Instalación y configuración segura de sistemas Unix

Sergio Talens-Oliag

InfoCentre (http://www.infocentre.gva.es)

stalens@infocentre.gva.es

En este curso hablaremos de la instalación y configuración segura de sistemas Unix, centrándonos principalmente en sistemas GNU/Linux y Solaris.

Introducción

En el curso trataremos la instalación y configuración segura de sistemas Unix, centrándonos en sistemas GNU/Linux y Solaris, aunque parte de los contenidos serán aplicables a otros sistemas.

La idea es proporcionar a los alumnos una guía y una serie de referencias para que instalen y configuren los servidores Unix teniendo siempre en cuenta los aspectos relacionados con la seguridad.

El curso se estructurará según los siguientes apartados:

- Seguridad física; donde se hablará de la seguridad de los servidores desde el punto de vista físico.
- El sistema de arranque; comentaremos como funcionan los sistemas de arranque de los PC y las estaciones SPARC y sus problemas de seguridad.
- Elección de la distribución; donde hablaremos de los factores a considerar a la hora de elegir el sistema operativo a instalar.
- Planificación de la instalación; donde se hablará de los datos que debemos conocer antes de una instalación.
- Particionado del disco y elección de tipo de sistema de archivos, donde se comentará que es importante pensar como vamos a organizar el disco y los sistemas de archivos que vamos a emplear, ya que una vez creados suele ser problemático modificarlos.
- El cargador de arranque; donde hablaremos de los *bootloaders* empleados en sistemas PC y de sus características de seguridad.
- Instalación de Debian GNU/Linux; donde se describirán los pasos a seguir para instalar una Debian GNU/Linux 3.0 (Woody).
- Herramientas administrativas y de actualización de Debian; en donde se hablará del formato de paquetes `.deb`, de `APT` y de los mecanísmos de instalación y actualización de Debian.
- Instalación de RedHat 9.
- Herramientas administrativas y de actualización de RedHat; aquí hablaremos del `rpm` y de otras herramientas de esta distribución.
- Instalación de Solaris; donde se describirán los pasos a seguir para instalar Solaris 9.

- Herramientas administrativas y de actualización de Solaris; comentaremos el formato de paquetes del SysV y el sistema de parcheado de Sun.
- Configuración inicial del sistema; en este apartado se describirán algunos procesos a seguir antes de pasar a producción una máquina recién instalada.

En el curso siempre que hablemos de instalación de programas nos referiremos a la instalación de *paquetes* en formato binario, no a la instalación a partir de la compilación de código fuente.

En principio lo haremos así por que el uso de los sistemas de paquetes nos simplifica enormemente el mantenimiento de los sistemas, permitiéndonos controlar perfectamente las versiones instaladas y actualizar de manera sencilla cuando aparecen vulnerabilidades.

Guía rápida

Para los impacientes, incluimos en este apartado una guía rápida de lo que vamos a estudiar en el resto del curso.

El proceso de instalación segura de un sistema Unix cualquiera se podría resumir en los siguientes pasos:

1. Estudio del emplazamiento físico de la máquina, para minimizar riesgos.
2. Determinar para qué se va a emplear la máquina y los servicios que va a proporcionar. Para los servicios que transmitan información no pública o requieran autentificación intentaremos emplear sistemas que cifren la la información importante cuando viaja por la red.
3. Elegir los programas servidores más adecuados a nuestros conocimientos y objetivos (p. ej. en el caso del servidor `smtp`)
4. En función de los programas elegidos, decidiremos qué tipo de sistema operativo y versión vamos a instalar.
5. Revisar el Hardware y la documentación del S. O. para saber si necesitamos controladores de dispositivo adicionales o cualquier otra actualización.
6. Generar un listado de datos de configuración del equipo:
    a. Nombre de la máquina,
    b. Datos de la configuración de la red (dirección y máscara ip, rutas por defecto, direcciones de los servidores de nombres y dominio a emplear),
    c. Esquema de particiones y tipos de sistemas de archivos a emplear,
    d. Sistema de autentificación y contraseña(s),
    e. Usuarios y máquinas que van a tener acceso a los distintos servicios,
7. Una vez tenemos los datos realizamos una instalación mínima del sistema (preferiblemente sin conectarnos a la red),
8. Detenemos todos los servicios innecesarios y desactivamos su arranque al inicio del sistema; si sabemos que no vamos a necesitarlos también es buena idea desinstarlos completamente,
9. Si se va a emplear la red para instalar o actualizar más servicios colocaremos la máquina detrás de un cortafuegos que bloquee las conexiones entrantes o configuraremos el sistema de *firewalling* de nuestro S. O. para que haga lo mismo.
10. Una vez instalado el sistema básico, añadiremos los paquetes necesarios para proporcionar los servicios. Es importante que seamos conscientes de si alguno de los programas que proporcionan servicios necesita conectarse como clientes a otros equipos, por ejemplo para consultar nombres de máquinas o enviar mensajes de correo.

11. Después comprobaremos que tenemos instaladas todas las actualizaciones de seguridad de los paquetes que tenemos en el sistema.
12. Luego configuraremos los distintos servidores, intentando cerrar todos los agujeros de seguridad que pudieran tener, empezando por no usar ninguna configuración por defecto.
13. Empleando el sistema cortafuegos de nuestro servidor, limitaremos la conexiones de entrada al mínimo número de puertos posibles y sólo permitiremos conexiones de salida relacionadas con las entrantes o las que sepamos con certeza que son necesarias (p. ej. las de consultas del DNS o el acceso al *mail relay*).

Una vez hecho todo esto podremos pasar nuestra máquina a producción.

Hay que indicar que una vez instalado el equipo será necesario mantenerlo, en lo relativo a la seguridad y en estrecha relación con la instalación del mismo deberemos:

1. Monitorizar las listas de seguridad en las que se publican vulnerabilidades e instalar las actualizaciones del fabricante que nos afecten.
2. Auditar periódicamente la seguridad del sistema para comprobar que no tiene ninguna vulnerabilidad conocida.

Para realizar estas tareas de manera eficaz necesitaremos conocer las herramientas de gestión de paquetes de los distintos sistemas y las aplicaciones que vayamos a administrar.

En el curso describiremos brevemente como funcionan las distintas herramientas de gestión de paquetes, los sistemas cortafuegos de Linux y Solaris y comentaremos algunas cosas sobre la configuración de los servicios de red, aunque no hablaremos con detalle de ninguno de ellos.

Referencias

Para comodidad del alumno se presenta aquí la lista de referencias empleada en la elaboración del curso, clasificadas por áreas a las que se refieren.

Referencias generales

Seguridad en general:

- *Seguridad en Unix y Redes*, de Antonio Villalón Huerta: Manual sobre seguridad en Unix, se ha empleado fundamentalmente para la primera parte del curso, disponible en http://andercheran.aiind.upv.es/toni/personal/
- *Recomendaciones de seguridad en una instalación de Linux*, de Gunnar Wolf: http://www.gwolf.cx/seguridad/recom_seg_linux/

Sistemas de paquetes:

- *Comparing Linux/UNIX Binary Package Formats* de Joey Hess: http://www.kitenet.net/~joey/pkg-comp/

Sistemas cortafuegos:

- *Net Filter / IP Tables*: Sistema cortafuegos para Linux, disponible en http://www.netfilter.org/
- *IP Filter*: Sistema cortafuegos para *BSD y Solaris, disponible en http://coombs.anu.edu.au/ipfilter/

Sitios web sobre seguridad:

- CERT: http://www.cert.org/

- Linux Security: http://www.linuxsecurity.com/
- PacketStorm Security: http://www.packetstormsecurity.org/
- SANS Institute: http://www.sans.org/
- Security Focus: http://www.securityfocus.com/

Avisos de seguridad:

- Boletines de seguridad del SANS Institute: http://www.sans.org/newsletters/
- Listas de Security Focus: http://www.securityfocus.com/archive/
- Listas del CERT: http://www.cert.org/contact_cert/certmaillist.html
- Área de seguridad de la revista LWN (*Linux Weekly News*): Disponible en http://lwn.net/security, tiene una base de datos con avisos de la mayoría de distribuciones de Linux.
- Boletín diario de Hispasec: http://www.hispasec.com/unaaldia/

Debian GNU/Linux

Documentación:

- Área de documentación del servidor web de Debian GNU/Linux: http://www.debian.org/doc/
- *Manual de instalación de debian*: http://www.debian.org/releases/stable/installmanual
- *The Very Verbose Debian 3.0 Installation Walkthrough*, de Clinton De Young: http://osnews.com/story.php?news_id=2016
- *Guía de instalación de debian 3.0*: traducción al castellano del artículo anterior disponible en http://es.tldp.org/Manuales-LuCAS/doc-instalacion-debian-3.0/
- *Manual de seguridad de debian*: http://www.debian.org/doc/manuals/securing-debian-howto/

Software, avisos de seguridad y parches:

- Área de seguridad del proyecto Debian GNU/Linux: http://www.debian.org/security/

Hay que indicar que los almacenes de paquetes de la distribución *estable* y del área de seguridad ya contienen todo el software y los parches que nos puedan interesar.

Foros y listas de correo accesibles desde http://lists.debian.org/:

- `debian-security`: Lista de discusión sobre temas de seguridad en debian
- `debian-security-announce`: Lista donde se publican avisos de seguridad
- `debian-firewall`: Lista de discusión sobre cortafuegos

Red Hat

Documentación:

- Área de documentación del servidor web de RedHat: http://www.redhat.com/docs/
- Documentación técnica sobre seguridad de Red Hat: http://www.redhat.com/solutions/security/techdocs.html
- *Red Hat Linux Security Guide*: http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/

- Libro sobre el `RPM`: http://www.redhat.com/docs/books/max-rpm/

Software, avisos de seguridad y parches:

- Área de alertas de seguridad y corrección de errores de RedHat: Accesible en el URL http://www.redhat.com/apps/support/errata/
- *AutoUpdate*: script de actualización de RedHat similar a `up2date`, http://www.mat.univie.ac.at/~gerald/ftp/autoupdate/
- *apt for rpm based distributions*: Herramienta para convertir un almacén de `rpm` en un almacén de `apt`. En la web hay una página de referencias y otra con un listado de almacenes que contienen muchos enlaces interesantes. Está disponible en http://apt4rpm.sourceforge.net/.
- *APT-RPM Red Hat repository*: almacén con las últimas distribuciones de RedHat accesibles para `apt`, incluyendo la distribución original, las actualizaciones y algunos paquetes extra. En http://apt-rpm.tuxfamily.org/

Foros y listas de correo accesibles desde el URL https://www.redhat.com/mailman/listinfo:

- `redhat-watch-list`: Anuncios de seguridad y de corrección corrección de errores para productos software de RedHat.

Solaris

Documentación:

- Manuales de Solaris 9 (4/03): http://docs.sun.com/db/prod/solaris.9u403
- *Learn SolarisTM 9 Through Step by Step Lessons*: Libro gratuíto sobre Solaris 9, disponible en http://www.teachmesun.com/
- Portal para administradores de Solaris: http://www.sun.com/bigadmin/
- Blueprints de Sun: http://www.sun.com/solutions/blueprints/
- Trucos para solaris: http://www.bolthole.com/solaris/

Software, avisos de seguridad y parches:

- SunSolve: http://sunsolve.sun.com/
- BigAdmin: Desde http://www.sun.com/bigadmin/patches/ tenemos enlaces a listas de correo y parches del sistema.
- Herramientas seguridad: http://www.sun.com/software/security/blueprints/
- Sun Freeware: almacén de software libre pre empaquetado para las distintas versiones de Solaris, http://www.sunfreeware.com/
- CSW - Community SoftWare for Solaris: Otro almacén de software libre empaquetados para instalar en Solaris, http://www.blastwave.org/

Foros y listas de correo:

- `security-alert@sun.com`: Lista de avisos de seguridad de Sun, hay instrucciones para suscribirse en http://sunsolve.sun.com/
- Foros de soporte y alertas de Sun: http://supportforum.sun.com/salerts/

Seguridad física

Cuando hablamos de *seguridad física* nos referimos a todos aquellos mecanismos --generalmente de prevención y detección-- destinados a proteger físicamente cualquier recurso del sistema; estos recursos son desde un simple teclado hasta una cinta de backup con toda la información que hay en el sistema, pasando por la propia CPU de la máquina.

Dependiendo del entorno y los sistemas a proteger esta seguridad será más o menos importante y restrictiva, aunque siempre deberemos tenerla en cuenta.

A continuación mencionaremos algunos de los problemas de seguridad física con los que nos podemos enfrentar y las medidas que podemos tomar para evitarlos o al menos minimizar su impacto.

Protección del hardware

El *hardware* es frecuentemente el elemento más caro de todo sistema informático y por tanto las medidas encaminadas a asegurar su integridad son una parte importante de la seguridad física de cualquier organización.

Problemas a los que nos enfrentamos:

- Acceso físico
- Desastres naturales
- Alteraciones del entorno

Acceso físico

Si alguien que desee atacar un sistema tiene acceso físico al mismo todo el resto de medidas de seguridad implantadas se convierten en inútiles.

De hecho, muchos ataques son entonces triviales, como por ejemplo los de denegación de servicio; si apagamos una máquina que proporciona un servicio es evidente que nadie podrá utilizarlo.

Otros ataques se simplifican enormemente, p. ej. si deseamos obtener datos podemos copiar los ficheros o robar directamente los discos que los contienen.

Incluso dependiendo el grado de vulnerabilidad del sistema es posible tomar el control total del mismo, por ejemplo reiniciándolo con un disco de recuperación que nos permita cambiar las claves de los usuarios.

Este último tipo de ataque es un ejemplo claro de que la seguridad de **todos** los equipos es importante, generalmente si se controla el PC de un usuario autorizado de la red es mucho más sencillo atacar otros equipos de la misma.

Para evitar todo este tipo de problemas deberemos implantar mecanismos de **prevención** (control de acceso a los recursos) y de **detección** (si un mecanismo de prevención falla o no existe debemos al menos detectar los accesos no autorizados cuanto antes).

Para la **prevención** hay soluciones para todos los gustos y de todos los precios:

- analizadores de retina,
- tarjetas inteligentes,
- videocámaras,
- vigilantes jurados,
- ...

En muchos casos es suficiente con controlar el acceso a las salas y cerrar siempre con llave los despachos o salas donde hay equipos informáticos y no tener cableadas las tomas de red que estén accesibles.

Para la **detección** de accesos se emplean medios técnicos, como cámaras de vigilancia de circuito cerrado o alarmas, aunque en muchos entornos es suficiente con qué las personas que utilizan los sistemas se conozcan entre si y sepan quien tiene y no tiene acceso a las distintas salas y equipos, de modo que les resulte sencillo detectar a personas desconocidas o a personas conocidas que se encuentran en sitios no adecuados.

Desastres naturales

Además de los posibles problemas causados por ataques realizados por personas, es importante tener en cuenta que también los *desastres naturales* pueden tener muy graves consecuencias, sobre todo si no los contemplamos en nuestra política de seguridad y su implantación.

Algunos desastres naturales a tener en cuenta:

- Terremotos y vibraciones
- Tormentas eléctricas
- Inundaciones y humedad
- Incendios y humos

Los terremotos son el desastre natural menos probable en la mayoría de organismos ubicados en España, por lo que no se harán grandes inversiones en prevenirlos, aunque hay varias cosas que se pueden hacer sin un desembolso elevado y que son útiles para prevenir problemas causados por pequeñas vibraciones:

- No situar equipos en sitios altos para evitar caídas,
- No colocar elementos móviles sobre los equipos para evitar que caigan sobre ellos,
- Separar los equipos de las ventanas para evitar que caigan por ellas o qué objetos lanzados desde el exterior los dañen,
- Utilizar fijaciones para elementos críticos,
- Colocar los equipos sobre plataformas de goma para que esta absorba las vibraciones,

Otro desastre natural importante son las tormentas con aparato eléctrico, especialmente frecuentes en verano, que generan subidas súbitas de tensión muy superiores a las que pueda generar un problema en la red eléctrica. A parte de la protección mediante el uso de pararrayos, la única solución a este tipo de problemas es desconectar los equipos antes de una tormenta (qué por fortuna suelen ser fácilmente predecibles).

En entornos normales es recomendable que haya un cierto grado de humedad, ya que en si el ambiente es extremadamente seco hay mucha electricidad estática. No obstante, tampoco interesa tener un nivel de humedad demasiadoa elevado, ya que puede producirse condensación en los circuitos integrados que den origen a un cortocircuito. En general no es necesario emplear ningún tipo de aparato para controlar la humedad, pero no está de más disponer de alarmas que nos avisen cuando haya niveles anómalos.

Otro tema distinto son las inundaciones, ya que casi cualquier medio (máquinas, cintas, routers ...) que entre en contacto con el agua queda automáticamente inutilizado, bien por el propio líquido o bien por los cortocircuitos que genera en los sistemas electrónicos. Contra ellas podemos instalar sistemas de detección que apaguen los sistemas si se detecta agua y corten la corriente en cuanto estén apagados. Hay que indicar que los equipos deben estar por encima del sistema de detección de agua, sino cuando se intente parar ya estará mojado.

Por último mencionaremos el fuego y los humos, que en general provendrán del incendio de equipos por sobrecarga eléctrica. Contra ellos emplearemos sistemas de extinción, que aunque pueden dañar los equipos que apaguemos (aunque actualmente son más o menos inocuos), nos evitarán males mayores. Además del fuego,

también el humo es perjudicial para los equipos (incluso el del tabaco), al ser un abrasivo que ataca a todos los componentes, por lo que es recomendable mantenerlo lo más alejado posible de los equipos.

Alteraciones del entorno

En nuestro entorno de trabajo hay factores que pueden sufrir variaciones que afecten a nuestros sistemas que tendremos que conocer e intentar controlar.

Deberemos contemplar problemas que pueden afectar el régimen de funcionamiento habitual de las máquinas como la alimentación eléctrica, el ruido eléctrico producido por los equipos o los cambios bruscos de temperatura.

Electricidad

Quizás los problemas derivados del entorno de trabajo más frecuentes son los relacionados con el sistema eléctrico que alimenta nuestros equipos; cortocircuitos, picos de tensión, cortes de flujo ...

Para corregir los problemas con las subidas de tensión podremos instalar tomas de tierra o filtros reguladores de tensión.

Para los cortes podemos emplear *Sistemas de Alimentación Ininterrumpida* (SAI), que además de proteger ante cortes mantienen el flujo de corriente constante, evitando las subidas y bajadas de tensión. Estos equipos disponen de baterias que permiten mantener varios minutos los aparatos conectados a ellos, permitiendo que los sistemas se apaguen de forma ordenada (generalmente disponen de algún mecanísmo para comunicarse con los servidores y avisarlos de que ha caido la línea o de que se ha restaurado después de una caida).

Por último indicar que además de los problemas del sistema eléctrico también debemos preocuparnos de la corriente estática, que puede dañar los equipos. Para evitar problemas se pueden emplear esprais antiestáticos o ionizadores y tener cuidado de no tocar componentes metálicos, evitar que el ambiente esté excesivamente seco, etc.

Ruido eléctrico

El ruido eléctrico suele ser generado por motores o por maquinaria pesada, pero también puede serlo por otros ordenadores o por multitud de aparatos, y se transmite a través del espacio o de líneas eléctricas cercanas a nuestra instalación.

Para prevenir los problemas que puede causar el ruido eléctrico lo más barato es intentar no situar el *hardware* cerca de los elementos que pueden causar el ruido. En caso de que fuese necesario hacerlo siempre podemos instalar filtos o apantallar las cajas de los equipos.

Temperaturas extremas

No hace falta ser un genio para comprender que las temperaturas extremas, ya sea un calor excesivo o un frio intenso, perjudican gravemente a todos los equipos. En general es recomendable que los equipos operen entre 10 y 32 grados Celsius. Para controlar la temperatura emplearemos aparatos de aire acondicionado.

Protección de los datos

Además proteger el *hardware* nuestra política de seguridad debe incluir medidas de protección de los datos, ya que en realidad la mayoría de ataques tienen como objetivo la obtención de información, no la destrucción del

medio físico que la contiene.

En los puntos siguientes mencionaremos los problemas de seguridad que afectan a la transmisión y almacenamiento de datos, proponiendo medidas para reducir el riesgo.

Eavesdropping

La *interceptación* o *eavesdropping*, también conocida por ”passive wiretapping” es un proceso mediante el cual un agente capta información que va dirigida a él; esta captación puede realizarse por muchísimos medios: *sniffing* en redes ethernet o inalámbricas (un dispositivo se pone en modo promiscuo y analiza todo el tráfico que pasa por la red), capturando radiaciones electromagnéticas (muy caro, pero permite detectar teclas pulsadas, contenidos de pantallas, ...), etc.

El problema de este tipo de ataque es que en principio es completamente pasivo y en general difícil de detectar mientras se produce, de forma que un atacante puede capturar información privilegiada y claves que puede emplear para atacar de modo *activo*.

Para evitar que funcionen los *sniffer* existen diversas soluciones, aunque al final la única realmente útil es cifrar toda la información que viaja por la red (sea a través de cables o por el aire). En principio para conseguir esto se deberían emplear versiones seguras de los protocolos de uso común, siempre y cuando queramos proteger la información. Hoy en día casi todos los protocolos basados en TCP permiten usar una versión cifrada mendiante el uso del TLS.

Copias de seguridad

Es evidente que es necesario establecer una política adecuada de copias de seguridad en cualquier organización; al igual que sucede con el resto de equipos y sistemas, los medios donde residen estas copias tendrán que estar protegidos físicamente; de hecho quizás deberíamos de emplear medidas más fuertes, ya que en realidad es fácil que en una sola cinta haya copias de la información contenida en varios servidores.

Lo primero que debemos pensar es dónde se almacenan los dispositivos donde se realizan las copias. Un error muy habitual es almacenarlos en lugares muy cercanos a la sala de operaciones, cuando no en la misma sala; esto, que en principio puede parecer correcto (y cómodo si necesitamos restaurar unos archivos) puede convertirse en un problema serio si se produce cualquier tipo de desastre (como p. ej. un incendio). Hay que pensar que en general el *hardware* se puede volver a comprar, pero una pérdida de información puede ser ireemplazable.

Así pues, lo más recomendable es guardar las copias en una zona alejada de la sala de operaciones; lo que se suele recomendar es disponer de varios niveles de copia, una que se almacena en una caja de seguridad en un lugar alejado y que se renueva con una periodicidad alta y otras de uso frecuente que se almacenan en lugares más próximos (aunque a poder ser lejos de la sala donde se encuentran los equipos copiados).

Para proteger más aun la información copiada se pueden emplear mecanísmos de cifrado, de modo que la copia que guardamos no sirva de nada si no disponemos de la clave para recuperar los datos almacenados.

Soportes no electrónicos

Otro elemento importante en la protección de la información son los elementos no electrónicos que se emplean para transmitirla, fundamentalmente el papel. Es importante que en las organizaciones que se maneje información confidencial se controlen los sistemas que permiten exportarla tanto en formato electrónico como en no electrónico (impresoras, plotters, faxes, teletipos, ...).

Cualquier dispositivo por el que pueda salir información de nuestro sistema ha de estar situado en un lugar de acceso restringido; también es conveniente que sea de acceso restringido el lugar donde los usuarios recogen los documentos que lanzan a estos dispositivos.

Además de esto es recomendable disponer de trituradoras de papel para destruir todos los papeles o documentos que se quieran destruir, ya que evitaremos que un posible atacante pueda obtener información rebuscando en nuestra basura.

El sistema de arranque

Todas las estaciones de trabajo actuales disponen de algún sitema básico de arranque grabado en una memoria de sólo lectura (generalmente una EPROM, que permite actualizarla) denominado *firmware*.

En general este sistema se encarga de inicializar el *hardware*, cargar el operativo o un sistema de arranque más complejo y ejecutarlo.

Dado que este curso se centra principalmente en Solaris y Linux, y que las principales plataformas para ambos sistemas son SPARC e Intel respectivamente, sólo discutiremos los sistemas de arranque de estas dos arquitecturas, aunque probablemente lo dicho aquí será aplicable a otros sistemas.

En los PCs basados en tecnología Intel el *firmware* se suele denominar BIOS (*Basic Input Output System*) y suele tener una funcionalidad bastante limitada.

En las estaciones SPARC actuales el *firmware* se denomina **OpenBoot**. Se trata de una implementación del *IEEE Standard 1275-1994*, que define un estándar para los *firmware de arranque*. En la actualidad, además de los equipos SPARC de Sun, tambien los PowerPC de Apple emplean un *firmware* basado en este estándar.

En los apartados siguientes explicaremos brevemente como funcionan ambos tipos de sistemas de arranque y las implicaciones que tienen para la seguridad. Hay que indicar que en general las modificaciones que se hagan en las configuraciones de estos sistemas pueden ser evitadas desmontando los equipos, pero para un entorno de trabajo normal pueden ser suficientes (no es normal que alguien no autorizado que acceda a un equipo se ponga a desmontarlo sin que nadie le diga nada).

La BIOS del PC (Intel)

En general las BIOS de los equipos basados en Intel son muy similares, aunque exiten varios fabricantes que se reparten el mercado, los más famosos son Phoenix (http://www.phoenix.com/) y AMI (http://www.ami.com/).

Generalmente el acceso a la BIOS se hace pulsando una o varias teclas cuando arranca el equipo; una vez pulsado se entra en un entorno de configuración sencillo, habitualmente en modo texto, que nos permite definir parámetros del sistema como la fecha y la hora, la geometría y tamaño de los discos, si debemos habilitar o no determinadas controladoras, etc.

Uno de los parámetros que más nos interesa desde el punto de vista de la seguridad es la elección de los dispositivos de arranque, ya que generalmente es la BIOS la que decide el orden en el que se intenta arrancar empleando distintos soportes (disco duro, CD-ROM, floppy, ...). Si permitimos el arranque de disquete o CD-ROM es extremadamente fácil para cualquiera que se siente ante el equipo el arrancarlo con cualquier cosa y modificar nuestros sistemas de ficheros.

Aunque configuremos correctamente los dispositivos de arranque es necesario limitar el acceso a la BIOS para que nadie pueda entrar en ella de nuevo y cambiar los ajustes. Para ello la mayor parte de las BIOS ofrecen la posibilidad de establecer dos contraseñas independientes; una para impedir el acceso a la configuración de la BIOS (hay que introducirla despues de pulsar la secuencia de teclas que carga el programa de configuración de la BIOS) y otra de arranque (si no se introduce la clave no se inicia el procedimiento de arranque y por tanto no se puede ejecutar nada). El problema de esta última contraseña es que es necesario estar delante del PC para arrancar un equipo, algo poco práctico si reiniciamos remotamente el ordenador o simplemente si se va la luz ...

Por último mencionar que la BIOS tiene un sistema de carga de S. O. extremadamente simple: en el caso de los discos duros se carga el programa que hay en unos sectores concretos del disco (el MBR o **Master Boot Record**) o, en caso de no haber nada en ese sector, lo que hay en el sector de arranque de la partición marcada como **activa**.

Por todo esto, es imposible usar la BIOS para pasarle parámetros a un sistema operativo o simplemente decidir cual vamos a arrancar si tenemos más de uno. Si instalamos linux en el equipo es habitual instalar en el disco algún **bootloader** que nos permita arrancar varios sistemas operativos distintos (o incluso el mismo con parámetros diferentes) como LILO o GRUB. Hablaremos de estos programas en el apartado que trata la instalación de Linux.

La OpenBootProm de Sun (SPARC)

El *firmware* de las estaciones SPARC de SUN es una memoria NVRAM denominada `OpenBoot PROM` o simplemente `OBP`.

El acceso a la misma se consigue pulsando la combinación de teclas `Stop-A` en teclados Sun o `Ctrl-Break` en terminales serie. Sin importar el estado en que se encuentre el sistema, cuando se pulse esta combinación se se detendrán automáticamente todos los procesos en ejecución y se mostrará en consola el prompt `ok`, que indica que podemos comenzar a teclear órdenes de la OBP.

La máquina no pierde en ningún momento su estado a no ser que explícitamente la detengamos: al salir de la OBP podemos continuar la ejecución de todos los procesos que teníamos al entrar, desde el mismo punto en que los detuvimos y con el mismo entorno que poseían, pero mientras estemos interactuando con la `EEPROM` ningún proceso avanzará en su ejecución.

Un problema con este modo de funcionamiento es que al interactuar con la EEPROM, cualquier persona puede interrumpir al operativo y rearrancarlo desde un disco, un CD-ROM, o un sistema remoto, lo que evidentemente le proporciona un control total sobre el sistema.

Para evitarlo podemos deshabilitar la función de las teclas `Stop-A` mediante la directiva del kernel `abort_enable` en el fichero `/etc/system`, o proteger mediante contraseña el reinicio de una máquina desde su memoria NVRAM.

La **OBP** ofrece tres niveles de seguridad: `none-secure`, `command-secure`, y `full-secure`.

El primero de ellos, `none-secure` es el que está habilitado por defecto y no ofrece ningún tipo de seguridad.

Si activamos `command-secure` será necesaria una clave para reiniciar el sistema de cualquier dispositivo que no sea el utilizado por defecto (que generalmente será el disco, disk), y si elegimos `full-secure` la contraseña es obligatoria independientemente del dispositivo elegido para arrancar (con lo que no deberemos usar este modo si queremos permitir reinicios automáticos).

Hay que indicar que esta contraseña no tiene nada que ver con la del usuario `root` del sistema, aunque este usario puede usar desde línea de órdenes el comando `eeprom` para modificar (o consultar) cualquier parámetro de la NVRAM, passwords incluidos.

Si perdemos la contraseña de la EEPROM y no podemos arrancar la máquina será necesario sustituir nuestra memoria NVRAM por una nueva, por lo que hemos de tener cuidado con las claves que utilicemos para proteger la OBP.

Como hemos adelantado, para consultar o modificar el modo en el que se encuentra nuestra memoria NVRAM podemos ejecutar la orden `eeprom`.

Para conocer el estado de una variable (`security-mode`) haríamos lo siguiente:

```
lem:/# eeprom security-mode
security-mode=none
```

en este caso nuestra máquina no tiene habilitado ningún tipo de seguridad; si quisiéramos habilitar el modo `command-secure`, ejecutaríamos:

```
lem:/# eeprom security-mode=command
Changing PROM password:
New password:
Retype new password:
lem:/# eeprom security-mode
security-mode=command
```

También es posible realizar estos cambios desde el propio prompt de la memoria NVRAM, mediante la orden `setenv`:

```
ok setenv security-mode command
security-mode =       command
```

A partir de este momento, cuando el sistema inicie desde un dispositivo que no sea el utilizado por defecto, se solicitará la clave que acabamos de teclear; de forma similar podríamos habilitar el modo `full-secure`.

Para eliminar cualquier clave de nuestra memoria no tenemos más que restaurar el modo `none-secure`, de la forma habitual:

```
lem:/# eeprom security-mode=none
lem:/# eeprom security-mode
security-mode=none
```

Si en los modos `command-secure` o `full-secure` queremos cambiar la contraseña de la NVRAM podemos utilizar de nuevo la orden `eeprom` y el parámetro `security-password`:

```
lem:/# eeprom security-password=
Changing PROM password:
New password:
Retype new password:
lem:/# eeprom security-password
security-password= data not available.
```

Como podemos ver, al consultar el valor de la variable, este nunca se muestra en pantalla.

El tercer y último parámetro relacionado con la seguridad de la memoria EEPROM es `security-#badlogins`, que no es más que un contador que indica el número de contraseñas incorrectas que el sistema ha recibido; podemos resetear su valor sencillamente asignándole `0`:

```
lem:/# eeprom security-#badlogins
security-#badlogins=4
lem:/# eeprom security-#badlogins=0
lem:/# eeprom security-#badlogins
security-#badlogins=0
```

Por último indicar que los parámetros de seguridad de la memoria `EEPROM` que acabamos de ver sólo existen en máquinas `SPARC`; aunque en la versión de Solaris para arquitecturas Intel también existe una orden denominada ‘eeprom’ que nos mostrará los valores de ciertos parámetros si la ejecutamos, únicamente se trata de una simulación llevada a cabo en un fichero de texto denominado ‘bootenv.rc’. Es posible dar valor a las variables que hemos visto, pero no tienen ningún efecto en máquinas Intel ya que estas suelen proteger el arranque mediante contraseñas en la BIOS, como hemos visto.

Elección de la distribución

En el caso de Solaris es evidente que instalaremos la distribución del sistema proporcionada por Sun, ya que es la única existente, pero en el caso de Linux tenemos una gran cantidad de distribuciones entre las que elegir.

Las distribuciones de Linux tienen muchas características comunes, pero tambien existen diferencias en aspectos importantes:

- Formatos de paquete (`.deb` para Debian, `.rpm` para RedHat, SuSE o Mandrake, `.tgz` para Slackware, etc.),
- Herramientas de actualización (`apt` para Debian, `up2date` para RedHat, etc.),
- Disponibilidad de parches y actualizaciones de seguridad para distintas versiones de la distribución,
- Soporte comercial o a través de foros de usuarios,
- Número y calidad de los paquetes disponibles,
- Certificaciones de fabricantes de *hardware* y *software*,
- Costes de los distintos servicios (p. ej. el acceso a los almacenes de Debian es gratuito, mientras que para acceder a la RHN hay que pagar)

En general la distribución a instalar vendrá determinada por los requisitos del software servidor que queramos instalar; en este curso recomendaremos el uso de **Debian GNU/Linux** siempre que sea posible, ya que consideramos que actualmente es la mejor distribución de Linux desde un punto de vista técnico, aunque estudiaremos tambien la instalación de **RedHat** por ser una de las distribuciones más extendidas.

Planificación de la instalación

Antes de comenzar una instalación debemos plantearnos una serie de preguntas que nos permitan realizarla de modo efectivo:

- ¿Qué hardware vamos a emplear (arquitectura, tarjetas de red, etc.)?,
- ¿Qué sistema operativo y versión?,
- ¿Como vamos a instalar? ¿desde disco flexible, CD-ROM, vía red, ...?,
- ¿Como vamos a configurar la BIOS?,
- ¿Cómo vamos a particionar los discos?,
- ¿Qué sistemas de ficheros vamos a emplear?
- ¿Cual es la configuración de la red? (en principio no deberíamos emplear DHCP para un servidor):
  - Dirección IP
  - Máscara de red
  - Dirección del *gateway*
  - Nombre y dominio del equipo
  - Direcciones de los servidores de nombres
- ¿Qué sistemas de autentificación vamos a emplear?
- ¿Contraseña del administrador?
- ¿Qué *bootloader* vamos a emplear?

Una vez tengamos las respuestas a estas preguntas podremos instalar un sistema básico, configurar los servicios mínimos (incluyendo un cortafuegos si es necesario), actualizar los componentes a las últimas versiones sin vulnerabilidades conocidas y por último instalar y configurar los paquetes adicionales que necesitemos.

Particionado del disco y elección de tipo de sistema de archivos

Independientemente del sistema Unix a instalar tendremos que decidir como particionamos el disco y que tipo de sistema de archivos vamos a emplear.

Al tratarse de un curso orientado a la instalación de servidores supondremos que todo el disco está disponible para nuestro sistema operativo, de modo que no nos tendremos que preocupar de mantener particiones para otros S.O.

En el caso más simple lo normal es definir 1 partición para la memoria de intercambio o *swap* y otra para el sistema de archivos raiz.

Además de estas dos particiones es habitual incluir otra partición dedicada exclusivamente a la información de los usuarios (`/home` en **Linux**, `/export/home` en **Solaris**), de manera que sea lo más independiente posible de la versión del sistema (de hecho podremos usarla desde varias máquinas o mantener la misma partición aunque actualicemos la versión del sistema operativo).

En sistemas **Linux** para **Intel** tambien es común definir una pequeña partición donde se instala el núcleo del sistema operativo y todo lo que necesita para arrancar (partición `/boot`). En origen se empezó a usar esta partición por las limitaciones de la **BIOS** del PC y el programa `LILO`, que hacían necesario que la imágen del nucleo que queremos arrancar esté accesible empleando el sistema de direccionamiento de la *BIOS* (nunca más allá del cilindro 1024 del disco).

Visto lo anterior, ¿qué sentido tiene emplear más particiones en nuestros discos? Dependiendo del uso que le vayamos a dar al sistema y las aplicaciones que empleemos usar multiples particiones nos puede resultar útil, ya que nos permite emplear sistemas de archivos diferentes (p. ej. con o sin *journaling*, con soporte de listas de control de acceso o ACLs, más optimizados para trabajar con tipos concretos de archivos, etc.) y montarlas con opciones distintas (partición montada en modo de sólo lectura, sin permiso de ejecución de programas, etc.).

En cuanto a la elección del sistema de archivos, dependerá del uso que le vayamos a dar al sistema, pero en principio en **Solaris** usaremos el sistema de archivos por defecto `UFS` (*Unix File System*) y en **Linux** el `ext3` (versión con *journaling* del `ext2`).

El cargador de arranque

Como ya hemos comentado en puntos anteriores, para arrancar un sistema Solaris o Linux se pueden emplear diversos medios: instalar un cargador de arranque en el disco duro o en un disco flexible, arrancar desde otro sistema operativo, etc.

En el caso que nos ocupa hablaremos sólo de los sistemas de arranque que se instalan en disco duro, comentando brevemente sus diferencias y sus características de seguridad.

Sólo hablaremos de los sistemas de arranque empleados en los PC y de estos sólo entraremos en detalles en los que se emplean con sistemas Linux, el que esté interesado en conocer como funciona el sistema de arranque incluido en la versión de Solaris para Intel tiene un buen documento de referencia en: http://developers.sun.com/solaris/developer/support/driver/wps/realmode_env/

LILO

LILO (*LInux LOader*) es el sistema de arranque más extendido en el mundo Linux, aunque es un poco arcáico. Se instala en un sector de arranque - de una partición o de un disco flexible - o en el *Master Boot Record* (*MBR*) del disco duro y permite arrancar Linux y otros sistemas operativos instalados en el PC.

La configuración se encuentra generalmente en el archivo `/etc/lilo.conf`, aunque en realidad se almacena en el sector de arranque o en el MBR, por lo que cada vez que queramos aplicar los cambios hechos en el fichero será necesario ejecutar la orden `/sbin/lilo` para que reinstale el programa.

El formato del fichero de configuración es sencillo, tiene una parte global y secciones para cada sistema operativo que queramos arrancar:

```
# /etc/lilo.conf
# Global options:
boot=/dev/hda
map=/boot/map
lba32
compact
vga=normal
read-only
delay=20
# bootable kernel images:
image=/boot/vmlinuz-2.4.21-2-686
  label=linux
  root=/dev/hda2
  initrd=/boot/initrd-2.4.21-2.img
image=/boot/vmlinuz-2.4.20-3-686
  label=linux.old
  root=/dev/hda2
  initrd=/boot/initrd-2.4.20-3.img
# other operating systems:
other=/dev/hda1
  label=windows
  table=/dev/hda
```

Al arrancar el PC, LILO permite elegir la imagen que queremos arrancar y pasar parámetros al núcleo; aunque esto sea necesario para inicializar el sistema en ciertas ocasiones - principalmente cuando hay errores graves en un arranque normal - el hecho es que los parámetros pasados a un kernel antes de ser arrancado pueden facilitar a un atacante un control total sobre la máquina, ya que algunos de ellos llegan incluso a ejecutar un shell con privilegios de root sin necesidad de ninguna contraseña.

Para proteger el arranque podemos habilitar el uso de contraseñas en LILO, de modo que se solicite antes de que se cargue cualquier sistema operativo instalado en el ordenador o cuando se intenten pasar parámetros a una imágen.

Para poner la contraseña usaremos la palabra reservada `password`; si la ponemos en la sección general se aplicará a todas las imágenes definidas y si queremos que se aplique a imágenes concretas la la pondremos dentro de la sección específica.

Una vez tenemos definida una contraseña, el comportamiento del LILO vendrá determinado por la aparición de las palabras `mandatory`, `restricted` y `bypass`:

1. La primera implica que para arrancar la imagen o imágenes seleccionadadas debemos introducir la clave, esta es la opción que se aplica por defecto.
2. La segunda implica que la contraseña sólo se solicitará si se desean pasar parámetros adicionales al núcleo.

3. La tercera deshabilita el uso de la contraseña (se puede usar para quitar el uso de una contraseña global al arrancar sistemas que no son linux).

GRUB

GRUB (GRand Unified Bootloader) es un sistema de arranque más potente que el anterior. Una vez instalado en un sector de arranque (de una partición o un disco flexible) o en el Master Boot Record (MBR) del disco duro, ejecuta un *interprete de comandos* cada vez que iniciamos el sistema que nos permite arrancar prácticamente cualquier sistema operativo actual.

Este *interprete* se puede usar de modo interactivo o puede leer un fichero de configuración almacenado en el disco (que por defecto estará en `/boot/grub/menu.lst`). Una característica importante de GRUB es que es capaz de reconocer gran cantidad de sistemas de ficheros, de modo que no es necesario reinstalarlo cuando cambiamos ese fichero de configuración, que es simplemente un fichero de texto.

Un ejemplo de `/boot/grub/menu.lst` similar al ejemplo anterior del `LILO` sería el siguiente:

```
# Sample boot menu configuration file
# ----------------------------------
# Boot automatically after 10 secs.
timeout 10
# By default, boot the first entry.
default 0
# Fallback to the second entry.
fallback 1
# Kernel 2.4.21-2-686
title  Linux 2.4.21-2-686 (hda2)
kernel (hd0,1)/boot/vmlinuz-2.4.21-2-686 root=/dev/hda2 hdb=ide-scsi hdc=ide-scsi
initrd (hd0,1)/boot/initrd.img-2.4.21-2-686
# Kernel 2.4.20-3-686
title  Linux 2.4.20-3-686 (hda2)
kernel (hd0,1)/boot/vmlinuz-2.4.20-3-686 root=/dev/hda2 hdb=ide-scsi hdc=ide-scsi
initrd (hd0,1)/boot/initrd.img-2.4.20-3-686
# For booting Windows
title Windows
rootnoverify (hd0,0)
makeactive
chainloader  +1
```

El problema con el interprete de comandos es que nos da control total sobre el arranque; por suerte `GRUB` nos permite proteger el acceso al mismo empleando una contraseña en el fichero de configuración, de modo que sea necesario introducirlo para acceder al menú o cuando queramos modificar parámetros de una sección.

Esta contraseña se pone en el fichero empleando la instrucción `password`:

```
password ['--md5'] clave [nuevo-fichero-de-config]
```

Donde clave es la contraseña en claro o cifrada con `md5` si ponemos `--md5` delante. El nombre de fichero que aparece detrás de la clave es un fichero de configuración con el mismo formato que el `menu.lst` que se cargará si se introduce la clave adecuada.

Si la clave aparece en la sección global del fichero de configuración sólo se empleara si el usario quiere editar el menu de arranque, si además queremos *bloquear* el arranque de alguna sección podemos emplear el comando `lock`, que detiene la ejecución del `GRUB` hasta que se introduce una clave válida. Para que sea efectivo lo tendremos que poner detrás del título de la sección que queramos proteger.

Instalación de Debian GNU/Linux

Actualmente existen 4 versiones activas de la distribución de **Debian**:

`estable`

: Es la distribución recomendada para sistemas en producción, ya que sigue un proceso riguroso de control de calidad y se generan versiones actualizadas de los paquetes cuando aparecen vulnerabilidades.

`inestable`

: Es la distribución empleada por los desarrolladores para ir añadiendo y modificando paquetes, en general tiene las últimas versiones *estables* de los programas (*inestable* se refiere al empaquetado, no a los programas) y es muy usable. Es adecuada para sistemas de usuario que no precisen la máxima estabilidad y quieran *probar* todo lo que va saliendo.

`testing`

: Pretende ser una versión *consistente* de la distribución, los paquetes que contiene se pasan desde la distribución *inestable* si trás unas semanas de prueba no han aparecido *bugs* de importancia y todos los programas y librerías de los que depende el paquete ya han sido pasados a *testing*. De este modo, cuando se decide que hay material suficiente para liberar una nueva versión de la distribución se *congela* la versión de pruebas (es decir, no se añaden más paquetes y sólo se actualizan los que tienen algún tipo de problema) hasta que se eliminan los errores más graves, se valida el proceso de actualización desde la antigua *estable* y se publica.

`experimental`

: Es una distribución pensada para probar versiones inestables de programas o paquetes con posibles problemas. En realidad no se utiliza como una distribución, sino más bien como un almacén de conveniencia del que cojer paquetes que realmente nos interesa probar.

En principio el sistema de instalación sólo está disponible y probado para la distribución estable, aunque periódicamente se generan CD's de instalación para la distribución de prueba.

Un buen esquema de los pasos que hay que seguir en el proceso de instalación de **Debian** es el siguiente:

1. Crear espacio particionable para Debian en el disco duro
2. Localizar y/o descargar los ficheros del núcleo y los controladores (salvo los usuarios de un CD Debian)
3. Crear los discos de arranque o usar un CD de Debian
4. Arrancar el sistema de instalación
5. Elegir el idioma
6. Configurar el teclado
7. Crear y montar particiones Debian
8. Señalar al instalador la localización del núcleo y los controladores
9. Seleccionar qué controladores de periféricos cargar
10. Configurar la interfaz de red
11. Iniciar la descarga/instalación/configuración automática del sistema base
12. Configurar la carga del arranque de Linux o arranque múltiple
13. Arrancar el sistema recién instalado y hacer algunas configuraciones finales
14. Instalar tareas y paquetes adicionales

En nuestro caso todo el disco será para la instalación, así que no tendremos que hacer nada antes de arrancar. La instalación la haremos desde CD-ROM, por lo que ya tendremos los ficheros del núcleo y los controladores y no nos hará falta crear discos adicionales.

La Debian 3.0 puede instalarse empleando núcleos de Linux de la serie 2.2 o de la 2.4, en principio nosotros usaremos los de esta última serie. Para ello cuando nos aparezca la línea `boot:` pondremos:

```
boot: bf24
```

Esto nos iniciará el sistema de instalación usando el kernel 2.4.

A partir de aquí seguiremos los pasos descritos en el documento Un paseo detallado por la instalación de Debian 3.0 (http://es.tldp.org/Manuales-LuCAS/doc-instalacion-debian-3.0/doc-instalacion-debian-3.0-html/index.html), terminando antes de la explicación de como instalar paquetes adicionales y configurar el sistema gráfico *XFree86*.

En el punto siguiente veremos como instalar programas adicionales.

Herramientas administrativas y de actualización de Debian GNU/Linux

Debian emplea el formato de paquete deniominado `.deb`, y para gestionar su sistema de paquetes dispone de varias herramientas:

- El programa de más bajo nivel se denomina `dpkg-deb` y se emplea para manipular directamente los paquetes en formato `.deb`. Este programa no suele ser utilizado por los usuarios directamente.
- Un peldaño por encima tenemos el programa `dpkg`, que es una herramienta de nivel intermedio que permite instalar, compilar, eliminar o gestionar los paquetes de Debian, manteniendo una base de datos de paquetes disponibles y el estado de los mismos.
- `APT` (Advanced Package Tool), es un sistema de gestión de paquetes de software; se desarrolló en el proyecto Debian empleando paquetes en formato `.deb` pero es bastante independiente del formato de archivos, lo que ha permitido que existan versiones del sistema para paquetes en formato `.rpm`. El sistema incluye herramientas para gestionar una base de datos de paquetes que nos permite obtenerlos e instalarlos, detectando y solucionando problemas de dependencias (seleccionando paquetes que nos hagan falta) y conflictos entre paquetes (eliminando los problemáticos).
- El *front-end* tradicional para la gestión de paquetes en Debian es el programa `dselect`, que es un interfaz de usuario que permite actualizar las listas de paquetes, ver el estado de los paquetes disponibles e instalados, alterar las selecciones de paquetes gestionando las dependencias e instalar, actualilzar o eliminar los paquetes. Antiguamente se empleaban varios métodos distintos para obtener las listas de paquetes disponibles y descargarlos, pero últimamente sólo se emplea el proporcionado por `apt`.

dpkg

A continuación damos una lista de ejemplos de uso, para más información se recomienda leer la página de manual `dpkg(8)`:

`dpkg --help`

Muestra las opciones del programa.

`dpkg --contents nompaq_VVV-RRR.deb`

Muestra los ficheros que contiene el paquete.

`dpkg --info nompaq_VVV-RRR.deb`

Imprime el fichero de control y otros datos del paquete.

`dpkg --install nompaq_VVV-RRR.deb`

Instala el paquete especificado, incluyendo el desempaquetado y la configuración, en el disco duro.

`dpkg --unpack nompaq_VVV-RRR.deb`

Desempaqueta un archivo `.deb` en el disco y lo deja sin configurar.

`dpkg --configure nompaq`

Configura un paquete previamente desempaquetado.

`dpkg --configure --pending`

Configura todos los paquetes previamente desempaquetados. Suele ser útil cuando herramientas de más alto nivel fallan y no terminan de configurar paquetes por errores no relacionados entre sí.

`dpkg --listfiles nompaq`

Lista los ficheros que contiene un paquete ya instalado.

`dpkg --remove nompaq`

Borra un paquete, pero deja los ficheros de configuración que tuviera.

`dpkg --purge nompaq`

Borra un paquete eliminando todos sus ficheros de configuración.

`dpkg --status nompaq`

Muestra el estado de un paquete.

`dpkg --search expreg`

Muestra los paquetes que contienen ficheros que coinciden con la expresión regular.

Hay que indicar que para emplear los paquetes con `dpkg` es preciso tenerlos en el disco, ya que el programa trabaja con el fichero que contiene el paquete o con paquetes ya instalados.

apt

Para utilizar `APT` es necesario indicarle al sistema donde tiene que buscar los recursos; esto se hace en el fichero `/etc/apt/sources.list`.

Cada línea índica de donde obtener la lista de paquetes binarios o ficheros fuente disponibles. El formato de las entradas es el siguiente:

```
deb uri distribución [componente1] [componente2] [...]
```

El el primer elemento de la línea puede ser `deb` (para paquetes binarios) o `deb-src` (si nos referimos a ficheros fuente). El segundo nos indica la dirección del directorio base de una distribución tipo Debian y acepta cuatro tipos de `URI`: `file`, `cdrom`, `http` y `ftp`. El tercer parámetro nos indica la distribución que vamos a emplear (en

nuestro caso emplearemos `woody` o `stable`) y los siguientes parámetros indícan secciones de la distribución (generalmente `main`, `contrib` y `non-free`, aunque puede haber otras).

Para interactuar con *APT* se suelen emplear dos programas, `apt-get` y `apt-cache`. El primero es el cliente de línea de comandos del sistema y el segundo es el que se emplea para manipular la *cache* de paquetes.

A continuación presentamos ejemplos de uso de ambos programas, explicando lo qué hacen, como siempre recomendamos la lectura de las páginas de manual de `apt-get(8)` y `apt-cache(8)` para el que desee más información.

Uso de `apt-get`:

`apt-get --help`

Muestra un resumen del uso del programa.

`apt-get install <paquete>`

Descarga el paquete <paquete> y todas sus dependencias, instalando o actualizando los paquetes descargados.

`apt-get remove [--purge] <paquete>`

Elimina el paquete <paquete> y los paquetes que dependan de él. Si se emplea la opción `--purge` se eliminan además los ficheros de configuración del paquete.

`apt-get update`

Actualiza la lista de paquetes desde los origenes definidos en el fichero `/etc/apt/sources.list`. Se debe ejecutar siempre que queramos instalar o actualizar algo o cuando cambiemos la configuración.

`apt-get upgrade [-u]`

Actualiza todos los paquetes instalados a las versiones más nuevas disponibles, sin instalar paquetes nuevos ni eliminar los antiguos. Si la actualización de un paquete requiere instalar paquetes nuevos dejamos ese paquete no se actualiza. La opción `-u` nos mostrará los paquetes que se van a actualizar.

`apt-get dist-upgrade [-u]`

Similar a lo anterior, excepto que instala o elimina paquetes para satisfacer las dependencias en lugar de no actualizar.

`apt-get [install|upgrade|dist-upgrade] -f`

Se usa para corregir problemas de dependencias cuando una operación como `install` falla. Es interesante indicar que se invoca sin referirnos a ningún paquete en concreto.

Uso de `apt-cache`:

`apt-cache --help`

Muestra un resumen del uso del programa.

`apt-cache search <patrón>`

Busca en los paquetes y el las descripciones el patrón <patrón>.

`apt-cache show <paquete>`

Muestra la descripción completa del paquete <paquete>

`apt-cache showpkg <paquete>`

Muestra detalles sobre el paquete y su relación con otros paquetes.

dselect

Herramienta de más alto nivel que nos permite seleccionar, instalar, borrar o dejar paquetes en espera (no actualizarlos, p. ej. por qué la nueva versión genera conflictos con otras versiones u otros programas, etc.).

Utiliza un interfaz en modo texto que nos permite ver los paquetes agrupados por categorías e importancias, acceder a descripción de los paquetes y cuando seleccionamos un paquete nos selecciona sus dependencias, nos recomienda instalar más paquetes si el que estamos instalando o sus dependencias tienen recomendaciones y nos ayuda a resolver conflictos (p. ej. nos dice que tenemos que eliminar versiones incompatibles de programas o bibliotecas).

El menú de `dselect` es bastante explicativo:

```
0. [M]étodo    Escoger el método de acceso que se usará.
1. [A]ctualiza Actualizar la lista de paquetes disponibles, si se puede.
2. [S]eleccion Solicitar qué paquetes desea en el sistema.
3. [I]nstalar  Instalar y actualizar los paquetes deseados.
4. [C]onfigura Configurar los paquetes que no estén configurados.
5. [D]esinstal Desinstalar los paquetes no deseados.
6. sa[L]ir     Salir de dselect.
```

En la actualidad siempre se suele emplear como *método* de acceso a los paquetes el `apt`, de modo que esa opción no se suele emplear.

La siguiente opción se debe emplear siempre que queramos instalar o actualizar el sistema, ya que se encarga de obtener las últimas listas de paquetes disponibles.

El siguiente punto, *selección*, nos muestra la lista de paquetes diciéndonos cuales son nuevos, cuales tienen actualización disponible, cuales han desaparecido de la lista de paquetes, etc. Además, nos indica el estado de cada uno de ellos y nos permite ordenar la visualización de varias maneras. Usando esta vista podemos marcar los paquetes para que sean instalados, borrados, purgados o que se mantengan sin actualizar.

El punto 3, *instalar*, es el último realmente útil en la actualidad, ya que al usar `apt`, cuando le damos a esta opción se descargan, instalan, configuran y eliminan los paquetes de una vez. Empleando otros métodos de acceso si que era necesario pasar por el cuarto y quito paso.

Podemos decir que `dselect` es una buena herramienta para gestionar la instalación y actualización de paquetes de forma global, ya que nos permite acceder comodamente a la descripción de los mismos, conocer rápidamente su estado y es muy útil a la hora de resolver conflictos. Por todo ello recomendamos a todo aquel que quiera trabajar seriamente con debian que aprenda a utilizar este programa o alguno equivalente (como por ejemplo `synaptic`, que es similar aunque necesita X).

Actualizaciones de seguridad

En principio sólo la distribución *estable* (`woody`) tiene soporte *oficial* en lo relativo a las actualizaciones de seguridad.

Cada vez que aparece un error de seguridad y se generan paquetes corregidos se anuncian en la lista `debian-security-announce` (hay un archivo de mensajes disponible en http://lists.debian.org/debian-security-announce/) y se dejan en un almacén específico de actualizaciones de seguridad preparado para funcionar con `apt`.

Es recomendable incluirlo siempre en nuestra lista de origenes de paquetes añadiendo la siguiente línea en el fichero `/etc/apt/sources.list`:

```
deb http://security.debian.org/ woody/updates main contrib non-free
```

Cada vez que aparezcan paquetes en este repositorio deberemos actualizar las listas de paquetes e instalar (siempre que se trate de programas que usemos, claro).

Instalación de Red Hat

Red Hat tiene gran cantidad de versiones de su distribución, algunas pensadas para usuarios finales y otras para servidor. En realidad, salvo que queramos hace uso del soporte de la compañía o queramos instalar algún entorno muy especial la distribución normal nos sirve para instar servidores sin ningún problema.

A continuación describiremos el proceso de instalación que vamos a seguir.

Arrancamos con el CD y el el prompt `boot:` seleccionamos la instalación en modo texto poniendo:

```
boot: linux text
```

Una vez arranca el sistema nos preguntará si queremos comprobar el CD de instalación, cosa que hacemos por si acaso.

Después de una pantalla de presentación seleccionamos el idioma, la disposición del teclado y el tipo de ratón.

A continuación nos aparece un menú que nos permite seleccionar el tipo de instalación que queremos hacer, dado que queremos controlar los programas que instalamos seleccionaremos la opción `personalizada`.

Seguimos con el particionado del disco, podemos optar por el particionado automático o por el particionado manual usando `disk druid`. En princpio le damos al automático, que nos prepara un esquema de particiones simple y nos permite editarlo como en el caso del particionado manual. Una vez tenemos el esquema de particiones que queremos le damos al `OK` para continuar.

El siguiente paso nos permite seleccionar el gestor de arranque que queremos instalar: `GRUB`, `LILO` o ninguno. Dado que en `Debian` hemos probado el `LILO`, aquí seleccionaremos el `GRUB`.

Las pantallas que siguen nos permitirán configurar el `GRUB`:

1. Primero pregunta si necesitamos parámetros especiales para el núcleo,
2. Luego nos preguntará si deseamos proteger con contraseña el gestor de arranque y nos pedirá la contraseña,
3. A continuación nos preguntará los sistemas que queremos arrancar (en nuestro caso sólo tendremos un **Linux**),
4. Y por último preguntará dónde queremos instalar el *bootloader*, en principio siempre usaremos el `MBR`.

La siguiente pantalla nos pide la configuración de la red, deshabilitaremos el uso de bootp/dhcp, seleccionaremos que se active al inicio y rellenaremos los datos que nos solicita.

A continuación nos pedirá el nombre del host.

Después nos permitirá configurar un *cortafuegos*, en principio seleccionaremos un nivel de seguridad alto, personalizándolo para permitir el acceso al `ssh`.

La siguiente sección nos permitirá escoger los idiomas que vamos a emplear, ajustaremos la zona horaria marcando que el reloj del sistema está en GMT (si no usamos otros sistemas esto hace que se cambie automáticamente al horario de verano cuando corresponde).

A continuación el sistema nos solicitará la contraseña del administrador y nos permitirá elegir el sistema de autentificación que vamos a emplear, de entrada usaremos *shadow* y *md5*.

Una vez completado todo este proceso nos aparecerá el sistema de selección de paquetes. Como se trata de una instalación segura eliminaremos las selecciones por defecto de los distintos grupos de paquetes y marcaremos la opción de selección individual para eliminar aquellos paquetes de la instalación mínima que consideremos innecesarios o añadir algunos que sepamos que nos van a hacer falta.

Cuando terminemos con la selección de paquetes comenzará la instalación del sistema.

Una vez termina la instalación nos pregunta si queremos generar un disco de arranque, en principio innecesario, ya que tenemos el CD de RedHat para arrancar en caso de problemas.

Después de ese momento se reinicia el equipo y ya hemos terminado la instalación básica.

Herramientas administrativas y de actualización de RedHat

RedHat emplea el formato de paquete `rpm` y la herramienta principal de manipulación de los paquetes tiene el mismo nombre.

A continuación damos una guía rápida de como usar el `rpm` para realizar tareas comunes de mantenimiento de paquetes:

`rpm --help`

: Muestra las opciones del programa.

`rpm -ivh nompaq-VVV-RRR.rpm`

: Instalar un paquete RPM.

`rpm -Uvh nompaq-VVV-RRR.rpm`

: Actualiza o instala un paquete RPM.

`rpm -Fvh nompaq-VVV-RRR.rpm`

: Refresca un paquete RPM existente.

`rpm -e nompaq`

: Desinstala el paquete RPM de nombre `nompaq`.

`rpm -q nompaq`

: Consulta qué versión del paquete está instalada.

`rpm -qf fichero`

: Consulta a qué paquete (instalado) pertence un fichero.

`rpm -ql nompaq`

: Muestra la lista de ficheros instalados por `nompaq`.

Hay que señalar que, a diferencia de lo que sucede con `dpkg`, en el programa `rpm` no es imprescindible que el paquete esté en el disco, podemos poner un url y el programa se encarga de descargarlo por nosotros.

Actualizaciones de seguridad

Para estar al tanto de las actualizaciones de RedHat se pueden acceder periódicamente al directorio `updates` de la versión que tengamos instalada o se pueden emplear varias herramientas automáticas:

- `up2date`: herramienta de RedHat para actualizar la distribución, actualmente se trata de un servicio de pago.
- `autoupdate`: sistema de actualización capaz de emplear distintos repositorios de paquetes RPM, disponible en: http://www.mat.univie.ac.at/~gerald/ftp/autoupdate/
- `apt-rpm`: versión del `apt` para RedHat, se puede acceder a almacenes con las últimas actualizaciones de la distribución en el la dirección http://apt-rpm.tuxfamily.org/

Instalación de Solaris

Antes de comenzar comentar que existen varios *blueprints* útiles para realizar una instalación segura de Solaris 9:

- *Minimizing the Solaris Operating Environment for Security* http://www.sun.com/solutions/blueprints/1102/816-5241.pdf
- *Solaris Operating Environment Security*: http://www.sun.com/solutions/blueprints/1202/816-5242.pdf

La instalación de Solaris la haremos empleando la *Solaris Webstart Installation* que es la que se ejecuta cuando arrancamos con el *Install CD*.

Los pasos que seguiremos para la instalación son más o menos como sigue:

- Arrancamos desde el CD.
- Selección idioma: Spanish
- Saltamos la configuración del entorno gráfico dándole a F4.
- Formateamos y particionamos el disco, la memoria de intercambio coje 512Mb y se coloca al principio del disco, redistribuimos el espacio de las particiones a automáticas para no quedarnos cortos.
- El programa de instalación copia el ’mini-root’ en el disco local y reinicia.

En el reinicio se debe arrancar del disco duro, hay que revisar la BIOS y el lector de CD.

Una vez rearrancamos nos vuelve a pedir la configuración del entorno gráfico, la evitamos pulsando de nuevo el F4 y se inicia el programa de instalación en modo texto.

El programa nos hace una serie de preguntas para configurar la red, el uso de kerberos, los servicios de nombres, la fecha y la hora, la contraseña del administrador y la gestión de energía. La secuencia de respuestas es más o menos como sigue:

1. Máquina conectada a la red.
2. No se usa DHCP.
3. Le asignamos un nombre a la máquina (sin dominio).
4. Escribimos su IP y su máscara de subred.
5. No usamos IPv6.
6. Especificamos la dirección del encaminador manualmente.
7. No se habilita Kerberos.

8. Usamos DNS como servicio de nombres, especificamos nuestro dominio, las direcciones de los servidores de nombres y los dominios de búsqueda.
9. Seleccionamos zona horaria: Europa/España/península.
10. Configuramos la hora del sistema.
11. Introducimos la contraseña del administrador.

Ahora el sistema nos muestra los valores introducidos y nos pide confirmación.

Una vez a validado la configuración se arranca el asistente de la instalación, en principio le decimos que no queremos ni rearrancar el sistema ni expulsar los CDs de modo automático.

Después seleccionamos CD/DVD como origen de datos para instalar Solaris, el sistema nos pide que insertemos el primer disco de software de Solaris.

Se nos presenta la opción de realizar una instalación predeterminada o personalizada; elegimos esta última.

A partir de aquí nos solicita los idiomas a instalar (de momento los dejamos tal cual), dejamos el entorno nacional español (es_ES.ISO8859-1).

El sistema nos deja seleccionar software adicional para instalar, de momento no instalamos nada de la documentación, quizás sería interesante seleccionar el *Sun Screen 3.2* del disco *Solaris 9 Extra Value Software*, que es el sistema cortafuegos que pensamos utilizar.

Cuando nos pregunta si queremos añadir más software le decimos que no.

A continuación nos permite elegir grupos de software para instalar, de entrada seleccionamos el quinto, *Grupo de núcleo*, que es el mínimo necesario para un sistema Solaris.

Una vez seleccionado se nos da la opción de personalizar la selección de paquetes, cosa que hacemos para instalar algunas utilidades que nos interesan y eliminar otras cosas.

Eliminamos las siguientes opciones:

- FTP Server (clusters 30 y 31).
- NFS Server (cluster 92).
- NIS (cluster 93).
- PCMCIA (cluster 100).
- Remote network services and commands (cluster 117).

Añadimos las siguientes opciones:

- Freeware Shells, BASH (cluster 41, seleccionamos la 41,0).
- Freeware Compresión Utilities (cluster 39, las seleccionamos todas).
- Freeware Other Utilities (cluster 40, las seleccionamos todas).
- gcmn - GNU common package (cluster 189).
- GNU wget (cluster 43)
- NTP (cluster 94).
- Sun Workshop Compilers Bundled libC (cluster 148).
- Documentation Tools (cluster 27).
- On-Line manual pages (cluster 96).
- Secure Shell (cluster 126).

- ggrep (cluster 190).
- gtar (cluster 191).
- tcpd (cluster 197).

Para muchos programas puede que necesitemos usar las X Windows, un conjunto mínimo de paquetes que nos permite usarlas es:

- SUNWctpls: Portable layout services for Complex Text Layout support
- SUNWmfrun: MotifRunTime Kit
- SUNWxwdv: X Window System Kernel Drivers
- SUNWxwfnt: X Window System Fonts
- SUNWcpp: Solaris C Pre-Processor (cpp)
- SUNWxwplt: X Window System platform software
- SUNWxwice: X Window System Inter-Client Exchange (ICE) Components
- SUNWxwpls: X Server x86 platform software
- SUNWxwrtl: X Window System & Graphics Runtime Library Links in /usr/lib

Por desgracia para muchas cosas nos puede hacer falta `Java`, así que lo instalamos:

- SUNWj3rt: J2SDK 1.4 runtime environment

Para instalar más tarde el Sun Screen debemos instalar:

- SUNWsprot: Solaris Bundled tools
- SUNWtoo: Programming Tools
- SUNWeu8os: American English/UTF-8 L10N For OS Environment User Files
- SUNWapchr: The Apache HTTP server program (root components)
- SUNWapchu: The Apache HTTP server program (usr components)

Por requisitos seleccionamos:

- 175,3
- 164

Una vez seleccionado todo el sistema nos pregunta si queremos borrar el disco, le decimos que sí y que lo haga todo automático. El tamaño de particiones queda un poco extraño y lo corregimos, asignando más de un giga al sistema de archivos raiz.

A partir de aqui comienza la instalación automática, hasta que se reinicia el sistema, ya instalado.

Herramientas administrativas y de actualización de Solaris

Solaris emplea el formato de paquetes de SysV, en general los paquetes usan la extensión `.pkg`. Para manipularlos desde la línea de comandos emplearemos varios programas:

- `pkgadd`: añade un paquete de software
- `pkgrm`: elimina un paquete de software

- `pkgchk`: verifica si un paquete está correctamente instalado
- `pkginfo`: muestra información sobre los paquetes
- `pkgask`: guarda las respuestas de una instalación en fichero, se emplea para automatizar instalaciones.
- `pkgparam`: muestra parámetros asociados a un paquete

Para instalar un paquete haremos:

```
# pkgadd -d <ruta> [<nom_paquete>]
```

Donde ruta indica un dispositivo o directorio que contiene paquetes. Si no ponemos el nombre del paquete se instalan todos los paquetes que hay en el directorio.

Para eliminar un paquete

```
# pkgrm <nom_paquete>
```

Para obtener la lista de paquetes instalados

```
# pkginfo
```

Para obtener información sobre un paquete

```
# pkginfo -x <nom_paquete>
# pkginfo -l <nom_paquete>
```

Para saber a qué paquete pertenece un fichero

```
# pkgchk -l -p /ruta/al/archivo
```

Para saber qué ficheros hay en un paquete haremos

```
# grep <paquete> /var/sadm/install/contents
```

Hay que indicar que la información de instalación de los paquetes se guarda en subdirectorios de `/var/sadm/pkg` que tienen el mismo nombre que el paquete instalado.

Actualizaciones de seguridad

Para acceder a los parches oficiales de Sun se puede consultar la web http://sunsolve.sun.com/, en general los parches se distribuyen en formato `.zip` con una nomenclatura estándar: `numparche_rev.zip`, donde el primer elemento es en número de parche y el segundo la revisión del mismo.

Para instalar y desinstalar los parches se usarán las herramientas `patchadd` y `patchrm`.

Para aplicar un parche:

```
# patchadd numparche_rev
```

Siendo `numparche_rev` el directorio donde hemos desempaquetado el `.zip` proporcionado por **Sun**.

Para ver la lista de parches aplicados haremos:

```
# patchadd -p
```

Para ver los parches aplicados a un paquete haremos:

```
# pkgparam nombre_paquete PATCHLIST
```

Para ver la información de un parche concreto aplicado a un paquete haremos:

```
# pkgparam nombre_paquete PATCH INFO numpatch
```

Para borrar un parche haremos:

```
# patchrm numparche_rev
```

La información sobre los parches instalados en un solaris se encuentra en el directorio `/var/sadm/patch`; para cada parche aplicado se guarda información en subdirectorios con el nombre ..numparche_rev„. Además de esto en el directorio „/var/sadm/pkg/NOMBRE_PAQUETE/save/numparche_rev qué contiene los ficheros que fueron eliminados al aplicar el parche, por si hace falta dar marcha atrás con el `patchrm`.

Además del uso de los progaramas que acabamos de comentar, existe una herramienta para automatizar la instalación de parches accesible en el URL https://sunsolve.sun.com/patchpro/.

Si instalamos software empaquetado por terceros como el que hay en http://www.sunfreeware.com/ podemos usar las herramientas estándar de manejo de paquete y también tenemos a nuestra disposición herramientas similares a APT como `pkg-get` (http://www.bolthole.com/solaris/pkg-get.html) y `pkgadm` (http://www.bolthole.com/solaris/pkgadm.html).

Configuración inicial del sistema

Protección del sistema de arranqe

Como ya hemos explicado anteriormente, si queremos que nuestra máquina no pueda ser manipulada por alguien con acceso físico *límitado* a la misma deberemos proteger el acceso a la BIOS para evitar que se pueda arrancar desde un dispositivo que no sea el disco duro (si es un servidor en producción no pondremos clave de arranque al PC, ya que si se va la luz haría falta entrar directamente en la máquina para reiniciar) y además deberemos proteger el *bootloader* para que no permita modificar los parámetros de arranque del núcleo.

El método para hacerlo ya ha sido explicado en puntos anteriores del curso.

Sistemas de autentificación de usuarios y elección de contraseñas

En principio usaremos la autentificación tradicional de Unix (usando los ficheros `/etc/passwd` y `/etc/group`) pero con *shadow passwords* (el fichero de claves, que es legible para todos los usuarios, no contendrá las versiones cifradas de las contraseñas, estas estarán en el fichero `/etc/shadow`, que sólo puede leer el administrador) y cifrado de claves con `MD5` en lugar de `crypt` (nos permite usar claves más largas y es más seguro criptográficamente hablando).

Aunque hemos dicho que tanto en **Linux** como en **Solaris** vamos a usar ficheros tradicionales, lo cierto es que lo que vamos a usar son los módulos de autentificación tradicional de Unix de **PAM** (”Pluggable Autentication Modules”), que es una biblioteca con la que se enlazan la mayoría de programas que requeren identificar a los usuarios.

PAM es un mecanísmo flexible para la autentificación de usuarios, permite que el administrador configure para cada aplicación un sistemas de identificación distinto sin necesidad de modificar el código del programa.

El PAM permite configurar 4 tipos distintos de módulos:

auth

Módulos de autentificación, comprueban que una clave corresponde a un usuario.

account

Módulos de contabilidad, realizan verificaciones que no tienen que ver con las claves, como comprobar la caducidad de los 'passwords' o ver

si el usuario puede acceder a una determinada hora.

session

Módulo de session, se emplea para realizar tareas que no tienen nada que ver con la gestión de claves, como guardar un log de inicio de sesión, montar o crear directorios, etc.

password

Módulo empleados para el cambio de claves.

Dependiendo del uso que se vaya a hacer del sistema puede ser muy interesante conocer como funciona el PAM, para más información se pueden mirar los siguientes sitios WEB:

- Linux-PAM: http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/
- Solaris PAM: http://wwws.sun.com/software/solaris/pam/

Sistemas cortafuegos

Cortafuegos en Linux

En **Debian** y **RedHat** usaremos `netfilter`, que es el subsistema cortafuegos incluído en los núcleos de Linux en las series 2.4.x y posteriores. Este sistema nos proporciona funcionalidad de cortafuegos (con y sin estado), varios tipos de NAT (*Network Address Translation*) y nos permite manipular los paquetes.

En muchas ocasiones emplearemos el nombre `iptables` al nos referiremos a `netfilter`. En realidad lo que sucede es que el sistema se llama `netfilter`, pero la herrmienta que emplearemos para configurarlo se llama `iptables`.

Este sistema es muy potente y permite hacer gran cantidad de cosas con los paquetes que entran, salen y atraviesan nuestros sistemas; aquel que esté interesado encontrará mucha información de como usarlo en: http://www.netfilter.org/.

En lo que sigue explicaremos como instalar el cortafuegos inicial en un sistema Debian GNU/Linux, ya que la última versión de *RedHat* nos ayuda a configurar un cortafuegos sencillo durante la instalación y nos genera un script similar al que proponemos para debian.

Para configurar el cortafuegos emplearemos el programa `iptables`; en principio lo que haremos será arrancar el sistema sin conexión física a la red o en modo mono usuario (*runlevel 1*) y una vez identificados como administradores haremos:

```
cseg:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
```

```
Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Esto nos dice que no tenemos ninguna regla configurada.

A continuación presentamos los comandos que ejecutaríamos para configurar un cortafuegos sencillo:

```
# Por defecto no aceptamos nada en la entrada ni en el forwarding
iptables -P INPUT   DROP
iptables -P FORWARD DROP
# Pero aceptamos los paquetes de salida
iptables -P OUTPUT  ACCEPT

# Definimos una tabla para registrar conexiones y aceptarlas
iptables -N LOGACCEPT
iptables -A LOGACCEPT -m limit --limit 3/hour -j LOG --log-prefix "LOG_ACCEPT: "
iptables -A LOGACCEPT -m limit --limit 3/hour -j ACCEPT

# Definimos una tabla para registrar conexiones e ignorarlas
iptables -N LOGDROP
iptables -A LOGDROP -m limit --limit 3/hour -j LOG --log-prefix "LOG_DROP: "
iptables -A LOGDROP -m limit --limit 3/hour -j DROP

# Input
# -----
# Loopback -- aceptamos todo
iptables -A INPUT -i lo -j ACCEPT
# Aceptamos conexiones establecidas
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# Aceptamos paquetes ICMP
iptables -A INPUT -p icmp --icmp-type pong -j ACCEPT
iptables -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
iptables -A INPUT -p icmp --icmp-type ping -j ACCEPT
iptables -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
# Aceptamos conexiones SSH
# Logging de inicios de conexión al SSH
iptables -A INPUT -p tcp -m state --state NEW --dport 22 -j LOGACCEPT
iptables -A INPUT -p tcp -m tcp   --dport 22 -j ACCEPT
# Para aceptar conexiones HTTP hariamos:
# iptables -A INPUT -p tcp -m tcp   --dport 80 -j ACCEPT
# Para ignorar BROADCASTS hariamos:
# iptables -A INPUT -p tcp -m pkttype --pkt-type broadcast -j DROP
# Registrar e ignorar el resto de paquetes
iptables -A INPUT -j LOGDROP

# Forward
# -------
# Registrar paquetes que intenten hacer forwarding
iptables -A FORWARD -j LOGDROP

# Output
# ------
# Filtrar paquetes ICMP desconocidos
iptables -A OUTPUT -m state -p icmp --state INVALID -j LOGDROP
```

Una vez ejecutados todos estos comandos la configuración del sistema de cortafuegos se puede ver haciendo:

```
# iptables -L
```

Para que la configuración no se pierda al reiniciar el equipo usaremos el script `/etc/init.d/iptables`; en primer lugar almacenaremos como configuración activa el firewall que acabamos de instalar:

```
# /etc/init.d/iptables save active
```

Despues limpiaremos las reglas, volveremos a poner el *forwarding* a DROP (por si acaso) y guardaremos las reglas de firewall inactivo:

```
# /etc/init.d/iptables clear
# iptables -P FORWARD DROP
# /etc/init.d/iptables save inactive
```

Para hacer que el firewall se active cada vez que arranquemos la máquina reconfiguraremos el paquete `iptables`:

```
# dpkg-reconfigure iptables
```

Diciéndole que si queremos que active el cortafuegos al arrancar.

Ahora activaremos el firewall y conectaremos el equipo a la red o saldremos del modo mono usuario.

Por último indicar que si no conocemos bien como funciona el sistema de firewalling de Linux debemos tener cuidado con la configuración y en lugar de usar un sistema tan manual como el presentado es recomendable emplear alguno de los muchos sistemas de configuración de *cortafuegos* disponibles en la distribución como por ejemplo *firestarter*, *fwbuilder* o *shorewall*.

Cortafuegos en Solaris

Para **Solaris** podemos emplear varios sitemas cortafuegos, el que viene en los CD de Solaris 9 es el *Sun Screen 3.2*. Hay que indicar que está como software adicional, es decir, no va incluido en la instalación estándar del sistema.

La documentación oficial de esta versión está en en la web de documentos de Sun (http://docs.sun.com/db/coll/557.4). Además de esta documentación, en el URL http://www.bolthole.com/solaris/sunscreen.html hay una guía rápida de como configurarlo y hay un *Blueprint* denominado *Securing Systems with Host-Based Firewalls - Implemented With SunScreen Lite 3.1 Software* que está disponible en http://www.sun.com/solutions/blueprints/0901/sunscreenlite.pdf.

Además de este firewall existe otro de dominio público que tambien es muy interesante; se llama *IP Filter* y es el incluido por defecto el FreeBSD y NetBSD. Se puede conseguir en http://coombs.anu.edu.au/ipfilter/

En el resto de este punto vamos a explicar como instalar y configurar el SunScreen 3.2 para funcionar como sistema cortafuegos de *host*.

Para empezar insertamos el segundo CD de software de Solaris 9 y nos ponemos en el directorio de software adicional:

```
# cd /cdrom/Solaris_9/ExtraValue/CoBundled/SunScreen_3.2/i386
```

Para ver la lista de paquetes hacemos:

```
# pkgadd -d .
```

De estos instalamos los paquetes 2, 1, 3, 6-13, 18-20, 14-17.

Para configurar el firewall haremos:

```
# ssadm configure
```

E iremos contestando lo siguiete a las preguntas que nos aparecen:

```
1. ROUTING
2. STEALTH

Screen Type? 1

1. LOCAL
2. REMOTE
3. BOTH

Local, remote or both methods of administration? 1

The security levels are as follow:

1. Restrictive
2. Secure (routing screens only)
3. Permissive

Select the initial security level for this Screen: 3

The following name resolution method was detected on this machine:DNS -
Domain Name Service.

1. YES
2. NO

Is this is the name service that you want to use on this machine? 1
```

A partir de aquí reiniciamos la máquina y ya tenemos un firewall activo que en principio no hace nada.

Para ajustar las políticas miramos la que está activa:

```
# ssadm active
Active configuration: cseg default Initial.1
Activated by root on Tue Jul 15 15:08:30 2003
```

Y la editamos:

```
# ssadm edit Initial
Loaded common objects from Registry version 1
Loaded policy from Initial version 1
edit> list rules
1 "common" "*" "*" ALLOW
```

Lo primero que hacemos es eliminar la regla ’promiscua’:

```
edit> delete rule 1
```

A partir de este punto podemos añadir *objetos* de varios tipos: direcciones, servicios, interfaces, etc.

Por ejemplo podemos añadir algunos elementos de red:

```
edit> add ADDRESS dns_srv HOST 192.168.96.1 COMMENT "Servidor DNS"
edit> add ADDRESS smtp_srv HOST 192.168.96.1 COMMENT "Servidor SMTP"
edit> add ADDRESS adm_net RANGE 192.168.96.65 192.168.96.91
```

Y también algunos servicios de red que no están disponibles por defecto, como el `https`:

```
edit> add SERVICE https SINGLE FORWARD "tcp" PORT 443
```

Podemos agrupar el `http` (`www` en SunScreen) y el `https` en un sólo grupo:

```
edit> add SERVICE web GROUP www https
```

Ahora añadiremos reglas para habilitar el acceso desde la máquina local a la `web` (`http` y `https`) y al `dns` y `smtp` de los servidores de la red:

```
edit> add rule "web" "localhost" "*" ALLOW COMMENT "Acceso web de salida"
edit> add rule "dns" "localhost" "dns_srv" ALLOW COMMENT "Acceso al DNS"
edit> add rule "smtp" "localhost" "smtp_srv" ALLOW COMMENT "Acceso al SMTP"
```

Para poder administrar la máquina habilitaremos el acceso al `ssh` desde la red de los administradores:

```
edit> add rule "ssh" "adm_net" "*" ALLOW LOG SUMMARY COMMENT "Permitir ssh"
```

Para evitar problemas aceptamos todos los paquetes `icmp` (podríamo ser más estrictos, pero esto es cómodo).

```
edit> add rule "icmp all" "*" "*" ALLOW COMMENT "Aceptamos todos los ICMP"
```

Y al final denegamos el resto con *Logging*:

```
edit> add rule "*" "*" "*" DENY LOG SUMMARY COMMENT "Logging"
```

Nuestras reglas quedarán:

```
edit> list rule
1 "web" "localhost" "*" ALLOW COMMENT "Acceso web de salida"
2 "dns" "localhost" "dns_srv" ALLOW COMMENT "Acceso al DNS"
3 "smtp" "localhost" "smtp_srv" ALLOW COMMENT "Acceso al SMTP"
4 "ssh" "adm_net" "*" ALLOW LOG SUMMARY COMMENT "Permitir ssh"
5 "icmp all" "*" "*" ALLOW COMMENT "Aceptamos todos los ICMP"
6 "*" "*" "*" DENY LOG SUMMARY COMMENT "Logging"
```

Para activar el firewall grabamos las reglas:

```
edit> save
Saved common objects to Registry version 2
Saved policy to Initial version 2
```

Las validamos:

```
edit> verify
Configuration verified successfully (not activated).
```

Salimos y la activamos:

```
edit> quit
# ssadm activate Initial
Configuration activated successfully on cseg.
```

Si por lo que sea preferimos volver a la configuración anterior haremos:

```
# ssadm activate Initial.1
```

Para más detalles del uso del SunScreen 3.2 se puede consultar la documentación de Sun sobre el producto, disponible en http://docs.sun.com/db/coll/557.4.

Deshabilitar servicios inútiles

En general, después de reiniciar la máquina y configurarle el cortafuegos, intentaremos ver que servicios tenemos activos empleando los programas `netstat` y `lsof`. El primero de ellos nos servirá para saber que puertos tenemos escuchando y el segundo nos dirá que procesos son los que escuchan.

El primer demonio que deberemos parar será el `inetd`. Una vez esté parado miraremos el fichero de configuración `/etc/inetd.conf` y comentaremos todos los servicios que no necesitemos. Si cuando terminamos no queda ningún servicio podemos desactivar el arranque del demonio al inicio del sistema o incluso desinstalar el programa.

A continuación pararemos y deshabilitaremos el `portmap` y los demonios del `nfs` (salvo que los vayamos a utilizar, claro está).

Seguiremos haciendo lo mismo con los servicios que nos vayan apareciendo, en los casos en los que los programas si sean necesarios intentaremos limitar el acceso a los mismos al mínimo número de máquinas posibles; incluso si sólo es necesario para los usuarios o programas locales lo configuraremos para que escuche únicamnte en la dirección IP `127.0.0.1`.

Debian

Miramos los servicios de red arrancados por defecto, podemos hacerlo mirando la salida del proceso de arranque, mirando los procesos en marcha con `top` o `ps`, o, dado que se trata de procesos de red, usando la salida del comando `netstat` y/o `lsof`:

```
# netstat -tulpen
```

Nos dará una lista de puertos TCP (opción `t`) y UDP (opción `u`) en estado LISTEN (opción `l`) indicándonos los programas a los que pertenecen los *sockets* (opción `p`). Las opciones que restan nos dan información extendida (opción `e`) y evitan el uso de los sistemas de resolución de nombres (opción `n`).

Podemos obtener información similar del comando `lsof` haciendo `lsof -i`, que nos lista los ficheros de red abiertos y los programas que los están usando.

Una vez sabemos que procesos queremos que no se reinicien en el arranque podemos eliminar los enlaces de arranque en los distintos runlevels:

```
# rm /etc/rc*.d/S*nombre_demonio
```

o simplemente eliminar los paquetes con `dpkg --purge`.

RedHat

Al igual que en *debian*, miramos los puertos que están escuchando y los programas que los tienen abiertos y cerraremos lo que no nos haga falta.

En principio detendremos el `portmap` y el `rpc.statd`:

```
[root@cseg root]# /etc/init.d/portmap stop
Parando portmapper:                                        [  OK  ]
[root@cseg root]# /etc/init.d/nfslock stop
Parada de NFS statd:                                       [  OK  ]
```

Ahora tendremos:

```
[root@cseg root]# netstat -tuln
Active Internet connections (only servers)
```

```
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN
[root@cseg root]# lsof -i -n
COMMAND   PID USER   FD   TYPE DEVICE SIZE NODE NAME
sshd     1637 root    3u  IPv4   2043       TCP *:ssh (LISTEN)
sendmail 1661 root    4u  IPv4   2108       TCP 127.0.0.1:smtp (LISTEN)
```

Dejaremos activo el `ssh` para poder administrar el equipo de forma remota y en principio mantendremos el `sendmail` funcionando en la dirección de *loopback*, aunque sería recomendable revisar su configuración (en la validación de la configuración de los servicios básicos de *Solaris* comentaremos como configurarlo el `sendmail` para que funcione únicamente como servidor local, sin escuchar en el puerto de `smtp`, ver http://www.samag.com/documents/s=8228/sam0306a/0306a.htm).

Si no queremos que en el próximo arranque se lancen los servicios que hemos detenido haremos:

```
[root@cseg root]# chkconfig portmap off
[root@cseg root]# chkconfig nfslock off
```

Esto los deshabilitará en todos los `runlevels`. Si sabemos con certeza que no vamos a necesitar ni el `portmap` ni el `nfs`, podemos borrar los paquetes con `rpm -e`.

A partir de aquí podemos comenzar a instalar los servicios que necesite nuestro sistema, recordando que tenemos un *firewall* habilitado.

Solaris

Hacemos lo mismo que en los otros sistemas, aunque empleamos instrucciones diferentes, para el `netstat` haremos:

```
# netstat -an | grep LISTEN
```

Para la detección de demonios que tienen abiertos los distintos puertos no hay ninguna herramienta (al menos que yo conozca) que venga de serie con Solaris, por lo que instalaremos el `lsof` de `www.sunfreeware.com`.

Parámetros del kernel

En principio en la mayoría de sistemas operativos existen parámetros ajustables que son importantes ante distintos ataques contra nuestro sistema de red, por ejemplo podemos controlar si nuestros sistemas *reenvían* paquetes `TCP` entre distintas interfaces de red, podemos decidir si aceptamos o no los `ICPM Redirect`, podemos hacer que se validen las rutas de los paquetes para evitar el IP Spoofing, etc.

En los siguientes documentos se pueden encontrar análisis de los parametros que podemos manipular en los núcleos de Solaris (usando el programa `ndd`) y Linux (usando el programa `sysctl` o manipulando directamente ficheros de `/proc`):

- *System and Network Security - Kernel Options* (BSD, Linux y Solaris): http://www.seifried.org/security/technical/20020307-kernel-options.html
- *Ipsysctl tutorial 1.0.4* (ajuste de variables en Linux 2.4.x): http://ipsysctl-tutorial.frozentux.net/ipsysctl-tutorial.html
- *Solaris Operating Environment Network Settings for Security*: http://www.sun.com/solutions/blueprints/0603/816-5240.pdf

Además de los ajustes del núcleo relacionados con la red, no está de más mencionar que existen varios conjuntos de parches para el nucleo de Linux que lo hacen menos vulnerable a todo tipo de ataques (no sólo de red), entre ellos podemos mencionar los siguientes:

- Security Enhaced Linux o SE Linux (http://www.nsa.gov/selinux/)
- Rule Set Based Access Control (RSBAC) for Linux (http://www.rsbac.org/)
- Grsecurity (http://www.grsecurity.net/)
- Linux Security Modules (http://lsm.immunix.org/)

Existen distribuciones orientadas a la seguridad basadas en RedHat y Debian que incluyen estos núcleos:

- Security Enhaced Linux se distribuye junto con una RedHat 7.2.
- SE Linux para Debian (http://www.coker.com.au/selinux/).
- Adamantix (http://www.adamantix.org/), basada en Debian, usa RSBAC.

Epílogo

Después de una instalación como la propuesta aquí podemos tener unas ciertas garantías de que nuestro equipo es menos vulnerable que con una instalación por defecto, pero aun nos quedán multitud de tareas por hacer; en el planteamiento inicial del curso se plantearon varios apartados más, que quizás serán parte de una segunda parte de este curso:

- Seguridad local; en donde se hablaría de problemas de seguridad relacionados con los usuarios ’locales’ del equipo (ya sean reales o los empleados por programas servidores) y los mecanísmos de protección que podemos utilizar.
- Seguridad de red; donde se discuten los problemas relacionados con los servicios accesibles a través vía red y las soluciones que podemos emplear. En este apartado tambien se tratarán los problemas relacionados con la administración remota de los equipos.
- Herramientas de auditoría y monitorización; en este apartado se tratarían las herramientas que nos permiten comprobar si nuestras máquinas son vulnerables a ataques y detectar los ataques que pueden sufrir.
- Configuración segura de algunos servicios; donde se hablaría de la configuración de servidores de protocolos muy habituales como el SMTP, HTTP, etc.