo el tráfico originado desde o destinado a la red local, permite paquetes para cualquier conexión TCP establecida, permite email entrante y cualquier conexión TCP saliente, y permite peticiones DNS y NTP por UDP. Cualquier otra cosa es denegada.
- Simple - Creación de un filtro de paquetes en un sistema que ruta entre diferentes redes. Dados los números de red, máscaras, y direcciones IP de la red externa (insegura) e interna (segura), esta opción en rc.firewall crea un filtro que:

- Previene la entrada de paquetes externos con el campo origen de la red interna, y evita la salida de paquetes que en su campo origen no contengan una dirección de la red interna.

- Permite todos los paquetes para conexiones tcp establecidas.

- Permite conexiones TCP para correo entrante, DNS y peticiones HTTP.

- Rechaza y "loguea" (vía syslogd(8)) todos los intentos de conexiones a servicios TCP diferentes de los permitidos anteriormente.

- Permite todas las peticiones y respuestas de DNS y NTP UDP.

- Deniega cualquier cosa no explícitamente permitida.

Si el nivel de seguridad del sistema es inferior a 2, no se permiten cambios en la lista de filtrado de paquetes. Esto permite al administrador de sistemas prevenir las posibles modificaciones en el listado de filtro de paquetes en el caso de que un sistema se vea comprometido. La opción IP firewall nos puede ayudar a monitorizar los posibles ataques que se produzcan a nuestro sistema. La opción del kernel IPFIREWALL_VERBOSE activa el log del filtrado de paquetes vía syslogd(8). La información recogida puede ser útil en la detección de escaneos de puertos o intentos de entrada no autorizados del sistema.

**One-Time Passwords**
El sniffing de passwords ha sido uno de los constantes problemas que se encuentran los administradores de red y sistemas. Los usuarios que realizan telnet o ftp remotos a través de Internet deberían ser avisados para que usen el sistema de "one-time password" para evitar que su password "reusable" pueda ser capturado y usado ilícitamente.
FreeBSD usa el software S/Key para proveer el servicio de "one-time password". Los usuarios pueden configurarse su "one-time password" en FreeBSD usando el comando:

keyinit

el cual preguntará por un password privado que solo conoce el usuario. Este password debería ser diferente del password usado regularmente, pero no es obligatorio. El keyinit mostrará algo como:

ID fred s/key is 99 sp99609

BUY OUR BOLD LEER YOKE COW

El usuario puede ahora generar una serie de one-time passwords para usar en sus accesos remotos usando el comando:

key -n 10 98 sp99609

El cual preguntará por el password privado usado en el comando keyinit anterior, generando los passwords para los próximos 10 logins. El usuario puede imprimir estos passwords y llevárselos consigo (por supuesto, la impresión de estos passwords comporta otra serie de implicaciones de seguridad). Un sistema FreeBSD puede requerir el uso de passwords S/Key para todos los logins remotos. El fichero /etc/skey.access define cuando debe ser usado el sistema S/Key. Por ejemplo, si el administrador confía en los sistemas de su propia red (172.16.xxx.xxx), pero quiere que todos los usuarios se vean obligados a usar S/Key desde cualquier sistema externo, podría usar un fichero /etc/skey.access como este:

permit 172.16.0.0 255.255.0.0

deny

Ten en cuenta que /etc/skey.access no afecta a todos los métodos de acceso remoto, solo a telnet, ftp y rlogin. Por ejemplo, ssh, pop daemons e imap daemons ignoran las restricciones S/Key.

**Desactivación de cuentas**
Si un password llega a manos de un cracker, el cracker intentará acceder a la cuenta comprometida a través de protocolos como telnet y ftp, lo que puede comprometer la seguridad global del sistema o sistemas donde resida

la cuenta comprometida.
Un administrador puede restringir completamente el acceso telnet o ftp desde una cuenta determinada o host, ajustando el fichero /etc/login.access. Esta opción puede reducir el peligro de passwords o cuentas comprometidas limitando los lugares desde los que esta cuenta es accesible. De todas maneras, la mejor solución, si es posible, es dar de baja la cuenta comprometida. Si esto no es posible por cualquier motivo, el paso mínimo a realizar es el cambio de password de dicha cuenta.
Ten en cuenta que login.access no afecta a todos los métodos de acceso remoto, solo a telnet, ftp y rlogin. Por ejemplo, ssh, pop daemons e imap daemons ignoran las restricciones establecidas en login.access.
Por ejemplo, si un administrador quiere permitir logins para todas sus cuentas desde dos redes, 172.16 y 192.168.32, y permitir a todos los usuarios incluidos en el grupo wheel hacer login desde la máquina con dirección 192.168.2.2, este es el fichero /etc/login.access que puede ser usado:

```
+:ALL:172.16.

+:ALL:192.168.32.

+:wheel:192.168.2.2

-:ALL:ALL
```

**Sendmail**
Desgraciadamente, durante los últimos tiempos se están produciendo ataques indiscriminados, que si bien no afectan a la propia seguridad del sistema, si afectan al rendimiento de éstos y al uso de recursos propios de manera no autorizada. Me estoy refiriendo a la nefasta técnica del spam, que se basa en usar servidores smtp ajenos para envíos masivos de correo no solicitado. Esta técnica empezó como algo anecdótico (de hecho, hasta hace poco tiempo, no se solía instalar ningún filtro en los servidores smtp confiando en la buena intención de los usuarios de Internet y administradores de sistemas conectados a ella), pero se está convirtiendo en una auténtica plaga. La política a seguir se basa en limitar el acceso a los servidores smtp de la siguiente manera:

- Aceptamos correo para cualquier destino con origen en nuestra/s red/es.
- Aceptamos correo de redes remotas con destino solo a nuestros usuarios/dominios.
- Si hacemos mail backup, aceptamos relay solo para dominios autorizados.

En versiones de sendmail 8.8.8 es aconsejable el uso de las check_rules de Claus Asmann, disponibles en la dirección:

http://www.informatik.uni-kiel.de/%7Eca/email/check.html

Desde hace muy pocos días está disponible la versión 8.9.0 release de sendmail que incluye toda una serie de herramientas anti-spam y anti-relay. Sendmail está disponible en la dirección:

http://www.sendmail.org

Solo decir que este problema no es específico de FreeBSD, si no que afecta a todas y cada una de las plataformas sobre las que funciona Sendmail.

**Conclusión**
Gracias al equipo de desarrollo, FreeBSD es un sistema sólido que puede ser configurado para aguantar ataques de seguridad importantes. Mientras que la instalación por defecto es todavía relativamente abierta y amigable para los usuarios, FreeBSD contiene muchos mecanismos de seguridad y packages añadidos que permiten aumentar la resistencia del sistema. Estos aumentos de seguridad pueden asegurar una operación y viabilidad del sistema, así como la confidencialidad de la información contenida en él.

**Agradecimientos**
Gracias al "core team" de FreeBSD y contribuyentes al proyecto FreeBSD, que han hecho de él un excelente y extremadamente estable sistema operativo. Gracias también a todos los desarrolladores que trabajan en los parches de seguridad disponibles días e incluso horas después de su descubrimiento. (El soporte de seguridad de FreeBSD online incluye ftp://ftp.freebsd.org/pub/FreeBSD/CERT/ para avisos "advisories" y parches, y la lista de distribución freebsd-security@freebsd.org.
Los archivos de las listas están disponibles en la dirección http://www.freebsd.org/search.html.)

**Referencias**

Steven M. Bellovin. Security problems in the TCP/IP protocol suite. Computer Communications Review, 19(2), April 1989.

CERT Advisory 94.01: Ongoing network monitoring attacks. [Online] ftp://info.cert.org/pub/cert_advisories/CA-94:01.network.monitoring.attacks, August 17 1997.

Bill Cheswick and Steve Bellovin. Firewalls and Internet Security: Repelling the Wily Hacker. Addison-Wesley, 1994.

Simson Garfinkel and Eugene H. Spafford. Practical UNIX and Internet Security. O'Reilly & Associates, Inc., Sebastapol, California, 1996. Second Edition.

Neil M. Haller. The S/KEY one-time password system. In Proceedings of the Internet Society Symposium on Network and Distributed System Security, San Diego, CA, February 1994.

Robert T. Morris. A weakness in the 4.2BSD UNIX TCP/IP software. Science of Computer Programming, February 25 1985.

Jennifer G. Steiner, B. Clifford Neuman, and Jeffrey I. Schiller. Kerberos: An authentication service for open network systems. In Proceedings of the Winter 1988 Usenix Conference, February 1988. (Version 4).

Wietse Venema. TCP wrapper: Network monitoring, access control, and booby traps. In Proceedings of the UNIX Security III Symposium. USENIX, September 1992.

**Nota:** este artículo está basado en un original escrito por Guy Helmer de la Iowa State University.