Implementación de seguridad con sistemas operativos y herramientas libres

Gunnar Wolf
Departamento de Seguridad en Cómputo, DGSCA, UNAM
FES Iztacala, UNAM
gwolf@campus.iztacala.unam.mx

Congreso de Seguridad en Cómputo 2001

Índice general

Capítulo 1

Definiciones básicas

1.1. ¿Qué es software libre?

Concepto

- No tener las restricciones que nos son comunes en el mundo *propietario*
- La ambigüedad de esta definición nos lleva a grandes problemas

La GPL

- Creada por Richard M. Stallman en 1985, tras el GNU Manifesto [1]
- Protegida por la Free Software Foundation [2]
- La Licencia Pública GNU (GPL) [3] marca que todo software protegido por esta licencia debe poder ser redistribuído libremente, siempre que vaya acompañado por su código fuente, y debe poder ser modificado, embebido, o reutilizado de cualquier manera, siempre que el resultado permanezca bajo la licencia GPL.

LaBSD

- Licencia académica, creada por la Universidad de California en Berkeley para las modificaciones que hicieron al código fuente original del Unix de AT&T, conocido como Berkeley Software Distribution [4]
- La licencia BSD permite la redistribución de los programas que protege bajo cualquier esquema, siempre y cuando se otorgue crédito a los autores originales. El espíritu de esta licencia puede apreciarse claramente en las políticas de copia del sistema operativo OpenBSD [5].

El OpenSource

- La Open Source Initiative [6] nace para evitar las confusiones comunes entre angloparlantes, al utilizar la misma palabra para *software libre* y *software gratuito* (*free* software), y presentarlo de una manera más aceptable para las entidades comerciales.
- Hoy en día hay fuertes diferencias entre el movimiento del Software Libre y los defensores del Open Source. Muchas licencias han sido aceptadas como Open Source [7] que no pueden ser aceptadas como software libre por diferentes razones [8].

1.2. ¿Qué es un sistema operativo?

1.2.1. Funciones del Sistema Operativo

- Abstracción del hardware
- Compartir los recursos justamente
- Proteger a todos los procesos de los demás procesos
- Proteger a los datos de todos los usuarios de los demás usuarios
- Asegurar la integridad de la información

1.2.2. Sistemas operativos libres y sus caracterísitcas principales

Linux [9]

- El sistema operativo libre más popular, y por mucho.
- Desarrollado completamente bajo un esquema cooperativo y no centralizado, lo que ha llevado a la aparición de muchas distribuciones
- Bajo casi todas las distribuciones, su seguridad es bastante débil en la configuración default
- Muy alta velocidad de desarrollo e inovación
- Soporte a una amplia gama de plataformas de hardware

FreeBSD [10]

- Su objetivo es crear un sistema operativo libre con la máxima estabilidad y eficiencia para plataformas Intel; recientemente agregó soporte para plataformas Alpha.

- Derivado de las fuentes originales de BSD, con una trayectoria de tres décadas de desarrollo
- El sistema derivado de BSD más popular, y por mucho
- Desarrollado abiertamente por un núcleo cerrado de desarrolladores, con contribución de los miembros de la comunidad
- Alta velocidad de desarrollo e inovación

OpenBSD [11]

- Reconocido sin lugar a dudas como el sistema operativo más seguro del mundo
- Derivado de NetBSD, el cual a su vez deriva de las fuentes originales de BSD, con una trayectoria de tres décadas de desarrollo
- Desarrollado abiertamente por un núcleo cerrado de desarrolladores, con contribución de los miembros de la comunidad
- Auditoría proactiva constante de seguridad en el código
- Primer sistema operativo en integrar el soporte a la criptografía fuerte como característica núcleo del sistema.
- Da mucha mayor importancia a la seguridad y estabilidad que a adoptar nuevas características
- Soporte a una amplia gama de plataformas de hardware

Sistemas menores

NetBSD Su objetivo es ser un sistema operativo libre que soporte tantas arquitecturas de hardware como sea posible [12]

AtheOS Busca crear un sistema operativo para escritorio eficiente, fácil de utilizar y GPL [13]

HURD Implementa un diseño avanzado de microkernel, se convertirá en el núcleo central del sistema GNU [14]

FreeDOS Reimplementación libre compatible con MS-DOS [15]

MIT-Exokernel Propone optimizar al sistema operativo para cada aplicación, logrando resultados sorprendentes [16]

1.3. ¿Qué es seguridad?

Seguridad es una palabra con una definición demasiado amplia, y aún entre expertos es difícil llegar a un acuerdo acerca de qué significa. En el ámbito informático, la seguridad equivale —principalmente— a garantizar los siguientes puntos:

Consistencia Comportarse como esperamos que se comporte y mantener su comportamiento sin cambios inesperados.

Servicio El sistema debe prestar todos los servicios que ofrecemos de manera confiable, constante y consistente.

Protección y

Si un programa tiene errores y sufre una caída, no debe afectar a la ejecución de otros procesos. Un programa diseñado expresamente para hacer daño debe tener un impacto mínimo en el sistema. Los segmentos de memoria de un proceso deben ser invisibles e inmodificables para cualquier otro proceso.

Control de acceso Los datos generados por un usuario no deben ser accesibles a otro usuario a menos que así sea específicamente solicitado por su dueño. Soportar diferentes modos de acceso a un archivo, de modo que el sistema pueda exigir que un archivo pueda ser leído pero no ejecutado o abierto para escritura. Los mecanismos de control de acceso deben ser tan granulares como sea posible.

Autenticación El sistema debe poseer los mecanismos necesarios para asegurarse que un usuario es quien dice ser y tiene suficientes privilegios para llevar a cabo todas las operaciones que desee realizar, y debe ser capaz de notificar al administrador acerca de cualquier anomalía.

1.3.1. Imposibilidad de lograr la seguridad absoluta

- Siempre habrá agujeros (fallas en la lógica de los programas) desconocidos para nosotros.
- Siempre habrá riesgos desconocidos para nosotros, inclusive para el programador de cada uno de los componentes del sistema.
- La seguridad es inversamente proporcional a la usabilidad.

1.4. ¿Qué es una herramienta de seguridad?

- Un programa que corre en espacio de usuario diseñado para ayudar al administrador —sea alertándolo o realizando por sí mismo las acciones necesarias— a mantener su sistema seguro.
- Orientadas a host: Trabajan exclusivamente con la información disponible dentro del host (configuración, bitácoras, etc.)
- Orientadas a red: Trabajan exclusivamente con la información proveniente de la red (barridos de puertos, conexiones no autorizadas, etc.)
- Muy importante: *Toda herramienta de seguridad útil para el administrador es también útil para un atacante, y toda herramienta de seguridad disponible para un administrador debemos asumir que está también disponible para un atacante.*

Capítulo 2

Configuración básica de seguridad en sistemas Unix genéricos

Una instalación segura inicia con una buena instalación. Los siguientes son consejos muy básicos acerca de cómo instalar un sistema conectado a una red potencialmente hostil.

2.1. Hacer la instalación sin tener conectividad a red

- De ser necesario, hacer la instalación sobre una red privada confiable
- Riesgos de no seguir este consejo

2.2. No brindar servicios no requeridos

- Mención honorífica: NetBSD, OpenBSD, Debian Linux
- Mención horrorífica: IRIX, Solaris, RedHat

2.3. Instalaciones mínimas

- Instalación mínima seguida de aumentar capacidades
- Nunca realizar una instalacion comprehensiva seguida de recortar capacidades

2.4. Los riesgos más comunes

- Portmap y RPC
- Los infames *r-commands*
- Manejo de protocolos inseguros
- Configuración y permisos incorrectos en los servicios públicos
- Uso de software no auditado
- Uso de versiones viejas

Capítulo 3

Características avanzadas de seguridad integradas a sistemas operativos libres

3.1. Manejo de reglas de filtrado de paquetes (Linux, *BSD)

- Hay paquetes para manejar filtrado de paquetes para prácticamente todos los Unixes.
- En Linux este paquete depende de la versión del sistema (ipfwadm [17], ipchains [18], iptables [19]).
- Para casi todos los demás Unixes está disponible IPFilter [20].
- FreeBSD puede manejar tanto ipfilter como ipfw [21].
- La próxima versión de OpenBSD utilizará un nuevo filtro llamado PF [22]; actualmente utiliza IPFilter.
- Un filtro de paquetes convierte a cualquier sistema Unix en un firewall.

3.2. Atributos extendidos del sistema de archivos (Linux, *BSD)

- Atributos tradicionales de los sistemas Unix: Permisos rwxrwxrwx (+SUID, +SGID, +Sticky)
- Atributos extendidos bajo Linux [23] (comando chattr): *A* (access time), *a* (append only), *c* (compresión), *d* (no-dump), *i* (inmutable), *s* (borrado destructivo), *S* (grabar cambios síncronamente), *u* (undeletable).

- Atributos extendidos bajo *BSD [24] (comando chflags): *arch, opaque, nodump, sappnd, schg, uappnd, uchg*

3.3. Atributos de montaje (Linux, *BSD)

- Siguiendo los estándares, es posible determinar qué tipos de archivo pueden/deben existir en determinados lugares del sistema de archivos.
- Los sistemas de archivos montables por usuario (p.ej., floppy, CD-ROM) deben sujetarse a diferentes reglas que aquellos montados por el administrador.
- Atributos interesantes de montaje permitidos con la opción -o del programa `mount` en ambos sistemas: `async` / `sync` (escritura síncrona / asíncrona), `noatime` (no actualiza atime a menos que cambie también mtime y ctime), `nodev` (no permitir nodos de dispositivo), `noexec` (no permitir archivos ejecutables), `nosuid` (no permitir archivos SUID)[25], [26]
- Atributos interesantes de montaje permitidos con la opción -o del programa `mount` en Linux: `nouser` (no permitir que un usuario monte la partición — este es el valor por default), `remount` (volver a montar cambiando alguna opción), `ro` / `rw` (montar de sólo lectura o de lectura/escritura), `encryption` + `keybits` (montar un sistema de archivos cifrado con una llave de `<keybits>` de longitud), `resgid` / `resuid` (permitir que los usuarios indicados con resuid o que forman parte del grupo indicado por resuid escriban al sistema de archivos si está a más del 95 % de su capacidad) [25]
- Atributos interesantes de montaje permitidos con la opción -o del programa `mount` en *BSD: `softdep` (montar con SoftUpdates, acelerando el funcionamiento general de la partición), `rdonly` (montar la partición de sólo lectura), `update` (actualizar los parámetros de montaje de una partición ya montada), `union` (hacer una unión del filesystem especificado y el que existe en el directorio donde se montó)[26]

3.4. Niveles de seguridad en la ejecución del Kernel (OpenBSD)

- El kernel de OpenBSD permite hacer más estricta la seguridad del sistema por medio de la bandera securelevel del `sysctl`. Hay cuatro niveles de seguridad, y el administrador puede subir el nivel en tiempo de ejecución, pero únicamente el proceso init puede bajarlo una vez colocado en modo seguro. El modo en el que el sistema va a correr se indica en el script /etc/rc.securelevel. [27]

- El nivel de seguridad -1 (*permanentemente inseguro*) indica que init no debe subir el nivel de seguridad, y sólo puede ser llamado desde sysctl cuando el sistema está en modo inseguro.
- El nivel de seguridad 0 (*inseguro*) es utilizado durante la inicialización del sistema, mientras éste está en modo monousuario. El administrador puede modificar los atributos extendidos de archivo, y todos los dispositivos pueden ser accesados acorde a los permisos naturales que tengan sus descriptores.
- El nivel de seguridad 1 (*seguro*) es el modo normal de ejecución del sistema. Una vez en este modo, sólo init puede bajarlo a inseguro. El kernel no permite la escritura a los archivos especiales /dev/mem y /dev/kmem, los dispositivos *crudos* de disco duro con particiones montadas son de sólo lectura (las particiones individuales sí son de lectura/escritura), los atributos extendidos de inmutable y append-only no pueden ser modificados aún por el administrador, y no se permite cargar/descargar módulos del kernel.
- El nivel de seguridad 2 (*altamente seguro*) agrega a las características del nivel 1: Todos los dispositivos *crudos* (tengan o no particiones montadas) son de sólo lectura, la hora del sistema no puede ser retrocedida, no se pueden modificar las tablas de filtrado de paquetes de red, y las banderas de depuración ddb.panic y ddb.console no pueden ser activadas.

3.5. Revisión de firmas MD5 de todos los paquetes instalados (Linux)

- La gran mayoría de las distribuciones de Linux (a excepción de Slackware) ofrecen un sistema maduro de manejo de paquetes
- Parte de la información que forma parte de un paquete es su firma MD5
- En sistemas basados en RedHat (que utilizan el formato de paquetes .RPM), un paquete individual puede ser verificado utilizando el comando `rpm -V` *`paquete`*. El sistema completo puede ser verificado con el comando `rpm -Va` [28]
- En sistemas basados en Debian es necesario instalar el paquete debsums. El comando para verificar un paquete individual es `debsums` *`paquete`*, y para hacer la verificación en todo el sistema es simplemente `debsums` [29]. Este programa permite también, a través de `debsums_gen`, firmar archivos que no pertenecen a ningún paquete para que puedan ser incluídos en la verificación [30].

3.6. Recompilación del 'mundo' (*BSD) [31]

- Si bien revisando MD5 podemos ver qué archivos han sido modificados, no podemos encontrar cuáles fueron los cambios. En los sistemas *BSD, el árbol completo de código fuente necesario para construir el sistema base puede ser bajado por CVS para ser comparado y compilado
- Desde el árbol de fuentes podemos hacer instalaciones parciales de ciertas ramas o subsistemas, con la sintaxis tradicional — `make clean && make dep && make && make install`
- Podemos también regenerar el sistema operativo completo a partir de las fuentes con `make world`
- Este sistema también nos permite ir siguiendo la rama `-current` (de desarrollo) del sistema operativo

3.7. Soporte a hardware acelerador de cifrado (OpenBSD)

- El cifrado criptográficamente fuerte se ha convertido cada vez más en una necesidad para los usuarios y administradores de redes de todo tipo. Sin embargo, éste es matemáticamente muy complejo, y puede consumir rápidamente los recursos del sistema.
- El sistema operativo OpenBSD fue el primero en poner a la criptografía como su mayor fuerza y tomarla como estandarte. [32]
- Hace relativamente poco tiempo, diferentes fabricantes de hardware han comenzado a distribuir tarjetas de aceleración criptográfica así como generadores de números aleatorios, indispensables para la criptografía fuerte. En la mayoría de los casos el soporte ha aparecido primero para OpenBSD que para cualquier otro sistema operativo [33].

3.8. Verificación diaria de sanidad: mtree (*BSD) [34]

- Los sistemas BSD mantienen una lista de directorios y archivos sensibles/importantes, y a diario revisa para reportar si ha habido algún cambio o hay alguna configuración incorrecta, reportándolo por correo al administrador. No sólo reporta dónde hubo cambios, sino que en qué consistieron estos.
- Las definiciones de los archivos a revisar con mtree están en el directorio /etc/mtree.

3.9. Sistemas de archivos con bitácora (Linux)

- Toda la información que está en nuestros discos duros puede ser localizada gracias a estar estructurada en uno o varios sistemas de archivos. Mantener la integridad de nuestros sistemas de archivos es, por tanto, un punto de gran importancia para la seguridad de nuestros sistemas.
- En un sistema de archivos tradicional se guarda información acerca de dónde está cada archivo, cuánto mide y diferentes atributos; en un sistema de archivos con bitácora se guarda, tal como en una base de datos transaccional, qué movimientos fueron realizados recientemente y qué movimientos faltan por realizar, para que si ocurre una falla eléctrica, un pánico de kernel o algo por el estilo sea no sea necesario buscar en la partición entera buscando inconsistencias.
- Linux ofrece actualmente cuatro diferentes sistemas de archivo con bitácora: ReiserFS [35] (muy veloz y eficiente en el manejo de espacio gracias a que organiza su información en un árbol binario balanceado; es actualmente el sistema de archivos predeterminado en SuSE y Mandrake, y puede realizarse una instalación Debian sobre ReiserFS con discos de arranque terceros), Ext3fs [36] (Se enfoca en mantener compatibilidad con Ext2, el estándar actual en Linux. Es el sistema de archivos predeterminado en RedHat 7.2), XFS [37] (Sistema de archivos muy rápido y altamente escalable, donado a Linux por Silicon Graphigs) y JFS [38] (Sistema de archivos altamente escalable, donado a Linux por IBM).

3.10. Sistemas de archivos con SoftUpdates (*BSD)

- Un enfoque alterno para mantener la consistencia en los sistemas de archivos es el que toman los sistemas derivados de BSD: Los SoftUpdates [39]. La idea consiste en que el sistema operativo se obligue a sí mismo a hacer las cosas en órden — A planificar todas las escrituras de modo que no haya posibilidad de que lo que haya en el disco en algún momento presente inconsistencias. En vez de planificar las escrituras a disco pensando en un disco físico (sector por sector), lo hace pensando en la estructura lógica del disco, guardando los metadatos en una sóla operación.
- Las mismas ventajas de un sistema de archivos con bitácora las encontramos en un sistema con SoftUpdates; hay guerras religiosas en torno a cuál de estos sistemas es mejor... Y la respuesta no podrá darla más que cada usuario. Margo I. Seltzer et. al. presentaron [40] un artículo muy interesante explicando las diferencias entre ellos a fondo en el congreso Usenix del 2000.

Capítulo 4

Herramientas básicas de seguridad

4.1. Herramientas orientadas a host

- Verificando la integridad del sistema: TripWire [41] (Reportes periódicos de archivos modificados)
- Monitoreo de bitácoras: Logcheck [42], [43] (Reportes por correo al administrador del sistema de las líneas importantes que aparezcan en las bitácoras), Swatch [44] (Monitoreo constante y reacción instantánea a determinados patrones en las bitácoras)
- Errores comunes de configuración: COPS [45] (Verificación de diferentes archivos de configuración. Muy viejo.)
- Manejo de archivos de contraseñas: npasswd [46] (reemplazo de passwd para evitar que el usuario proporcione contraseñas débiles), passwd+ [47] (similar a npasswd, aunque bastante más extensible), crack [48] (ayuda a encontrar contraseñas débiles), shadow [49] (esconde las contraseñas cifradas en un archivo de lectura sólo para el administrador)
- Reportes condensados de bitácoras: Analog [50], Webalizer [51] (ambos presentan repotes fáciles de comprender y de seguir de las bitácoras del sistema, orientados principalmente a bitácoras de Web, pero capaces de entender algunos otros formatos. Webalizer produce resultados más bonitos y permite mantener archivos históricos mucho más fácilmente, Analog es mucho más flexible y funciona en más ambientes).

4.2. Herramientas orientadas a red

- Encontrando barridos de puertos: Portsentry [52], [53] (escucha a los puertos en los que no hay servicios legítimos en ejecución y no deberían recibir conexiones, puede tomar respuesta inmediata automática)
- Sistema Detector de Intrusos en Red: Snort [54] (detector de intrusos con una amplia base de datos y capacidades de conectarse con muy diferentes paquetes para análisis), ACID [55] (analizador y reportador de registros de IDS en base de datos)
- Realizar búsquedas de vulnerabilidades: nmap [56] (realiza barridos de puertos buscando sistemas con vulnerabilidades), SATAN [57] (busca vulnerabilidades de configuración ya conocidas desde una interfaz sencilla de usar. Muy viejo.), SAINT [58] (Similar a SATAN, más actualizado), nessus [59] (Similar a SAINT y SATAN, más actualizado aún, con una base de ataques muy amplia, y con un control mucho más fino acerca de qué hacer)
- Monitoreo centralizado de sistemas: MRTG (Recoge periódicamente el estado actual de diferentes sistemas y lo presenta en gráficas con interfaz Web, con registro histórico de un año. Si bien está hecho para monitorear ruteadores por SNMP, puede utilizarse para monitorear todo tipo de servidores utilizando conexiones simples TCP/IP, lo que representa un riesgo mucho menor a la seguridad que SNMP)

4.3. Herramientas específicas a un sistema operativo

Hay herramientas de seguridad que, si bien no pertenecen directamente al núcleo de un sistema operativo, trabajan estrechamente ligados con él. Merecen, por sus características, un apartado propio.

4.3.1. Bastille Linux [61]

- Bastille Linux es una serie de scripts que revisan una instalación de Linux (soporta las hechas por distribuciones RedHat y Mandrake) y "aprieta" la seguridad en muchos aspectos.
- La instalación de Bastille Linux es sencilla. Está estructurada por áreas de acción. Explica al administrador qué es lo que está haciendo y por qué es recomendable hacerlo, y le permite no aplicar cualquiera de sus configuraciones si él así lo indica.
- En un futuro el equipo de desarrollo de Bastille Linux planea soportar otras distribuciones, y probablemente otros tipos de Unix.

4.3.2. Parches de seguridad al kernel (Linux)

- El diseño modular del kernel de Linux ha permitido la aparición de numerosos parches que aumentan la seguridad del sistema.
- Los parches provistos por el equipo OpenWall [62], entre otras cosas, crean un área de stack no ejecutable para los usuarios (previniendo buffer overflows), imponen controles más estrictos sobre los archivos localizados en /tmp y /proc, modifican el comportamiento de los descriptores de archivo privilegiados 0, 1 y 2, y otras varias mejorías del punto de vista de seguridad.
- Immunix [63] es un sistema derivado de RedHat, construído utilizando el compilador Stackguard en lugar del mucho más común gcc. Este compilador ordena de diferente manera los archivos objeto, de modo que es más difícil provocar un buffer overflow, al menos uno que provoque acceso privilegiado al sistema. Incluye también FormatGuard, que previene las vulnerabiildades de formato en cadena, y SubDomain, que aísla a un programa, impidiéndole accesar áreas del sistema para las que no tiene permiso explícito.
- Linux Security Module Interface [64] provee una interfaz para desarrollar módulos de seguridad e integrarlos al kernel de una manera más limpia que si cada desarrollador implementara su propio método para integrar características de seguridad al sistema. Si piensas desarrollar un módulo de seguridad, va una fuerte recomendación a aprovechar este trabajo en vaz de iniciar desde cero.
- SELinux (Security Enhanced Linux) [65] es un proyecto financiado por la NSA (Agencia de Seguridad Nacional de Estados Unidos) para llevar a Linux niveles de seguridad contemplados por su famoso Libro Naranja. Para esto, agregan controles de acceso mandatorios (MAC) al sistema, logrando de esta manera que el que haya un compromiso de root no signifique que la totalidad de información del sistema está a disposición del atacante.

4.4. Desarrollando sus propias herramientas

- Una herramienta no requiere ser compleja. Las mejores herramientas son las más simples. (Filosofía de componentes Unix)
- Lenguajes principales para desarrollar herramientas: Perl, Python, los diferentes shells
- Por qué *no* desarrollar herramientas de seguridad en C/C++ (y qué características de C/C++ lo pueden hacer indispensable en ciertos casos)
- Algunos ejemplos de herramientas sencilas en el sitio de Gunnar [66]

Capítulo 5

¿Qué sigue?

5.1. Nuevos agujeros - Siguiéndoles la pista

- La importancia de mantener la guardia siempre arriba
- Recursos para enterarse de nuevas vulnerabilidades: Bugtraq [67], las listas de correo de los fabricantes de sus sistemas operativos/distribuciones favoritos
- Al aparecer una advertencia por parte del fabricante, parchar el sistema *de inmediato.*

5.2. La nueva moda – Gusanos

- Los gusanos no son nuevos; el primero reportado tiene más de 12 años y logró que más de la mitad de los hosts de Internet dejaran de operar
- Es relativamente fácil escribir código autoreplicable que explote una vulnerabilidad en red
- Gusanos famosos del último año: Lion (sistemas RedHat 7.0), Ramen (sistemas RedHat 6.2 y 7.0), Adore (sistemas RedHat 7.0), Code Red (sistemas Windows NT 4.0) y variantes
- Características de los gusanos: Cientos o miles de intentos de acceso por parte de sistemas de todo el mundo con exactamente el mismo patrón de comportamiento. Después de un ataque exitoso, alto consumo de recursos para reproducirse.
- Al tener patrones tan predecibles, es fácil bloquear un gusano con filtros de paquetes

5.3. Estar siempre al día en seguridad

5.3.1. Sitios importantes para visitar frecuentemente

Sitios generales de seguridad

- Departamento de Seguridad en Cómputo (DGSCA-UNAM), http://www.seguridad.unam.mx
- SecurityFocus, http://www.securityfocus.org
- OpenBSD México, http://www.openbsd.org.mx

Centros de atención a incidentes

- UNAM-CERT, http://www.cert.unam.mx
- CERT, http://www.cert.org
- Computer Incident Advisory Capability (CIAC), http://ciac.llnl.gov
- Forum of Incident Response and Secutity Teams (FIRST), http://www.first.org
- COAST, http://www.cs.purdue.edu/coast/

Sitios *underground*

- Phrack Magazine, http://www.phrack.com
- 2600 Magazine, http://www.2600.com

5.3.2. Congresos y seminarios importantes en seguridad

Seguridad en Cómputo http://www.seguridad.unam.mx

Organizado por el Departamento de Seguridad en Cómputo de la UNAM, del 24 al 30 de noviembre. El congreso de seguridad más importante en el mundo de habla hispana.

SANS http://www.sans.org

El Instituto SANS da cursos especializados a responsables de seguridad. Sus encuentros se llevan a cabo varias veces por año y en diferentes continentes — en el 2001 hubo cuatro encuentros, dos de ellos en Estados Unidos, uno en Inglaterra y uno en Australia.

Tienen disponible material de sus cursos en http://www.sans.org/giactc.htm

USENIX http://www.usenix.org

USENIX es una organización techo que organiza congresos de todo tipo relacionados con la administración de sistemas en todo el mundo. De sus congresos, el más importante relacionado con seguridad es LISA.

Bibliografía

[1] The GNU Manifesto, http://www.gnu.org/gnu/manifesto.html

[2] Free Software Foundation, http://www.fsf.org

[3] GNU General Public Licence (GPL), http://www.gnu.org/copyleft/gpl.html

[4] BSD, Berkeley System Distribution, de la sección 'History' de libro UNIX Basic Introduction, Claude Cantin, National Research Council Canada, http://www.sao.nrc.ca/imsb/rcsg/documents/basic/node6.html

[5] OpenBSD Copyright Policy, OpenBSD Project, http://www.openbsd.org/policy.html

[6] Open Source Initiative, http://www.opensource.org

[7] Licencias aprobadas por la Open Source Initiative, http://www.opensource.org/licenses/index.html

[8] Comentarios acerca de varias licencias, Free Software Foundation, http://www.gnu.org/philosophy/license-list.html

[9] Sistema operativo Linux, http://www.linux.org

[10] Sistema operativo FreeBSD, http://www.freebsd.org

[11] Sistema operativo OpenBSD, http://www.openbsd.org

[12] Sistema operativo NetBSD, http://www.netbsd.org

[13] Sistema operativo AtheOS, http://www.atheos.cx

[14] Sistema operativo HURD, http://www.gnu.org/software/hurd/hurd.html

[15] Sistema operativo FreeDOS, http://www.freedos.org/

[16] Sistema operativo MIT ExoKernel, http://www.pdos.lcs.mit.edu/exo/

[17] Preguntas frecuentes de ipfwadm, filtro de paquetes para sistemas Linux con kernel 2.0.x, http://www.fwtk.org/ipfwadm/faq/ipfwadm-faq.html

[18] Guía rápida de ipchains, filtro de paquetes para sistemas Linux con kernel 2.2.x, http://www.linuxdoc.org/HOWTO/IPCHAINS-HOWTO.html

[19] Guía rápida de iptables, filtro de paquetes para sistemas Linux con kernel 2.4.x, http://www.linuxguruz.org/iptables/howto/iptables-HOWTO.html

[20] IPFilter — Filtro de paquetes para diferentes Unixes, Darren Reed, http://www.ipfilter.org

[21] Sección 6.4 (firewalls) del FreeBSD Handbook, http://www.infran.ru/TechInfo/BSD/handbook83.html

[22] PF — Filtro de paquetes de OpenBSD, http://www.benzedrine.cx/pf.html

[23] Atributos extendidos en sistemas de archivos ext2, Michael Shaffer, http://www.securityfocus.com/frames/?focus=linux&content=/focus/linux/articles/ext2attr.html

[24] Atributos extendidos en *BSD, página de manual de chflags en OpenBSD, http://www.openbsd.org/cgi-bin/man.cgi?query=chflags

[25] Opciones de montaje en Linux, página de manual de mount en http://linux.ctyme.com/man/man1254.htm

[26] Opciones de montaje en *BSD, página de manual de mount en OpenBSD, http://www.openbsd.org/cgi-bin/man.cgi?query=mount

[27] Página de manual de securelevel en OpenBSD, http://www.openbsd.org/cgi-bin/man.cgi?query=securelevel

[28] HOWTO de RPM, http://www.rpm.org/support/RPM-HOWTO-4.html

[29] Paquete debsums de Debian, http://packages.debian.org/stable/admin/debsums.html

[30] Debsums_gen, página de manual, http://www.fifi.org/cgi-bin/man2html/usr/share/man/man8/debsums_gen.8.gz

[31] Using make world, FreeBSD handbook, http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/makeworld.html

[32] La fuerte relación de OpenBSD con la criptografía, http://www.openbsd.org/crypto.html

[33] Soporte a hardware criptográfico en OpenBSD, http://www.openbsd.org/crypto.html#hardware

[34] Página de manual de mtree de OpenBSD, http://www.openbsd.org/cgi-bin/man.cgi?query=mtree

[35] ReiserFS, http://www.namesys.com/

[36] Ext3fs, http://people.spoiled.org/jha/ext3-faq.html

[37] XFS, http://people.spoiled.org/jha/ext3-faq.html

[38] JFS, http://oss.software.ibm.com/developerworks/opensource/jfs/

[39] Soft Updates, http://www.mckusick.com/softdep/

[40] Journaling Versus Soft Updates: Asynchronous Meta-data Protection in File Systems, et. al., USENIX Annual 2000 Technical Conference, http://www.usenix.org/publications/library/proceedings/usenix2000/general/seltzer.html

[41] Tripwire, http://www.tripwire.org/

[42] Logcheck, http://www.psionic.com/abacus/logcheck

[43] Tutorial de Logcheck de Gunnar Wolf, http://www.gwolf.cx/seguridad/logcheck/

[44] Swatch, http://oit.ucsb.edu/~eta/swatch/

[45] COPS, http://www.fish.com/cops/

[46] npasswd, http://www.utexas.edu/cc/unix/software/npasswd/

[47] passwd+, http://www.funet.fi/pub/unix/security/passwd/passwd+/

[48] crack, http://www.users.dircon.co.uk/~crypto/

[49] Linux Shadow Password HOWTO, http://www.linuxdoc.org/HOWTO/Shadow-Password-HOWTO.html

[50] Analog, http://www.statslab.cam.ac.uk/~sret1/analog

[51] Webalizer, http://www.mrunix.net/webalizer

[52] Portsentry, http://www.psionic.com/abacus/logcheck

[53] Tutorial de Portsentry de Gunnar Wolf, http://www.gwolf.cx/seguridad/portsentry/

[54] Snort, http://www.snort.org

[55] ACID, http://acidlab.sourceforge.net/

[56] nmap, http://www.insecure.org/nmap/index.html

[57] SATAN, http://www.fish.com/satan/

[58] SAINT, http://www.wwdsi.com/saint/

[59] nessus, http://www.nessus.org/

[60] MRTG, http://ee-staff.ethz.ch/~oetiker/webtools/mrtg

[61] Bastille Linux, http://www.bastille-linux.org/

[62] Parches de OpenWall para el kernel de Linux, http://www.openwall.com/linux y http://www.openwall.com/linux/README

[63] Immunix / Stackguard, http://www.immunix.org

[64] Linux Security Module Interface, http://lsm.antisoft.com/

[65] Security Enhanced Linux, http://www.nsa.gov/selinux/

[66] Tutorial *Scripts de Seguridad en Perl* expuesto en el Congreso de Seguridad en Cómputo 2000, organizado por la UNAM. Gunnar Wolf, http://www.gwolf.cx/seguridad/script_seg_perl/

[67] Lista de correo Bugtraq, bugtraq@securityfocus.com, http://www.securityfocus.org/templates/archive.pike?list=1