Diseño de Sistemas Operativos:

Introducción a la Administración UNIX

Estas trasparencias han sido desarrolladas para la asignatura "*Diseño de sistemas operativos*" © J.M. Peña

Índice

- Introducción
- Directorios del sistema
- Gestión de usuarios
- Arranque del sistema
- Variantes de shells
- Discos y sistemas de ficheros
- Servicios de red
- Servicios internos
- Instalación de nuevo software
- Terminales gráficos
- Interoperabilidad con otros SSOO
- Auditoría del sistema
- Seguridad en sistemas Unix

Introducción

- Tareas del administrador:
  - Servicio a los usuarios.
  - Mantenimiento y actualización del software.
  - Auditoría de seguridad y rendimiento del sistema.
  - Gestión de recursos.

Herramientas de Administración

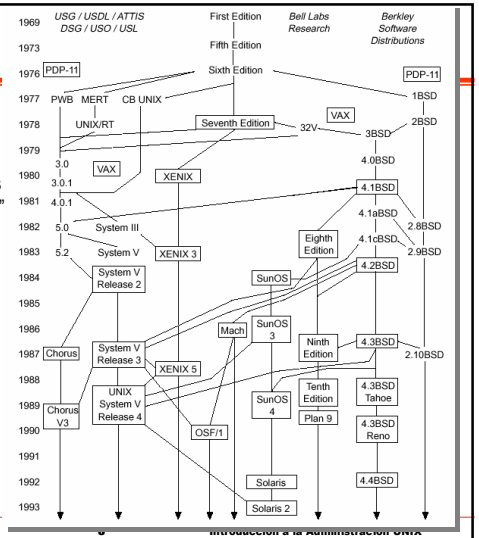
Casi todos los sistemas operativos UNIX tienen su propio conjunto de herramientas de administración:

- admintool (Sun Solaris)
- control-panel (Linux-RedHat)
- Yast (Linux-SuSe).
- smit (IBM AIX).
- sysadmsh (XENIX).

La administración de sistemas del "día a día" se hace por medio de estas herramientas.

La evolución de los sistemas UNIX ha causado que esta familia de sistemas operativos guarde “pequeñas diferencias” entre ellos.

A nivel de administración estas diferencias se agudizan más.



Tipos de Instalaciones

Se pueden dividir las instalaciones en tres diferentes categorías:

- Estaciones de trabajo monousuario:
  - Administración sencilla (uno o muy pocos usuarios).
  - Administrador == usuario.
  - Instalaciones “poco críticas”.
- Servidores multiusuario:
  - Mayor número de usuarios.
  - Arbitraje de recursos (limitaciones y privilegios).
  - Modificaciones más delicadas.
- Clusters de máquinas:
  - Red: Problemas de seguridad y mayor complejidad.
  - Para gran cantidad de máquinas: automatización de tareas.
  - La complejidad crece sustancialmente.

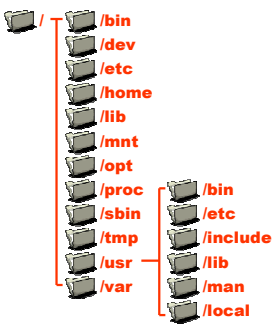
Conocimientos del Administrador

Es recomendable para el administrador el conocimiento de:

- Funcionamiento interno del sistema:
  - Diseño interno del sistema operativo.
  - Permite comprender qué hace cada operación, causas y motivos del funcionamiento del sistema.
- Seguridad y comunicaciones:
  - En la actualidad los equipos están en red.
  - Los servicios de red de una máquina son cruciales.
- Programación (scripts, perl, awk, ...):
  - Automatización de tareas: “Si lo necesitas una vez lo vas a hacer varias”.
- Instalación de componentes hardware:
  - Discos duros, periféricos, ...
- Otros sistemas operativos:
  - “Conoce a tu enemigo” ☺

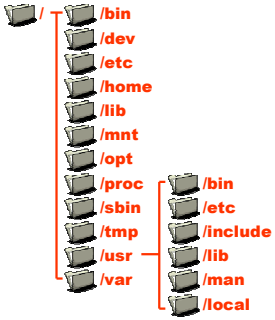
Árbol de Directorios UNIX

- `/bin`: Ejecutables básicos del SSOO. Incluye los mandatos tipo ls, cp, ... En algunos UNIX aquí están los mandatos enlazados estáticamente.
- `/dev`: Ficheros especiales asociados a dispositivos. UNIX proyecta los dispositivos como ficheros de este directorio (interfaz común).
- `/etc`: Configuración del sistema. Ficheros de configuración de servicios, arranque, etc.
- `/home`: Directorio de cuentas de usuarios. Cada usuario poseerá un directorio aquí. Puede estar dividido en varios niveles (por organización).
- `/lib`: Librerías básicas del sistema. Librerías del Kernel o comunes a muchos ejecutables.



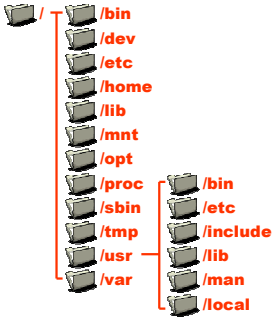
Árbol de Directorios UNIX

- `/mnt`: Directorio de montaje de ciertos sistemas de ficheros. Por lo general, vacío o con un nivel de directorios. Se utiliza para montar otros dispositivos.
- `/opt`: Aplicaciones adicionales del sistema. En principio todo aquello que se instale fuera del SSOO estándar (el del fabricante).
- `/proc`: Sistema de ficheros virtual para la gestión de recursos. El kernel presenta en este directorio información del sistema.
- `/sbin`: Ejecutables de administración del SSOO. Subconjunto de mandatos con privilegios. En principio un usuario no tiene por qué tenerlos en el PATH.



Árbol de Directorios UNIX

- `/tmp`: Directorio para ficheros temporales. Ficheros auxiliares de aplicaciones. Todos los usuarios puedes escribir en este directorio.
  - En algunos UNIX se comparte con el swap.
- `/usr`: Aplicaciones adicionales del SSOO. Este directorio contiene los mandatos que no son básicos (ls, cp, etc.) pero que se distribuyen con la instalación del fabricante.
- `/usr/local`: Programas locales del sistema. Análogo al `/opt` (a veces un enlace simbólico).
- `/var`: Directorio para ficheros de log y colas de trabajos. Lo usa el sistema para guardar el registro de operaciones (accesos, errores y otros mensajes), así como colas de determinados servicios (correo o impresora).



Directorio `/dev`

Agrupa entradas de tres diferentes tipos:

- Dispositivos de tipo carácter: terminales o cintas.
- Dispositivos de tipo bloque: discos principalmente.
- Dispositivos virtuales (`/dev/zero`: dispositivo que genera el carácter 0 o `/dev/null`: sumidero de cualquier escritura).

Todos estos dispositivos tienen asociado un *minor* y un *major number*. Estos parámetros le valen al kernel para saber cómo tratar una lectura/escritura sobre el dispositivo:

- Major number: Identifica el manejador que lo va a tratar, qué código del SSOO maneja ese dispositivo (`/proc/devices`).
- Minor number: Parámetro adicional que recibe el manejador.

```
crw-rw-rw-   1 root     root        1,   5 May 31  2002 zero
```

Dispositivo carácter — Major number — Minor number

Directorio `/proc`

Se corresponde con un sistema de ficheros virtual (no tiene soporte en disco). Las entradas del directorio son:

- Procesos en ejecución.
- Información del sistema.
- *Mapping* de recursos del sistema.

El contenido de este directorio depende mucho del sistema.

El sistema operativo:

- Al registrar entradas en el directorio guarda una pareja de funciones (una para lectura y otra para escritura).
- Una operación de lectura (e.g.: `cat /proc/cpuinfo`) en realidad ejecuta una función dentro del kernel (no lee verdaderamente de ningún dispositivo).
- Ídem para la escritura.

Propietarios de los Directorios

- La mayoría de directorios y ficheros pertenecen al usuario root o a otros usuarios privilegiados del sistema.
- Las excepciones son:
  - Cuentas de usuario: /home
    - Cada usuario tiene un directorio del cual es propietario.
  - Ficheros temporales: /tmp
    - El directorio /tmp tiene asociado el flag +t (*sticky*) que permite que cualquier usuario pueda crear entradas aun sin ser propietario de dicho directorio pero no escribirlas o borrarlas.
  - Entradas de las colas de trabajos: /var
    - En concreto, los mensajes de correo (habitualmente /var/spool/mail).
  - Las imágenes de los procesos: /proc
    - Por motivos de coherencia y para ciertas operaciones, la información de procesos pertenece al mismo propietario al que está asociado el proceso.

Gestión de Usuarios

Creación de un usuario:

1. Insertarlo en el fichero de usuarios:
   - Se le asigna un nombre, un identificador y un directorio de trabajo (entre otras cosas).
2. Asignarle un *password*:
   - La *pasword* se cifra y se asocia al usuario.
3. Definir parámetros (límites):
   - Número de procesos máximo, memoria, etc...
4. Crear el directorio *home*:
   - Verificar espacio disponible y organizar las cuentas de forma manejable.
5. Copiar ficheros iniciales:
   - El directorio `/etc/skel` suele contener los ficheros básicos para una nueva cuenta de usuario (esqueleto de una cuenta).
6. Cambiar el propietario del *home*:
   - Se ajustan los permisos del directorio (y su contenido).
   - El directorio creado originalmente pertenecerá al administrador, ahora se hace un **chown** para que pertenezca al usuario.
7. Dar de alta en *mail*, *quota*, ...
   - Dependerá de los servicios disponibles en el sistema.

Gestión de Usuarios

Los ficheros habituales para la gestión de usuarios son:

- Fichero de usuarios: **`/etc/passwd`**
  `usuario:passwd:uid:gid:desc:home:shell`
- Fichero de grupos: **`/etc/group`**
  `grupo:<reserved>:gid:usuarios...`

```
chema:x:1500:200:Jose M. Peña:/home/chema:/bin/bash
```

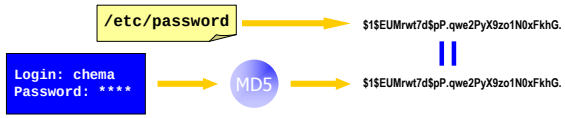
La información almacenada es:

- Usuario (**`chema`**): Identificador interno del sistema. Login en la máquina.
- *Password* (**`x`**): Para evitar que el fichero **`/etc/passwd`** (de lectura pública) contenga las *passwords* cifradas (para evitar ataques de fuerza bruta).
- UID/GID (**`1500:200`**): Identificadores de sistema para el usuario y su grupo principal.
- Descripción del usuario (**`Jose M. Peña`**): Información personal del usuario.
- Home (**`/home/chema`**): Directorio desde donde arranca el login, raíz de su cuenta.
- Shell (**`/bin/bash`**): Programa que se ejecuta en el login. Generalmente un *shell*.

Passwords

Formato de las claves:

- Proceso de traducción no invertible. Funciones unidireccionales.
- La validación de passwords se hace cifrando la secuencia tecleada por el usuario y comparándola con la password cifrada. No se descifra.



Protección de los ficheros de claves: (Problema **`/etc/passwd`** de lectura para todos)

- Fichero de Passwords: **`/etc/shadow`**
  `usuario:passwd:parámetros...`
  - Se eliminan las passwords del fichero **`/etc/passwd`**
  - Se crea un fichero **`/etc/shadow`** con las claves y de acceso más restrictivo.

```
-rw-r--r--    1 root     root         1156 Jul  3  2002 /etc/passwd
-rw-r-----    1 root     shadow        977 Jul  3  2002 /etc/shadow
```

Login de un Usuario

Al conectarse un usuario al sistema:

- Se evalúa si el modo de conexión (local o remoto) es válido.
  - Puede haber restricciones a nivel del usuario (por ejemplo, en base al `getty`, el *root* no se debe conectar de forma remota).
- Se arranca el programa *shell* asociado.
  - Este programa puede no ser un shell propiamente dicho.
- Configuración de la sesión:
  - Configuración general (`/etc/profile`) y
  - Configuración por usuario (`~/.profile`).

Deshabilitar Usuarios

- Para cerrar o deshabilitar una cuenta (sin borrar su contenido):
  - Bloquear el password de la cuenta sustituyéndola por '`*`'.
    - También podemos usar: `passwd -l` y `passwd -u`
  - Cambiar el *shell* de acceso por `/bin/false` o un mensaje.
    - Por ejemplo, el shell del usuario podría ser un *script* como éste:

```
#!/bin/tail +2
La cuenta se encuentra bloqueada.
Hable con el administrador.
```

El código de este *script* muestra el mensaje en pantalla.

- Si dicho *script* se pone como *shell* por defecto del usuario le saldrá el mensaje.
- Una vez finalizado el *shell* de un usuario se sale de la cuenta.

Cambio de Usuario

Para cambiar de usuario al iniciar una sesión se usa el mandato `su`:

- `su - usuario`: Cambia de usuario…
  - Carga los ficheros de configuración (.profile, por ejemplo) del usuario.
  - Cambia al directorio HOME del usuario.
- `su usuario`: Sólo cambia de usuario.

Si no se indica el usuario se cambia al usuario `root`.

Algunos sistemas disponen de una herramienta (llamada `sudo`), que permite restringir de qué usuario se puede pasar a qué otros e, incluso, limitar el conjunto de programas que puede usar como ese usuario.

Usuarios de Sistema

Los usuarios de sistema de cada UNIX son diferentes, pero la gran mayoría de ellos disponen de los siguientes usuarios:

- `root`: Administrador (UID 0).
- `daemon`: Ejecuta procesos de servicio del sistema (UID 1).
- `bin`: Propietario de ejecutables (UID 2).
- `sys`: Ficheros de sistema (UID 3).
- `adm`: Ciertos log (UID 4).
- `nobody`: Usuario sin privilegios.
- ...

Arranque del Sistema

1. Arranque del kernel:
   - El cargador (e.g. LILO o BIOS del sistema) inicia la carga del núcleo desde un bloque de disco.
2. Montar el sistema de ficheros raíz.
   - El núcleo o el cargador conocen cuál es el directorio raíz. También puede ser un parámetro de arranque.
   - El montaje se hace en modo sólo lectura.
3. Arranque del proceso `init` (PID 1).
   - Una vez montado el sistema de fichero se busca el ejecutable `/sbin/init` y se arranca el primer proceso.
4. Montaje del resto de sistemas de ficheros:
   - La tabla de ficheros a montar está en un fichero de configuración.
   - Se vuelve a montar el sistema de ficheros raíz en modo lectura/escritura.
5. Inicialización de los terminales:
   - Se arranca tanto los terminales texto como los gráficos.
6. Activación del *runlevel* (demonios):
   - Se empieza a arrancar los restantes servicios del sistema.
   - El encargado de saltar al *runlevel* indicado en la configuración es el proceso `init`.

Runlevels Estándar

- Los *runlevels* son "niveles de ejecución", es decir, diferentes configuraciones del sistema con diferentes servicios.
- La parada y re-arranque con también *runlevels*.
- El *runlevel* es un parámetro de arranque del sistema.

- Los *runlevel* más comunes son (depende mucho de la variante UNIX):
  - ***Runlevel* 0**: Parada del sistema.
  - ***Runlevel* 1**: Modo mantenimiento.
  - ***Runlevel* 2**: Multiusuario sin red (NFS).
  - ***Runlevel* 3**: Multiusuario.
  - ***Runlevel* 4**: <Reservado>
  - ***Runlevel* 5**: Terminal gráfico.
  - ***Runlevel* 6**: Re-arranque del sistema.

Fichero `/etc/inittab`

Este fichero configura al proceso `init`.

Indica cosas como:
- El *runlevel* por defecto.
- La inicialización de terminales.
- Secuencia o condiciones de parada (e.g. *power-fail*)

Cada entrada tiene el formato:

```
id:runlevels:action:process args
```

- `id`: Identificador único.
- `runlevels`: Niveles en los que se ejecuta.
- `action`: Modo de ejecución.
- `process args`: Proceso a ejecutar.

Tipos de Acciones Estándar

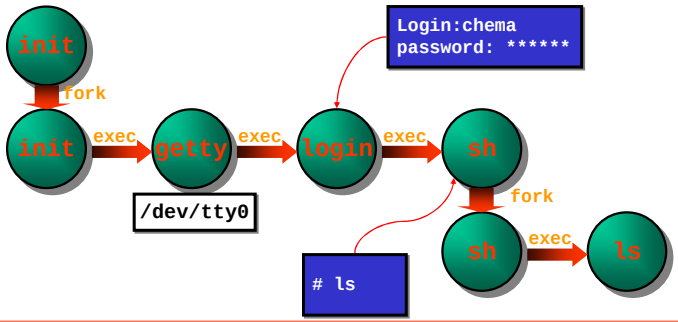
El campo `action` representa cómo se ejecuta esa línea por parte del proceso `init`.

Hay que tener en cuenta que la gran mayoría de las entradas del fichero `inittab` implican el arranque de un determinado proceso.

Las acciones habituales son:
- `wait`: Arranca el proceso y espera a su finalización antes de seguir.
- `respawn`: Arranca el proceso automáticamente en el caso de que muera.
- `once`: Si no esta arrancado (sin esperar).
- `boot`: Ejecuta sólo en el arranque (sin esperar).
- `off`: Si el proceso está en ejecución, lo mata.

Inicialización de Terminales

La inicialización de terminales las hace el `init` por medio de una serie de gestores de terminal (e.g. `getty`).
Después se arranca un proceso `login` que es el que se queda a la espera de que el usuario teclee.



Modificación del *Runlevel* en Ejecución

Se puede reiniciar el proceso `init` (por ejemplo, al cambiar la configuración del fichero) mandando una señal HUP.

El nivel de ejecución actual se puede cambiar invocando a `init` con el nuevo *runlevel* como argumento.

Por ejemplo:

`$ init 0`	Apagaría la máquina.
`$ init 6`	Re-arrancaría la máquina.
`$ init 1`	Entraría en modo recuperación.

Scripts del Sistema

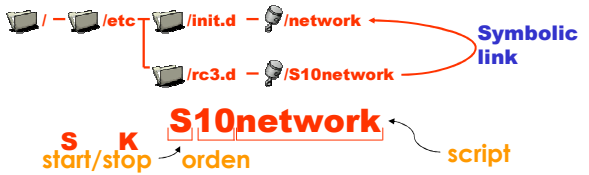
El directorio `/etc/init.d/` es el usado para mantener los *scripts* de arranque de los servicios del sistema.

- Son *scripts* (`/bin/sh`), no binarios.
- Reciben diferentes argumentos (`start`, `stop`, `status`, `restart`, ...).
- Si se quiere crear un nuevo servicio, se programa un *script* en este directorio.
- El código de cada uno de estos *scripts* hace tareas del tipo:
  - Verificar las condiciones para arrancar el servicio.
  - Leer otros archivos de configuración.
  - Quitar o poner ficheros *lock* (para no arrancar 2 veces un servicio).
  - Arrancar y detener el servicio en el orden apropiado. Para ello ejecutará los programas correspondientes a dicho servicio.

Scripts del *Runlevel*

Para asociar los servicios a cada *runlevel*, se hace un **enlace simbólico** desde el directorio del *runlevel* al *script* apropiado de arranque.
El nombre del enlace simbólico es una convención para que el proceso `init` sepa en qué orden hay que arrancarlo.

- Directorio de scripts del sistema: `/etc/init.d/`
- Directorio de cada *runlevel*: `/etc/rc3.d/`



Parada del Sistema

El proceso de parada consiste en:

- Notificación a los usuarios.
  - Por lo general con un mensaje a todos los terminales.
- Envía una señal a los procesos para su terminación.
  - Se envía una señal SIG_TERM, que puede ser capturada por los procesos para "morir" de forma controlada.
- Entrada en modo monousuario:
  - Mata el resto de procesos (esta vez con SIG_KILL).
  - Y desconecta a los usuarios.
- Sincronización de los sistemas de ficheros (`sync`).
  - Copia los datos aún en memoria a disco.
  - Y desmonta los sistema de ficheros.

Runlevel 0

Por lo general se usa un *script* (`shutdown`), que es el encargado de enviar el mensaje y luego saltar al *runlevel* 0.

Shells del Sistema

Los shell (o intérpretes de mandatos) son los procesos con los que interactúa el usuario para solicitar la ejecución de procesos al sistema:

Existen diversas familias de *shells*:

- Bourne shell (`sh`/`bash`).
- Korn shell (`ksh`).
- C shell (`csh`).
- TC shell(`tcsh`).

Con otras funcionalidades:

- Restricted shell (`r-sh`):
  - Permisos de ejecución limitados.
- Secure shell (`ssh`):
  - Acceso remoto seguro.
  - En el servidor se arranca después otro *shell* en modo interactivo.

Shells gráficos:

- *File manager* (depende del entrono gráfico y del escritorio).

Otros Servicios

Ciertos servicios se filtran basándose en el *shell* del usuario que lo invoca:

- Un ejemplo típico es el servicio FTP:
  - Este servicio no ejecuta el shell del usuario en el servidor.
  - En su lugar, ejecuta una instancia del servidor FTP (e.g. in.ftpd).
  - No tiene mucho sentido que un usuario creado para un propósito específico, por ejemplo ejecutar un programa o apagar la máquina, use otros servicios.

El fichero `/etc/shells` indica qué ejecutables son *shells* válidos para el resto de servicios.

- [Consejo]: Si un FTP falla, mira si el *shell* está en este fichero.

Shells de Programación

Otra utilidad de los *shell* es la programación (de *scripts*). Además de los *shells* estándar, se dispone de:

- Perl:
  - Tratamiento de expresiones regulares.
  - Interfaz con C y Librerías de utilidades.
- Python:
  - Muy potente y "*más limpio*" que el anterior.
- AWK:
  - Procesador de campos.
- Tcl / Tk:
  - Componentes gráficos (ventanas).

Es **muy importante** para un administrador programar *scripts*.

Gestión de Dispositivos de Almacenamiento

Las fases de uso de un soporte de almacenamiento son:
- Dar formato al soporte:
  - Separación física entre sectores, pistas, etc.
  - Operación de muy bajo nivel.
  - Casi nunca necesario (en discos viene de fábrica).
- Particionamiento:
  - División del disco en zonas asignables a diferentes sistemas de ficheros.
  - Operación de alto nivel de reparto del disco.
  - Sólo para discos o similares (no *floppies*).
- Creación del sistema de ficheros:
  - Creación de las estructuras lógicas de un formato específico de SF.
  - Realizado sobre particiones (discos) o sobre dispositivos enteros (*floppies*).
- Utilización del soporte.

Tipos de Dispositivos

UNIX define dos tipos de dispositivos:
- Dispositivos de tipo bloque (discos).
- Dispositivos de tipo carácter (cintas).

En ciertos UNIX el mismo dispositivo físico puede ser gestionado en modo bloque y modo carácter.
- Operaciones en modo bloque: Montaje y uso.
- Operaciones en modo carácter: Ciertas operaciones de recuperación.

Particionamiento de Discos

El formato de las particiones y características, depende del SO y del tipo de disco (IDE, SCSI).

Las herramientas también depende del SO:
- Linux: **`fdisk`**, **`diskdruid`**.
- Solaris/SunOS: **`format`**.
- AIX: **`smit`**.

La información relativa a las particiones se guarda en una Tabla de Particiones al comienzo del disco.
- Esta tabla indica dónde comienza cada partición y su tamaño.
- Información adicional puede incluir el tipo de SF o el punto de montaje.

Creación de un Sistema de Ficheros

Esta operación crea sobre un dispositivo físico una serie de estructuras que permiten organizar directorios y ficheros.

También depende del SO y del sistema de ficheros que soporte.
Opciones:
- Espacio reservado al root.
- Número de i-nodos.
- Opciones de verificación.

Linux: **`mkfs`**
Solaris: **`newfs`**

[Aviso]: Esta operación es a la que SSOO de tipo MSDOS/Windows llaman “dar formato”.

Modelos de Sistemas de Ficheros

Existen diferentes modelos de sistemas de ficheros:
- Sistemas de ficheros tradicionales: `ext2fs`, `ufs`, `minix`, ...
  - La información a almacenar son datos y metadatos.
  - Dispone de estructuras para gestionar el espacio libre eficientemente.
- Sistemas de ficheros transaccionales: `ext3fs`, `jfs`, `afs`, ...
  - Extienden los SF anteriores añadiendo un log de operaciones.
  - Ventaja: El tiempo de recuperación tras un error es menor.

Verificación de integridad:
- Comprueba la coherencia entre datos y metadatos.
- Posibles problemas:
  - Bloques marcados como libre y ocupado a la vez.
  - Bloques de datos referenciados por múltiples ficheros.
- Herramientas: `fsck`.

Estrategias

Es recomendable la creación de los siguientes SF independientes:
- `/` (Sistema, tamaño justo).
- `/usr` (Aplicaciones, tamaño justo).
- `/home` (Cuentas, mucho espacio).
- `/usr/local` - `/opt` (Gran tamaño).
  - Depende de la las aplicaciones (actuales/futuras).
- `/var` (Logs, bastante espacio).
  - Especialmente si se habilita el servicio de correo.
- `swap` - `/tmp` (Depende de la carga).

Estrategias

Se recomienda:
- Mantener las cuentas en un disco diferente al del sistema.
  - En caso de necesidad, se pueden sacar esos discos y montar en otra máquina.
- Separar los SF de mayor acceso en diferentes discos (*swap* y sistema).
  - Repartes mejor el trabajo entre discos. Muy importante si SCSI.
- Ubicar las particiones de forma que sea posible redistribuir los discos.
  - Colocar las particiones prescindibles (`swap`, `/tmp`,...) entre las que pueden requerir un crecimiento (`/home`).
- Vigilar el porcentaje de disco libre.
  - Un "*file system full*" puede ser muy peligroso dependiendo del caso.

Automatización del Montaje de Sistemas de Ficheros

La operación de montaje implica mostrar el sistema de ficheros residente en una partición como los subdirectorios por debajo de un punto de montaje:
- Tras montar la partición del SF raíz el sistema monta el resto de sistemas de ficheros.
- Dependiendo del SO existen determinados ficheros de configuración que contienen las tablas de montaje:
  - Linux: `/etc/fstab`
  - Solaris: `/etc/ufstab`
  - AIX: `/etc/filesystems`
  - ...
- Estos ficheros indican: dispositivo o partición, punto de montaje y opciones.

Dispositivos sin Sistema de Ficheros

Ciertas utilidades pueden usarse para acceder a dispositivos sin hacer uso del sistema de ficheros:

- **`dd`**, **`cpio`** o **`tar`**.

Ejemplo: **`dd if=data.img of=/dev/fd0`**
Copia "bloque a bloque" el fichero **`data.img`** sobre la unidad *floppy*.

Permite el acceso a dispositivos a muy bajo nivel:

- Duplicar una partición.
- Recuperar datos borrados o perdidos.
- Pueden usarse como utilidades de *backup* elementales.

Sistemas de *Backup*

Backup: Copia de seguridad de determinados datos de un sistema.

Esquemas de *backup*:

- Backups completos: Se copia toda la información.
- Backups incrementales: Sólo los ficheros modificados son copiados.

Habitualmente se combinan los dos esquemas.

Decisiones de *Backup*

Una estrategia de *backup* debe incluir:

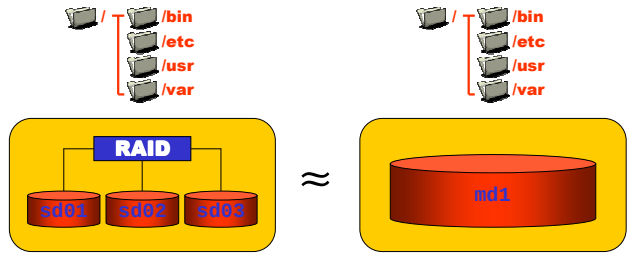
- Estimación del volumen de datos:
  - Tamaño original de los datos a copiar.
  - Estimación del ratio de compresión.
- Selección de los ciclos de *backup:*
  - Cuándo se realizan y de qué tipo son (completo/incremental).
- Automatización (cliente/servidor):
  - Programación de los *backups*.
- Verificación del sistema.

Herramientas de *backup*: **`Amanda, KDat, Tivoli….`**

Dispositivos Redundantes

Dispositivos RAID:

- Dispositivo virtual compuesto por varios dispositivos físicos reales agrupados.
- Proporciona redundancia y mejores prestaciones.
- A nivel del SF que hay por encima es trasparente.

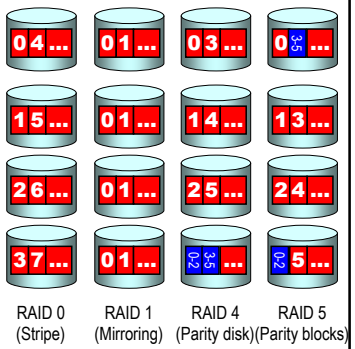


Tecnología RAID

Hay varios modelos de RAID:
- Modo lineal: Concatena volúmenes.
- RAID 0: Modo alternado de bloques.
- RAID 1: Redundancia (*Mirroring*).
- RAID 4: Disco de paridad.
- RAID 5: Bloques de paridad.

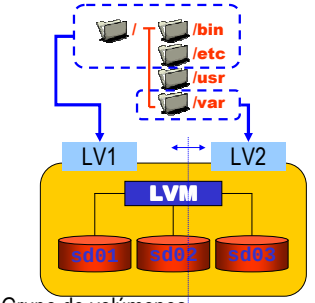
Discos de reserva: *spare disks:*
- Discos no asignados que se incorporar sustituyendo un disco que falla.

Configuración software:
`/etc/raidtab`



Volume Manager

- Gestión del espacio de almacenamiento de forma dinámica:
  - Permite redimensionar particiones creadas (Volúmenes lógicos)
  - Utilizar varios dispositivos reales como soporte (Volúmenes físicos)



Grupo de volúmenes

- Soportado por:
  - LVM (AIX, Linux)
  - Veritas VM (HP-UX)
  - Sun Volume Manager
- Requiere que el sistema de ficheros también se pueda redimensionar (JFS/ReiserFS)
- Su comportamiento es similar a RAID0 pero con varias particiones dinámicas encima.

Cuotas de Disco

Asocia a cada usuario/grupo un límite de espacio en disco.

Las cuotas limitan:
- El número máximo de archivos (i-nodos).
- El número máximo de bloques. La suma total del tamaño de todos los archivos.

Dos límites:
- Soft limit: Límite informativo.
- Hard limit: Espacio máximo disponible.

Límites aplicables a cada sistema de ficheros. Se verifican en el arranque de la máquina y en cada login.

Configuración de TCP/IP

Configuración del interfaz:
- Dispositivo de red (e.g. `/dev/le0`).
- Asignar dirección IP.
- Máscara de red.
- Dirección Broadcast.
- Subred.

Se utiliza una herramienta (`ifconfig`: *Interface configurator*).

Por lo general los parámetros se guardan en ficheros de configuración y un *script* del sistema los lee e invoca a `ifconfig`.

Ejemplo:
```
ifconfig eth0 138.100.9.101 netmask 255.255.248.0 up
```

Configuración de TCP/IP

Encaminamiento: indica al sistema cómo hacer llegar un paquete IP a una dirección destino. Para ello se indica cuál será el siguiente salto.

La herramienta **`route`** permite configurar las tablas de encaminamiento.

Encaminamiento IP:

- Encaminamiento local: "me lo quedo yo".
- Encaminamiento dentro de la subred: "esta por aquí cerca".
- Encaminamiento externo (Router): "pues ni idea, para afuera".

Ejemplos:

```
route add -host 127.0.0.1 lo
route add -net 138.100.8.0 netmask 255.255.248.0 eth0
route add default gw 192.168.1.1 eth0
route add default ppp0
```


Configuración de TCP/IP

Una alternativa para la configuración de red es la configuración dinámica.

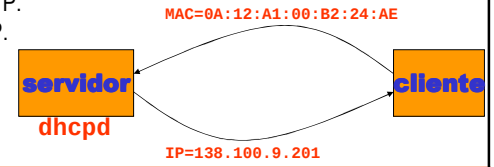
Por medio de protocolos específicos se intercambia con un servidor (conocido):

- Del cliente al servidor: La dirección física de la tarjeta de red.
- Del servidor al cliente: Los parámetros de configuración.

Usado especialmente en redes muy grandes (por comodidad).

Configuración dinámica, protocolos:

- Protocolo BOOTP.
- Protocolo DHCP.



Configuración de un *Router*

Un router hace de encaminador entre dos o más redes.

Servicio de encaminamiento:

- Máquina con dos interfaces de red.
- Intercambio de tablas de encaminamiento entre *routers*:
  - Aunque se pueden configurar de forma estática (**`route`**).
  - Lo lógico es que se intercambie con otros *routers* información sobre las redes próximas, usando ciertos protocolos: RIP/OSPF.
- Emisor de ciertos mensajes ICMP. Son mensajes de control de red.

La configuración dinámica requiere de un servicio del sistema que implemente alguno de estos protocolos: **`routed`** (RIP), **`gated`** (RIP/OSPF)

Resolución de Nombres

El proceso de resolución de nombres implica traducir:

- Un nombre simbólico: *laurel.datsi.fi.upm.es*
- En una dirección IP: *138.100.8.101*

Orden de resolución, indica dónde se busca para resolver un nombre:

- **`/etc/host.conf`**

Una de las opciones es la resolución de nombres local:

- **`/etc/hosts`** contiene una tabla de traducciones estática.

La otra es la resolución de nombres remota:

- **`/etc/resolv.conf`** contiene la dirección IP de un servidor de nombres. Servidor DNS.
- Asimismo, contiene el identificador y dominio de la máquina.
- La dirección del servidor de nombres tiene que ser conocida (no se puede resolver).

Configuración de un Servidor de Nombres

Existen dos tipos de servidores de nombres.

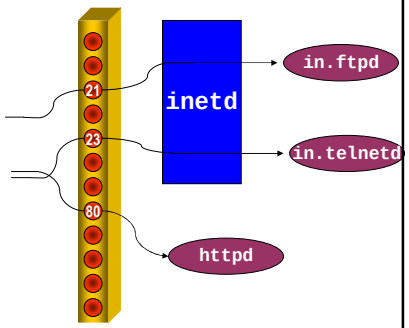
- DNS en modo caché:
  - Resuelve peticiones y almacena los resultados.
  - Siempre tiene un DNS superior.
- DNS autónomo:
  - Mantiene una BD propia.
  - Incluye el anterior.

Al igual que en el caso del encaminamiento, este servicio requiere de un demonio que interprete el protocolo DNS, **named** por ejemplo.

Demonios de Red

Existen dos modalidades de servicios:

- Dependiente del **inetd**.
  - Sólo existe un proceso en ejecución que reserva varios puertos.
  - Este demonio, si recibe un mensaje a un puerto estándar, arranca el servidor correspondiente.
  - Permite tener menos procesos arrancados.
- Autónomo (modo *standalone*).
  - Un programa que arranca y reserva él solo el puerto que corresponda.
  - Típico: **httpd** (serv. web)



Demonio inetd

Este demonio se configura en dos partes:

- Puertos estándar de servicio: **/etc/services**
  - Dado un puerto reconoce qué servicio se está invocando.

```
#nombre    puerto/protocolo   alias
telnet     23/tcp
time       37/udp             timeserver
```

- Programas de servicio: **/etc/inetd.conf**
  - Para cada servicio se conoce qué servidor hay que arrancar.

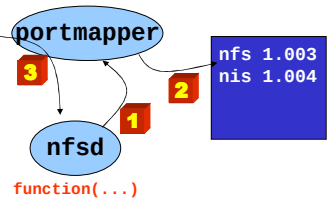
```
#servicio  socket   proto   flags   usr   serv
telnet     stream   tcp     nowait  root  in.telnetd
time       dgram    udp     wait    root  internal
```

Variantes más avanzadas como **xinetd** tiene algunas opciones adicionales y el formato es ligeramente diferente.

Servicios RPC de Sun

Una tercera alternativa para servicios de red son las RPC.

- Define un formato de comunicación propio.
- Estos servicios disponen de un único servidor estándar (en el puerto 111): **portmapper**
- En este se registran los servicios RPC 1 con un número de programa 2.
- Los clientes consultan al **portmapper** sobre los programas 3



Servicios de red de SUN:

- NFS (Network File System).
- NIS (Network Information System).

Sistemas de Ficheros en Red

Permite exportar sistemas de ficheros a otras máquinas.

- Servidor (exporta un directorio de su árbol de ficheros):
  - Asignación de permisos (lectura/escritura, *root-squash*).
  - Todo esto se indica en: `/etc/exports` o `share`.
- Cliente:
  - Realiza el montaje pero, en lugar de indicar un dispositivo, indica el servidor, dos puntos y el directorio (`laurel:/usr/local`).
  - Montaje manual:
    ```
    mount -t nfs laurel:/usr/local /opt
    ```
  - Montaje automático (en `/etc/fstab`, por ejemplo):
    ```
    /opt laurel:/usr/local nfs (rw,no-root-squash)
    ```


Usuarios de Dominio (NIS)

Un servicio común en los sistemas en red es el servicio NIS / NIS+:

- Mantiene bases de datos compartidas por varias máquinas.
  - Usuarios y *passwords* (`/etc/passwd` y `/etc/shadow`).
  - Grupos (`/etc/groups`).
  - Otras opciones de configuración.
- Permite tener centralizada gran parte de la configuración (más mantenible para grandes redes).
- Organización Cliente/Servidor.

NIS+ añade:

- Mayor nivel de seguridad (certificados y cifrado).
- Organización jerárquica y replicada de servidores.
- Una configuración más compleja ☹

Usuarios de Dominio (NIS)

La configuración de NIS, a grandes rasgos, consiste en:

- Maestro o Servidor NIS:
  - Define un dominio (`domainname`).
  - Iniciar las bases de datos de NIS.
  - Publicar las bases de datos del sistema.
- Cliente NIS:
  - Definir el mismo dominio.
  - Modificar el mecanismo de búsqueda de datos en las BD locales/remotas.

Terminales Remotos

Otro servicio importante de red son los servicios de acceso remoto.

Existen tres servicios de terminal remoto:

- `rsh` (*remote shell*).
- `rlogin` (*remote login*).
- `telnet` (*terminal remoto*).

En los casos de `rsh` y `rlogin` la configuración puede permitir acceso basado en *host*:

- Se confía en la identificación hecha por el *host* desde el cuál se conecta. Por ejemplo, el usuario *chema* de *laurel* puede acceder.
- No se transmite *password* alguna.
- Fichero `/etc/hosts.equiv` y `~/.rhosts`

Otros Servicios

- Servidor Web:
  - NCSA, Apache.
- Servidor FTP:
  - WU-ftp, ProFTPd.
- Agente de correo:
  - Sendmail, Postfix, Exim.
- Servicios de correo (IMAP, POP):
  - Courier, Cyrus.
- Servicios de noticias:
  - NNTP.
- Protocolos *peer-to-peer* (compartición de ficheros):
  - GNUtella, eMule, ...

Opciones del Protocolo IP

El protocolo IP ofrece:

- IP Masquerading:
  - Usar una sola dirección IP para varias máquinas.
  - Vale para que varios equipos de una red compartan una única IP de salida.
- IP Accounting:
  - Estadísticas y análisis de paquetes.
- IP Aliasing:
  - Varias direcciones IP a las misma tarjeta.
  - Si se quiere discriminar determinados servicios.
- IP Forwarding:
  - Redirección de paquetes.
  - Permite hacer *firewalls* y *routing*.

Servicios Internos

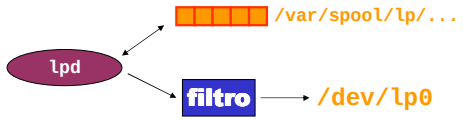
Otros servicios adicionales UNIX:

- Servicio de Impresión (lp).
- Servicio de Programación de tareas:
  - Tareas pendientes (puntuales).
  - Tareas periódicas.

Servicio de Impresión

Demonio de Impresión (lpd):

- Asociado al dispositivo de impresora:
  - Local o remota.
- Gestiona la cola de trabajos:
  - Por lo general por debajo del directorio `/var`
- Preprocesa el documento a imprimir:
  - Filtros.



Filtros de Impresión

Pequeños *scripts* que procesan la entrada (documento) y transmiten su salida a la impresora:

- Filtros texto:
  - Permiten configurar retornos de carro y otras opciones de impresión.
- PostScript:
  - Interpretación *postscript* de la impresora (`gs`).
  - Permite cambiar opciones o incluso pasar a formato nativo de la impresora (si no interpreta PostScript), por ejemplo PCL.

Configuración del Demonio

La configuración de este servicio se hace por medio del fichero:
`/etc/printcap`

Este fichero define:

- Todas las impresoras del sistema.
  - Locales.
  - Remotas (otros UNIX o independientes).
  - Compartidas (otros SSOO).
- Ficheros de configuración y filtros.

Programación de Tareas

- Demonio `atd`:
  - Ejecuta un mandato en un instante determinado.
  - Informa al usuario (vía *mail*).
- Demonio `crond`:
  - Mantiene una serie de tablas de tareas habituales:
    `/etc/crontab`
  - Tareas de administración:
    - `/etc/cron.daily`
    - `/etc/cron.weekly`
    - `/etc/cron.monthly`

Instalación de Nuevo Software

La instalación de nuevo software en el sistema se puede realizar a partir de diferentes tipos de distribución de software:

- Paquetes de instalación.
- Binarios comprimidos.
- Código fuente (para dicho SSOO).
- Código fuente a portar.

En ciertos casos es posible instalar software en las cuentas de usuario, pero, por lo general, se hace por parte del administrador.

Paquetes de Instalación

El formato de paquete depende del SO:
- RedHat y otros linux: RPM.
- Solaris: PKG.
- AIX: SMIT.
- ...

Incluye software y configuración del mismo (un *script*).

Cada SO incluye algún tipo de herramienta de instalación:
- Linux RedHat: `rpm -ivh` *`paquete.rpm`*
- Linux Debian: `dpkg -i` *`paquete.deb`*

En algunos casos, la herramienta de instalación puede descargar de servidores autorizados el software:
- Linux Debian: `apt-get install` *`paquete`*

Código Binario

En otros casos se distribuyen los directorios de binarios, documentación, etc., comprimidos en un fichero.

Posibles problemas:
- Sistemático en la instalación.
- Problemas de versiones de librerías y otros programas...
- Problemas de seguridad.

Es necesario verificar que el software sea para la arquitectura y SO apropiados.

Formatos (comprimidos):
`.tar.gz`, `.tgz`, `.tar.Z`, `.shar` o `.bz`

Código Fuente

Otra alternativa (más portable) es distribuir las fuentes y que cada cual se las compile en su sistema.

Posibles problemas:
- Configuración de las fuentes: `IMake` o `configure` (generan los `Makefiles`).
- Compilador adecuado (C o C++).
- Instalación del programa.

En las aplicaciones tipo GNU esto se suele hacer:

```
$ ./configure
$ make
$ make install
```


Configuración/Actualización del Núcleo

- Un elemento clave del software del sistema es el propio núcleo o *kernel*:
  - La configuración (en el caso de Linux se hace recompilando las fuentes):
    `make {xconfig dep clean bzImage modules modules_install}`
  - En otros sistemas operativos por medio de la inclusión de módulos dinámicos:
    - Por ejemplo, el directorio `/kernel` de Solaris
  - La configuración del núcleo es muy delicada pero puede tener un efecto importante en el rendimiento del sistema.

Terminales Gráficos

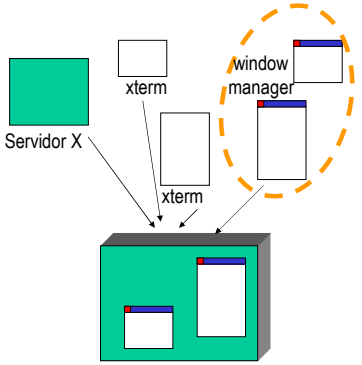
X Window:
- Entorno gráfico de los sistemas UNIX.
- Arquitectura Cliente/Servidor
- Diferentes niveles (librerías) de desarrollo.
- Nuevos problemas de seguridad.

El sistema gráfico de ventanas es muy dependiente del SO. En la actualidad, parece que tiende a converger a un conjunto de librerías más o menos portables para la gran mayoría de sistemas.

X Window

Servidor:
- Asociado al terminal gráfico.
- Muestra en pantalla los pixels.

Cliente:
- Aplicación con salida gráfica.
- Está asociada a un servidor.



X Window

Servidor:
- Interactúa con el hardware gráfico.
- Acepta mensajes X11.

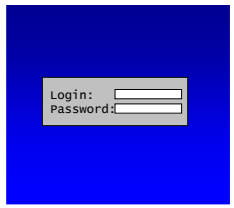
Clientes:
- Usan primitivas para dibujar en el servidor.
- Pueden ser aplicaciones remotas.
- Gestor de ventanas: *window manager*
  - El gestor de ventanas es un cliente más.
  - Se encarga de controlar el comportamiento de las aplicaciones ante determinados eventos.
  - Controla acciones como minimizar, maximizar, etc.
  - Define, por ejemplo, el marco de las ventanas de cada aplicación.

XDM

El servicio XDM (X desktop manager):
- Proporciona un login gráfico.
- Asociado al *runlevel* 5 (gráfico).
- Se re-arranca si el servicio cae.
- Puede ser accedido desde puestos remotos.

Para la conexión remota a un sistema se puede invocar al XDM desde el servidor X:

X –query *servidor*



Interoperabilidad con MS Windows

El protocolo SMB (Server Message Block) lo utilizan los sistemas MSWin para compartir discos e impresoras.

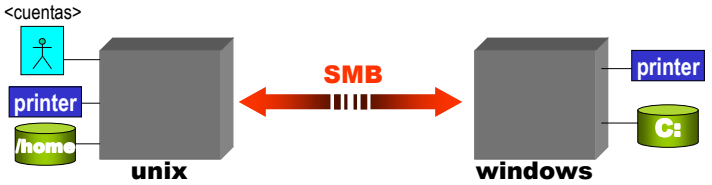
La implementación de este protocolo en UNIX se denomina Samba.

Por medio de Samba es posible acceder a unidades y recursos compartidos de un entorno Windows.

Cliente/Servidor Samba

- Gestión de usuarios (cuentas).
- Impresoras compartidas.
- Directorios compartidos.

- Impresoras compartidas.
- Directorios compartidos.



Servidor Samba

La configuración de un servidor Samba arranca dos demonios:
- **smbd**: Demonio de SMB.
- **nmbd** : Servicio de nombres NetBIOS.

Configuración: `/etc/smb.conf`

Cliente Samba

Existen las siguientes herramientas:
- **smbclient**: Conexión a recursos (modo FTP).
- **smbrun**: Ejecución de programas.
- **smbprint**: Impresión remota.
- **smbmnt**: Permite montar discos compartidos.

También se puede montar automáticamente incluyendo una línea en el fichero `/etc/fstab`, indicando como tipo de SF **smbfs** o **cifs**.

Auditoría del Sistema

Los directorios `/var/log` o `/var/adm` contienen información sobre ciertas operaciones registradas en el sistema:

- Accesos de usuarios.
- Mensajes del kernel.
- Arranque y parada del sistema.
- Errores de ciertos demonios.

Conexiones al Sistema

Ficheros de acceso:

- Registra las conexiones.
- En algunos casos, tiene formato binario.
- Se consulta por medio del comando **`last`**.
  - Los datos retornados por este mandato están almacenados originalmente en formato binario (no en texto).
- Las conexiones activas (de todo tipo de protocolos) se pueden obtener por medio de un mandato **`netstat -ta`**
  - Muestra los puertos abiertos y conexiones activas.

Gestión de Logs

La gestión de los ficheros log comprende:

- Selección de eventos a registrar:
  - Accesos, errores, etc.
  - Determinar de qué servicios y qué tipo de acciones.
- Los ciclos de rotación:
  - Cada cuánto tiempo los logs se reinician.
- La compresión de los logs:
  - Los logs antiguos (después de la rotación), se renombran y se comprimen.
- Verificaciones de integridad:
  - Programar algún *script* que recorra los logs a diario e informe de cosas "extrañas".

Carga del Sistema

La utilidad **`uptime`** proporciona una estimación de la carga del sistema:

Otras utilidades:
`ps`, **`top`**, **`xload`**, **`perfmeter`**...

```
# uptime
3:24pm up 6 days, 2:30, 5 users, load average: 0.23, 0.32, 0.26
```

Memoria del Sistema

Utilidades como **vmstat** o **sar** miden el uso de la memoria del sistema:

- Acceso a swap.
- Estado de los procesos.
- Memoria usada del sistema.
- Porcentaje de CPU usada.

Límites de Recursos

- A un usuario se le restringen los límites de recursos a utilizar por medio de la llamada **ulimit**.
  - Por lo general, los límites generales se definen en el fichero **/etc/profile** que todos los usuarios ejecutan.
  - Se pueden establecer límites por:
    - Número de procesos.
    - Cantidad de memoria.
    - Número de recursos (ficheros abiertos, semáforos, etc..)
    - ...

Seguridad

Puntos de interés:

- Seguridad Interior:
  - ¿Qué cosas pueden y deben hacer mis usuarios?
  - Programas con permisos.
- Seguridad Exterior:
  - ¿Hacia el exterior cuáles son los servicios que se ofrecen?
  - Servicios de red.
- Detección de Intrusiones:
  - Una vez que han superado la seguridad.
  - ¿Qué es lo que han hecho?

La mejor defensa es tener una máquina con instalación **no** estándar.

Seguridad Interior

Programas con permisos de ejecución privilegiada: Bit **s**.

- Este bit otorga temporalmente la identidad del propietario del fichero a quien lo ejecute. (Para delegar privilegios)
- Si el programa no se usa: eliminarlo.
- Si se usa: instalar la versión más actualizada.
- Restringir los privilegios de ciertos usuarios (*restricted shells*).

Muchos de los exploits (ataques sobre vulnerabilidades del sistema) se realizan sobre programas con estos permisos.

Si consiguen hacerlos fallar se puede forzar a ejecutar otros programas (shells, por lo general) como ese usuario.

Seguridad Exterior

Servicios de red del sistema:
- Si no se usa: eliminarlo.
- Si se usa: tenerlo actualizado.
- Saber quién debe usar cada servicio (desde dónde se usa).

A este nivel también se dan ataques similares a los anteriores.

Servicios mal configurados o antiguos pueden ser vulnerables haciendo que el intruso acceda al sistema.

Una vez dentro ya es más fácil.

Filtrado de Conexiones

Los *TCP wrappers* son un paquete de seguridad basado en filtrar conexiones al inetd.

Configuración:
- `/etc/hosts.allow`
- `/etc/hosts.deny`

```
# /etc/hosts.deny
ALL: PARANOID   # Direciones sospechosas
ALL: ALL        # Todos los servicios
```

```
# /etc/hosts.allow
telnetd, ftpd: LOCAL, .fi.upm.es
fingerd: ALL: (finger @%h |
               mail -s "finger @%h" root)
```


Firewalls

La mejor opción de seguridad externa son los firewalls:
- Proporcionan las mismas funcionalidades de los TCPwrappers pero no a nivel del inetd.
- El control de accesos se realiza dentro de la pila de protocolos TCP/IP.

En Linux, por ejemplo:
- La configuración de los firewalls se realiza por medio de una extensión de las opciones de IP.
- Por medio de unas herramientas de usuario: `iptables`
- Estas herramientas configuran unos filtros que se aplican dentro de la pila de protocolos.

Sistemas de Detección de Intrusos

- Sistemas que buscan patrones de comportamiento malicioso:
  - En los ficheros log del sistema.
  - En el tráfico de red (inyección de paquetes, DoS, análisis de puertos).
- Existen tres niveles de IDS (*Intrusion Detection Systems*):
  - IDS de Host: Protege una máquina (análisis de logs).
  - IDS de Red: Una tarjeta en modo promiscuo analiza el tráfico de un segmento de red.
  - IDS de pila de protocolos: Analizan no sólo el tipo de paquetes sino también el contenido y opciones de los protocolos.
- Ejemplos: `Tripwire`, `Abacus Sentry`, `Snort`

Salvaguarda de los Log

Una intrusión en la máquina intenta borrar sus "*huellas*":
- Análisis de los ficheros de log:
  - Automatizado a ser posible.
- Protecciones especiales (*append*):
  - Si los sistemas de ficheros lo permiten.
- Salvaguarda periódica de los log.
- Renombrar ficheros de log.

Integridad del Sistema

La verificación periódica del sistema:
- Automatizada (`crontab`).
- Comparar los directorios del sistema.
  - Extraer el listado de los directorios y periódicamente cotejar los datos.
- Checksums o CRCs de los ficheros:
  - Ídem con los ficheros, pero en lugar del contenido se hace un checksum (`md5sum`, por ejemplo) y se compara.
- Fecha/Hora de arranque de los servicios:
  - Si alguien ha reconfigurado un servicio y lo ha re-arrancado, el PID y la hora de arranque serán muy distintas a las del sistema.
- Puertos de servicio del sistema:
  - Por medio de `netstat` se saca la lista de puertos activos y conexiones del sistema.

Passwords por la Red

La transmisión de passwords por la red puede ser interceptada por medio de *sniffers*:
- Uso de autorización basada en hosts (`~/.rhosts`, `/etc/host.equiv`)
- Transmisiones cifradas (*secure shell*: `ssh`).
- Topologías de red aisladas.

Un sniffer es una aplicación que configura el interfaz de red (la tarjeta) en modo "promiscuo" de forma que captura todo el tráfico del mismo segmento de red.

IP Spoofing

Otro posible ataque es por medio de suplantar la identidad de otro host:
- Estos tipos de ataques se denominan *man-in-the-middle*.
- No sólo implican la escucha de mensajes (sniffing).
- Además la captura de algunas y la reemisión de otros.
- Se pueden detectar debido a rutas de mensaje extrañas.
- Servicios de autenticación:
  - Garantiza la identidad de los interlocutores.
  - Certificados, Kerberos, ...

Servicios de Red Peligrosos

Servicios de red más problemáticos:

- Servicio de correo (sendmail):
  - Complejo, difícil de configurar.
- Servicio FTP (ftp anónimo):
  - Si está mal configurado puede permitir que se use como repositorio de información para otros ataques.
- Servicio Web (CGIs):
  - Ejecución de código por medio de un formulario web.
  - Especialmente peligroso si cualquier usuario de la máquina puede ofrecer esta opción.
- RPCs (ataques al `portmapper`):
  - Si está mal configurado las RPCs buscan el servidor en modo *broadcast*. Susceptible de *spoofing*.
- Protocolos de red: ICMP, SNMP, ARP...
  - Problemas "históricos" con estos protocolos.

Bibliografía

- Generales
  - *Unix System Administration Handbook*. Evi Nemeth, Garth Snyder,. Prentice Hall 4th Edition 2006
  - *Essential System Administration - Help for UNIX System Administration.* Æleen Frisch. O'Reilly - 2nd Edition 1995
  - *The Practice of System and Network Administration*. Thomas A. Limoncelli, Christine Hogan. Addison Wesley 2001
- Linux
  - Linux System Administrator's Guide (http://tldp.org/LDP/sag)
  - Security & Optimizing Linux (http://tldp.org/LDP/solrhe)
  - Linux Network Administrator's Guide (http://tldp.org/LDP/nag2)