

Por Kurt Seifried (seifried@seifried.org) (Llave de cifrado), la GSAL tiene licencia OpenContent.

Traducción al español de José Antonio Revilla (revilla@segurinet.com) (Llave de cifrado)

La versión original de la GSAL está disponible en <http://www.securityportal.com/lasg>

Última versión en español disponible en <http://segurinet.com/gsal>

GSAL se encuentra en constante desarrollo.

Envíe un correo en blanco a gsal@segurinet.com si desea ser informado cada vez que se actualice.

Últimos cambios de la versión en español: 28-Nov-99

Burocracia

- * Licencia y garantía
- * Prefacio
- * Acerca del Autor
- * Acerca del Traductor
- * Comentarios a la traducción
- * Contribuciones
- * Colaboradores
- * Qué es y qué no es esta guía
- * Modificaciones

Empezando

- * Cómo determinar qué asegurar y cómo asegurarlo
- * Instalación segura de Linux
- * Conceptos generales, servidor versus estaciones de trabajo, etc.

El núcleo del sistema

- * Ficheros del sistema
- * Seguridad de Ficheros / Sistema de Ficheros
- * PAM

Seguridad física / consola

- * Seguridad Física / de Arranque

Contraseñas

- * Seguridad de contraseñas
- * Almacenamiento de contraseñas

Seguridad básica de red

- * Seguridad básica de servicios de red
- * Ficheros básicos de configuración de red

Seguridad TCP-IP

- * TCP-IP y seguridad de red
- * Seguridad PPP
- * Seguridad IP (IPSec)
- * Cifrado de datos / servicios
- * Rutado

Cortafuegos / Proxies

- * Software de Proxy
- * Cortafuegos

Servicios de Red

- * Telnet
- * SSH
- * FTP
- * HTTP / HTTPS
- * SMTP
- * POP
- * IMAPD
- * DNS
- * NNTP
- * DHCP
- * rsh, rexec, rcp
- * NFS
- * TFTP
- * BOOTP
- * SNMP
- * FINGER
- * IDENTD
- * NTP
- * CVS
- * rsync
- * lpd
- * SMB (SAMBA)
- * LDAP
- * Sistema X Window
- * Conectividad SNA
- * Software de Autoridad de Certificación para Linux

Seguridad del Kernel

- * El Kernel de Linux
- * Parches de seguridad del Kernel y del Compilador

Administración del Sistema

- * Herramientas administrativas
- * Gestión de software

Registro de logs y monitorización

- * Herramientas de monitorización de Hosts
- * Ficheros de log y otros métodos de monitorización
- * Limitación y monitorización de usuarios

Seguridad de Red

- * Lista de comprobación para la conexión a Internet
- * Métodos de compartición de ficheros

- * Lectores de correo basados en WWW
- * Autenticación basada en red
- * Software de listas de correo

Intrusiones y detección de intrusos

- * Herramientas de escaneo / intrusos
- * Herramientas de detección de escaneos / intrusos
- * Sniffers de paquetes
- * Normas de comportamiento / integridad del sistema

Copias de seguridad y auditorías

- * Gestión de auditorías
- * Copias de seguridad

Acciones hostiles

- * Enfrentándose a los ataques
- * Ataques de negación de servicio
- * Ejemplos de ataques
- * Virus, Cabayos de Troya y Gusanos

Distribuciones

- * Distribuciones seguras de Linux
- * Información específica por Distribución
- * Información de contacto con el Vendedor

Seguridad de aplicaciones

- * Programación segura

Apéndices

- * Apéndice A: Libros y revistas
- * Apéndice C: Otras documentaciones de seguridad en Linux
- * Apéndice D: Documentación de seguridad en línea
- * Apéndice E: Sitios generales de seguridad
- * Apéndice F: Sitios generales de Linux

Secciones obsoletas / interrumpidas

- * Historial de versiones (interrumpida)
- * Páginas LASG antiguas con enlaces a la versión PDF

Licencia y Garantía

Se define OpenContent (OC) como el conjunto de páginas web que comprenden la "Guía de Seguridad del Administrador de Linux", o el documento PDF titulado "Guía de Seguridad del Administrador de Linux"

LICENCIA

Términos y condiciones para la Copia, Distribución y Modificación

Otros elementos más allá de la copia, distribución y modificación del Contenido con el que se distribuye esta licencia (como el uso, etc.) se encuentran fuera del ámbito de esta licencia.

1. Se puede copiar y distribuir réplicas exactas del OpenContent (OC) cuando se reciban, en cualquier medio, con tal de que se publique de forma visible y apropiada en cada copia el correspondiente aviso de copyright y renuncia de garantía; se mantengan intactos todos los avisos que hacen referencia a esta Licencia y a la ausencia de cualquier garantía; y se le proporcione a cualquier otro receptor del OC una copia de esta Licencia junto con el OC. Se puede cobrar, a su elección, unas tasas por los medios y/o la manipulación que conlleve el crear una única copia del OC para uso offline (desconectado), puede a su elección ofrecer soporte instructivo sobre el OC a cambio de una cuota, o a su elección, puede ofrecer garantías a cambio de unas tasas. No se puede cobrar una tasa por el OC en sí mismo. No se puede cobrar una tasa por el simple servicio de proporcionar acceso y/o uso al OC vía red (por ejemplo, Internet) ya sea mediante world wide web, FTP, o cualquier otro método.

2. Puede modificar su copia o copias del OpenContent o cualquier porción del mismo, de tal modo que se creen trabajos basados en el Contenido, y distribuir tales modificaciones o trabajo bajo los términos de la Sección 1 más arriba detallada, con tal de que se cumplan las siguientes condiciones:

a) En el material modificado, debe incluirse anuncios visibles de que ha sido modificado, así como la naturaleza exacta y contenido de los cambios, y la fecha de cualquier cambio.

b) Es necesario que cualquier trabajo que Vd. distribuya o publique, que contenga en todo o en parte contenidos, o esté derivado del OC de cualquier forma, tenga licencia como un todo sin cargo alguno a terceros bajo los términos de esta licencia, a menos que esté permitido de otra forma bajo la ley de Uso Adecuado aplicable.

Estos requerimientos se aplican al trabajo modificado como a un todo. Si se identifican secciones de ese trabajo que no estén derivadas del OC, y se puedan considerar razonablemente como independientes y trabajos aislados por sí mismos, entonces no se aplican esta Licencia ni sus términos, cuando se distribuyan como trabajos separados. Pero cuando se distribuyan las mismas secciones como parte de un todo el cual es un trabajo basado en el OC, la distribución del todo debe ser acorde con los términos de esta licencia, cuyo permiso para otras licencias se extiende a la totalidad, y de esta forma a todas y cada una de las partes sin importar quién lo escribió. Se hacen excepciones a estos requerimientos para emitir trabajos modificados libres de cargos bajo esta licencia, sólo en cumplimiento de la ley de Uso Adecuado allí donde sea aplicable.

No se le requiere aceptar esta Licencia, puesto que no la ha firmado. Sin embargo, nada más le garantiza permiso para copiar, distribuir o modificar el OC. Si no se acepta la Licencia, tales acciones están prohibidas por la ley. De

tal forma que, al distribuir o modificar el OC, o trabajos derivados del mismo, se indica la aceptación de esta Licencia para hacerlo, y todos sus términos y condiciones para la copia, distribución o traducción del OC.

SIN GARANTÍA

4. PUESTO QUE EL OPENCONTET (OC) TIENE LICENCIA LIBRE DE CARGOS, NO EXISTE GARANTÍA EN CUANTO AL OC, ALLÍ HASTA DONDE ESTÉ PERMITIDO POR LA LEY APLICABLE. EXCEPTO CUANDO ASÍ CONSTE DE OTRA FORMA POR PARTE DE LOS PROPIETARIOS DEL COPYRIGHT Y/O CUALESQUIERE OTRAS PARTES, EL OC SE PROPORCIONA "TAL CUAL" SIN GARANTÍA DE NINGUNA CLASE, YA SEA EXPRESA O IMPLÍCITA, INCLUYENDO, PERO NO LIMITÁNDOSE, A LAS GARANTÍAS IMPLÍCITAS DE MERCADERÍA Y CONVENIENCIA PARA UN PROPÓSITO PARTICULAR. AL UTILIZAR EL OC TODO EL RIESGO LO ASUME VD. SI SE PROBASE QUE EL OC FUESE ERRÓNEO, DEFECTUOSO O INACEPTABLE DE CUALQUIER OTRA FORMA, ES VD. QUIEN ASUME EL COSTE DE LAS REPARACIONES O CORRECCIONES NECESARIAS.

5. EN NINGÚN CASO, A MENOS QUE ASÍ LO REQUIERA LA LEY APLICABLE O SE ACUERDE POR ESCRITO CON CUALQUIER PROPIETARIO DEL COPYRIGHT, O CUALQUIER OTRA PARTE QUE PUEDA HACER UN MIRROR Y/O REDISTRIBUIR EL OC SEGÚN ESTÁ PERMITIDO MÁS ARRIBA, TENDRÁ RESPONSABILIDADES ANTE VD. EN CUANTO A DAÑOS, INCLUIDOS CUALQUIER DAÑO GENERAL, ESPECIAL, ACCIDENTAL O COMO CONSECUENCIA DEL USO O INCAPACIDAD DE USO DEL OC, INCLUSO SI TAL PROPIETARIO O CUALESQUIERA OTRA PARTE HA SIDO ADVERTIDO DE LA POSIBILIDAD DE TALES DAÑOS.

bfqRI20jRcGRQt5wIwIDAQABo4HTMIHQMAkGA1UdEwQCMAAwga8GA1UdIASBpzCA
MIAGC2CGSAGG+EUBBwEBMIAwKAYIKwYBBQUHAgEWHGh0dHBzOi8vd3d3LnZlcmllz
aWduLmNvbS9DUFMwYgYIKwYBBQUHAgIwVjAVFg5WZXXJpU2lnbiwgSW5jLjADAgEB
Gj1WZXXJpU2lnbidzIENQUyBpbmNvcnAuIGJ5IHJlZmVyZW5jZSBsaWFiLiBsdGQu
IChjKtK3IFZlcmllTaWduAAAAAAMBEQCWCGSAGG+EIBAQQEAWIHgDANBgkqhkiG
9w0BAQQFAAOBgQAwfnV6AKAetmcIs8lTkgp8/KGbJCbL94adYgfhGJ99M080yhCk
yNuZJ/o6L1VlQCxjntcws+VMtMziJNELDCR+FzAKxDmHgal4XCinZMhp8YdqWsfC
wdXnRMPqEDW6+6yDQ/pi84oIbPlujDdajN141YLuMz/c7JKsuYCKkk1TZQ==
-----END CERTIFICATE-----

Firmo todo mi correo con ese certificado y la llave PGP, de modo que si no está firmado, no es mío. Tienes libertad para cifrar tus correos dirigidos a mi con mi certificado, estoy intentando promover el correo seguro a nivel mundial (aunque no parece que esté funcionando).

Se puede contactar con el traductor en la siguiente dirección:

José Antonio Revilla - revilla@segurinet.com

También firmo todo mi correo con PGP, si no va firmado no es mío. Prefiero recibir correo cifrado. Si por correo tradicional nos escribimos cartas en vez de postales, escribamos también correos electrónicos cifrados.

Escribe GSAL en el Tema, para que lo detecte el filtro. No des por hecho recibir respuesta, aunque al menos trato de leer todos los correos.

Mi llave pública PGP:

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGPfreeware 6.0.2i

mQGIBDQoFagRBADDXri20uPsfHoG1WQ+mt83CtuteB4TnaI9ZqgBwVVPzDUY0TUT
L6OTgIH5CD6uAlFK+FKPQeGO42FbYnvikFk5drrkVG5ubsUWoEpLCMA40wwG7i/5
4jd4LgXmKrq78jlpTxBYMMHumaXP1ZCJNKC+wSHJPymOUrvzcg+lWmphRwCg/73D
S/7B+abccwh4hZEGRNQ+uueEAMIn/8XlYkre40xCptpEORA8YBg2XZ3wONomhhN3
UgkdNA4BLcxWiAu64vG89FdxvIqczX5t71B8QPPHp96n1ORIwFBTtmpj/HqDaZP8B
4XDwaXroqzEABm8gYW3WjB/CEQcwaDRY6X8zGyCUmZPuE+8+KAbq6IapNQbakXqs
fZ74A/0ZkRbvDLvdrKyaR44lFT2KhkPt304Q/XY8eAlBswzFWDESPsxfEvX8cmNI
VkeMf42F8xoiuaKhqHVIHLqnIcnoCObEMdKBX8R0k6m/2ssaEFKuy2e75Q6HtFJP
vNeQvxPw0mVXnzLQBR274bKQhk94uQZUD3y5U+n81eA2uezhDbQjSm9z6SBBbnRv
bmlvIFJldmlsbGEgPHJldmlsbGFAAs5hbT6JAESeeBECAAsFAjQoFagECwMCAQAK
CRDguZgz02L8EUIEAKD/FeOPeRaxGbmGO6hRptYfI6CJEQCg052IP6L7DPeSIjVO
qr5ou94RqhO5Ag0ENCgVqBAIAPZCV7cIfwgXcqK61q1C8wXo+VMROU+28W65Szgg
2gGnVqMU6Y9AvfPQB8bLQ6mUrfdMZIZJ+AyDvWXPf9Sh01D49Vlf3HZSTz09jdvO
meFXklnN/biude/F/ha8g8VHMGHOfMlm/xX5u/2RXscBqtNbn02gpXI61Brwv0YA
WCvl9Ij9WE5J280gtJ3kkQc2azNsOAlFHQ98iLMcfFstjvbzySPAQ/ClWxiNjrtV
jLhdONM0/XwXV00jHRhs3jMhLLUq/zzhSslAGBGNfISnCNLWhsQDGCgHKXrKlQzZ
lp+r0ApQmwJG0wg9ZqRdQZ+cfL2JSyIZJrqr0l7DVekyCzsAAGIIAN3k+XPKMzqE
ovOwc8nyybVvJv4C6Ktd/XMR4Z6GZD2EX2NFIf+LVEDGIBVdd89Z3gIdqIIRJ5yU
SSFNMIAoQhAtmhK9/MOx1x87qrBsJmil0hMKPBA/mf4KKxaUZJyFkiHnSgJafyJF
ZJeLvxnbYlgXLazU7BnL5kay3cvQd9ofFSIa6AMrnFuOihSJKQDlKbTSXOYBBbvN
ckFcNgzvJlmQUQXd2ElgTzdnL6ng0pm8BI/1FqUMWUKOthP9a4TFjSz8C0FsJXn
LryqrQfVv5H+JZlWMBpLnE93bkesY3qNGMrI6t4c5w8XAiujiQ4v8veBSxlcyB/h
VlJskJpTm2OJAeyEGBECAAYFAjQoFakACgkQ4LmYM9Ni/BHPUACfZLuWyjAEupRE
dXA3NmFYxZft+OIAN34ox331zuFJEbI7Pb7XsOyRqFoa
=Z6kI
-----END PGP PUBLIC KEY BLOCK-----

Acerca del autor

Tuve mi segundo (el primero no cuenta, un TRS-80 que murió en pocos meses) ordenador en Navidades de 1993, 4 meses más tarde me quité de encima el windows pasándome a OS/2, conseguí un segundo ordenador en primavera de 1994, en el que cargué Linux (Slackware 1.?) en Julio de 1994. Estuve con Slackware durante 2-3 años, y me cambié a Red Hat después de que me lo presentasen, me cambié después de estar expuesto a Red Hat durante 2 o 3 meses. Desde entonces también he alcanzado un MCSE y un MCP+Internet (ven al lado oscuro Luke...). ¿Que por qué he escrito esta guía? Pues porque nadie más lo ha hecho. ¿Por qué está disponible gratuitamente en línea? Porque quiero que alcance la mayor audiencia posible.

También he recibido ayuda en esta guía (directa e indirectamente) de la inmensa comunidad de Internet, mucha gente que ha puesto en marcha páginas de seguridad excelentes, las cuales listo, así como de listas de correo como Bugtraq, que me ayudan a estar en la cima de lo que está ocurriendo. Suena a tópico (y espero que Dios les prohíba a los periodistas que escojan esta frase) pero esto no hubiera sido posible sin la ayuda de la comunidad del código abierto. Os doy las gracias a todos.

[Guía de Seguridad del Administrador de Linux - GSAL]

Acerca del traductor

Me decidí a traducir esta guía como contribución personal a la comunidad Linux en general y a la hispanohablante en particular, como pequeña muestra de agradecimiento.

Mi primer acercamiento a la telemática fue a 300 bps., el mundo de la BBS y Fidonet, siendo punto de ACCE BBS (Asociación Cántabra de Correo Electrónico), de donde guardo un grato recuerdo de su gente. Linux lo conocí en el 94, con la distribución Slackware, conectando por las mismas fechas con la Red, cuando Internet era de color negro, y pronto comencé a interesarme por temas de (in)seguridad. En la actualidad compagino mi interés por Linux con el desarrollo de mi actividad profesional como administrador de sistemas y analista de seguridad.

Correo electrónico, preferentemente cifrado, a: revilla@segurinet.com - Mi llave pública PGP

Última versión de esta guía: <http://segurinet.com/gsal>

Escribe GSAL en el Tema, para que lo detecte el filtro. No des por hecho recibir respuesta, aunque al menos trato de leer todos los correos.

No preguntes qué puede hacer Linux por ti, pregúntate qué puedes hacer tú por Linux.

Comentarios a la traducción

Última versión de la guía, cambios, correcciones, actualizaciones, etc.:
<http://segurinet.com>

Se ha abordado la traducción desde un punto de vista pragmático, no purista, libre cuando así lo requiriese, no literal, en ocasiones conservando el fondo en sacrificio de la forma. La vigésima primera edición del diccionario de la lengua española define el término "traducir" como "expresar en una lengua lo que está escrito o se ha expresado antes en otra", "explicar, interpretar" en su tercera acepción. En la medida de lo posible, se ha tratado de evitar traducciones literales, visualmente incómodas y fútiles para la adecuada comprensión del texto.

Así, se dejan en su forma original términos como `host`, `exploit`, `sniffer`, `spoofing`, `script` o `fichero shadow`, `backbone` en lugar de anfitrión, explotar, esnifador, guión, fichero sombra, espina dorsal respectivamente, entre otros tantos, si bien se traducen otros más aceptados, como `password` por contraseña, `gateway` por puerta de enlace o pasarela, `backup` por copia de seguridad o `to crypt`, `to encrypt` por cifrar (en español no existe "encriptar", está más cercano de la tanatopraxia que de la criptografía).

Ni mucho menos quiere esto decir que la traducción sea impoluta. Es más, quedan por revisar, a medida que se vayan detectando, muchos giros y expresiones, que se irán corrigiendo en lo sucesivo. El traductor también se ha tomado ciertas licencias, en la utilización de términos como "mapear", "escanear" (el diccionario admite escáner), o "resetear", o la traducción de `command` por comando, cuando sería más apropiado orden o mandato, términos que aunque incorrectos, facilitan la comprensión, si bien es de reconocer que empobrecen el lenguaje.

Se han traducido los scripts allí donde fuese posible, en los comentarios e incluso ciertas variables, por facilitar su análisis.

Se invita al lector interesado en profundizar en estos y otros aspectos relacionados a visitar las siguientes direcciones:

<http://www.rae.es/NIVEL1/buscon/AUTORIDAD2.HTM> - Diccionario académico

http://www.ati.es/novatica/glosario/glosario_internet.txt

<http://www.uco.es/webuco/si/ccg/glosario/glosario.html>

<http://goya.eunet.es/listserv/spanglish/>

<http://pagina.de/interdic>

<http://www.el-castellano.com>

[Guía de Seguridad del Administrador de Linux - GSAL]

Contribuciones

Serán bien recibidas las contribuciones de URL's y punteros a recursos y programas que no aparecen listados (echar un vistazo a la lista de URLs del final para asegurarse que no haya sido ya incluida). Si deseas contribuir a la GSAL con material escrito, ediciones o cualquier otro tipo de trabajo, por favor escíbeme a seifried@seifried.org. En general, si es bueno y no he escrito todavía sobre ello, acabará formando parte de la GSAL, y se te incluirá en la lista de contribuyentes.

Contribuciones a la versión en español

Para informar acerca de recursos relativos a seguridad para Linux en español, escribir a revilla@segurinet.com

[Guía de Seguridad del Administrador de Linux - GSAL]

Colaboradores

Alan Mead - amead@soltec.net - edición masiva (muchas, muchas correcciones), también intentó enseñarme acerca del uso adecuado del apóstrofe (en lo relativo al lenguaje escrito y la gramática soy increíblemente malo).

[Guía de Seguridad del Administrador de Linux - GSAL]

Qué es y qué no es esta guía

Esta guía no es un documento general de seguridad. Esta guía está específicamente orientada a asegurar el sistema operativo Linux contra amenazas generales y específicas. Si necesitas un vistazo general sobre seguridad, por favor, cómprate "Practical Unix and Internet Security", disponible en www.ora.com. O'Reilly y asociados, el cual es uno de mis editores de libros de ordenadores preferidos (también tienen bonitas camisetas) y en el apéndice vienen listados una gran variedad de libros recomendados.

[Guía de Seguridad del Administrador de Linux - GSAL]

Modificaciones

Esto es algo parecido a una lista de modificaciones de LASG más o menos actualizada (AÑO/MES/DÍA)

1999/09/11 - cambios sustanciales a la tabla de contenidos, no estoy muy seguro de que me gusten.

Cómo determinar qué asegurar y cómo asegurarlo

¿Proteges datos (propietarios, confidenciales o de cualquier otro tipo), estás intentando mantener ciertos servicios en marcha (tu servidor de correo, servidor www, etc.), simplemente quieres proteger el hardware de daños? ¿Contra qué lo estás protegiendo? Los daños malintencionados (8 estaciones Sun Enterprise 10000), borrados (datos estadísticos, la colección de recetas de tu madre), etc. ¿Qué probabilidades hay de que ocurra un "mal" suceso, pruebas de red (a mi me ocurren a diario), intrusiones físicas (todavía no me ha ocurrido), ingeniería social ("Hola, soy Paco, del departamento de informática, necesito tu clave para que podamos cambiarla...").

Es necesario listar todos aquellos recursos (servidores, servicios, datos y otros componentes) que contengan datos, den servicios, formen parte de la infraestructura de tu compañía, etc. A continuación se detalla una pequeña lista:

- * Servidores físicos
- * Servidores de correo y servicios
- * Servidores de DNS y servicios
- * Servidores de WWW y servicios
- * Servidores de ficheros y servicios
- * Datos internos de la compañía, como registros contables
- * Infraestructura de la red (cables, hubs, switches, routers, etc.)
- * Sistema telefónico (PBX, buzones de voz, etc.)

Después necesitas averiguar contra qué lo quieres proteger:

- * Daños físicos (humo, agua, comida, etc.)
- * Borrado / modificación de datos (registros contables, deterioro de tu sitio web, etc.)
- * Exposición de datos (registros contables, etc.)
- * Continuidad de servicios (mantenimiento activo de los servidores de correo/www/ficheros)
- * Evitar que otros hagan uso ilegal/impropio de tus servicios (envíos masivos de correos, etc.)

Finalmente, ¿cuál es la probabilidad de que se dé un suceso determinado?

- * Escaneos de red - puedes apostar que a diario
- * Ingeniería social - varía, normalmente suelen ser objetivo la gente más vulnerable
- * Intrusión física - depende, bastante rara, pero un empleado hostil con un par de alicates podría causar mucho daño en un armario de telecomunicaciones
- * Empleados que venden datos a la competencia - ocurre

- * La competencia, que alquile a gente especializada para penetrar activamente en tu red - nadie suele hablar de esto, pero también ocurre

Una vez que has redactado una lista de tus recursos y de aquello que es necesario hacer, tienes que empezar a implementar la seguridad. Algunas técnicas (seguridad física para servidores, etc.), son bastante evidentes, en estos temas existen unas pautas de comportamiento de seguridad que por lo general están implementadas (poner claves a las cuentas, etc.). La gran mayoría de los problemas de seguridad suelen ser de origen humano, y la mayoría de los problemas que yo haya visto están debidos a la falta de educación/comunicación entre gente, no existe la "bala de plata" técnicamente hablando, incluso el mejor software necesita alguien para instalarse, configurarse y estar mantenido.

Al grano. He aquí una pequeña lista de los posibles resultados derivados de un incidente de seguridad:

- * Pérdidas de datos
- * Perdida directa de beneficios (ventas vía web, el servidor de ficheros inactivo, etc)
- * Costes en tiempo de personal
- * Pérdida de productividad del departamento de informática, así como de los trabajadores dependientes de su infraestructura
- * Implicaciones legales (registros médicos, registros contables de clientes, etc.)
- * Pérdida de la confianza por parte del cliente
- * Publicidad por parte de los medios de comunicación

Instalación segura de Linux

Una adecuada instalación de Linux es el primer paso para tener un sistema estable y seguro. Hay varios trucos que hacen más fácil la instalación, así como algunos asuntos que es mejor llevarlos a cabo durante la instalación (como la estructura de un disco duro).

Elección de los medios de instalación

Este es el elemento nº 1, que afectará a la velocidad de instalación y en buena medida a la seguridad. Personalmente mi método preferido es la instalación vía ftp, ya que colocar temporalmente una tarjeta de red en una máquina es algo rápido e indoloro, y alcanzar velocidades superiores a 1 Megabyte/seg acaba siendo una instalación rápida. La instalación desde el CD-ROM suele ser la más fácil, ya que se puede arrancar desde ellos, el Linux encuentra el CD y tira millas, sin tener que apuntar a directorios o tener que preocuparse por los nombres de ficheros con mayúsculas o minúsculas (al contrario que ocurre en la instalación basada en disco duro). Se trata de un medio original de Linux, y puedes tener la relativa certeza de que es seguro (dando por hecho que la fuente es reputada), aunque si eres algo paranoico tienes toda libertad para comprobar las firmas de los ficheros.

FTP - rápida, requiere una tarjeta de red, y un servidor de ftp (Una máquina Windows ejecutando algo como un warftpd también funciona).

HTTP - también rápida, y algo más segura que hacer una llamada a un FTP público

Samba - rápida, un buen método si dispones de una máquina windows (comparte el cdrom).

NFS - no tan rápida, pero puesto que el nfs está implementado en la mayoría de las redes UNIX existentes (y NT tiene un servidor NFS de MS gratis), es casi indolora. El método por NFS es el único método soportado por la opción kickstart de Red Hat.

CDROM - si tienes un lector de cdrom rápido, la mejor apuesta es introducir el cd y arrancar desde él, pulsar enter unas cuantas veces y ya estás listo. Ahora, se puede arrancar desde la mayoría de los CDROMs de Linux.

Disco duro - generalmente la más dolorosa, las ventanas confunden los nombres de fichero, la instalación desde una partición ext2 suele ser algo menos dolorosa.

Imágenes ISO en CD

Si quieres tostar tu propia distribución X en CD, vete a <http://freeiso.linuxsw.net> y pásala a CD.

Red Hat kickstart

Red Hat proporciona una característica para instalaciones automatizadas, que puede ser bastante útil. Simplemente hay que crear un fichero de texto con especificaciones de la instalación, y apuntar el instalador de Red Hat al mismo, después sentarse y dejarlo funcionar. Es muy útil si se ejecuta en múltiples máquinas, o si se les da a los usuarios un método de recuperación (suponiendo que sus ficheros de datos están seguros). Se puede conseguir más información en: <http://www.redhat.com/mirrors/LDP/HOWTO/KickStart-HOWTO.html>

Y aquí no acaba todo...

Así que tienes fresca una instalación de Linux (Red Hat, Debian, o lo que sea, por favor, por favor, NO instales versiones viejas, e intentes actualizarlas, es una pesadilla), pero hay probabilidades de que tengas una gran cantidad de software extra, y paquetes que quizás quisieras actualizar, o cosas que más te vale actualizar si no quieres que el sistema se vea comprometido en los primeros 15 segundos de vida (en el caso de BIND/Sendmail/etc.). Mantener una copia local del directorio de actualizaciones de tu distribución es una buena idea (hay una lista de erratas en distribuciones al final de este documento), y tenerlo disponible vía nfs/ftp o tostar un CD generalmente suele ser la forma más rápida de tenerlo disponible. De igual forma, existen otros elementos que podrías querer actualizar, yo por ejemplo utilizo una versión no root con chroot de Bind 8.1.2, disponible en el servidor de contribuciones (<ftp://contrib.redhat.com>), en lugar de la que viene por defecto, sin chroot, ejecutándose como root, la Bind 8.1.2 que viene con Red Hat Linux. También querrás eliminar cualquier tipo de software que no estés usando, así como reemplazarlo por otras versiones más seguras (tal como reemplazar rsh por ssh).

Conceptos generales, servidores versus estaciones de trabajo, etc

Hay muchos elementos que afectan a la seguridad de un ordenador. ¿Cómo de seguro necesita ser? ¿Está en red la máquina? ¿Habrán cuentas interactivas de usuarios (telnet/ssh)? ¿Se utilizarán las máquinas como estaciones de trabajo o se trata de un servidor? Esta última es de gran relevancia, pues las "estaciones de trabajo" y los "servidores" tradicionalmente han sido animales muy diferentes, aunque la línea se difumina con la introducción de potentes y baratos PCs, a medida que los sistemas operativos se aprovechan de ello. En el mundo de hoy en día, la principal diferencia entre los ordenadores no suele estar en el hardware, ni siquiera el SO (Linux es Linux, NT Server y NT Workstation son familia cercana, etc), sino en los paquetes de software que traen instalados (Apache, X, etc) y en el modo en que los usuarios acceden a la máquina (interactivamente, en la consola, etc.). Algunas reglas de carácter general que te ahorrarán bastantes penurias son las siguientes:

- * Mantén a los usuarios alejados de los servidores. Es decir: no les proporciones shells de login interactivos, a menos que sea un requerimiento absoluto.
- * Bloquea las estaciones de trabajo, da por hecho que los usuarios intentarán "arreglar" las cosas (qué caramba, incluso podrían ser hostiles, trabajadores temporales/etc).
- * Utiliza la criptografía allí donde sea posible para almacenar claves en texto simple, números de tarjetas de crédito y otro tipo de información delicada.
- * Escanea regularmente la red en busca de puertos abiertos, software instalado que no debería estar ahí, compáralo con resultados anteriores.

Recuerda: la seguridad no es una solución, es un modo de vida.

Hablando en términos generales, las estaciones de trabajo/servidores son utilizados por gente que no se suele preocupar en absoluto por la tecnología que llevan por debajo, lo único que quieren es tener listo su trabajo y recoger su correo de forma periódica. Sin embargo, hay muchos usuarios que tendrán la habilidad suficiente como para modificar sus estaciones de trabajo, para bien o para mal (instalar sniffers de paquetes, sitios ftp de warez, servidores www, robots de irc, etc). A esto hay que añadirle que la mayoría de los usuarios tienen acceso físico a sus estaciones de trabajo, lo cual quiere decir que tienes que bloquearles el acceso si quieres hacer las cosas bien.

- * Utiliza las claves de las BIOS para mantener a los usuarios alejados de la BIOS (nunca deberían estar ahí, recuerda también que las BIOS viejas tienen claves universales.)
- * Configura la máquina para que arranque únicamente del disco duro adecuado.
- * Pon clave al prompt de LILO.
- * No les des acceso de root a los usuarios, utiliza sudo para proporcionar acceso a comandos privilegiados cuando sea necesario.
- * Utiliza cortafuegos para que incluso si instalan servicios estos no sean accesibles al resto del mundo.
- * Observa regularmente la tabla de procesos, puertos abiertos, software instalado, etc. en busca de cambios.

- * Ten una política de seguridad escrita que los usuarios puedan entender, y fortalécela.
- * Borra todo tipo de objetos peligrosos (compiladores, etc), a menos que sean necesarios en un sistema.

Recuerda: seguridad en profundidad

Con la configuración adecuada, una estación Linux es casi a prueba de usuarios (nada es 100% seguro), y generalmente mucho más estable comparado con máquinas Wintel. Con el goce añadido de la administración remota (SSH/Telnet/NSH), puedes tener a tus usuarios contentos y productivos.

Los servidores son harina de otro costal, y por lo general son más importantes que las estaciones de trabajo (cuando se muere una estación de trabajo, el afectado es un usuario, si el que se muere es el servidor de correo/www/ftp/, es tu jefe el que telefona de mal humor). A menos que existan fuertes necesidades, mantén el número de usuarios con shells interactivos (bash, pine, basados en lynx, lo que sea) en el mínimo posible. Segmenta los servicios (ten un servidor de correo, un servidor de www, etc.) para minimizar los puntos únicos de fallo. En términos generales, un servidor adecuadamente configurado se ejecutará y no necesitará mayor mantenimiento (tengo un servidor de correo de un cliente que ha sido utilizado durante 2 años con unas 10 horas de tiempo total de mantenimiento). Cualquier actualización debería estar cuidadosamente planeada y ejecutada en un banco de pruebas. Algunos puntos importantes a recordar en cuanto a los servidores:

- * Restringe el acceso físico a los servidores.
- * Política del menor privilegio, así pueden estropearse menos cosas.
- * ¡HAZ COPIAS DE SEGURIDAD!
- * Comprueba con regularidad los servidores en busca de cambios (puertos, software, etc), para esto son estupendas las herramientas automatizadas.
- * Los cambios de software deberían estar cuidadosamente planeados/probados, pues pueden implicar efectos inversos (como que el kernel 2.2.x ya no utiliza ipfwadm, imagínate que se te olvidara instalar ipchains).

Minimizar los privilegios significa darles a los usuarios (y a los administradores en lo que a ellos respecta) la mínima cantidad de acceso requerido para hacer su trabajo. Darle a un usuario acceso de "root" a su estación de trabajo tendría sentido si todos los usuarios tuviesen grandes conocimientos de Linux, y fuesen de fiar, pero a menudo no lo son (en ambos sentidos). E incluso si lo fueran, sería una mala idea, pues crecerían las probabilidades de que instalasen software inseguro/defectuoso. Si todo lo que necesita un usuario es hacer un apagado/reiniciado de la estación de trabajo, es entonces ese el nivel de acceso que se les debería proporcionar. Con seguridad, no dejarías registros contables en un servidor con permisos de lectura al mundo para que los contables pudieran leerlos, y este concepto se extiende a la totalidad de la red. Limitar el acceso también limitará los daños, en caso de que se produzcan penetraciones en las cuentas (¿has leído alguna vez los post-it que mucha gente dejan pegados en los monitores?).

Ficheros del sistema

/etc/passwd

El fichero de contraseñas es sin discusión el fichero más crítico en Linux (y en la mayoría de otros Unix). Contiene el mapa de nombres de usuarios, identificaciones de usuarios y la ID del grupo primario al que pertenece esa persona. También puede contener el fichero real, aunque es más probable (y mucho más seguro) que utilice contraseñas con shadow para mantener las contraseñas en /etc/shadow. Este fichero TIENE que ser legible por todo el mundo, o si no comandos tan simples como ls dejarían de funcionar correctamente. El campo GECOS puede contener datos tales como el nombre real, el número de teléfono y otro tipo de cosas parecidas en cuanto al usuario, su directorio personal, que es el directorio en que se coloca al usuario por defecto si hace un login interactivo, y el shell de login tiene que ser un shell interactivo (como bash, o un programa de menús), y estar listado en /etc/shells para que el usuario pueda hacer login. El formato es:

```
nombreusuario:contraseña_cifrada:UID:GID:campo_GECOS:direct_personal:login_shell
```

Las contraseñas se guardan utilizando un hash de un sólo sentido (el hash utilizado por defecto es crypt, las distribuciones más nuevas soportan MD5, que es significativamente más robusto). Las contraseñas no pueden obtenerse a partir de la forma cifrada, sin embargo, se puede tratar de encontrar una contraseña utilizando fuerza bruta para pasar por el hash cadenas de texto y compararlas, una vez que encuentres una que coincide, sabes que has conseguido la contraseña. Esto no suele ser un problema por sí mismo, el problema surge cuando los usuarios escogen claves que son fácilmente adivinables. Las encuestas más recientes han demostrado que el 25% de las contraseñas se pueden romper en menos de una hora, y lo que es peor es que el 4% de los usuarios utilizan su propio nombre como contraseña. Los campos en blanco en el campo de la contraseña se quedan vacíos, así que se vería ":", lo cual es algo crítico para los cuatro primeros campos (nombre, contraseña, uid y gid).

/etc/shadow

El fichero de shadow alberga pares de nombres de usuario y contraseñas, así como información contable, como la fecha de expiración, y otros campos especiales. Este fichero debería protegerse a toda costa, y sólo el root debería tener acceso de lectura a él.

/etc/groups

El fichero de grupos contiene toda la información de pertenencia a grupos, y opcionalmente elementos como la contraseña del grupo (generalmente almacenado en gshadow en los sistemas actuales), este fichero debe ser legible por el mundo para que el sistema funcione correctamente. El formato es:

```
nombregroupo:contraseña_cifrada:GID:miembro1,miembro2,miembro3
```

Un grupo puede no contener miembros (p. ej., no está usado), sólo un miembro o múltiples miembros, y la contraseña es opcional (y no se suele usar).

/etc/gshadow

Similar al fichero shadow de contraseñas, este fichero contiene los grupos, contraseñas y miembros. De nuevo, este fichero debería ser protegido a toda costa, y sólo el usuario root debería tener permiso de lectura al mismo.

/etc/login.defs

Este fichero (/etc/login.defs) te permite definir algunos valores por defecto para diferentes programas como useradd y expiración de contraseñas. Tiende a variar ligeramente entre distribuciones e incluso entre versiones, pero suele estar bien comentado y tiende a contener los valores por defecto.

/etc/shells

El fichero de shells contiene una lista de shells válidos, si el shell por defecto de un usuario no aparece listado aquí, quizás no pueda hacer login interactivamente. Mira la sección sobre Telnetd para más información.

/etc/securetty

Este fichero contiene una lista de tty's desde los que el root puede hacer un login. Los tty's de la consola suelen ir de /dev/tty1 a /dev/tty6. Los puertos serie (pongamos que quieres hacer login como root desde módem) son /dev/ttyS0 y superiores por lo general. Si quieres permitirle al root hacer login vía red (una muy mala idea, utiliza sudo) entonces añade /dev/tty1 y superiores (si hay 30 usuarios conectados y el root intenta conectar, el root aparecerá como procedente de /dev/tty31). Generalmente, sólo se debería permitir conectar al root desde /dev/tty1, y es aconsejable deshabilitar la cuenta de root, sin embargo antes de hacer esto, por favor, instala sudo o un programa que permita al root acceder a comandos.

Seguridad de Ficheros / Sistema de ficheros

Una casa sólida necesita cimientos sólidos, si no se derrumbará. En el caso de Linux, esto es el sistema de ficheros ext2 (EXTendido, versión 2). Algo así como el standard UNIX de toda la vida. Soporta permisos de ficheros (lectura, escritura, ejecución, sticky bit, suid, guid, etc.), propiedad de ficheros (usuario, grupo, otros), y otro tipo de standards. Alguna de sus desventajas son: no se puede hacer journaling, y especialmente no hay Listas de Control de Acceso, las cuales se rumorea vendrán con el ext3. En la parte positiva, Linux tiene excelente software RAID, soportando bastante bien los Niveles 0, 1 y 5 (RAID no tiene que ver con la seguridad, pero por supuesto tiene que ver con la estabilidad).

Las utilidades básicas para interactuar con ficheros son: "ls", "chown", "chmod" y "find". Otras incluyen ln (para creación de enlaces), stat (muestra información de un fichero) y muchas más. En cuanto a la creación y mantenimiento de sistemas de ficheros por sí mismos, tenemos "fdisk" (el viejo fdisk), "mkfs" (MaKe FileSystem, que formatea particiones), y "fsck" (FileSystem Check, que suele arreglar problemas). De modo que, ¿qué es lo que estamos tratando de evitar que haga la gente hostil? (generalmente usuarios, y/o demonios de red alimentados con información maligna). Se puede comprometer con facilidad un sistema Linux si se consigue acceso a ciertos ficheros, por ejemplo la capacidad para leer un fichero de claves sin shadow da como resultado la posibilidad de ejecutar contraseñas cifradas contra crack, encontrando con facilidad las contraseñas débiles. Es un objetivo típico de los atacantes que vienen de la red (scripts CGI pobremente escritos suelen ser los favoritos). De otra forma, si un atacante puede escribir en el fichero de contraseñas, el o ella puede irrumpir en el sistema, o (presumiblemente peor) conseguir cualquier nivel de acceso que quiera. Este tipo de situaciones suelen estar causadas por lo general por "razas tmp", en las cuales un programa setuid (uno que se esté ejecutando con privilegios de root), escribe ficheros temporales, por lo general en /tmp , sin embargo muchos no comprueban la existencia de un fichero, y cuando se ejecuta el fichero setuid, boom, se borra /etc/passwd o incluso se le añaden entradas. Hay muchos ataques más similares a este, de modo que ¿cómo se pueden prevenir?

Es simple: configurando el sistema de ficheros correctamente cuando se instale. Dos directorios habituales a los que los usuarios tienen acceso son /tmp y /home, dividir esto en particiones separadas también evita que los usuarios llenen cualquier sistema de ficheros crítico (un / lleno es algo bastante malo). Un /home lleno podría dar como resultado la incapacidad de que los usuarios pudieran hacer un login (por eso el directorio del root está en /root). Poner /tmp y /home en particiones separadas es algo así como obligatorio si los usuarios tienen acceso al servidor, poner /etc, /var, y /usr en particiones separadas también es una muy buena idea.

Las herramientas principales para conseguir información sobre ficheros y sistemas de ficheros son relativamente simples y fáciles de usar, "df" (muestra el uso del disco) también mostrará el uso de los ínodos, "df -i" (los ínodos contienen información acerca de los ficheros tal como su localización en el disco duro, te puedes quedar sin ellos antes de quedarte sin disco duro si tienes muchos ficheros muy pequeños. Lo cual da mensajes de error del tipo "disco lleno", cuando en realidad "df" te dirá que queda espacio libre, "df -i" sin embargo mostraría los ínodos como todos usados). Esto es parecido a la reserva de entradas de ficheros de Windows, con vfat en realidad se almacenan los nombres en formato 8.3, utilizando múltiples entradas para nombres largos de ficheros, con un máximo de 512 entradas por directorio, lo cual es muy útil para averiguar dónde se ha ido todo el espacio, se usa como "du" (saca un listado de todo lo que hay en el directorio actual y por debajo de él a lo que

tengas acceso) o "du /nombre/directorio", opcionalmente utilizando "-s" para sacar un sumario, lo cual es útil para directorios como /usr/src/linux. Para conseguir información sobre ficheros específicos, la herramienta principal es ls (similar al comando "dir" del DOS), "ls" sólo saca nombres de ficheros/directorios, "ls -l" muestra información como los permisos de los ficheros, el tamaño, etc., y "ls -la" muestra los directorio y los ficheros que empiezan con ".", generalmente ficheros de configuración y directorios (.bash_history, .bash_logout, etc.). La utilidad "ls" tiene una docena de opciones para ordenar, basada en el tamaño, fecha, orden inverso, etc.; "man ls" para ver todos los detalles. Para detalles sobre un fichero en particular (fecha de creación, último acceso, ínodo, etc.) está el "stat", que simplemente da información de las estadísticas vitales de un fichero(s) dado, y es muy útil para ver si un fichero está en uso, etc.

Para manipular ficheros y carpetas, están las herramientas generales como cp, mv, rm (CoPy, MoVe y ReMove, copiar, mover y eliminar), al igual que herramientas para manipular la información de seguridad. chown es responsable de cambiar la propiedad del usuario y grupo de un determinado fichero (el grupo otros es siempre otros, similar al grupo 'todos' de NT o Novell). chmod (CHange MODE, cambio de modo) cambia los atributos de un fichero, siendo los básicos lectura, escritura y ejecución, al igual que está el setuid, setgid (establecer la id del ususario y grupo bajo la cual se ejecuta el programa, a menudo root), sticky bit, etc. Con el uso adecuado de asignaciones de usuarios a grupos, chmod y chwon, se pueden emular las ACL's (Listas de Control de Accesos) hasta cierto punto, pero es bastante menos flexible que los permisos para Sun/AIX/NT (aunque se rumorea que así será con ext3). Por favor, ten especial cuidado con los setuid/setgid, ya que cualquier problema en un programa/script de esos se puede agrandar enormemente.

Creo que también sería de mencionar "find". Encuentra ficheros (en esencia, lista ficheros), y también se puede utilizar con filtros basados en permisos/propiedad (también por tamaño, fecha y otros criterios diferentes). Un par de ejemplos rápidos para cazar programas setuid/guid:

para encontrar todos los programas setuid:

```
find / -perm +4000
```

para encontrar todos los programas setgid:

```
find / -perm +2000
```

La mayor parte de la seguridad son los permisos de usuarios. En Linux, un fichero es 'propiedad' de 3 entidades separadas, un Usuario, un Grupo y Otros (que es el resto). Se puede asignar a qué usuario pertenece un fichero y a qué grupo pertenece mediante:

```
chown usuario:grupo objeto
```

donde objeto es un fichero, directorio, etc. Si se quiere denegar el acceso de ejecución a los 3 propietarios, simplemente escribir:

```
chmod x="" objeto
```

donde x es a|g|u|o (All/User/Group/Other), fuerza que los permisos sean igual a "" (null, nada, ningún acceso) y el objeto es un fichero, directorio, etc. Este es con mucho, el método más rápido y efectivo de eliminar permisos y denegar totalmente el acceso a usuarios/etc (="" forzar a borrarlo). Recuerda que el root SIEMPRE puede cambiar los permisos de un fichero y ver/editar el fichero, Linux no aporta todavía protección a los usuarios sobre el root (lo cual muchos lo consideran algo bueno). Igualmente, cualquiera que sea dueño del directorio

en que está el objeto (sea un usuario/grupo/otros con los permisos adecuados sobre el directorio padre) puede, potencialmente, editar los permisos (y puesto que el root es dueño de / puede hacer cambios que involucren cualquier lugar del sistema de ficheros).

Borrado seguro de ficheros

Algo que muchos de nosotros olvidamos es que cuando se borra un fichero, en realidad no se ha ido. Incluso se sobrescribe, se vuelve a formatear el disco duro o se lo intenta destruir de cualquier otra forma, hay posibilidades de que pueda ser recuperado, y por lo general, los servicios de recuperación de datos sólo cuestan unos pocos cientos de miles de pesetas, de modo que quizás les mereciese la pena a los atacantes en tiempo y en dinero. El truco consiste en desordenar los datos, alterando los bits magnéticos (alias los 1's y 0's) para que una vez hubiera terminado, no quedasen trazas del original (es decir, bits magnéticos cargados de la misma forma que estaban originariamente). Se han escrito dos programas (ambos llamados wipe) a tal efecto.

wipe (durakb@crit2.univ-montp2.fr)

wipe borra datos con seguridad, sobrescribiendo el fichero múltiples veces con varios patrones de bits, p. ej., todo 0's, luego todo 1's, luego alternando 1's y 0's, etc. Se puede utilizar wipe en ficheros o en dispositivos, si se utiliza con ficheros, hay que recordar que las fechas de creación de ficheros, permisos, etc., no serán borrados, de modo que asegúrate de borrar el dispositivo si se necesita eliminar toda traza de cualquier cosa. Se puede conseguir en: <http://gsu.linux.org.tr/wipe/>

wipe (thomassr@erols.com)

Este también borra datos de forma segura, sobrescribiéndolos múltiples veces, sin embargo este no soporta el borrado de dispositivos. Se puede conseguir en: <http://users.erols.com/thomassr/zero/download/wipe/>

Listas de Control de Acceso (ACL's)

Uno de los componentes que se echan en falta es un sistema de ficheros Linux Listas de Control de Acceso (ACL's) en lugar de los habituales Usuario, Grupo, Otros, con su docena o así de permisos. Las ACL's te permiten un control de accesos del sistema de ficheros más afinado, por ejemplo, se puede otorgar al usuario "paco" acceso total a un fichero, a "maría" permiso de lectura, al grupo de ventas, permiso de cambio, al grupo de contabilidad, permiso de lectura, y ningún permiso para el resto. Bajo los permisos existentes para Linux, no se podría hacer esto. De aquí la necesidad de las ACL's.

El proyecto de trustees (ACL) de Linux es una serie de parches y utilidades del kernel para configurar accesos ACL al sistema de ficheros. Esta solución es algo cutre, ya que mantiene los permisos en un fichero, y actúa como una capa de filtrado entre el fichero y los usuarios, lo cual no es un sistema basado en ACL propiamente dicho (pero es un empuje). Se puede conseguir en: <http://www.braysystems.com/linux/trustees.html>

PAM

"Pluggable Authentication Modules" para Linux, es una suite de librerías compartidas que permiten al administrador local del sistema escoger cómo autentifican a los usuarios las aplicaciones. Literalemente extraído de la documentación de PAM, yo no lo hubiera podido decir mejor. ¿Pero qué significa en realidad? Por ejemplo, tomemos el programa "login", cuando un usuario se conecta a un tty (vía puerto serie o sobre la red), un programa responde la llamada (getty para líneas en serie, normalmente telnet o ssh para conexiones de red) e inicia el programa "login", y luego pide el típico nombre de usuario, seguido de la contraseña, lo cual se compara con el fichero /etc/passwd. Todo esto está bien y es muy mono, hasta que tienes una fenomenal tarjeta de autenticación nueva y quieres utilizarla. Bueno, pues tendrás que recompilar login (y cualquier otra aplicación que vaya a hacer la autenticación según el nuevo método) de modo que soporten el sistema nuevo. Como te puedes imaginar, esto lleva bastante trabajo y está sujeto a errores.

PAM introduce una capa de middleware entre la aplicación y el mecanismo real de autenticación. Una vez que el programa está PAMificado, podrá ser utilizado por el programa cualquier método de autenticación que soporte PAM. Además de esto, PAM puede manejar cuentas y datos de sesiones, lo cual no suelen hacer bien los mecanismos habituales de autenticación. Por ejemplo, usando PAM se puede deshabilitar con facilidad el acceso de login a los usuarios normales entre las 6pm y las 6am, y cuando hagan login, se les puede autenticar vía scanner retinal. Por defecto, los sistemas Red Hat son conscientes de PAM, y las versiones más recientes de Debian también (más abajo puedes echar un vistazo a la tabla de sistemas PAMificados). De esta forma, en un sistema con soporte PAM, todo lo que tengo que hacer para implementar el shadow en contraseñas es convertir los ficheros de contraseñas y de grupos, y posiblemente añadir una o dos líneas a algunos ficheros de configuración de PAM (si no las tienen ya añadidas). En resumen, PAM proporciona una gran cantidad de flexibilidad al manejar la autenticación de usuarios, y soportará otras características en el futuro, como firmas digitales, con el único requerimiento de uno o dos módulos PAM para manejarlo. Es necesario este tipo de flexibilidad si se pretende que Linux sea un sistema operativo de tipo empresarial. Las distribuciones que no vengan como "PAMificadas" se pueden convertir, pero requiere mucho esfuerzo (tienes que recompilar todos los programas con soporte PAM, instalar PAM, etc), probablemente sea más fácil cambiarse a una distribución PAMificada si va a suponer un requisito. PAM suele venir con documentación completa, y si estás buscando una vista general, deberías visitar: <http://www.sun.com/software/solaris/pam/>.

Otros beneficios de un sistema orientado a PAM es que ahora se puede hacer uso de un dominio NT para autenticar usuarios, lo cual quiere decir que se pueden plantar estaciones Linux en una red Microsoft ya existente sin tener que comprar NIS / NIS+ para NT y pasar por el calvario de instalarlo.

| Distribución | Versión | Soporte PAM |
|--------------|--------------------|---------------|
| Red Hat | 5.0, 5.1, 5.2, 6.0 | Completamente |
| Debian | 2.1 | Sí |
| Caldera | 1.3, 2.2 | Completamente |
| TurboLinux | 3.6 | Completamente |

Existen más distribuciones que soportan PAM y que no aparecen en la lista, así

que por favor, infórmame de ellas.

PAM Cryptocard Module

<http://www.jdimedia.nl/igmar/pam/>

Pam Smart Card Module

<http://www.linuxnet.com/applications/applications.html>

Seguridad Física / de Arranque

Acceso Físico

Este área viene cubierta en profundidad en el libro "Practical Unix and Internet Security", pero daré un breve repaso a lo básico. Alguien apaga el servidor principal de contabilidad, lo vuelve a encender, arranca desde un disquete especial y transfiere el fichero pagas.db a un ftp en el extranjero. A menos que el servidor de contabilidad esté bloqueado, ¿qué le impide a un usuario malintencionado (o al personal de limpieza del edificio, el chico de los recados, etc.) hacer tal cosa? He escuchado historias de terror acerca de personal de limpieza desenchufando los servidores para poder enchufar sus aparatos de limpieza. He visto cómo la gente pulsaba por accidente el pequeño botón de reset y reiniciaban los servidores (no es que lo haya hecho yo alguna vez). Tiene sentido bloquear los servidores en una habitación segura (o incluso en un armario). También es una buena idea situar a los servidores en una superficie elevada, para evitar daños en el caso de inundaciones (ya sea por un agujero en el techo o lo que sea).

La BIOS del ordenador

La BIOS del ordenador es uno de sus componentes de más bajo nivel, controla la forma en que el ordenador arranca y otro tipo de cosas. Las BIOS viejas tiene fama de tener claves universales, asegúrate de que tu bios es reciente y que no contiene semejante puerta trasera. La bios se puede utilizar para bloquear la secuencia de arranque de un equipo, limitándola a C: únicamente, por ejemplo, al primer disco duro, lo cual es una buena idea. Deberías utilizar la bios para eliminar la disquetera (el servidor típico no va a necesitar utilizarla), y puede evitar que los usuarios copien datos de la máquina a disquetes. También puedes eliminar los puertos serie en las máquinas de los usuarios, de tal forma que puedan instalar módems, la mayoría de los ordenadores modernos utilizan teclados y ratones PS/2, así que quedan pocas razones por las que podría necesitarse un puerto serie (además de que se comen IRQ's). Lo mismo sirve para el puerto paralelo, permitiendo a los usuarios imprimir obviando la red, o dándoles la oportunidad de instalar una grabadora de CDRom o un disco duro externo, lo cual puede disminuir la seguridad en buena medida. Como se puede ver, esto es un añadido a la política del menor privilegio, y puede disminuir considerablemente los riesgos, al igual que facilitar la administración de la red (menos conflictos de IRQs, etc.) Por supuesto que existen programas para obtener las contraseñas de la BIOS de un ordenador, hay uno disponible en http://www.esiea.fr/public_html/Christophe.GRENIER/, y está disponible para DOS y Linux.

LILO

Una vez que el ordenador ha decidido arrancar de C:, LILO (o cualquier otro gestor de arranque que utilices) despega. La mayoría de los gestores de arranque permiten algún tipo de flexibilidad en el modo en que se arranca el sistema, especialmente LILO, pero también es una espada de dos filos. Puedes pasarle argumentos a LILO a la hora de arrancar, siendo el argumento más dañino (desde el punto de vista de la seguridad) "imagenname single", lo cual arranca Linux en modo de único usuario, y por defecto, la mayoría de las distribuciones te vuelcan a un prompt de root en un shell de comandos sin preguntar contraseñas u otro tipo de mecanismos de seguridad. Hay varias técnicas para minimizar este riesgo.

delay=x

esto controla la cantidad de tiempo (en décimas de segundo) que LILO espera a

que el usuario introduzca datos antes de arrancar la opción por defecto. Uno de los requerimientos de la seguridad de nivel C2 es que este intervalo sea puesto a 0 (obviamente, cualquier máquina con arranque dual acaba con cualquier tipo de seguridad). Es una buena idea poner esto a 0, a menos que el sistema arranque dualmente.

```
prompt
```

fuerza al usuario a introducir algo, LILLO no arrancará el sistema automáticamente. Esto podría ser útil en servidores, como una forma de eliminar los reinicios sin que esté presente una persona, pero lo típico es que si el hacker tiene capacidad para reiniciar la máquina, podría reescribir el MBR con nuevas opciones de arranque. Si le añades una opción de cuenta atrás, el sistema continuará arrancando después de que haya terminado la cuenta atrás.

```
restricted
```

pide una contraseña si se pasan opciones de tiempo de arranque (tales como "linux single"). Asegúrate de que utilizas esto en cada imagen (si no, el servidor necesitará una contraseña para arrancar, lo cual está bien si no planeas arrancarlos remotamente nunca).

```
boot=/dev/hda
```

```
map=/boot/map
```

```
install=/boot/boot.b
```

```
prompt
```

```
timeout=100
```

```
default=linux
```

```
image=/boot/vmlinuz-2.2.5
```

```
label=linux
```

```
root=/dev/hda1
```

```
read-only
```

```
restricted
```

```
password=aqu1_va_la_c0ntRaSeña
```

Esto reinicia el sistema utilizando el kernel /boot/vmlinuz-2.2.5, almacenado en el MBR del primer disco IDE del sistema, el prompt impediría hacer reinicios desatendidos, sin embargo está implícito en la imagen, de modo que puede arrancar "linux" sin problemas, pero pediría una contraseña si introduces "linux single", de modo que si quiere ir al modo "linux single", tienes 10 segundos para escribirlo, en cuyo punto te preguntaría por la contraseña ("aqu1_va_la_c0ntRaSeña"). Combina esto con una BIOS configurada para arrancar sólo desde C: y protegida con contraseña y has conseguido un sistema bastante seguro. Una medida menor de seguridad que puedes tomar para asegurar el fichero lilo.conf es dejarlo invariable, utilizando el comando "chattr". Para hacer el fichero invariable, simplemente teclea:

```
chattr +i /sbin/lilo.conf
```

y esto evitará cualquier cambio (accidental o de otro tipo) en el fichero

lilo.conf. Si quieres modificar el fichero lilo.conf necesitarás quitar el flag de invariable:

```
chattr -i /sbin/lilo.conf
```

sólo el root tiene acceso al flag de invariable.

Seguridad de contraseñas

En todo sistema operativo tipo UNIX se dan varias constantes, y una de ellas es el fichero `/etc/passwd` y la forma en que funciona. Para que la autenticación de usuario funcione correctamente se necesitan (como mínimo) algún tipo de fichero(s) con UID a mapas de nombres de usuarios, GID a mapas de nombres de grupos, contraseñas para todos los usuarios y demás información variada. El problema es que todo el mundo necesita acceso al fichero de contraseñas, cada vez que se hace un `ls`, se verifica el fichero de contraseñas, de modo que ¿cómo se consigue almacenar todas las contraseñas con seguridad y a la vez mantenerlas legibles por el mundo? Durante muchos años, la solución ha sido bastante simple y efectiva, simplemente, haz un hash de las contraseñas y guarda el hash, cuando un usuario necesite autenticar, toma la contraseña que introduce, pásala por el hash y si coincide, evidentemente se trataba de la misma contraseña. El problema que tiene esto es que la potencia computacional ha crecido enormemente, y ahora se puede coger una copia del fichero de contraseñas e intentar abrirlo mediante fuerza bruta en una cantidad de tiempo razonable. Para resolver esto hay varias soluciones:

- * Utiliza un algoritmo de hashing "mejor", como MD5. Problema: se pueden romper muchas cosas si están esperando algo más.
- * Almacena las contraseñas en alguna otra parte. Problema: el sistema/usuarios siguen necesitando tener acceso a ellas, y podría hacer que fallasen algunos programas si no están configurados de esta forma.

Varios SO's han escogido la primera opción, Linux ha implementado la segunda desde hace tiempo, se llama contraseñas con shadow. En el fichero de contraseñas, se reemplaza la contraseña por una 'x', lo cual le indica al sistema que verifique tu contraseña contra el fichero shadow (se hace lo mismo con el fichero de grupos y sus contraseñas). Parece lo suficientemente simple, pero hasta hace bien poco, implementar el shadow era una ardua tarea. Había que recompilar todos los programas que verificasen la contraseña (login, ftpd, etc, etc) y esto, por supuesto, implica una considerable cantidad de esfuerzo. Es aquí donde brilla Red Hat, con su confianza en PAM.

Para implementar contraseñas con shadow hay que hacer dos cosas. La primera es relativamente simple, cambiar el fichero de contraseñas, pero la segunda puede ser un calvario. Hay que asegurarse que todos tus programas tienen soporte para contraseñas con shadow, lo cual puede ser bastante penoso en algunos casos (esta es una más que importante razón por la cual un mayor número de distribuciones deberían venir con PAM).

Debido a la confianza de Red Hat en PAM para la autenticación, para implementar un esquema nuevo de autenticación todo lo que se necesita es añadir un módulo PAM que lo entienda y editar el fichero de configuración para cualquier programa (digamos el login) permitiéndole que use ese módulo para hacer la autenticación. No hace falta recompilar, y hay poco tejemaneje, ¿a que sí? En Red Hat 6.0, durante la instalación se te da la opción de elegir contraseñas con shadow, o puedes implementarlas más tarde vía las utilidades `pwconv` y `grpconv` que vienen con el paquete de utilidades shadow. La mayoría del resto de distribuciones también tienen soporte para contraseñas con shadow, y la dificultad de implementación varía de un modo u otro. Ahora, para que un atacante mire las contraseñas con hash, tiene que esforzarse un poco más que simplemente copiar el fichero `/etc/passwd`. También asegúrate de ejecutar cada cierto tiempo `pwconv`, para tener la certeza de que todas las contraseñas efectivamente tienen shadow. Hay veces que las contraseñas se quedan en `/etc/passwd` en lugar de enviarse a `/etc/shadow` como deberían, lo cual hacen algunas utilidades que editan el fichero de contraseñas.

Reventando contraseñas

En Linux las contraseñas se guardan en formato hash, sin embargo ello no las hace irrecuperables, no es posible hacer ingeniería inversa de la contraseña a partir del hash resultante, sin embargo sí que puedes hacer un hash de un lista de palabras y compararlas. Si el resultado coincide, entonces has encontrado la contraseña, es por esto que es crítica la elección de buenas contraseñas, y las palabras sacadas de un diccionario son una idea horrible. Incluso con un fichero de contraseñas con shadow, el root puede acceder a las contraseñas, y si se han escrito scripts o programas que se ejecuten como root (pongamos un script CGI para www) los atacantes pueden recuperar el fichero de contraseñas. La mayoría del software para reventar contraseñas también te permite la ejecución en múltiples hosts en paralelo para acelerar las cosas.

John the ripper ("Juan el destripador")

Un eficiente revienta contraseñas disponible en:
<http://www.false.com/security/john/>

Crack

El revienta contraseñas original y ampliamente extendido (según me consta), lo puedes conseguir en: <http://www.users.dircon.uk/~crypto/>

Saltine cracker

Otro revienta contraseñas con capacidades de red, lo puedes descargar de:
<http://www.thegrid.net/gravitino/products.html>

VCU

VCU (Velocity Cracking Utilities) es un programa basado en windows para ayudar a reventar contraseñas "VCU intenta facilitar la tarea de reventar contraseñas a usuarios de ordenadores de cualquier nivel". Lo puedes descargar desde:
<http://wilter.com/wf/vcu/>

Espero que esto sea suficiente motivo para utilizar contraseñas con shadow y un hash más robusto como el MD5 (el cual soporta Red Hat 6.0, no conozco otras distribuciones que lo soporten).

Almacenamiento de Contraseñas

Esto es algo que la mayoría de la gente no suele tener en cuenta. ¿Cómo se pueden almacenar las contraseñas de forma segura? El método más obvio es memorizarlas, pero suele tener sus inconvenientes, si se administran 30 sitios diferentes, por lo general se tendrán 30 contraseñas diferentes, y una buena contraseña tiene más de 8 caracteres de longitud, y por lo general no es la cosa más fácil de recordar. Esto conduce a que mucha gente utilice la misma contraseña en diferentes sistemas (vamos, admítelo). Una de las formas más sencillas es escribir las contraseñas. Por lo general, esto suele ser un grandísimo NO-NO; te sorprendería saber lo que encuentra la gente echando un vistazo, y lo que encuentran si lo están buscando. Una mejor opción es almacenar las contraseñas en un formato cifrado, generalmente de forma electrónica en tu ordenador o en el palm pilot, de forma sólo hay que recordar una contraseña para desbloquear el resto. Para esto se puede utilizar algo tan simple como PGP o GnuPG.

Gpasman

Gpasman es una aplicación que requiere el gtk (es relativamente fácil de instalar en un sistema que no sea Gnome, sólo hay que cargar las librerías gtk). Cifra tus contraseñas utilizando el algoritmo rc2. Al inicio del programa se introduce la contraseña maestra, y (suponiendo que es correcta) se presenta una lista de tus cuentas de usuarios, sitios, contraseñas y un campo de comentario. Gpasman está disponible en:

<http://www.student.wau.nl/~olivier/gpasman/>

Strip

Strip es un programa de almacenamiento seguro de contraseñas para palm pilot, y también se puede utilizar para generar contraseñas. Tiene licencia GNU y se encuentra disponible en: <http://www.zetetic.net/products.html>

Seguridad básica de servicios de red

¿Qué se está ejecutando y con quién se está hablando?

No se pueden empezar a asegurar servicios hasta que no se sepa qué se está ejecutando. Para este tipo de tareas, `ps` y `netstat` no tienen precio; `ps` dice qué se está ejecutando (`httpd`, `inetd`, etc) y `netstat` te dirá cuál es el estado de los puertos (llegados a este punto, estamos interesados en los puertos que están abiertos y escuchando, es decir, esperando conexiones). Se les puede echar un vistazo a los diferentes ficheros de configuración que controlan los servicios de red.

Salida de PS

El programa `ps` nos muestra el estado de procesos (información disponible en el sistema de ficheros virtual `/proc`). Las opciones más comúnmente utilizadas son "`ps -xau`", que muestra algo así como toda la información que siempre quisiste saber. Por favor, ten en cuenta: estas opciones cambian entre sistemas UNIX, Solaris, SCO, todos se comportan de manera diferente (lo cual es increíblemente molesto). Lo que viene a continuación es una salida típica de una máquina (utilizando "`ps -xau`").

```
USER PID %CPU %MEM SIZE RSS TTY STAT START TIME COMMAND
bin 320 0.0 0.6 760 380 ? S Feb 12 0:00 portmap
daemon 377 0.0 0.6 784 404 ? S Feb 12 0:00 /usr/sbin/atd
named 2865 0.0 2.1 2120 1368 ? S 01:14 0:01 /usr/sbin/named -u named -g named
-t /home/named
nobody 346 0.0 18.6 12728 11796 ? S Feb 12 3:12 squid
nobody 379 0.0 0.8 1012 544 ? S Feb 12 0:00 (dnsserver)
nobody 380 0.0 0.8 1012 540 ? S Feb 12 0:00 (dnsserver)
nobody 383 0.0 0.6 916 416 ? S Feb 12 0:00 (dnsserver)
nobody 385 0.0 0.8 1192 568 ? S Feb 12 0:00 /usr/bin/ftppget -S 1030
nobody 392 0.0 0.3 716 240 ? S Feb 12 0:00 (unlinkd)
nobody 1553 0.0 1.8 1932 1200 ? S Feb 14 0:00 httpd
nobody 1703 0.0 1.8 1932 1200 ? S Feb 14 0:00 httpd
root 1 0.0 0.6 776 404 ? S Feb 12 0:04 init [3]
root 2 0.0 0.0 0 0 ? SW Feb 12 0:00 (kflushd)
root 3 0.0 0.0 0 0 ? SW Feb 12 0:00 (kswapd)
root 4 0.0 0.0 0 0 ? SW Feb 12 0:00 (md_thread)
root 64 0.0 0.5 736 348 ? S Feb 12 0:00 kerneld
root 357 0.0 0.6 800 432 ? S Feb 12 0:05 syslogd
```

```

root 366 0.0 1.0 1056 684 ? S Feb 12 0:01 klogd
root 393 0.0 0.7 852 472 ? S Feb 12 0:00 crond
root 427 0.0 0.9 1272 592 ? S Feb 12 0:19 /usr/sbin/sshd
root 438 0.0 1.0 1184 672 ? S Feb 12 0:00 rpc.mountd
root 447 0.0 1.0 1180 644 ? S Feb 12 0:00 rpc.nfsd
root 458 0.0 1.0 1072 680 ? S Feb 12 0:00 /usr/sbin/dhcpd
root 489 0.0 1.7 1884 1096 ? S Feb 12 0:00 httpd
root 503 0.0 0.4 724 296 2 S Feb 12 0:00 /sbin/mingetty tty2
root 505 0.0 0.3 720 228 ? S Feb 12 0:02 update (bdflush)
root 541 0.0 0.4 724 296 1 S Feb 12 0:00 /sbin/mingetty tty1
root 1372 0.0 0.6 772 396 ? S Feb 13 0:00 inetd
root 1473 0.0 1.5 1492 1000 ? S Feb 13 0:00 sendmail: accepting connections on
port 25
root 2862 0.0 0.0 188 44 ? S 01:14 0:00 /usr/sbin/holelogd.named
/home/named/dev/log
root 3090 0.0 1.9 1864 1232 ? S 12:16 0:02 /usr/sbin/sshd
root 3103 0.0 1.1 1448 728 p1 S 12:16 0:00 su -root 3104 0.0 1.3 1268 864 p1 S
12:16 0:00 -bash
root 3136 0.0 1.9 1836 1212 ? S 12:21 0:04 /usr/sbin/sshd

```

Los interesantes son: portmap, named, Squid (y su servidor dns, los procesos hijos unlinkd y ftpget), httpd, syslogd, sshd, rpc.mountd, rpc.nfsd, dhcpd, inetd, y sendmail (este servidor parece estar proveyendo servicios de puerta de enlace, correo y compartición de ficheros FNS). La forma más fácil de aprender a leer la salida de ps es irse a la página del manual de ps y aprender a qué se refiere cada campo (la mayoría se explican por sí mismos, tales como el %CPU, mientras que algunos como SIZE son un poco más oscuros: SIZE es el número de páginas de memoria de 4k que está utilizando un programa). Para averiguar qué programas se están ejecutando, una apuesta segura es hacer 'man <nombre_de_comando>'; lo cual casi siempre suele sacar la página del manual que pertenece a ese servicio (como httpd). Te darás cuenta de que servicios como telnet, ftpd, identd y otros no aparecen aunque estén ahí. Esto es debido a que se ejecutan desde inetd, el "superservidor". Para encontrar estos servicios, mira en /etc/inetd.conf o en la salida de "netstat -vat".

Salida de Netstat

netstat informa acerca de casi cualquier cosa que se pueda imaginar relacionada con la red. Es especialmente buena para sacar listados de conexiones y sockets activos. Al usar netstat se puede encontrar qué interfaces están activas en qué puertos. Lo que viene a continuación es la salida típica de un servidor, con netstat -an.

Active Internet connections (including servers)

Proto Recv-Q Send-Q Local Address Foreign Address State

```
tcp 0 0 24.108.11.200:80 205.253.183.122:3661 ESTABLISHED
tcp 0 0 0.0.0.0:1036 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:80 0.0.0.0:* LISTEN
tcp 0 0 10.0.0.10:53 0.0.0.0:* LISTEN
tcp 0 0 28.208.55.254:53 0.0.0.0:* LISTEN
tcp 0 0 127.0.0.1:53 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:139 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:2049 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:635 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:21 0.0.0.0:* LISTEN
tcp 0 0 0.0.0.0:111 0.0.0.0:* LISTEN
udp 0 0 127.0.0.1:1031 0.0.0.0:*
udp 0 0 0.0.0.0:1029 0.0.0.0:*
udp 0 0 0.0.0.0:800 0.0.0.0:*
udp 0 0 0.0.0.0:1028 0.0.0.0:*
udp 0 0 10.0.0.10:53 0.0.0.0:*
udp 0 0 28.208.55.254:53 0.0.0.0:*
udp 0 0 127.0.0.1:53 0.0.0.0:*
udp 0 0 10.1.0.1:138 0.0.0.0:*
udp 0 0 10.1.0.1:137 0.0.0.0:*
udp 0 0 10.0.0.10:138 0.0.0.0:*
udp 0 0 10.0.0.10:137 0.0.0.0:*
udp 0 0 0.0.0.0:138 0.0.0.0:*
udp 0 0 0.0.0.0:137 0.0.0.0:*
udp 0 0 0.0.0.0:2049 0.0.0.0:*
udp 0 0 0.0.0.0:635 0.0.0.0:*
udp 0 0 0.0.0.0:514 0.0.0.0:*
udp 0 0 0.0.0.0:111 0.0.0.0:*
```

```
raw 0 0 0.0.0.0:1 0.0.0.0:*
```

```
raw 0 0 0.0.0.0:6 0.0.0.0:*
```

En mi opinión la salida numérica es más fácil de leer (una vez que se memoriza /etc/services). Los campos que nos interesan son el primero, el tipo de servicio, el cuarto campo, que es la dirección IP de la interfaz y el puerto, la dirección externa (si no es 0.0.0.0.* significa que alguien le está hablando activamente), y el estado del puerto. La primera línea es un cliente remoto hablando con el servidor de Web de esta máquina (puerto 80). Cuando se ve el servidor www escuchando en 0.0.0.0:80 que significa, todos los interfaces, puerto 80, seguidos del servidor DNS ejecutándose en las 3 interfaces, un servidor samba (139), un servidor de correo (25), un servidor NFS (2049), etc. Observarás listado el servidor de ftp (21), que aunque se ejecuta desde inetd, y aunque actualmente no está en uso (p. ej., no hay nadie activo haciendo un ftp), sale en el listado de la salida de netstat. Lo cual convierte a netstat en una herramienta especialmente útil para averiguar qué es lo que está activo en una máquina, haciendo más sencillo el inventariado en el servidor del software activo e inactivo.

```
lsof
```

lsof es un práctico programa cuya idea es similar a la de ps, excepto en que muestra qué ficheros/etc están abiertos, lo cual puede incluir sockets de red. Desafortunadamente, el lsof medio saca bastante información, de modo que será necesario utilizar grep o redireccionarlo mediante less ("lsof | less") para hacerlo más cómodo de leer.

```
squid 9726 root 4u inet 78774 TCP localhost:2074->localhost:2073 (ESTABLISHED)
```

```
squid 9726 root 5u inet 78777 TCP localhost:2076->localhost:2075 (ESTABLISHED)
```

```
squid 9726 root 6u inet 78780 TCP localhost:2078->localhost:2077 (ESTABLISHED)
```

```
squid 9726 root 7w CHR 1,3 6205 /dev/null
```

```
squid 9726 root 14u inet 78789 TCP host1:3128 (LISTEN)
```

```
squid 9726 root 15u inet 78790 UDP host1:3130
```

```
squid 9726 root 16u inet 78791 UDP host1:3130
```

```
squid 9726 root 12u inet 167524 TCP host1:3128->host2:3630 (ESTABLISHED)
```

```
squid 9726 root 17u inet 167528 TCP host1:3424->www.ejemplo.org:http (SYN_SENT)
```

Este ejemplo muestra que se tiene ejecutándose a Squid, escuchando en los puertos 3128 y 3130, las últimas dos líneas muestran una conexión abierta desde un host interno al servidor de Squid y la acción resultante que ha emprendido Squid para cumplir con la solicitud (ir a www.playboy.com) host1 es el servidor de Squid y host2 es la máquina cliente haciendo la petición. Es una herramienta que no tiene precio para hacerse una idea exacta de qué es lo que está ocurriendo con tu servidor en la red. Se puede conseguir lsof con algunas distribuciones. Ten en cuenta que las versiones de losf compiladas para las versiones del kernel 2.0.x no funcionarán con el kernel 2.2.x y vice versa, pues hay bastantes cambios. El sitio primario de lsof es:
<ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/>

Ficheros básicos de configuración de red

Hay varios ficheros de configuración importantes, que controlan qué servicios ejecuta Linux y cómo lo hacen. Por desgracia, muchos de ellos se encuentran en diferentes localizaciones dependiendo de qué/cómo instalaras Linux y los servicios. Los lugares habituales son:

Fichero de configuración del servidor Inetd:

/etc/inetd.conf

Ficheros de inicio de varios tipos:

/etc/rc.d/*

/etc/*

Lo mejor que se puede hacer es imaginarse qué servicios se quiere ejecutar, y deshabilitar/borrar el resto. Échale un vistazo a la sección apropiada de gestión de paquetes de tu sistema (RPM, dpkg, tarballs)

inetd.conf

inetd.conf es el responsable de iniciar los servicios, generalmente aquellos que no necesitan ejecutarse de continuo, o que están basados en sesiones (como telnet o ftpd). Ello es debido a que la sobrecarga que supondría ejecutar un servicio constantemente (como telnet) sería mayor que el costo de inicio ocasional (o que arrancar in.telnetd) cuando el usuario quisiera utilizarlo. Para algunos servicios (como DNS) que sirven a muchas conexiones rápidas, la sobrecarga de arrancar servicios cada pocos segundos sería mayor que tenerlo constantemente ejecutándose. De igual forma ocurre con servicios como DNS y el correo, donde el tiempo es crítico, sin embargo unos pocos segundos de retraso en empezar una sesión de ftp no le hacen daño a nadie. La página del manual de inetd.conf cubre los básicos ("man inetd.conf"). El servicio en sí se llama inetd y se ejecuta al arrancar, de modo que se le puede parar/arrancar/recargar manipulando el proceso inetd. Cada vez que se hagan cambios a inetd.conf, hay que reiniciar inetd para hacer efectivos los cambios, killall -1 inetd lo reiniciará correctamente. Como de costumbre, las líneas del inetd.conf se pueden comentar con un # (lo cual es una forma muy simple y efectiva de deshabilitar servicios como rexec). Se aconseja deshabilitar tantos servicios de inetd.conf como sea posible, por lo general los que se suelen usar son ftp, pop e imap. Se debería reemplazar telnet y los servicios r por el SSH y servicios como systat/netstat y finger proporcionan demasiada información. El acceso a programas arrancados por inetd se puede controlar con facilidad mediante el uso de TCP_WRAPPERS.

TCP_WRAPPERS

Usar TCP_WRAPPERS hace que el asegurar servidores contra intrusiones externas sea bastante más simple y menos doloroso de lo que te imaginas. TCP_WRAPPERS se controla desde dos ficheros:

/etc/hosts.allow

/etc/hosts.deny

Primero se comprueba hosts.allow, y las reglas se comprueban desde la primera a la última. Si encuentra una regla que te permita específicamente entrar (p. ej., una regla que permita a tu host, dominio, máscara de subred, etc.) te deja

conectarte al servicio. Si no puede encontrar ninguna regla que te corresponda en `hosts.allow`, entonces va a comprobar `hosts.deny` en busca de una regla que te deniegue la entrada. De nuevo comprueba las reglas de `hosts.deny` desde la primera a la última, y la primera regla que encuentre que te deniega acceso (p. ej., una regla que deshabilite tu host, dominio, máscara de subred, etc.) significa que no te deja entrar. Si tampoco puede encontrar una regla denegándote la entrada, entonces por defecto te deja entrar. Si eres tan paranoico como yo, la última regla (o la única regla si se va a utilizar una política por defecto no optimista en cuanto a seguridad) debería ser:

```
ALL: 0.0.0.0/0.0.0.0
```

lo que significa que todos los servicios, todos los lugares, de modo que cualquier servicio al que no se le permita específicamente acceder, queda bloqueado (recuerda que por defecto es permitir). Quizás también preferirías simplemente denegar el acceso por defecto a, digamos, `telnet`, y dejar el `ftp` abierto al mundo. Habría que hacer lo siguiente:

en `hosts.allow`:

```
in.telnetd: 10.0.0.0/255.255.255.0 # permitir acceso desde la red interna de 10.0.0.*
```

```
in.ftpd: 0.0.0.0/0.0.0.0 # permitir acceso desde cualquier parte del mundo
```

en `hosts.deny`:

```
in.telnetd: 0.0.0.0/0.0.0.0 # denegar acceso a telnetd desde cualquier parte
```

o si quieres estar realmente a salvo:

```
ALL: 0.0.0.0/0.0.0.0 # denegar el acceso a todo desde cualquier parte
```

Lo cual puede afectar a servicios como `ssh` y `nfs`, de modo que ¡ten cuidado!

Quizás simplemente prefieras listar por separado todos los servicios que se están usando:

```
in.telnetd: 0.0.0.0/0.0.0.0
```

```
ipop3d: 0.0.0.0/0.0.0.0
```

Si se deja activado un servicio que no debería figurar en `inetd.conf` y NO se tiene una política de denegación por defecto, se pueden tener problemas. Es más seguro (y lleva un poco más de trabajo, pero a la larga es menor que tener que reinstalar el servidor) tener reglas de denegación por defecto para el cortafuegos y `TCP_WRAPPERS`, de modo que si se olvida algo por accidente, por defecto no tendrá acceso. Si se instala algo para lo cual necesitan tener acceso los usuarios y se olvida poner reglas, enseguida se quejarán y se podrá rectificar el problema rápidamente. Fallar por precaución y denegar accidentalmente algo es más seguro que dejarlo abierto.

Las páginas del manual de `TCP_WRAPPERS` son bastante buenas y están disponibles haciendo:

```
man hosts.allow
```

y/o (son la misma página del manual)

```
man hosts.deny
```

Una pequeña advertencia con TCP_WRAPPERS de aparición reciente en Bugtraq.

TCP_WRAPPERS interpreta la líneas de hosts.allow y hosts.deny de la forma siguiente:

- * se eliminan todos los \`'s` (continuación de línea), completando todas las líneas (también hay que darse cuenta de que la longitud máxima de una línea es de unos 2k, en algunos casos es mejor utilizar múltiples líneas)
- * se eliminan las líneas que empiezan con `#'s`, p. ej. todas las líneas comentadas. De modo que:

```
# esto es una prueba
```

```
# in.ftpd: 1.1.1.1 \
```

```
in.telnetd: 1.1.1.1
```

esto significa que la línea "in.telnetd: 1.1.1.1" también se ignoraría.

```
/etc/services
```

El fichero de servicios es una lista de números de puertos, el protocolo y el nombre correspondiente. El formato es:

```
nombre-de-servicio puerto/protocolo alias
```

```
# comentario opcional
```

por ejemplo:

```
time 37/udp timserver
```

```
rlp 39/udp resource # localización de recursos
```

```
name 42/udp nameserver
```

```
whois 43/tcp nickname # generalmente al sri-nic
```

```
domain 53/tcp
```

```
domain 53/udp
```

Por ejemplo, este fichero se utiliza cuando se ejecuta 'netstat -a', y por supuesto no se utiliza cuando se ejecuta 'netstat -an'

TCP-IP y seguridad de redes

El TPC-IP se creó en una época y en una situación donde la seguridad no era algo que concerniera demasiado. Inicialmente, 'Internet' (entonces llamada Arpanet), consistía en unos pocos hosts, todo eran sitios académicos, grandes empresas o gobiernos. Todo el mundo se conocía, y acceder a Internet era un asunto serio. La suite de protocolos TCP-IP es bastante robusta (todavía no ha fallado estrepitosamente), pero desafortunadamente no está prevista para la seguridad (p. ej., autenticación, verificación, cifrado, etc.). Hacer spoofing de paquetes, interceptar paquetes, leer la carga de los datos, y demás, es algo bastante fácil en el Internet de hoy en día. Los ataques más comunes son los ataques de negación de servicio, ya que son los más fáciles de ejecutar y los más difíciles de impedir, seguidos del sniffing de paquetes, escaneo de puertos y otras actividades relacionadas.

Los nombres de hosts no apuntan siempre a la dirección IP correcta, y las direcciones IP's no siempre se pueden resolver al nombre de host adecuado. Si es posible, no utilices autenticación basada en nombres de hosts. Puesto que el envenenamiento de cachés DNS es algo relativamente sencillo, confiar la autenticación en una dirección IP reduce el problema al spoofing, lo cual es algo más seguro, pero de ningún modo completamente seguro. No existen mecanismos extendidos para verificar quién envió los datos y quién los está recibiendo, excepto mediante el uso de sesiones o cifrado a nivel IP (sin embargo están empezando a cobrar auge el IPSec/IPv6 y otras tecnologías VPN).

Se puede empezar por denegar los datos entrantes que dicen originarse desde tu red(es), puesto que estos datos son evidentemente falsos. Y para evitar que tus usuarios, u otros que hayan irrumpido en tu red puedan lanzar ataques falsificados, se deberían bloquear todos los datos salientes que no provengan de tu dirección IP. Es algo relativamente fácil y sencillo de gestionar, pero la inmensa mayoría de redes no lo hacen (me pasé casi un año dando la coña a mi PSI antes de que empezaran a hacerlo). Si todo en Internet tuviera filtros salientes (es decir, restringir todo el tráfico saliente a aquel que se originase desde las direcciones IP internas), los ataques mediante spoofing serían imposibles, y a la vez sería mucho más fácil tracear el origen de los atacantes. También se deberían bloquear las redes reservadas (127.*, 10.*, etc.). Me he dado cuenta de que muchos ataques provenientes de Internet lo hacen con IPs etiquetadas con esos rangos de direcciones. Si se utiliza traducción de direcciones de red (como IPMASQ) y no se tiene correctamente instalado el cortafuegos, se puede ser atacado con facilidad, o ser utilizado para lanzar un ataque a terceros.

Si tienes que comunicarte de forma segura con otra gente, deberías considerar el uso de la tecnología VPN. La única tecnología disponible que goza de una amplia aceptación y está destinada a convertirse en standard (en IPv6) es IPSec, un standard abierto soportado por muchos vendedores y la mayoría de los vendedores tienen implementaciones funcionales nativas para sus SO (aunque algunas están limitadas para cumplir con las leyes de exportación de los EE.UU.). Por favor, lee el Apéndice B o la sección Cifrado de Servicios y Datos para más detalles.

IPSec

El IPSec tiene su propia sección. Creo que es el futuro de la tecnología VPN (es el standard más comúnmente aceptado en la actualidad, y una parte integral de IPv6).

IPv6

IPv6 no proporciona seguridad per se, pero tiene conectores integrados para futuras mejoras de seguridad, soporte para IPSec, etc. Si se utiliza en una red, por supuesto que haría más difícil la vida de los atacantes, puesto que el uso de IPv6 todavía no está muy extendido. Si quieres aprender más, visita <http://www.bieringer.de/linux/IPv6/>. Actualmente, Linux soporta IPv6 casi completamente (uno de los pocos SO's que lo hacen).

Programas de ataque TCP-IP

Existen una variedad de programas para causar desorganización en TCP-IP (la mayoría son ataques de Negación de Servicio) sin embargo sólo unos pocos pueden ser útiles para administradores.

Proyecto HUNT

El Proyecto HUNT es un conjunto de herramientas para manipular conexiones TCP-IP (generalmente en una LAN Ethernet), como cerrar conexiones, espiarlas y otro tipo de cosas "desagradables". Incluye también una variedad de ataques basados en ARP y otro tipo de maléficas fuentes de diversión. Se puede conseguir en: <http://www.cri.cz/kra/index.html>

Seguridad PPP

PPP permite conexiones TCP-IP, IPX/SPX y NetBEUI sobre líneas serie (las cuales pueden estar conectadas a módems, por supuesto). Este es el método principal que utiliza la gente para conectarse a Internet (prácticamente todas las cuentas de dial-up son PPP). La esencia de una conexión PPP consiste en dos dispositivos informáticos (un ordenador, un Palm Pilot, un servidor de terminales, etc.) conectados sobre enlaces de serie (generalmente vía módems). Ambos extremos llaman al PPP, se negocia la autenticación (mediante uno de entre varios métodos), y se establece el enlace. PPP no tiene soporte real para cifrado, de modo que si se necesita un enlace seguro hay que invertir en algún tipo de software VPN.

La mayoría de los sistemas llaman a PPP de una forma bastante cutre, se hace un login al equipo (servidor de terminales, etc.) y luego se invoca al login shell del PPP. Por supuesto que esto significa que el nombre de usuario y contraseña se envían en texto claro sobre la línea, y que hay que tener una cuenta en ese tipo de equipo. En este caso el PPP no negocia la autenticación en absoluto. Un método algo más seguro de gestionarlo es utilizar PAP (Password Authentication Protocol, Protocolo de Autenticación de Contraseñas). Mediante PAP, la autenticación se hace internamente mediante PPP, de modo que no se requiere una cuenta "real" en el servidor. Sin embargo el nombre de usuario y la contraseña se siguen enviando en texto claro, pero al menos el sistema es algo más seguro dada la inexistencia de cuentas de usuario "reales".

El tercer (y mejor) método para la autenticación es utilizar CHAP (Challenge Handshake Authentication Protocol, Protocolo de Autenticación Desafío-Respuesta). Ambas partes se intercambian llaves públicas y las utilizan para cifrar los datos que se envían durante la secuencia de autenticación. De modo que el nombre de usuario y la contraseña están relativamente a salvo de fisgones, y sin embargo las transmisiones de datos se hacen con normalidad. Una advertencia con CHAP: La implementación de Microsoft utiliza DES en lugar de MD5, lo cual lo hace fallar si se conecta con un cliente Linux. Sin embargo existen parches para arreglarlo. PPP viene con cada distribución de Linux como parte del núcleo del SO, el Linux PPP-HOWTO se encuentra disponible en <http://www.interweft.com.au/other/ppp-howto/ppp-howto.html>

Seguridad IP (IPSec)

Seguridad IP (IPSec) es el cifrado del tráfico de red. No se puede cifrar la información de la cabecera ni el trailer (p. ej. la dirección IP y puerto de donde viene el paquete y su destino, los checksums de CRC, etc.), pero se puede cifrar la carga útil. Esto permite asegurar protocolos como POP/WWW sin tener que cambiarlos de ninguna forma, puesto que el cifrado se hace en el nivel IP. También permite conectar de forma segura LANs y clientes entre sí, sobre redes inseguras (como Internet). En la actualidad, IPSec para Linux está en fase de pruebas, sin embargo ya se han lanzado varias versiones estables, y yo mismo he desarrollado servidores seguros basados en IPSec. IPSec es un standard, y parte el protocolo IPv6, de modo que ya se puede comprar software IPSec para Windows 95/98/NT, Solaris y otros Unix, que interoperarán con Linux IPSec.

Soporte IPSec del kernel

Para utilizar IPSec es necesario tener soporte IPSec en el kernel. Desafortunadamente, ninguna distribución Americana de Linux puede exportar criptografía robusta fuera de Norte América, de modo que en general, suelen escoger no incluirla en absoluto, de las distribuciones extranjeras de Linux, en la actualidad ninguna viene con soporte IPSec dentro del kernel. Es necesario conseguir el código fuente del kernel (recomiendo la 2.2.10, la más reciente a la hora de escribir esto), y el código fuente del Linux IPSec, disponible en: <http://www.xs4all.nl/~freeswan/>

Instala el fuente del kernel (generalmente en /usr/src/linux) y compila el nuevo kernel, instálalo, arráncalo y pruébalo. Asegúrate de que tus redes funcionan correctamente, si no funcionan, hacer que lo haga IPSec será imposible. Ahora hay que descargar la última instantánea de IPSec (la versión 1.0 NO funcionará con los kernels 2.2.x). Después ir a /usr/local/src (o dondequiera que hayas puesto el código fuente de tus programas), desempaquetar el fuente y ejecutar el programa de instalación (make menugo suele ser lo habitual para la configuración basada en ncurses). Lo cual parcheará los ficheros del kernel, luego ejecuta la configuración del kernel y después construye las herramientas IPSec y el kernel.

```
cd /usr/local/src/
```

```
tar -zvxvf /path/del/tarball/snapshot.tar.gz
```

```
chown -R root:root freeswan-snap1999Jun14b
```

```
cd freeswan-snap1999Jun14b
```

```
make menugo
```

asegúrate de guardar la configuración del kernel, incluso aunque se hayan elegido las opciones, no han sido guardadas. También tendrás que reconstruir el kernel, puesto que el comando "make menugo" ejecuta un "make zImage", lo cual suele fallar, debido a los grandes tamaños del kernel de la 2.2.x. Una vez que se ha hecho la compilación, debería dar uno o dos mensajes de error, simplemente haz:

```
cd /usr/src/linux
```

```
make bzImage
```

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-2.2.10-ipsec
```

Ahora hay que editar lilo.conf, ejecutar lilo de nuevo y reiniciar para hacer uso del nuevo kernel.

lilo.conf debería tener el siguiente aspecto:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=100
image=/boot/vmlinuz-2.2.10-ipsec
    label=linux-ipsec
    root=/dev/hda1
    read-only
image=/boot/vmlinuz-2.2.10
    label=linux
    root=/dev/hda1
    read-only
```

vuelve a ejecutar lilo y verás:

```
linux-ipsec *
linux
```

después reinicia y deberías estar ejecutando el kernel 2.2.10 con soporte IPsec. A medida que la máquina se reinicia y empieza el IPsec se verán varios errores, por defecto IPsec está configurado para utilizar el interfaz eth999, el cual por supuesto no existe. Deberías añadir /usr/local/lib/ipsec en la frase del path o si no tendrás que escribir el path completo un montón.

Configuración de redes IPsec

Tendrás que habilitar el TCP-IP forwarding en el servidor de enlace, en Red Hat Linux se hace cambiando la línea de /etc/sysconfig/network:

```
FORWARD_IPV4="false"
```

por:

```
FORWARD_IPV4="yes"
```

o se puede habilitar vía sistema de ficheros en /proc:

```
cat 1 > /proc/sys/net/ipv4/ip_forward
```

Puesto que la mayoría de la gente tiene por defecto políticas de denegación de paquetes de forwarding, tendrás que permitir que los paquetes atraviesen la red remota / máquina de tu red / máquina y vice versa. Además de esto,

cualquier regla de enmascaramiento para redes internas que también estén usando IPsec debe venir después de las reglas que permiten el tráfico IPsec, o la máquina intentará enmascarar los paquetes, en lugar de pasarlos al IPsec.

El siguiente ejemplo es para dos redes protegidas (usando direcciones IP no rutables, escondidas tras máquinas Linux haciendo enmascaramiento de IPs) conectadas vía Internet:

```
10.0.0.2 192.168.0.2
```

```
10.0.0.1 192.168.0.1
```

```
1.2.3.4 5.6.7.8
```

```
1.2.3.1 5.6.7.1
```

```
2.3.4.5 6.7.8.9
```

```
INTERNET
```

Conexión manual de llaves

Primero configuraremos un enlace utilizando la conexión manual de llaves (por simplicidad), hay que editar `ipsec.conf`, y las reglas del cortafuegos. La mayoría de las opciones por defecto del fichero `ipsec.conf` son correctas, pero hay que cambiar lo siguiente:

```
conn ejemplo
```

```
type=tunnel
```

```
left=
```

```
leftnexthop=
```

```
leftsubnet=
```

```
right=
```

```
rightnexthop=
```

```
rightsubnet=
```

```
spibase=0x200
```

```
esp=3des-md5-96
```

```
espenckey=
```

```
espauthkey=
```

reemplaza la `espenckey` y la `espauthkey` con las nuevas llaves (utilizando `ranbits` para generar un número, recuerda dejar el `0x` por delante, que especifica que es un número hexadecimal) de modo que es algo así:

```
conn mi-tunnel
```

```
type=tunnel
```

```
left=1.2.3.4
```

```
leftnexthop=1.2.3.1
leftsubnet=10.0.0.0/24
right=5.6.7.8
rightnexthop=5.6.7.1
rightsubnet=192.168.0.0/24
spibase=0x200
esp=3des-md5-96
espenckey=cualquier_llave_de_autenticación (ranbits 192)
espauthkey=cualquier_otra_llave (ranbits 128)
```

Una vez que has acabado, copia los ficheros ipsec.conf e ipsec.secrets desde la máquina en que los editaste hasta el otro servidor, de forma segura. Ahora, lo único que queda es añadir reglas al cortafuegos para que no se enmascaren los paquetes (en lugar de eso lo que queremos es redirigirlos, hacer un forward).

En el servidor 1.2.3.4 se deberían añadir las siguientes reglas:

```
ipchains -A forward -p all -j ACCEPT -s 10.0.0.0/24 -d 192.168.0.0./24
```

```
ipchains -A forward -p all -j ACCEPT -s 192.168.0.0/24 -d 10.0.0.0/24
```

asegúrate de que estas reglas aparecen antes de la regla de enmascaramiento, algo así:

```
#
```

```
# FORWARD RULES
```

```
#
```

```
ipchains -P forward DENY
```

```
#
```

```
ipchains -A forward -p all -j ACCEPT -s 10.0.0.0/24 -d 192.168.0.0/24
```

```
ipchains -A forward -p all -j ACCEPT -s 192.168.0.0/24 -d 10.0.0.0/24
```

```
ipchains -A forward -p all -j MASQ -s 10.0.0.0/24 -d 0.0.0.0/0
```

Y en el servidor 5.6.7.8 se repetiría el proceso:

```
ipchains -A forward -p all -j ACCEPT -s 192.168.0.0/24 -d 10.0.0.0/24
```

```
ipchains -A forward -p all -j ACCEPT -s 10.0.0.0/24 -d 192.168.0.0/24
```

asegúrate de que estas reglas aparecen antes de la regla de enmascaramiento, algo así:

```
#
```

```
# FORWARD RULES
```

```
#
ipchains -P forward DENY
#
ipchains -A forward -p all -j ACCEPT -s 192.168.0.0/24 -d 10.0.0.0/24
ipchains -A forward -p all -j ACCEPT -s 10.0.0.0/24 -d 192.168.0.0/24
ipchains -A forward -p all -j MASQ -s 192.168.0.0/24 -d 0.0.0.0/0
```

Ahora deberías ser capaz de establecer el túnel ipsec manualmente en ambas máquinas, y las máquinas de la Red A deberían ser capaces de hablar con las máquinas de la red B sin problemas.

```
ipsec manual -up mi-tunnel
```

lo cual mostraría una salida similar a:

```
/usr/local/lib/ipsec/spi: message size is 36
/usr/local/lib/ipsec/spi: message size is 132
/usr/local/lib/ipsec/spi: message size is 132
```

Para probarlo, haz un ping a 192.168.0.2 desde el cliente 10.0.0.2. Si funciona, lo has configurado correctamente. Si no funciona, verifica tu red para asegurarte de que 1.2.3.4 puede ver 5.6.7.8 y que está habilitado el TCP-IP forwarding, y asegúrate de que no hay reglas del cortafuegos que estén bloqueando paquetes o intentando enmascararlos. Una vez que has establecido y probado la conexión con éxito, deberías pasar a la conexión automática de llaves (especialmente en entornos de producción).

Conexión automática de llaves

Si se intenta usar IPSec en un entorno de producción, la conexión manual de llaves es una mala idea, en términos generales. Con la conexión automática se tiene un secreto compartido de 256bit que se copia a ambos lados del túnel, el cual se utiliza durante el intercambio de llaves para asegurarse que no se dan ataques del tipo "man in the middle, hombre de por medio". Con la conexión automática, la vida media de una llave es de 8 horas, lo cual se puede ajustar a cualquier intervalo, y si alguien se las arregla para atacar por fuerza bruta la llave, sólo será válida durante ese período de tráfico de 8 horas. El ejemplo siguiente se construye sobre el anterior:

ipsec.secrets contiene el secreto compartido. Este fichero debe ser puesto a salvo a toda costa. Para una conexión entre los servidores 1.2.3.4 y 5.6.7.8 se necesitaría una línea como:

```
1.2.3.4 5.6.7.8
```

```
"0xa3afb7e6_20f10d66_03760ef1_9019c643_a73c7ce0_91e46e84_ef6281b9_812392bf"
```

Esta línea necesita estar en ambos ficheros ipsec.secrets. Después se necesitaría editar la configuración del túnel en ipsec.conf por la siguiente:

```
conn mi-tunnel
```

```
type=tunnel
```

```
left=1.2.3.4
leftnexthop=1.2.3.1
leftsubnet=10.0.0.0/24
right=5.6.7.8
rightnexthop=5.6.7.1
rightsubnet=192.168.0.0/24
keyexchange=ike
keylife=8h
keyingtries=0
```

Entonces se arrancaría el demonio pluto, intenta conectar al demonio Pluto desde el otro extremo del túnel, y establece una conexión. Una advertencia, Pluto se ejecuta en el puerto 500, udp, de modo que lo más probable es que tengas que abrir un hueco en el cortafuegos para permitirle pasar:

```
ipchains -A input -p udp -j ACCEPT -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 500
```

Encuentro conveniente el uso de la clave "%search" en lugar de listar el túnel a arrancarse, añadiendo:

```
auto=start
```

para cada configuración del túnel y editar el ipsec.secrets:

```
plutoload=%search
plutostart=%search
```

Lo cual a la larga te hará la vida más fácil. Si todo va bien, deberías ver algo parecido a esto en /var/log/messages:

```
+-----+
| Jun 26 02:10:41 server ipsec_setup: Starting FreeS/WAN IPSEC...
| Jun 26 02:10:41 server ipsec_setup: /usr/local/lib/ipsec/spi: message size
| is 28
| Jun 26 02:10:41 server ipsec_setup: KLIPS debug 'none'
| Jun 26 02:10:41 server ipsec_setup: KLIPS ipsec0 on eth0
| 1.2.3.4/255.255.255.0 broadcast
| 24.108.11.255
| Jun 26 02:10:42 server ipsec_setup: Disabling core dumps:
| Jun 26 02:10:42 server ipsec_setup: Starting Pluto (debug 'none'):
| Jun 26 02:10:43 server ipsec_setup: Loading Pluto database 'mi-tunnel':
| Jun 26 02:10:44 server ipsec_setup: Enabling Pluto negotiation:
| Jun 26 02:10:44 server ipsec_setup: Routing for Pluto conns 'mi-tunnel':
+-----+
```

```
Jun 26 02:10:45 server ipsec_setup: Initiating Pluto tunnel 'mi-tunnel':
Jun 26 02:10:45 server ipsec_setup: 102 "mi-tunnel" #1: STATE_MAIN_I1:
initiate
Jun 26 02:10:45 server ipsec_setup: 104 "mi-tunnel" #1: STATE_MAIN_I2: from
STATE_MAIN_I1; sent MI2, expecting MR2
Jun 26 02:10:45 server ipsec_setup: 106 "mi-tunnel" #1: STATE_MAIN_I3: from
STATE_MAIN_I2;sent MI3, expecting MR3
Jun 26 02:10:45 server ipsec_setup: 003 "mi-tunnel" #1: STATE_MAIN_I4: SA
established
Jun 26 02:10:45 server ipsec_setup: 110 "mi-tunnel" #2: STATE_QUICK_I1:
initiate
Jun 26 02:10:45 server ipsec_setup: 003 "mi-tunnel" #2: STATE_QUICK_I2: SA
established
Jun 26 02:10:46 server ipsec_setup: ...FreeS/WAN IPSEC started
```

Y en el fichero /var/log/secure se debería ver algo parecido a esto:

```
Jun 26 02:10:42 server Pluto[25157]: Starting Pluto (FreeS/WAN Version
snap1999Jun14b
Jun 26 02:10:42 server Pluto[25157]: added connection description "mi-tunnel"
Jun 26 02:10:42 server Pluto[25157]: listening for IKE messages
Jun 26 02:10:42 server Pluto[25157]: adding interface ipsec0/eth0 1.2.3.4
Jun 26 02:10:42 server Pluto[25157]: loading secrets from
"/etc/ipsec.secrets"
Jun 26 02:10:42 server Pluto[25157]: "mi-tunnel" #1: initiating Main Mode
Jun 26 02:10:42 server Pluto[25157]: "mi-tunnel" #1: ISAKMP SA established
Jun 26 02:10:42 server Pluto[25157]: "seifried-mosqueado" #2: initiating
Quick Mode POLICY_ENCRYPT+POLICY_TUNNEL+POLICY_PFS
Jun 26 02:10:42 server Pluto[25157]: "mi-tunnel" #2: sent QI2, IPsec SA
established
Jun 26 02:10:42 server Pluto[25157]: "mi-tunnel" #3: responding to Main Mode
Jun 26 02:10:42 server Pluto[25157]: "mi-tunnel" #3: sent MR3, ISAKMP SA
established
Jun 26 02:10:42 server Pluto[25157]: "mi-tunnel" #4: responding to Quick Mode
Jun 26 02:10:42 server Pluto[25157]: "mi-tunnel" #4: IPSec SA established
Jun 26 02:10:42 server Pluto[25157]: "mi-tunnel" #5: responding to Main Mode
Jun 26 02:10:42 server Pluto[25157]: "mi-tunnel" #5: sent MR3, ISAKMP SA
established
```


Cifrado de servicios / datos

Cifrado de servicios de red

Prácticamente todo el tráfico de red viaja sin cifrar y puede ser leído con facilidad por un atacante. Si alguien revienta una máquina del lado de Internet e instala un sniffer de contraseñas (en realidad no es más que el sniffer de paquetes básico con un filtro), la red entera puede verse comprometida en cuestión de horas. Un PSI que permanecerá en el anonimato, colocó máquinas de clientes en la misma LAN, utilizando un hub ethernet normal y corriente, lo cual quiere decir que todas las máquinas podían ver el tráfico entre sí (usuarios recogiendo su correo vía pop, sesiones de telnet, etc.). Esta es una de las razones por las cuales es una buena idea el cifrado del tráfico de datos.

Existen o se están desarrollando varios mecanismos para cifrar el tráfico de red, en diferentes niveles de la pila de red. Algunos esquemas sólo cifran los datos que se envían (como el correo cifrado con PGP), algunos cifran la sesión (SSL), y algunos cifran la carga de datos de los paquetes (IPSec y otros VPN's). A la larga, la mejor solución será el IPSec (en mi opinión), puesto que no requiere modificar las aplicaciones, y proporciona un nivel de seguridad entre ordenadores bastante elevado. Actualmente no existen soluciones de cifrado ampliamente utilizadas, en parte porque Microsoft no soporta muchas, lo cual es un serio obstáculo para cualquier solución masiva. Para ser justos, Microsoft tiene soporte para IPSec en fase beta, pero todavía no está preparado, y quedará restringido al mercado Norte Americano debido a la ley de los EE.UU. En la actualidad, el mejor esquema disponible es el SSL, Secure Sockets Layer, propuesto originalmente por Netscape. El SSL cifra los datos a nivel de sesión, de modo que si tu aplicación soporta SSL y el servidor soporta SSL, se está de suerte. Hoy en día, la mayoría de visores www, algunos lectores de news/correo, y unos pocos clientes de ftp y de telnet soportan SSL. En cuanto a servidores Linux, la mayoría de los servicios se pueden "SSLificar". Sin embargo el SSL necesita clientes con capacidad SSL, algo que no se podrá conseguir que soporte la mayoría de la gente. Esto significa que por lo general, los servicios "SSLificados" están restringidos dentro de una organización. Las librerías SSL se encuentran disponibles en <http://www.openssl.org/>.

SSL

HTTP - SSL

El servidor www más habitual, Apache, tiene bastante buen soporte SSL, el cual se puede descargar gratuitamente fuera de los EE.UU. (las patentes de los EE.UU. sobre RSA/etc significan tener que pagar derechos dentro de los EE.UU, de modo que el software gratuito es ilegal) desde <http://www.Apache-ssl.org/>. Hay muchos servidores comerciales de www que soportan SSL, la mayoría de los cuales están basados en Apache, como el Servidor Seguro de Red Hat, Stronghold, etc.

Telnet - SSL

Para reemplazar al telnet, SSLtelnet y MZtelnet proporcionan un nivel de seguridad mucho más alto que el telnet original, aunque el SSLtelnet y el MZtelnet no son tan flexibles como el SSH, son perfectamente libres (es decir, con licencia GNU) lo cual el SSL no lo es. Los paquetes cliente y servidor se encuentran disponibles como tarballs en:

<ftp://ftp.uni-mainz.de/pub/internet/security/ssl/>, y como paquetes RPM en: <ftp://ftp.replay.com/pub/replay/linux/Red Hat/>.

FTP - SSL

También existe un reemplazo para nuestro ftpd favorito (probablemente el WU-FTPD), también disponible como un conjunto de parches al WU-FTPD. Es altamente apropiado, pues la mayoría de los servidores tienen muchos usuarios que necesitan acceso ftp. El tarball se encuentra en:
<ftp://ftp.uni-mainz.de/pub/internet/security/ssl/>, y los paquetes RPM están en:
<ftp://ftp.replay.com/pub/replay/linux/Red Hat/>.

Soluciones para Redes Privadas Virtuales

IPSec

El IPSec tiene su propia sección. Creo que es el futuro de la tecnología VPN (hoy en día es el standard más soportado, y una parte integral del IPv6).

PPTP (Point to Point Tunneling Protocol)

El PPTP es un protocolo propietario creado por Microsoft para soluciones VPN. Hasta la fecha se ha demostrado que contiene numerosos fallos serios. Sin embargo, si se necesita integrar Linux en un entorno PPTP, no está todo perdido, en <http://www.moretonbay.com/vpn/pptp.html> se encontrará una implementación de PPTP para Linux.

CIPE (Crypto IP Encapsulation)

CIPE es un esquema gratuito de cifrado a nivel IP, orientado al uso entre routers. Es adecuado para hacer "bridging" de redes con seguridad sobre redes inseguras (como Internet). El sitio oficial está en:
<http://sites.inka.de/~W1011/devel/cipe.html>. Sin embargo recomendaría el FreeS/WAN como una mejor solución a largo plazo.

ECLiPt Secure Tunnel (actualmente en fase beta)

Otra solución con licencia GNU para VPN's bajo Linux. Actualmente en fase beta (y no está recomendado para uso masivo) aunque he pensado que lo mencionaría, puesto que parece ser un esfuerzo serio. La página oficial se encuentra en:
<http://eclipt.uni-klu.ac.at/projects/est/>. De nuevo, recomendaría FreeS/WAN como una mejor solución a largo plazo.

Stunnel

Stunnel es una solución basada en SSL para asegurar servicios de red. Tiene una porción servidor que se ejecuta en el servidor UNIX, y una parte cliente que se ejecuta en UNIX o en Windows. <http://mike.daewoo.com.pl/computer/stunnel/>

Cifrado de datos

También se encuentran disponibles diferentes programas para cifrar tus datos, algunos a nivel de fichero (PGP, GnuPG, etc.) y algunos a nivel de unidad de disco (Cryptographic FileSystem, Sistema de Ficheros Criptográfico, por ejemplo). Estos sistemas son muy adecuados para el almacenamiento de datos seguros, y en cierta medida para la transmisión segura de datos. Sin embargo se necesita el software adecuado en ambos extremos, versiones compatibles, y de alguna forma tendrá que llevarse a cabo un intercambio de llaves públicas, lo cual, por desgracia es una onerosa tarea para la mayoría de las personas. Además de esto, no existe una forma sencilla de confiar en la llave pública de una persona a menos que se reciba directamente de esa persona (como por ejemplo en una fiesta de firmado de llaves), o a menos que esté firmada por alguien en quien se confía (¿pero cómo se consigue de forma segura la llave del firmante?). Sistemas para cifrar unidades, como el CFS (Cryptographic

FileSystem, Sistema de Ficheros Criptográfico) suelen ser fácil de implementar, y sólo necesitan que el usuario proporcione una contraseña o llave, o alguna forma de acceder a sus ficheros.

PGP (Pretty Good Privacy, Privacidad Bastante Buena)

El abuelito del cifrado público, es con mucho uno de los programas más populares, soportado por Unix, Windows y Macintosh. Por desgracia ahora se ha comercializado, lo cual ha dado como resultado una pérdida de calidad para los usuarios. Personalmente, creo que cualquier software utilizado para cifrar o asegurar datos de cualquier otra forma, DEBE ser código libre, o si no cómo se va a asegurar que es seguro. El PGP ahora lo vende Network Associates y no puedo recomendarlo de buena fe como un mecanismo de seguridad para el almacenamiento y transmisión de ficheros. El PGP se encuentra disponible para descargar de: <ftp://ftp.replay.com/>

GnuPG (Gnu Privacy Guard, Guardián de Privacidad Gnu)

La alternativa al PGP, GnuPG (GPG) es un reemplazo directo completamente de código abierto y con licencia GNU (como si no se adivinase por el nombre). Esta herramienta está disponible en: <http://www.gnupg.org/>, como código fuente o en binarios precompilados para windows y RPM's.

pgp4pine

El pgp4pine es un shell PGP para pine, que facilita el uso de PGP/GnuPG con el pine. La firma / cifrado, etc. también se han hecho más sencillas. Se puede conseguir en: <http://members.home.com/cdwiegand/pgp4pine/>

HardEncrypt

HardEncrypt es un generador pad de un sólo uso y un conjunto de herramientas para utilizarlo. En teoría, los pads de un sólo uso son una forma de cifrado casi irrompible. Utilizando un conjunto de datos aleatorios, criptográficamente seguros se pueden exprimir por completo los datos privados, para descifrarlos se necesita el pad de un sólo uso. Esta forma de cifrado es ideal para la comunicación de datos sensibles, salvo una desventaja, que primero hay que transferir el pad de un sólo uso a la otra parte. Es posible descargar el HardEncrypt desde la siguiente dirección:

<http://www.csuglab.cornell.edu/Info/People/jcr13/HardenedCriminal/main.html>

Secret-share

El secret-share te permite dividir un fichero en tantos trozos como se deseen, todos los cuales se necesitan para reconstruir satisfactoriamente el fichero. Todos los trozos menos uno son datos aleatorios cifrados, lo cual lo enmaraña de alguna forma. Se puede descargar de:

<http://www.mindrot.org/code/secret-share.php3>

Cifrado del disco duro

CFS (Sistema de Ficheros Criptográfico)

El CFS te permite guardar los datos en el disco duro en formato cifrado, y es significativamente más sencillo de utilizar que un programa de cifrado de ficheros (como PGP) si lo que se quiere es almacenar muchos ficheros y directorios de los que se quiere tener alejados a los curiosos. El sitio oficial de distribución está en: <http://www.cryptography.org/>, y también se encuentran disponibles RPM's en: [ftp://ftp.replay.com/pub/replay/linux/Red Hat/](ftp://ftp.replay.com/pub/replay/linux/RedHat/), y los ficheros binarios Debian están en: <http://www.debian.org/Packages/stable/otherosfs/cfs.html>.

TCFS

El TCFS es una utilidad de cifrado de datos a nivel del kernel, similar al CFS. Sin embargo tiene algunas ventajas sobre el CFS; puesto que está implementado en el kernel, es significativamente más rápido. Está fuertemente arraigado con el NFS, lo cual quiere decir que se pueden servir datos de forma segura en una máquina local, o a través de la red. Descifra datos en la máquina cliente, de modo que cuando se utiliza sobre la red, nunca se transmite la contraseña. La única pega es que todavía no se ha adaptado a la serie 2.2 del kernel. El TCFS se puede conseguir en: <http://tcfs.dia.unisa.it/>

PPDD

El PPDD permite crear una partición de disco cifrada, puede ser una partición real o un dispositivo loopback (el cual reside en un fichero, pero se monta como un sistema de ficheros). Utiliza el algoritmo blowfish, el cual es relativamente rápido y está probado. El PPDD se puede conseguir desde: <http://linux01.gwdg.de/~alatham/>

Directorio Raíz Cifrado

El Encrypted Home Directory, Directorio Raíz Cifrado, funciona de forma parecida al CFS, sin embargo está orientado a proporcionar un sólo directorio cifrado. Resumiendo, crear un fichero de tamaño X en /crypt/ con tu UID, y lo monta en un dispositivo loopback de forma que se pueda acceder a él. El truco está en que los datos se cifran y descifran al vuelo, cuando se accede a él (como con el CFS). La única pega es que el software todavía está en desarrollo, así que haz copia de seguridad de cualquier dato importante. Se puede descargar desde: <http://members.home.net/id-est/>

StegFS

Steganographic File System, en realidad oculta los datos en tu disco duro, haciendo difícil incluso probar que existan. Puede ser muy útil, puesto que el atacante primero tiene que encontrar los datos, y ya no digamos romper el fuerte cifrado utilizado para protegerlos. Se puede conseguir de: <http://ban.joh.cam.ac.uk/~adm36/StegFS/>.

BestCrypt

BestCrypt es un producto comercial, con código fuente, disponible para Windows y Linux. Se puede conseguir aquí: <http://www.jetico.com/>

Fuentes de datos aleatorios

Para que el cifrado sea efectivo, especialmente a gran escala como con IPSec a lo largo de muchos hosts, se necesitan buenas fuentes de datos aleatorios y criptográficamente seguros. En Linux, están el /dev/random y el /dev/urandom, que son buenos pero no siempre fenomenales. Parte de la ecuación está en medir sucesos "aleatorios", manipulando esos datos y después poniéndolo disponible (vía (u)random). Estos sucesos aleatorios incluyen: entrada de teclado y ratón, interrupciones, lecturas de disco, etc.

Sin embargo, a medida que los discos duros que van saliendo tienen más y más caché (los IBM Deskstars vienen con 4 Megabytes de caché en disco) y muchos servidores no tienen teclado/ratón, puede ser más difícil encontrar fuentes de datos aleatorios. Algunas fuentes, como la actividad de red, no son del todo apropiadas, puesto que los ataques también se pueden medir (por supuesto que sería un ataque muy exótico, pero lo suficiente como para preocupar a la gente). Existen varias fuentes de datos aleatorios que se pueden utilizar (o al

menos que parecen ser aleatorios), los decaimientos radiactivos y las manipulaciones de radiofrecuencias son dos bastante famosos. Por desgracia, la idea de colocar un dispositivo radiactivo al lado de un ordenador suele poner nerviosa a la mayoría de la gente. Y utilizar frecuencias de radio manipuladas está sujeto a error, y la posibilidad de manipulación externa. Para la mayoría de nosotros, no es algo a tener demasiado en cuenta, sin embargo para pasarelas IPSec que manejan muchas conexiones, puede presentar un problema. Una solución potencial es el PIII, que viene con un generador aleatorio de números, que mide la variación térmica de la CPU, creo que a medida que progreseemos serán más comunes soluciones como esta.

Ocultación de datos

Un aspecto que olvida mucha gente es que el simple hecho de cifrar los datos puede llamar la atención. Por ejemplo, si un administrador corporativo buscase entre las estaciones de trabajo por ficheros terminados en .pgp, y fueses el único con semejante tipo de ficheros..

StegHide

El StegHide oculta los datos en ficheros, tales como sonidos e imágenes, en los cuales no se utilizan todos los bits de cada byte. Puesto que los datos van cifrados, pareceran aleatorios, y probar que los datos están ahí puede ser difícil. La única desventaja es que para almacenar un fichero de un megabyte se necesita una imagen o sonido de varios megabytes, lo cual puede resultar extraño (pero los discos duros y las altas velocidades de acceso se están poniendo baratas). El StegHide se puede conseguir en:

<http://www2.crosswinds.net/~shetzl/steghide/>

Virtual Private Server

El Virtual Private Server (VPS) utiliza Perl y SSH para crear VPN's. Se puede conseguir en: <http://www.strongcrypto.com/>.

Virtual Tunnel

El Virtual Tunnel (VTUN) soporta una variedad de métodos de establecimiento de un enlace, y varios algoritmos. Se puede conseguir de:

<http://vtun.netpedia.net/>.

Zebedee

Zebedee proporciona cifrado del tráfico TCP entre hosts y se encuentra disponible para UNIX y windows. Se puede conseguir de:

<http://www.winton.org.uk/zebedee/>.

Rutado

routed

routed es uno de los paquetes de rutado standard disponibles para Linux. Soporta RIP (uno de los protocolos de rutado más viejos todavía en servicio), y ya está. El RIP es muy simple, los routers simplemente hacen un broadcast de sus tablas de rutado a los routers vecinos, lo cual da como resultado (en teoría) una tabla completa de rutado que tiene entradas para cada destino en Internet. Este método es fundamentalmente inseguro, y muy ineficiente más allá de pequeñas redes seguras (en cuyo caso probablemente no sea necesario). Asegurarlo no es posible, se puede configurar un cortafuegos en los puertos 520 y 521, que son los que utiliza RIP para transferir, sin embargo puede dar como resultado rutas a través de las cuales preferirías no atravesar, y los atacantes pueden seguir falsificando las rutas. Ejecutar este servicio es una idea muy mala.

gated

gated es un software de rutado más avanzado que routed. Soporta versiones de RIP 1 y 2, DCN HELLO, OSPF versión 2, EGP versión 2, BGP versiones 2 a 4. Actualmente el protocolo de rutado más popular parece ser el BGP (Border Gateway Protocol), ganando en popularidad el OSPF (OSPF tiene seguridad incorporada, es muy eficiente y bastante más complicado).

zebra

zebra tiene bastantes más características que gated, y ostenta una bonita línea de interfaz de comandos al estilo de Cisco. Se ejecuta como un demonio, multi-hilo para el rendimiento, cada protocolo (RIP, OSPF, etc.) tiene su propia configuración, y se pueden ejecutar múltiples protocolos a la vez (aunque podría originar confusiones/problemas). Existe un puerto de configuración maestro, y un puerto para cada protocolo:

```
zebrasrv 2600/tcp #zebra service
zebra    2601/tcp  #zebra vty
ripd     2602/tcp  #RIPd vty
ripngd   2603/tcp  #RIPngd vty
ospfd    2604/tcp  #OSPFd vty
bgpd     2605/tcp  #BGPd vty
ospf6d   2606/tcp  #OSPF6d vty
```

Recomendaría filtrar estos puertos con el cortafuegos. El acceso se controla vía contraseña de login, y el acceso a las funciones de comandos solicita otra contraseña (utilizando la misma sintáxis que en Cisco, "enable"). Se puede descargar de: <http://www.zebra.org>

Software de Proxy

Existe una variedad de paquetes de software de proxy para Linux. Algunos son a nivel de aplicación (como SQUID) y otros son a nivel de sesión (como SOCKS)

SQUID

SQUID es un proxy a nivel de aplicación para HTTP, HTTPS y FTP. También puede ejecutar peticiones DNS bastante más rápido de lo que puede hacerlo la mayoría del software cliente. SQUID es ideal para acelerar el acceso a www, y para controlar el acceso a sitios web (utilizando paquetes como squidGuard)

Cortafuegos

Un cortafuegos consiste en filtrar el tráfico TCP-IP, generalmente en el punto donde la red se conecta a otra (p. ej. a Internet, a una LAN de clientes, etc), que puede ser no fiable (en el caso de Internet) o quizás incluso fiable (otro piso de tu edificio). Al igual que los cortafuegos de los grandes edificios, un cortafuegos de red puede evitar e incluso bloquear la extensión del ataque si un segmento se ve comprometido con éxito, al igual que su homónimo cortafuegos puede evitar que la red se siga viendo comprometida.

Linux ha tenido capacidad de cortafuegos desde hace ya un tiempo, en forma de ipfwadm, que era un filtro a nivel de paquetes muy sencillo. Con la llegada del kernel 2.1, se ha visto reemplazado por ipchains, que es un poco más sofisticado. Este a su vez se verá reemplazado en el propuesto kernel 2.4, con un filtrado de paquetes todavía más avanzado, que es más independiente. Sin embargo, ambos todavía siguen siendo filtros de paquetes, y no permiten características más avanzadas como la inspección de estados o algunos tipos de conexiones proxy. Sin embargo Linux soporta IPMASQ, una forma de NAT (Traducción de Direcciones de Red, Network Address Translation). El IPMASQ permite enganchar una red de ordenadores a Internet, pero haciendo un proxy de sus conexiones a nivel de IP. De tal forma que todo el tráfico parezca provenir y dirigirse a una máquina (la máquina Linux con IPMASQ), lo cual proporciona un alto grado de protección a la red interna. Como plus añadido, los clientes de la red interna NO necesitan configurar su proxy; mientras el servidor IPMASQ del Linux esté bien configurado y los clientes lo utilicen como su puerta de enlace por defecto, todo irá bien.

Ambos ipchains e ipfwadm proporcionan las siguientes funcionalidades:

- * bloqueo / permiso del paso de datos basado en IP/puerto/interface origen/destino
- * enmascaramiento de conexiones, basado en IP/puerto/interface origen/destino

Además, ipchains soporta:

- * redireccionamiento de puertos
- * creación de cadenas, para reglas y condiciones más complejas, más fácil de mantener
- * routing de calidad de servicio (QOS, Quality of Service), útil en conexiones de baja velocidad o saturadas
- * especificación de IP/puerto/interface además de especificación inversa (utilizando el !)

El HOWTO del cortafuegos y las páginas "man <command>" (ipchains o ipfwadm) se ocupan en gran detalle de la mecánica para la configuración de las reglas, pero en realidad no se ocupan de la estrategia para hacer un filtrado de forma segura. La primera elección que se debe hacer es si se va a seguir una política de denegación o de permiso por defecto, seguido de qué servicios y hosts se quiere permitir y bloquear.

Cuando se vaya a decidir la política, se debería escoger aquella que deniego todo por defecto, a menos que esté específicamente permitido (de forma que si existe un fallo, con suerte se vea minimizado vía política por defecto) o una política que permita todo y bloquee ciertos servicios/hosts. Generalmente suelo utilizar una política de denegación por defecto, pues de esta forma puedo

arreglar errores y cambios de forma más segura que una política que permita el flujo de datos por defecto.

Pongamos por caso, se tiene un servidor asegurado vía filtrado con cortafuegos, ejecutando Apache, se instala WU-FTPD para uso interno (de modo que la gente pueda subir ficheros) a las 3 de la mañana, y se te olvida cambiar las reglas del cortafuegos. Si se ha escogido una política permisiva por defecto, cualquiera puede acceder al servidor ftp desde Internet, y además, cometiste el error de instalar una versión antigua que permitía a cualquiera comprometer la máquina. Si por otra parte, se sigue una política de denegación por defecto, no hubiesen accedido al servidor de ftp, ni lo hubiesen hecho tus usuarios, pero te darías cuenta más rápidamente. Los usuarios enfadados son algo más sencillo de tratar que una red que haya sido comprometida.

En esta sección, he decidido no tratar específicamente las reglas de filtrado del cortafuegos, daré ejemplos para cada servicio de red, puesto que para filtrar adecuadamente un protocolo primero hay que entender cómo se comporta. Por ejemplo, existe una gran diferencia entre filtrar el www y el ftp en cuanto a accesos internos y externos. Algunas reglas/conceptos generales:

IPFWADM

El Ipfwadm es un sólido paquete de filtrado para Linux, aunque carece de muchas características disponibles en Ipchains. Ipfwadm sólo soporta 3 objetivos para cada paquete: aceptar, denegar o rechazar, mientras que las reglas del ipchains se pueden dirigir a 6 objetivos, o a un objetivo definido por el usuario. En realidad, el Ipfwadm sólo es apropiado para un cortafuegos sencillo a nivel IP, enmascaramiento de IP y si se tiene previsto utilizar FreeS/WAN. Las opciones básicas son: especificar una dirección (dentro, fuera, o ambas, útil con el flag de interface), reglas de entrada, reglas de salida, reglas de redireccionamiento (pongamos que se tienen múltiples interfaces, también se ocupa de las reglas de enmascaramiento) y reglas de enmascaramiento que controlan el comportamiento del enmascaramiento (timeouts, etc). Se pueden insertar, añadir y borrar reglas, configurar políticas por defecto y listar todas las reglas. Aparte de eso es muy parecido a ipchains, con pequeñas variaciones. Lo que viene a continuación es un script apropiado para un servidor que está haciendo de bridge entre 2 redes (10.0.0.x en eth0, 10.0.0.1 y 192.168.0.x en eth1, 192.168.0.1) ejecutando un servidor de correo.

```
#!/bin/bash

#

# Primero limpiar todas las reglas

#

ipfwadm -f -I

ipfwadm -f -O

ipfwadm -f -F

#

# Permitir el redireccionamiento entre las dos redes y si no es entre # ellas,
denegarlos

#

ipfwadm -F -a accept -P all -S 10.0.0.0/24 -i eth0 -D 192.168.0.0/24
```

```

ipfwadm -F -a accept -P all -S 192.168.0.0/24 -i eth1 -D 10.0.0.0/24

ipfwadm -F -p deny

#

# Y por supuesto hay que dejar que entren los paquetes

#

ipfwadm -I -a accept -P tcp -S 10.0.0.0/24 -i eth0 -D 192.168.0.0/24

ipfwadm -I -a accept -P tcp -S 192.168.0.0/24 -i eth1 -D 10.0.0.0/24

#

# Dejarles acceder al servidor de correo pero a nada más

#

ipfwadm -I -a accept -P tcp -S 10.0.0.0/24 -i eth0 -D 10.0.0.1 25

ipfwadm -I -a accept -P tcp -S 192.168.0.0/24 -i eth0 -D 192.168.0.1 25

ipfwadm -I -p deny

```

Ahora el FreeS/WAN soporta la serie 2.2.x del kernel, nunca se debería escoger ipfwadm sobre ipchains. Ipchains ofrece un grado de control mucho más afinado y es mucho más flexible que ipfwadm.

IPCHAINS

El ipchains contiene algunas características nuevas comparado con ipfwadm; se pueden crear cadenas de reglas (de aquí el nombre) y enlazarlas juntas, haciendo más sencilla la administración de cortafuegos. El ipchains soporta más objetivos que ipfwadm; se puede apuntar una regla a: ACCEPT, DENY, REJECT, MASQ, REDIRECT o RETURN o a una cadena definida por el usuario. Como tal es bastante potente, se podría, por ejemplo, redireccionar todos los paquetes destinados al puerto 80 (tráfico www) de mi puerta de enlace para que se redirigiesen al puerto local 3128, el servidor proxy de Squid. También se puede utilizar esto junto con el routing de calidad de servicio, el ejemplo dado en la documentación de ipfwadm es priorizar el tráfico sobre un enlace PPP, se le puede dar una prioridad más alta al tráfico de telnet que al de, pongamos, ftp, reduciendo los problemas de latencia causados por un enlace saturado. Por lo general, creo un fichero /etc/rc.d/init.d/ipchains-sh (o en cualquier otro lugar apropiado) y lo llamo inmediatamente después de que arranca la red, lo cual deja un tiempo pequeño durante el cual el servidor es vulnerable, pero mínimamente, pues todavía no se están ejecutando demonios de red.

El siguiente script es apropiado para una puerta de enlace ejecutándose con 2 interfaces, que es por lo que he utilizado el objetivo DENY en lugar de REJECT, de modo que se descarte el paquete y no se responda de ninguna manera, lo cual ralentiza los escaneos de red (puesto que esperan el timeout en lugar de recibir una respuesta) y revela menos información. También desaconsejaría guardar logs de los datos, a menos que se disponga de la suficiente cantidad de espacio en disco duro, puesto que cada paquete que se envía (varios bytes) se utilizan muchos bytes de disco duro para crear la entrada del log, siendo fácil saturar el syslog y/o el disco duro en una conexión rápida. La página del ipchains se encuentra en: <http://www.rustcorp.com/linux/ipchains/>

```
#!/bin/bash

#
# Este script configura las reglas apropiadas de un cortafuegos para un
# servidor con 2 interfaces ejecutándose como puerta de enlace.
#
# Si se planea utilizarlo, es necesario editar este script.
#
# Se supone que las máquinas internas hacen todas una llamada a la puerta
# de enlace, de modo que las reglas no bloquean el tráfico interno.
#
# Un par de variables
#
# ETH0IP es la dirección IP de ETH0 (el interfaz externo)
# ETH0NET es la red
# ETH0NETMASK es la máscara de red
# HOSTFIABLE1 es un host fiable (para administración de web/ssh)
# HOSTFIABLE2 es un host fiable (para administración de web/ssh)
# ETH1IP es la dirección IP de ETH1 (el interfaz interno)
# ETH1NET es la red
# ETH1NETMASK es la máscara de red
#
ETH0IP=1.1.1.1
ETH0NET=1.1.1.0
ETH0NETMASK=24
HOSTFIABLE1=1.5.1.1
HOSTFIABLE2=1.5.1.2
ETH1IP=10.0.0.1
ETH1NET=10.0.0.0
ETH1NETMASK=24
#
PATH=/sbin
```

```

# LIMPIAR TODAS LAS REGLAS

ipchains -F input

ipchains -F output

ipchains -F forward

# ANTI-SPOOFING

ipchains -A input -p all -j DENY -s 10.0.0.0/8 -i eth0 -d 0.0.0.0/0

ipchains -A input -p all -j DENY -s 127.0.0.0/8 -i eth0 -d 0.0.0.0/0

ipchains -A input -p all -j DENY -s 192.168.0.0/16 -i eth0 -d 0.0.0.0/0

ipchains -A input -p all -j DENY -s 172.16.0.0/16 -i eth0 -d 0.0.0.0/0

ipchains -A input -p all -j DENY -s $ETH0IP -i eth0 -d 0.0.0.0/0

# PRIMERO ICMP

ipchains -A input -p icmp -j ACCEPT -s $ETH0NET/$ETH0NETMASK -i eth0 -d
0.0.0.0/0

ipchains -A input -p icmp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0

# SSH

ipchains -A input -p tcp -j ACCEPT -s $HOSTFIABLE1 -i eth0 -d 0.0.0.0/0 22

ipchains -A input -p tcp -j ACCEPT -s $HOSTFIABLE2 -i eth0 -d 0.0.0.0/0 22

# BLOQUEO 1:1023

ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 1:1023

ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 1:1023

# BLOQUEO DE OTRAS COSAS

ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 1109

ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 1524

ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 1600

ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 2003

ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 2049

ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 2105

ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 3001

ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 3001

ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 3128:3130

ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 3128:3130

```

```

ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 3306
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 3306
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 4444
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 6000:6100
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 6000:6100
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 6667
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 7000

# ADMINISTRACIÓN DE WEB

ipchains -A input -p tcp -j ACCEPT -s $HOSTFIABLE1 -i eth0 -d 0.0.0.0/0 10000
ipchains -A input -p tcp -j ACCEPT -s $HOSTFIABLE2 -i eth0 -d 0.0.0.0/0 10000
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -i eth0 -d 0.0.0.0/0 10000

# REGLAS DE REDIRECCIONAMIENTO

ipchains -P forward DENY

ipchains -A forward -p all -j MASQ -s $ETH1NET/$ETH1NETMASK -d 0.0.0.0/0

```

NETFILTER

El NETFILTER es la siguiente generación de filtrado de paquetes para Linux. Debería de hacer otro tipo de cosas más fácilmente, como cortafuegos, IPsec, cualquier cosa que tenga que ver con la gestión de paquetes. El HOWTO se encuentra disponible en: <http://netfilter.kernelnotes.org/>

IPF

El IPF es un paquete de cortafuegos alternativo, disponible para Linux (y la mayoría de sistemas operativos). Se puede conseguir en: <http://cheops.anu.edu.au/>

SINUS Firewall

El SINUS Firewall es un cortafuegos alternativo para Linux (kernel 2.0.x and 2.2.x). Se puede conseguir en: <http://www.sinusfirewall.org/>.

Phoenix Adaptive Firewall

Estoy en proceso de evaluar el producto, sin embargo parece bastante prometedor. Reemplaza ipchains completamente y añade un montón de inteligencia al proceso de filtrado. Sin embargo es un producto comercial (sobre los 3000\$ US), y es el primer cortafuegos para Linux en estar certificado por la ICASA. Está disponible en: <http://www.progressive-systems.com/products/phoenix/>

Creación de Reglas

ipfwadm2ipchains

Un simple script que convierte las reglas de ipfwadm a ipchains, haciendo más fácil la migración. El script se encuentra disponible en la siguiente

dirección: <http://users.dhp.com/~whisper/ipfwadm2ipchains/>

mason

Mason es un generador automático de reglas para ipfwadm e ipchains. Se carga y monitoriza el flujo de paquetes a través de la máquina, y después, basándose en eso crea un conjunto de reglas para permitir ese tipo de actividad (p. ej. si se hace un ftp al servidor desde un sitio remoto, permitirá ese tipo de acceso en las reglas que crea). Es una buena herramienta para administradores de cortafuegos novatos, disponible en: <http://users.dhp.com/~whisper/mason/>

firewall.sh

Un script basado en diálogo que te conduce a través de la creación de reglas de filtrado, bastante bien hecho y está orientado a nuevos usuarios, disponible en: <http://devplanet.fastethernet.net/Utilities/>

Mklinuxfw

Mklinuxfw es una herramienta perl dirigida a proporcionar una variedad de interfaces (CGI, KDE, línea de comandos, etc.) para la creación de reglas de cortafuegos. Actualmente soporta interfaz CGI y está en progreso el GTK. Se puede descargar de: <http://www.madhouse.org.uk/~red/framepage.phtml?mklinuxfw/index.html>

kfirewall

kfirewall es una aplicación basada en GUI para la creación de reglas ipfwadm o ipchains. Se puede conseguir en: <http://megaman.ypsilonia.net/kfirewall/>

fwconfig

fwconfig es una herramienta interesante para configurar ipfwadm e ipchains, basada en www. Se puede descargar desde: <http://www.mindstorm.com/~sparlin/fwconfig.shtml>

xipfwadm

xipfwadm es una aplicación Tcl/Tk para X que simplifica la creación de reglas ipfwadm. Se puede conseguir en: <http://www.x25.org/xipfwadm.html>

Firewall Manager

Firewall Manager es una aplicación Tcl/Tk orientada a ser ejecutada desde X-Window que proporciona un GUI para gestión de cortafuegos. Se puede descargar desde: <http://www.tectrip.net/arg/>

Linux Firewall Tools

Un sitio interesante, tiene un cgi online para crear scripts de cortafuegos, aunque a mi no me funcionó (muy lento). Se puede ver en: <http://www.linux-firewall-tools.com/>.

FCT - Firewall Configuration Tool

Una de las herramientas de configuración online mediante cgi más avanzadas. Se puede probar en <http://www.fen.baynet.de/~ft114/FCT/index.htm>.

DNi

DNi es un cgi online que te ayuda a crear reglas de cortafuegos para ipfwadm.

Se puede probar en: <http://members.tripod.com/~robeldni/>.

Telnet

Telnet fue uno de los primeros servicios de lo que ahora se conoce como Internet, permitiéndote hacer login interactivo en una máquina remota, lanzar comandos y ver sus resultados. Todavía sigue siendo la herramienta primaria por defecto para administración remota en la mayoría de los entornos, y cuenta con soporte casi universal (incluso el NT tiene un demonio y un cliente de telnet). También es uno de los protocolos más inseguros, susceptible al sniffing, hijacking, etc. Si se tienen clientes utilizando telnet para llegar hasta el servidor, se debería hacer un chroot de sus cuentas si esto es posible, de igual forma que restringir el telnet a los hosts que se utilicen mediante TCP_WRAPPERS. La mejor solución para asegurar el telnet es deshabilitarlo y utilizar telnet con SSL o el ssh.

Los problemas con telnet incluyen:

- * Autenticación en texto claro, nombre de usuario y contraseña.
- * Texto en claro de todos los comandos.
- * Ataques de adivinación de contraseñas (como mínimo acabarán en los ficheros de log)

La mejor solución es desactivar el telnet y utilizar ssh. Sin embargo esto no es práctico en todas las situaciones. Si es necesario utilizar telnet, sugeriría encarecidamente filtrarlo mediante un cortafuegos, tener reglas para permitir a los hosts/redes acceso a puerto 23, y después tener una regla general denegando acceso al puerto 23, al igual que utilizar TCP_WRAPPERS (lo cual es más eficiente, puesto que el sistema sólo comprueba cada conexión de telnet y no cada paquete contra las reglas del cortafuegos) sin embargo utilizar TCP_WRAPPERS le permitirá a la gente dar por hecho que se está ejecutando telnet, les permite conectar, se evalúa la conexión, y después se cierra si no se está listado como permitido el acceso. Un ejemplo de reglas del cortafuegos:

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 23
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiabile -d 0.0.0.0/0 23
```

```
ipchains -A input -p all -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 23
```

Un ejemplo de lo mismo utilizando TCP_WRAPPERS:

En /etc/hosts.allow

```
in.telnetd: 10.0.0.0/255.0.0.0, un.host.fiabile
```

Y en /etc/hosts.deny

```
in.telnetd: ALL
```

Existen varias alternativas cifradas al telnet, como ya se mencionó más arriba, ssh, SSlEay Telnet y otras utilidades de terceros, a mi personalmente me parece que la "mejor" alternativa si te vas a tomar la molestia de cambiar el telnet por algo mejor es utilizar ssh.

Para asegurar las cuentas de los usuarios con respecto a telnet, se pueden hacer varias cosas. La primera sería no permitir al root hacer login vía telnet, lo cual se controla mediante el /etc/securetty y por defecto en la

mayoría de las distribuciones el root tiene restringido el acceso a la consola (una buena cosa). Para que un usuario haga login con éxito, su shell tiene que ser válido (lo cual viene determinado por la lista de shells de /etc/shells), de modo que configurar cuentas de usuario a las que se les permita hacer login es simplemente cuestión de configurar su shell a alguno de los listados en /etc/shells. Es hora de algunos ejemplos prácticos de lo que se puede conseguir configurando el shell del usuario para otro tipo de cosas además de para hacer shell.

Para un PSI que quiere permitir a sus clientes cambiar sus contraseñas con facilidad, pero no permitirles acceso al sistema (mi PSI utiliza Ultrasparcs y por alguna razón se niega a distribuir cuentas de usuario, me pregunto porqué).

en /etc/shells se lista:

```
/usr/bin/passwd
```

Y se cambia el shell de los usuarios por /usr/bin/passwd, de modo que se tiene algo así:

```
nombreusuario:x:1000:1000::/home/nombreusuario:/usr/bin/passwd
```

et voilà. El usuario hace un telnet al servidor, se le pregunta su nombre de usuario y contraseña, y después se le pide cambiar la contraseña. Si se hace correctamente, passwd termina y se les desconecta. Si no tienen éxito, passwd sale y se les desconecta. Lo que sigue es una transcripción de tal configuración cuando un usuario hace telnet:

```
Trying 1.2.3.4...
```

```
Connected to localhost
```

```
Escape character is '^['.
```

```
Red Hat Linux release 5.2 (Apollo)
```

```
Kernel 2.2.5 on an i586
```

```
login: tester
```

```
Password:
```

```
Changing password for tester
```

```
(current) UNIX password:
```

```
New UNIX password:
```

```
Retype new UNIX password:
```

```
passwd: all authentication tokens updated successfully
```

```
Connection closed by foreign host.
```

Telnet también muestra un banner por defecto cuando se conecta alguien. El banner suele contener información del sistema, como el nombre, el SO, la versión y a veces otro tipo de información detallada, como la versión del kernel. Antaño esto era útil cuando se trabajaba en múltiples SO's, sin embargo en la Internet hostil de hoy suele ser más perjudicial que útil. Telnet muestra los contenidos del fichero /etc/issue.net (generalmente es idéntico a /etc/issue el cual se muestra en los terminales, etc.), este fichero se suele

volver a crear al arrancar, en la mayoría de las distribuciones de Linux, desde el fichero de arranque rc.local. Simplemente edita el fichero rc.local, ya sea modificando lo que pone en /etc/issue y /etc/issue.net, o comentando las líneas que crean esos ficheros, y después editando los ficheros con información estática.

Los contenidos de un fichero rc.local típico pertenecientes a /etc/issue y /etc/issue.net:

```
# This will overwrite /etc/issue at every boot. So make any changes
# you want to make to /etc/issue here or you will lose them when you
# reboot.

echo "" > /etc/issue

echo "$R" >> /etc/issue

echo "Kernel $(uname -r) on $a $(uname -m)" >> /etc/issue

cp -f /etc/issue /etc/issue.net

echo >> /etc/issue
```

simplemente comenta las líneas o elimina los comandos uname. Si es absolutamente necesario habilitar el telnet para hacer logins de usuarios, asegúrate de mostrar una advertencia:

Este sistema es exclusivamente para usos autorizados.

Los infractores serán perseguidos.

o algo similar. Legalmente se está en una posición más fuerte si alguien revienta el sistema o abusa de cualquier otra forma de tu demonio telnet.

SSH

SSH es un protocolo seguro y un conjunto de herramientas para reemplazar otras más comunes (inseguras). Fue diseñado desde el principio para ofrecer un máximo de seguridad y permitir el acceso remoto a servidores de forma segura. SSH se puede utilizar para asegurar cualquier tráfico basado en red, configurándolo como un pipe (p. ej., vinculándolo a cierto puerto en ambos extremos). Es bastante cutre, pero está bien para utilizar X a través de Internet. Además de esto, los componentes del servidor se ejecutan en la mayoría de sistemas UNIX, y NT, y los componentes del cliente se ejecutan en casi cualquier cosa. Por desgracia SSH ya no es gratis; sin embargo hay un proyecto para crear una implementación gratis del protocolo SSH.

No existen tantos problemas con el SSH per se como existen con telnet, todo el tráfico de la sesión va cifrado y el intercambio de llaves se hace de forma relativamente segura (opcionalmente se pueden precargar las llaves al final, para evitar que sean transmitidas y ser vulnerables a ataques tipo 'man in the middle', hombre de por medio. SSH se suele ejecutar como un demonio, y se puede cerrar utilizando el fichero `sshd_config`. También se puede ejecutar `sshd` desde `inetd`, y de tal forma utilizar `TCP_WRAPPERS`, y por defecto los rpm's de `ftp://ftp.replay.com/` tienen la opción de `TCP_WRAPPERS` compilada. De modo que utilizar "`sshd: blahblah`" en `hosts.allow` y `hosts.deny` te permite restringir con facilidad el acceso a ssh. Por favor, ten en cuenta que las primeras versiones de ssh contienen bugs, y se han hackeado sitios (generalmente con ataques tipo 'man in the middle' o problemas de desbordamientos de pila en el código ssh), pero la última versión de ssh tiene en cuenta estos problemas. El principal asunto con ssh es su licencia, sólo es gratis para usos no comerciales, sin embargo se puede descargar el código fuente de una gran variedad de sitios. Si se quiere instalar ssh con facilidad, hay un script llamado "install-ssh" que descargará, compilará e instalará el ssh sin dolor, está disponible en:

```
ftp://ftp.yellowdoglinux.com/pub/yellowdog/install-ssh/
```

Las reglas del cortafuegos para ssh son bastante parecidas a telnet. Por supuesto que está `TCP_WRAPPERS`, el problema con `TCP_WRAPPERS` es que un atacante se conecta al puerto, pero no consigue un demonio, SIN EMBARGO sabe que hay algo en ese puerto, mientras que mediante el cortafuegos, ni siquiera se consigue conexión con el puerto. Lo siguiente es un ejemplo de cómo permitir a la gente ejecutar ssh desde máquinas internas, y ciertas clases C en Internet (por ejemplo la clase C que utiliza tu PSI para su batería de módems de acceso).

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 22
```

```
ipfwadm -I -a accept -P tcp -S bateria.módems.psi/24 -D 0.0.0.0/0 22
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 22
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 22
```

```
ipchains -A input -p tcp -j ACCEPT -s bateria.módems.psi/24 -d 0.0.0.0/0 22
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 22
```

O vía `TCP_WRAPPERS`

```
hosts.allow:
```

```
sshd: 10.0.0.0/255.0.0.0, bateria.módems.psi/255.255.255.0
```

```
hosts.deny:
```

```
sshd: 0.0.0.0/0.0.0.0
```

Además de esto, por defecto el ssh trae un fenomenal fichero de configuración, /etc/sshd/sshd_config en la mayoría de las instalaciones. Se puede restringir con facilidad a quién se le permite hacer login, qué hosts, y qué tipo de autenticación les está permitido utilizar. El fichero de configuración por defecto es relativamente seguro, pero lo que sigue es uno más seguro con explicaciones. Ten en cuenta que toda esta información se puede obtener con un "man sshd", la cual es una de las pocas páginas que están bien escritas. Lo que sigue es un fichero sshd-config típico:

```
Port 22
```

```
# se ejecuta en el puerto 22, el standard
```

```
ListenAddress 0.0.0.0
```

```
# escucha en todos los interfaces, quizás sería preferible vincularlo
```

```
# sólo a un cortafuegos interno, etc.
```

```
HostKey /etc/ssh/ssh_host_key
```

```
# dónde se encuentra la llave del host
```

```
RandomSeed /etc/ssh/ssh_random_seed
```

```
# dónde se encuentra la simiente aleatoria
```

```
ServerKeyBits 768
```

```
# durante cuanto tiempo dura la llave del servidor
```

```
LoginGraceTime 300
```

```
# cuánto tiempo se tiene para introducir las credenciales
```

```
KeyRegenerationInterval 3600
```

```
# cada cuánto tiempo se regeneran las llaves del servidor
```

```
PermitRootLogin no
```

```
# permitir hacer login al root? ni hablar
```

```
IgnoreRhosts yes
```

```
# ignorar los ficheros .rhosts de los usuarios? Pues claro
```

```
StrictModes yes
```

```
# para asegurarse de que los usuarios no hacen tonterías
```

```
QuietMode no
```

```
# Si es sí no hace log de nada. Queremos hacer log de logins/etc.
```

```
X11Forwarding no
# ¿reenviar X11? no habría por qué en un servidor

FascistLogging no
# quizás no querramos hacer demasiado log

PrintMotd yes
# mostrar el mensaje del día? Siempre está bien

KeepAlive yes
# se asegura de que las sesiones se desconectan correctamente

SyslogFacility DAEMON
# ¿quién está haciendo el logging?

RhostsAuthentication no
# la autenticación está usando rhosts o /etc/hosts.equiv No está
# en mi mente. Por defecto es sí, de modo que se desactiva.

RSAAuthentication yes
# permitir autenticación RSA pura? Es bastante segura

PasswordAuthentication yes
# permitir a los usuarios que utilicen su login/contraseña habitual?
# Por qué no.

PermitEmptyPasswords no
# permitir cuentas con contraseñas vacías? no

Otras directivas sshd_conf útiles incluyen:

AllowGroups - permitir a grupos explícitamente (/etc/group) hacer login
utilizando ssh

DenyGroups - deshabilitar explícitamente hacer login a grupos (/etc/groups)

DenyUsers - bloquear explícitamente a los usuarios el hacer login

AllowHosts - permitir ciertos hosts, al resto se les denegará

DenyHosts - bloquea ciertos hosts, al resto se les permitirá

IdleTimeout time - tiempo en minutos/horas/días/etc, que fuerza un logout
haciendo un SIGHUP del proceso.

Software SSH

Fresh Free FiSSH

La mayoría de nosotros todavía nos tenemos que sentar frente a estaciones
```

windows, y los clientes ssh para windows son bastante difíciles de encontrar. Fresh Free FiSSH es un cliente ssh gratuito para Windows 95/NT 4.0. Aunque todavía no está completado, recomendaría echarle un vistazo si eres como yo y tienes muchas estaciones Windows. La URL es: <http://www.massconfusion.com/ssh/>

Tera Term

Tera Term es un cliente gratuito para Windows y tiene una DLL añadida para soportar ssh. Tera Term está disponible en: <http://hp.vector.co.jp/authors/VA002416/teraterm.html>. La DLL añadida para soporte SSH se encuentra disponible en: <http://www.zip.com.au/~roca/ttssh.html>

putty

putty es un cliente SSH para Windows, bastante bueno, y completamente gratis, además de pequeño (184k en la actualidad). Se puede descargar de: <ftp://rak.isternet.sk/mnt/rhcd/misc/putty/>

mindterm

mindterm es un cliente gratuito de ssh en java, se puede conseguir en: <http://www.mindbright.se/mindterm/>

LSH

LSH es una implementación gratuita del protocolo SSH (ambos cliente y servidor), LSH trae licencia GNU y se está empezando a parecer a la alternativa (comercialmente hablando) a SSH (que ya no es gratis). Se puede descargar de: <http://www.net.lut.ac.uk/psst/>, ten en cuenta que está bajo desarrollo.

Secure CRT

Un Telnet/Cliente SSH comercial de software Vandyke. Se puede descargar / comprar en: <http://www.vandyke.com/>

Fsh

Fsh significa "Fast remote command execution", "Ejecución remota rápida de comandos", y en concepto es similar al rsh/rcp. Evita el gasto de tener que estar constantemente creando sesiones cifradas, mediante el uso de un túnel cifrado utilizando SSH o LSH, y ejecutando todos los comandos sobre él. Se puede conseguir en: <http://www.lysator.liu.se/fsh/>

SSH Win32 ports

Existen ports del SSH a Win32 disponibles en: <http://guardian.htu.tuwien.ac.at/therapy/ssh/>

FTP

El FTP solía ser el protocolo más usado en Internet por el tráfico puro de datos hasta que fue sobrepasado por el HTTP hace unos años (sí, una vez hubo un Internet libre de WWW). El FTP hace una cosa, y lo hace bien, transferir ficheros entre sistemas. El protocolo en sí mismo es inseguro, contraseñas, datos, etc, se transfieren en texto claro y pueden ser esnifados con facilidad, sin embargo la mayoría del uso del ftp es 'anónimo', de modo que no es un gran problema. Uno de los principales problemas con que se suelen encontrar los sitios de ftp son los permisos incorrectos en directorios que permiten a la gente utilizar el sitio para distribuir sus propios datos (por lo general material con copyrights). Al igual que con telnet, se debería utilizar una cuenta para hacer ftp que no se utilizara para hacer trabajos administrativos, puesto que la contraseña viaja por la red en texto claro.

En general, los problemas con el ftp incluyen:

- * Autenticación en texto claro, nombre de usuario y contraseña.
- * Texto en claro en todos los comandos.
- * Ataques de adivinación de contraseñas.
- * Configuración inadecuada del servidor y el consiguiente abuso de servidores.
- * Todavía existen varios desagradables ataques de Negación de Servicio en algunos servidores de ftp.
- * Las versiones más antiguas de WU-FTPD y sus derivados tienen ataques de root.

Asegurar FTP no es demasiado difícil, entre los cortafuegos y los TCP_WRAPPERS se puede restringir bastante bien el acceso basado en la dirección IP / hostname. Además, la mayoría de los servidores ftp se ejecutan bajo chroot por defecto para cualquiera con acceso anónimo, o en una cuenta definida como invitado. Con un poco de trabajo, se puede configurar a los usuarios que están haciendo ftp para hacer chroot su directorio personal, o cualquier otra cosa apropiada. También se pueden ejecutar servidores de ftp que cifren los datos (utilizando cosas como SSL/etc.) sin embargo eso significa que tus clientes ftp deben hablar el protocolo de cifrado, y ello no siempre es práctico. También asegúrate de que no se tienen directorios de acceso público en el servidor de ftp que sean a la vez legibles y que se puedan escribir, o si no la gente los explotará para distribuir su propio software (generalmente warez o porno).

Un ejemplo de reglas de filtrado para el cortafuegos:

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 21
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 21
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 21
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 21
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 21
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 21
```

Un ejemplo de lo mismo utilizando TCP_WRAPPERS:

En /etc/hosts.allow

```
in.ftpd: 10.0.0.0/255.0.0.0, un.host.fiable
```

Y en /etc/hosts.deny

```
in.ftpd: 0.0.0.0/0.0.0.0
```

Existen diferentes alternativas cifradas a ftp como se mencionó más arriba, SShLeay FTPD y otras utilidades de terceros. puesto que la mayoría de cuentas ftp no se utilizan como cuentas de administración (contraseñas en texto claro, ya has sido advertido), y es de esperar que se ejecuten con chroot, el riesgo de seguridad se minimiza. Ahora que tenemos cubiertas todas las partes de ftp basadas en red, vamos con la forma de asegurar cuentas de usuarios y el entorno.

WU-FTPD

No recomendaría el uso del WU-FTPD, puesto que tiene muchos problemas de seguridad, y bastantes vendedores de Linux no utilizan WU-FTPD en sus propios servidores de ftp. Recomendaría encarecidamente ProFTPD, que está disponible gratuitamente y se desarrolla en la siguiente sección.

Uno de los principales mecanismos de seguridad de WU-FTPD es el uso de chroot. Por ejemplo: por defecto toda la gente haciendo login de forma anónima tiene /home/ftp/ como el directorio raíz. No pueden salir de aquí y, digamos, mirar el contenido de /home/ o /etc/. Lo mismo se aplica a grupos de usuarios y/o individuos, por ejemplo, se podría configurar a todos los usuarios para que fuesen hechos chroot a /home/ cuando hacen el ftp, o en casos extremos de privacidad de usuarios (digamos un servidor de www albergando múltiples dominios) configurar cada usuario con chroot a su propio directorio personal. Esto se hace mediante el uso de /etc/ftpaccess y /etc/passwd (haciendo man ftpaccess se obtienen toda la información). Daré unos cuantos ejemplos de lo que es necesario hacer para llevarlo a cabo, puesto que al principio puede resultar algo confuso. ftpd también verifica /etc/ftpusers y si el usuario que está intentando hacer login aparece listado en ese fichero (como debería estarlo el root) no le dejará acceder vía ftp.

Hacer chroot a usuarios a medida que van haciendo login en el servidor de ftp es bastante simple, pero está pobremente documentado. La comprobación del servidor de ftp de /etc/ftpaccess en busca de "guestgroup"'s, que son simplemente "guestgroup cualquier-grupo-del-sistema" p. ej. "guestgroup usuarios". El nombre de grupo tiene que estar definido en /etc/group y tener añadidos miembros. Es necesario editar la línea de su fichero de contraseñas, de forma que el servidor de ftp sepa dónde volcarlos. Y puesto que ahora están hechos chroot a ese directorio del sistema, no tienen acceso a /lib, etc, así que hay que copiar ciertos ficheros a sus directorios para que cosas como "ls" funcionen correctamente (siempre un toque elegante).

Configurar un usuario (felipefroilan) de forma que pueda entrar por ftp y acabe siendo hecho chroot a su directorio personal (porque sigue amenazando al administrador con llevarle a cazar gamusinos). Además de esto, felipefroilan puede entrar por telnet y cambiar su contraseña, pero nada más, porque sigue intentando ejecutar bots de irc. El sistema en que está utiliza contraseñas con shadow, por eso hay una 'x' en el campo de contraseñas de felipefroilan.

Lo primero de todo es que felipefroilan necesita tener una cuenta correctamente configurada en /etc/passwd:

```
felipefroilan:x:500:Felipe Juan Froilan:/home/felipefroilan/./:/usr/bin/passwd
```

Lo que significa que el servidor de ftp hará un chroot de felipefroilan y luego le hará un chdir en lo que ahora se conoce como / (/home/felipefroilan para el resto de nosotros). La página del manual de ftpaccess cubre bien todo esto, y por supuesto que /usr/sbin/passwd tiene que estar listado en /etc/shells.

Segundo, para que el servidor de ftp sepa que se le está haciendo chroot, tiene que ser miembro de un grupo (usuariosmalos, genteftp, etc) que venga definido en /etc/group. Y después ese grupo tiene que aparecer listado en /etc/ftpaccess.

Ahora hay que copiar algunas librerías y binarios en la "cárcel" chroot, si no "felipefroilan" no va a poder hacer demasiadas cosas una vez que haya entrado por ftp. Los ficheros necesarios están disponibles como paquete (normalmente llamado "anonftp"), una vez que esté instalado, los ficheros se copiarán a /home/ftp/, te darás cuenta de que hay un /etc/passwd, que sólo se usa para mapear UID's a nombres de usuarios, si quieres que felipefroilan vea su nombre de usuario y no su UID, añade una línea para él (p. ej., copia esta línea desde el /etc/passwd real por esta otra). Lo mismo se aplica al fichero de grupos.

```
sin "felipefroilan*:500:500:::" en /home/felipefroilan/etc/passwd:
```

```
drwxr-xr-x 2 500 500 1024 Jul 14 20:46 felipefroilan
```

y con la línea añadida a /home/felipefroilan/etc/passwd:

```
drwxr-xr-x 2 felipefroilan 500 1024 Jul 14 20:46 felipefroilan
```

y con una línea para el grupo de felipefroilan añadida a /home/felipefroilan/etc/group:

```
drwxr-xr-x 2 felipefroilan felipefroilan 1024 Jul 14 20:46 felipefroilan
```

Ahora felipefroilan puede hacer un ftp al sistema, subir y descargar ficheros desde el directorio /home/felipefroilan, cambiar él mismo su contraseña y no dañar el sistema, ni descargar el fichero de contraseñas u otro tipo de incordios.

El FTP también es un protocolo bastante especial, pues los clientes se conectan al puerto 21 (por lo general) del servidor de ftp, y luego, en el puerto 20 del servidor de ftp se conecta al cliente, y es sobre esta conexión donde viajan los datos reales. Lo cual significa que el puerto 20 tiene que hacer conexiones externas. Hay que tener esto en cuenta cuando se configure un cortafuegos, ya sea para proteger servidores ftp o clientes utilizando ftp. De igual forma, existe un ftp "pasivo", y generalmente se suele utilizar por visores www/etc, lo cual significa conexiones entrantes al servidor en puertos altos (en lugar de utilizar el 20, se ponen de acuerdo en algún otro). Si se pretende tener un servidor ftp público que SÓLO vaya a servir ftp, y nada más, colócalo preferiblemente fuera de nuestra LAN interna (ver Practical Unix and Internet Security para discusiones del concepto 'DMZ'). Se puede conseguir WU-FTPD de <ftp://ftp.wu-ftp.org/>

ProFTPD

ProFTPD es un servidor con licencia GPL que se ejecuta en una variedad de plataformas UNIX. Soporta nuevas características como ftp virtual, configuración por directorio (utilizando ficheros .ftpaccess, similares a los ficheros .htaccess de Apache), soporte para cuentas que han expirado y más. También soporta características bastante útiles como la limitación de descargas y controles de seguridad más férreos que WU-FTPD. Lo recomendaría encarecidamente por encima de otros servidores de FTP para UNIX gratuitos.

El fichero de configuración principal de ProFTPD es /etc/proftpd.conf, tiene un estilo de configuración Apachesca que me gusta mucho. ProFTPD se puede ejecutar desde inetd (y hacer uso de TCP_WRAPPERS) o se puede ejecutar como servidor autónomo. También soporta ficheros de configuración por directorio para limitar el acceso, etc. ProFTPD también soporta ftp virtual (aunque al contrario que con un servidor virtual de www, son necesarias IPs extra) y se puede configurar cada sitio de forma diferente (diferente acceso anónimo, si es que lo hay, y más cosas de entre esas líneas). El fichero general proftpd.conf suele tener una sección que cubre los parámetros globales (inetd o autónomo, número máximo de procesos a ejecutar, como quien se ejecuta, etc.), seguido de un fichero de configuración por defecto, seguido de una configuración específica del sitio (sitios virtuales). En un servidor haciendo hosting virtual probablemente sea una buena idea desactivar "DefaultServer", de modo que cualquier cliente que haga ftp sin ningún objetivo sea denegada, en lugar de volcada dentro del sitio por defecto.

Una configuración de ejemplo de un servidor ProFTPD ejecutándose desde inetd sin acceso anónimo:

```
ServerName "Instalación por defecto de ProFTPD"

ServerType inetd

DefaultServer on

Port 21

Umask 022

MaxInstances 30

User nobody

Group nobody

<Directory /*>

AllowOverWrite on

</Directory>
```

Digamos que, al igual que yo, eres paranoico y quieres controlar el acceso al servidor de ftp mediante direcciones IP, nombres de hosts y nombres de dominio (aunque recomendaría confiar sólo en las IP's). Esto se puede conseguir mediante reglas del cortafuegos, pero eso acaba ralentizando la máquina (especialmente si se añaden multitud de reglas, como tiende a ocurrir). Se puede utilizar TCP_WRAPPERS , pero no sería posible limitar selectivamente el acceso a sitios virtuales, sitios anónimos, sólo al servidor en sí mismo. O se puede hacer en el fichero proftpd.conf utilizando la directiva "<Limit LOGIN>".

El siguiente ejemplo limitará el acceso a 10.1.*.* y 1.2.3.4, a cualquier otra máquina se le denegará el acceso.

```
<Limit LOGIN>

Order Allow, Deny

Allow from 10.1., 1.2.3.4

Deny from all
```

</Limit>

Si se coloca esto dentro de las directivas "<VirtualHost>" o "<Anonymous>", se aplica sólo a ese sitio virtual o configuración anónima, si se coloca en la directiva "<Global>" se aplicará a todas las secciones "<VirtualHost>" y "<Anonymous>", y si se coloca en la configuración del servidor (p. ej. con "ServerName" y elementos relacionados) se comportará como lo haría TCP_WRAPPERS, cualquiera desde 10.1.*.* o desde 1.2.3.4 impacta cuando se intenta conectar al puerto 21, al contrario que si simplemente le negase el login si no está en la sección "<Global>", "<VirtualHost>" o "<Anonymous>".

Si se quiere añadir más accesos anónimos, tan sólo añadir:

```
<Anonymous ~ftp>
```

```
User      ftp
```

```
Group     ftp
```

```
RequireValidShell  off
```

```
UserAlias  anonymous ftp
```

```
MaxClients  10
```

```
DisplayLogin  bienvenida.msg
```

```
DisplayFirstChdir  .message
```

```
<Directory *>
```

```
<Limit WRITE>
```

```
DenyAll
```

```
</Limit>
```

```
</Directory>
```

```
</Anonymous>
```

Esto asignaría el directorio personal de los usuarios "ftp" (suponiendo que la configuración normal de "~ftp" probablemente fuese /home/ftp) como el directorio raíz anónimo, cuando la gente hiciese login anónimo, el ProFTPD se ejecutaría como el usuario "ftp" y el grupo "ftp" (al contrario que si se hiciera login como un usuario normal), y los logins anónimos se limitarían a 10. De igual forma, el fichero /home/ftp/bienvenida.msg se mostraría cuando hiciesen ftp los usuarios anónimos, y cualquier directorio con un fichero .message que contuviera texto, el cual se mostraría al cambiar a tal directorio. El "<Directory >" se ocupa de /home/ftp/*, y después deniega el acceso de escritura a todos, lo cual quiere decir que nadie puede subir ficheros. Si se quisiera añadir in directorio entrante, simplemente se añadiría lo siguiente después de las directivas "<Directory *>":

```
<Directory incoming>
```

```
<Limit WRITE>
```

```
AllowAll
```

</Limit>

<Limit READ>

DenyAll

</Limit>

</Directory>

Lo cual le permitiría a la gente escribir ficheros en /home/ftp/incoming, pero no leerlos (es decir, descargarlos). Como se puede ver, ProFTPD es muy flexible, lo cual da como resultado mayores requerimientos de potencia que el WU-FTPD, pero definitivamente merece la pena por el control añadido. ProFTPD y su documentación se pueden conseguir en: <http://www.proftpd.org/>

proftpd-ldap

El proftpd-ldap te permite hacer consultas por contraseñas utilizando un directorio LDAP, se puede descargar de:
<http://horde.net/~jwm/software/proftpd-ldap/>

NcFTPD

El NcFTPD es un servidor ftp de gran volumen, sin embargo sólo se encuentra disponible para uso personal o educativo. Se puede conseguir en:
<http://www.ncftpd.com/ncftpd.>

BSD ftpd

El servidor ftp de BSD (ftpd) también se ha transportado a Linux, de modo que si necesitas ejecutarlo, se puede descargar de:
<ftp://quatramaran.ens.fr/pub/madore/ftpd-BSD/>

Muddleftpd

El Muddleftpd is un pequeño servidor de ftp. Se puede conseguir en:
[http://www.computing.edu.au/~kuiperba/muddleftpd/.](http://www.computing.edu.au/~kuiperba/muddleftpd/)

Troll ftpd

El Troll ftpd es un servidor ftp extremadamente pequeño y relativamente seguro. No puede ejecutar programas externos, y es bastante fácil de configurar. Se puede conseguir en: <http://www.troll.no/freebies/ftpd.html>.

BetaFTPD

El BetaFTPD es un pequeño servidor ftp de un solo hilo. Se puede conseguir en:
http://members.xoom.com/_XOOM/sneeze/betaftpd.html.

FTP - SSL

Otro reemplazo de tu ftpd favorito (probablemente WU-FTPD), también disponible como un conjunto de parches para el WU-FTPD. Es altamente apropiado, puesto que la mayoría de los servidores tienen muchos usuarios que necesitan acceso ftp. El tarball está disponible en:
<ftp://ftp.uni-mainz.de/pub/internet/security/ssl/>, y como paquetes RPM en <ftp://ftp.replay.com/pub/replay/linux/redhat/>

FTP - SRP

El SRP también se puede utilizar para cifrar la porción nombre de usuario / contraseña de tu sesión ftp, o de la sesión completa. Se puede conseguir en: <http://srp.arcot.com/srp/>

HTTP / HTTPS

El tráfico WWW es uno de los mayores componentes del uso de Internet hoy en día. Existen una variedad de servidores WWW bastantes populares para Linux, siendo el más famoso el Apache (con más del 50% del mercado). La mayoría de los servidores WWW también tienen la capacidad de utilizar SSL para asegurar las sesiones (para comercio electrónico, etc.). Esta sección se centra en Apache, pero tiene sentido, dado que es el servidor por defecto para casi todas las distribuciones Linux (y *BSD). También estoy escribiendo para la versión 1.3.9 de Apache, que ya no utiliza el fichero `access.conf` o `srm.conf`, sino que en su lugar lo ha cambiado por el `httpd.conf`.

Apache

¿Qué puedo decir acerca de asegurar Apache? En realidad no demasiado. Por defecto Apache se ejecuta como el usuario 'nobody', lo cual le da muy poco acceso al sistema, y por lo general el equipo Apache ha hecho un buen trabajo evitando desbordamientos de pila/etc. En general, la mayoría de los servidores www simplemente toman datos del sistema y los envían fuera, los mayores peligros no vienen del Apache sino de programas descuidados que se ejecutan vía Apache (CGI's, server side includes, etc.).

Si se va a ejecutar Apache, recomendaría utilizar la serie 1.3, a menos que se tengan algún tipo de extraños requerimientos para ceñirse a la 1.2, la versión de desarrollo activa es la 1.3, e incluye muchas características nuevas desde el punto de vista de la seguridad, estabilidad y el rendimiento. La mayoría de los servidores basados en Apache (Red Hat Secure Server, Stronghold, etc.) suelen, en general, estar libres de bugs, pero ocasionalmente se dan problemas.

Hacer chroot del Apache

Si se quiere ser paranoico, sugeriría ejecutar Apache en un entorno de chroot, aunque a veces esto puede dar más problemas de lo que merece la pena. Hacer esto estropeará muchas cosas. También es preciso instalar numerosas bibliotecas, perl y cualquier otra utilidad que vaya a utilizar el Apache, así como cualquier fichero de configuración al que se quiera tener acceso. Cualquier script CGI y otras cosas que interactúen con el sistema serán algo problemáticas y en general, más difíciles de reparar.

La forma más simple de configurar el Apache con chroot es instalarlo y mover/editar los ficheros necesarios. Una buena idea es crear un directorio (como `/chroot/apache/`), preferiblemente en un sistema de ficheros separado de `/`, `/usr`, `/etc` (enlaces simbólicos, llenado accidental de particiones, etc...) y después crear para el Apache una estructura de ficheros por debajo. Lo siguiente es un ejemplo, simplemente reemplaza `/chroot/apache/` por algo de tu elección. Por supuesto que hay que ejecutar estos pasos como root para que funcione. RPM lo soporta mediante la directiva "`--root /cualquier/directorio`", simplemente hay que instalar Apache y las bibliotecas necesarias utilizando rpm (y de paso ganando soporte de dependencias/etc, haciéndote la vida más fácil). Si no se está en un sistema basado en RPM, simplemente hay que utilizar ldd para encontrar las librerías compartidas que se necesitan, y mover todo lo que se necesite al directorio `/chroot/apache`

```
[seifried@host seifried]$ ldd /usr/bin/httpdlibm.so.6 => /lib/libm.so.6
(0x40017000)libc.so.6 => /lib/libc.so.6 (0x40060000)/lib/ld-linux.so.2 =>
/lib/ld-linux.so.2 (0x40000000)
```

Las peticiones de logs de Apache se hacen internamente, de modo que no hay porqué preocuparse de instalar pseudo-demonios para hacer logging, como

holelogd, para que pasen esta información al syslog.

Configuración de Apache

En cuanto a la forma más simple de asegurar Apache y asegurarse de que no tiene acceso innecesario al sistema de ficheros es crear un directorio /www/ o algo similar, y situar por ahí debajo TODOS los sitios web, contenido web, cgi's, etc. Después sólo es necesario configurar access.conf para que deniegue el acceso a / , y se lo permita a /www/ y sus varios directorios cgi-bin.

Ejemplo de httpd.conf:

```
<Directory />
```

```
Options None
```

```
AllowOverride None
```

```
</Directory>
```

```
<Directory /www >
```

```
Options Indexes FollowSymLinks Includes
```

```
AllowOverride None
```

```
</Directory>
```

Control de Accesos

El acceso a los directorios también se puede controlar con facilidad, el Apache soporta la definición y localización de ficheros (generalmente conocidos como ficheros htaccess) que controlan el acceso basado en nombre de usuario y contraseña, IP de origen, etc. Esto se define en srm.conf:

```
AccessFileName .htaccess
```

El formato del este fichero viene desarrollado en la documentación del Apache, y es idéntico a directivas que se colocarían en access.conf (bueno, casi). La autenticación de usuario vía nombre de usuario y contraseña también vien desarrollada en profundidad en <http://www.apacheweek.com/features/userauth/>

También querrás evitar que la gente vea los ficheros .htaccess, coloca esto en tu srm.conf:

```
<Files .htaccess>
```

```
order allow,deny
```

```
deny from all
```

```
</Files>
```

Conseguir Apache

Se puede descargar Apache desde <http://www.Apache.org/>, y se puede descargar Apache-SSL desde <http://www.apache-ssl.org/>, de modo que también se necesitará software OpenSSL, disponible desde: <http://www.openssl.org>. Ten en cuenta que el uso de Apache-SSL es ilegal en USA debido a las patentes que mantiene sobre RSA. El servidor comercial de Apache-SSL más barato es el Red Hat Secure Server, a 100\$ USA.

Apache con extensiones SSL

Existen diferentes alternativas gratuitas al Apache con SSL, y varias comerciales. Si se está en los EE.UU., el RSA está patentado, así que hay que utilizar o bien DSA (para el cual es difícil conseguir certificados) o comprar un servidor comercial basado en Apache (como Stronghold). Si se está en Europa, se puede vivir en un país donde el IDEA esté patentado, de modo que asegúrate primero.

Apache-SSL

Es el que uso actualmente (sencillamente porque lo probé antes que Apache con mod_ssl y funcionó). Tienes que conseguir Open-SSL, compilarlo e instalarlo, y después parchear Apache con el parche Apache SSL, compilar Apache, y ya está. El Open-SSL se encuentra disponible en: <http://www.openssl.org/>, tan sólo hay que conseguir el último tarball, desempaquetarlo y ejecutar:

```
./config
```

```
make
```

```
make test
```

```
make install
```

A mi me ha funcionado siempre. Hay que conseguir el material de Apache-SSL de <http://www.apache-ssl.org>, desempaquetar el código fuente del Apache en algún lugar, hacer un cd al directorio de nivel superior (/usr/local/src/apache_1.3.9/) y después desempaquetar el material de Apache-SSL dentro (te dice que hagas eso en los docs). Después sólo hay que ejecutar:

```
./FixPatch
```

Lo cual debería funcionar (si no funciona, lee el README.SSL), después configura el Apache como siempre, y ejecuta make seguido de make install. Sáltate hasta la sección "Crear un certificado".

Apache con mod_ssl

El Apache con mod_ssl se encuentra disponible en <http://www.modssl.org> Todavía no lo he probado.

Crear un certificado

Esta es la parte fácil, el siguiente paso es crear el conjunto de llaves, y después configurar el httpd.conf para utilizarlo correctamente. Busca dónde está instalado el "openssl" y asegúrate de que está en el path, después haz un cd allí donde quiera que tengas ubicados tus ficheros de configuración del Apache (cualquiera que fuese el prefijo como el raíz de Apache seguido de /conf). Si se necesita crear un certificado de prueba, para uso interno, se puede hacer:

```
openssl genrsa -des3 > httpsd.key
```

```
openssl req -new -x509 -key httpsd.key > httpsd.crt
```

Los navegadores se quejan sobre este certificado, puesto que está creado por la persona que lo firma, y no son fiables. Si quieres generar un certificado, y una petición de certificado para enviar a alguien como Thawte o Verisign,

entonces hay que hacer:

```
openssl genrsa -des3 > httpsd.key
```

```
openssl req -new -key httpsd.key > httpsd.csr
```

También se pueden conseguir certificados reales con un tiempo de vida limitado (generalmente de una o dos semanas) de Verisign, para utilizarlos en un entorno más real.

Configurar el Apache para SSL

Se necesitan añadir varias cosas al fichero de configuración de Apache para conseguir que el Apache con extensiones SSL haga algo útil con tus certificados. Habrá que añadir algunas configuraciones globales (ten en cuenta que esto se refiere a la 1.3.9, y que no funcionará con versiones más antiguas de Apache):

```
# Hay que decirle al Apache que escuche en el puerto 443
```

```
# por defecto sólo escucha en el 80
```

```
Listen 443
```

```
# Si utilizas más de un sitio seguro en una IP (MALA IDEA)
```

```
# necesitarás:
```

```
NameVirtualHost 10.1.1.1:443
```

```
# Es una buena idea deshabilitar el SSL globalmente y habilitarlo
```

```
# basado en hosts
```

```
SSLDisable
```

```
# SSL cache server, sin esto el servidor morirá
```

```
dieSSLCacheServerPath /usr/bin/gcache
```

```
# Puerto en el que se ejecuta el servidor
```

```
SSLCacheServerPort 12345
```

```
# timeout del SSL cache, acortarlo para hacer pruebas
```

```
# 300 es un buen valor del "mundo real"
```

```
valueSSLSessionCacheTimeout 300
```

Ahora puedes crear un host virtual con SSL habilitado:

```
<VirtualHost www.example.com:443>
```

```
DocumentRoot /www/secure/
```

```
ServerName www.example.com
```

```
ServerAdmin example@example.com
```

```
ErrorLog logs/https_error.log
```

```

TransferLog logs/https_access.log

# Habilitar SSL para este host virtual

SSLEnable

# esto prohíbe el acceso excepto cuando se utiliza el SSL. Muy
# cómodo para defenderse contra errores de configuración que
# ponen al descubierto elementos que deberías estar protegidos
SSLRequireSSL

SSLCertificateFile /usr/conf/httpsd.crt

# Si la llave no está combinada con el certificado,
# utiliza esta directiva para apuntar al fichero de la llave
# [OPCIONAL]

SSLCertificateKeyFile /usr/conf/httpsd.key

# Si se requiere que los usuarios tengan un certificado, se
# necesitarán un montón de certificados raíz, para que se puedan
# verificar sus certificados personales
# SSLCACertificateFile /etc/ssl/ca-cert-bundle.pem

SSLVerifyClient none

</VirtualHost>

```

Filtrado con el cortafuegos del HTTP / HTTPS

El HTTP se ejecuta en el puerto 80, con tcp, y si es sólo para uso interno (una Intranet, o un mecanismo de control basado en www, para, digamos, un servidor cortafuegos) definitivamente habría que filtrarlo con el cortafuegos.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 80
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 80
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 80
```

o en ipchains:

```
ipchains -A input -p all -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 80
```

```
ipchains -A input -p all -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 80
```

```
ipchains -A input -p all -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 80
```

El HTTPS se ejecuta en el puerto 443, con tcp, y si sólo es para uso interno también debería filtrarse con el cortafuegos:

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 443
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 443
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 443
```

o en ipchains:

```
ipchains -A input -p all -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 443
```

```
ipchains -A input -p all -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 443
```

```
ipchains -A input -p all -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 443
```

Añadidos al Apache

apache-userdirldap

El apache-userdirldap te permite utilizar el directorio LDAP para buscar en los directorios de los usuarios. En otras palabras, si se quiere mover todos los usuarios a un directorio LDAP y hacer toda la autenticación a través de él, no habrá que romper el Apache. Se puede conseguir en:
<http://horde.net/~jwm/software/apache-userdirldap/>

Servidores WWW alternativos

Red Hat Secure Server

Red Hat Secure Server es un producto basado en Apache de (adivina quien) software Red Hat. En esencia es un Apache de fábrica con módulos de cifrado RSA (que es por lo que en realidad estás pagando) y también puede servir peticiones de http standard sin cifrado. Sólo se puede vender en USA y en Canadá, y es la mejor opción (en mi opinión) en cuanto a servidores www seguros que sean de uso legal en US (debido a las patentes RSA). En cuanto a la seguridad, lee la sección anterior sobre Apache / Apache-SSL, todo es aplicable. Red Hat Secure Server cuesta 100\$ USA y se consigue un descuento de 25\$ en el sitio de certificación Thawte (de modo que el certificado de sitio sólo cuesta 100\$). Personalmente me gusta bastante, pues está basado en un software que se ejecuta en más de la mitad de los sitios www del mundo, y como tal es fácil conseguir soporte/actualizaciones/etc. Red Hat Secure Server se puede comprar en:
<http://store.Redhat.com/commerce/>

Roxen

Roxen es otro servidor comercial de www capaz de hacer https y tiene licencia GPL. Se puede descargar gratuitamente si se está en la Unión Europea o en Australia, Canadá, Japón, Nueva Zelanda, EE.UU. o Suiza. Se puede descargar una versión con criptografía "débil" (40 bits) sin ningún problema y desde cualquier país. Roxen es un producto extremadamente sólido y está disponible en: <http://www.roxen.com>

AOL Server

Ya lo sé, parece extraño pero es cierto. El AOL es un servidor www gratuito, con código fuente disponible. No sólo eso, sino que soporta SSL y otras características avanzadas. Definitivamente merece ser tenido en cuenta. Se puede conseguir en: <http://aolserver.com/>

Hay más trabajo en asegurar tu servidor www que instalar el Apache y configurarlo correctamente. La mayoría de los servidores necesitan permitir acceso a sus sistemas de ficheros, de forma que los usuarios puedan subir y modificar ficheros del servidor. Para ello existen 4 métodos que lo cubren en

detalle:

Webfs

Webfs es un servidor ligero de www que implementa funcionalidad básica y se encuentra disponible en: <http://www.in-berlin.de/User/kraxel/webfs.html>

Flash Web Server

Un servidor www ligero y rápido, se puede conseguir en:
<http://www.cs.rice.edu/~vivek/flash/>

Acceso a los ficheros del servidor WWW

En algún momento habrá que acceder a los ficheros del servidor para actualizarlos. Hacer un logging y utilizar un editor de texto como el emacs no suele ser una sabia decisión a largo plazo, si valoras tu tiempo. Hay varios paquetes de autoría de HTML que pueden acceder al website vía FTP o compartición de ficheros de Windows.

FTP

Este es el método clásico de garantizar acceso a los usuarios a los servidores ftp, las preocupaciones habituales incluyen que los usuarios puedan verse sus ficheros entre sí, que vean ficheros del sistema que no deberían, etcétera. Hacer un chroot de las sesiones de los usuarios resolverá la mayoría de estos problemas, sin embargo el principal problema que existe con el FTP es que cifrar el nombre de usuario y la contraseña no suele ser posible debido al hecho de que la mayoría de la gente está utilizando clientes FTP de Windows. Recomendaría el ProFTPD antes que el WU-FTPD para una aplicación de este tipo, el ProFTPD tiene mejores controles de acceso.

Acceso Samba

El Samba es bastante útil para compartir directorios www con clientes Windows, se pueden mantener los nombres de usuarios y contraseñas separados del sistema (utilizando smbpasswd en vez del passwd del sistema) y el cifrado de logins no es ningún problema. Simplemente hay que hacer los directorios compartidos no visibles, y utilizar la directiva "valid users" para restringir qué usuarios pueden ver los datos compartidos. Por ejemplo:

```
[www-example]
```

```
path = /www/www.example.org/
```

```
valid users = example
```

```
read only = No
```

```
browseable = No
```

configurará un directorio compartido bastante seguro del directorio
"/www/www.example.org/" al que sólo podrá acceder el usuario "example".

Acceso Frontpage

El FrontPage es uno de los programas más famosos para edición de HTML entre los usuarios de Windows (qué caramba, incluso yo lo utilizo). Puede comunicarse directamente con los servidores WWW y descargar / subir ficheros de un sitio (llamado el "sitio Frontpage") si el servidor soporta extensiones FrontPage. Las extensiones FrontPage se encuentran disponibles para diferentes plataformas

UNIX, gratuitamente, en Ready To Run Software: <http://www.rtr.com> En el pasado, las extensiones FrontPage de RTR para UNIX han sido un tanto desastrosas. Sin embargo existen alternativas comerciales, una es el Instant ASP, disponible en: <http://www.halcyonsoft.com>

RearSite

El RearSite es un programa cgi que proporciona a los usuarios acceso a sus directorios vía un navegador normal. Se puede conseguir en: <http://listes.cru.fr/rs/fd>

Fast Webpage Exchanger

El Fast Webpage Exchanger mantiene sincronizados los ficheros utilizando un interesante fichero de configuración en el que se especifica todo. Se puede descargar desde: http://www.enjoy.ne.jp/~gm/program/iew_en.html

SMTP

Simple Mail Transfer Protocol (SMTP), es uno de los servicios más importantes que proporciona Internet. Ahora casi todas las compañías tienen o dependen del correo, y por extensión todos los servidores SMTP. Se encuentran disponibles muchos paquetes SMTP, siendo el más viejo y el más probado el Sendmail (ahora con soporte comercial, etc.), y hay dos nuevos contendientes, Postfix y Qmail, ambos dos escritos desde cero teniendo en cuenta la seguridad. Filtrar con el cortafuegos el SMTP no tiene pérdida, se ejecuta en el puerto 25, tcp:

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 25
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 25
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 25
```

o bien con ipchains:

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 25
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 25
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 25
```

Sendmail

Sendmail es otro de esos servicios con los que la mayoría de nosotros hemos tenido relaciones. Odiamos administrarlo y nos encantaría reemplazarlo (en realidad he empezado a eliminar el Sendmail de las máquinas que administro y lo estoy reemplazando con el Postfix).

Sendmail se ha ganado por sí mismo una muy mala reputación en cuanto a seguridad, sin embargo es difícil echarle la culpa al software cuando te encuentras con sistemas ejecutando versiones antiguas del sendmail. La raíz del problema (si se me permite el mal juego de palabras) es que casi todo el mundo ejecuta el sendmail como root (y algo así como el 70% del correo de Internet se maneja mediante máquinas con Sendmail, de modo que hay un montón de ellas), de forma que tan pronto se encuentra un bug, encontrar un sistema que explotar no es tan difícil. Las últimas versiones de sendmail han sido bastante buenas, sin hacks del root, etc, y con las nuevas características anti spam finalmente ha alcanzado la mayoría de edad. Más información en Sendmail y su código fuente está disponible en: <http://www.sendmail.org/>

Hacer un chroot del sendmail es una buena opción, pero lleva demasiado trabajo, y puesto que se ejecuta como root, algo bastante discutible en cuanto a su efectividad (puesto que el root se puede escapar de una cárcel chroot). Personalmente pienso que es mejor invertir el esfuerzo en cambiarse a Postfix o a Qmail.

Mantener actualizado el sendmail es relativamente simple, recomendaría como mínimo la versión 8.9.3 (la serie 8.9 tiene más características anti-spam, la 8.8.x tiene la mayoría de estas características, suponiendo que se ha configurado correctamente el sendmail.cf). La mayoría de las distribuciones vienen con la 8.8.x, aunque las versiones más recientes suelen venir con la 8.9.x. Se puede conseguir el código fuente desde <ftp://ftp.sendmail.org/>, pero compilar el sendmail no es algo para el débil de corazón o para aquellos que no tengan un montón de tiempo para dedicárselo.

Sendmail sólo tiene que estar accesible desde el mundo exterior si se está

utilizando para recibir correo de otras máquinas y repartirlo localmente. Si sólo se quiere ejecutar sendmail de forma que funcione el reparto local (p. ej., una estación de trabajo autónoma, un servidor de prueba u otros) y que se pueda enviar con facilidad el correo a otras máquinas, simplemente filtra con el cortafuegos el sendmail, o mejor, no lo ejecutes en modo demonio (en el cual escucha conexiones). Sendmail se puede ejecutar en la cola de refresco de un nodo, donde simplemente se "despierta" cada cierto tiempo y procesa el correo local, ya sea distribuyéndolo localmente o enviándolo a través de la red. Para configurar la ejecución del Sendmail en modo cola:

edita el script de inicio del Sendmail y cambia la línea que contiene:

```
sendmail -bd -qlh
```

por:

```
sendmail -qlh
```

Ten en cuenta que: si utilizas el sistema para enviar mucho correo quizás prefieras disminuir el tiempo de refresco, quizás "-q15m" (refrescar la cola cada 15 minutos), ahora el correo saliente y el correo interno del sistema se comportarán bien, lo cual a menos que se ejecute un servidor de correo, es perfecto.

Ahora vienen todas esas maravillosas características anti-spam del sendmail. Los ficheros de configuración del Sendmail consisten en (se aplica al Sendmail 8.9.x):

```
/etc/sendmail.cf
```

El fichero de configuración principal, también dice dónde se encuentran el resto de ficheros de configuración.

```
/etc/mail/
```

Se puede definir la localización de los ficheros de configuración en sendmail.cf, generalmente la gente los coloca en /etc/ o en /etc/mail (lo cual lo lía menos).

```
access
```

La base de datos de la lista de accesos, permite rechazar el correo proveniente de ciertas fuentes (IP o dominio), y controlar con facilidad las transmisiones. Mi fichero de acceso es así:

```
* RELAY
```

```
spam.com REJECT
```

lo que quiere decir que a 10.0.0.* (los hosts de mi red interna) se les permite utilizar el servidor de correo para enviar correo donde quieran, y el correo de *.spam.com se rechaza. Hay listas en línea de spammers conocidos, generalmente suele tener entre 5-10.000 entradas, lo cual puede afectar seriamente el rendimiento del sendmail (pues cada conexión se comprueba contra esta lista), por otra parte, tener una máquina sendmail para enviar spam es incluso peor.

```
aliases
```

El fichero de alias, te permite controlar el reparto del correo local al sistema, es útil para hacer una copia de seguridad del correo entrante de un usuario a un spool por separado. La mayoría del software de servidores de

listas utiliza este fichero para enviar el correo que recibe a los programas que realmente procesan las listas. Recuerda ejecutar el comando "newaliases" después de editar este fichero, y después reiniciar el sendmail.

domaintable

la tabla de dominios (añadir dominios) que se maneja, útil para hacer hosting virtual.

majordomo

fichero de configuración de majordomo, personalmente recomendaría SmartList a Majordomo.

sendmail.cw

fichero que contiene nombres de hosts de los que se recibe correo, útil si se alberga más de un dominio.

sendmail.hf

situación del fichero de ayuda (telnet al puerto 25 y escribir "HELP")

virtusertable

Tabla de usuario virtual, mapea usuarios entrantes, p. ej., mapear ventas@ejemplo.org a john@ejemplo.org

Sendmail 8.9.x (y versiones anteriores) en realidad no tienen soporte para hacer un log de todo el correo de una forma agradable (un requisito indispensable por muchas compañías por motivos legales). Esta es una de las características sobre las que se trabaja en la versión de Sendmail 8.10.x. Hasta entonces, hay dos formas de hacer log del correo, la primera es algo agraciada, y registra un log de los correos entrantes a usuarios según cada usuario. El segundo método no es agraciado, e implica un simple log de todas las transacciones SMTP a un fichero, habría que escribir algún tipo de procesador (probablemente en perl) para que el log fuese útil.

El correo (o las conexiones SMTP para ser más precisos) primero se filtra con el fichero access, y es aquí donde se pueden RECHAZAR correos de ciertos dominios/IP's, y TRANSMITIR correo de ciertos hosts (p. ej. tu red interna de máquinas windows). Cualquier dominio local para el que se hospeda el correo tendrá que ir al sendmail.cw. Suponiendo que el correo cumple con las reglas y va a la cola para reparto local, el siguiente fichero que se comprueba en virtusertable, que es una lista de direcciones de correo mapeadas al nombre de la cuenta/otra dirección de correo. p.ej.:

seifried@seifried.org alias-seifried

listuser@seifried.org usuariolista

@seifried.org correos-estropeados

La última regla es para evitar que reboten los correos estropeados, y que en lugar de eso se envíen a un buzón. Después se comprueba el fichero de alias, si se encuentra una entrada se hace lo que dice, y si no, se intenta entregar el correo a un buzón de un usuario local, mi entrada seifried del fichero de alias es:

alias-seifried: seifried, "/var/backup-spool/seifried"

De esta forma mi correo se reparte a mi buzón normal, y a un buzón de copia de seguridad (por si acaso he borrado un correo que no quisiera), o en el peor de los casos, Microsoft Outlook decide cascar un día y cargarse mis buzones. Esto también sería útil en empresas, puesto que ahora se tiene una copia de seguridad de todo el correo entrante según cada usuario, y se les puede permitir (o no) acceder al fichero que contiene el correo en la copia de seguridad.

Una advertencia, cuando se esté usando una regla general para un dominio (p. ej. @seifried.org) hay que crear un alias para CADA cuenta, y para las listas de correo. Si no, cuando se examina la lista y no se encuentra una entrada específica (para, digamos, lista-correo@seifried.org) lo enviará al buzón especificado por la regla general. Ya sólo por este motivo no se debería utilizar una regla general.

El segundo método es muy simple, sencillamente se arranca el sendmail con la opción -x y se especifica un fichero para hacer un log de todas las transacciones. Este fichero crecerá con enorme rapidez, NO recomendaría utilizar este método para hacer log del correo, al menos que sea absolutamente necesario.

Qmail

Qmail (al igual que Postfix) fue creado como una respuesta directa a los fallos percibidos en Sendmail. Qmail es GPL con una cláusula sin distribución binaria que obliga a instalarlo desde el código fuente. Muy poco del código del Qmail se ejecuta como root, y es muy modular comparado con el sendmail (el cual es un trozo de código monolítico). Se puede descargar de: <http://www.qmail.org/>

Postfix

El Postfix es un agente de transferencia de correo (MTA) orientado a la seguridad, velocidad, y facilidad de configuración, cosas en las que Sendmail suele fallar por lo general. Recomendaría encarecidamente reemplazar el Sendmail por el Postfix. La única parte de Postfix que se ejecuta como root es un programa de control maestro, llamado "master", que llama a otros programas para procesar el correo a la cola ("pickup"), un programa para gestionar la cola, espera conexiones entrantes, repartos de correo retrasados, etc. ("qmgr"), un programa que en realidad envía y recibe el correo ("smtpd") etc. Cada parte de Postfix está muy bien pensada, y generalmente hace una o dos tareas, muy bien. Por ejemplo, en lugar del modelo de sendmail, donde el correo simplemente se volcaba a /var/spool/mqueue, en Postfix existe un directorio accesible por el mundo llamado "maildrop" el cual se comprueba mediante "pickup", el cual alimenta los datos a "cleanup", el cual mueve el correo (si está correctamente formateado, etc.) a un directorio seguro de cola para el procesado real.

Los ficheros primarios de configuración están en /etc/postfix, y existen varios ficheros primarios de configuración que es necesario tener:

master.cf

Controla el comportamiento de varios programas de "ayuda", están hechos chroot, el máximo número de procesos que pueden ejecutar, etcétera. Probablemente sea mejor dejar las configuraciones por defecto en la mayoría de los servidores de correo, a menos que se necesite ajustar algo por altas cargas o asegurar el servidor (p. ej. haciéndolo chroot).

main.cf

Este fichero está muy próximo al sendmail.cf (en cuanto a propósito, en cuanto

al diseño es bastante diferente). Está bien comentado y configura todas las variables principales, y las situaciones y formato de diferentes ficheros que contienen información tal como los mapeos a usuarios virtuales e información relativa.

He aquí una lista de variables y localización de ficheros que se suele tener que configurar, el fichero /etc/postfix/main.cf por lo general suele estar comentado densamente. Ten en cuenta que los siguientes ejemplos de entradas main.cf no son un main.cf completo.

```
# ¿cuál es el nombre de la máquina?
myhostname = correo.ejemplo.org

# ¿cuál es el nombre de dominio?
mydomain = ejemplo.org

# ¿cómo etiqueto el "from" del correo?
myorigin = $mydomain

# ¿en qué interfaces lo ejecuto? Por lo general, en todas.
inet_interfaces = all

# un fichero que contiene una lista de nombres de hosts y nombres de
# dominio cualificados desde los cuales recibo correo,
# generalmente están listados así:
# mydestination = localhost, $myhostname, etc
# pero prefiero mantener el listado en un fichero.
mydestination = /etc/postfix/mydestination

# mapa de nombres de usuarios entrantes. "man 5 virtual"
virtual_maps = hash:/etc/postfix/virtual

# mapeo de alias (como /etc/aliases en sendmail), "man 5 aliases"
alias_maps = hash:/etc/postfix/aliases

# base de datos de alias, se pueden tener diferentes configuraciones.
# "man 5 aliases"
alias_database = hash:/etc/postfix/aliases

# dónde repartir el correo, formato Mailbox o Maildir
# (el tradicional /var/spool/mail).
home_mailbox = Maildir/

# dónde guardar el correo, generalmente en /var/spool/mail/ pero se
# puede cambiar con facilidad
```

```
mail_spool_directory = /var/spool/mail

# ¿qué comando utilizamos para repartir el correo? /usr/bin/procmail
# es el comando por defecto, pero yo utilizo scanmail que es el sello
# del software antivirus AMaViS

mailbox_command = /usr/sbin/scanmails

# para quién retransmito el correo, de nuevo se pueden listarlos o
# guardarlos en un fichero (uno por línea).

relay_domains = /etc/postfix/relaydomains

# lista de redes locales (por defecto se retransmite el correo de
# estos hosts).

mynetworks = 10.0.0.0/24, 127.0.0.0/8

# ¿qué se le muestra a la gente que conecte al puerto 25? Por defecto
# muestra el número de versión, lo cual yo no hago.

smtpd_banner = $myhostname ESMTP $mail_name
```

En términos generales, cualquier fichero que simplemente liste un elemento por línea (como /etc/postfix/mydestination o /etc/postfix/relaydomains) se suelen almacenar como simple texto llano. Los ficheros que contienen mapeados (p. ej. alias, donde se tienen entradas como "root: cualquierusuario") deberían transformarse en ficheros hash de base de datos por velocidad (se puede especificar el tipo de fichero como hash, dbm, etc.).

Al igual que la mayoría de productos de IBM, Postfix tiene una licencia bastante curiosa, pero parece que la mayoría es código abierto y libre. Postfix se encuentra disponible en: <http://www.postfix.org/>. Se pueden conseguir los rpm's de postfix en: <ftp://contrib.Redhat.com/>, y aparentemente SuSE ahora viene con Postfix.

Sendmail Pro

Sendmail Pro es una versión comercial de Sendmail con soporte, y se encuentra disponible en: <http://www.sendmail.com/>. No me ha sido posible conseguir una demo o encontrar a alguien que lo utilice, de modo que no estoy seguro al 100% de lo cercano que se encuentra al Sendmail "original", aunque la compañía me ha dicho que utiliza el mismo código de base.

Zmailer

Zmailer es un gestor de correo GPL disponible en: <http://www.zmailer.org/>. Tiene ganchos criptográficos y por lo general parece bien construido.

DMail

DMail es un servidor de correo comercial, y no es código abierto. Se puede descargar una versión de prueba de: http://netwinsite.com/dmail_first.htm

nullmailer

El nullmailer envía correo a hosts inteligentes (relays) de forma que la máquina local no tenga que ejecutar ningún software de servidor. Está en: <http://em.ca/~bruceg/nullmailer/>.

MasqMail

El MasqMail manda el correo a una cola mientras está offline y después lo envía cuando conectas a tu ISP. Se puede configurar para múltiples ISP's, con dirección de respuesta, etc. Se puede descargar en: <http://merlin.uni-sw.gwdg.de/~okurth/masqmail/>

Dynamic Relay Authorization Control

El Dynamic Relay Authorization Control (DRAC) se une a tu servidor POP/IMAP para garantizar temporalmente acceso de reenvío SMTP a los hosts que se han autenticado con éxito y recogen correo (asumiendo que estos hosts enviarán correo, y no van a abusar de este privilegio. Se puede conseguir en: <http://mail.cc.umanitoba.ca/drac/index.html>.

POP

WU IMAPD (popd original)

POP e IMAP están relacionados pero son muy diferentes, de modo que los he separado aparte. POP significa "Protocolo de Oficina de Correos, Post Office Protocol" y simplemente te permite listar mensajes, recibirlos y borrarlos. Existen muchos servidores POP disponibles para Linux, el original que viene con la mayoría de las distribuciones suele ser adecuado para la mayoría de los usuarios. Los problemas principales con POP son similares a los de muchos otros protocolos; los nombres de usuarios y sus contraseñas se transmiten en texto claro, haciendo de ello un buen objetivo para un sniffer de paquetes. El POP se puede "SSLificar", sin embargo no todos los clientes de correo soportan POP seguro mediante SSL. La mayoría de los servidores POP vienen configurados para utilizar TCP_WRAPPERS, que es un método excelente de restringir el acceso. Para más información, ver anteriormente la sección sobre TCP_WRAPPERS. POP se ejecuta como root (ya que tiene que acceder a buzones) y en el pasado se han detectado varios ataques de root a varios servidores POP. POP se ejecuta en el puerto 109 y 110 (aunque el 109 está obsoleto), utilizando el protocolo tcp. El servidor IMAPD de la Universidad de Washington también viene con un servidor POP y en general se trata del servidor POP standard que viene con la mayoría de distribuciones Linux. Se puede conseguir de: <http://www.washington.edu/imap/>.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 110
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 110
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 110
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 110
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 110
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 110
```

Cyrus

Cyrus es un servidor imap (también soporta pop y kpop) dirigido a entornos "cerrados". Es decir, que los usuarios no tendrán ningún acceso al servidor de correo más que mediante los protocolos imap o pop. Esto le permite a Cyrus almacenar el correo de forma más segura, y permite una gestión más sencilla en instalaciones más grandes. Cyrus no tiene licencia GNU, pero es relativamente "libre", y está disponible en: <http://andrew2.andrew.cmu.edu/cyrus/imapd/>. También existe un conjunto de herramientas añadidas a Cyrus disponible en: <ftp://ftp.hr.vc-graz.ac.at/cyrus-tools/>.

IDS POP

IDS (No apesta, "It Doesn't Suck") POP es un reemplazo más ligero del popd dirigido a instalaciones más pequeñas. Es GPL y se encuentra disponible en: <http://www.nodomainname.net/software/ids-pop/>.

Qpopper

Qpopper es freeware, está producido por Qualcomm (los autores de Eudora). No lo recomendaría (el código fuente no está disponible). Se puede conseguir de: <http://eudora.qualcomm.com/freeware/qpop.html>.

IMAPD

WU IMAPD (imapd original)

IMAP es un POP con esteroides. Permite mantener con facilidad múltiples cuentas, permitir a múltiples personas acceso a una cuenta, dejar correo en el servidor, simplemente descargar los encabezados, o los cuerpos sin attachments, etc. IMAP es ideal para cualquiera o con serias necesidades de correo. Los servidores POP e IMAP que traen la mayoría de las distribuciones (empaquetados en un único paquete llamado `imapd`, por extraño que parezca) cubren la mayoría de las necesidades.

IMAP también se ejecuta como `root`, aunque `imapd` suele tener el privilegio del usuario que está accediendo, y no se puede configurar con facilidad para que se ejecute como usuario no-`root`, puesto que tienen buzones abiertos (y en el caso de IMAP, crea carpetas, ficheros, etc. en el directorio personal del usuario), de modo que no se pueden quitar los privilegios tan pronto como alguien quisiera. Ni se puede hacer `chroot` con facilidad (IMAP necesita tener acceso a `/var/spool/mail`, e IMAP necesita acceder al directorio personal del usuario). La mejor política es tener el software actualizado. Y si es posible, filtrar `pop` e `imapd` con el cortafuegos, lo cual funciona bien no hay nadie de gira que necesite recoger su correo vía Internet. El IMAP de la Universidad de Washington (WU) se encuentra disponible en: <http://www.washington.edu/imap/>.

IMAP se ejecuta en el puerto 143 y la mayoría de los servidores IMAPD soportan `TCP_WRAPPERS`, lo cual le hace relativamente sencillo de bloquear.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 143
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 143
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 143
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 143
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 143
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 143
```

Cyrus

Lectores de correo basados en WWW

Una de las mejores soluciones es utilizar un cliente basado en `www`, los cuales se pueden ejecutar sobre un servidor de `www` seguro con un mínimo trabajo extra, y añadir la opción de dejar que los usuarios comprueben el correo con seguridad desde lugares desde los que se les haría difícil comprobar su correo (mientras se está de vacaciones por Europa, por ejemplo). Por desgracia, la mayoría de los lectores de correo basados en `www` apestan, y los buenos cuestan un ojo de la cara.

No comerciales

IMP

IMP necesita el módulo Horde (disponible en el mismo sitio) y un servidor de correo con soporte para PHP3. Se puede descargar IMP y Horde desde:

<http://www.horde.org/imp/>

AtDot

AtDot tiene licencia GNU y está escrito en Perl. Tiene varios modos de operación, lo cual le hace útil para una gran variedad de soluciones de correo (proveedores al estilo de hotmail, PSI's, etc.). Se puede descargar de: <http://www.nodomainname.net/software/atdot/>.

acmemail

<http://www.nodomainname.net/software/atdot/>

IMHO

<http://www.lysator.liu.se/~stewa/IMHO/>

Comerciales

DmailWeb

<http://netwinsite.com/dmailweb/index.htm>

WebImap

<http://netwinsite.com/webimap/index.htm>

Coconut WebMail Pro

<http://www.coconutsoftware.com/>

DNS

Bind

DNS es un servicio extremadamente importante para redes IP. No dudaría en decir que probablemente sea el servicio MÁS importante (sin él, nadie puede encontrar nada más). También requiere conexiones provenientes del mundo exterior, y debido a la naturaleza y estructura del DNS, la información que los servidores de DNS dicen tener puede no ser cierta. El principal proveedor de software DNS (named, el standard de facto) actualmente está buscando la forma de añadir información de autenticación (utilizando RSA para firmar criptográficamente los datos, probando que es "cierto"). Si se tiene planeado administrar servidores DNS, yo diría que es de obligada lectura "DNS & BIND", de O'Reilly and Associates.

La mayoría de las distribuciones vienen con bind 8.x, sin embargo ninguna (según mis conocimientos) lo trae configurado para no-root, utilizan chroot por defecto. Sin embargo hacer el cambio es sencillo:

-u

especifica a qué UID cambiará bind una vez que esté vinculado al puerto 53 (me gusta utilizar un usuario llamado 'named' sin permisos de login, similar a 'nobody').

-g

especifica el directorio al que bind se hará chroot a sí mismo una vez que esté arrancado. /home/named es una buena apuesta, es en este directorio donde se deberían situar todas las librerías y ficheros de configuración que va a necesitar bind.

Una forma incluso más sencilla de ejecutar bind con chroot es descargar el paquete bind-chroot, disponible como paquete de contribución en la mayoría de las distribuciones, e instalarlo. Antes de la instalación, se necesitará un usuario y un grupo llamados named (al cual cambiará el servidor bind su UID/GID), simplemente utilizar groupadd y useradd para crear el usuario/grupo. Algunos paquetes utilizan holelogd para hacer un log de la información de bind en /var/log/messages (de igual forma que haría bind). Si no está disponible, habrá que instalarlo a mano, lo cual es una faena. Además de esto, el fichero de configuración por defecto de bind se suele configurar de forma segura (p. ej., no se puede hacer una petición a bind acerca de su versión).

Otro aspecto de bind es la información que contiene sobre tu(s) red(es). Cuando alguien hace una petición a un servidor DNS, por lo general envían una petición pequeña por cada información. Por ejemplo, ¿cuál es la dirección IP de www.seifried.org? Y existen transferencias de dominios, en las cuales un servidor DNS solicita toda la información disponible sobre, digamos, seifried.org, la recibe y después la pone disponible a otros (en el caso de un servidor DNS secundario). Esto es potencialmente peligroso, ya que puede ser tanto o más peligroso que enviar el número de teléfono de la compañía a cualquiera que llame y lo solicite. La versión 4 de Bind no se preocupaba demasiado sobre la seguridad, se podían limitar las transferencias a ciertos servidores, pero no de la forma lo suficientemente selectiva como para ser realmente útil. Esto ha cambiado en Bind 8, la documentación se encuentra disponible en <http://www.isc.org/bind.html>. Resumiendo, en Bind 8 existen configuraciones globales, la mayoría de las cuales se pueden aplicar basadas en el dominio. Se pueden restringir con facilidad las transferencias Y las peticiones, hacer log de las peticiones, configurar los tamaños máximos de los

datos, etcétera. Recuerda, cuando se está restringiendo las peticiones de zona, se deben asegurar TODOS los servidores de nombres (principal y secundarios), ya que se pueden transferir zonas desde un secundario con igual facilidad que desde el principal.

He aquí un fichero de configuración named.conf relativamente seguro (robado del paquete bind-chroot disponible en ftp.tux.org):

```
options {  
  
// Para este chroot se necesitan los siguientes paths  
  
directory "/var/named";  
  
dump-file "/var/tmp/named_dump.db"; // _PATH_DUMPFILE  
  
pid-file "/var/run/named.pid"; // _PATH_PIDFILE  
  
statistics-file "/var/tmp/named.stats"; // _PATH_STATS  
  
memstatistics-file "/var/tmp/named.memstats"; // _PATH_MEMSTATS  
  
// Fin de los paths necesarios  
  
check-names master warn; /* default. */  
  
datasize 20M;  
  
};  
  
zone "localhost" {  
  
type master;  
  
file "master/localhost";  
  
    check-names fail;  
  
    allow-update {  
  
        none;  
  
    };  
  
    allow-transfer {  
  
        any;  
  
    };  
  
};  
  
zone "0.0.127.in-addr.arpa" {  
  
type master;  
  
file "master/127.0.0";  
  
    allow-update {  
  
        none;  
  
    };
```

```

};

allow-transfer {
    any;
};

};

// Denegar y registrar peticiones de versión excepto desde localhost
zone "bind" chaos {
type master;

file "master/bind";

    allow-query {
        localhost;
    };
};

zone "." {
type hint;

file "named.zone";

};

zone "ejemplo.org" {
type master;

file "zones/ejemplo.org";

    allow-transfer {
        10.2.1.1;
        10.3.1.1;
    };
};

```

Bind se ejecuta en el puerto 53, utilizando udp y tcp, udp se utiliza para las peticiones normales de dominios (es ligero y rápido), tcp se utiliza para las transferencias de zonas y peticiones más grandes (como excavar www.microsoft.com). De modo que filtrar con el cortafuegos el tcp es relativamente seguro y eliminará cualquier transferencia de zonas, pero la petición ocasional al DNS podría no funcionar. Es mejor utilizar `named.conf` para controlar las transferencias de zonas:

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 53
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 53
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 53
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 53
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 53
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 53
```

lo cual bloquearía las transferencias de zonas y las peticiones grandes, lo siguiente bloquearía las peticiones normales (pero no las transferencias de zona, de modo que si se está bloqueándolo, recordar utilizar ambos conjuntos de reglas)

```
ipfwadm -I -a accept -P udp -S 10.0.0.0/8 -D 0.0.0.0/0 53
```

```
ipfwadm -I -a accept -P udp -S un.host.fiable -D 0.0.0.0/0 53
```

```
ipfwadm -I -a deny -P udp -S 0.0.0.0/0 -D 0.0.0.0/0 53
```

o

```
ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 53
```

```
ipchains -A input -p udp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 53
```

```
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 53
```

Dents

Dents es un servidor DNS con licencia GPL, actualmente en fase de pruebas (versión 0.0.3). Dents se está escribiendo desde cero, con soporte para SQL, integración con SNMP, utilizando CORBA internamente. Todo en conjunto debería compensar a Bind, tengo planeado probarlo y evaluarlo, pero hasta entonces tendrás que intentarlo tú mismo. Dents se encuentra disponible en: <http://www.dents.org/>.

NNTP

INN

El servidor de usenet INN ha tenido una larga y variopinta historia, durante un gran periodo no salían versiones oficiales, y parecía estar en el limbo. Sin embargo, parece que ha vuelto para bien. El software de servidor es el responsable de mantener una carga potencial enorme, si se coge una bandeja de noticias completa, el servidor tiene que procesar varios cientos de artículos por segundo, varios kilobytes de tamaño. Tiene que indexarlos, escribirlos a disco y entregárselos a los clientes que los han solicitado. INN es relativamente seguro en sí mismo, puesto que maneja datos dentro de un directorio y por lo general no tiene acceso más allá de ahí, sin embargo, hay que tener cuidado, al igual que con cualquier sistema de mensajería que se utilice para mantener material privado/confidencial. En la actualidad, INN lo mantiene el ISC, y se encuentra disponible en: <http://www.isc.org/inn.html>

Una de las principales amenazas de seguridad de INN es el consumo de recursos del servidor. Si alguien decide hacer un flood al servidor con artículos sin sentido o hay un repentino aumento de la actividad, se podrían tener problemas si escasea la capacidad. INN ha tenido diferentes agujeros de seguridad en el pasado, pero en el entorno actual, parece que los programadores han sabido cazarlos y eliminarlos todos (recientemente no ha surgido ninguno). Es altamente recomendable (por más motivos que los meramente de seguridad) ubicar el spool de noticias en un disco aparte, de forma que no pueda hacer caer al servidor.

En cuanto al acceso, definitivamente no se debería permitir acceso público. Cualquier servidor de noticias que esté accesible públicamente va a ser inmediatamente utilizado por la gente para leer noticias, enviar spam y asuntos parecidos. Restringe la lectura de noticias a tus clientes/red interna y si se está realmente preocupado, fuerza a la gente a hacer login. El acceso de clientes a INN se controla con el fichero `nntp.access`. Se pueden especificar direcciones IP, nombres de dominios (como `*.yo.com`), al igual que existen niveles de acceso (leer y publicar), los grupos de noticias a los que se tiene o no acceso, y también se puede especificar un nombre de usuario y contraseña. Sin embargo, puesto que la contraseña está enlazada con el host/dominio, es algo lioso.

Ejemplo de fichero `nntp.access`:

```
:: -no - : -no- :!*  
  
# deniega acceso desde todos los sitios, para todas las acciones  
# (publicar y leer), a todos los grupos.  
.yo.com::Read Post:::  
  
# los hosts de yo.com tienen acceso total a todos los grupos  
.ellos.com::Read::*, !yo.*  
  
# los hosts de ellos.com tienen acceso de lectura a todo salvo  
# a la jerarquía yo.  
.aol.com:Read Post:minombre:miclave:
```

```
# darme acceso a la cuenta de AOL utilizando usuario y clave
```

Si se va a ejecutar un servidor de noticias, recomendaría encarecidamente la lectura del libro de O'Reilly "Managing Usenet". Usenet es parecida al Sendmail, toda una bestia para conseguir que funcione y esté contenta.

Las noticias se deberían filtrar con un cortafuegos, puesto que la mayoría de los servidores alojan un grupo interno y las conexiones se hacen en uno o dos sentidos de los siguientes flujos:

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 119
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 119
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 119
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 119
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 119
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 119
```

Diablo

Diablo es un software gratuito dirigido al transporte de la backbone, es decir, aceptar artículos desde otros servidores NNTP y alimentar con ellos a otros servidores, no está dirigido a usuarios finales para lectura o publicación. Se puede conseguir en: <http://apollo.backplane.com/diablo/>.

DNews

Servidor NNTP comercial para diferentes plataformas. Disponible en:

<http://netwinsite.com/dnews.htm>

Cyclone

Otro servidor NNTP comercial, al igual que Diablo. Disponible en:

<http://bcandid.com/>

Typhoon

Typhoon es un servidor NNTP comercial dirigido a usuarios finales, es decir, les permite publicar y leer artículos. Se puede conseguir en:

<http://bcandid.com/>

DHCPD

El DHCPD es algo que deberían utilizar todos los administradores de redes. Permite servir información a clientes en cuanto a configuraciones de red/etc, lo cual significa que la única configuración que se necesita implantar en el cliente es aquella por defecto, y encender la máquina. También permite reconfigurar máquinas cliente con facilidad (es decir, pasar de utilizar 10.0.1.0 a 10.0.2.0, o un conjunto nuevo de servidores DNS). A la larga (e incluso a la corta) DHCP ahorra un montón de trabajo, dinero y stress. Yo lo ejecuto en casa con sólo 8 máquinas clientes y he descubierto que la vida es mucho más fácil. DHCPD lo mantiene el ISC, y se encuentra en:
<http://www.isc.org/dhcp.html>

También recomendaría ejecutar la versión 2.x (la versión 3.x está en pruebas), tiene muchas características nuevas, es más fácil de configurar y de trabajar con ella. Sin embargo, la ultimísima versión tiende a ser algo más neurótica, ten en cuenta que es software en fase beta. Definitivamente hay que filtrar el DHCPD de Internet. El tráfico DHCP sólo debería existir en segmentos locales, posiblemente reenviado a un servidor DHCP en otro segmento, pero el único tráfico DHCP que se vería proveniente de Internet sería un ataque/DOS (podrían reservar todas tus IP's, dejando secos a los auténticos clientes). Si estás reenviando el tráfico DHCP a Internet, NO lo hagas. Es una idea muy mala por multitud de motivos (primero por rendimiento / consistencia, pero también por seguridad).

Recomiendo que el servidor DHCPD sea exclusivamente servidor de DHCP, bloqueado en alguna parte (si confías en DHCP para tu red y el servidor DHCP se cae, la red tiene un serio problema), permitiéndole hacer su trabajo en silencio. Si se necesita abarcar subredes (p. ej., se tienen múltiples segmentos ethernet, sólo uno de los cuales tiene un servidor DHCP físicamente conectado) utiliza un transmisor DHCP (NT viene con uno, el software DHCP de Linux tiene esta característica, etc.). También existen problemas conocidos con NT y DHCP, NT RAS tiene la mala costumbre de consumir direcciones IP como un loco (he visto a un servidor NT coger 64 y quedárselas indefinidamente), debido a que está intentando reservar IP's para los clientes a que vayan a hacer un dial in. Esto no tiene porqué ser problema, pero puede (y lo ha hecho) conducir al agotamiento de recursos (en concreto, la cola de direcciones IP se puede terminar). O bien se apaga el RAS del NT o se le coloca en su propia subred, la dirección MAC que envía al servidor DHCP es muy extraña (y deletrea RAS en los primeros bytes) y no es fácil de mapear.

Haciendo Chroot al DHCPD

El DHCPD consiste en 2 ejecutables principales:

- * dhcpd - el DHCP
- * dhcrelay - un transmisor DHCP (para transmitir peticiones a un servidor DHCP central, puesto que el DHCP está basado en broadcasts, las cuales por lo general no se extienden (o no deberían) a routers.

DHCPD requiere 2 librerías:

- * /lib/ld-linux.so.2
- * /lib/libc.so.6

Un fichero de configuración:

* /etc/dhcpd.conf - información de configuración, situación de los ficheros de arranque, etc.

Y varios otros ficheros:

* /etc/dhcpd.leases - una lista de las conexiones activas

* un fichero de inicio, se puede modificar el que viene o hacerte el tuyo propio

La forma más sencilla de configurar el dhcpd con chroot es sencillamente instalar el dhcpd (preferiblemente la última versión) y mover/editar los ficheros necesarios. Una buena idea es crear un directorio (como /chroot/dhcpd/), preferiblemente en un sistema de ficheros separado de /, /usr, etc (enlaces simbólicos...), y después crear una estructura de ficheros por debajo para dhcpd. Lo siguiente es un ejemplo, simplemente reemplaza /chroot/dhcpd/ por tu elección. Por supuesto que es necesario ejecutar estos pasos como root para que funcione.

```
# Se instala bind para tener los ficheros apropiados
```

```
#
```

```
rpm -i dhcpd-2.0b1p10-1.i386.rpm
```

```
#
```

```
# Se crea la estructura de directorios
```

```
#
```

```
cd /chroot/dhcpd/ # o dondequiera que esté
```

```
mkdir ./etc
```

```
mkdir ./usr/sbin
```

```
mkdir ./usr
```

```
mkdir ./var/dhcpd
```

```
mkdir ./var
```

```
mkdir ./lib
```

```
#
```

```
# Se empiezan a llenar los ficheros
```

```
#
```

```
cp /usr/sbin/dhcpd ./usr/sbin/dhcpd
```

```
cp /etc/dhcpd.conf ./etc/dhcpd.conf
```

```
cp /etc/rc.d/init.d/dhcpd ./etc/dhcpd.init
```

```
cp /etc/rc.d/init.d/functions ./etc/functions
```

```
#
```

```

# Para conseguir las últimas librerías, cambiar lo oportuno

#

cp /lib/ld-linux.ld-linux.so.2 ./lib/

cp /lib/libc.so.6 ./lib/

#

# Y se crean los enlaces simbólicos necesarios para que funcione
# Recuerda que el dhcpd piensa que /chroot/dhcpd/ es /, de modo
# que utiliza enlaces relativos

```

Después modifica o crea tu script de inicio.

Una vez que se ha hecho esto, simplemente hay que borrar el fichero de inicio original y crear un enlace simbólico desde donde apuntaba al nuevo, y el dhcpd se portará con "normalidad" (es decir, se iniciará automáticamente al arrancar), mientras que en realidad se encuentra separado de tu sistema. Quizás también quieras eliminar los ficheros DHCPD originales que anden por ahí, aunque no es necesario.

Si se ha hecho lo anterior correctamente, deberías tener un /chroot/dhcpd/ (u otro directorio si se especifica algo diferente) que contenga todo lo requerido para ejecutar dhcpd. Un ps -xau debería mostrar algo así:

```

USER PID %CPU %MEM SIZE RSS TTY STAT START TIME COMMAND
root 6872 0.0 1.7 900 532 p0 S 02:32 0:00 ./usr/sbin/dhcpd -d -q
root 6873 0.0 0.9 736 288 p0 S 02:32 0:00 tee ./etc/dhcpd.log

```

Definitivamente DHCPD debería de filtrarse con el cortafuegos de hosts externos, pues no hay ninguna razón para que un host externo lance una petición a tu servidor DHCP en busca de IP's, además, tenerlo disponible al mundo exterior podría dar como resultado que el atacante consumiera los recursos de direcciones del servidor DHCP, suponiendo que se utilicen colas dinámicas de direcciones, se le podría acabar la suerte a tu red interna, y aprender acerca de la estructura de tu red interna. El DHCP se ejecuta en el puerto 67, con udp, porque las cantidades de datos involucradas son pequeñas, y la rapidez en la respuesta es crítica.

```

ipfwadm -I -a accept -P udp -S 10.0.0.0/8 -D 0.0.0.0/0 67
ipfwadm -I -a accept -P udp -S un.host.fiable -D 0.0.0.0/0 67
ipfwadm -I -a deny -P udp -S 0.0.0.0/0 -D 0.0.0.0/0 67

o

ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 67
ipchains -A input -p udp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 67
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 67

```


RSH, REXEC, RCP

Los servicios R como rsh, rcp, rexec, etc., son muy inseguros. Sencillamente no hay otra forma de decirlo. Sus fundamentos de seguridad se basan en la dirección hostname/IP de la máquina desde la que se conecta, la cual se puede falsificar con facilidad, utilizando técnicas como envenenamiento de DNS, comprometido de cualquier forma. Por defecto, no están todos deshabilitados, por favor, hazlo inmediatamente. Edita el fichero /etc/inetd.conf y busca rexec, rsh, etc., y coméntalos, seguido de un "killall -1 inetd" para reiniciar inetd.

Si es absolutamente necesario ejecutar estos servicios, utiliza los TCP_WRAPPERS para restringir el acceso, no es mucho pero ayudará. También asegúrate de que se filtran con el cortafuegos, puesto que los TCP_WRAPPERS le permiten a un atacante ver que se están ejecutando, lo cual puede dar como resultado un ataque falsificado, algo contra lo que los TCP_WRAPPERS no se pueden defender si se hace correctamente. El acceso a varios servicios R se controla vía ficheros rhosts, por lo general cada usuario tiene su propio fichero rhosts, por desgracia esto es susceptible a spoofing de paquetes. El problema con los servicios r también estriba en que existe una pequeña brecha que se puede utilizar para modificar ficheros, editar el fichero rhosts de un usuario (como el del root) es una sencilla forma de reventar ampliamente un sistema.

Si se necesitan herramientas de administración remotas que sean fáciles de usar y similares a rsh/etc, recomendaría utilizar nsh (Network SHell) o SSH, ambos soportan cifrado, y un grado de seguridad mucho más alto. Alternativamente, la utilización de software de VPN reducirá algunos de los riesgos, puesto que se puede denegar a los falsificadores de paquetes la oportunidad de comprometer tu(s) sistema(s) (parte del IPsec es la autenticación del emisor y la fuente, lo cual a veces es algo más importante que cifrar los datos).

Webmin

Webmin es una de las mejores herramientas de administración remota para Linux, escrita primariamente en Perl, fácil de usar y de instalar. Se puede asignar diferentes 'usuarios' (webmin guarda internamente nombres de usuarios y contraseñas) variando los niveles de acceso, por ejemplo, se podría asignar a paco acceso solamente para hacer un shutdown del servidor, y darle a juan acceso sólo para crear/borrar y manipular usuarios. Además de esto, funciona en la mayoría de plataformas de Linux y en una variedad de otras plataformas UNIX. El principal 'problema' que hay con webmin es su pobre documentación de uso en algunas partes, y el hecho de que el par nombre de usuario/contraseña se envía en texto claro sobre la red (lo cual se minimiza ligeramente por la posibilidad de permitir el acceso sólo a ciertos hosts y redes). Lo más importante es que hace el sistema accesible a personas no técnicas, que tienen que administrar sistemas de tal forma que no tengas que proporcionarles cuentas reales en el servidor. Webmin está disponible en: <http://www.webmin.com/webmin/> y actualmente es gratuito. Por defecto, Webmin corre en el puerto 10000 y se debería filtrar con el cortafuegos:

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 10000
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 10000
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 10000
```

o en ipchains:

```
ipchains -A input -p all -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 10000
```

```
ipchains -A input -p all -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 10000
```

```
ipchains -A input -p all -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 10000
```

NFSD

NFS significa Sistema de Ficheros de Red, Network File System, y es sencillamente eso, una buena forma de distribuir sistemas de ficheros, de sólo lectura y lectura/escritura, mientras se conserva un grado de seguridad y de control suponiendo que la red está cerrada y es segura. El NFS se pensó para ser utilizado en entornos con un gran ancho de banda (p. ej., en una LAN), en la cual los riesgos de seguridad no son altos, o la información que se comparte no es sensible (p. ej., una LAN pequeña y fiable detrás de un cortafuegos intercambiando diagramas de CAD/CAM, o un gran laboratorio de una universidad, que utilice el nfs para montar el /usr/. Si se necesita un alto nivel de seguridad, tal como el cifrado de datos entre hosts, el NFS no es la mejor elección. Personalmente lo utilizo dentro de mi LAN interna (la máquina tiene 2 interfaces, adivina cuál es la que tiene los filtros del cortafuegos más severos), para compartir sistemas de ficheros que contienen rpm's, este sitio web, etc. Alternativas más seguras incluyen el SAMBA (gratuito) y ahora IBM está portando el AFS a Linux (costoso, pero el AFS es un buen trozo de código).

El NFS tiene algunos controles de seguridad rudimentarios. El primero sería el filtrado mediante cortafuegos; en cualquier caso, utilizar NFS a través de una red grande, lenta y pública como Internet no es una buena idea, de modo que hay que filtrar el puerto 2049, UDP. Puesto que el NFS se ejecuta como un conjunto de demonios, los TCP_WRAPPERS no son útiles, a menos que se haya compilado el NFS para soportarlos. El fichero de configuración del NFS tiene unas cuantas directivas, un montón de las cuales tratan de las configuraciones de id del usuario y grupo (mapear todo el mundo a nobody, quizás mapear todos los motores clientes a "motor", etc, etc.) pero no existen mecanismos de autenticación reales (que tu cliente diga tener UID 0, que es por lo que el id del root se limita por defecto a nobody). Las exportaciones del NFS de sólo lectura son bastante seguras, sólo hay que preocuparse de que la gente equivocada sea la que le eche un vistazo a tu información (si ésta es sensible) y/o creen ataques de negación de servicio (pongamos que tienes un directorio legible por el mundo/etc para compartir los fuentes del kernel, y algún tipejo empieza a succionar datos como un loco...).

Las exportaciones de escritura son harina de otro costal, y se deberían utilizar con extrema precaución, puesto que la única "autenticación" está basada en IP/nombre de host (ambas fácilmente sujetas a spoofing), y UID (tú mismo puedes ejecutar Linux y ser UID 0). Viene un cliente con un ataque DOS, toma su IP, monta el compartido con permiso de escritura y se va para casa. Te dices "pero tendría que haber conocido la IP y el UID", amigos, el sniffing de paquetes no es una ciencia aeroespacial, ni lo es el "showmount".

De modo que, ¿cómo se asegura el NFS? Lo primero que hay que hacer es filtrarlo con el cortafuegos, especialmente si la máquina es del tipo multi-homed, con un interfaz conectado a una red públicamente accesible (Internet, el laboratorio de estudiantes, etc.). Si se está pensando en ejecutar NFS sobre una red pública, es mejor que sea de sólo lectura, y definitivamente se estará mejor con un producto diferente a NFS.

Lo segundo y la parte más interesante es el fichero /etc/exports. Controla qué es lo que se les permite hacer a los clientes, y cómo lo hacen.

Un fichero exports de ejemplo:

```
# Permitir a una estación de trabajo editar el contenido web
/www 10.0.0.11(rw,no_root_squash)
```

```
#  
  
# Otro compartido que permita a un usuario editar un sitio web
```

```
/www/www.bobo.org 10.0.0.202(rw,no_root_squash)
```

```
#  
  
# directorio ftp público
```

```
/home/ftp *.ejemplo.org(ro,all_squash)
```

La estructura del fichero exports es bastante simple, el directorio que se quiere exportar, el cliente (utiliza siempre IP's, los nombres de hosts se pueden falsear), y cualquier opción. El cliente puede tener una única IP (10.0.0.1), nombre de host (tipejo.poncho.net), una subred (10.0.0.0/255.255.255.0), o un wildcard (*.abuelito.mil). Algunas de las directivas más interesantes (y útiles) del fichero de configuración son: secure - la sesión nfs se debe originar desde un puerto privilegiado, es decir, el root TIENE que ser el que esté intentando montar el directorio. Esto es útil si el servidor que se está exportando también está asegurado.

ro - uno bueno, Sólo Lectura, ya se ha hablado lo suficiente.

noaccess - utilizado para cortar el acceso, p. ej. exportar /home/ pero poner como noaccess el /home/root

root_squash - limita el UID del root a la UID/GID del usuario anónimo (normalmente "nobody"), muy útil si se están exportando los directorios a servidores de administradores en los que no se confía al 100% (el root casi siempre puede leer cualquier fichero...PISTA)

no_root_squash - útil si se quiere perder el tiempo en directorios exportados como root para arreglar cosas (como los permisos del site www)

squash_uids y squash_gids - limitar ciertos UID(s) o GID(s) a los del usuario anónimo, en Red Hat un buen ejemplo sería 500-10000 (por defecto, Red Hat comienza a añadir usuarios y grupos en el 500), permitiendo a cualquier usuario con UID's más bajas (p. ej. cuentas especiales) acceder a cosas en especial.

all_squash - uno bueno, todos los privilegios se eliminan y todo el mundo es guest.

anonuid y anongid - configuran específicamente el UID / GID del usuario anónimo (quizás se quiera algo especial como "anonnfs").

En realidad, la página man exports es bastante buena.

Más allá de esto no hay mucho que asegurar en NFS, aparte de eliminarlo y sustituirlo por algún otro producto (como AFS, Coda, etc). El NFS es relativamente robusto, casi cualquier sabor de Unix lo soporta, y suele ser fácil de configurar, trabajar y mantener. También es "un viejo conocido", que ha estado entre nosotros durante mucho tiempo. Échale un vistazo a "Practical Unix and Internet Security", también ponen en negrita que no se utilice NFS si la seguridad es de especial interés.

El NFS se debería restringir al mundo exterior, se ejecuta en el puerto 2049, udp, al igual que utiliza RPC en el puerto 111, udp/tcp, y hace uso del mountd, que se ejecuta en el puerto 635, udp. Cambia el 2049 por el 111, y pon el 635 y tcp para asegurar esos servicios (de nuevo, la mejor idea es una regla en blanco para denegar los puertos 1 al 1024, o mejor aún, una política por

defecto de denegación).

```
ipfwadm -I -a accept -P udp -S 10.0.0.0/8 -D 0.0.0.0/0 2049
```

```
ipfwadm -I -a accept -P udp -S un.host.fiabile -D 0.0.0.0/0 2049
```

```
ipfwadm -I -a deny -P udp -S 0.0.0.0/0 -D 0.0.0.0/0 2049
```

o

```
ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 2049
```

```
ipchains -A input -p udp -j ACCEPT -s un.host.fiabile -d 0.0.0.0/0 2049
```

```
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 2049
```

tftp

El tftp (Protocolo Trivial de Transferencia de Ficheros, Trivial File Transfer Protocol) se utiliza en dispositivos que solicitan información desde un servidor de red, generalmente a la hora de arrancar. Es una forma extremadamente simple de ftp, con la mayoría de la seguridad y comandos avanzados eliminados, básicamente se utiliza casi en exclusiva por estaciones sin disco, datos de configuraciones de routers, y cualquier dispositivo que se arranque, y necesite información que no pueda almacenar permanentemente. Como tal, presenta un agujero de seguridad bastante grande, imagínate que alguien se conecta al servidor tftp y coge el fichero de arranque del router Cisco principal.

TFTP

El tftp por defecto se puede bloquear, acepta un nombre de directorio que está bastante limitado (muy similar al chroot), y se pueden utilizar TCP_WRAPPERS para limitar el acceso a ciertos hosts, pero si se quiere controlar el acceso a ficheros, habrá que utilizar utftp. Por defecto tftp (al menos en Red Hat) sólo permite acceso al directorio /tftpboot (el cual por lo general no suele existir, de modo que créalo si lo necesitas). Es una muy buena idea mantener el directorio tftp lo más separado del sistema que sea posible. Esto se consigue especificando el directorio o directorios a los cuales se quiere que tftp tenga acceso, después del comando tftp en el inetd.conf. El ejemplo que viene a continuación arranca normalmente el tftp y garantiza el acceso al directorio /tftpboot y al directorio /kickstart.

```
tftp dgram udp wait root /usr/sbin/tcpd in.ftpd /tftpboot  
  
/kickstart
```

Recuerda también que el tftp utiliza el UDP, de modo que un ps xau no mostrará necesariamente quién está conectado o qué está haciendo (al contrario de lo que muestra el ftp) a menos que se estén bajando un fichero (puesto que la mayoría de las aplicaciones tftp utilizan ficheros pequeños, es poco probable coger a alguien haciéndolo). El mejor desde el que monitorizar el tftp es desde el syslog, pero incluso así el tftp no guarda un log de las direcciones IP o algo realmente útil. Lo siguiente es una salida de ps , y syslog durante una sesión activa de tftp.

```
nobody 744 0.0 0.6 780 412 ? R 14:31 0:00 in.tftpd/tftpboot
```

```
Apr 21 14:31:15 hostname tftpd[744]: tftpd: trying to get file: testfile
```

```
Apr 21 14:31:15 hostname tftpd[744]: tftpd: serving file from /tftpboot
```

El TFTP se puede restringir con facilidad utilizando TCP_WRAPPERS y filtrándolo con el cortafuegos, tftp se ejecuta en el puerto 69, UDP, de modo que sólo hay que restringir el acceso al necesario por las diferentes estaciones de trabajo sin disco, los routers y similares. También es una buena idea bloquear todo el tráfico tftp en los perímetros de tu red, puesto que una máquina no necesita reiniciar remotamente utilizando tftp a través de Internet/etc. De igual forma, el tftp se ejecuta bajo el usuario nobody. Puesto que no se hace autenticación, y todos los dispositivos que acceden al servidor tftp lo están haciendo bajo "nobody", la seguridad a nivel de ficheros es bastante inútil. Resumiendo, un servidor muy inseguro de TFTP se ejecuta en el puerto 69, udp.

```
ipfwadm -I -a accept -P udp -S 10.0.0.0/8 -D 0.0.0.0/0 69
```

```
ipfwadm -I -a accept -P udp -S un.host.fiable -D 0.0.0.0/0 69
```

```
ipfwadm -I -a deny -P udp -S 0.0.0.0/0 -D 0.0.0.0/0 69
```

o

```
ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 69
```

```
ipchains -A input -p udp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 69
```

```
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 69
```

```
utftpd
```

El utftpd es un recambio seguro del tftpd por defecto, permite un control más afinado y soporta otras características interesantes (como control de revisión). También se puede basar el acceso según la IP de los clientes, lo cual significa que la configuración del router y de las estaciones de trabajo sin disco se pueden mantener aparte unas de otras. utftpd tiene licencia GPL y se encuentra disponible en: <http://www.nrw.net/uwe/utftpd.html>

BOOTP

El bootp es el precursor del dhcpd, tiene menos opciones y es menos configurable, pero en esencia hace las mismas tareas: ayuda a que los dispositivos se arranquen en la red, y les da la información que necesitan. No recomendaría ejecutar el bootp a menos que se tengan equipos antiguos que den conflicto con un servidor DHCP. Si alguien quiere que escriba algo más acerca de esto, que me escriba un email y lo haré. Si no, consideraré el bootp exclusivamente de interés histórico. Al igual que el DHCP, el bootp se ejecuta en el puerto 67, UDP.

```
ipfwadm -I -a accept -P udp -S 10.0.0.0/8 -D 0.0.0.0/0 67
```

```
ipfwadm -I -a accept -P udp -S un.host.fiable -D 0.0.0.0/0 67
```

```
ipfwadm -I -a deny -P udp -S 0.0.0.0/0 -D 0.0.0.0/0 67
```

o

```
ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 67
```

```
ipchains -A input -p udp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 67
```

```
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 67
```

SNMP

El SNMP (Protocolo Simple de Gestión de Red, Simple Network Management Protocol) se diseñó para permitir que sistemas heterogéneos y equivalentes hablasen entre sí, generasen informes y permitiesen modificaciones de sus configuraciones sobre una red TCP-IP. Por ejemplo, un dispositivo SNMP (como un router Cisco) se puede monitorizar/configurar desde un cliente SNMP, y se pueden escribir con sencillez scripts, para, digamos, alertarte si los paquetes denegados/segundo suben por encima de 20. Por desgracia, SNMP no trae seguridad incluida. El SNMPv1 se propuso originalmente en el RFC 1157 (Mayo de 1990) y la sección 8 (Consideraciones de Seguridad) dice: "En esta memoria no se discuten asuntos de seguridad". Creo que eso lo resume todo. En 1992/1993, salió el SNMPv2, y contenía consideraciones de seguridad, sin embargo éstas se eliminaron más tarde cuando demostraron ser totalmente rompibles. De modo que hoy hemos acabado con un SNMPv2 sin seguridad.

Actualmente, la única forma de proteger los dispositivos SNMP consiste en configurar el nombre de la comunidad por algo difícil de adivinar (pero es muy fácil hacer un sniffing del cable y encontrar el nombre), y filtrar con el cortafuegos el SNMP de modo que sólo los hosts que necesiten hablar entre sí puedan hacerlo (lo cual te deja abierto al spoofing). Los ataques al nombre de la comunidad mediante fuerza bruta son fáciles de llevar a cabo, y suelen ser efectivos, y existen varios métodos para monitorizar específicamente las transmisiones SNMP y reventar por completo una comunidad SNMP, es un mundo peligroso el que existe por ahí fuera.

Estos riesgos se pueden mitigar ligeramente dada la utilidad que tiene el SNMP, puesto que si está soportado e implementado de forma correcta, puede hacer significativamente más sencilla la administración de la red. Casi en cada implementación de SNMP, el nombre por defecto de la comunidad es "public" (en cuanto a Linux, NT, etc), hay que cambiarlo, por algo más oscuro (el nombre de la empresa es una mala idea). Una vez que alguien ha conseguido el nombre de la comunidad, puede llevar a cabo un "snmpwalk" y entrar en la red. El SNMP se ejecuta sobre UDP en los puertos 161 y 162; hay que bloquear esto en todas las entradas a la red (la backbone, el pool de acceso telefónico, etc.). Si un segmento de la red no tiene habilitados dispositivos SNMP o una consola SNMP, habría que bloquear el SNMP hacia y desde esa red. Esta es la única línea de defensa en cuanto a SNMP.

Por añadidura, el uso de IPsec (u otro software VPN) puede reducir considerablemente el riesgo de sniffing. Los RFC's del SNMPv3 se centran con intensidad en la seguridad (específicamente el RFC 2274, Ene 1998) de modo que queda algo de esperanza en el futuro. Si se están comprando productos SNMP, asegúrate de que soportan SNMPv3, pues sólo así tendrás oportunidades reales de seguridad.

Con el cu-snmpd no existen problemas específicos per-se, aparte de los problemas generales del SNMP anteriormente cubiertos. Las herramientas y utilidades cu-snmp sólo soportan SNMPv1 y SNMPv2, de modo que recuerda tener cuidado al utilizarlas sobre redes no fiables, puesto que la principal línea de seguridad (el nombre de la comunidad) quedará expuesta a que lo vea cualquiera.

```
ipfwadm -I -a accept -P udp -S 10.0.0.0/8 -D 0.0.0.0/0 161:162
```

```
ipfwadm -I -a accept -P udp -S un.host.fiable -D 0.0.0.0/0 161:162
```

```
ipfwadm -I -a deny -P udp -S 0.0.0.0/0 -D 0.0.0.0/0 161:162
```

o

```
ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 161:162
```

```
ipchains -A input -p udp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 161:162
```

```
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 161:162
```

Finger

El Finger es una de esas cosas que la mayoría de los administradores deshabilitan e ignoran. Es una herramienta útil según la ocasión, pero si se quiere permitir que otros administradores se hagan una idea de cuál de tus usuarios está intentando reventar el sistema, utiliza `identd`. El Finger revela demasiada información, y es la herramienta favorita para pruebas iniciales, y recopilación de datos sobre los objetivos. Existen otro tipo de ataques DOS, la mayoría consistentes en el envío de cientos de peticiones de finger y en ciertas configuraciones simplemente en observar el murmullo del servidor. Por favor, no ejecutes el finger. Muchas distribuciones vienen con él habilitado, pero citando el `inetd.conf` de Red Hat:

```
# Finger, systat y netstat proporcionan información que puede ser
# valiosa para potenciales "revienta sistemas". Muchos sitios eligen
# deshabilitar alguno de estos servicios para mejorar la seguridad.
```

Si todavía se tiene la sensación de que es absolutamente imprescindible utilizarlo, ejecútalo con `-u` para denegar las peticiones de tipo finger @host que sólo se utilizan para reunir información para futuros ataques. Deshabilita finger, de veras. Recientemente Fingerd también ha sido la causa de unos pocos ataques severos de denegación de servicio, especialmente si se ejecuta el NIS con grandes mapas, NO, repito NO ejecutes fingerd. El finger se ejecuta en el puerto 79, y el cfinger se ejecuta en el puerto 2003, ambos utilizan tcp.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 79
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 79
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 79
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 79
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 79
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 79
```

Cfingerd

Cfingerd (fingerd configurable) es un buen reemplazo del fingerd original, se construyó teniendo en cuenta la seguridad, normalmente se ejecuta como usuario `no-root`, y los usuarios lo pueden configurar con facilidad de modo que no se les pueda hacer un finger. Cfingerd se encuentra disponible en:
<http://ftp.bitgate.com/cfingerd/>

PFinger

El PFinger es parecido al Cfingerd en la forma en que supone un reemplazo seguro del fingerd original. Se puede conseguir en:
<http://www.xelia.ch/unix/pfinger/>

Identd

El servicio `identd` se utiliza para mapear usuarios/procesos a puertos en uso. Por ejemplo, la mayoría de los servidores `irc` intentan averiguar quién se está conectando a ellos haciendo una petición `identd`, lo cual consiste básicamente en preguntarle al servidor `identd` desde el ordenador cliente qué información tiene sobre un número de puerto, y la respuesta puede variar desde ninguna (si nadie está utilizando ese puerto en particular) a un nombre de usuario, un nombre de grupo, un `id` de proceso y otra información interesante. La configuración por defecto de la mayoría de las distribuciones es que el `identd` está activado (es elegante ejecutarlo, los servidores de `irc` y las versiones más recientes de `sendmail` comprueban las respuestas de `identd`), y sólo distribuirán el nombre de usuario. El uso principal del `identd` es permitir a los sistemas remotos algún tipo de forma de seguir la pista de los usuarios que se están conectando a sus servidores, `irc`, `telnet`, correo, u otros, por propósitos de autenticación (no es una buena idea, ya que es muy fácil de falsear). La universidad local de Edmonton requiere que se ejecute el `identd` si se quiere hacer un `telnet` a cualquiera de los servidores de shell, principalmente de forma que puedan seguir con rapidez la pista de las cuentas comprometidas.

Ejecutar el `identd` en tu máquina ayudará a otros administradores a la hora de hacer el seguimiento de problemas, puesto que no sólo consiguen la dirección IP y la hora del problema, sino que utilizando `identd` pueden averiguar el nombre del usuario. Esta forma es una espada de doble filo, mientras que proporciona información útil para seguir a usuarios maliciosos (definitivamente la gente a la que se quiere mantener alejada de los servidores) también se puede utilizar para conseguir información de los usuarios del sistema, lo cual dé como resultado que sus cuentas sean comprometidas. Ejecutar `identd` en los servidores sólo tiene sentido si están albergando cuentas de shell.

`Identd` soporta bastantes características, y se puede configurar con sencillez para que se ejecute como usuario `no-root`. Dependiendo de las políticas de seguridad, se puede querer o no dar mucha información, o se puede querer informar lo máximo posible. Simplemente activa la opción en `inetd.conf`, después en `in.identd` (las configuraciones por defecto son `-l -e -o`).

`-p port`

`-a address`

Se puede utilizar para especificar a qué puerto y en qué dirección se enlaza (en el caso de una máquina con IP's en forma de alias, o a múltiples interfaces), generalmente sólo es útil si se quiere que conecten máquinas internas, puesto que a las máquinas externas probablemente no les sea posible imaginarse a qué puerto se cambió.

`-u uid`

`-g gid`

Se utilizan para configurar el usuario y el grupo bajo el que el `identd` tendrá privilegios después de conectar al puerto, lo cual da como resultado el ser menos susceptible de comprometer la seguridad del sistema. En cuanto al manejo de la cantidad de información que proporciona:

`-o`

Especifica que `identd` no devuelva el tipo de sistema operativo, decir

simplemente "UNKNOWN" es una muy buena opción.

-n

Hará que identd devuelva números de usuarios (p.ej. su UID) y no el nombre de usuario, lo cual todavía les proporciona suficiente información para ti y para seguir la pista del usuario con facilidad, sin proporcionar valiosas pistas a posibles atacantes.

-N

Permite a los usuarios hacer crear un fichero ~/.noident , lo cual forzará que el identd devuelva "HIDDEN-USER" en lugar de información. Esto les permite a los usuarios la opción de tener un cierto grado de privacidad, pero un usuario malicioso lo utilizará para evadir la identificación.

-F format

Te permite especificar mucha más información que la standard, cualquier cosa desde el nombre y número del usuario hasta el PID real, nombre de comando, ¡y los argumentos dados! Esto sólo lo recomendaría para uso interno, puesto que es mucha información que los atacantes encontrarían útil.

En general, recomendaría utilizar el identd en servidores con cuentas de usuario shell, o si no deshabilitarlo, principalmente debido al número de ataques de denegación de servicio a los que es susceptible. Ejecutar identd le hará la vida mucho más fácil a otros usuarios a la hora de seguir pistas provenientes de tu sitio, algunas con mejoras de seguridad (no lo aseguro todavía pues aún no he podido comprobarlo):

<http://insecurity.net/> - El identd seguro de Paul, escrito en Perl

<http://www.ojnk.un/~odin/> - ojnk identd

<http://www.tildeslash.org/nullidentd.html> - null identd

<http://www.ajk.tele.fi/~too/sw/> - identd falso

<http://p8ur.op.het.net/midentd/> - midentd

<http://www.nyct.net/~defile/programs/ident2/> - ident2

Identd se ejecuta en el puerto 113 utilizando tcp, y por lo general sólo se necesitará si que quiere hacer IRC (muchas redes irc requieren una respuesta identd), o ser amable con los sistemas ejecutando demonios (tales como tcp_wrapped telnet, o sendmail) que hagan revisiones identd en las conexiones.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 113
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 113
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 113
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 113
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 113
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 113
```


NTPD

El NTP (Protocolo de Red de Tiempo, Network Time Protocol) es bastante simple en cuanto a su misión, mantiene sincronizados los relojes de los ordenadores. ¿Y qué? Intenta comparar ficheros de log desde 3 servidores separados si sus relojes están fuera de sincronismo por unos cuantos minutos. El NTP funciona simplemente mediante un cliente que se conecta a un servidor de tiempo, averiguando el retraso entre ellos (en una red de área local podría ser de sólo 1-2ms, a través de internet puede ser de varios cientos de ms), y después pregunta la hora y configura el reloj propio. Además, los servidores se pueden colocar en "cluster" para mantenerse sincronizados entre ellos, las posibilidades de que 3 o más servidores pierdan la pista de la hora que es (también llamado "deriva", "drift") es relativamente baja.

Habitualmente, la señal de tiempo se suele generar por un reloj atómico o una señal GPS, medida por un ordenador, estos son los servidores "stratum 1", más abajo se indican servidores de tiempo "stratum 2" que se encuentran generalmente abiertos al público, una compañía podría mantener sus propios servidores de tiempo "stratum 3" si es lo suficientemente necesario, etcétera.

Los datos que intercambia el NTP por supuesto que no son sensibles, es una señal de tiempo, sin embargo, si a un atacante le fuese posible interferirla, podrían ocurrir todo tipo de cosas desagradables: los ficheros de log se podrían volver inutilizables, las cuentas podrían expirar antes de tiempo, los trabajos del cron que hacen la copia de seguridad del servidor se podrían ejecutar en hora punta causando retrasos, etc. De modo que es una buena idea ejecutar tu propio servidor de tiempo y configurar el ajuste máximo que harán a sólo unos pocos segundos (en cualquier caso, tampoco deberían derivar mucho más). Si se es realmente paranoico, o se tiene un gran número de clientes, habría que considerar comprar una unidad de tiempo GPS.

Vienen de todo tipo de formas y tamaños, desde un rack de 1Unidad que se enchufa directamente a la LAN hasta tarjetas ISA o PCI que se conectan en un servidor y tienen una antena. Es una buena idea filtrar con el cortafuegos el servidor de tiempo, puesto que podrían darse ataques de negación de servicio en detrimento de la red. Además de esto, si es posible, utilizar el cifrado disponible en ntpd, basada en DES, suele ser suficiente para desalentar a la mayoría de atacantes. El NTP se encuentra disponible en:

<http://www.eecis.udel.edu/~ntp/>. Generalmente, ntpd o xntpd no suelen traer páginas de manual (genial, eh?) pero se puede encontrar documentación en /usr/doc/ntp-xxx/, o en:

http://www.eecis.udel.edu/~ntp/ntp_spool/html/index.htm. NTP se ejecuta en el puerto 123 utilizando udp y tcp, de modo que filtrarlo con el cortafuegos es relativamente sencillo:

```
ipfwadm -I -a accept -P udp -S 10.0.0.0/8 -D 0.0.0.0/0 123
```

```
ipfwadm -I -a accept -P udp -S un.host.fiable -D 0.0.0.0/0 123
```

```
ipfwadm -I -a deny -P udp -S 0.0.0.0/0 -D 0.0.0.0/0 123
```

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 123
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 123
```

o

```
ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 123
```

```
ipchains -A input -p udp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 123
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 123
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 123
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 123
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/123
```

CVS

CVS permite trabajar juntos a múltiples desarrolladores en proyectos con enorme cantidad de código fuente, y mantiene una gran base de código en alguna parte de forma limpia. Los mecanismos internos de seguridad del CVS son bastante simples por sí mismos; de hecho, si alguien dijera que son débiles, le tendría que dar la razón. La autenticación del CVS se suele conseguir a través de la red, utilizando pserver, los nombres de usuario se envían en texto claro, y las contraseñas tienen un hash trivial (en realidad no existe seguridad).

Para evitar esto, se cuenta con varias opciones buenas. En un entorno Unix, probablemente el método más simple sea utilizar SSH para pasar las conexiones por un túnel entre las máquinas clientes y el servidor. "Tim el PierdeTiempo" (Tim Hemel) ha escrito una excelente página ocupándose de esto, disponible en: <http://cuba.xs4all.nl/~tim/scvs/>. Una aproximación algo más complicada (pero a la larga mejor en cuanto a grandes instalaciones) es "kerberizar" el servidor CVS y los clientes.

Las redes grandes (especialmente en entornos universitarios) ya han establecido una infraestructura con Kerberos. Detalles en la "kerberización" del CVS disponibles en: <http://www.cyclic.com/cyclic-pages/security.html>. Aparte de que recomendaría encarecidamente filtrar con el cortafuegos el CVS, a menos que se esté utilizando para cualquier tipo de propósito público (como un proyecto de código abierto a través de Internet).

Otra herramienta que acaba de aparecer para asegurar CVS es "cvsd", un wrapper para pserver que hace chroot y/o suid el pserver al de un usuario no dañino. cvsd se encuentra disponible en: <http://cblack.mokey.com/cvsd/> en formato rpm y en tarball fuente.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 2401
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 2401
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 2401
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 2401
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 2401
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 2401
```

rsync

El rsync es un método extremadamente eficiente para hacer mirroring de ficheros, ya sea de ficheros de código desde un árbol CVS, un sitio web, o incluso este documento. rsync mantiene los permisos de ficheros, enlaces, hora de los ficheros y más. Además de esto, soporta modo anónimo (lo cual por cierto, utilizo para hacer el mirroring de este documento) lo cual hace la vida muy sencilla en lo concerniente. El programa rsync puede actuar por sí mismo como cliente (se ejecuta desde una línea de comandos o un script) y como servidor (generalmente se ejecuta desde inetd.conf). El programa en sí es bastante seguro: no requiere privilegios de root para ejecutarse como cliente ni como servidor (aunque se puede hacer si se quiere) y se puede hacer chroot a sí mismo en el directorio raíz o cualquiera que esté siendo hecho mirror (sin embargo esto requiere privilegios de root y puede ser más peligroso de lo que merezca la pena). También se puede mapear el id del usuario y el id del grupo con acceso al sistema (por defecto, es nobody para la mayoría de los paquetes de rsync precompilados y probablemente sea la mejor elección). En modo no-anónimo, rsync soporta nombres de usuarios y contraseñas que se cifran fuertemente utilizando MD4 de 128 bit. La página del manual "man rsyncd.conf" cubre claramente la configuración de un servidor rsync y lo hace relativamente seguro. El fichero de configuración por defecto es /etc/rsyncd.conf. Tiene una sección global y sección modular (básicamente cada directorio compartido es un módulo):

ejemplo de rsyncd.conf

```
motd file = /etc/rsync.motd # Especifica fichero a mostrar

etcmax connections = 5 # número máximo de conexiones, para evitar saturación

[pub-ftp]

comment = public ftp area # comentario simple

path = /home/ftp/pub # path al directorio exportado

read only = yes # hacerlo de sólo lectura, para directorios exportados

chroot = yes # chroot a /home/ftp/pub

uid = nobody # configurar explícitamente el UID

gid = nobody # configurar explícitamente el GID

[asuntos-secretos]

comment = mis asuntos secretos

path = /home/user/secretos # path a mis secretos

list = no # ocultar este módulo cuando pidan una lista

secrets file = /etc/rsync.users # fichero de contraseñas

auth users = me, bob, santa # lista de usuarios a los que se permite ver mis secretos

hosts allow = 1.1.1.1, 2.2.2.2 # lista de hosts permitidos
```

Como se puede ver, el rsync es bastante configurable, y generalmente es bastante seguro, excepción hecha de que las transiciones de ficheros no van cifradas de ninguna forma. Si se necesita seguridad, sugeriría utilizar el SSH para abrir una conexión mediante un túnel, o alguna solución VPN como FreeS/WAN. Igualmente asegúrate de estar ejecutando rsync 2.3.x o una versión más alta, ya que se encontró un compromiso de root en la 2.2.x. Rsync se encuentra disponible en: <http://rsync.samba.org/>. Rsync se ejecuta en el puerto 873, tcp.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 873
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 873
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 873
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 873
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 873
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 873
```

lpd

El lpd es la utilidad UNIX de impresión (Demonio de Impresión de Línea, Line Printer Daemon). Te permite entregar trabajos de impresión, ejecutarlos a través de filtros, gestionar las colas de impresión, etcétera. lpd puede aceptar trabajos locales de impresión, o sobre la red, y acceder a varias partes del sistema (impresoras, demonios de logging, etc), lo cual lo convierte en un potencial agujero de seguridad. Históricamente el lpd ha sido motivo de diferentes ataques root. Aunque estos bugs parecen haberse limado en su mayoría, todavía quedan muchos potenciales ataques de denegación de servicio, debido a su función (algo tan simple como mandar trabajos de impresión enormes y dejar sin papel a la impresora). Por suerte, con la llegada de las impresoras que tienen conocimiento de red, el lpd se está desfasando, sin embargo todavía existe una gran cantidad de impresión hecha vía lpd. El acceso a lpd se controla vía /etc/hosts.equiv y /etc/hosts.lpd. El lp se debería filtrar del mundo exterior con el cortafuegos. Si se necesita enviar trabajos de impresión a través de redes públicas, recuerda que cualquiera los puede leer, de modo que sería interesante una solución de VPN. El lpd se ejecuta en el puerto 515, utilizando tcp. El fichero hosts.lpd debería contener una lista de los hosts (estacion1.tudominio.org, etc), uno por línea, a los que se les permite utilizar los servicios lpd del servidor, de igual forma se podría utilizar ipfwadm/ipchains.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 515
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 515
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 515
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 515
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 515
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 515
```

LPRng

Una alternativa al lpd por defecto es "LPRng" (LPR Generación Siguiendo, Next Generation), que aporta nuevas mejoras y también soporta un más alto grado de seguridad. LPRng soporta Kerberos y autenticación basada en PGP, al igual que restricción de ficheros, /etc/lpd.perms, lo cual te permite controlar el acceso basado en usuario, grupo, autenticación, IP, etcétera, permitiendo configuraciones extremadamente flexibles y seguras. LPRng tiene una documentación excelente y se encuentra disponible en:
<http://www.astart.com/lprng/LPRng.html>

pdq

El pdq es otro reemplazo del LPD, no se hace especial hincapié en la seguridad mejorada, pero parece ofrecer algunas mejoras de gestión y rendimiento comparado con el LPD por defecto. Se puede conseguir en:
<http://feynman.tam.uiuc.edu/pdq/>

CUPS

Systema Común de Impresión Unix, Common UNIX Printing System (CUPS), tiene licencia GPL y actualmente se encuentra en fase beta. CUPS se encuentra

disponible en: <http://www.cups.org/>

Samba

El SAMBA es de lo mejorcito desde el pan de molde, claro, siempre y cuando se tenga que compartir ficheros e impresoras entre Windows y *NIX. También está bastante incomprendido, y sufre severamente de la interacción con varios clientes Window (a veces estropeados). El SAMBA tiene muchas cosas que lo hacen parecer sano, pero puede conducir a algo que a veces pueda parecer que esté estropeado. El SAMBA simplemente da acceso al sistema de ficheros vía SMB (Servidor de Bloque de Mensajes, Server Message Block), el protocolo que utiliza Windows para compartir ficheros e impresoras. Verifica el nombre de usuario y la contraseña que se le da (si es necesaria) y después da acceso a los ficheros según los permisos de los ficheros. Sólo me voy a ocupar del Samba 2.x, el Samba 1.x es bastante viejo y obsoleto.

El Samba 2.x se controla vía `smb.conf`, por lo general en `/etc` (`man smb.conf`). En `/etc/smb.conf` hay 4 zonas de configuración: `[globals]`, `[printers]`, `[homes]`, y cada `[sharename]` tiene su propia configuración (ya sea una impresora o un disco compartido). Hay algo así como un centenar de switches, la página del manual de `smb.conf` se ocupa de ellos en profundidad. Algunos de los importantes (en cuanto a seguridad) son:

`security = xxxx` donde `xxxx` es un compartido, servidor o dominio, la seguridad compartida se hace por compartimiento, con una contraseña que utiliza todo el mundo, el servidor significa que el servidor de samba autentifica él mismo a los usuarios, ya sea vía `/etc/password`, o `smbpasswd`. Si se configura por dominio, el samba autentifica al usuario vía controlador de dominio NT, integrándolo de esta elegante forma en la red NT existente (si se tiene una).

`guest account = xxxx` donde `xxxx` es el nombre de usuario de la cuenta a la que se quiere que mapear el usuario `guest`. Si se ha definido un compartido como público, entonces todas las peticiones que se le hagan las maneja este usuario.

`hosts allow = xxxx` donde `xxxx` es una lista separada por espacios de hosts / bloques IP a las que se permite conectar al servidor.

`hosts deny = xxxx` donde `xxxx` es una lista separada por espacios de hosts / bloques IP a las que no les está permitido conectar con el servidor.

`interfaces = xxxx` donde `xxxx` es una lista separada por espacios de bloques IP con los que enlazará samba.

El SMB utiliza una variedad de puertos, la mayoría de las veces haciendo uso de los puertos 137, 138 y 139, `udp` y `tcp` en todos salvo para el 139.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 137:139
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 137:139
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 137:139
```

```
ipfwadm -I -a accept -P udp -S 10.0.0.0/8 -D 0.0.0.0/0 137:139
```

```
ipfwadm -I -a accept -P udp -S un.host.fiable -D 0.0.0.0/0 137:139
```

```
ipfwadm -I -a deny -P udp -S 0.0.0.0/0 -D 0.0.0.0/0 137:139
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 137:139
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 137:139
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 137:139
```

```
ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 137:139
```

```
ipchains -A input -p udp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 137:139
```

```
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 137:139
```

También recomendaría la instalación y el uso de SWAT (Herramienta de Administración de Web de samba) puesto que recortará los errores que se tiendan a cometer. El Samba y el SWAT se encuentran disponibles en <http://www.samba.org/> y vienen con cada distribución.

SWAT

SWAT es una herramienta de administración muy interesante, para configurar smb.conf. El principal problema es que esto requiere utilizar la cuenta del root y su contraseña para acceder, y se ejecuta como un proceso separado de inetd.conf, de modo que no hay forma de cifrarlo, hasta donde yo sé no hay forma de asignar acceso administrativo a SWAT a otros usuarios. Habiendo dicho esto, sin embargo es una buena herramienta para corregir errores a la hora de editar el smb.conf. También se puede ejecutar SWAT con la opción -a, lo cual quiere decir que no se solicitará contraseña, y utilizar TCP_WRAPPERS para restringir el acceso a ciertas estaciones de trabajo (aunque todavía se seguiría siendo vulnerable a IP spoofing). En resumen, el SWAT no se pensó como una herramienta administrativa, pero es útil. El SWAT por lo general viene con el samba, y se encuentra disponible en: <http://www.samba.org/>, y hay una versión demo en línea en: <http://anu.samba.org/cgi-bin/swat/>

Servidores LDAP Linux

El protocolo de acceso a directorio ligero parece ser el futuro del almacenamiento de la información de usuario (contraseñas, directorios de usuarios, números de teléfonos, etc.) Muchos productos (ADS, NDS, etc.) soportan interfaces LDAP, haciendo importante para Linux soportar el LDAP, puesto que será necesario que se adapte a entornos empresariales.

Servidores LDAP

OpenLDAP

El OpenLDAP es un paquete completamente de código abierto (ten en cuenta que no es GPL) que proporciona servicios LDAP, servidores de réplicas y utilidades. Se puede conseguir en: <http://www.openldap.org>

Herramientas LDAP

ldap-client-cgi.py

El ldap-client-cgi.py es un programa en Python que se ejecuta como un cgi y que proporciona un interfaz www a un directorio LDAP. Se puede conseguir en: <http://sites.inka.de/ms/python/ldap-client-cgi/>

NSS LDAP Module

El NSS LDAP Module te permite hacer autenticación de usuarios vía LDAP. Se puede conseguir de: http://www.padl.com/nss_ldap.html

Sistema X Window

El sistema X Window proporciona un método transparente a red de compartir datos gráficos, o más específicamente de exportar el display de un programa a un host remoto (o local). Utilizándolo se puede ejecutar un potente paquete de renderizado 3D de una SGI origin 2000 y mostrarlos en un 486. En esencia, es el abuelito de todos estos "delgados clientes" que se están haciendo bastante populares hoy en la actualidad. Se creó en el MIT, durante una época en la que la seguridad no era un asunto que preocupase demasiado. Esto, por supuesto, ha llevado a más de un bug desagradable. Peor incluso, el nivel de control que se le ha dado a X (maneja pulsaciones de teclado, movimientos de ratón, dibuja la pantalla, etc.) significa que si se compromete pueden ocurrir cosas muy desagradables. Estos datos, si se envían sobre la red (p. ej., el programa X que se ejecuta se está mostrando en un host remoto) se pueden registrar en logs con facilidad, de modo que la información sensible (como un xterm que esté siendo utilizado para hacer login en otro sistema remoto) es vulnerable. Además de estos problemas, el protocolo de autenticación que utiliza X es relativamente débil (aunque se ha mejorado). Ejecutarlo en una sesión gráfica de xemacs en un servidor situado 3 zonas horarias más lejos puede ser algo bastante útil.

El X es bastante predecible en cuanto al uso de puertos, casi todas las implementaciones

e instalaciones de X utilizan el puerto 6000 para la primera sesión, e incrementan en uno

para otras sesiones, lo cual lo hace bastante sencillo de escanear. Si no se va a utilizar el

X para mostrar un programa ejecutándose en sistemas remotos, sugeriría que se filtrase

el puerto 6000 con el cortafuegos.

```
ipfwadm -I -a accept -P tcp -S 10.0.0.0/8 -D 0.0.0.0/0 6000:6100
```

```
ipfwadm -I -a accept -P tcp -S un.host.fiable -D 0.0.0.0/0 6000:6100
```

```
ipfwadm -I -a deny -P tcp -S 0.0.0.0/0 -D 0.0.0.0/0 6000:6100
```

o

```
ipchains -A input -p tcp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 6000:6100
```

```
ipchains -A input -p tcp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 6000:6100
```

```
ipchains -A input -p tcp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 6000:6100
```

El control sobre lo que está permitido ejecutar al servidor X se puede hacer de diferentes formas.

xhost

El xhost te permite especificar a qué máquinas se le permite o no conectar al servidor X,

es un mecanismo de seguridad bastante simplicista, y no es apropiado en

cualquier entorno moderno, sin embargo, si se utiliza en conjunción con otros mecanismos, puede ayudar. El comando es bastante simple: "xhost +nombredehost.com" añade nombredehost.com, "xhost -hostname.com" elimina nombredehost.com de la lista, es necesario especificar "xhost -" para activar la lista de control de accesos, o si no por defecto se le dejará entrar a cualquiera.

mkxauth

Definitivamente, el mkxauth es un paso adelante desde xhost. El mkxauth ayuda a crear ficheros ~/.Xauthority, y juntarlos, lo cual se utiliza para especificar nombres de hosts y las magic cookies relativas. Estas cookies se pueden utilizar para conseguir acceso a un host X remoto (en esencia, cada extremo tiene una copia de la cookie) y se transmiten bien en texto plano (inseguro) o cifrado con DES (bastante seguro). Utilizando este método se puede estar relativamente a salvo y seguro. Los ficheros Xauthority también se pueden utilizar en conjunción con Kerberos, eliminando la necesidad de copiar los ficheros Xauthority y manteniéndolos sincronizados. Los hosts se autentifican entre sí a través de un servidor central de llaves Kerberos de forma cifrada, este método es apropiado para la mayoría de grandes instalaciones/etc. El mkxauth tiene una página de manual excelente "man mkxauth" y se pueden obtener detalles generalizados disponibles en la página del manual del Xsecurity (no estoy seguro de lo común que es el nombre de esta página) "man Xsecurity".

[Guía de Seguridad del Administrador de Linux - GSAL]

Conectividad SNA

El SNA es un protocolo de red muy habitual que se remonta a los días de IBM y el "hierro pesado"

ICE Linux-SNA

<http://www.icenetworking.com/products/sna/fire/index.html>

[Guía de Seguridad del Administrador de Linux - GSAL]

Software de Autoridad de Certificación para Linux

OpenCA

Un proyecto para proporcionar un extenso conjunto de herramientas para una autoridad "recién estrenada", capaz de manejar certificados X.509. Se encuentra disponible en <http://www.openca.org>

pyCA

pyCA es una colección de scripts de software escritos en python para configurar una autoridad de certificación. Se puede descargar de:
<http://sites.inka.de/ms/python/pyca/>

El kernel de Linux

En realidad Linux (GNU/Linux según Stallman, si te estás refiriendo a una distribución Linux completa) tan sólo es el kernel del sistema operativo. El kernel es el corazón del sistema, maneja el acceso al disco duro, los mecanismos de seguridad, la red y casi todo. Más te vale que sea seguro o la has fastidiado.

Además de esto, hay problemas como el fallo F00F del Pentium, y problemas inherentes al protocolo TCP-IP, que el kernel de Linux ha podido eliminar. Las versiones del kernel van etiquetadas como X.Y.Z, siendo Z los números de revisiones menores, Y define si el kernel es una prueba (números impares) o de producción (número par), y X define si se trata de una revisión mayor (hasta ahora hemos tenido la 0, la 1 y la 2). Yo recomendaría encarecidamente ejecutar el kernel 2.2.x, puesto que a Julio de 1999 este es el 2.2.10. Las series del kernel 2.2.x tienen importantes mejoras sobre las series 2.0.x. Utilizar kernels 2.2.x también te permite tener acceso a nuevas características tales como ipchains (en lugar de ipfwadm) así como otras características de seguridad avanzadas. La serie 2.0.x ha quedado retirada oficialmente desde Junio de 1999.

Para saber cuál es la versión más reciente del kernel, simplemente haz un finger a @linux.kernel.org:

```
[seifried@mail kernel-patches]$ finger @linux.kernel.org[linux.kernel.org]
```

```
The latest stable version of the Linux kernel is: 2.2.12
```

```
The latest beta version of the Linux kernel is: 2.3.20
```

```
The latest prepatch (alpha) version *appears* to be: none
```

Actualización y Compilación del Kernel

La actualización del kernel consiste en conseguir un kernel y unos módulos nuevos, editando /etc/lilo.conf, volviendo a ejecutar lilo para escribir el nuevo MBR. El kernel se suele colocar en /boot, y los módulos van en /lib/modules/kernel.numero.version/.

Conseguir un kernel y módulos nuevos se puede hacer de dos formas diferentes, descargando el paquete apropiado del kernel e instalándolo, o descargando el código fuente de ftp://ftp.kernel.org/ (por favor, utiliza un mirror), y compilándolo.

Compilar un kernel va todo seguido:

```
cd /usr/src
```

Debería de haber un symlink (enlace simbólico) llamado "linux" apuntando al directorio que contiene el kernel actual, bórralo si está ahí, si no está, no hay problema. Quizás quieras hacer un "mv" del directorio de linux a /usr/src/linux-kernel.version.number y crear un enlace apuntando a /usr/src/linux

Desempaqueta el código fuente utilizando tar y gzip según sea apropiado, de modo que ahora tienes un /usr/src/linux con aproximadamente 50 megabytes de código fuente dentro. El siguiente paso es crear una configuración del kernel de linux (/usr/src/linux.config), lo cual se puede conseguir mediante "make config", "make menuconfig" o "make xconfig", mi método preferido es "make menuconfig" (para lo cual necesitarás ncurses y ls librerías de desarrollo de

ncurses). Este es discutiblemente el paso más difícil, hay cientos de opciones, que se pueden categorizar en dos áreas principales: soporte de hardware y soporte de servicios. En cuanto al soporte de hardware, haz una lista del hardware sobre el que se ejecutará el kernel (por ejemplo, P166, controladora Adaptec 2940 SCSI, tarjeta de red NE2000, etc.) y activa las opciones apropiadas. En cuanto al soporte de servicios, necesitarás imaginarte qué sistema de ficheros (fat, ext2, minix, etc) estás planeando utilizar, lo mismo en cuanto a la red (cortafuegos, etc.).

Una vez que hayas configurado el kernel, necesitas compilarlo, el siguiente comando crea las dependencias, asegurando que las librerías y demás se vayan construyendo en el orden apropiado, luego se limpia cualquier información de compilaciones anteriores, posteriormente se construye el kernel, los módulos y se instalan los módulos.

```
make dep (crea las dependencias)
```

```
make clean (elimina basura anterior)
```

```
make bzImage (make zImage casca si el kernel es demasiado grande, y los kernels 2.2.x tienden a ser bastante grandes)
```

```
make modules (crea todos los módulos que hayas especificado)
```

```
make modules_install (instala los módulos en /lib/modules/numero.version.kernel
```

Luego tienes que copiar /usr/src/linux/arch/i386/boot/bzImage (zImage) a /boot/vmlinuz-numero.version.kernel. Después edita /etc/lilo.conf, añadiendo una nueva entrada para el nuevo kernel y configurarlo como la imagen por defecto es la forma más segura (utilizando el comando default=x, si no arrancará el primer kernel de la lista), si falla puedes reiniciar y volver a la versión anterior del kernel que estaba funcionando.

```
boot=/dev/hda
```

```
map=/boot/map
```

```
install=/boot/boot.b
```

```
prompt
```

```
timeout=50
```

```
default=linux
```

```
image=/boot/vmlinuz-2.2.9
```

```
label=linux
```

```
root=/dev/hda1
```

```
read-only
```

```
image=/boot/vmlinuz-2.2.5
```

```
label=linuxviejo
```

```
root=/dev/hda1
```

```
read-only
```

Una vez que termines de editar /etc/lilo.conf tienes que ejecutar /sbin/lilo para sobrescribir el MBR (Master Boot Record, el sector de arranque). Cuando se ejecuta lilo, mostrará una salida similar a:

```
Added linux *
```

```
Added linuxviejo
```

Sacará una lista de las imágenes que estén cargadas en el MBR e indicará con un * cuál es la imagen por defecto (lo normal es que por defecto cargue la primera imagen de la lista, a menos que se le especifique otra usando la directiva default).

Versiones del Kernel

Actualmente la serie estable del kernel es la 2.2.x, y las series de desarrollo son las 2.3.x. Las series de desarrollo 2.1.x no son recomendadas, tienen muchos problemas e inconsistencias. La serie 2.0.x del kernel, aunque vieja y falta de algunas características, es relativamente sólida, pero por desgracia, la actualización de la 2.0.x a la 2.2.x es un paso bastante grande, recomendaría precaución. Hay que actualizar algunos paquetes de software, librerías, ppp, modutils y otras (viene descrito en los documentos del kernel / dependencias rpm, etc.). Puedes conservar el kernel antiguo, añadir una entrada en lilo.conf como "linuxviejo" o algo parecido, así podrás recuperarte con facilidad en caso de que el 2.2.x no funcionase como fuera de esperar. No pienses que la serie 2.2.x está libre de errores, a la 2.2.9 se le encontrarán fallos y quedará obsoleta, al igual que cualquier software del mundo.

Parches de seguridad del kernel y del compilador

Existe una variedad de parches a nivel de kernel que pueden mejorar la seguridad de un sistema linux. Algunos evitan exploits de desbordamiento de buffer, otros proporcionan fuerte criptografía.

Parches del Kernel

Secure Linux - patch del kernel

Este parche soluciona un buen número de elementos, y proporciona otro nivel de seguridad al sistema. Por desgracia sólo se encuentra disponible para la serie 2.0 del kernel. Se puede conseguir en:

<http://www.false.com/security/linux/index.html>

International Kernel Patch

Este patch (¡de más de un megabyte de tamaño!) añade una enorme cantidad de potente criptografía y elementos relacionados. Incluyo diferentes algoritmos de cifrado candidatos AES (incluyendo el MARS de IBM). Se puede conseguir en:

<http://www.kerneli.org/>

Parches del Compilador

Stackguard

El Stackguard es un conjunto de parches para GCC que compila programas, para evitar que escriban en direcciones de memoria que no deberían (explicación simplificada, para información detallada acudir a la página web de Stackgard). Sin embargo Stackguard recorta algo de funcionalidad, fallarán programas como gdb y otros debuggers, pero esto último no suele ser una gran preocupación en servidores de producción de alta seguridad. Se puede conseguir en:

<http://www.cse.ogi.edu/DISC/projects/immunix/StackGuard/>

Herramientas administrativas

Acceso Remoto

Telnet

Telnet es con mucho la herramienta remota más vieja y conocida, prácticamente cualquier Unix viene con ella, incluso lo soportan sistemas como NT. Telnet sólo tiene uso cuando puedas administrar el sistema desde modo comandos (algo para lo que NT no es tan bueno), lo cual lo convierte en perfecto para sistemas Unix. Telnet es increíblemente inseguro, las contraseñas y los nombres de usuarios, al igual que los datos de las sesiones vuelan en texto simple, siendo el objetivo preferido de los sniffers. Telnet viene con todas las distribuciones de Linux. No deberías utilizar nunca el telnet de fábrica para administrar remotamente un sistema.

SSL Telnet

SSL Telnet es telnet con el añadido de cifrado SSL, lo cual lo hace bastante más seguro. Usando certificados X.509 (también conocidos como certificados personales) se pueden administrar sistemas con facilidad. Al contrario de sistemas como el SSH, SSL Telnet es completamente GNU y gratis para cualquier uso. Puedes encontrar el servidor Telnet SSL en <ftp://ftp.replay.com>

SSH

SSH era gratis al principio, pero ahora está bajo licencia comercial, sin embargo tiene numerosas características que lo hacen merecer la pena. Soporta diferentes tipos de autenticación (contraseña, basada en rhosts, llaves RSA), permite redireccionar puertos, y se puede configurar fácilmente a qué usuarios se les permite usarlo. SSH está disponible en <ftp://ftp.replay.com>. Si vas a utilizarlo para uso comercial, o si quieres la última versión, dirígete a <http://www.ssh.fi/>

LSH

LSH es una implementación gratuita del protocolo SSH, LSH tiene licencia GNU y está empezando a perfilarse como la alternativa (comercialmente hablando) a SSH (que ya no es gratis). Lo puedes descargar de <http://www.net.lut.ac.uk/psst/>, pero ten en cuenta que está bajo desarrollo.

REXEC

REXEC es una de las utilidades UNIX más antiguas, te permite ejecutar comandos en un sistema remoto, aunque tiene el serio fallo de no tener un modelo de seguridad real. La seguridad se consigue mediante el uso de ficheros 'rhosts', que especifican qué hosts/etc. pueden ejecutar comandos, lo cual está sujeto a spoofing y otro tipo de exploits. Jamás deberías utilizar el REXEC standard para administrar un sistema.

Slush

Slush está basado en OpenSSL, y actualmente soporta certificados X.509, lo cual, para grandes organizaciones, es una apuesta mucho mejor (y más sana) que intentar recordar varias docenas de contraseñas en diferentes servidores. Slush es GPL, pero todavía no está terminado (implementa la mayoría de la funcionalidad que se requiere para ser utilizable, pero tiene límites). Por otra parte, está completamente basado en software de código abierto, dejando pocas posibilidades a que pueda tener puertas traseras/etc. En último caso,

podría reemplazar al SSH por algo mejor. Lo puedes conseguir en <http://violet.ibs.com.au/slush/>.

NSH

NSH es un producto comercial con todos sus detalles (y he dicho todos). Tiene soporte para cifrado, de modo que es relativamente seguro de usar (esto no puedo verificarlo completamente, ya que no es código abierto). Es de una gran facilidad de uso, haces un `cd //nombredeordenador` y con eso ya haces el log en ese ordenador, puedes copiar/modificar/etc ficheros con facilidad, ejecutar `ps` y ver la lista de procesos de ese ordenador, etc. NSH también dispone de un módulo Perl, lo cual hace sencilla la redacción de scripts de comandos, y es ideal para administrar muchos sistemas similares (como estaciones de trabajo). Además de eso, NSH está disponible en múltiples plataformas (Linux, BSD, Irix, etc.) con RPM's disponibles para sistemas Red Hat. NSH está disponible en: <http://www.networkshell.com/>, y se pueden descargar versiones de evaluación de 30 días.

Fsh

Fsh significa "Ejecución rápida de comandos remotos", y el concepto es similar al de `rsh/rcp`. Evita el costo de estar creando continuamente sesiones cifradas, habilitando un túnel cifrado utilizando `ssh` o `lsh`, y ejecutando todos los comandos sobre él. Lo puedes conseguir en: <http://www.lysator.liu.se/fsh/>.

secsh

`secsh` (Shell Seguro) aporta otra capa más de seguridad de login, una vez que has hecho log vía `ssh` o `telnet` SSL te pide otra contraseña, si introduces una errónea, `secsh` mata el intento de login. Se puede conseguir en: <http://www.leenux.com/scripts>.

Locales

YaST

YaST (Yet Another Setup Tool, "otra herramienta más de seguridad") es un comando gráfico de líneas bastante interesante, (muy similar a `scoadmin`) que aporta una sencilla interfaz para la mayoría de las tareas administrativas. Sin embargo, no está pensado para limitar accesos a usuarios, así que sólo es útil para depurar errores y para permitir administrar su sistema a nuevos usuarios. Otro problema es que al contrario que `Linuxconf`, no está orientado a redes, lo cual quiere decir que hay que hacer un log en cada sistema que quieras manipular.

sudo

`Sudo` le da a un usuario acceso `setuid` a un programa(s), se le puede especificar desde qué `host(s)` se les permite (o no) hacer login y tener acceso `sudo` (de modo que si alguien vulnera una cuenta pero está bloqueado, se minimizan los daños). Se puede especificar bajo qué usuario se ejecutará un comando, lo cual te da un grado de control relativamente preciso. Si tienes que dar acceso a los usuarios, asegúrate de especificar los `hosts` desde los que les está permitido hacer un login cuando estén utilizando `sudo`, de igual forma, da la ruta completa a los binarios, lo cual te evitará problemas a la larga (p. ej. si le das acceso a `"adduser"` a un usuario, no hay nada que le impida editar su `path` y copiar `bash` a `/tmp/adduser` obteniendo el control de la máquina). Esta herramienta es muy similar a `super` pero con un control ligeramente inferior. `Sudo` está disponible para la mayoría de las distribuciones, como paquete interno, o paquete contribuido. `Sudo` está disponible en: <http://www.courtesan.com/sudo/>, por si tu distribución no viene con él.

Sudo te permite definir grupos de hosts, grupos de comandos, y grupos de usuarios, haciendo la administración más sencilla, a largo plazo. Aquí van varios ejemplos de /etc/sudoers :

Dar acceso total al usuario 'seifried'

```
seifried ALL=(ALL) ALL
```

Crear un grupo de usuarios, un grupo de hosts, y permitirles apagar el servidor como root

```
host_alias WORKSTATIONS=localhost, estacion1, estacion2
```

```
User_alias SHUTDOWNUSERS=rober, maria, juana
```

```
Cmnd_Alias REBOOT=halt, reboot, sync
```

```
Runas_Alias REBOOTUSER=admin
```

```
SHUTDOWNUSERS WORKSTATIONS=(REBOOTUSER) REBOOT
```

Super

Super es una de las pocas herramientas que se pueden utilizar hoy en día para dar a ciertos usuarios (y grupos) diferentes niveles de acceso a la administración del sistema. Además, se pueden especificar horas y permitir el acceso a scripts, puesto que dar acceso setuid, incluso a comandos comunes, puede tener resultados inesperados (cualquier editor, cualquier herramienta de manipulación de ficheros como chown, chmod, incluso herramientas como lp podrían comprometer partes del sistema). Debian viene con super, y existen rpm's disponibles en el directorio contrib. Es una herramienta potente (a veces deja a sudo a la altura del betún), pero necesita una sustancial cantidad de esfuerzo para implementarse correctamente (como cualquier herramienta potente), y creo que merece la pena. Suele haber ejemplos de ficheros de configuración en el directorio /usr/doc/super-xxxx/. El sitio principal de distribución de super es: <ftp://ftp.ucolick.org/pub/users/will/> .

runas

runas es muy parecido a sudo y Super, con algunas variaciones. Se crea un fichero de configuración listando el comando, como quien se ejecuta, y a qué usuarios/grupos/etc. se les permite ejecutarlo como tal. Además de esto, se pueden restringir el uso de opciones (argumentos), y se le puede solicitar al usuario el motivo (lo cual queda registrado con syslog). Esta es una característica de mis preferidas, ya que con un poco de entrenamiento, se puede conseguir que el staff de administración documente lo que está haciendo de forma liviana (p. ej. "quería reiniciar el sistema debido a fugas de memoria"). Runas se puede descargar de: <http://www.mindspring.com/~carpinello/runas/index.html>.

Remotas basadas en WWW

Webmin

Webmin es (actualmente), una herramienta de administración no comercial. Es un conjunto de scripts de perl con un servidor www autocontenido al cual se accede utilizando un visor de www. Tiene módulos para la mayoría de las funciones de administración del sistema, aunque algunas son un poco temperamentales. Una de sus características de mis preferidas es el hecho de que mantiene su propio usuario y contraseña para acceder a webmin, y se puede personalizar a qué tiene

acceso cada usuario (p. ej. usuario1 sólo puede administrar usuarios, usuario2 sólo puede reiniciar la máquina y usuario3 puede modificar la configuración del Apache). Webmin está disponible en: <http://www.webmin.com/>.

Linuxconf

Linuxconf es una herramienta de administración Linux de propósito general, que se puede utilizar desde la línea de comandos, desde X, o vía su propio servidor www. Es una de mis herramientas favoritas para administración automatizada del sistema (lo utilizo principalmente para hacer configuraciones de red complejas), ya que es relativamente ligera desde la línea de comandos (en realidad está dividida en varios módulos). Desde X proporciona un vistazo general de todo aquello que puede configurarse (PPP, usuarios, discos, etc.). Para utilizarlo vía visor www, primero hay que ejecutar Linuxconf en la máquina y añadir el o los hosts o red(es) a las que quieres permitir conectarse (Conf> Misc> Linuxconf network access), salvar los cambios y salir. Luego, cuando te conectes a la máquina (por defecto Linuxconf sólo acepta root como la cuenta, y Linuxconf no soporta ningún tipo de cifrado (se ejecuta independientemente en el puerto 901), de modo que desaconsejaría con vehemencia la utilización de esta característica entre redes, a menos que se tenga IPsec o algún otro tipo de nivel de seguridad IP. Linuxconf viene con Red Hat Linux y está disponible en: <http://www.solucorp.qc.ca/linuxconf/>. Linuxconf no parece que venga con páginas de manual/etc., la ayuda está contenida internamente, lo cual es ligeramente cabreante.

COAS

El proyecto COAS (Caldera Open Administration System) es un proyecto muy ambicioso para proporcionar un marco abierto de administración de sistemas, desde línea de comandos (con interfaz semi-gráfico), desde X (utilizando el componente qt) hasta el web. Hace una abstracción de los datos reales de configuración aportando una capa intermedia, permitiendo de esta forma su uso en variadas plataformas Linux. Acaba de salir la versión 1.0, parece que finalmente Caldera va adelante con ello. El sitio de COAS está en: <http://www.coas.org>.

Otras herramientas basadas en red

VNC

El Virtual Network Computer (VNC) es parecido a X o a PCAnywhere. Se puede mostrar un escritorio gráfico, y controlarlo remotamente, con NT o Linux como servidor y/o cliente. El VNC es bastante bueno a través de una Ethernet de 10 megabit, sin embargo tiende a utilizar un montón de potencia computacional relativamente comparado con otros métodos de administración remota. Se puede conseguir en: <http://www.uk.research.att.com/vnc/>. La seguridad del VNC no es tan buena, pero hay varios sitios con información acerca de asegurar VNC, utilizando SSL, SSH y otros métodos. MindVNC es un cliente java que utiliza SSH, disponible en: <http://www.mindbright.com/english/technology/products/mindvnc.html>. Existe un parche disponible para añadir cifrado al VNC en: <http://web.mit.edu/thouis/vnc/>.

Gestión de Software

RPM

RPM es una herramienta de gestión de software creada originalmente por Red Hat, y más tarde hecha GNU y puesta a disposición del público (<http://www.rpm.org/>). Forma parte del corazón de la administración en la mayoría de sistemas, ya que una de las tareas más importantes de cualquier administrador es la instalación y mantenimiento de software actualizado. Varias estimaciones otorgan la mayor parte de la culpa de los incidentes de seguridad a las malas contraseñas, y al software viejo con vulnerabilidades conocidas. Lo cual no es tan sorprendente como uno podría pensar, pero mientras que el servidor medio contenga entre 200 y 400 paquetes de software de media, uno empieza a ver por qué mantener el software actualizado puede ser una tarea importante.

La página del manual de RPM es bastante mala, no existe una forma de hacerla agradable. El libro "Maximum RPM" (ISBN: 0-672-31105-4) por otra parte es realmente estupendo (disponible gratuitamente en <http://www.rpm.org/> en formato post script). Le sugeriría este libro a cualquier administrador de Red Hat, y puedo asegurar que es de lectura necesaria si está pensando en construir paquetes RPM. Las bases de RPM son autoexplicativas, los paquetes vienen en formato rpm, con una simple convención de nombre de fichero:

nombre_paquete-versión_paquete-rpm-versión_construcción-arquitectura.rpm

servidor-nfs-2.2beta29-5.i386.rpm sería "servidor-nfs", versión "2.2beta29" del "servidor-nfs", la quinta construcción de ese rpm (p. ej. se ha empaquetado y construido 5 veces, modificaciones menores, cambios en la localización de ficheros, etc.), para la arquitectura Intel, y es un fichero rpm.

| Comando | Función |
|------------------------|--|
| -q | Consulta información de los paquetes / bases de datos |
| -i | Instala el software |
| -U | Actualiza o instala el software |
| -e | Extrae el software del sistema (borra) |
| -v | dar más información |
| -h | Marcas hash |
| Ejemplo de comando | Función |
| rpm -ivh paquete.rpm | Instala 'paquete.rpm', da más información, muestra marcas hash |
| rpm -Uvh paquete.rpm | Actualiza 'paquete.rpm', da más información, muestra marcas hash |
| rpm -qf /algún/fichero | Comprueba a qué paquete pertenece un fichero |
| rpm -qpi paquete.rpm | Consulta 'paquete.rpm', saca un listado con información |

| | |
|-----------------------|--|
| rpm -qpl paquete.rpm | Consulta 'paquete.rpm', saca un listado de todos los ficheros |
| rpm -qa | Consulta la base de datos RPM, lista todos los paquetes instalados |
| rpm -e nombre-paquete | Elimina 'nombre-paquete' del sistema (según listado por rpm -qa) |

Red Hat linux 5.1 venía con 528 paquetes, y Red Hat Linux 5.2 venía con 573, cuando te paras a pensar te das cuenta que es un montón de software. (SuSE 6.0 viene en 5 CD's, no me he molestado en contar cuántos paquetes). Por lo general acabarás con unos 200-300 paquetes instalados (más aplicaciones en estaciones de trabajo, los servidores tienden a ser más livianos, pero no siempre es el caso). Así que cuáles se deberían instalar y cuáles se debería evitar instalar si fuese posible (como los paquetes de servicios remotos). Tengo que decir algo, los RPM's que vienen con las distribuciones Red Hat suelen ser bastante buenos, y por lo general duran entre 6-12 meses antes de que se descubra que están estropeados.

Hay una lista de URL's y listas de correo con erratas específicas por distribución y más adelante, dentro de este documento se indicarán dónde están disponibles las actualizaciones

dpkg

El sistema de paquetes de Debian es un paquete similar a RPM, sin embargo carece de cierta funcionalidad, aunque en conjunto hace un trabajo excelente gestionando los paquetes de software de un sistema. Combinado con la utilidad dselect (ya un poco desfasada), se puede conectar con sitios remotos, recorrer los paquetes disponibles, instalarlos, ejecutar scripts de configuración necesarios (como para gpm), todo desde la comodidad de tu consola. La página del manual de dpkg "man dpkg" es bastante grande.

El formato general de un fichero con un paquete de Debian (.deb) es:

nombrepaquete_versiónpaquete-debversion.deb

ncftp2_2.4.3-2.deb

Al contrario que los ficheros rpm, los ficheros .deb no traen indicativo de la arquitectura (no es que tenga demasiada importancia, pero es algo que hay que tener en cuenta).

| Comando | Función |
|---------------------|--|
| -I | Consulta un paquete |
| -i | Instala el software |
| -l | Saca un listado del software instalado (equivalente a rpm -qa) |
| -r | Elimina el software del sistema |
| Ejemplo de comando | Función |
| dpkg -i paquete.deb | Instala paquete.deb |
| dpkg -I paquete.deb | Saca un listado de la información de paquete.deb (rpm |

| | -qpi) |
|---------------------|--|
| dpkg -c paquete.deb | Saca un listado de todos los ficheros de un paquete (rpm -qpl) |
| dpkg -l | Muestra todos los paquetes instalados |
| dpkg -r | Elimina 'nombre-paquete' del sistema (según listado por dpkg -l) |

Debian tiene más de 1.500 paquetes disponibles para el sistema. Aprenderás a amar al dpkg (funcionalmente tiene todo lo necesario, sólo me dejo algunas de las florituras que tiene rpm, por otra parte, dselect tiene algunas características que ya quisiera yo que tuviera rpm).

Más adelante hay una lista de URL's y listas de correo con información específica sobre erratas por distribuciones.

tarballs / tgz

La mayoría de las distribuciones modernas de Linux utilizan un sistema de gestión de paquetes para instalar, llevar un seguimiento y eliminar software del sistema. Sin embargo hay muchas excepciones, Slackware no utiliza un sistema de gestión de paquetes per se, sino que tiene tarballs precompilados (un fichero tar comprimido que contiene ficheros), los cuales tan sólo hay que desempaquetar desde el directorio raíz para instalarlos, algunos de los cuales también pueden ser eliminados, pero funciones tales como consultas, comparación de ficheros instalados contra ficheros de paquetes (intentando encontrar ficheros falsificados, etc.) simplemente no están. O quizás quieras probar la última copia de X, y nadie se las ha apañado todavía para hacer un bonito fichero .rpm o .deb, así que tienes que pillar el código fuente (normalmente viene comprimido en un tarball), desempaquetarlo e instalarlo. Esto no supone más peligro real que un paquete, ya que la mayoría de los tarballs tienen asociadas firmas MD5 o PGP, que puedes descargar y comprobar. La verdadera preocupación con estos ficheros es la dificultad que existe a veces para hacer el seguimiento de si se tiene cierto tipo de software instalado, determinar la versión, y después eliminarlo actualizarlo. Recomendaría que no se utilizasen tarballs si fuese posible, si hay que usarlos, es una buena idea hacer una lista de los ficheros del sistema antes de instalarlo, y otra después de instalarlo, y posteriormente compararlas utilizando 'diff', para averiguar qué fichero ha sido colocado en qué lugar. Simplemente ejecuta 'find /* > /listaficheros.txt' antes y 'find /* > /listaficheros2.txt' después de instalar el tarball, y utiliza 'diff -q /listaficheros.txt /listaficheros2.txt > /listadiferencia.txt' para obtener un listado de qué es lo que ha cambiado. Alternativamente, un 'tar -tf blah.tar' sacará un listado de los contenidos del fichero, pero como la mayoría de los tarballs, acabarás ejecutando un script de instalación/compilación e instalando el software, de modo que un simple listado de ficheros no te va a dar una idea exacta de qué fue lo que ha sido instalado/modificado. Otro método de llevar un seguimiento de qué es lo que se ha instalado vía tar es utilizar un programa como 'stow'; stow instala el paquete en un directorio aparte (/opt/stow/), por ejemplo, y después crea links desde el sistema a ese directorio, según sea necesario. Para ejecutar Stow es necesario tener Perl instalado, y se encuentra disponible en: <http://www.gnu.ai.mit.edu/software/stow/stow.html>

| Comando | Función |
|---------|-----------------------------|
| -t | Saca un listado de ficheros |
| -x | Extrae ficheros |

| Ejemplo de comando | Función |
|-----------------------------------|--|
| <code>tar -xf fichero.tar</code> | descomprime fichero.tar |
| <code>tar -xt filename.tar</code> | Saca un listado de los ficheros de fichero.tar |

Comprobación de la integridad de ficheros

Algo que he pensado cubrir por semi-separado es la verificación de la integridad del software adquirido desde sitios remotos. Generalmente, la gente no se preocupa, pero hace poco entraron en ftp.win.tue.nl, y se insertó un trojano en el paquete (entre otros) TCP_WRAPPERS. Se hicieron 59 descargas antes de que el site eliminase los paquetes ofensivos e iniciase un procedimiento de control de daños. Siempre se debería verificar la integridad de los ficheros que descargues desde sitios remotos, algún día pueden entrar en un sitio de los grandes y mucha gente sufriría bastantes daños.

RPM

Los paquetes RPM pueden estar (y de hecho lo suelen estar) firmados con PGP por el autor. Esta firma se puede comprobar para asegurar que el paquete no ha sido falsificado o es una versión con trojano. Esto queda descrito más adelante en el capítulo 7 de "Maximum RPM" (disponible en línea en <http://www.rpm.org/>), pero consiste en añadir las llaves de los desarrolladores a tu llavero público PGP, y después utilizar la opción -K, que cogerá la llave apropiada del llavero y verificará la firma. De esta forma, para insertar un trojano en un paquete firmado correctamente, se necesitaría robar la llave PGP privada y la contraseña para descifrarla, lo cual sería casi imposible.

dpkg

dpkg soporta MD5, de modo que hay que conseguir las firmas MD5 mediante un canal seguro (como correo firmado con PGP). MD5 viene con la mayoría de las distribuciones.

PGP

Se distribuyen muchos tarballs con firmas PGP en ficheros ASCII separados, para verificarlos, se añade la llave de los desarrolladores a tu llavero y se utiliza PGP con la opción -o. De esta forma, para insertar un trojano en un paquete y firmarlo correctamente, habría que robar la llave privada PGP del desarrollador y la contraseña para descifrarla, lo cual debería ser casi imposible. PGP para Linux está disponible en <ftp://ftp.replay.com/>.

MD5

Otra forma de firmar un paquete es crear un checksum MD5. La razón por la que se utilizaría MD5 (ya que cualquiera puede crear una firma MD5 válida para un paquete de software con trojano), es que MD5 es bastante universal y no está controlado por leyes de exportación. La debilidad es que de alguna manera hay que distribuir con antelación las firmas MD5 por un canal seguro, lo cual se suele hacer vía correo, cuando se anuncia un paquete (vendedores como Sun lo hacen con los parches).

Actualizaciones automáticas

RPM

Hay una gran variedad de herramientas disponibles para instalaciones automáticas de ficheros rpm.

<ftp://ftp.kaybee.org/pub/linux/>

AutoRPM es probablemente la mejor herramienta para mantener actualizados los rpm's, se le hace apuntar a un directorio ftp, y descarga e instala cualquier paquete que sea más nuevo que los que se tienen. Por favor, ten en cuenta que si alguien envenena tu caché de dns, te pueden comprometer con facilidad, de modo que asegúrate que utilizas la dirección IP del ftp y no su nombre. También deberías considerar apuntar a un sitio ftp interno con los paquetes que has probado, y mantener un mayor control sobre él. AutoRPM requiere tener instalado el paquete libnet Net::FTP para perl.

<ftp://missinglink.darkorb.net/pub/rhlupdate/>

rhlupdate también se conectará a un sitio ftp y cogerá todas las actualizaciones necesarias, se aplican las mismas advertencias que más arriba, y de nuevo requiere tener instalado el paquete libnet Net::FTP para perl.

<http://www.iaehv.nl/users/grimaldo/info/scripts/>

RpmWatch es un sencillo script en perl que instalará actualizaciones en lugar de ti, ten en cuenta que no descargará los paquetes que necesitas, tienes que hacer un mirror de ellos localmente, o tenerlos accesibles localmente vía algo como NFS o CODA.

dpkg

dpkg tiene un instalador automático interesante, llamado 'apt', además de instalar software, capturará e instalará el software necesario para cumplir las dependencias, se puede descargar de:

<http://www.debian.org/Packages/stable/admin/apt.html>

tarballs / tgz

No se han encontrado herramientas, por favor, dime si conoces alguna (aunque más allá de hacer mirror, desempaquetar automáticamente y ejecutar `./configure; make; make install`", no se me ocurre nada más).

Seguimiento de cambios

installwatch

installwatch monitoriza lo que hace un programa, y lleva un log al syslog de cualquier cambio que hace al sistema. Se parece al programa "time" en que ejecuta el programa de forma escalonada, de modo que pueda monitorizar lo que ocurre, el programa se ejecuta como `installwatch /usr/src/algo/make` por ejemplo (opcionalmente se puede utilizar `-o nombrefichero` para hacer un log a un fichero específico), installwatch está disponible en:

<http://datanord.datanord.it/~pdemauro/installwatch/>

instmon

instmon se ejecuta antes y después de instalar un paquete tarball / tgz (o cualquier paquete, por lo que respecta). Genera una lista de fichero que han cambiado, la cual se puede utilizar más adelante para deshacer cualquier cambio. Está disponible en: <http://hal.csd.auth.gr/~vvas/instmon/>

Conversión de formatos

Otra forma de manejar paquetes/etc. es convertirlos. Existen diferentes utilidades para convertir ficheros rpm a tarballs, rpm's a deb's, etc.

alien

alien es probablemente la mejor utilidad para convertir ficheros, maneja rpm's, deb's y tarballs bastante bien. Se puede descargar de:
<http://kitenet.net/programs/alien/>

slurp

Slurp adopta un enfoque interesante, de alguna forma se comporta como installwatch, pero también tiene alguna de las características de alien. Monitoriza el sistema a medida que se instala el paquete, para lo cual crea un rpm. Slurp se puede conseguir en: <http://students.vassar.edu/~jajohnst/slurp/>

Encontrar software

Uno de los mayores problemas con Linux es encontrar software que no viene con tu distribución. Buscar en Internet es un coñazo. Sin embargo existen algunos recursos que pueden aliviar tus penas:

- * <http://www.rpmfind.net/>
- * <http://www.linuxapps.com/>
- * <http://www.freshmeat.net/>

Herramientas de monitorización de Hosts

La monitorización de tu servidor(es) y host(s) es importante por una serie de razones, desde el seguimiento de las irrupciones hasta los requerimientos legales. Recuerda, más vale prevenir que curar. Existen una variedad de herramientas de monitorización generales disponibles en la sección de logs, desde el syslog hasta el auditd (que te permite auditar los ficheros que tienen abiertos los usuarios, programas en ejecución, etc). Sugeriría encarecidamente su uso. De igual forma, existe un número de programas más específicos para monitorizar el estado del sistema y evitar que los usuarios hagan cosas que no deberían.

bgcheck

El bgcheck se ejecuta en background y comprueba la tabla de procesos en busca de elementos que deberían estar ejecutándose (p. ej. robots de irc, crackers de contraseñas, etc.). Se puede descargar desde: <http://blue.dhs.org/bgcheck/>

Sxid

El Sxid comprueba el setuid y el setgid en busca de cambios, genera firmas MD5 de ficheros y generalmente te permite hacer un seguimiento de los cambios efectuados. Se puede conseguir en: <ftp://marcus.seva.net/pub/sxid/>

ViperDB

El ViperDB comprueba los programas setuid/setgid y las carpetas, y te puede notificar (vía syslog) acerca de cualquier cambio o resetear los permisos y la pertenencia a la forma en que deberían estar. El ViperDB crea una serie de bases de datos (en realidad textos planos) en el directorio raíz, p. ej.:

```
/etc/.ViperDB podría contener:
```

```
/etc/login.defs,1180,-,root,rw-,root,r--,r--,Apr,15,18:03
```

```
/etc/minicom.users,1048,-,root,rw-,root,r--,r--,Mar,21,19:11
```

```
/etc/CORBA,1024,d,root,rwx,root,r-x,r-x,Jun,14,16:51
```

```
/etc/X11,1024,d,root,rwx,root,r-x,r-x,Jun,14,23:05
```

```
/etc/cron.d,1024,d,root,rwx,root,r-x,r-x,Apr,14,17:09
```

Por desgracia el ViperDB no parece ser capaz de manejar subdirectorios, de modo que tendrás que añadirlos al fichero viperdb.ini con algo como esto:

```
find /etc/ -type d >> /usr/local/etc/viperdb.ini
```

El viperdb.pl tiene 3 opciones, -init (crea un conjunto de bases de datos), -check (comprueba los ficheros contra las bases de datos, envía cualquier mensaje al syslog, y después vuelve a crear las bases de datos) y -checkstrict (comprueba los ficheros contra las bases de datos, resetea los permisos si es necesario, envía cualquier mensaje al syslog y después vuelve a crear las bases de datos). Lo cual significa que si se utiliza -check, se obtendrá una advertencia diciendo que /etc/passwd no es tiene permiso de escritura por el mundo, y puesto que vuelve a crear las bases de datos, la próxima vez que se ejecute viperdb NO mostrará la advertencia. Recomendaría ejecutar el viperdb exclusivamente en modo checkstrict, y asegurarte de que se ejecuta el viperdb con la opción -init después de manipular los permisos de cualquier fichero / carpeta de directorios protegidos. El ViperDB se encuentra disponible para

descarga en: <http://www.resentment.org/projects/viperdb/>

Pikt

El Pikt se vio en la sección anterior, y se puede utilizar para monitorizar la actividad de los usuarios. Lo recomendaría para grandes instalaciones, pues es extremadamente flexible y potente. Se encuentra disponible en:
<http://pikt.uchicago.edu/pikt/>

DTK

El Deception ToolKit es un conjunto de programas que emulan servicios bastante conocidos, para proporcionar una serie de lecturas falsas a los atacantes. El objetivo es confundir y retardar a los atacantes llevándoles a falsas conclusiones, el DTK se puede descargar desde: <http://all.net/dtk/>

Ficheros de Log y otros métodos de monitorización

Una de las partes integrales de cualquier sistema UNIX son las facilidades para hacer logging. La mayoría del logging de Linux viene proporcionado por dos programas principales, `sysklogd` y `klogd`, el primero proporciona servicios de logging a programas y aplicaciones, el segundo proporciona capacidad de logging al kernel de Linux. `klogd` en realidad envía la mayoría de los mensajes al `syslogd`, pero de vez en cuando lanzará mensajes a la consola (p. ej., cuando el kernel entra en pánico). `sysklogd` en realidad maneja la tarea procesar la mayoría de los mensajes y de enviarlos al fichero o dispositivo apropiado, lo cual se configura desde `/etc/syslog.conf`. Por defecto, la mayoría del logging a ficheros tiene lugar en `/var/log/`, y generalmente los programas que manejan su propio logging (la mayoría de servidores `httpd` manejan sus logs internamente) lo hacen a `/var/log/nombreprograma/`, lo cual te permite centralizar los ficheros de log y hace más sencillo situarlos en particiones diferentes (algunos ataques pueden consumir los logs rápidamente, y tener una partición / no es algo divertido). Además, existen programas que manejan internamente sus logs, siendo uno de los más interesantes el shell de comandos `bash`. Por defecto, `bash` mantiene un historial de los comandos ejecutados en la línea de comandos. `Apache` maneja todos sus logs internamente, configurable desde `httpd.conf` y extremadamente flexible con la aparición de `Apache 1.3.6` (soporta logging condicional). `Sendmail` maneja sus logs vía `syslogd`, pero también tiene la opción (vía línea de comandos con el atributo `-X`) de hacer un log de todas las transacciones SMTP directamente a un fichero. Esto es altamente desaconsejable, puesto que el fichero crecerá enormemente en un corto espacio de tiempo, pero es útil para hacer debugging. Para más información, ver las secciones acerca de seguridad de red en `Apache` y `sendmail`.

Seguridad general de logs

Por lo general, no querrás permitir que los usuarios vean los ficheros de log de un servidor, y especialmente que puedan ser capaces de modificarlos o borrarlos. En general, la mayoría de los ficheros de log son propiedad del usuario y grupo `root`, y no tienen asignados permisos para otros, de modo que en la mayoría de los casos, el único usuario que será capaz de modificar los logs será el `root` (y si alguien revienta la cuenta del `root`, pues apaga y vámonos). Se pueden tomar medidas de seguridad adicionales, siendo la más simple el uso de "chattr" (CHange ATTRIBUTES, el comando para cambiar los atributos), para dejar el exclusivamente permiso de sólo-añadir a los ficheros de log. De tal forma que si se da un problema de raza en `/tmp` que permita a la gente sobrescribir ficheros del sistema no se puedan dañar ficheros de logs. Para dejar un fichero en modo sólo-añadir, utiliza:

```
chattr +a nombrefichero
```

A la función `chattr` sólo tiene acceso el superusuario. Si dejas todos los ficheros en modo sólo-añadir, recuerda que fallarán los programas de rotación de logs, puesto que no pueden dejar en cero el fichero de log. Añade una línea al script para deshabilitar el atributo sólo-añadir:

```
chattr -a nombrefichero
```

y añade una línea después del script de rotación de logs para reiniciar el flag de sólo-añadir. Si los ficheros de logs se dejan en el sistema, quizás prefieras activar también el flag de inmutables, de forma que no se puedan falsificar con facilidad. Para activar el fichero como inmutable, simplemente:

```
chattr +i nombrefichero
```

lo que evitará cualquier cambio (debida a razas en el /tmp, etc.) en el fichero, a menos que el atacante tenga acceso de root (en cuyo caso vas a sufrir de todas formas).

```
chattr -i nombrefichero
```

sólo el usuario root tiene acceso al flag de immutable.

```
sysklogd / klogd
```

En resumen, klogd maneja los mensajes del kernel, dependiendo de tu configuración, puede ir desde casi ninguno a una buena cantidad, si por ejemplo activas la opción de contabilidad de procesos. Después se pasa la mayoría de los mensajes al syslogd, para el manejo real (es decir, es el que escribe los datos físicamente al fichero). Las páginas del manual del sysklog, klogd y syslog.conf son bastante buenas, con ejemplos claros. Una característica del syslog muy potente y a menudo despreciada es la posibilidad de hacer un log de mensajes a hosts remotos que estén ejecutando syslog. Puesto que se pueden definir múltiples ubicaciones para los mensajes del syslog (p. ej., enviar todos los mensajes del kernel al fichero /var/log/messages, y a consola, a uno o múltiples hosts remotos), lo cual te permite centralizar las tareas de logs a un único host, y verificar con facilidad violaciones en la seguridad o demás anomalías. Sin embargo, existen varios problemas con syslogd y klogd, siendo el principal la facilidad con la que un atacante, una vez que ha conseguido acceso de root, puede borrar/modificar ficheros de log, no existe un método de autenticación construido dentro de las capacidades de hacer logging.

Los ficheros de log standard y que normalmente vienen definidos en syslog.conf son:

```
/var/log/messages
```

```
/var/log/secure
```

```
/var/log/maillog
```

```
/var/log/spooler
```

El primero (messages) por lo general captura la mayoría de la información; el usuario hace el login, los TCP_WRAPPERS vuelcan aquí la información, el cortafuegos de paquetes IP también vuelca información aquí, etc. El segundo generalmente registra entradas de eventos tales como usuarios cambiando su UID/GID (vía su, sudo, etc.), intentos fallidos cuando se requieren contraseñas, etc. El fichero maillog guarda por lo general entradas de cada conexión pop/imap (nombre de usuario y logout), y el encabezamiento de cada uno de los correos que entran o salen del sistema (de quien, a quien, msgid, estado, etc.). El fichero spooler ya no se suele usar, puesto que el número de gente que utiliza usenet o uucp ha caído en picado, el uucp ha sido reemplazado por el ftp y el correo, y la mayoría de los servidores de usenet suelen ser por lo general, máquinas extremadamente potentes como para manejar un newsfeed parcial o completo, lo cual quiere decir que no hay muchos de ellos (generalmente uno por cada PSI o más, dependiendo del tamaño). La mayoría de los usuarios desde casa o de pymes no van a estar ejecutando un servidor de usenet (al menos no deberían, en mi opinión), la cantidad de ancho de banda y potencia de máquina que se requiere es inmensa, y no digamos ya los riesgos de seguridad.

También se pueden definir ficheros de log adicionales, por ejemplo, se podría añadir:

```
kern.* /var/log/kernel-log
```

Y se puede hacer un log selectivamente a otro host para logs separado:

```
*.emerg    @syslog-host
```

```
mail.*     @mail-log-host
```

Lo cual daría como resultado que todos los mensajes del kernel fuesen a parar a /var/log/kernel-log, lo cual es útil en servidores sin monitor, ya que por defecto todos los mensajes del kernel van a /dev/console (p. ej. si alguien está conectado en las máquinas). En el segundo caso, todos los mensajes de emergencia irían al host "syslog-host", y todos los ficheros del log de correo se enviarían al servidor "mail-log-host", permitiéndote mantener ficheros de log de varios servicios centralizados

secure-syslog

El problema principal con el syslog es que falsificar los ficheros es algo trivial. Sin embargo existe una versión segura del syslogd, disponible en <http://www.core-sdi.com/ssyslog/> (estos chicos suelen hacer buenas herramientas de seguridad y tiene una buena reputación, en cualquier caso es software de fuente abierta, para aquellos que sean realmente paranoicos). Esta te permite firmar criptográficamente los logs, para asegurarse de que no han sido falsificados. Sin embargo, en último lugar, un atacante podría borrar los ficheros de log, de modo que es una buena idea enviarlos a otro host, especialmente en el caso de un cortafuegos, para prevenir que el disco duro se llene.

next generation syslog

Otra alternativa es "syslog-ng" (Next Generation Syslog, Syslog de Nueva Generación), que parece bastante más personalizable que incluso el syslog o secure-syslog, soporta firmas digitales para prevenir falsificación de ficheros, y puede filtrar basándose en el contenido del mensaje, no sólo la procedencia y la prioridad (algo bastante útil para reducir el volumen). Syslog-ng está disponible en: <http://www.balabit.hu/products/syslog-ng.html>

Nsyslogd

Nsyslogd soporta tcp y SSL para enviar el log a sistemas remotos. Se ejecuta en una gran variedad de plataformas UNIX, lo puedes descargar de: <http://coombs.anu.edu.au/~avalon/nsyslog.html>

Monitorización de Logs

Psionic Logcheck

Psionic Logcheck navega por los ficheros de mensajes (y otros) a intervalos regulares (normalmente se invoca vía crontab), y envía un correo con un sumario de cualquier actividad sospechosa. Es fácilmente configurable, con varios "tipos" de elementos, intentos activos de penetración, sobre los que enseguida te grita, actividad dañina y actividad que se debe ignorar (por ejemplo, estadísticas del servidor DNS o regeneración de llaves SSH). Psionic Logcheck se encuentra disponible en: <http://www.psionic.com/abacus/logcheck/>.

colorlogs

colorlogs colorea los ficheros de logs, lo cual te permite identificar con facilidad actividad sospechosa. Basado en un fichero de configuración, busca palabras clave y colorea las líneas (en rojo, cyan, etc.), recibe la entrada desde STDIN, de modo que se puede utilizar para revisar ficheros de log

rápidamente (utilizando "cat", "tail" u otras utilidades para alimentar el fichero de logs a través del programa). Se puede conseguir en:
<http://www.resentment.org/projects/colorlogs/>.

WOTS

WOTS recolecta ficheros de logs desde múltiples orígenes, y genera informes o toma medidas basándose en lo que le digas que haga. WOTS busca expresiones regulares que se le definan, y después ejecuta los comandos que le listas (enviar un informe, hacer sonar una alerta, etc.). WOTS requiere que tengas instalado perl, y está disponible en: <http://www.vcpc.univie.ac.at/~tc/tools/>.

swatch

swatch es muy parecido a WOTS, y la configuración de los ficheros de log es muy similar. Puedes descargarlo de:
<ftp://ftp.stanford.edu/general/security-tools/swatch/>

Logs del Kernel

auditd

auditd te permite utilizar las capacidades de log del kernel (una herramienta muy potente). Se puede hacer un log de mensajes de correo, eventos de sistema y los elementos habituales que cubriría syslog, pero además se pueden cubrir eventos como que usuarios específicos abran ficheros, la ejecución de programas, de programas setuid, etc. Si necesitas sólidos seguimientos de auditoría, esta es la herramienta que necesitas, la puedes conseguir en:
<ftp://ftp.hert.org/pub/linux/auditd/>.

Logs del Shell

bash

También trataré bash, ya que es el shell por defecto en la mayoría de las instalaciones de Linux, y como tal su posibilidad de hacer logging suele la habitualmente utilizada. bash tiene una gran cantidad de variables que se pueden configurar en tiempo de ejecución o durante su uso, las cuales modifican su comportamiento. Cualquier cosa desde el estilo del prompt hasta cuántos ficheros se pueden almacenar en el fichero de logs.

HISTFILE

nombre del fichero de historial, por defecto es `~nombreusuario/.bash_history`

HISTFILESIZE

número máximo de comandos a mantener en el fichero, se rotan a medida que sea necesario.

HISTSIZE

el número de comandos que recuerda (p. ej. cuando se usa el cursor arriba).

Las variables se suelen configurar en `/etc/profile`, lo cual configura el bash globalmente para todos los usuarios, sin embargo, estos valores pueden pasarse por alto con el fichero `~nombreusuario/.bash_profile`, y/o usando manualmente el comando `export` para configurar variables como `export EDITOR=emacs`. Esta es una de las razones por las que los directorios de los usuarios no deberían ser legibles por el mundo; el fichero `.bash_history` puede almacenar una gran cantidad de información valiosa para partes hostiles. Se puede hacer que el

fichero no sea legible, que no se haga log del fichero `.bash_profile`, hacer que no se pueda escribir en el fichero (denegando de esta forma al shell la posibilidad de escribir y hacer log de ello) o enlazarlo a `/dev/null` (lo cual casi siempre suele ser síntoma claro de actividad sospechosa por parte del usuario, o usuarios paranoicos). En cuanto a la cuenta del root, recomendaría encarecidamente configurar el `HISTFILESIZE` y `HISTSIZE` con un valor bajo, tal como 10. Por otra parte, si quieres hacer un log del historial de shell de los usuarios, para así reforzar la seguridad, recomendaría dejar los ficheros de configuración de los directorios de los usuarios como inmutables, utilizando el comando `chattr`, y dejar los ficheros de logs (como `.bash_history`) como sólo-añadir. Sin embargo, hacer esto tiene implicaciones legales, de modo que asegúrate de que los usuarios son conscientes de que se les está registrando y que están de acuerdo con ello, o si no podrías tener problemas.

Limitación y monitorización de usuarios

Limitación de usuarios

Existen muchas cosas malas que puede hacer un usuario a un sistema si tiene la cuenta de shell interactiva. También existen muchas formas de evitar que esto ocurra. Las cuotas de usuario de utilización del disco, de CPU, etc. son un buen punto de partida, también pueden ayudar técnicas de monitorización más avanzadas, como monitorizar a usuarios con amplios entornos, etc. Una de las cosas más simples que puede hacer un usuario es ocupar toda la memoria lanzando múltiples copias de un programa que se coma la memoria, o utilizar todos los descriptores de ficheros con una "bomba fork".

PAM

La mayoría de los Linux modernos vienen con soporte PAM, y una de las cosas que proporciona PAM es la configuración de entorno. Configuraciones tales como limitar la cantidad de memoria que le está permitido utilizar al usuario. En Red Hat y Caldera se configura desde el directorio /etc/security, el cual contiene un determinado número de ficheros. El fichero más interesante es: /etc/security/limits.conf, que permite definir reglas para usuarios o grupos, ya sea una regla "soft" o "hard" (información sobre esto más adelante), y a qué se le aplica la regla, que puede ser CPU, memoria, tamaño máximo de fichero, etc. Por ejemplo:

```
* hard core 0
pepe soft nproc 100
pepe hard nproc 150
```

La primera regla deshabilita los core dumps para cualquiera, la segunda regla establece un límite soft para pepe de 100 procesos, y la tercera establece un límite hard para pepe de 150 procesos. Un límite soft se sobrepasar, suele ser una nota de advertencia, el límite hard no se puede sobrepasar. Como te puedes imaginar es bastante útil puesto que se le aplica a todos los logins de shells, y a otros servicios como ftp.

Bash

El bash viene con un limitador interno, al que se accede mediante 'ulimit'. No se puede configurar más alto cualquier límite hard, de forma que si se tiene definido un límite en /etc/profile o en el fichero .bash_profile (suponiendo que no puedan borrar / editar esos ficheros) se pueden reforzar los límites de los usuarios de shell Bash. Esto es útil para distribuciones obsoletas que carecen de soporte PAM. También habrá que asegurarse de que el usuario no puede cambiar su shell de login. Configurar los límites es parecido al método de PAM. Se definen varios como:

```
ulimit -Sc 0
ulimit -Su 100
ulimit -Hu 150
```

Con estas tres reglas se conseguirían los mismos resultados que con el ejemplo PAM. La primera regla deshabilita los core dumps, la segunda establece un límite soft de 100 procesos y la tercera un límite hard de 150 procesos. Más información sobre ulimit disponible tecleando "help ulimit" desde el intérprete

de shell.

Quota

La cuota es un sistema para restringir la utilización del disco por parte de los usuarios. Viene con la mayoría de las distribuciones y se encuentra disponible la ayuda mediante "man quota".

Monitorización de usuarios

Una tarea típica en la mayoría de servidores de shells es asegurarse de que los usuarios no abusan del servidor. Es algo bastante sencillo de monitorizar en cuanto a los recursos standard (como uso del disco duro, uso de CPU, etc.) sin embargo uno de los abusos más frecuentes es el de ancho de banda, aunque afortunadamente existe una variedad de métodos de monitorizar tal abuso.

ttysnoop

Por supuesto que todo esto va bien mientras no vaya mal. Pero qué pasa si realmente se quiere monitorizar lo que está haciendo un usuario (estás advertido, esto tiene implicaciones legales que pueden buscarte problemas, primero consulta con tus abogados). Es aquí donde entra en juego el ttysnoop. El ttysnoop te permite monitorizar lo que está haciendo un usuario y grabarlo. Se puede conseguir en: <http://uscan.cjb.net>

UserIPAcct

UserIPAcct te permite monitorizar el uso de ancho de banda por usuario, lo cual implica parchear el kernel y configurar las reglas (un concepto similar al del cortafuegos) para monitorizar la cantidad de datos que pueden enviar o recibir los programas de un usuario. Sin embargo, no se pueden contabilizar los datos para conexiones PPP, puesto que el demonio PPP no se ejecuta bajo el mismo nombre de usuario (aunque se podría modificar para que así lo hiciera). Lo recomendaría en servidores de shells para monitorizar a los usuarios (generalmente, sólo una pequeña minoría hará un uso excesivo). El paquete completo se puede descargar de: <http://zaheer.grid9.net/useripacct/>

Lista de comprobación para la conexión a Internet

Lo siguiente es una lista de comprobación para la gente que vaya a conectar el equipo a Internet (PPP, *DSL, Cablemodem, etc). De ninguna forma se considera completa esta lista, pero debería ayudar.

Desactiva todos los demonios que no sean necesarios, y el software de red - deshabilita telnet, ftp, ntalk, auth, pop, imap, etc. a menos que se tenga planeado utilizarlo. Si está desactivado, presenta menos riesgos.

Si es posible, utiliza ipfwadm/ipchains para filtrar con cortafuegos los servicios, si se filtra todo esto por defecto, ralentizará cualquier escaneo que la gente ejecute sobre tus máquinas. Por lo general, servicios tales como el NFS / Samba, imap y pop sólo necesitarán acceso de usuarios internos, bloquear los accesos externos simplifica las cosas.

Utiliza TCP_WRAPPERS para asegurar los servicios que se dejen activados, cuando sea posible, restringir el acceso a los clientes internos o ciertos clientes Internet de servicios como imap, pop y ftp. También recuerda que la mayoría de las distribuciones vienen con nfs configurado para utilizar TCP_WRAPPERS, y el SSH también puede utilizar TCP_WRAPPERS. Esto te permite centralizar el control del acceso a los servicios con facilidad.

Actualiza el software (especialmente el software de red) a las últimas versiones, comprueba las páginas de erratas / seguridad que tengan que ver con tu distribución.

Utiliza un herramienta de integridad de sistema, como L5 o Gog&Magog para establecer una lista y las firmas de los ficheros actuales. Si alguien entra en el sistema, esto te hará la vida más fácil.

Ejecuta pruebas de penetración desde un sitio externo, p. ej., ejecuta nmap / strobe, nessus y herramientas similares contra tu máquina, para asegurarte de que está convenientemente bloqueada. Recuerda, estas herramientas las tienen los chicos malos, de modo que tu también deberías utilizarlas.

Mantén el software / listas de accesos /etc. actualizado. Audita a intervalos regulares tus ficheros de log utilizando herramientas como el Psionic Logcheck.

Si es posible, utiliza herramientas como Pikt, para monitorizar y evitar problemas del sistema (p. ej. usuarios con enormes buzones de correo).

Si es necesaria una auditoría detallada, instala auditd y habilita la auditoría en el kernel de eventos tales como la apertura de ficheros y la ejecución de programas. Asegúrate de disponer de suficiente espacio de disco.

Métodos de compartición de ficheros

He pensado que también daré un breve repaso a los diferentes métodos de compartición de ficheros que existen para Linux, y mostraré los pros y los contras.

SAMBA

El Samba es la mejor opción (orientado al rendimiento, orientado a la seguridad, etc.) para compartir ficheros con máquinas clientes Windows (y también es gratis). Sin embargo no lo recomendaría para compartir ficheros entre máquinas Linux. Haz click aquí para saber más sobre SAMBA.

NFS

El NFS no es muy seguro, pero es fácil de usar (especialmente junto con el demonio de auto mount), y puede ser relativamente seguro si el entorno no es muy hostil. No recomendaría este método para compartir ficheros si la seguridad es un problema (lo cual significa que no se debería utilizar en estaciones sin disco, pero así es la vida). Haz click aquí para saber más sobre NFS.

Coda

Un sistema de ficheros de red avanzado, no muy divertido de implementar.
<http://www.coda.cs.cmu.edu/>

Drall

Un sistema de compartición de ficheros entre máquinas de forma segura basado en https. <http://www.edlund.org/projects/drall/index.html>

AFS

Una aplicación comercial de alto nivel para compartir ficheros en grandes entornos. El FAQ se encuentra disponible en:
<http://www.angelfire.com/hi/plutonic/afs-faq.html>. Una implementación gratuita del cliente para una variedad de Unixes (incluido Linux, por supuesto) se encuentra disponible en: <http://www.stacken.kth.se/projekt/arla/>

Lectores de correo basados en WWW

Una de las mejores soluciones es utilizar un cliente basado en www, los cuales se pueden ejecutar sobre un servidor de www seguro con un mínimo trabajo extra, y añadir la opción de dejar que los usuarios comprueben el correo con seguridad desde lugares desde los que se les haría difícil comprobar su correo (mientras se está de vacaciones por Europa, por ejemplo). Por desgracia, la mayoría de los lectores de correo basados en www apestan, y los buenos cuestan un ojo de la cara.

No comerciales

AtDot

AtDot tiene licencia GNU y está escrito en Perl. Tiene varios modos de operación, lo cual le hace útil para una gran variedad de soluciones de correo (proveedores al estilo de hotmail, PSI's, etc.). Se puede descargar de: <http://www.nodomainname.net/software/atdot/>.

acmemail

<http://www.nodomainname.net/software/atdot/>

IMHO

<http://www.lysator.liu.se/~stewa/IMHO/>

IMP

IMP necesita el módulo Horde (disponible en el mismo sitio) y un servidor de correo con soporte para PHP3. Se puede descargar IMP y Horde desde: <http://www.horde.org/imp/>

TWIG

<http://twig.screwdriver.net/>

WebMail

<http://webmail.wastl.net/>

Comerciales

Coconut WebMail Pro

<http://www.coconutsoftware.com/>

DmailWeb

<http://netwinsite.com/dmailweb/index.htm>

WebImap

<http://netwinsite.com/webimap/index.htm>

Autenticación Basada en Red

NIS / NIS +

El NIS y el NIS+ (antiguamente conocido como "páginas amarillas") significa Servicio

de Información de Red. En resumen, el NIS y el NIS+ proporcionan una forma de distribuir ficheros de contraseñas, ficheros de grupos y otros ficheros de configuración a lo largo de muchas máquinas, proporcionando sincronización de cuentas y contraseñas (entre otros servicios). El NIS+ es en esencia el NIS con algunas mejoras (la mayoría relativas a seguridad), por lo demás son bastante parecidos.

Para utilizar el NIS se configura un servidor NIS maestro, que contendrá los registros y se le permitirá cambiarlos (añadir usuarios, etc), este servidor puede distribuir los registros a máquinas NIS esclavas que contienen copias de sólo lectura de los registros (pero pueden promocionar a maestro y configurarse de lectura/escritura, si ocurre algo malo). Los clientes de la red NIS solicitan porciones de la información y la copian directamente a sus ficheros de configuración (como /etc/passwd), de modo que puedan estar accesibles localmente. Utilizando NIS se les proporciona a varios miles de estaciones de trabajo y servidores del mismo conjunto de nombres de usuario, información de usuario, contraseñas y similar, reduciendo significativamente las pesadillas de administración.

Sin embargo, esto es parte del problema: al compartir esta información, se hace accesible a atacantes. El NIS+ intenta resolver esta circunstancia, pero es una auténtica pesadilla de configurar.

Una estrategia alternativa sería utilizar algún tipo de VPN (como FreeS/WAN, ¿a que parece que resuelve casi cualquier problema?) y cifrar los datos antes de que lleguen a la red. Existe un howto de NIS / NIS+ en:

<http://metalab.unc.edu/LDP/HOWTO/NIS-HOWTO.html>, y O'Reilly tiene un libro excelente al respecto. El NIS / NIS+ se ejecuta sobre RPC, el cual utiliza el puerto 111, ambos tcp y udp. Definitivamente, esto tendría que bloquearse en el perímetro de la red, pero no protegerá totalmente al NIS / NIS+. Puesto que NIS y NIS+ son servicios basados en RPC, tienden a utilizar los puertos más altos (p. ej., los superiores al 1024) de forma bastante aleatoria, haciendo bastante difícil el filtrado mediante el cortafuegos. La mejor solución es situar el/los servidores NIS en una red interna que esté completamente bloqueada a Internet, hacia dentro y hacia fuera. Hay un excelente documento sobre la forma de asegurar NIS disponible en: <http://www.eng.auburn.edu/users/doug/nis.html>

```
ipfwadm -I -a accept -P udp -S 10.0.0.0/8 -D 0.0.0.0/0 111
```

```
ipfwadm -I -a accept -P udp -S un.host.fiable -D 0.0.0.0/0 111
```

```
ipfwadm -I -a deny -P udp -S 0.0.0.0/0 -D 0.0.0.0/0 111
```

o

```
ipchains -A input -p udp -j ACCEPT -s 10.0.0.0/8 -d 0.0.0.0/0 111
```

```
ipchains -A input -p udp -j ACCEPT -s un.host.fiable -d 0.0.0.0/0 111
```

```
ipchains -A input -p udp -j DENY -s 0.0.0.0/0 -d 0.0.0.0/0 111
```

SRP

El SRP es un recién llegado, relativamente, sin embargo tiene algunas ventajas sobre los programas antiguos. El SRP es gratuito para uso no comercial, y no utiliza cifrado por sí mismo para asegurar los datos, de modo que la exportación fuera de EE.UU. no es un problema. El SRP utiliza hash de un sólo sentido y proporciona autenticación a ambas partes. La desventaja es que el SRP sólo cifra el login (nombre de usuario y contraseña) de modo que cualquier dato transferido (como la sesión telnet o los sitios ftp) son vulnerables. SRP se puede conseguir en: <http://srp.stanford.edu/srp/>. En la actualidad, el SRP tiene soporte para Telnet y FTP (también para windows) aunque habilitar SRP para otros protocolos es relativamente sencillo.

Kerberos

Kerberos es un moderno sistema de autenticación de red basado en la idea de expedir un ticket a un usuario una vez que se ha autenticado en el servidor Kerberos (similar al uso de testigos en NT). Kerberos se encuentra disponible en: <http://web.mit.edu/kerberos/www/>. El FAQ de Kerberos está en: <http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html>. Kerberos es adecuado para instalaciones grandes, pues escala mejor y es más seguro que NIS / NIS+. "Kerberizar" programas como telnet, imap y pop es posible, con algo de esfuerzo, sin embargo es más difícil encontrar clientes Windows con soporte para Kerberos.

[Guía de Seguridad del Administrador de Linux - GSAL]

Software de Listas de correo

No comercial

SmartList

<http://www.procmail.org/>

Majordomo

<http://www.greatcircle.com/majordomo>

Minordomo

<http://www.nodomainname.net/software/minordomo/>

Sympa

<http://listes.cru.fr/sympa>

Listar

<http://www.listar.org>

Herramientas de escaneo y detección de intrusos

Si la sección anterior te tiene preocupado, así debería ser. Sin embargo existen muchas defensas, activas y pasivas, contra esos tipos de ataques. Las mejores formas de combatir escaneos de red son mantener el software actualizado, sólo ejecutar lo que se necesite, y restringir fuertemente lo demás mediante el uso de cortafuegos y otros mecanismos.

Por suerte, en Linux estas herramientas son gratis y se encuentran fácilmente disponibles, de nuevo sólo me dedicaré a las herramientas de código abierto, puesto que la idea de un cortafuegos propietario es bastante preocupante. La primera línea de defensa debería ser un cortafuegos robusto, seguido de filtros de paquetes en todas las máquinas accesibles desde Internet, uso liberal de TCP-WRAPPERS, logs y lo más importante, software automatizado para que examine los logs en tu lugar (hoy en día para un administrador es impracticable que pueda leer los ficheros de log).

Herramientas para logs

Psionic PortSentry

El tercer componente de la suite Abacus, detecta y guarda un log de los escaneos de puertos, incluyendo escaneos clandestinos (stealth) (básicamente debería ser capaz de detectar cualquier cosa que sea posible hacer con Nmap). Se puede configurar el Psionic Portsentry para que bloquee la máquina atacante (en mi opinión es una mala idea, pues se podría utilizar para generar un ataque de denegación de servicio en hosts legítimos), haciendo difícil el completar un escaneo de puertos. Puesto que esta herramienta está en fase beta, no recomendaría su uso, sin embargo, con el tiempo debería madurar hasta convertirse en una herramienta sólida y útil. Psionic Portsentry se encuentra disponible en: <http://www.psionic.com/abacus/portsentry/>

Detección de ataques basada en Host

Cortafuegos

La mayoría de los cortafuegos soportan guardar logs de los datos, y el ipfwadm/ipchains no son una excepción, utilizando la opción -l se debería generar una entrada en el syslog para cada paquete, utilizando filtros automatizados (para esto es bueno el Perl) se pueden detectar tendencias/ataques hostiles, etcétera. Puesto que la mayoría de los cortafuegos (basados en UNIX, y Cisco en cualquier caso) guardan un log vía syslog, se puede centralizar todo el log de paquetes del cortafuegos en un único host (con mucho espacio en disco duro).

TCP-WRAPPERS

El TCP-WRAPPERS de Wietse te permite restringir las conexiones a varios servicios basándose en direcciones IP, pero incluso más importante es el hecho de que te permite configurar una respuesta, se puede hacer que te envíe un correo, haga finger a la máquina atacante, etcétera (sin embargo, hay que utilizarlo con cuidado). El TCP-WRAPPERS viene standard con la mayoría de las distribuciones y está disponible en: <ftp://ftp.porcupine.org/pub/security/>

Klaxon

Si bien ha quedado obsoleto por el TCP-WRAPPERS y los logs de los cortafuegos, Klaxon todavía puede ser útil para detectar escaneos de puertos, si no se quiere bloquear por completo la máquina. Se encuentra disponible en:

<ftp://ftp.eng.auburn.edu/pub/doug>

Psionic HostSentry

Aunque este software todavía no está listo para el consumo en masa, he pensado que lo mencionaría de todas formas, pues forma parte de un proyecto más grande (el proyecto Abacus, <http://www.psionic.com/abacus/>). En resumen, el Psionic HostSentry construye un perfil de accesos del usuario y después lo compara con la actividad actual, para resaltar cualquier actividad sospechosa. Se encuentra disponible en: <http://www.psionic.com/abacus/hostsentry/>

Pikt

El Pikt es una herramienta extremadamente interesante, en realidad es más un trozo de lenguaje script dirigido a la administración de sistemas que un simple programa. El Pikt te permite hacer cosas como matar procesos libres (idle) de usuarios, reforzar las cuotas de correo, monitorizar el sistema en busca de patrones de uso sospechosos (fuera de horas, etc) y mucho más. El único problema de Pikt es la empinada curva de aprendizaje de las herramientas, puesto que utiliza su propio lenguaje de scripts, pero creo que dominar este lenguaje te recompensará si se tienen muchos sistemas que administrar (especialmente debido a que actualmente Pikt se ejecuta en Solaris, Linux y FreeBSD). El Pikt se encuentra disponible en: <http://pikt.uchicago.edu/pikt>

Detección de ataques basada en red

NFR

El NFR (Registro de Vuelo de Red, Network Flight Recorder) es mucho más que un sniffer de paquetes, en realidad guarda un log de los datos y detecta en tiempo real los ataques, escaneos, etcétera. Es una herramienta muy potente y para ejecutarse requiere una significativa inversión de tiempo, energías y potencia de máquina, pero está en la cima de la cadena alimenticia en cuanto a detección. El NFR está disponible en: <http://www.nfr.com/>

Documentos de Detección de Intrusos

FAQ: Sistemas de Red de Detección de Intrusos, un FAQ excelente que se ocupa de los principales (y otros menores) asuntos relativos a los sistemas IDS. Disponible en: <http://www.robertgraham.com/pubs/network-intrusion-detection.html>

Escaneo / herramientas de prueba de intrusos

Durante los últimos años ha crecido de forma dramática el número de herramientas de seguridad para Windows y UNIX, siendo incluso más sorprendente el hecho de que la mayoría de ellas estén disponibles gratuitamente en Internet. Sólo me ocuparé de las herramientas gratuitas, puesto que la mayoría de las herramientas comerciales son ridículamente caras, no son código abierto, y en muchos casos han demostrado incluir graves fallos (como almacenar las contraseñas en texto plano después de la instalación). Cualquier cracker/hacker serio tiene estas herramientas a su disposición, de modo que ¿por qué no tenerlas tú?

Existen diferentes categorías principales de herramientas, las que escanean hosts desde el mismo host, las que escanean otros hosts e informan acerca del tipo de SO que están ejecutando (utilizando una técnica llamada huella TCP-IP), los servicios disponibles, etc., estando en la cima de la cadena alimenticia las herramientas de intrusión que intentan ejecutar exploits, e informan acerca de los que funcionaron y los que no, y finalmente incluyo la categoría de exploits, que si bien no son una herramienta de intrusión per se, existen y se debería tener conocimiento de ellos.

Escaneo de Hosts

Cops

El Cops es extremadamente obsoleto y su lugar habitual del ftp del CERT ha desaparecido.

Tiger

El Tiger es obsoleto, pero he pensado mencionarlo por exactitud histórica, la Universidad de Agricultura y Mecánica de Texas solía requerir que un host UNIX pasara por el tiger antes de que se le permitiese conectar a la red desde fuera. Se puede conseguir en: <ftp://net.tamu.edu/pub/security/TAMU/>

SBScan

El SBScan es un scanner basado en host, busca una variedad de problemas como ficheros rhosts malignos, puertos abiertos, cuentas de contraseñas, y escanea la red buscando otro tipo de inconveniencias. El SBScan ya no se encuentra en desarrollo, pero está disponible en: <http://www.hagd.demon.co.uk/>

check.pl

check.pl es un interesante programa en perl que comprueba los permisos de ficheros y directorios, e informará acerca de cualquier cosa sospechosa o de los "malos" (setuid, setgid, directorios con permiso de escritura, etc.). Muy útil pero tiende a encontrar un montón de falsos positivos. Se encuentra disponible en: <http://opop.nols.com/proggie.html>

Scanners de Red

Strobe

El Strobe es una de las más veteranas herramientas de escaneo de puertos, simplemente se intenta conectar a varios puertos de una máquina(s) e informa del resultado (si existe). Es simple de utilizar y muy rápido, pero no tiene ninguna de las características que tienen los nuevos escaners de puertos. El Strobe se encuentra disponible desde la mayoría de las distribuciones, como

parte de la misma, o como un paquete contribuido, los fuentes se encuentran disponibles en: <ftp://suburbia.net/pub/>

Nmap

El Nmap es una herramienta de escaneo más reciente y con más características. Tiene técnicas avanzadas, como las huellas TCP-IP, un método por el cual se examinan los paquetes TCP-IP devueltos, y se deduce el SO del host basándose en diferentes peculiaridades presentes en todas las pilas TCP-IP. El Nmap soporta un número de métodos de escaneo, desde los escaneos normales de TCP (simplemente tratar de abrir una conexión como es habitual) hasta escaneos clandestinos (stealth scanning) y escaneos SYN semi-abiertos (fenomenales para cascar pilas TCP-IP inestables). Este es indiscutiblemente uno de los mejores programas de escaneo de puertos disponibles, ya sea comercial o de cualquier otro tipo. Nmap se encuentra disponible en: <http://www.insecure.org/nmap/index.html>. También existe un interesante artículo disponible en: <http://raven.genome.washington.edu/security/nmap.txt> sobre nmap y acerca del uso de sus características más avanzadas.

MNS

<http://www.thegrid.net/gravitino/products.html>

Bronc Buster vs. Michael Jackson

<http://www.thegrid.net/gravitino/products.html>

Leet scanner

<http://www.thegrid.net/gravitino/products.html>

Soup scanner

<http://www.thegrid.net/gravitino/products.html>

Network Superscanner

http://members.tripod.de/linux_progz/

Portscanner

Portscanner es un pequeño escaneador de puertos (¡qué sorpresa!) que tiene diferentes niveles de salida, lo cual le hacen sencillo de utilizar en scripts y por humanos. Es código abierto y de uso gratuito, se puede conseguir en: <http://www.ameth.org/~veilleux/portscan.html>

Queso

El Queso no es un scanner per se, pero te dirá con un grado de exactitud bastante alto el SO que está utilizando un host remoto. Utilizando una variedad de paquetes tcp válidos e inválidos para probar el host remoto, compara la respuesta con una lista de respuestas conocidas para diferentes sistemas operativos, y te dirá qué SO está ejecutando el host remoto. Se puede conseguir desde: <http://www.apostols.org/projectz/queso/>

Escáners de Intrusos

Nessus

Nessus es relativamente nuevo pero rápidamente se está perfilando como una de las mejores herramientas de escaneo de intrusos. Tiene una arquitectura

cliente/servidor, el servidor se ejecuta en Linux, FreeBSD, NetBSD y Solaris, y los clientes están disponibles para Linux, Windows y hay un cliente Java. La comunicación entre el servidor y el cliente va cifrada, para mayor seguridad, todo en un hábil trozo de código. El Nessus soporta escaneo de puertos, y ataques, basado en direcciones IP o nombres de host(s). También puede buscar a través de la información DNS de la red, y atacar hosts los relativos, según tu interés. Nessus es relativamente lento en modo ataque, lo cual no es sorprendente. Sin embargo, en la actualidad cuenta con 200 ataques y un lenguaje de plug-ins, de forma que puedas escribir los tuyos propios. Está disponible en: <http://www.nessus.org/>

Saint

Saint es el sucesor de Satan, un escáner de seguridad de red hecho famoso por los medios de comunicación hace unos años (había serias preocupaciones de que los malos acabaran con Internet haciendo uso de el). Saint también utiliza una arquitectura cliente /servidor, pero utiliza un interfaz www en lugar de un programa cliente. El Saint produce una salida muy fácil de leer y entender, graduando por prioridad los problemas de seguridad (aunque no siempre de forma correcta) y también soporta módulos de escaneo añadidos, lo cual le hace muy flexible. Saint está disponible en: <http://www.wwdsi.com/saint/>

Cheops

Si bien no es un escáner per se, es útil para detectar el SO de un host y manejar un gran número de hosts rápidamente. El Cheops es un "entorno de red" con esteroides, construye una imagen de un dominio, o bloque IP, que está ejecutando cada host y así sucesivamente. Es extremadamente útil para preparar un escaneo inicial, pues se pueden localizar elementos interesantes (Impresoras HP, routers Ascend, etc) con rapidez. El Cheops está disponible en: <http://www.marko.net/cheops/>

Ftpcheck / Relaycheck

Dos sencillas utilidades que escanean en busca de servidores ftp y de correo que permitan retransmisión, bueno para hacerse una idea de qué usuarios molestos están instalando servicios que no deberían (o simplemente desconfigurándolos), disponible desde: <http://david.weekly.org/code/>

SARA

Asistente de Búsqueda para el Auditor de Seguridad, Security Auditor's Research Assistant, es una herramienta de funciones similares a las de SATAN y Saint. El SARA soporta múltiples hilos para escaneos más rápidos, guarda sus datos en una base de datos para facilidad de acceso y genera interesantes informes en HTML. SARA es de uso gratuito y está disponible en: <http://home.arc.com/sara/>

BASS

El BASS, "Escáner de Seguridad de Auditoría Masiva", "Bulk Auditing Security Scanner", te permite escanear internet en busca de una variedad de exploits bien conocidos. En esencia, fue una prueba del concepto de que Internet no es seguro. Se puede conseguir en: <http://www.securityfocus.com/data/tools/network/bass-1.0.7.tar.gz>

Escáners de Cortafuegos

Firewalk

Firewalk es un programa que utiliza un estilo similar al traceroute para escanear un cortafuegos e intentar deducir las reglas impuestas en ese

cortafuegos. Al enviar paquetes con diferentes tiempos de vida y ver dónde mueren o si son rechazados, se puede engañar al cortafuegos para que revele sus reglas. No existe una defensa real contra esto, aparte de denegar silenciosamente los paquetes en lugar de enviar un mensaje de rechazo, lo cual con suerte revelará menos cosas. Te recomendaría utilizar esta herramienta para escanear tus sistemas, pues los resultados te pueden ayudar a reforzar la seguridad. El Firewall se encuentra disponible en:
<http://www.packetfactory.net/firewalk/>

Exploits

No me voy a dedicar a tratar los exploits específicamente, puesto que existen cientos, si no miles de ellos rondando por ahí. Simplemente indicaré los principales archivos:

<http://www.rootshell.com/>

Uno de los principales archivos en cuanto a exploits, tiene casi cualquier cosa y de todo, un motor de búsqueda adecuado y generalmente exploits completos.

Sniffers de Paquetes

El sniffing de paquetes es la práctica de capturar datos de red que no están destinados a tu máquina, generalmente con el propósito de ver tráfico confidencial/sensible, como sesiones telnet o gente leyendo su correo. Por desgracia no existe una forma de detectar un sniffer de paquetes, puesto que es una actividad pasiva, sin embargo mediante la utilización de switches de red y backbones de fibra óptica (que son muy difíciles de pinchar) se puede minimizar la amenaza.

tcpdump

El abuelito de los sniffers de paquetes para Linux, esta herramienta ha existido desde que me es posible recordar, y su uso principal es hacer un debug de problemas de red. No es muy configurable y carece de las características avanzadas de los más novedosos sniffers de paquetes, pero todavía puede ser útil. La mayoría de las distribuciones vienen con tcpdump.

sniffit

Mi sniffer de paquetes favorito, es muy robusto, tiene interesantes posibilidades de filtrado, convierte la carga de los paquetes en texto ASCII para una fácil lectura (como las sesiones telnet), e incluso tiene un modo gráfico (interesante para monitorizar de forma general la actividad y conexiones). Se encuentra disponible en:

<http://sniffit.rug.ac.be/~coder/sniffit/sniffit.html>

Ethereal

Un analizador de protocolos con buena apariencia (alias, un sniffer trucado) con una interfaz muy similar al monitor de red de NT. Permite una sencilla vista de la carga de los paquetes para la mayoría de protocolos de red (tftp, http, Netbios, etc). Está basado en GTK, lo cual significa que probablemente habrá que tener el gnome ejecutándose para utilizarlo. Todavía no lo he probado (aunque tengo intenciones). Se encuentra disponible en: <http://ethereal.zing.org>

Snort

El Snort es una interesante herramienta de para sniffing de paquetes, que también se puede utilizar para detectar diferentes ataques. Puede vigilar actividades como escaneos mediante huellas TCP-IP con el Queso, escaneos con Nmap y similares. Se encuentra disponible en:

<http://www.clark.net/~roesch/security.html>

SPY

SPY es un sniffer multiprotocolo avanzado que se ejecuta en diferentes plataformas. No es un programa gratuito, sin embargo existe una licencia para un sólo usuario para uso no comercial, con un máximo de 5 hosts. El coste comercial es de alrededor de 6000\$ dólares americanos, pero echándole un rápido vistazo a sus características, yo diría que merece la pena si lo que se necesita es un sniffer industrial. Se puede conseguir en:

<http://pweb.uunet.de/trillian.of/SpY/>

Otros sniffers

Existe toda una variedad de sniffers para Linux, basados en la librería libpcap entre otras, he aquí una pequeña lista:

<http://www.mtco.com/~whoop/ksniff/ksniff.html> - KSniff

<http://ksniffer.veracity.nu/> - Ksniffer

<http://mojo.calyx.net/~btx/karpski.html> - karpski

<http://www.ozemail.com.au/~peterhawkins/gnusniff.html> - Gnusniff

<http://elektra.porto.ucp.pt/snmpsniff/> - SNMP Sniffer

<http://www.xnet.com/~catchmike/mike/Software/> - ipgrab

AntiSniff

Como ya se mencionó anteriormente, AntiSniff es una herramienta que prueba dispositivos de red para ver si se están ejecutando en modo promiscuo, al contrario que los modos normales de operación. Se supone que es efectivo, y funcionará con la mayoría de sniffers. Se puede conseguir en:

<http://www.l0pht.com/antisniff>

Normas de comportamiento / integridad de ficheros

Una de las cosas que más se suelen pasar por alto por mucha gente que administra sistemas es olvidarse de crear una línea de fondo del sistema, es decir, un perfil del sistema, el uso de sus recursos, etcétera. Por ejemplo, algo tan simple como un "netstat -a -n > netstat-output" puede darte una referencia para comprobar más tarde y ver si están presentes algunos puertos abiertos que no lo deberían estar. El uso de memoria y de disco también son un par de cosas sobre las que echar un vistazo. Un incremento repentino del uso de la memoria podría dar como resultado que el sistema viese consumidos sus recursos. Lo mismo en cuanto al uso del disco. Podría ser un accidente de un usuario, un usuario malicioso, o un programa gusano que ha comprometido el sistema y ahora está escaneando otros sistemas. Existen diferentes herramientas para medir el uso de disco y de memoria: vmstat, free, df, du, todos los cuales vienen desarrollados por sus páginas del manual respectivamente.

Como mínimo haz una copia de seguridad completa del sistema, y regularmente haz copias de seguridad de los ficheros de configuración y logs, lo cual también puede ayudar a descubrir cuándo se ha producido una intrusión (la cuenta de usuario "rewt" se añadió después de la copia del 4 de Abril, pero no está en la copia del 20 de Marzo). Una vez que el sistema se ha visto comprometido, por lo general se suele instalar un rootkit, que consiste en binarios con troyanos, y es casi imposible de eliminar de forma segura, es mejor formatear el disco duro y empezar desde cero. Por supuesto hay una notable excepción a esta regla, si se fue diligente y se utilizaron herramientas de integridad de ficheros / directorios tales como L5, se sería capaz de descubrir los ficheros afectados con facilidad y tratar con ellos.

Tripwire

El Tripwire ya no es una herramienta de código abierto. No tengo absolutamente NINGÚN problema con el software comercial. Sin embargo, cuando se espera de mi que confíe en un programa para proporcionar seguridad, cuando ni yo ni nadie puede ver el código fuente (está disponible bajo algún tipo de licencia especial, probablemente un NDA) debo declinar. El Tripwire cuesta aproximadamente 70\$ para Linux, y sólo está disponible como paquete RPM, destinado a Red Hat Linux (tripwire cuesta 500\$ para otros sistemas operativos). Opino que está en la parte alte para un tipo de software que se puede reemplazar con facilidad por alternativas como el L5 o el Gog&Magog. Tripwire está disponible en: <http://www.tripwiresecurity.com>

AIDE

El AIDE es un reemplazo del tripwire que intenta ser mejor que el tripwire. Tiene licencia GPL, lo cual le hace más deseable que el tripwire, desde un punto de vista de fiabilidad. Soporta varios algoritmos de hashing, y se puede descargar desde: <http://www.cs.tut.fi/~rammer/aide.html>

L5

Sin embargo, hay una alternativa al tripwire, el L5, disponible en: <ftp://avian.org/src/hacks/>, que es completamente gratuito y muy efectivo. Definitivamente, recomendaría utilizar esta herramienta.

Gog&Magog

Gog&Magog crea una lista de propiedades del sistema de ficheros, propietario, permisos, una firma MD5 del fichero y similar (parecido al tripwire). Se puede hacer que compare automáticamente esto y se asegure de que cualquier fichero

que haya sido cambiado capte de inmediato tu atención. De igual forma, hace que la recuperación a partir de una irrupción sea más simple, puesto que se conocerá qué ficheros han sido comprometidos. Se puede descargar desde:
<http://www.multimania.com/cparisel/gog/>

nannie

nannie es una herramienta relativamente simple, que se sirve de stat para construir una lista de cómo deberían ser los ficheros (tamaño, timestamp, etc). Crea una lista que contiene el nombre de fichero, el ínodo, información de enlace, etcétera, es una útil aunque simple alarma. Se puede conseguir en:
<ftp://tools.tradeservices.com/pub/nannie/>

confcollect

confcollect es un simple script que recolecta información del sistema como tablas de rutado, rpm's instalados y similar. Se puede descargar desde:
<http://www.skagelund.com/confcollect/>

Copias de Seguridad

Algo de lo que la gente se suele olvidar, pero se pueden comparar los ficheros actuales con las copias de seguridad viejas, y muchos formatos de copias de seguridad (Cinta, floppy, CDR, etc.) se pueden hacer de sólo lectura, de forma que una copia de seguridad de los sistemas recién instalados proporciona un buen banco de pruebas con el que comparar las cosas. Se pueden utilizar la utilidad "diff" y "cmp" para comparar ficheros entre sí. Mira la sección de copias de seguridad para un listado de software gratis y comercial.

Gestión de auditorías

De forma que has asegurado las máquinas, y has hecho todas las cosas que es necesario hacer. De modo que ¿cómo te aseguras de que en realidad está haciendo lo que se supone que debe hacer, o cómo probarle a alguien que es tan seguro como dices que es? Pues llevando a cabo una auditoría. Puede ser algo tan simple como revisar el software instalado, ficheros de configuración y demás, o tan complejo como alquilar un tiger team (o hackers éticos, o cualquier otra palabreja) para intentar activamente entrar y penetrar en tu seguridad. Si no pueden, hiciste un buen trabajo (o si no apestan), y si entran, ya sabes que es lo que hay que arreglas (es un buen método para enseñarle al jefe que la seguridad no es un asunto a corto plazo, sino que es una batalla constante).

También existen muchas herramientas gratuitas y técnicas que se pueden utilizar para llevar a cabo tú mismo una auditoría y asegurarte de que los sistemas reaccionan como crees que deberían (todos cometemos errores, pero cazarlos rápidamente y corregirlos es parte de lo que te convierte en un gran administrador). Herramientas tales como nmap, nessus, crack, etcétera, se pueden emplear de forma rápida para escanear la red y los hosts, encontrando de forma rápida cualquier problema obvio. También sugeriría echarle un vistazo cada cierto tiempo a los ficheros de configuración (en cuanto a mi, intento "visitar" cada servidor una vez al mes, a veces descubro algún pequeño error, o algo que olvidé configurar anteriormente). Mantener los sistemas en un relativo estado de sincronización (acabo de terminar de trasladar a TODOS mis clientes al Kernel 2.2.x, con ipchains) te ahorrará una buena cantidad de tiempo y energías.

Utilizando las herramientas mencionadas anteriormente en la sección "Normas de comportamiento" se pueden comprobar la integridad de los ficheros utilizando tripwire, L5, copias de seguridad u otros métodos. Otra herramienta que es útil para comprobar binarios es el comando "strings", que muestra información legible de los ficheros binarios, y es especialmente útil si alguien olvidó ejecutar "strip" en sus binarios después de compilarlos (la gente ha tenido suerte y ha conseguido el directorio desde el que se compiló el exploit, permitiéndoles trazar el usuario exacto).

Copias de Seguridad

No recuerdo cuantas veces se lo he dicho a la gente, pero no deja de impresionarme cuántas veces le sorprende a la gente que si no hacen copia de seguridad de sus ficheros los perderán, si el disco duro casca o si aprietan la tecla "borrar" sin pensar. Siempre ten copias de seguridad del sistema, incluso si sólo son los ficheros de configuración, a la larga te ahorrarás tiempo.

Para hacer una copia de seguridad de tus datos bajo Linux existen muchas soluciones, todas tienen sus pros y sus contras. También existen diferentes programas de copia de seguridad industriales, los mejores soportan copias de seguridad en red, los cuales son definitivamente una ventaja en entornos grandes no homogéneos.

Tar y Gzip

Los viejos rockeros nunca mueren, tar y gzip. ¿Por qué? Porque al igual que el vi, puedes apostar por el hecho de que cualquier sistema UNIX tendrá tar y gzip. Pueden ser lentos, cutres y empezar a enseñar su edad, pero son una herramienta universal que harán su trabajo. Me he dado cuenta de que con Linux, la instalación de un sistema típico suele llevar entre 15-30 minutos, dependiendo de la velocidad de la red/cdrom, la configuración otros 5-15 minutos (suponiendo que tenga copias de seguridad o que sea muy simple) y la restauración de datos lleva lo que lleva (definitivamente no es algo en lo que uno debería apresurarse). Un buen ejemplo: recientemente hice una copia de seguridad de un servidor y acto seguido procedí a cargarme el sistema de ficheros (y eliminar físicamente 2 discos duros que ya no necesitaba), después instalé Red Hat 5.2 y reconfiguré 3 tarjetas de red, Apache (para cerca de 10 sitios virtuales), Bind y algunos otros servicios en 15 minutos. Si lo hubiese hecho desde cero me hubiera llevado varias horas. Simplemente:

```
tar -cvf nombre-de-archivo.tar dir1 dir2 dir3....
```

para crear un tarball de tus ficheros favoritos (por lo general /etc, /var/spool/mail, /var/log, /home y cualquier otros datos de usuarios/sistema), seguido de un:

```
gzip -9 nombre-de-archivo.tar
```

para comprimirlo lo máximo posible (por supuesto que el espacio de disco duro es más barato que la promesa de un político, pero comprimirlo lo hace más fácil de transportar). Quizás preferirías utilizar bzip, que es bastante mejor que gzip comprimiendo texto, pero es algo más lento. Por lo general después hago una copia del archivo en un servidor remoto, ya sea mediante ftp o enviándolo por correo como un attachment si no es grande (p. ej, la copia de seguridad de un cortafuegos suele ser de alrededor de 100kb de ficheros de configuración).

Programas no comerciales de Copias de Seguridad para Linux

Amanda

Amanda es un programa de copia de seguridad cliente/servidor basado en red, con soporte para la mayoría de sistemas UNIX y Windows (vía SAMBA). Amanda tiene licencia al estilo de BSD y está disponible en: <http://www.amanda.org/>

afbackup

Afbackup es otro programa cliente/servidor con una licencia general GPL con una pequeña excepción, el desarrollo de la porción servidor en Windows está

prohibida. Afbbackup tiene soporte de servidor para Linux, HP-UX y Solaris, y tiene clientes para esos y para windows. Se puede descargar en:
<ftp://ftp.zn-gmbh.com/pub/linux>

Burt

Burt es un conjunto de extensiones basadas en Tcl/Tk que permite hacer copias de seguridad de estaciones UNIX fácilmente, lo cual te permite ejecutarlo en casi cualquier sistema. Burt tiene arquitectura cliente/servidor y parece ser bastante escalable, está disponible en: <http://www.cs.wisc.edu/~jmelski/burt/>

Programas comerciales de Copias de Seguridad para Linux

BRU

BRU (Backup and Restore Utility), ha estado en el mundo Linux desde hace tanto tiempo como el Linux Journal (han puesto anuncios desde el principio). Este programa proporciona un conjunto de herramientas relativamente completo de forma unificada, mediante línea de comandos y entorno gráfico (en otras palabras, sencillez de automatizar). Soporta copias de seguridad completas, incrementales y diferenciales, al igual que catálogos, y puede escribir a un fichero o a una cinta, se trata básicamente de un programa sólido, simple y fácil de utilizar. El BRU se encuentra disponible en:
<http://www.estinc.com/features.html>

Quickstart

El Quickstart está destinado a crear una imagen del sistema, de forma que cuando falle el disco duro se pueda cargar de forma rápida un disco en blanco y tener un sistema funcionando. También se puede utilizar para crear un "master" de un sistema y después cargar otros sistemas (como alternativa al kickstart de Red Hat). Tiene un precio bastante razonable y ha recibido buenas críticas en el Linux Journal (Nov 1998, pág. 50). Se puede conseguir en:
<http://www.estinc.com/qsdr.html>

Backup Professional

<http://www.unitrends.com/bp.html>

CTAR

<http://www.unitrends.com/ctar.html>

PC ParaChute

<http://www.unitrends.com/pcpara.html>

Arkeia

Arkeia es un programa de copias de seguridad muy potente, con arquitectura cliente - servidor, que soporta muchas plataformas. Es un producto de potencia "industrial" y apropiado para entornos heterogéneos, se hizo una crítica en el Linux Journal (Abril 1999, pág. 38) y se puede descargar una versión shareware en línea y darle una oportunidad, la URL es: <http://www.arkeia.com>

Legato Networker

El Legato Networker es otro programa de copias de seguridad para la empresa, con clientes gratuitos (aunque sin soporte) para Linux. Está disponible en: http://www.legato.com/Products/html/legato_networker.html y los clientes Linux están disponibles en: <http://feral.com/networker.html>

Ventajas e Inconvenientes de los medios para Copias de Seguridad

Existen más cosas sobre las que poder hacer una copia de seguridad que kilómetros conducir un range rover hasta que casque, pero aquí van algunas de las alternativas más famosas:

| Nombre del soporte | Ventajas | Inconvenientes |
|--------------------|--|--|
| Disco Duro | Es rápido. Es barato. Es bastante fiable. (Entre 3000-4500 pts. por giga) | Puede no ser suficiente, y fallan, generalmente en el peor momento. Son más complicados de sacar fuera del sitio. Sin embargo RAID es una opción viable. Los discos de 20 Gb están a unas 55.000 pts. |
| CDROM | Cualquiera en el mundo desarrollado dispone de una unidad de CDROM. Los medios suelen ser bastante robustos y baratos (Unas 300 pts. por cada 650 pts. o algo así) | Los CDROM's tienen una vida limitada de entre 5-15 años, y no todos los grabables son iguales. Mantenlos alejados de la luz solar, y asegúrate de disponer de un CDROM que los lea |
| Cinta | Es fiable, se pueden comprar cintas GRANDES, carruseles de cintas y robots de cintas, y están poniéndose lo suficientemente baratas como para que cualquiera se pueda permitir tener una | Soporte magnético, vida finita y algunas cintas se pueden dañar con facilidad (tienes lo que pagas), también asegúrate de que las cintas se puedan leer en otra unidad de cinta (por si se quema el servidor...) |
| Disquettes | No estoy bromeando, corren rumores de que algunos todavía los utilizan para hacer copias de seguridad | Es un disquette. ¿Tú qué crees? |
| Discos Zip | Todavía no he conseguido cargarme uno, ni tampoco mi gato. Guardan 100 Megas, lo cual es suficiente para la mayoría de usuarios de una máquina | No todo el mundo tienen una unidad zip, y son soporte magnético. Los modelos IDE y SCSI son aceptablemente rápidos, pero los modelos de puerto paralelo son abismalmente lentos |
| Discos Jazz | Discos extraíbles de 1 o 2 Gb, el mío SCSI da una media de 5 megabytes/segundo de escritura | Mueren. Ya voy por el tercer disco. Tienen tendencia a estropearse si se utilizan muy de continuo. Y no son |

| | | |
|-----------|--|--|
| | | baratos. |
| Syquest | 1.6 Gb, discos sellados, igual que el anterior | Los cartuchos sellados son más fiables. Aunque recientemente la compañía se declaró en quiebra. No hay servicio de garantía. |
| LS120 | 120 Mb, y baratos, ganando en popularidad | Leeeeeeeento. No estoy bromeando. 120 Mb sobre una controladora de disquettes para algo anunciado como "hasta 3-4 veces más rápido que un disquette" |
| Impresora | Muy larga vida. Necesita de un humano standard Mark 1 como dispositivo lector. Útil para mostrar consultas y como material de referencia. No se puede alterar con facilidad. | ¿Quieres volver a introducir un fichero de contraseñas de 4000 entradas? El OCR también es otra opción. |

Enfrentándose a los ataques

Enfrentarse con un ataque depende de diferentes factores, ¿el ataque está en progreso? ¿Has descubierto que el plan de tu empresa se está enviando por el servidor de correo a una dirección de hotmail? ¿Te han llamado para localizar un servidor o un cluster muertos? ¿Cuáles son tus prioridades? ¿Restaurar el servicio? ¿Asegurar que los datos confidenciales están a salvo? ¿Perseguir al/los atacante(s)? Varias cosas a tener en cuenta:

- * La respuesta del administrador dependerá del entorno en que está. El atacante puede haber comprometido las cuentas administrativas, de forma que enviar correo puede no funcionar.
- * La mayoría de los sitios no quiere dar publicidad de sus ataques (ya sean exitosos o no), debido al avergonzamiento potencial y a problemas relativos a las relaciones públicas.
- * La mayoría de los ataques rápidos, ataques de negación de servicio y similares, están falseados. Llegar hasta el atacante es muy difícil y consume muchos recursos.
- * Incluso si todo va bien, existe una posibilidad de que las fuerzas de la ley confisquen tu equipo como prueba, y se queden con él, no es algo que se deba tomar a la ligera.
- * ¿Sabes cómo entró el atacante? (p. ej. si NFR lo registró) Si es así, quizás prefieras corregir los fallos y dejarlo estar.
- * No intentes ignorar los ataques, si bien al mismo tiempo ten en cuenta que existe mucha gente ejecutando ataques basura para hacer perder el tiempo y las energías de los administradores (y posiblemente distraerles de ataques más subrepticios).

Igualmente, antes de enfrentarse un ataque, tendrías que consultar con la política de la empresa. Si no se dispone de una, consulta con el jefe, el departamento legal, etc. También es una buena idea contar con un plan para tratar los ataques (p. ej, el servidor de correo es de prioridad uno, comprobar los servidores de ficheros es prioridad dos, a quién se notifica, etc.) lo cual evitará un montón de problemas a medida que vayan surgiendo (estate preparado). El libro "Practical Unix and Internet Security" se encarga de este tema detalladamente, de modo que no voy a repetirlo. Compra el libro.

Existe un documento excelente acerca de esto, mira el Apéndice D, "Cómo manejar e identificar pruebas de Red".

Ataques de Negación de Servicio

Los ataques DoS (Denial of Service) son un follón en el que no me he querido meter. Los ataques DoS son con mucho los más molestos y los que causan más problemas, puesto que no se pueden bloquear fácilmente; no importa lo rápido que se refresquen las tablas de conexiones SYN o se limite el tiempo de CPU para los usuarios, el suficiente número de ataques a la suficiente rapidez causará penurias. Los ataques locales son los más sencillos de manejar, puesto que una vez que uno se da cuenta de qué cuenta de usuario es la responsable, se puede cancelar. Los ataques DoS remotos suelen provenir de un amplio espectro de direcciones falseadas, haciendo inefectivo el filtrado mediante cortafuegos (o simplemente se enmascaran tras un lugar con el cual necesites comunicarte, como el sitio de un cliente).

Ejemplos de ataques

Sin entrar en demasiado detalle para no ayudar a los del sombrero negro, quiero dar un par de ejemplos de ataques, para mostrar cómo cosas aparentemente inócuas pueden ser problemáticas y otras complicarte la vida.

Ping flooding, (alias smurfing)

El simple hecho de inundar de datos una red es una táctica pasada de moda pero efectiva, que empeora por el hecho de que la mayoría de las redes tienen configuraciones de cortafuegos defectuosas. Haciendo un ping a la dirección de red de una red remota (por ejemplo un PSI de cablemodem) se pueden recibir varios cientos de pings de respuesta por cada paquete que se envíe. Si se falsea la dirección IP y se etiquetan los paquetes salientes como provenientes de una red de alguien que te disguste, se puede hacer que la mala configuración de una red sea la que haga el trabajo sucio y sature a la víctima.

Envenenamiento de caché DNS

Puesto que muchos servicios confían en el buen funcionamiento del DNS, supone una parte de la red interesante para atacar. Alterar la información de los servidores DNS es más sencillo de lo que debería ser, y si se consigue, se pueden introducir datos falsos. Por ejemplo, si le convenciera a tu servidor de nombres que updates.Redhat.com en realidad apuntase a updates.losmalos.com, probablemente conseguiría engañarte para que te bajases e instalases mi software. Por supuesto que esto lo niega el hecho de que Red Hat firma sus paquetes con PGP, pero ¿verificas las firmas? De igual forma, si utilizas una herramienta automatizada, como autorpm, puede ocurrirte sin intervención del usuario, los paquetes comprometidos se descargan y se instalan, y todo lo que tengo que hacer es echarle un vistazo al log de ftp y después explotar los sitios que han hecho download de los paquetes. Si me las apañase para convencer a tu servidor de correo que otraempresa.com es en realidad uno de mis servidores, no sólo podría recibir el correo que le envías a otraempresa.com, sino que podría leer el correo, y quizás volverlo a reenviar con pequeñas modificaciones (como añadirle un 0 de más al precio de tu oferta).

Virus, Caballos de Troya y Gusanos

Linux no es susceptible a los virus de la misma forma que lo son plataformas Dos/Windows o Mac. En UNIX, los controles de seguridad son una parte fundamental del sistema operativo. Por ejemplo, a los usuarios no se les permite escribir de forma promiscua en cualquier dirección de memoria, algo que Dos/Windows y Mac sí permiten.

Para ser justos, existen virus para UNIX. Sin embargo, el único que he visto para Linux se llamaba "bliss", tenía una opción de desinstalación ("--uninstall-please, --desinstalar-porfavor") y debía ejecutarse como root para que fuese efectivo. O citando una vieja frase para UNIX "si no sabes lo que hace un ejecutable, no lo ejecutes como root". Los gusanos perduran más en el mundo UNIX, siendo la primera incidencia la causada por el gusano Internet de Morris, que explotaba una vulnerabilidad en el sendmail. Los gusanos Linux de la actualidad explotan versiones de imapd, sendmail, WU-FTPD y otros demonios. La forma más sencilla es mantenerlos actualizados y no hacer accesibles los demonios a menos que sea necesario. Estos ataques pueden tener bastante éxito, especialmente si encuentran una red de hosts que no está actualizada, pero por lo general su efectividad se desvanece a medida que la gente actualiza sus demonios. En general, no me preocuparía específicamente de estos dos asuntos, y definitivamente no hay necesidad de comprar un software antivirus para Linux.

Los gusanos gozan de una larga y orgullosa tradición en el mundo UNIX, explotando agujeros de seguridad conocidos (generalmente, muy pocos explotan agujeros nuevos/desconocidos) y replicándose pueden exprimir una red. En la actualidad existen diferentes gusanos que están abriéndose paso en máquinas Linux, la mayoría explotando software Bind 4.x e IMAP viejo. Vencerlos es sencillo, como lo es mantener actualizado el software.

Los caballos de Troya también son populares. Hace poco alguien entró en ftp.win.tue.nl y modificó el paquete TCP_WRAPPERS (entre otros) de forma que enviase contraseñas a una cuenta anónima. Se detectó cuando alguien comprobó la firma PGP del paquete y encontró que no estaba autorizada. ¿Cuál es la moraleja? Utiliza software proveniente de sitios fiables, y comprueba la firma PGP.

Desinfección de virus / gusanos / troyanos

Haz copias de seguridad de tus datos, formatea y reinstala el sistema desde medios conocidos. Una vez que el atacante consigue root en un sistema Linux, puede hacer literalmente cualquier cosa, desde comprometer el gcc/egcs hasta cargar interesantes módulos del kernel al arrancar. No confíes en el software no fiable como root. Comprueba las firmas PGP de los ficheros que descargas, etc. Un poco de prevención bloqueará la diseminación de virus, gusanos y troyanos bajo Linux.

La forma más fácil de tratar con virus y similares es utilizar herramientas de integridad de sistema como tripwire, L5 y Gog&Magog, con las cuales se podrá encontrar fácilmente qué ficheros han sido comprometidos y restaurar/reemplazar/actualizarlos. Existen muchos escáners antivirus disponibles para Linux (pero por lo general no existen virus para Linux).

Escáners de virus para Linux

Como ya se ha dicho anteriormente, los virus no son un problema real en el mundo Linux, sin embargo los escáners de virus que se ejecutan en Linux pueden ser útiles. Filtrar el correo u otras formas de contenido en las puertas de

enlace de la red (todo el mundo tiene máquinas Windows) puede proporcionar una línea extra de defensa, puesto que las plataformas que proporcionan defensa contra la amenaza no se pueden ver comprometidas por esa amenaza (ojalá). Quizás también se quiera escanear ficheros almacenados en servidores de ficheros de Linux a los que se accede desde clientes Windows. Por suerte existen bastantes buenos antivirus disponibles para Linux.

Sophos Antivirus

El antivirus Sophos es un escáner de virus comercial que se ejecuta en una variedad de plataformas Windows y UNIX. Es gratuito para uso personal, y es relativamente barato para uso comercial. Se puede conseguir en:
<http://www.sophos.com/>

AntiVir

AntiVir es otro escáner comercial que se ejecuta en una variedad de plataformas Windows y Linux. Se puede conseguir de: <http://www.hbedv.com/>

Escaneo de Correo Electrónico

AMaViS

El AMaViS utiliza software de terceros (como McAfee) para escanear el correo entrante en busca de virus. Se puede conseguir en: <http://aachalon.de/AMaViS/>. Asegúrate de descargar la última versión, las anteriores han tenido compromisos de root. A fecha de 19 de Julio, la última es:
<http://aachalon.de/AMaViS/amavis-0.2.0-pre5.tar.gz>

Sendmail

Utilizar el AMaViS con el sendmail es relativamente simple, tiene un programa llamado "scanmail" que actúa como reemplazo del procmail (generalmente el programa que maneja el reparto local del correo). Cuando llega un correo, en lugar de utilizar el procmail para repartirlo, el Sendmail llama al scanmail, el cual descomprime y descodifica cualquier attachment, y después utiliza un escáner de virus (de tu elección) para escanear los attachments. Si no se encuentra un virus, el correo se reparte como es habitual. Sin embargo, si se encuentra un virus, se envía un correo al emisor, informándole de que han enviado un virus, y se envía un correo al receptor del correo, informándole acerca de la persona que le ha enviado un virus. Las instrucciones se encuentran en: <http://satan.oih.rwth-aachen.de/AMaViS/amavis.html>

Postfix

Puesto que Postfix puede hacer uso del procmail para repartir correo de forma local, en teoría debería funcionar sin ningún problema. En la práctica necesita que se modifiquen algunas cosas para que funcione correctamente. Para habilitarlo, reemplazar la línea de main.cf:

```
mailbox_command = /usr/bin/procmail
```

por la línea:

```
mailbox_command = /usr/sbin/scanmails
```

y reiniciar el postfix. Para que funcione la advertencia local (se envía una advertencia al receptor del mensaje) el nombre de host de la máquina (sundog, servidorcorreo01, etc) tiene que aparecer listado en "mydestino" en main.cf, o si no se reparte la advertencia. Se debería (y por lo general lo hacen la mayoría de los sitios) redireccionar el correo del root a una cuenta de

usuario, utilizando el fichero de alias, o si no las advertencias no le llegarán al root de forma correcta. Por defecto, el correo de "virusalert" también se redirige a root, también habría que redirigir este correo a una cuenta normal de usuario.

Distribuciones seguras de Linux

Existen varios intentos de crear distribuciones "seguras" de Linux, hay dos (kha0S y Secure Linux) que son al estilo OpenBSD, con serios esfuerzos en la auditoría del código. El tercero es algo menos ambicioso, promovido por SANS y VA Research, han cogido Red Hat y han reemplazado/eliminado/añadido varios paquetes y han utilizado configuraciones seguras por defecto, en un esfuerzo por asegurar rápidamente Linux.

Bastille Linux

Bastille Linux es un derivado de Red Hat Linux. Para conocer más cosas:
<http://www.bastille-linux.org/> y se puede descargarlo desde:
<ftp://ftp.bastille-linux.org>

kha0S

El kha0S ha estado en fase de desarrollo durante algún tiempo, e incluye mucho paquetes de software criptográfico. Se puede leer más acerca del mismo en:
<http://www.kha0S.org/> y se puede descargar desde:
<ftp://ftp.replay.com/pub/replay/linux/kha0s>

Secure Linux

El Secure Linux todavía está en las primeras fases de desarrollo, todavía no han escogido ningún nombre ni han sacado ningún software. Para saber más,
<http://www.resseau.nl/securelinux/>

Información específica por Distribuidor / Vendedor

Red Hat

Red Hat Linux 6.0

Durante la instalación de Red Hat 6.0, llegará un momento en que te deje implementar contraseñas con shadow, y contraseñas MD5, ambas están habilitadas por defecto, y es una buena idea dejarlas así. También se debería actualizar el kernel de la Red Hat 6.0, pues sufre de un molesto ataque de negación de servicio (basado en icmp).

SuSE

SuSE Linux 6.1

Uno de los empleados de SuSE (Marc Heuse) ha escrito unas cuantas utilidades interesantes para SuSE Linux, disponibles en: <http://www.suse.de/~marc/> La primera se llama "Harden SuSE" y trata de eliminar objetos punzantes, reforzar los permisos de los ficheros, eliminar demonios, etcétera. El segundo, "SuSE security check" es un conjunto de scripts que comprueban el fichero de contraseñas por limpieza, una vez al mes saca un listado de todos los paquetes instalados, etc.

Caldera

Caldera OpenLinux 2.2

Caldera tiene una instalación gráfica para la 2.2 llamada "lizard", con un buen número de características interesantes. Durante la instalación, te forzará a crear una cuenta de usuario, lo cual con un poco de suerte ayudará a que la gente no haga login como root constantemente. De igual forma, hay una entrada para "sulogin" en el fichero /etc/inittab, lo cual quiere decir que no se puede escribir simplemente "linux single" en el prompt del boot de lilo y volcarte directamente a modo comandos como root, primero hay que introducir la contraseña del root. Sin embargo existen ciertos problemas con la instalación por defecto que será necesario corregir.

inetd.conf

El fichero inetd.conf es el que controla diferentes servicios relativos a Internet, y tiene algunos servicios activados que son peligrosos:

```
echo stream tcp nowait root internal
```

```
echo dgram udp wait root internal
```

```
discard stream tcp nowait root internal
```

```
discard dgram udp wait root internal
```

```
daytime stream tcp nowait root internal
```

```
daytime dgram udp wait root internal
```

```
chargen stream tcp nowait root internal
```

```
chargen dgram udp wait root internal
```

```

gopher stream tcp nowait root /usr/sbin/tcpd gn
shell stream tcp nowait root /usr/sbin/tcpd in.rshd
login stream tcp nowait root /usr/sbin/tcpd in.rlogind
exec stream tcp nowait root /usr/sbin/tcpd in.rexecd
talk dgram udp wait nobody.tty /usr/sbin/tcpd in.talkd
ntalk dgram udp wait nobody.tty /usr/sbin/tcpd in.ntalkd
uucp stream tcp nowait uucp /usr/sbin/tcpd /usr/sbin/uucico -l

```

Todos deberían estar comentados (situando un "#" al principio de cada línea), y reiniciando inetd con "killall -l inetd"

portmap

Uno de los servicios que la gente preferirá desactivar es el portmap, que es utilizado por varios servicios, como el nfs, y tiene un historial de problemas. Desactivarlo en OpenLinux es algo penoso, puesto que se arranca desde el mismo script que inicia el inetd. Se puede eliminar el paquete portmap ("rpm -e portmap") o se puede ir al /etc/rc.d/init.d/inet y editar lo siguiente:

```
NAME1=inetd
```

```
DAEMON1=/usr/sbin/$NAME1
```

```
NAME2=rpc.portmap
```

```
DAEMON2=/usr/sbin/$NAME2
```

```
por:
```

```
NAME1=inetd
```

```
DAEMON1=/usr/sbin/$NAME1
```

```
#NAME2=rpc.portmap
```

```
#DAEMON2=/usr/sbin/$NAME2
```

```
y:
```

```
# Bail out if neither is present
```

```
[ -x $DAEMON1 ] || [ -x $DAEMON2 ] || exit 2
```

```
por:
```

```
# Bail out if neither is present
```

```
[ -x $DAEMON1 ] || exit 2
```

```
y:
```

```
[ -x $DAEMON1 ] && ssd -S -n $NAME1 -x $DAEMON1 -- $INETD_OPTIONS
```

```
[ -x $DAEMON2 ] && ssd -S -n $NAME2 -x $DAEMON2 -- $PORTMAP_OPTIONS
```

por:

```
[ -x $DAEMON1 ] && ssd -S -n $NAME1 -x $DAEMON1 -- $INETD_OPTIONS  
# [ -x $DAEMON2 ] && ssd -S -n $NAME2 -x $DAEMON2 -- $PORTMAP_OPTIONS
```

y después comentar por completo lo siguiente:

```
NFS=""  
  
cat /etc/mtab | while read dev mpoint type foo; do  
[ "$stype" = "nfs" ] && NFS="$mpoint $NFS"  
  
done  
  
if [ -n "$NFS" ]; then  
  
echo -n "Unmounting NFS filesystems: "  
  
POLICY=I # Ignore 'device busy' during shutdown  
[ "$PROBABLY" != "halting" ] && POLICY=1 # exit on 'busy'  
  
for mpoint in $NFS; do  
  
SVIrun S $POLICY "$mpoint" "!$mpoint" \  
umount $mpoint  
  
done  
  
echo "."  
  
fi  
  
amd
```

Otro servicio que se instala por defecto en OpenLinux 2.2 es el demonio Auto Mount (amd). Permite definir directorios y dispositivos en lugares nfs, de forma que se puede definir /auto/cdrom como /dev/cdrom, de forma que cuando se haga un "cd /auto/cdrom" el sistema monte automáticamente el /dev/cdrom como /auto/cdrom con las opciones adecuadas (sólo-lectura, etc.). El servicio amd utiliza un número de puerto semi-aleatorio, normalmente en el rango 600-800. Por supuesto que este servicios es muy útil en una estación de trabajo, les ahorra a los usuarios el tener que montar manualmente cada dispositivo removible cada vez que quieran utilizarlo (siendo el cdrom y el disquette los más habituales). Sin embargo no lo recomendaría para servidores, debido al historial de problemas que ha tenido el amd. Desactivarlo es sencillo, sencillamente hay que eliminar los enlaces simbólicos de "S30amd" a "K70amd".

```
mv /etc/rc.d/rc3.d/S30amd /etc/rc.d/rc3.d/K70amd
```

```
mv /etc/rc.d/rc5.d/S30amd /etc/rc.d/rc5.d/K70amd
```

SSH

Los rpm's del SSH no se encuentran disponibles para OpenLinux 2.2 (quiero decir, no he encontrado ninguno). Desgraciadamente, los rpm's de Red Hat fallan, y los rpm's fuente también fallan al compilarse, el SSH se compila limpiamente desde el código fuente, sin problemas. El código fuente se puede

obtener de: <ftp://ftp.replay.com/pub/replay/crypto/SSH/> . Para empezar con sshd como mínimo es necesario ejecutar el `"/usr/local/bin/sshd"` al arrancar, buscará sus ficheros de configuración dentro de `/etc` y debería iniciarse sin problemas.

Novell

Todavía no he probado el software de Novell, desconozco si existe alguna incidencia.

Actualizaciones

Las actualizaciones de Caldera OpenLinux 2.2 se encuentran disponibles en:

<ftp://ftp.calderasystems.com/pub/openlinux/2.2/current/RPMS/>

TurboLinux

TurboLinux 3.6

El TurboLinux tiene una instalación muy parecida a la de Red Hat, te va guiando a través de consolas basadas en texto y va haciendo preguntas, después el sistema instala los paquetes y hay que configurar algunas cosas (como el X). Existen un par de pequeñas incidencias con TurboLinux que habrá que "reparar", y existen varias utilidades que vienen de forma standard con TurboLinux, que ya pudieran venir incluidas en las otras distribuciones (como sudo).

inetd.conf

El `inetd.conf` de TurboLinux viene de una forma relativamente adecuada, sin embargo ciertos servicios como `rsh` y `rlogin` vienen habilitados por defecto, recomendaría desactivarlos.

```
shell stream tcp nowait root /usr/sbin/tcpd in.rshd
```

```
login stream tcp nowait root /usr/sbin/tcpd in.rlogind
```

```
talk dgram udp wait nobody.tty /usr/sbin/tcpd in.talkd
```

```
ntalk dgram udp wait nobody.tty /usr/sbin/tcpd in.ntalkd
```

Todos estos deberían estar comentados (colocando un `"#"` delante de cada línea), y reiniciando el `inetd` con `"killall -1 inetd"`

inittab

El TurboLinux (como la mayoría de las distribuciones) te deja arrancar en modo monousuario, sin pedir contraseña para acceder al sistema como root. Habrá que poner la palabra `"restricted"` en `lilo.conf` y añadir una contraseña para evitar que la gente arranque el sistema en modo monousuario sin contraseña.

ipchains

El `ipchains` no viene con el CD de instalación, se encuentra en el CD que lo acompaña, o en el sitio <ftp://ftp.turbolinux.com/pub/TurboLinux/tlw-3.6-companion/TurboContrib/RPMS/>.

Por supuesto que recomendaría instalar el `ipchains` y filtrar con el cortafuegos la máquina.

SSH

Los rpm's del SSH no se encuentran disponibles para TurboLinux 3.6. Se aplica

lo mismo que para el párrafo de Red Hat.

Tripwire

Una cosa que viene incluida en el CD que acompaña a TurboLinux es una copia del Tripwire, recomendaría utilizarlo. No estoy seguro del tipo de licencia que tiene (p. ej., si es gratuita exclusivamente para uso no comercial, o una licencia, o qué). Parece ser la versión 1.3 del Tripwire, de modo que no es comercial.

El CD acompañante

Como ya se ha dicho anteriormente, el CD acompañante contiene un montón de utilidades extra (como Tripwire), al igual que:

Amanda (un interesante programa de copias de seguridad)

ipmasqadm (utilizado para hacer redireccionamiento de puertos a nivel de kernel)

ipchains (utilizado para configurar el cortafuegos)

ProFTPD (un servidor ftp mejor que el WuFTPD)

Squid (un servidor ftp y proxy www)

Tripwire (crea valores de checksum de los ficheros y te advierte si cambian)

Actualizaciones

Las actualizaciones del TurboLinux 3.6 (Miami) se encuentran disponibles en:
<ftp://ftp.turbolinux.com/pub/TurboLinux/turbolinux-updates/3.6/>

Debian

Debian 2.1

Aún sin evaluar.

Slackware

Slackware Linux 4.0

Aún sin evaluar.

Información de contacto con vendedores

Esta sección se está actualizando. Si es Vd. un vendedor con algún tipo de producto, por favor rellene lo siguiente y envíemelo. (seifried@seifried.org). Les iré enviando este "cuestionario" a tantos vendedores como pueda, pero por supuesto que no alcanzaré a todos. El formulario se puede encontrar (en formato texto) en: <https://www.seifried.org/lasg/vendor-contact.txt> o en <http://www.seifried.org/lasg/vendor-contact.txt>. Si no se ha recibido uno por correo y se desea aparecer, complételo y envíelo (las instrucciones se encuentran en el formulario).

Caldera OpenLinux

Bugs: bugs@caldera.com
Security: <http://www.caldera.com/news/security/index.html>
Support: support@caldera.com

Debian GNU/Linux

Bugs: bugs@debian.org
Security: <http://www.debian.org/security/>

LinuxCare

Si bien no son un vendedor, proporcionan soporte.

Support: <http://www.linuxcare.com>

Support: 1-888-546-4878

NetMAX

Support: support@netmax.com

Red Hat Linux

Bugs: <http://developer.redhat.com/bugzilla/>
Security: <http://www.redhat.com/support/>
Support: support@redhat.com

Slackware

<ftp://cdrom.com/pub/linux/>

SuSE

Bugs: bugs@suse.com
Security: <http://www.suse.de/security/index.html>
Support: support@suse.com

TurboLinux

Bugs: jht@turbolinux.com
Security: jht@turbolinux.com
Support: support@turbolinux.com
<http://www.turbolinux.com/support/>

Programación segura

Esta guía existe porque Linux y el software que se ejecuta en sistemas Linux está escrito de forma insegura o configurado de forma insegura. Muchos aspectos, como los desbordamientos de buffer, se deben a la mala programación o a la despreocupación. Estos problemas se tornan especialmente malos cuando el software en cuestión tiene setuid para ejecutarse como root, o cualquier otro grupo privilegiado. Existen multitud de técnicas y otras medidas que se pueden tomar para hacer más seguro el software.

FAQ de Programación Segura en UNIX

Este documento se ocupa de una variedad de técnicas para hacer más seguros los programas, al igual que ciertos elementos de bajo nivel, como el trust heredado, la compartición de credenciales, etc. Está disponible en: <http://www.whitefang.com/sup/> y recomiendo su lectura si se tiene previsto programar en Linux (o en UNIX en general).

Programación Segura en Internet

Secure Internet Programming (SIP) es un laboratorio (a falta de una palabra mejor) que estudia la seguridad de los ordenadores, y más específicamente problemas con el código móvil como Java y ActiveX. Están desarrollando un buen número de proyectos interesantes, y muchas publicaciones en línea que suponen una excelente lectura. Si se va a escribir código Java, yo diría que habría que visitar este sitio: <http://www.cs.princeton.edu/sip/>

Apéndice A: Libros y Revistas

Sendmail - <http://www.oreilly.com/catalog/sendmail2/>
Linux Network Admin Guide (NAG) - <http://www.oreilly.com/catalog/linag/>
Running Linux - <http://www.oreilly.com/catalog/runux2/noframes.html>
DNS & BIND - <http://www.oreilly.com/catalog/dns3/>
Apache - <http://www.oreilly.com/catalog/apache2/>
Learning The Bash Shell - <http://www.oreilly.com/catalog/bash2/>
Building Internet Firewalls - <http://www.oreilly.com/catalog/fire/>
Computer Crime - <http://www.oreilly.com/catalog/crime/>
Computer Security Basics - <http://www.oreilly.com/catalog/csb/>
Cracking DES - <http://www.oreilly.com/catalog/crackdes/>
Essential System Administration - <http://www.oreilly.com/catalog/esa2/>
Linux in a nutshell - <http://www.oreilly.com/catalog/linuxnut2/>
Managing NFS and NIS - <http://www.oreilly.com/catalog/nfs/>
Managing Usenet - <http://www.oreilly.com/catalog/musenet/>
PGP - <http://www.oreilly.com/catalog/pgp/>
Practical Unix and Internet Security - <http://www.oreilly.com/catalog/puis/>
Running Linux - <http://www.oreilly.com/catalog/runux2/>
Using and Managing PPP - <http://www.oreilly.com/catalog/umppp/>
Virtual Private Networks - <http://www.oreilly.com/catalog/vpn2/>

Red Hat/SAMS también publica libros interesantes:

Maximum RPM (disponible como documento en postscript en <http://www.rpm.org/>)
Red Hat User's Guide (disponible en HTML en <ftp://ftp.redhat.com/>)

SNMP, SNMPv2 and RMON - W. Stallings (ISBN: 0-201-63479-1)

Revistas:

Linux Journal (por supuesto, mensual)
Sys Admin (artículos inteligentes, mensual)
Perl Journal (quincenal)
Information Security - <http://www.infosecuritymag.com/>

[Guía de Seguridad del Administrador de Linux - GSAL]

Apéndice C: Otra documentación de seguridad sobre Linux

Firewalling and Proxy Server HOWTO

<http://metalab.unc.edu/LDP/HOWTO/Firewall-HOWTO.html>

Linux IPCHAINS HOWTO

<http://metalab.unc.edu/LDP/HOWTO/IPCHAINS-HOWTO.html>

Linux NETFILTER HOWTO

<http://netfilter.kernelnotes.org/>

Linux Security HOWTO

<http://metalab.unc.edu/LDP/HOWTO/Security-HOWTO.html>

Linux Shadow Password HOWTO

<http://metalab.unc.edu/LDP/HOWTO/Shadow-Password-HOWTO.html>

The Linux CIPE + Masquerading mini-HOWTO

<http://metalab.unc.edu/LDP/HOWTO/mini/Cipe+Masq.html>

Firewall Piercing mini-HOWTO

<http://metalab.unc.edu/LDP/HOWTO/mini/Firewall-Piercing.html>

Quota mini-HOWTO

<http://metalab.unc.edu/LDP/HOWTO/mini/Quota.html>

Secure POP via SSH mini-HOWTO

<http://metalab.unc.edu/LDP/HOWTO/mini/Secure-POP+SSH.html>

The VPN HOWTO (using SSH)

<http://metalab.unc.edu/LDP/HOWTO/mini/VPN.html>

Red Hat Knowledge Base

<http://www.redhat.com/cgi-bin/support?faq>

[Guía de Seguridad del Administrador de Linux - GSAL]

Apéndice D: Documentación de seguridad en línea

Bugtraq Archives

<http://www.geek-girl.com/bugtraq/>

CERT Incident Reporting Guidelines

http://www.cert.org/tech_tips/incident_reporting.html

SECURITY RISK ANALYSIS AND MANAGEMENT

<http://www.norman.com/local/whitepaper.htm>

An Introduction to Information Security

<http://www.certicom.com/ecc/wecc1.htm>

Site Security Handbook

<http://sunsite.cnlab-switch.ch/ftp/doc/standard/rfc/21xx/2196>

How to Handle and Identify Network Probes

<http://www.network-defense.com/papers/probes.html>

IANA Port Numbers

http://rlz.ne.mediaone.net/linux/papers/port_numbers

Free Firewall and related tools (large)

http://sites.inka.de/sites/lina/freefire-1/index_en.html

Internet FAQ Consortium (You want FAQ's? We got FAQ's!)

<http://www.faqs.org/>

An Architectural Overview of UNIX Network Security

<http://www.alw.nih.gov/Security/Docs/network-security.html>

The human side of computer security (an article on social engineering)

[http://www.sunworld.com/sunworlden línea/swol-07-1999/swol-07-security.html](http://www.sunworld.com/sunworlden_línea/swol-07-1999/swol-07-security.html)

IBM Redbooks

<http://www.redbooks.ibm.com/>

[Guía de Seguridad del Administrador de Linux - GSAL]

Apéndice E: Sitios de seguridad en general

SecurityPortal, tiene una sección de Linux y mi columna semanal (así que por supuesto es un gran sitio)

<http://www.securityportal.com/>

Open Security Solutions
<http://www.opensec.net/>

SANS
<http://www.sans.org/>

Listas de correo de Seguridad
<http://www.iss.net/vd/mail.html>

Computer Security Information
<http://www.alw.nih.gov/Security/security.html>

8 Little Green Men
<http://www.8lgm.org/>

Robert's Cryptography, PGP & Privacy Links
<http://www.interlog.com/~rguerra/www/>

Open Security Solutions
<http://www.opensec.net/>

[Guía de Seguridad del Administrador de Linux - GSAL]

Apéndice F: Sitios de Linux en general

Linux.com

<http://www.linux.com>

Linux.org

<http://www.linux.org>

Historial de versiones

- * Bosquejo inicial, decisión de la estructura, introducción de información básica procedente de antiguos escritos. 04/01/1999
- * Comprobación general de puntuación, cambios de formato. El libro se ocupa de la mayoría de temas generales y con instrucciones específicas para Red Hat. Añadidos ejemplos de ipchains allí donde se daban reglas de ipfwadm. 04/03/1999
- * Comienzo del Apéndice B (sitios www, etc.) y el glosario. Retocadas algunas secciones, añadidas otras. Creación de la tabla de contenidos. 04/07/1999
- * Limpieza general, añadidas las secciones sobre Encriptación, IPsec y similares. 04/11/1999
- * Herramientas de escaneo y detección de intrusos, sniffing de paquetes, añadidas secciones nuevas. Editadas secciones viejas que necesitaban limpieza. Me deshago de gráficos innecesarios, que eliminan cerca de 200k (actualmente la mitad del total). 04/12/1999
- * Añadidas auditorías, normas de comportamiento, inn, CVS, rsync, añadidas nuevas herramientas en varias secciones (ssh, etc.) 04/16/1999
- * Añadidos algunos títulos de sección nuevos (Autenticación basada en REd, Sistema X Window, PAM, etc.), terminadas otras secciones (PPP, Linuxconf, etc) y añadidos programas de copias de seguridad comerciales. Añadidos un montón de vendedores Linux. También añadidas herramientas como YaST, Super, Linuxconf. 04/19/1999
- * Añadidas algunas secciones (dpkg, tarballs/tgz's, tftp, etc), añadidas más herramientas. En realidad instalé Debian 2.1 y me pasé unos días jugando con él. Ahora falta experimentar con Slackware para la 0.0.9. Pequeña reorganización (Ahora SAMBA tiene su propio apartado, al igual que la conectividad de Novell). 04/22/1999
- * Reorganización masiva, terminadas la mayoría de las secciones. Comprobación general de gramática y ortografía. Añadidos ejemplos consistentes de filtrado de cortafuegos. Añadida la numeración de páginas (vaaaya). Se retrasa el lanzamiento de la 0.1.0. Nueva licencia, limpieza, etc. Nuevas secciones sobre gestión de ataques, tipos de ataques, etc. Lanzamiento oficial al mundo. 04/27/1999
- * Eliminada la sección de consultores de seguridad, posibles conflictos de intereses sin existir haber método para vetarlo. Añadidas un montón de herramientas nuevas. Eliminada la palabra "Red Hat" en multitud de ocasiones para hacer la guía algo más neutral. 05/08/1999
- * Añadidas muchas más herramientas, se empiezan a añadir herramientas comerciales en otras áreas además de la sección de copias de seguridad. Más correcciones de ortografía y gramática. Añadidas nuevas secciones (almacenamiento de contraseñas, ataques de negación de servicio, etc.) y se reorganiza la estructura. Actualizaciones masivas de Squid y herramientas administrativas. Se reescribe la sección de herramientas administrativas. 15/5/1999
- * Se corrige la sección de URL del final, se organizan varias secciones. Se añade la lista de comprobación de conexión a Internet. Se añaden más de

60 programas y aplicaciones nuevas. Se añade el borrado seguro de ficheros, herramientas de gestión de software, varias herramientas de escaneo y detección de pruebas. Nueva sección de documentos listando documentos de seguridad interesantes, y un listado de URL por sección al final. Añadidos nuevos apéndices listando sitios de Linux, sitios de seguridad, documentos y más. 29/5/1999

- * Correcciones de fallos en general, trabajo de edición, añadidos más listados de software y hecha una pequeñas reorganización. Añadida la sección de escaneo de virus, aparte de eso no hay más cambios. 8/6/1999
- * Más edición, se escriben algunas secciones (ProFTPD, Postfix, IPSec, etc). Otra versión de corrección de fallos. 21/6/1999
- * Se termina de editar, se añade bastante a la sección sobre Squid, y se escribe material específico por distribuciones para Caldera OpenLinux 2.2 (me enviaron una copia =). También se añade la estructura de nuevas secciones. 27/6/1999
- * Me arruino y compro PGP para Windows (lo sé, soy malo). Ahora firmo los correos importantes con PGP además de con la firma Verisign. También estoy firmando el LASG con ello. Se escribe la sección sobre Red Hat y se añaden las auto-claves a la sección IPSec. Se añade la sección sobre TurboLinux, material sobre ssh, y una multitud de otras utilidades. 12/7/1999
- * Se arreglan problemas de formato, ya no hay páginas en blanco, etc. 15/7/1999
- * No sé. Me imagino que pequeños cambios. Se me olvidó escribir la entrada en el log de cambios y ahora ya son casi 4 semanas después. 22/7/1999
- * Más parches, algunas correcciones, el material nuevo es una lista de ficheros de configuración importantes, mejor información de contacto con el vendedor, y planeadas varias secciones nuevas. Probablemente la penúltima versión sólo PDF. 22/8/1999

LASG - Enlaces a la versión antigua en PDF

LASG se ha trasladado a : <http://www.securityportal.com/lasg/>

He transformado LASG en HTM, todavía no está terminada. La versión antigua en PDF se ha interrumpido.

LASG 0.1.7 - 180+ páginas de puro contenido.

Haz click en uno de los mirrors para descargarla.

| | |
|---|---|
| USA - http://metalab.unc.edu/lasg/ | Spain - |
| | http://www.grupoaccessus.com/lasg/ |
| * USA - http://www.freek.com/lasg/ | |
| | |
| * USA - http://www.vadep.com/lasg/ | * Poland - |
| | http://www.cool.waw.pl/lasg/ |
| * USA - | |
| http://jezebel.rath.peachnet.edu/lasg/ | * Great Britain - |
| | http://www2.merton.ox.ac.uk/lasg/ |
| * USA - | |
| http://eeyore.cae.wisc.edu/lasg/ | * The Netherlands - |
| | http://www.nl.linux.org/lasg/ |
| * USA - | |
| http://csociety-ftp.ecn.purdue.edu/lasg/ | * The Netherlands - |
| | http://security.pine.nl/lasg/ |
| | |
| * USA - http://mirrors.hotdog.org/lasg/ | * The Netherlands - |
| | http://linux.infra.cx/lasg/ |
| * Brazil - | |
| http://www.supernet.com.br/lasg/ | * Greece - |
| | http://linux.forthnet.gr/lasg/ |
| * Germany - http://ftp.gwdg.de/lasg/ | |
| | * Slovakia - |

| | |
|--|---|
| * Germany - http://www.it-sec.de/lasg/ | http://www.sadman.sk/lasg/ |
| ----- | ----- |
| * Austria - http://info.ccone.at/lasg/ | * Slovenia - |
| ----- | http://www.camtp.uni-mb.si/lasg/ |
| * Denmark- http://sunsite.auc.dk/lasg/ | ----- |
| ----- | * Israel - |
| * Portugal - | http://ths000.tau.ac.il/lasg/ |
| http://mirrors.fct.unl.pt/lasg/ | ----- |
| | * South Africa - |
| | http://www.lantic.net/lasg/ |
| | ----- |
| | * Australia - |
| | http://mirror.aarnet.edu.au/lasg/ |
| | - restricted to within .au. |

+-----+

+
 LASG está disponible como fichero PDF protegido (de momento no quiero que la gente edite LASG) Se puede utilizar el Acrobat Reader de Adobe, o el xpdf con los parches de cifrado.

Se encuentra disponible una lista de correo, envía correo a Majordomo@lists.seifried.org con subscribe lasg-announce en el cuerpo del mensaje (sin las comillas) y quedarás añadido automáticamente. Los archivos de lasg-announce están en <http://www.securityportal.com/lasg/mailling-lists/lasg-announce/>