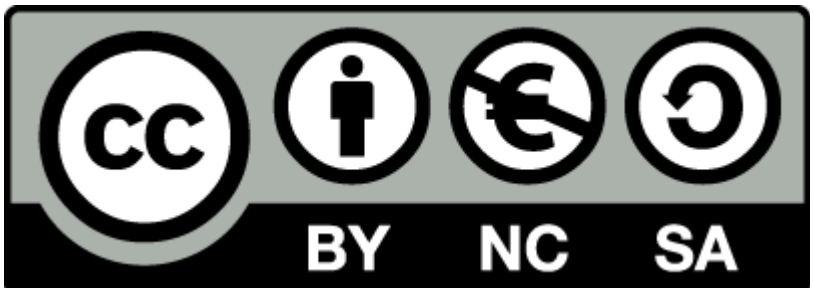
Apuntes para la certificación LPI nivel 2

Jorge Andrada Prieto

Licencia



Contacto

monino@gmail.com ***@monino*** ***jandradap***

Bibliografía

Aclarar que cubre los *objetivos LPIC-2 para la versión 3.5.0*
Para la realización de los apuntes me baso en páginas man, wiki lpi, Guía de estudio - Exámenes 201 y 202, Anaya Multimedia ISBN-13: 978-84-415-3014-0 (recomendable su compra) y por supuesto san google ;)

Agradecimientos

- **Grison** y **Rano**: por su apoyo y su gran amistad
- **Paquintosita**, **Carlos López** y **P.Quintas**: por esas risas y cafés en la biblioteca
- **Lolo**, **Bala**, **Jose**, **Skuleto**, **Arboleda** y demás colegas: por hacerme desconectar los fines de semana y pasarlo de arte
- **Carmen Pozo** por esos momentos de relax en sitios tan ricos y aguantarme
- **Colegas del curro** como "**Er Femenino**" por darme caña cuando estoy de descanso-estudio, **Rubén "Muñón"** por ser un compi fricazo de lo mejor que hay! **Mon** por las pechá de reir y **Jpagador**, Juanillo, **Negra**, Nefer, Pepe, **Adolfo**, etc.
- Colegas de **diablo3-esp.com**: por ayudarme a mendigar al no tener tiempo
- **Mi pelirroja**: por hacerme más amena las horas de estudio con su bonita sonrisa...ains, gracias desconocida...de momento
- Mis **padres** y mi **vaca asquerosa** (mi hermana) por muchas cosas.

Tema 201: Kernel de Linux

201.1 Componentes del Kernel

Se puede descargar el Kernel "**Linux Kernel Archives**" (www.kernel.org) o mediante actualización del sistema (yum, apt, etc).
Es complicado distinguir entre Kernel estable e inestable.

Formato: "**versión.subversión.nuevas funciones importantes.arreglos de esa versión**".

Podemos ver la versión que se está ejecutando actualmente con "**uname -a**".

***Nota**: actualizar el Kernel sin reiniciar (para sistemas que no pueden tener inactividad) se usa la herramienta "**ksplice**", que congela la ejecución de programas que gestiona el kernel, cambia el nuevo Kernel y retoma la operación de los programas. Es totalmente transparente.*

El código fuente del Kernel reside en "**/usr/src/linux-version_que_sea**", otras distros lo ubican en "**/usr/src/kernel/linux-version**".
Un vínculo simbólico de "**/usr/src/linux**" debe apuntar al directorio actual del Kernel fuente (modificamos con "**ls -ld /usr/src/linux**"). Con varios Kernels hay que hacerlo a mano "**rm /usr/src/linux"** y "**ln -s /usr/src/linux-version /usr/src/linux**".

Se puede instalar con "**yum** y **apt**", buscando "**linux-source** o **kernel-devel**", también podemos instalar sólo las cabeceras con "**linux-headers** o **kernel-headers**".

Documentación:
Se encuentra en "**/usr/src/linux/README**". La documentación específica y subsistemas del Kernel se encuentra en "**/usr/src/linux/Documentation**", pero es enorme, hay un índice "**00-INDEX**".

Binarios del Kernel:
Una vez compilado, el Kernel produce dos archivos, dos categorías:

- **Archivo principal del kernel**: contiene las partes centrales del Kernel, es procesado por el bootloader. Reside en "/".
- **Módulos del Kernel**: residen "**/lib/modules/**", con subdirectorios para versiones.

Tabla de archivos principales del Kernel:

- **vmlinux**: versión descomprimida del kernel, creado como paso intermedio. No arrancable.
- **vmlinuz**: vmlinux comprimida, arrancable con algunas funciones. Suelen ser nombres de Kernels binarios precompilados.
- **zImage**: obsoleto, tamaño limitado a 512 KB.
- **bzImage**: sustituye al obsoleto zImage. Nombra Kernels compilados localmente.
- **Kernel**: binario estilo bzImage, usado por bootloaders como Grub2.

201.2 Compilar un Kernel

Configurar las fuentes del Kernel: hay miles de opciones de configuración, para configurarlo bien hay que:

Investigar su hardware: tener un módulo innecesario añade tiempo de carga del SO. Hay que leer los manuales del hardware y usar: lspci, lsusb y lsmod.

Utilizar objetivos make del Kernel:
Algunos son:

- **mrproper**: elimina archivos de configuración y temporales antiguos.
- **oldconfig**: actualiza un archivo de configuración antiguo con elementos nuevos.
- **virtualdconfig**: oldconfig pero más ordenado por pantalla.
- **defconfig**: crea un archivo de config nuevo con los valores por defecto.
- **allmodconfig**: crea un archivo de configuración que emplea toda la configuración modular posible.
- **config**: configura todos los elementos del Kernel usando un interfaz basada en texto.
- **menuconfig**: configura el Kernel usando un sistema de menú basado en texto.
- **xconfig**: como menuconfig, pero con GUI basada en Qt.
- **gconfig**: xconfig, pero basado en Gtk.

Opciones de configuración:
Con menuconfig, xconfig o gconfig. Alguna configuración importante:

- **Configuración general>Versión local**: para montar el mismo Kernel con diferentes opciones.
- **Configuración general> Soporte Ram inicia y disco inicial**: (intran???/initrd): si se usa un disco RAM inicial.
- **Configuración general> Habilitar soporte de módulo cargable**: no podría cargar drivers que no vengan del Kernel. Muchos inconvenientes.
- **Tipo y funciones el procesador> Soporte de multiproceso**: habilita más de una CPU o núcleo por CPU.
- **Tipo y funciones del procesador> Familia del procesador**: se logra rendimiento especificando una CPU concreta.

Compilar un Kernel:
Hacemos "**make**" en el directorio fuente del kernel. Podemos compilar el Kernel por separado con "**make nzImage o zImage**" y los módulos con "**make modules**". Para localizar errores: "**Make zImage | grep -iw "error"**".

Preparar un disco de RAM inicial:
Es una colección de módulos fundamentales del Kernel y grupo de utilidades del sistema que el bootloader pasa al Kernel en el arranque. El Kernel accede a ellos en la memoria como si fuese un disco, carga módulos y ejecuta scripts para montar "/". Es necesario cuando usamos: RAID, LVM o NAS (SAN en español), pero recomendable para los demás.
Herramientas de creación:

- **mkinitrd**: **RedHat, Fedora y relacionados**. Hay que pasarle el nombre de imagen de disco RAM y su nº de versión del Kernel: "**mkinitrd /boot/initrd-2.6.32.4.img 2.6.35.4**". Esto crea el archivo "/boot/initrd-2-6-35-4". Con "**-f"** borra los anteriores antes de crear el nuevo.
- **mkinitramfs**: igual que "**mkinitrd**" pero hay que especificar el nombre con "**-o**", ejemplo: "mkinitramfs -o /boot/initramfs-2.6.35.4.img 2.6.35.4".

Con el comando "**rdev**" se cambia las opciones de modo VGA, tamaño de disco RAM, etc. **Sin argumentos** muestra la línea de "**/etc/mtab**" con el sistema root actual.

201.3 Parchear el Kernel:

No siempre es necesario, permite realizar cambios en las fuentes del Kernel sin tener que descargar el Kernel nuevo entero, se obtiene el archivo "**patch-version.gz** o **patch-version.bz2**".

Con "**gzip/bzip2 -cd ../ patch-version.gz/bz2 | patch -pl**".

Para quitar el parche "unzip -c ../patch-version.tar.bz2 | patch -Rp1".

201.4 Personalizar, compilar e instalar un kernel personalizado y módulos

Empaquetar el Kernel:

Para instalar el Kernel en otro sistema, se hace con "**rpm-pkg, binrpm-pkg o deb-pkg**" como parámetros a make.

Es recomendable instalarlo sin actualizar el anterior Kernel (lo borra) por si no arrancase el pc.

Instalación de un binario del Kernel:

Una vez compilado el Kernel lo instalamos, copiándolo a "**/boot**", que suele estar desmontada para evitar errores en algunas distros. La imagen a copiar es un bzImage en la mayoría de las distros "**cp arch???/x86/boot/bzImage /boot/bzImage-2.6.35.4**".

No es obligatorio, pero recomendable copiar también el archivo "**System.map**" que se usa para depuración del Kernel. Copiamos y creamos enlace simbólico "**/boot/System.map**".

Todo esto nos lo podemos ahorrar con "**make && make install**" que compila, copia el Kernel, copia System.map y modifica el bootloader.

Instalar los módulos del Kernel:

Con "**sudo make modules_install**". Esta llama también a "**depmod**" que regenera el archivo "**Modules.dep**" de dependencias.

Añadir el Kernel a Grub:

La manera más sencilla es copiar una entrada existente, pegarla y editarla.

- **Grub1**: cambiar el "**title**" para identificarlo, cambiar el nombre de archivo en la línea "Kernel". Si se usa otro disco RAM, cambiar la línea "**initrd**". No borrar la otra sin haber probado a iniciar el nuevo Kernel.
- **Grub2**: copia de seguridad a "**/boot/grub/grub.cfg**". Hacemos un "**sudo update-grub2** o **grub-mkconfig**". Si no lo encuentra porque tiene un nombre personalizado (no se llama vmlinuz o Kernel) hay que añadirlo a mano en "**/etc/grub.d/40-custom**". Modificar el "**menuentry** (el título)" y los nombres del Kernel y disco RAM. Si queremos cambiar el Kernel por defecto al inicio cambiamos el parámetro "**default**" en "**/boot/grub/grub.cfg**".

***Nota**: si se instala desde binario, probablemente lo añada de forma automática.*

201.5 Administrar módulos del Kernel durante la ejecución:

Obtener información sobre el Kernel:

Con "**uname -a**", con "**-o**" indica el SO (Linux, FreeBSD, Solaris, MacOS...). El directorio "**/proc**" tiene mucha información.

Obtener información de los módulos del Kernel en ejecución:

Usamos "**lsmod**" para ver los módulos cargados en el sistema. Tiene una columna llamada

"Module" que se le pasaría como parámetro a "**modinfo**" si queremos ver más información sobre ese módulo. La columna "**Used by**" de "**lsmod**" describe que usa el módulo. **Lsmod** no muestra info de los drivers que se compilan directamente sobre el Kernel.

Cargar módulos del Kernel: mediante dos programas:

- **ismod**: carga un único módulo en el Kernel (No sus dependencias). Requiere el nombre completo.
  ej: "**insmod /lib/modulos/2.6.35.4/Kernel/drivers/cdrom/cdrom.KO**".
- **modprobe**: carga el módulo y los módulos de los que depende. Es más sencillo.
  Ej: "**modprobe cdrom**".**Parámetros**:
  - **-r**: elimina.
  - **-f**: fuerza la carga del módulo.
  - **-l partedelnombre***: lista los módulos que coinciden.

En la práctica puede que no haga falta, ya que Linux los carga automáticamente, pero puede ser útil para probar nuevos módulos.

Eliminar módulos del Kernel:
Con "**rmmod**", que es lo **opuesto** a "**insmod**", funciona como modprobe "**rmmod nombre_módulo**". Con "**-f**" fuerza. Con "**-r**" borra el módulo y los que dependen de él.

Herramientas y archivos para el mantenimiento de módulos:

- **Dependencias de los módulos**: se almacenan en "**/lib/modules/versionKernel/modules.dep**". El comando "**depmod**" lo reconstruye.
- **Configuración de los módulos**: en "**/etc/modules.conf** o **/etc/modules.conf.d** o **/etc/modprobe.conf** o **/etc/modprobe.conf.d**".
- **Pasar opciones a los módulos del Kernel**: edita los archivos de config y pasarle opciones añadiendo una línea de este tipo "**options nombre_módulo opciones**" y luego reconstruir con "**depmod**".

***Nota**: para asegurarnos que un dispositivo usa siempre un determinado módulo, hay que añadir "**alias dispositivo módulo**" en "/etc/modules.conf" o "/etc/modprobe.conf", ejemplo: "alias eth0 3c59x".*

Tema 202 Inicio del sistema

202.1 Personalizar el arranque y procesos de inicio

Identificar los servicios de un modo de ejecución:
Hay 2 maneras:

- Editando el archivo "**/etc/inittab**" que tiene el siguiente formato "**id:runlevels:acción:proceso**". **Modos de ejecución:**

0	Apagado
1	Monousuario
2	Multiusuario en derivadas de Debian
3	Multiusuario en Fedora, RedHat y derivadas
4	No se usa
5	
6	Reinicio

  Para que los cambios surjan efecto hay que reiniciar o hacer un "**telinit Q**".
- **Scripts de inicio o SysV**: se encuentran en "**/etc/init.d/rc**" o en "**/etc/rc.d/rc"**. Los scripts específicos para cada nivel de ejecución se encuentran en "**/etc/init.d/rcX.d**" (X= runlevel).
  Se le pasa "**start**" a los nombres que empiezan por "**S**" y "**stop**" a los "**K**".
  Se ejecutan por orden numérico "**tipoS/KnºNombre**", ejemplo "S10 network".
  Modificar estado de un servicio "**/etc/init.d/nombre_servicio start/stop/restart/state**" o con "**service nombre acción**".
  Los scripts de inicio SysV de los directorios de los modos de ejecución, son vínculos simbólicos al script original.

- **Upstart**: más moderno que SysV. En distros modernas. **No** usa "**/etc/inittab**", **usa** "**/etc/init**" (en antiguos Upstart usa "/etc/event.d").
  Añadir servicio: crear un **nombre_servicio.conf** en "**/etc/init/**" y añadir con "**start on runlevel [34]**" y **parar** con "**stop on runlevel [016]**" (inicia en el 0,1 y 6).
  Para que Upstart relea los archivos de config, escribimos "**initctl reload**". Es compatible con SysV.

Herramientas para gestionar programas activos en modos de ejecución:

- **chkconfig**: listar servicios con "**-l**". Para un sólo servicio sería "**chkconfig -l nombre_servicio**".
  Para modificar modos "**chkconfig --level 23 nombre on/off**" (2, 3 son los runlevels que queramos).
  Si no aparece un servicio, lo registramos con "**chkconfig -add nombre**".
- **Update-rc.d**: en Debian, formato "**update-rc.d [opciones] nombre acción**".
  Opciones:
  - **remove**: limpia el servicio
  - **defaults**: crea niveles para iniciar en los runlevels 2,3,4,5 y detenerlos en 0,1,6.
  - **start nº runlevels**: empieza en esos runlevels.
  - **stop nº runlevels**: para en esos runlevels.

Cambiar el modo de ejecución:
Con "**init/telinit nº_runlevel**, **shutdown**, **halt**, **reboot** y **poweroff**".

Comprobar el modo de ejecución:
Buscar "**initdefault**" en "**/etc/inittab**" o escribir "**runlevel**".

Grub para indicar el runlevel:
Al final de la línea del **Kernel** escribir "**runlevel=nº_runlevel**".

Modo seguro:
En grub2 podemos entrar en modo recuperación si pulsamos "**Shift"** durante el inicio y podremos elegir la entrada de "recovery". En ella podemos usar fsck, actualizar, etc.

Grub shell:
Pulsando "**c**" podemos entrar en la línea de comandos.
Hay varias shells: normal "**sh:grub>**" y la de rescate cuando ocurre algo grave "**grub rescue>**".

Tema 203: Sistemas de archivos y Dispositivos

203.1 Manejando el sistema de archivos de Linux

Tipo de sistemas de archivos:

- **Ext2fs** o **ext2**: fue creado para Linux.
- **Ext3**: ext2 con respaldo de transacciones.
- **Ext4**: ext3 que permite trabajar con discos de 32 Tb y archivos de más de 2TB.
- **ReiserFS**: desde 0, popular para muchos archivos pequeños.
- **JFS**.
- **XFS.**
- **Btrfs**: el candidato futuro para sustituir a los demás.
- **FAT**: en Linux con nombres largos-> vfat.
- **NTFS**: linux lee y sobreescribe.
- **HFS** y **HFS**+: de MacOS.
- **ISO-9660**: de CD-ROM.
- **Joilet**: cdrom propietario de Windows.
- **UDF**: formato universal de disco, dvd, regrabables, etc.

Montar sistemas de archivos:
Usando "**mount -t tipo_fs -o opciones dispositivo punto_montaje**".

Parámetros de mount:

- **-a**: monta lo listado en "/etc/fstab".
- **-r**: en sólo lectura.
- **-w**: escritura.
- **-t**: tipo de sistema de archivos (automáticamente lo detecta).
- **-L etiqueta**: añade etiqueta.

Opciones de mount (-o):

- **defaults**.
- **loop**: para montar un disco ".img o .iso" como si fuera una partición.
- **auto/noauto**: monta en el arranque.
- **user/nouser**: permite a usuarios normales hacer mount.
- **users**: umount cualquiera.
- **owner**: sólo el propietario puede montarlo.
- **ro**: sólo lectura.
- **rw**: lectura escritura.
- **uid=x**: indica el propietario de todos los archivos, usando el UID de "/etc/passwd".
- **gid=x**: como UID pero para grupos, usa "/etc/group".
- **umask=x**: para archivos.
- **dmsk=x**: para directorios.

Montar sistemas de archivos de manera permanente:
El archivo "**/etc/fstab**" es la tabla de sistemas de archivos en Linux. Cada una de las líneas que contiene corresponden a una partición.

Cada línea contiene 6 campos:

- **Dispositivo**: indica el dispositivo montado. Ej: "/dev/sdb5". También por UUID (nº de partición largo) o LABEL. Con "blkid /dev/dispositivo" se devuelve el UUID.
- **Punto de montaje**: es donde se montará la partición o disco. Con excepciones "/" y "**swap**". Los medios estraibles se montan en "**/mnt** o **/media**".
- Tipo de sistema de archivos: con auto lo detecta automáticamente.
- **Opciones de montado**: indica el modo en el que el Kernel tratará el sistema de archivos. Se pueden especificar varios, separándolos por comas, ej: users, noauto, etc. Las mismas opciones que el comando mount.
- **Copia de seguridad**: 1 si la utilidad "**dump**" hace copia de seguridad y 0 en caso contrario. Ya no es popular el uso el "**dump**", **apenas se usa**, por lo que siempre es 0.
- **Orden de revision del sistema de archivos**: orden en el que la herramienta "**fsck**" revisa la integridad. Si es 0 no revisa. La partición "/" debe de tener valor 1 y los demás valor 2.

Ejemplo: UUID=0..12.. /media/Datos ext4 users 0 0

***Nota**: la opción "**credentials**" es para compartidos de samba que requieren usuario y contraseña, se puede poner "**username=nombre,password=contraseña**", pero se podría leer por cualquiera. En lugar de eso se usa "**credentials=archivo**", con permiso de sólo lectura para root y formato:*

username=nombre
password=contraseña

Montado automático de sistma de archivos:
Antiguamente se en encargaba "**Hal**", pero ahora lo gestiona "**Udev**". Se inserta el pen, Udev detecta el cambio y el gestor de archivos tiene el control.

Autofs:
Evita tener que montar siempre las particiones de Samba. El archivo de configuración está en "**/etc/auto.master**". Ejemplo, le añadimos "/remote /etc/auto.servers –timeout=10". Luego creamos "**/etc/auto.server**" y en el interior aladimos los puntos de montaje.

Determinar lo que está montado:
Usamos "**mount**" sin parámetros o "**cat /etc/mtab**" (muestra todo lo montado) o "**cat /proc/mounts**". Y con "**df**".

Desmontar:
Con "**umount**". Si da error porque está en uso, podemos usar las herramientas "**lsof**" y "**fuser**" para listar archivos abiertos y los procesos que están usando sosos archivos.

***Nota**: el comando "**sync**" sincroniza la caché de un disco, pero no desmonta el sistema de archivos anterior.*

203.2 Mantenimiento de los sistemas de archivos:

Crear un sistema de archivos:
Ejemplo "**mkfs** -t ext3 /dev/sda6". Con "**-C**" se comprueban baques defectuosos "**badblock**". Hay "mkfs" de : ext2, extg3 y ext4, en estos últimos 3, una opción importante es "-m porcentaje", que define el % de espacio reservado en un disco (5% por defecto), si queremos 2% pues sería "-m 2". En discos extraible m es 0.
***Nota**: con "**-L etiqueta**" se añade etiqueta, excepto con mkreiserfs (-l) y mkfs.vfat (-n).*

Revisar en busca de errores:
Con "**fsck**". Opciones

- **-A**: verifica todos los fs de /etc/fstab. Se suele usar al inicio.
- **-C**: muestra indicador de progreso.
- **-V**: resumen detallado.

Hay que usarlo con sistemas desmontados o de sólo lectura.

Obtener información de los sistemas de archivos:
Para la familia ext, se usa "**dumpe2fs opciones dispositivo**". En XFS "**xfs-info** dispositivo" y para ReiserFS "**debugreiserfs**".

Ajustar sistemas de archivos:
Familia ext con "**tune2fs** opciones dispositivo". Opciones de tune2fs:

- **-c nºde montado**: nº de montajes para que salte fsck.
- **-C nºde montado**: engaña al contador de montados.
- **-i nº dwm**: días, semanas, meses para que salte fsck.
- **-m %**: espacio de disco reservado.
- **-L etiqueta**: cambia la etiqueta.

Para **XFS** usamos "**xfs-admin opciones dispositivo**", Opciones: "**-l etiqueta**" cambia la etiqueta y "**-m nº de montados**" igual que "**tune2fs**".
Para **ReiserFs** usamos "**reiserfstune opciones dispositivo**", con las mismas opciones que "**xfs-admin**".

Depurar el fs de manera interactiva:
Con "**debugfs**, **xfs_db** y **debugraiserfs dispositivo**" y aparece un prompt para meter los siguientes comandos "**show_super_stats** o **stats**" que muestra la información de superbloques.

Manipular espacio de intercambio:
La swap con código 0x82. Creamos swap con "**mkswap opciones dispositivo**", como opciones se suele usar "**-L etiqueta**" para etiquetar. Normalmente se suele crear en particiones, pero también se puede usar un archivo.
Para activar "**swapon dispositivo**", si usamos "**-a**" activa todas las listadas en "**/etc/fstab**". Para desactivar usamos "**swapoff**".

Administrar discos ópticos:
No se pueden meter datos directamente, hay que crear el sistema de archivos completo con "**mkisofs** o **genisofs**" que genera un **ISO-960**0. Para DVD "**growisofs**".
Podemos usar GUI como Gnome Toaster, K3b, etc
Ejemplo: "mkisofs -J -r -V "volumen nan" -o ../imagen.iso /directorio_a_meter" y grabaríamos con "cdrecord dev=/dev/dvdrw Speady=4 ../imagen.iso".

***Truco**: montar iso "**mount -t iso 9669 -o loop image.iso /mnt/cdfalso**".*

UDF: "**mkisofs -UDF**". Será de lectura, para escribir tenemos que tener instalado "**udftools**". Usamos "**mkudffs /dev/dvdrw**".

203.4 UDEV:
Crea entradas para el hardware. Reglas Udev en "**/etc/udev/rules.d**", se controlan en orden numérico. Mostramos info con "**udevadm info**", en distros nuevas es "**udevinfo**".
Opciones de Udev:

- **==**: compara igualdad.
- **!=**: compara desigualdad.
- **=**: asigno el valor a la clave reemplazando el antiguo valor.
- **+=**: añade el valor al grupo de valores existentes.
- **:=**: asigna el valor a la clave y anula futuros cambios.

Crear regla: creamos archivo en "**/etc/udev/.d/99-my.rules**".

Monitorizar Udev: con "**udevmonitor**", escucha los eventos del kernel producidos por una regla udev y muestra la información por consola.

Para controlar el comportamiento usamos "**udevadm**".

Tema 204: Administración avanzada de disco

204.1 Raid

Crear particiones:
Podemos crear usando la librería "**libparted**", usada por los programas: Gparted, fdisk y fdisk de GPT (programas **gdisk** y **sgdisk**).
Todos los programas se usan de esta manera "**programa /dev/dispositivo**".
La configuración RAID se almacena en "**/etc/raidtab**".

Nota**: existe **sfdisk** que se usa en modo no interactivo para usarlo con **scripts

Comandos **fdisk**:

- **d**: borra partición.
- **l**: muestra la lista de códigos de tipo de particiones (0x07=NTFS, 0x82=swap, etc).
- **n**: nueva partición.
- **o**: destruye la tabla de particiones.
- **p**: muestra la tabla de la partición actual.
- **q**: sale sin guardar cambios.
- **t**: cambia el código tipo de partición.
- **w**: guarda y sale.

***Truco**: podemos visualizar la tabla de partición con "**fdisk -lu /dev/dispositivo**" y sale.*
Nota: los códigos de partición en GPT son valores GUID de 16 bytes, ejemplo: NTFS sería 0x0700, ext sería 0x8300, swap 0x8200.

Tipos de RAID:

- **Modo lineal**: no es técnicamente RAID, pero es gestionado por RAID en Linux. Combina discos sumando espacio.
- **RAID0**: combina discos sumando espacio. Se lee/escribe de forma intercalada, una parte en cada disco. Mejora velocidad. Si un disco falla, los demás serán inútiles. LVM también usa división por bloques.
- **RAID1**: (mirrow) crea copias exactas en discos. Mejora fiabilidad, pero es más lento. El disco adicional puede estar como: "hot stanby" o disco de reserva (host spare) que permanece inactivo hasta que el otro disco falla, por lo que debe de copiar los datos al otro disco.
- **RAID4**: intenta mantener beneficio del 0 y 1. Los datos se dividen como en RAID0, pero hay un disco que almacena sumas de control para regenerar datos perdidos. Necesita 3 discos mínimo.
- **RAID5**: como RAID4, pero sin el disco que almacena las sumas de control, estas sumas se intercalan en todos los díscos. Mínimo 3 discos.
- **RAID6**: Si más de una unidad falla en RAID5 o 6, se pierden los datos. Para evitar esto se usa un disco más, para sumas de control. X-2. Mínimo 4 discos.
- **RAID10**: mezcla entre RAID1 y 0, conocida como RAID 1-0. Parecido a 4 o 5.

***Nota**: spare significa unidad de repuesto.*
***Nota**: para calcular el tamaño de RAID 4 y 5: (nº de discos -1) x (suma de tamaño de discos restantes).*

Raid en Linux:
Las particiones se combinan con los drivers raid del Kernel, con nombre "**/dev/md#**" (en raid por software, en hardware aparecen normales). Esto permite usar RAID para dispositivos enteros, sólo para particiones o para discos con distinto tamaño.

***Nota**: Grub1 no soporta RAID, Grub2 si. Se puede usar Grub1, pero dejando una partición "**/boot**" aparte.*

Preparar disco para RAID software:
Dos opciones:

- Crear una partición RAID en cada disco y se define el conjunto entero.
- Se crean las particiones en cada disco y luego se unen. Más flexible.

RAID por hardware es más eficiente que por software.

Ensamblar un conjunto RAID:
Una vez preparadas las particiones que se incluirán en el conjunto RAID, usamos la herramienta "mdadm" para ensamblar las particiones.
"**mdadm [modo] dispositivo_raid [opciones] dispositivos_integrantes]**"

Tipos de **modo**:

- **-A**: ensambla un conjunto ya ensamblado previamente.
- **-B**: crea un conjunto RAID sin metadatos (sólo para expertos).
- **-C**: crea conjunto y metadatos.
- **-F**: monitoriza cambios de estado.
- **-G**: modifica el conjunto.
- **-I**: añade un único dispositivo RAID.
- **sin parámetros**: busca RAIDs y los activa.

Opciones:

- **-h**: ayuda.
- **-v**: info adicional.
- **-V**: versión.
- **-f**: fuerza.
- **-c archivo**: especifica el archivo de configuración, que por defecto es "**/etc/mdadm.conf**" o "**/etc/mdadm/mdadm.conf**".
- **-s**: obtiene información perdida sobre el archivo de configuración.
- **-e nºnivel**: tipo de metatados, por defecto 0,90.
- **-n**: establece el nº de dispositivos activos en el conjunto más el nº sobrantes. Es el total.
- **-x num**: establece el nº de dispositivos sobrantes extras.
- **-c tamaño**: tamaño de bloques, por defecto 64.
- **-l nivel de raid**: linear, raid0, etc.
- **-N nombre**: establece un nombre para el conjunto.
- **-a**: añade dispositivos al conjunto.
- **-r**: borra dispositivos del conjunto.
- **-S**: de stop, desactiva el conjunto.

Ejemplo "mdadm -- create /dev/md0 --level=5 --raid-devices=3 /dev/sda6 /dev/sdc1 /dev/sdd1".

Una vez creado, lo usamos como particiones, o particionarlos o usarlos como volúmenes LVM.
Ejemplo de particiones de /dev/md0: md0p1, md0p2, etc

Revisar configuración RAID:
Con "**cat /proc/mdstat**", que contiene la información del estado de dispositivos RAID.

204.2 Ajustar acceso al disco

Identificar el uso de recursos del disco:
Consultar los IRC con "**cat /proc/interrupts**". Si hay errores, podemos ajustar a mano las IRQ con la utilidad "**sysctl**", cuyo archivo de configuración es "**/etc/sysctl.conf**". Vemos errores fácilmente con "**sysctl -a | grep irq**".

Probar funcionamiento del disco:
Con "**hdparm -t /dev/dispositivo**". Con "**-T**" prueba la caché del disco, se suele poner "**-tT**" para probar las dos a la vez. Para discos SCSI es "**sdparm**".

Monitorizar fallos:
Con la función SMART "**smartctl -a /dev/dispositivo**", podemos monitorizar errores.

204.3 LVM

LVM es parecido a RAID, pero más flexible y fácil. Comprende 3 niveles de estructuras de datos: volúmenes físicos, grupos de volumen y volúmenes lógicos (equivalente a particiones).
Las particiones tradicionales empiezan en un sector del disco y terminan en otro, son inflexibles.

***Nota**: va por la versión 2, se instala lvm2, **apt-get install lvm2**.*

***Nota**: EVMS es un sistema para administrar RAID, LVM y otros.*

***Nota**: Grub1 no lee LVM, por lo que hay que sacar fuera "**/boot"**.*

Estructuras:
- **Volúmenes físicos**: como las particiones convencionales.
- **Grupos de volúmen**: colección de uno o más volúmenes físicos, que son gestionados como un único espacio de asignación. 2 discos de 1TB= 2TB.
- **Volúmenes lógicos**: se crean "particiones " sin hacer referencia a sectores del disco.

***Nota**: El cisco de RAM inicia debe de tener RAID y LVM.*

***Nota**: si hay 2 discos en LVM y falla 1, se pierde todo.*

Crear y manipular volúmenes físicos:
Código MBR de particiones **LVM= 0x8E**. Primero hay que marcar los volúmenes físicos, para usarlos como LVM. Luego hay que etiquetarlos con las herramientas "**/sbin/pv*: pvchange, pvck, pvcreate, pvdisplay, pvmove, pvrename**, etc."

Ejemplo: pvcreate /dev/sda6 /dev/sdb /dev/sdc, luego comprobamos con pvdisplay.

Crear y manipular grupos de volúmen:
Con las herramientas "**/sbin/vg*: vgcfgbackup, vgcfgrestore, vgck, vgcreate, vgdisplay, etc**".

Ejemplo: vgcreate nombre_volumen /dev/sda6..., comprobamos con vgdisplay.

Crear y manipular volúmenes lógicos:
Con las herramientas "**/sbin/lv*: lvcharge, lvconvert, lvcreate, lvdisplay, lvextend, lvreduce, lvremove, lvrename, lvscan**".

Ejemplo: "lvcreate -L tamaño -n mi_volumen nombre_volumen", hay que formatearlo con "mkfs.ext4 /dev/mapper/nombre_grupo-mi_volumen" y luego hay que montarlo con "mount /dev/mapper/nombre_grupo-mi_volumen /media/lvm".

Por cada Volúmen Lógico que tengamos, se creará un dispositivo dentro de "**/dev/mapper/**" que estará compuesto por el nombre del Grupo de Volumen, un guión y el nombre del Volúmen Lógico, por ejemplo, si el Grupo de Volúmen se llama servidores y la Volúmen Lógico que hemos creado se llama ftp, entonces se creará el dispositivo /dev/mapper/servidores-ftp. Por otro lado, también se creará un enlace simbólico con la forma "/**dev/grupo_de_volúmen/volúmen_lógico"**, en el caso del ejemplo, se llamaría: /dev/servidores/ftp.

Instantáneas:
Es un volúmen lógico, que preserva el estado de otro volúmen lógico. Se crean muy rápido.

Ejemplo: creo instantáneas, intento actualizar y no funciona bien, pues restauramos la instantánea.

***Nota**: Btrfs (sistema de archivos reciente) incluye instantáneas.*

Creamos instantáneas con "**lvcreate -s**". Ejemplo: "lvcreate -L 10GB -s -n snappy /dev/speaker/PCLOS", que crea un nuevo volúmen lógico llamadom snappy, que dublica el contenido de /dev/sepaker/PCLOS en una instantánea de 10Gb.

Podemos v**er la capacidad** de la instantánea: con "**lvs**" y "**lvdisplay**".

Restaurar "**lvconvert --merge /dev/speaker/snappy**".

Nota: el orden es: pvcreate (creo volumen físico), vgcreate (creo grupo de volumen), lvcreate (creo volumen lógico), mkfs.ext4 (formateo la partición) y monto.

Tema 205: Configuración de red

205.1 Configuración básica de redes

Conectar por wifi:
Buscamos redes con "**iwlist**". Sin root sólo muestra a la que está conectado y con root conectado más disponibles. Formato "**iwlist adaptador comando**". Comandos **iwlist**:

- **scan** o **scanning**: muestra la lista de redes disponibles como root y la conectada si se usa usuario normal.
- **frec**, **frequency** o **channel**: muestra la lista de canales disponibles para el adaptador.
- **rate**, **bit** o **bitrate**: muestra todas las velocidades soportadas para el dispositivo.
- **keys**, **enc**, **encryption**: lista las claves de encriptación disponibles.
- **auth**: lista parámetros de autenticación establecidos actualmente.
- **wpa** o **wpakeys**: lista las claves wpa establecidas en el dispositivo.
- **power**: lista de modos de gestión de energía disponible.
- **txpower**: parámetros de energía de transmisión del adaptador.
- **retry**: lista de parámetros de reintento actuales.
- **event**: lista de eventos inalámbricos soportados por el dispositivo.
- **mod** o **modulation**: lista la modulación usada por el dispositivo.

Conectamos con wifi mediante "**iwconfig adaptador essid NombreWifi channel X mode Manager key s:clave**". La "**s**" indica que es una secuencia de texto, en hexadecimal no hace falta. Hay herramientas gráficas de conexión como "**wicd**" (wicd-gtk), también en consola "**wicd-cli**". Con "**iwspy**" obtenemos estadísticas de los nodos, comprueba calidad del enlace (junto con iwconfig).

Clientes DHCP:
Algunos clientes son "**pump**, **dhclient** y **dhcpcd**".

Ponder IP estática:

- **Fedora**: en el archivo "**/etc/sysconfig/network-scripts/ifcfg-adaptador**" sustituir en la línea "**Bootproto=dhcp**" el **dhcp** por **static**. Debe de quedar así:

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.1.2
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
GATEWAY=192.168.1.1
ONBOOT=yes
```

- **Debian**: en el archivo "**/etc/network/interfaces**" buscamos la línea "**iface adaptador inet dhcp**" y sustituimos **dhcp** por **static**. Debe de quedar así:

```
auto lo
iface lo inet loopback auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
```

network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1

***Nota**: podemos añadir varias IPs a un único interfaz, en ese archivo crearíamos otra configuración con "**eth0:0**", luego "**eth0:1**", etc. Tanto en Debian como en Fedora.*
También lo podemos hacer **mediante comando**s:

- IP y máscara: "**ifconfig adaptador up IP netmask MASCARA**"
- Puerta de enlace "**route add default gw PUERTA_ENLACE**".

Podemos ver la configuración con "**ifconfig** y **route**" sin parámetros.

Asignar DNS:
Los DNS los cambiamos en "**/etc/resolv.conf**", con el formato:

#Máximo 3 DNS
nameserver 208.67.222.222
nameserver 8.8.8.8
nameserver 8.8.4.4

Apagar encender adaptador:
Podemos habilitar y deshabilitar un adaptador "**ifup/ifdown** adaptador".

Configurar nombres de host:
Podemos añadir equipos para la resolución de nombres, añadiéndolos a "**/etc/hosts**" con el formato "**IP nombre**".

Podemos configurar nuestro nombre de equipo con el comando "**hostname nombre**" o editando el archivo "**/etc/hostname**" (en Debian) o "**/etc/sysconfig/network"** (en Fedora).
Podemos comprobar si se ha cambiado usando "**uname -a** o **hostname**".

Comprobar la conectividad básica:
Usando **ping** y luego "**Control-C**" o "**ping -C nºpaquetes IP/hostname**".

ARP permite ver y modificar la caché ARP, parámetros:

- **-d IP**: borra la de la caché.
- **-s IP MAC**: añade a la caché.

Comando IP:
El comando IP es un "todo en uno", su formato es "**ip [opciones] objeto comando**".
Objetos:

- **link**: realiza acciones en el hardware local.
- **addr**: asocia IP.
- **addrlabel**: muestra o ajusta IPs en una red IPv6.
- **route**: modifica la tabla de enrutamiento.
- **rule**: modifica iptables.
- **neigh**: muestra/modifica entradas ARP.
- **tunnel**: funciones de tunneling.
- **maddr**: direcciones de multidifusión.
- **mroute**: rutas de multidifusión.
- **monitor**: monitoriza la actividad de red.

205.2 Configuración avanzada de red

Configurar Linux como router:
Usando el comando "**route**", podemos redirigir el tráfico. Para activar el redireccionamiento con: "**echo "1" > /proc/sys/net/ipv4/ip_forward**", aunque será de forma temporal, para hacerlo de forma permanente hay que modificar "**/etc/sysctl.conf**" con "**net.ipv4.ip_forward=1**".

VPN:
Dos segmentos privados de red se enlazan a través de un router. Usamos OpenVPN.
OpenVPN usa criptografía de claves públicas.

Configurar la VPN: editar "**/etc/openvpn/client.conf**" o "**/etc/openvpn/server.conf**". Los parámetros a modificar son "**ca**, **cert**, **key** y si es server también **dh**".
Si los clientes deben de comunicarse entre si, eliminamos el comentario de "**cliente-to-cliente**" del servidor. Por seguridad borramos los comentarios de las directivas "**user nobody** y **group nobody**" del server.
En los clientes hay que editar el parámetro "**remote**" con la IP del host y el puerto (1194 por defecto).

Establecer conexión VPN: iniciar VPN con "**openvpn ruta_archivo_config**". Si inicia hacemos ping a la IP 10.8.0.1 (que es la red que se crea por defecto). Esta red podemos cambiarla modificando el archivo de configuración.

Configurar un PC como CA para certificados: mediante scripts que se encuentra en "**/usr/share/openvpn/easy-rsa**", lo copiamos a "**/etc/openvpn/**", editamos "**/etc/openvpn/vars**" y luego desde "**/etc/openvpn/**":

./vars
./clean-all
./build-ca

Cuando termine: "**./build-key-server Server**", luego los clientes "**build-key cliente1, etc**". Para finalizar hacemos "**./build-dh**".

Archivo	Copiar a	Propósito	Secreto
ca.crt	Servidor y todos los clientes	Certificado CA	No
ca.key	Sólo al host que firma la clave	Clave CA	Si
dh1024.pem	Sólo al servidor	Parámetros Diffie Hellman	No
server.crt	Sólo al servidor	Certificado del servidor	No
server.key	Sólo al servidor	Clave del servidor	Si
clienteⁿº.crt	Sólo al cliente nº	Certificado del cliente nº	No
clientnº.key	Sólo al cliente nº	Clave del cliente nº	Si

Información: Disponemos de dos ficheros con información de estado "**/etc/openvpn/openvpn-status.log**", que contiene la información sobre los clientes conectados al servidor VPN y que es actualizado cada minuto, y el fichero "**/etc/openvpn/ipp.txt**",que contiene la información sobre las IPs asignadas a los clientes de la VPN.
Pasos para **instalación y configuración**:

- ***apt-get install openvpn***
- ***mkdir /etc/openvpn/easy-rsa***
- ***cp -r /usr/share/doc/openvpn/examples/easy-rsa/20/* /etc/openvpn/easy-rsa***
- Modificamos valores como export KEY_COUNTRY=”ES” con “***gedit /etc/openvpn/easy-rsa/vars***”
- Linkeamos configuración “***cd /etc/openvpn/easy-rsa***” y luego “***ln -s openssl-1.0.0.cnf openssl.cnf***”
- Generamos CA: “***source vars***”, luego “***./clean-all***” y luego “***./build-ca***”
- Generamos certificado del servidor con “***./build-key-server miservidor***” y luego generamos los parámetros Diffie Hellman con “***./build-dh***”.
- Generamos certificados para los clientes “***./build-key cliente1***” así con todos.
- Copiamos a “***cp /etc/openvpn/ miservidor.crt miservidor.key ca.crt dh1024.pem***”.
- En los clientes copiamos: ca.crt, cliente1.crt y cliente1.key.
- Copiamos el archivo de configuración de ejemplo para el servidor “***cp /usr/share/doc/openvpn/examples/sample-config-files /server.conf.gz /etc/openvpn/***” y lo descomprimimos “***gzip -d /etc/openvpn/server.conf.gz***”.
- Editamos configuración del servidor: “***gedit /etc/openvpn/server.conf***” y modificamos los nombres de “ca ca.crt , cert miservidor.crt , key miservidor.key , dh dh1024.pem”.
- Reiniciamos servidor con “***/etc/init.d/openvpn restart***”.

Ejemplo de configuración del servidor:

```
port 1194
proto udp
dev tun
ca ca.crt
cert MoninoMagicBox.crt
key  MoninoMagicBox.key
dh dh1024.pem
server 192.168.2.0 255.255.255.0
ifconfig-pool-persist ipp.txt
push “route 192.168.1.0 255.255.255.0”
push “redirect-gateway def1 bypass-dhcp”
push “dhcp-option DNS 208.67.222.222”
push “dhcp-option DNS 8.8.8.8”
client-to-client
keepalive 10 120
comp-lzo
persist-tun
persist-key
status openvpn-status.log
verb3
```

Conexión entre red virtual y real:

Para que las dos redes se vean:

- Editar “***gedit /etc/sysctl.conf***” y descomentar la línea “***net.ipv4.ip_forward=1***”.
- Configurar iptables para enrutamiento “***iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE***”.
- Probamos que funcione y guardamos iptables “***sudo sh -c “iptables-save > /etc/iptables.rules***”.
- Para hacer que esa configuración se cargue automáticamente hay que editar “***gedit /etc/network/interfaces***” y añadir “***pre-up iptables-restore < /etc/iptables.rules***” al interfaz correspondiente, por ejemplo:

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet static
address 192.168.1.2
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 208.67.222.222 8.8.8.8
pre-up iptables-restore < /etc/iptables.rules
```

Configuración de los clientes:

- **Ubuntu:** Instala OpenVPN: "***sudo apt-get install openvpn***". Copiar fichero de ejemplo de configuración de cliente. Este fichero también se puede emplear en la configuración de un cliente Windows sin hacer prácticamente ningún cambio. "***sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/***"
  Copiar los certificados a */etc/openvpn* y **modificar** ***/etc/openvpn/client.conf*** con los nombres correctos:
  ca ca.crt
  cert cliente1.crt
  key cliente1.key
  Habilitar o añadir los siguientes parámetros
  client
  remote miservidor.ejemplo.com 1194
  Reiniciar servicio OpenVPN: ***/etc/init.d/openvpn restart***

- **Win7:** Descargar e instalar el cliente desde la web de OpenVPN, y añadir el siguiente fichero de configuración y los certificados del cliente.

```
# C:\Program Files\OpenVPN\config\client.ovpn
client
remote server.example.com
port 1194
proto udp
dev tun
dev-type tun
ns-cert-type server
reneg-sec 86400
auth-user-pass
auth-retry interact
comp-lzo yes
verb 3
ca ca.crt
cert "C:\\Users\\username\\My Documents\\openvpn\\cliente1.crt"
key "C:\\Users\\username\\My Documents\\openvpn\\cliente1.key"
management 127.0.0.1 1194
management-hold
management-query-passwords
auth-retry interact
```

- **Mac OS X (Tunnelblick):** Descargar **Tunnelblick** y situar el siguiente fichero de

configuración de ejemplo cliente.ovpn y los certificados correspondientes en el directorio ***/Users/username/Library/Application Support/Tunnelblick/Configurations/***

```
# sample client.ovpn for Tunnelblick
client
remote blue.example.com
port 1194
proto udp
dev tun
dev-type tun
ns-cert-type server
reneg-sec 86400
auth-user-pass
auth-nocache
auth-retry interact
comp-lzo yes
verb 3
ca ca.crt
cert client.crt
key client.key
```

Monitorizar tráfico de red:

- **nc**: sirve para abrir conexiones TCP/UDP, enviar paquetes, etc:
  - Conexión simple "**nc 127.0.0.1 (IP) nºpuerto**", si hay alguna conexión corriendo sobre ese puerto, dará error de conexión cerrada.
  - Ver puertos abiertos "**nc -vz IP Puertos(21-25)**" indica si está abierto. Para ver los UDP sería "**nc -vzu IP puertos**" (v de verbose).
  - Cliente /servidor: es posible montar un servidor de prueba, montamos con "**nc -l puerto_escucha**" y en el cliente "**nc IP puerto**". Podemos mandar datos entre ellos, en server "**nc -l puerto_escucha > algo.out**" y en cliente "**nc IP puerto < algo.in**".
- **netstat**: muestra conexiones activas, podemos usar "**| less**" para leer poco a poco. Opciones:
  - **-a**: muestra las conexiones abiertas, con "**-t**" muestra las TCP y con "**-u**" las UDP.
  - **-i**: información del interfaz.
  - **-r**: información de enrutamiento.
  - **-M**: información de enmascaramiento.
  - **-p**: información de programas que usan la red (no funciona siempre).
  - **sin parámetros**: muestra los puertos abiertos.

  Ejemplo "**nestat -ap | grep smtp**". Podemos usar el programa "**watch**" para ejecutar cada 2 segundos, creando una salida, por si el problema es intermitente.
- **tcpdump**: es un sniffer, hay que usarlo como root.
- **wireshark**: antiguamente se llamaba Ethereal, es otro es sniffer. Podemos usarlo como comandos "**tshark**" o con **gui**.
- **Nmap**: es un escáner de red, no monitoriza el tráfico. Ej: comprobar puertos abiertos con "**nmap -sT host**" (T para TCP, U para UDP).
- **ARP**: modifica la caché ARP. Parámetros:
  - **-a [host]**: muestra todas las entradas o del host si se indica.
  - **-d host:** borra ese host de la caché.
  - **-s host mac**: crea la entrada.
- **lsoft**: muestra todos los procesos haciendo uso de archivos, podemos usar "**lsoft | less**". Podemos ver de un solo proceso con "**lsoft -p ID_proceso**". Con la opción "**-i**" muestra todos los archivos de red usados por el procesos actual "**lsoft -i | grep nombre_servicio**".

205.3 Solucionar problemas de red

Consultas sobre red:

- **ifconfig**: verificamos el interfaz.
- **ping**: verificar conectividad.
- **router**: verificar las rutas.
- **traceroute**: verifica rutas, con "**-n**" sólo muestra las IPs de las rutas que sigue. Algunos routers bloquean los traceroute (aparecerían paréntesis sólamente).
- **host dominio**: para hacer consulta de DNS, si usamos el comando "**dig**" podemos hacer una búsqueda más completa "**dig @IP_DNS dominio**".
- **netstat**: verificar conexiones abiertas.
- **hostname**: verifica el nombre de nuestro pc, con "**-f**" muestra FQDN. Podemos saber la IP escribiendo "**host nombre_hostname**". Se pueden combinar los dos comandos "**host 'hostname-f'**"".
- **iptables**: comprobar que IPtables no está capando "**iptables -L**" para ver reglas. Hay que observar en las cadenas "**INPUT** y **OUTPUT**", si son "**DROP** o **ACCEPT**".
- **TCPWrappers**: comprobar "**/etc/hosts.allow** y **.deny**", por si están denegando el acceso a equipos.

Verificar DNS del cliente:

- Comprobar que en el archivo "**/etc/nsswitch.conf**" esté puesto "**dns**" en el apartado "**host**".
- Verificar que en el archivo "**/etc/hosts**" no haya redirecciones erróneas.
- Verificar que en "**/etc/resolv.conf**" haya entradas de DNS
- Realizar **ping** y **host** a un dominio.

Otros: verificar "**/etc/network**", "**/etc/sysconfig/network-scripts/**" si están por DHCP o estático.

205.4 Notificar incidencias a usuarios

Definir mensajes de acceso:

Cuando los usuarios hacen login a través de un prompt, se muestran los siguientes mensajes:

- **Mensaje del día**: en el archivo "**/etc/motd**", se muestra local, telnet y ssh.
- **Mensajes de la suerte**: hay que añadir el programa "**fortune**" en "**/etc/bashrc**":

```
if [$TERM !="dumb"];
  fortune
fi
```

- **De acceso local**: en el archivo "**/etc/issue**"
- **De acceso de red**: en el archivo "**/etc/issue.net**", para telnet, pero no para ssh.

En los mensajes de acceso local y de red, se pueden usar **variables**:

- **\n**: indica el nombre del host.
- **\r**: indica el nº de versión del kernel.
- **\s**: indica el nombre del so.
- **\m**: indica la plataforma (x86, x64).

Mensajes en tiempo local:

- Alerta de apagado: "**shutdown -h +10 "se apaga el equipo por Error37""**".
- Escribir a todos los terminales: "**wall "Acceso a red se apaga a las 17""**".
- Los usuarios pueden bloquear estos mensajes con "**mesg n**" y activarlos con "**mesg y**".

Tema 206: Mantenimiento del sistema

206.1 Compilar e instalar programas desde la fuente

Hay que tener instalado: GCC, los headers, tar y obtenemos el código fuente.

*Nota: si nos encontramos con un archivo "**programa.src.rpm**", lo compilamos con "**rpmbuild --rebuild programa.src.rpm**".*

Desempaquetar:
Con el comando "**tar txvf programa.tgz**", con la opción "**t**" se fuerza a crear un subdirectorio con el contenido, ya que en los zip no los crea por defecto.
Si se quiere que esté accesible para todos los usuarios, en vez de en local, hay que copiarlo a "**/usr/src/**".

Compilar:

- **Configurar**: con "**./configure**" se escanea el sistema y se genera el "**Makefile**". Admite parámetros como "**--prefix=directorio**", que sirve para cambiar el directorio de instalación, por defecto "**/usr/local/**".
- **Compilar**: ejecutando "**make**".
- **Instalar**: "**sudo make install**", los binarios se copiarán a "**/usr/local/bin**".
- **Limpiar**: "**make crean**".
- **Desinstalar:** "**sudo make uninstall**".

206.2 Operaciones de Backup

Elegir hardware (precios 2010):

Dispositivo	Coste Unidad	Coste dispositivo	Capacidad descomprimido	Velocidad	Tipo de acceso
Cinta	150-4000€	0,25 - 3€/GB	36GB - 1,5TB	3 - 140MB/s	Secuencial
HDD interno	100€	0,20€/GB	80GB - 2TB	50 - 100MB/s	Aleatorio
HDD externo	50-7000€	0,20€/GB	80GB - 2TB	12 - 100MB/s	Aleatorio
Óptico	25-200€	0,04 - 0,50€/GB	650MB - 50GB	1 - 45MB/s	Aleatorio

- **Cintas**: el más popular para copias enteras, son menos fiables que otros medios. Se acceden desde "**/dev/st0**" para SCSI o "**/dev/ht0**" en cintas PATA.
- **Discos duros**: debe ser extraíbles (esata?).
- **Ópticos**: fiables a corto plazo, suelen tener fallos tras uno o dos años.

Muchos admins usan la estrategia "**3-2-1**": **3** copias de los datos, al menos en **2** tipos de dispositivos diferentes , **1** copia entera.

Elegir el Software:
Software básico:

- **tar**: los tarball son archivos creados por tar, generalmente comprimidos con "gzip o bzip2". Formato "tar comando calificador".
  Comandos:
  - **c** (**--create**): crea un archivo.
  - **A** (**--concatenate**): añade ficheros tar a un archivo.
  - **r** (**--append**): añade ficheros al final del archivo.
  - **u** (**--update**): añade ficheros que son más recientes que los del archivo.
  - **d** (**--diff** o **--compare**): compara un archivo con los ficheros del disco.
  - **t** (**--list**): lista el contenido de un archivo.
  - **x** (**--extract** o **--get**): extra ficheros de un archivo.

  Certificados:
  - **--directory directorio**: pasa al directorio indicando antes de llevar a cabo las operaciones.
  - **g archivo** (**--listed-incremental archivo**): realiza una copia o restauración incremental, usando archivo como lista de ficheros previamente archivados.
  - **M** (**--multi-volume**): crea o extrae un archivo multicinta.
  - **v** (**--verbose**): va mostrando lo que lee/extrae.
  - **z** (**--gzip** o **--unzip**): procesa mediante gzip.
  - **j** (**--bzip2**): procesa mediante bzip2.

- **cpio**: similar a "**tar**", pero puede almacenar en disco o escribir directamente en cinta (sin pasos intermedios). Tiene 3 modos de funcionamiento:
  - **Modo copy-out**: usando "**-o**" o "**--create**", crea un archivo y copia ficheros en este.
  - **Modo copy-in**: usando "**-i**" o "**--extract**", extrae información. Se le puede pasar parámetros para extraer sólo archivos solicitados.
  - **Modo copy-pass**: con "**-p**" o "**--pass-through**", combina los dos modos anteriores copiando un árbol de directorio.

  Opciones:
  - **-A** (**--append**): añade datos a un archivo existente.
  - **-H formato** (**--format=formato**): bin, crc y tar.
  - **-u** (**--unconditional**): reemplaza todas los archivos sin solicitar confirmación.
  - **-v** (**--verbose**): muestra más información.

- **dd**: Las copias deben de hacerse en sistema desmontados. La copia ocupará la partición entera, aunque esté vacía. Operandos:
  - **bs=tamaño**: opera sobre un bloque de ese tamaño.
  - **count=bloques**: copia el nº de bloques especificado.
  - **if=archivo**: entrada.
  - **of=archivo**: salida.
  - **skip=bloques**: se salta esos bloques.
  - **seek=bloques**: escribe esos bloques.

  Ejemplos:
  - **Copiar una partición**: "dd if=/dev/dispositivo of=/media/b/copia.img".
  - **Restaurar copia de partición**: hay que intercambiar el if por el of y listo.
  - **Clonar disco entero**: "dd if=/dev/hda of /deb/hdb bs=1M". Clonaría el disco IDE1 al 2, bs=1M indica que la escritura/lectura se haga en bloques de 1MB, menos es

más lento y más grande nos arriesgamos a perder datos. Graba MBR, particiones, etc.
    - **Crear imágen de CD**: “dd if=/dev/cdrom of=/home/imagen.iso”. Para montar la imagen “mount -o loop /ruta/imagen.iso /mnt/imagen”.
    - **Copiar MBR**: “dd if=/dev/sda of=mbr count=1 bs=512”. Restaurar “dd if=mbr of=/dev/sda”,
    - **Borrado seguro de disco**: LLenar el disco con carácteres aleatorios 5 veces para que no quede rastro de información en disco “for n in {1..5]; do dd if=/dev/urandom of=/dev/gda bs=8b conv=notrue;done”.

  ***Nota**: podemos usar compresión “dd if=/dev/hda | gzip > /ruta/copia.bin.gz”.*

- **rsync**: realiza pequeñas copias de seguridad de red. Tiene GUI (**grsync**). Ejemplo “rsync -r -t -v --progress --delete -s /media/Datos /media/Datos2”.
  Otro ejemplo “rsync -av /directoriolocal usuario@remote:/directorioremoto”.
  Otro “rsync -avz alberto@192.168.1.30:/home/alberto/M/ /home/ahornero/M/”.

- **mt**: sirve para controlar un dispositivo de cinta. Para acceder a cada una de las copias dentro de una cinta, hace falta mt, para moverse hacia adelante y atrás en la **cinta** “**mt -f dispositivo operacion [cinta] [argumentos]”**. Operaciones:
    - **fsf**: mueve hacia delante la “cuenta” de archivos.
    - **bsf**: mueve hacia atrás la “cuenta” de archivos.
    - **eod** o **seod**: se mueve al final de los datos.
    - **rewind**: rebobina la cinta.
    - **offline** o **vrewoffl**: rebobina y expulsa la cinta.
    - **erease**: borra los datos.
    - **status**: muestra info.
    - **load**: carga unidad en cinta.
    - **compresion**: habilita (1) / deshabilita(0) la compresión o descompresión.

Tema 7: Servidor de Nombre de Dominio

207.1 Configuración básica de un servidor DNS

Surgió porque "**/etc/hosts**" se quedó pequeño. Tiene forma jerárquica. Un **TLD** es un dominio con el punto final, ej: "**www.dominio.ext.**".**Servidor de nombres de sólo caché**: almacena en caché las resoluciones de nombres ya realizadas.

Modificar el archivo de **configuración** principal de BIND:

El archivo de configuración es "/**etc/named.conf**" (Bind9). Tres config distintas:

- Servidor de **sólo reenvío**: reenvía a otros servidores todas las solicitudes de resoluciones de nombres que reciba.
- Servidor que **sólo** realiza **búsquedas recurrentes completas**: se realiza una búsqueda recurrente completa para cada consulta que no pueda responderse con su caché, lo usan los ISP.
- Servidor con **reenvíos y búsquedas recurrentes completas**: consulta y si falla hace la búsqueda completa.

```
Options {
        directory "/var/named";
        forwarders {
                10.9.16.30;
                10.13.16.30;
                };
        listen-on {
                192.168.1.1;
                172.20.21,1;
        };
        allow-tranfer {"none";};
        forward first;
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
//sepecify the root zone files
zone "." IN {
        type hint;
        file "named.ca";
};
include "/etc/named.rfc1912.zones";
```

Nota: *en la configuración "//" es comentario y "**;**" fin de una opción.*
Nota*: para comprobar que BIND se está ejecutando "**lsof -n -P -i | grep 53**".*

En la opción de "**forwarders**" se introducen las IPs de DNS a las que Bind transferirá las solicitudes de búsqueda que recibe.

La línea "**forward first**" indica a bind que debe funcionar como servidor de transferencia si es posible, pero debe realizar búsquedas recurrentes completa si falla.

Si “**forward first**” se cambia por “**forward only**”, bind sólo intentará obtener respuesta de los sistemas especificados en la sección “**forwarders**”, siempre se realizarán búsquedas completas.

En el apartado “**listen-on**” se le indica a Bind las IPs que debe escuchar.

La sección “**allow-transfer**”, es una opción de seguridad.

La opción de “**logging**” y “**zone**” es para la creación de zonas.
Con “**allow-query {…;}**;” indica que redes/hosts pueden consultar.

***Nota**: Bind por defecto viene configurado para que sólo funcione localmente, por si se instaló por error, (172.0.0.1, ::1, localhost).*

Modificar archivos de zona:
Una zona DNS es una colección de ordenadores relacionados, cuyas relaciones **nombre/IP** son gestionadas por un servidor autoritario.
En un servidor de sólo caché no habrá que realizar apenas configuraciones..
Un servidor de sólo reenvíos no requiere mantenimientos de zona.

Si se usa un sistema de búsqueda recurrentes completas, se debe de incluir una sección zona “**.**” para la zona raíz DNS.

La línea “**file**” de las sección “**zone**” se refiere a un archivo en el directorio especificado en la línea “**directory**”, normalmente en “**/var/named/**”.

***Nota**: las IPs de raíz no cambian, pero si por error borramos el archivo, lo podemos recuperar con “**dig @a.root-servers.net ns>db.cache**”.*

Comprobar cambios:
Reiniciamos con “**/etc/init.d/named reload o restart**(más agresivo)”. Una vez reiniciado podemos controlarlo con “**rndc**”:

- rndc **reload**: recarga los archivos de config.
- rndc **stop**: para el servidor.
- rndc **flush**: elimina caché del servidor.
- rndc **status**: muestra información.

Podemos usar “**host**” y “**dig**” para hacer pruebas.

Configurar un servidor como esclavo:
Se configura igual que uno maestro, pero en “**zone**” hay que poner “**slave**” en el apartado “**type**”.
Se pueden tener varios servidores maestros.

207.2 Crear y mantener zonas:

Las zonas deben de ser de tipo master, para cada zona se creará un archivo. Estos archivos tienen que tener permisos 644 pertenecer a root:bind.

Configurar archivos de zona:
Se encuentra en "**/var/named"**.

```
$TTL 1D
pangaea.edu.        IN   SOA   dns1.pangaea.edu. admin..pangaea.edu. (
                        2011022003  ;  serial
                        3600        ;  refresh
                        600         ;  retir
                        604800      ;  expire
                        86400       ;  default_ttl
dns1.pangaea.edu.               IN A 192.168.1.1
coelophysis.pangaea.edu.        IN A 192.168.1.1
peteinosaurus                   IN A 192.168.1.1
                                IN A 192.168.1.1
pangaea.edu.                    IN A 192.168.1.1
dns1.wgener.pangaea.edu.        IN A 192.168.1.1
www                             IN CNAME webhosting.example.com.
ftp                             IN CNAME plateosaurus
@                               IN MX 10 peteinosaurus
@                               IN MX 20 mail.example.com
@                               IN NS dns1.pangaea.edu.
wegener                         IN NS dns1.wegener.pangaea.edu.
```

La mayoría de las líneas tienen el formato "**nombre IN tipo_registro contenido_registro"**:

- **Nombre**: es el nombre del PC o la IP en caso de resolución inversa (el nombre de dominio lleva "**.**" final). Si el nombre está vacío, las líneas siguientes y la actual se asocian al anterior. El signo "**@**" representa el propio dominio.
- **IN**: significa Internet.
- **Tipo de registro**:
  - **A**: host IPv4, nombre-> IP.
  - **AAAA**: host IPv6, nombre-> IPv6.
  - **CNAME**: registro de nombre canónico (los subdominios).
  - **NS**: proporciona el nombre de host de un servidor DNS para el dominio.
  - **MX**: de intercambio de correo. Se pueden tener varios y el nº de prioridad organiza a quien enviar.
  - **TXT**: añade texto explicativo asociado al dominio.
  - **SOA**: inicio de autoridad.
  - **PTR**: IP a nombre, registro inverso.

- **Contenido registro**:
  - **nº de serie**: es una fecha de última modificación para que los servers locales copien la info del más actualizado.
  - **Tiempo de actualización**: indica cada cuanto tiempo debe comprobar el servidor maestro para actualizar.
  - **Tiempo de nuevo intento:** tiempo entre intentos de contactar con el servidor maestro si la conexión falla.
  - **Mínimo de tiempo de vida (TTL)**: tiempo mínimo que se almacenará una respuesta de error de subdominio no encontrado.

Configuración de zonas inversas:
En algunos servicios se realiza una consulta normal y una inversa, si los resultados no coinciden, la identidad queda bajo sospecha. E archivo de configuración es igual, con su registro SOA, pero los demás son PTR y NS. Las direcciones van sin el último octeto y en orden inverso.

Ej: 192.168.10/24 sería “1.168.192.in-addr.arpa.”, si fuese IPv6 sería “.ip6.arpa.”.

Host:
Puede revelar los registros A, CNAME y PTR del servidor.
Formato “**host [opcion] dominio/IP**”. **Opciones**:

- **-c**: muestra el registro SOA del dominio.
- **-d** o **-v**: vista detallada.
- **-t tipo de dato**: consulta el tipo de datos (SOA; CNAME, etc).
- **-4**: realiza búsqueda IPv4.
- **-6**: realiza búsqueda IPv6.

Nslookup: ya no se usa, es parecido a host.

Dig: **Formato**: “**dig [@servidor] [opciones] [nombre] [tipo]**”. Opciones:

- **-b dirección**: usa la direccioón indicada por si ese host tiene varias IPs.
- **-f archivo**: consulta host de ese archivo (1 por línea).
- **-p puerto**: por defecto es el 53.
- **-t tipo**: tipo de registro a buscar.
- **-q nombre**: nombre del host o dominio, suele omitirse.
- **-x dirección**: realiza búsqueda inversa.
- **-4**: busca sólo IPv4.
- **-6**: busca sólo IPv6.

Ejemplo: “dig @8.8.8.8 topgamespro.com”.

207.3 Securizar un servidor DNS

Asegurar transferencias de zona:
Mediante la directiva “**allow-transfers**” se pueden negar las transferencias de zona, para que los atacantes no obtengan toda la información interna.

Para negar todas las transferencias “**allow-transfer{“name”;};**”, pero si tenemos servidores esclavos, pondremos las IPs de ellos para que se puedan emitir los archivos de zona.

DNSSEC:
Aparte de asgurar las transferencias de zonas, se pueden usar extensiones de seguridad DNS, que defiende el servidor DNS del caché envenenado.
Esto evita que en la caché del servidor se inserten datos falsos de direcciones IPs de otros dominios, permitiendo que no haya redirecciones maliciosas.
Pasos:

- cd “/var/named”.
- “**dnssec-keygen -a RSASHA1 -b 768 -n ZONE nombreZona**”.

Esto creará una clave privada y una pública, que debe distribuirse a otros servidores. Con la clave privada generamos los archivos de zona:

“**dnssec-signzone -o dominio named.dominio**”.

Esto creará “**zona.dominio.signed**” y es el que se usará como archivo de zona, en lugar del original.

Jaula Chroot:
Ejecuta Bind en una jaula chroot, lo que evita que un servidor erróneo o vulnerable el resto del sistema.
La idea es ejecutar el programa haciéndole creer que el directorio raíz del pc "/" es distinto del que es en realidad. Entonces el servidor sólo puede acceder a los archivos de ese árbol de directorio alternativo.

Pasos:

- Comprobar que exista una entrada "**named**" en "**/etc/passwd**". Ejemplo: "named:x:200:200:Nameserver:/chroot/named:/bin/false".
- Preparar el directorio "**/chroot/named**":

```
mkdir -p /chroot/named
cd /chroot/named
mkdir -p dev /etc/namedb/slave /var/run"
cp -p /etc/named.conf /chroot/named/etc/
cp -a /var/named/* /chroot/named/etc/namedb/
cp /etc/localtime /chroot/named/etc/
chown -R named:named /chroot/named/etc/namedb/slave
chown named:named /chroot/named/var/run
Monod /chroot/named/dev/null c 1 3
mknod /chroot/named/dev/random c 1 8
chmod 666 /chroot/named/dev/ {null,random}
```

- Iniciamos named con la opción "**-t /chroot/named**".

Tema 208: Servicios Web

208.1 Implementando un servidor web (Apache)

El archivo de configuración principal es "**apache.conf o httpd.conf**", o en Apache 2.x **"apache2.conf o http2.conf"**. Normalmente se encuentran en "**/etc/apache/**", "**/etc/apache2**", "**/etc/httpd**" o en "**/etc/httpd/conf**". La configuración es la misma.

Está dividido en 3 secciones: parámetros globales, directivas de funcionamiento y host virtuales.

Los comentarios empiezan con **#** y las líneas con "**Directiva Valor**". La directiva es el nombre del parámetro que se quiere ajustar y el Valor es el valor del parámetro. Algunas Directivas aparecen entres paréntesis angulares "< >".

<IfDefine Variable>
Directiva valor
</IfDefine>

Apache es modular, muchas de sus funciones se pueden compilar como módulos separados que se pueden cargar en tiempo de ejecución. Para cargar un módulo se usa la directiva **"LoadModule".** En Apache 1.3.x hay que usar la directiva **"AddModule"** en el binario principal, pero en Apache 2.x no hace falta.

***Nota**: es recomendable comentar las Directivas "**LoadModule**" que no se usen.*

La directiva "**Include**" carga archivos adicionales como si fueran parte de los archivos de configuración principal. Se usa con los módulos de subdirectorios "**mods-available** y **mods-enabled**".

Algunas directivas:

- **<Directory>**: sus parámetros sólo se apricarán en el directorio y subdirectorios.
- **<Directory Math>**: como Directory pero en el nombre del directorio acepta expresiones regulares.
- **<Files>**: control de acceso de los ficheros por su nombre.
- **<Files Math>**: permite control de acceso a ficheros y uso de expresiones regulares.
- **<Location>**: control de acceso de los ficheros mediante html (no puede llevar AllowOverride).
- **Etc**.

Otros archivos de configuración:

- **access.conf:** le indica a Apache como tratar ciertos directorios, muchos sistemas los incluyen en el archivo de configuración principal.
- **mime.types o apache-mime.types:** define tipos de MIME, que son códigos que ayudan a identificar el tipo de archivo. Asocia las extensiones de los archivos a tipo MIME, como: .txt, .html, etc.
- **magic**: es un segundo método para determinar el tipo de MIME. En lugar de basarse en extensiones, se basa en "huellas digitales".

Todos los archivos de configuración residen en el mismo directorio principal de configuración de Apache.

Nota: *algunas opciones se pueden anular localmente mediante un "****.htaccess****".*

Establecer el usuario y grupo de Apache: Apache empieza ejecutándose como root y luego bajo el usuario que queramos. Las directivas son: **User** y **Group**. Una vez establecidas, podemos comprobarlo con "**ps aux | grep apache**" o "**ps aux | grep httpd**".
La primera instancia seguirá ejecutándose como root.

Cambiar las ubicaciones de las páginas web: el valor predeterminado es la directiva "**DirectoryIndex**", si se proporciona más de un valor para esta directiva, Apache las busca todas.
La directiva "**DocumentRoot**" le indica a Apache donde buscar las páginas web.
Para emitir las páginas desde otro sitio hay dos opciones: Cambiar la directiva "**DocumentRoot**" al nuevo directorio, en algunos sistemas se encuentra separado en "**sites-available/defaults**" o sustituir las creadas por otras.
Si queremos usar otro directorio, debe ser legible para el usuario bajo el que se ejecuta Apache.

Habilitar páginas web de usuario: usando la directiva "**UserDir**" que toma el nombre de un directorio del home del usuario como argumento. Puede aparecer en "**mods-available/userdir.conf**".

Ej: "**UserDir public_html**", una vez creado, los usuarios pueden crear subdirectorios llamados "**public_html**" y almacenar sus webs allí. Accederíamos como http://localhost/~usuario/web.html.
Estas directivas tienen que tener permiso de lectura y ejecución para Apache.

Servir dominios virtuales: se usa la directiva "**VirtualDocumentRoot**" para indicarle a Apache que directorio usar dependiendo del nombre del host. Hay que poner "**UseCanonicalNames Off**", para que funcione que por defecto está en On.
Funciona como "**DocumentRoot**", pero tiene variables:

- **%%:** un único % en el nombre del directorio.
- **%N.M**: http://www.aaa.com
  %1 %2 %3

Los números negativos cuentan desde el final %-1=%3, cuando es 0, es el nombre del host completo. M es el número de caracteres que tiene el nombre, ej: %1.2= ww.
Ejemplo: "**/home/httpd/com/example**" como directorio raíz de http://www.example.com, su configuración sería "**VirtualDocumentRoot /home/httpd/%-1/%-2**".

Nota*: si se quieren establecer dominios virtuales en base a la IP del server, se usa "****VirtualDocumentRootIP****" y se usan direcciones IP en lugar de nombres de host.*

Otro modo de usar dominios virtuales es usar "VirtualHost": la ventaja es que es más personalizable. También debe de estar "**UseCanonicalNames Off**".

```
<VirtualHost *>
        ServerName www.example.com
        DocumentRoot /home/httpd/pelirroja
</VirtualHost>
```

*indica todos los interfaces, si queremos a través de uno, se pondría.

Ejemplo de casa:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost

    DocumentRoot /media/Datos/ServidorWeb
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /media/Datos/ServidorWeb/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews
+SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Configurar Scripts: Para activar el soporte CGI en Apache se usa la directiva "**ScriptAlias**" apuntando a un directorio CGI parte del directorio padre:
"**ScriptAlias /cgi-bin /usr/www/cgi-bin**".

Activar PHP en Apache: instalarlo y descomentar las líneas:

```
#Use for PHP 5.x
LoadModule php5_module modules/libphp5.so
AddHandler php-script php

#Add index.php to your DirectoryIndex line:
DirectoryIndex index.html index.php
AddType text/html php
```

Puede ser necesario modificar "**/etc/php.ini**".

Instalar Perl: permite ejecutar script Perl directamente. Hay que instalar: **apache-mod-perl, libapache2-mod-perl2** y luego activar copiando el archivo de configuración "**perl.load**" o "**perl.conf**", si estos archivos no están, añadimos la siguiente línea a Apache "**LoadModule perl_module /usr/lib/apache2/modules/mod_perl.so**".

208.2 Mantenimiento de servidores web

Habilitar HTTPS: para ello necesitamos:

- **Instalar Apache con SSL:** hay que tener instalado **OpenSSL** y "**apache-mod_ssl**".
- **Tener un certificado:** lo normal es comprar uno a un CA, pero también podemos crear nuestro propio certificado (generaría alertas ante firmas desconocidas). Una vez lo tengamos, lo copiamos a "**/etc/ssl/apache**". Son dos archivos: un archivo de certificado ".crt" y una clave ".key" (deben de tener permiso 0600).

- **Configurar Apache para que escuche el puerto https, aceptando solicitudes de cifrado**: se necesita cargar el módulo SSL con "**LoadModule ssl_module /usr/lib/apache2/modules/mod_ssl.so**" (puede que sea distinto en otras distro). Directivas SSL:
  - **Listen**: para cambiar el puerto por defecto 443.
  - **SSLEngine off/on**: para el habilitar o no SSL.
  - **SSLRequireSSL**: usando esta opción sólo funciona https.

Limitar el acceso a Apache:

<IfModule mpm_prefork_module>
StartServers 5 *inicia con 5 servidores*
MinSpareServers 5 *Mantendrá entre 5*
MaxSpareServers 10 *y 10 servidores disponibles*
MaxClients 150 *nº máximo de instancias por servidor*
MaxRequestPerChild 0 *impone límites al nº de solicitudes (0 no tiene)*
</IfModule>

Opciones de autenticación de usuarios: para usarlo hay que cargar el módulo "**mod_auth**", en versiones posteriores a la 2.1 se usan módulos más especializados: "**mod_auth_basic**" (igual que <2.1), "**mod_auth_pam**" (usa sistema PAM) y "**mod_auth_ldap**" (usa ldap).
Para el uso de archivos de contraseña, generamos el archivo con "**htpasswd -c /ruta/archivo usuario**" (debe de estar en un directorio fuera de los directorios web) y tras ello pide contraseña para ese usuario.
Para que pida pass hay que configurar el archivo principal, o usar un "**.htaccess**" que se encuentra en el directorio que se quiere proteger:

AuthType Basic
AuthName "Archivos restringidos"
AuthBasicProvider File
AuthUserFile /etc/apache/passwd/passwords
Require user usuario

Podemos añadir más usuarios como antes, pero sin "**-c**".
Podemos usar **grupos de acceso**: creamos el archivo "**/etc/apache/passwd/group**". Con la siguiente línea "**GroupName: usuario1 usuario2 usuarioN**", cambiamos la línea "**Require user usuario**" por "**Requiere group NombreGrupo**" y añadir el archivo de grupos "**AuthGroupFile /ruta/.htgroup**".

En vez de crear grupos, podemos pedir usuarios válidos con la línea "**Require valid-user**".
Nota*: Para que sólo sea accesible desde esa red:*

Order deny, allow
Deny from all
Allow from 192.168.1.0/24

Controlar Apache: mediante la herramienta "**apache2ctl**", comandos:
- **start**: inicia Apache.
- **stop**: finaliza Apache.
- **graceful-stop**: para, pero permite que se completen las solicitudes que se están atendiendo.
- **graceful**: reinicia,pero permite que se completen las solicitudes que se están atendiendo.
- **restart**: reinicia.

- **fullstatus**: muestra info de todo, incluye la lista de solicitudes que se están atendiendo. Requiere que esté habilitado el módulo "**mod_status**".
- **status**: muestra info, pero no incluye la lista de solicitudes.
- **configtest**: comprueba el archivo de configuración.

Logs de Apache: Se encuentran en "**/var/log/apache2 o httpd**" y son:
- **error.log**: mensajes de error y notificaciones de arranque/fin.
- **access.log**: información de accesos a la web.
- **other_vhosts_access.log**: información del host virtual.

Se pueden examinar de forma manual o usar herramientas de análisis: webalizer, AWStat, etc.

Redirección de directorios: "**Redirect 301 /webmail http://....**", si se accede a webmail se redirecciona (301=redirección permanente).

208.3 Servidor Squid

El archivo de configuración se encuentra en "**/etc/squid/squid.conf**". Se usan tres comandos para configurar:
- **auth_param**: indica que mecanismo usar para autenticar usuarios: PAM, Samba, LDAP, etc.
- **ACL**: ejemplo "acl myacl proxy_auth REQUIRED", hay muchas acl.
- **http_access**: define las acl que usa la acl creada.
- **http_port nºpuerto (3128):** define el puerto de escucha.

***Nota**: las contraseñas de acceso a Squid se transmiten en texto plano.*

Ejemplos de acl:
- tipo src (dirección de origen):
  - acl redlocal src 192.168.1.0/24
  - acl colegas_VIP src 192.168.1.10 192.168.1.20
- tipo dst (destino): acl webmail dst www.gmail.com
- tipo dstdomain (da permisos sobre destinos): acl denegados dstdomain www.xx.com

Aplicarlas con: http_access allow colegas_VIP o por ejemplo http_access deny redlocal.

Tema 209: Compartición de archivos

209.1 Samba

El paquete se llama "**samba** o s**amba-server**" y maneja el protocolo SMB/CIFS.
El archivo de configuración se encuentra en "**/etc/samba/smb.conf**". Con un "**;**" es parámetro comentado.

La configuración de "**smb.conf**" se organiza en dos secciones:

- **[global]:** define los valores globales. No distingue mayúsculas, pero las rutas de Linux si. Opciones:
  - **workgroup = moninocasa**: nombre del dominio o grupo de trabajo.
  - **netbios name**: nombre del ordenador.
  - **printing**: sistema de impresión LPRng o CUPS.
  - **printcap name**: nombre del archivo "/etc/printcap" o en CUPS, necesario para compartir impresoras.
  - **load printers Yes/No**: indica si la impresora está disponible localmente cuando está compartida.
  - **host allow** y **hosts deny nombrehost/IP**: permite o deniega el acceso.
  - **security Share** (win9x/Me), User, Server, Domain, ADS o Security.
  - **encrypt passwords Yes/No**: encripta contraseñas.
  - **smb passwd file archivo**: archivo de contraseñas cifradas.
  - **password server IP/nombrehost**: indentifica al controlador de dominio.
  - **username mal archivo**: archivo que contiene asociaciones de nombres de usuario.
  - **name resolve order** lmhost host wins bcast: indica a samba como resolver nombres. Se usa cuando hay un servidor wins (antiguo DNS).
  - **server string=%h server (Samba, Ubuntu)**: nombre del servidor.
  - **interfaces=127.0.0.0/8 eth0**: es el interfaz por el que escuchará.
- **Recursos compartidos**: impresoras, carpetas, etc.

Establecer opciones de contraseñas:
Con la opción global de "**encrypt passwords**":

- **No**: Samba usa la base de datos de usuarios y contraseñas de Linux.
- **Yes**: Samba requiere su propia base de datos de contraseñas (independiente). Ejecutamos "**sudo smbpasswd -a usuario**" y pedirá contraseña, para borrar de la bd usamos "**smbpasswd -x usuario**". La base de datos se llama "**passdb.tdb**", hay que especificar el tipo de bd con "**passdb backend=tdbsam**" en "**/var/lib/samba**", la bd está cifrada.
  Si hay que añadir muchos usuarios se puede usar el script "**mksmbpasswd o .sh**", que meterá todos los usuarios de Linux en la bd de Samba, pero no es capaz de convertir las contraseñas (es algo inútil).

***Nota**: para usar lo anterior hay que tener "**security = User**", ya que como controlador de dominio es distinto.*

Opciones de grupos de trabajo o dominios:
El parámetro "**workgroup nombregrupotrabajo**" indica que es un grupo de trabajo, pero para usar un controlador de dominio necesitamos:

- Establecer el nombre del controlador de dominio con la opción "**password server**".
- Establecer "**encrypt passwords = Yes**"

- Establecer "**security**" en:
  - **Server**: no es miembro completo del dominio.
  - **Domain**: dominios antiguos.
  - **ADS**: dominio moderno.

Si usa seguridad **Domain** o **ADS**, hay que unirse al dominio con "**net join member -U usuarioadmindeldominio**".
Con esta seguridad, Samba no necesita bd de usuarios, pero debe manter una bd de cuentas convencionales de Linux, para ello existe: **Script local (**con la opción "adduser script" de Samba), **LDAP** y **Windbind**.

Asociar nombres de Linux y Windows:
En Linux son cortos y en Win largo y con espacios, por lo que hay que traducirlo. Hay que usar la opción "**username map nombrearchivo**" (normalmente **username.map**) para el mapeo.
Formato de username.map "**usuarioLinux = usuarioWin**".
Si tiene espacios se usarán comillas dobles. Se puede asignar un usuario de Linux a un grupo Win con "**usuarioLinux = @grupoWin**". Se puede usar * para asignar nombres desconocidos.

Configurar archivos compartidos:
Comienza con el nombre entre corchetes y luego las opciones:
- **comment = comentario**: descripción del recurso compartido.
- **path o directory rutaacompartir**: lo que compartimos.
- **browseable = Yes/No**: indica si el recurso aparece en los navegadores, por defecto Yes.
- **writeable = Yes/No**: indica si se puede escribir en el recurso, por defecto No.
- **create mask = máscara**: permisos de los archivos creados por los clientes.
- **directory mask = máscara**: premisos para nuevos directorios.
- **nt acl suppport = Yes/No**: indica si asocia los permisos de archivo de Linux a una ACL estilo Win. Por defecto Yes.
- **force user usuario**: usuario que se asignará a todos, una vez conectados (usuario con el que se ejecutarán todas las tareas), también se puede forzar grupo.
- **available = Yes/No**: habilita/deshabilita ese archivo compartido.
- **valid users = usuario1, usuario2**...: indica los usuarios autorizados.
- **veto files=/*Security*/*.tmp**: no deja acceder ni son visibles para los clientes.

Existe un archivo compartido especial **[homes]** que muestra el home del usuario registrado, por defecto no es browseable.

***Nota**: con un **$** al final del compartido se oculta, por ejemplo **[Datos1$]**.*

Ejemplo de casa:

[Datos]
path=/media/Datos
writeable=yes
browseable=yes
valid users=monino

Configurar impresoras compartidas:
Es igual que los archivos compartidos, excepto la opción "**printeable = Yes/No**", que le indica a Samba que trate el recurso como una impresora.

El envió de archivo se hace a una cola de impresión del mismo nombre que la impresora compartida, pero lo podemos cambiar con "**printer=nombreimpresora**".
En muchas configuraciones se comparte una impresora global, en vez de una a una **[printers]**, la opción global "load printers" debe de estar en No.
La opción "**printing = BSD/CUPS/Etc**" indica el método.

Testeo y ejecución:
Tras modificar smb.conf, podemos testear la nueva configuración con "**testparm**", que **escanea smb.conf y marcará los errores.** Si usamos "**testparm -v**" se muestran los **valores que no se modificaron** por smb.conf.

Samba consiste en dos programas servidor:

- **smbd**: que maneja las taeas de intercambios de archivo.
- **nmbd**: que maneja las funciones de "unión ocultos de NetBios", como resolución de nombres.

Podemos usar la herramienta "**nmbloockup**" para consultar servidores sobre us nombres y funciones NetBios relacionadas, sirve también para localizar grupos de trabajo o nombres de dominio local.
La herramienta "**smbstatus**" indica el estado del servidor Samba, los clientes conectados y los archivos abiertos actualmente.

Archivos de registro de Samba:
Se encuentran en "**/var/log/samba/**":

- **log.nmbd** y l**og.smbd**: contienen información de los servidores.
- **log.nombreequipo**, **log.IPequipo**: contiene información de los intentos de conexión de esos clientes.

Uso de Linux como cliente SMB/CIFS:
El programa cliente se llama "smbcliente", con formato:
"**smbcliente [//servidor//recurso][contraseña][opciones]**", en Win las barras son "\\servidor".

Opciones:

- **-I IP**: se conecta por IP.
- **-L HOST**: lista los recursos compartidos por el Host.
- **-s nombrearchivo**: usa nombrearchivo como configuración, en lugar de smb.conf.
- **-N**: elimina la solicitud de contraseña, si el recurso lo requiere, la conexión fallará.
- **-A** nombre archivo: obtiene el nombre de usuario y contraseña del archivo especificado.
- **-U nombredeusuario [%contraseña]**: conecta con ese usuario y pass.
- **-n nombre**: pasarle nombre como el nombre NetBIOS.
- **-W grupotrabajo**: especifica el grupo de trabajo o dominio.
- **- c comandos smbcliente**: usa ls, cd, rename, get, put... (separados por comas).

También podemos **montar recursos compartidos**, para ello: "**mount -t cifs o smbfs //servidor/recurso /rutade/montaje**" (smbfs fue eliminado del kernel > 2.6.37.

Opciones de montado cifs (hay que usar **-o** y luego las opciones):

- **user = nombre**
- **password = pass.**
- **credential = nombrearchivo**: obtiene el usuario y pass del archivo especificado.

Se pueden **montar los recursos al inicio** del sistema modificando **fstab**, ejemplo: "//SERVICES/Datos /mnt/Datos cifs credentials=/etc/samba/arch 0 0".

***Nota**: para ocultar los archivos que empiezan por "." (ocultos de Linux), escribimos en el recurso compartido la opción "hide dot file=Yes".*

***Nota**: podemos denegar el acceso a ciertos archivos filtrando por extensión o por cadenas, usando "**veto files**". Ejemplo: "veto files=/*Security*/*.tmp/", que deniega lo que contenga Security y /o lo que termine en .tmp. Otro ejemplo: "veto files=/*.mp3/*.avi/", impide esa extensiones.*

***Nota**: se pueden añadir antivirus a samba con "**Samba-vscan**".*

***Nota**: podemos evitar el borrado de archivos de un compartido añadiendo una papelera de reciclaje con:*

> ***"vfs objects=recycle***
> ***recycle:repository=Nombrepapelera"***
> ***Se creará la carpeta NombrePapelera, si se borra de esta ya es permanente.***
> ***Otras opciones: "recycle:minsize=10" (en bytes), "recycle:maxize=5120"***
> ***y "recycle:keeptree" que guarda copias aunque coincida el nombre.***

209.2 NFS

NFS respeta los permisos de Linux. Hay clientes NFS para Windows, pero es mejor usar Samba.
La implementación de NFS se basa en varios programas RPC:

- **rpc.idmapd**: en versiones > NFSv3, asocia nombres de usuario y UID entre sistemas. Importante si dos pcs incluyen los mismos usuarios, pero tienen UID diferentes.
- **rpc.mound**: administra las peticiones de montado de clientes NFS y controla quienes están conectados. En ocasiones se denomina "rpc.mountd".
- **rpc.nfsd**: realiza la mayor parte del trabajo.
- **rpc.statd**: indica a los clientes cuando está a punto de volver a arrancar el servidor NFS.
- **portmap**: conocido como "portmapper", indica a los clientes que puerto usar.

El archivo "**/etc/exports**" contiene los directorios que exporta un servidor NFS, con el formato: "**/directorio cliente(opciones) [cliente(opciones)]...**".
Los clientes se pueden añadir por: nombre del host, comodines, IPs, direcciones de red o grupos NIS (sin uso ya).
Opciones de clientes:

- **secure** o **insecure**: especifica que el cliente debe conectar secure (con un puerto < 1024), por defecto es secure.
- **rw** o **ro:** acceso rw o sólo ro. Predeterminado ro, aunque en algunos es rw, por lo que debe especificarse.
- **sync** o **async**: async mejora el funcionamiento a costo del riesgo de corrupción de datos en caso de falla del sistema.
- **hide** o **nohide**: cuando se usa hide, se oculta subdirectorios montados en otros lados. Lo mejor es exportar a NFS cada partición y montarla como NFS.
- **root-squash** o **no_root_squash**: el acceso a root desde el cliente se transforma sustituyéndose su ID por la de root. Con no_root_squash se es root al 100% (el root remoto es considerado como local, con mismo UID y GUID).
- **all_squash** o **no_all_squash**: especifica si transforma los accesos de usuarios normales. (Con all_squash cambia todos los usuarios al usuario anónimo).

- **acl** o **no_acl**: con acl aplica reglas de acceso. Usar no_acl requiere un Kernel parcheado.
- **fsid=[num] root|uuid**: cambia el uuid(nº de serie del sistema de archivos) por uno corto para no compartir el uuid real.

Ejemplo de "/etc/exports": /home monino(w,insecure) pepito (ro).

Modificar un servidor NFS en ejecución:
Se pueden hacer cambios temporales en las exportaciones usando "exportfs":

- **sin opciones**: muestra la lista de exportaciones activas.
- **-a**: lee "/etc/exports" y exporta todos los directorios listados allí.
- **-u**: como -a pero elimina.
- **-r**: vuelve a exportar, pero elimina la exportación de los directorios de "/etc/exports".
- **-f**: vacía y vuelve a crear la tabla de exportaciones.
- **-v**: incluye mensajes detallados en la salida del programa.

Cuando se usa "**-u**" o "**-a**", hay que especificar un cliente y directorio con el siguiente formato: "cliente:/exportación/directorio".
Ejemplo de quitar un directorio para una red "exportfs -u 192.168.1.0/24:/home/monino".
Ejemplo de exportar "exportfs 192.168.1.0/24:/var/apache".

Todo es temporal y habrá que editar "**/etc/exports**".

Identificar exportaciones montadas:
Con la herramienta "**showmount**" se muestra las actividades actuales de nfs:

- **sin opciones**: revela las direcciones IPs de los pcs que están usando el servidor.
- **-a**: muestra las direcciones IPs de los clientes y los directorios en uso.
- **-d**: muestra los directorios actualmente compartidos.
- **-e**: igual que sin argumentos, pero en una sola línea, lista exportaciones montadas.
- **-h**: ayuda.
- **-v**: versión.
- **-- no-headers**: elimina la salida de cabeceras explicativas.

Por defecto muestra info del pc local, pero si se añade un host, muestra de ese host. Ejemplo: "showmount -a MoninoPC".

Medir la actividad de nfs:
Con "**nfsstat**" se proporciona estadísticas del kernel NFS. Muestra la info de cliente y servidor, por lo que puede usarse para consultar desde server o cliente para ver info (es distinta la información que se muestra):

- **-s**: muestra info sólo del server.
- **-c**: muestra info sólo del cliente.
- **-n**: muestra sólo estadísticas NFS.
- **-r**: muestra sólo estadísticas RPC.
- **2/3/4**: muestra estadísticas de la sesión NFS especificada.
- **-m**: muestra info sobre exportaciones NFS montadas.
- **-o**: muestra info sobre la instalación especificada: nfs, rpc, net, fh, rc o all.
- **-z**: toma una instantánea de estadisticas y las compara con otra instantánea al hacer SIGINT (Control-C).

Comprobar la actividad de RPC:
Con "**rpcinfo**" proporciona acceso a datos RPC. Se usa para identificar los servidores con RPC habilitados en una red. Opciones

- **-p host**: explora el host.
- **-u programa host**: usando UDP ejecuta el procedimiento e informa de los resultados.
- **-t programa host**: como "-u" pero en TCP.
- **-n numpuerto**: complementa a "**-u**" y "**-t**" indicando el número de puerto a usar.
- **-b versión(2,3,4) programa**: envía una solicitud de multidifusión para que se ejecute el procedimiento.

Administrar un servidor NFS:

- **Usar Linux como cliente**: se pueden usar con "mount" o como entrada en /etc/fstab".
  - Por **mount**: "mount -t nfs Ipserver:/directoriocompartido /mnt/dondemonto".
  - Por **fstab** "Ipserver o NombreServer:/directorioCompartido /mnt/dondemonto nfs defaults 0 0".

  Para ver lo que exporta un servidor "showmount -e nombreServer o IPServer".
- **Portmap**: este mapeador se encarga de gestionar la asignación entre aplicaciones y puertos. Lo normal es que el portmap se encuentre ya instalado y ejecutándose, comprobar con "**ps aux | grep portmap**". Usa el puerto 111 en TCP o UDP.
- **TCP Wrappers**: en "/etc/host.allow" y "/etc/host.deny" hay que especificar las Ips o rangos que pueden acceder al servicio Portmap de la siguiente manera: "portmap : 192.168.1.0/255.255.255.0".
  ***Nota**: se pone los que si pueden y en el deny se deniegan todos con "portmap :ALL", es necesario para que arranque nfs.*

Tema 210: Administrar clientes de red

210.1 Configuración DHCP

El paquete se llama "**dhcp, dhcp-server, dhcp3-server** o **dhcp4-server**". El **cliente** dhcp se llama "**dhcpd**" y en otros sistemas "**dhclient**". A parte de manejar DHCP, es capaz de manejar "**Bootstrap (BootP)**".
El archivo de **configuración** del servidor DHCP es "**/etc/dhcp3o4/dhcpd.conf**".

***Nota**: Windows necesita que las respuestas DHCP se manden a la máscara y Linux las envía a la broadcast. Si hay errores con ello se pueden cambiar el comportamiento con la máscara, ejemplo para /24: "route add-host 255.255.255.0 dev adaptador".*

Opciones globales de **dhcpd.conf**:

- **default-lease-time tiempo**: tiempo de vida de la concesión IP en segundos. Normalmente entre 2 horas y 2 días.
- **max-lease-time tiempo**: tiempo máximo de cesión IP, aunque el cliente solicite más tiempo, el servidor no se lo da.
- **min-lease-time tiempo**: tiempo mínimo de cesión IP.
- **get-lease-hostnames true/false**: el server busca el nombre de host asociado con la IP y lo devuelve al cliente.
- **Use-host-decl-names true/false**: almacena el nombre del host al darle IP.
- **Ping-check true/false**: si es true el server hace ping a una dirección antes de asignarla.
- **option subnet-mask mascaraentera**: establece la máscara que asignará.
- **option routers IP**: la dirección o direcciones para el router o routers de la red.
- **option domain-name-servers direccionesIP**: dirección o direcciones de los servidores DNS que se enviarán.
- **option domain-name nombreDominio**: nombre de dominio en el que reside.
- **option netbios-name-servers direcciones IPs**: IPs de los servidores WINS, es importante para SAMBA en redes Windows.
- **option netbios-node-type código binario**:
  - **1**: multidifusión.
  - **2**: servidor WINS.
  - **4**: 1+2.
  - **8**: servidor WINS con el parámetro options netbios-name-servers.

Configurar el envío de direcciones dinámicas:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
 range 192.168.1.100 192.168.1.254;
}
```

Configurar envío de direcciones fijas:
Conseguir la MAC del host al que se le enviará la IP fija y añadir entradas por host.

```
host OrdeGordo{
    hardware Ethernet 00:11:22:33:44:55;
    fixed-address 192.168.1.3;
} # el hw puede ser Ethernet, Token-ring, etc
```

***Nota:** ISC DHCP no puede configurarse para dar sólo Ips a los clientes conocidos.*

Configuración de un agente relay DHCP:
Si la red abarca varios segmento de red, podemos:

- Ejecutar varios servidores DHCP, uno por subred.
- Ejecutar el servidor DHCP en el router.
- Configurar el router para que envíe multidifusiones DHCP.
- Ejecutar un agente relay de DHCP, que es un programa que permite transmitir multidifusiones DHCP de una subred a otra. Este programa debe instalarse en un ordenador de cada subred que no tenga su propio servidor DHCP.

Con la opción de agente relay, hace falta instalar "**dhcrelay**".
Se usa la IP del servidor dhcp de manera remota "**dhcrelay IPDNSServer**".

Con la opción "**-i interfaz**" **indicamos** porque **interfaz** escuchará.
Para que al iniciar ya retransmita, modificamos "**/etc/default"dhcp3-relay**" y en la opción de SERVERS se indica la IP.

Logs: Se encuentra en "**/var/log/daemon.log**" y "**/var/log/messages**".

DHCP.Leases:
Como el servidor DHCP puede pararse y reiniciarse, necesita mantener la lista de direcciones asignadas. El fichero "**/var/lib/dhcp/dhcpd.leases"** o "**/var/state/dhcp/dhcpd.leases**" mantiene esta lista de asignaciones. Cuando se inicia el servidor, primero lee el fichero de configuración **dhcpd.conf**, después el fichero **dhcpd.leases** y marca qué sistemas tienen asignaciones activas.

210.2 Autenticación PAM

Permite cambiar el modo en el que Linux autentica a los usuarios, además de "**/etc/shadow**" y "**/etc/passwd**" se puede ampliar por un servidor de contraseñas, hardware biométrico, etc. PAM migró las contraseñas de "**/etc/passwd**" a "**/etc/shadow**".
El directorio principal de configuración es "**/etc/"pam.d**", que contiene archivos de configuración de todos los programas que usan PAM. Los archivos de configuración usan el nombre de la herramienta que los invoca, por ejemplo "**/etc/pam.d"gdm**" para GDM, "**/etc/pam.d/login**" para acceso al login de texto.

Formato de archivos **PAM**: "**grupo_gestión mrca_control módulo [opciones]**". Estos campos significan:

- **grupo_gestión**: hay cuatro grupos y cada uno controla un de la autenticación (auth-> autenticación, account-> gestión de la cuenta, session-> gestión de sesión y password-> gestión de las contraseñas).
- **marca_control**: cuando se llama a un módulo, este puede fallar o no (como una contraseña correcta o no). Este campo indica a PAM como reaccionar a este éxito o fracaso. Hay cuatro marcas:
  - **requisite**: OK continua ejecutándose, KO deja de ejecutarse.
  - **required**: OK continua, KO continua.
  - **sufficiente**: OK deja y tiene éxito, KO continua.
  - **optional**: OK continua, KO continua.
- **Módulo**: indentifica el archivo de módulo PAM. Incluye la ruta completa o relativa, normalmente es "/lib/security" (dónde se almacenan los módulos PAM).
- **Opciones**: pues eso mismo.

Editar una pila PAM:
Para modificar una configuración PAM, se debe editar una o más pilas PAM, que son grupos de módulos que se llaman para realizar tareas específicas. Cada archivo de configuración en "**/etc/pam.d**", consiste en una o más pilas PAM.
Cada grupo de gestión tiene su propia pila.

Módulos:
Algunos módulos vienen con el paquete PAM y otros con paquetes auxiliares, como LDAP. Con independencia de su origen, se pueden añadir a la pila PAM para cambiar la forma en la que opera.

Módulos comunes:

- **pam_unix**: pertenece a los 4 grupos, realiza la autenticación tradicional de Linux en base a "**passwd** y **shadow**".
- **pam_cracklib**: grupo password, comprueba la fuerza de una contraseña cuando se cambia.
- **pam_limits**: grupo sesion, establece un límite de sesión de acceso en la memoria, tiempo de CPU y otros recursos del sistema. Si no se indica archivo de configuración, por defecto será "**/etc/security/limits.conf**".
- **pam_listfile**: pertenece a los 4 grupos, busca reglas que permitan o denieguen el acceso en el archivo especificado.

Ajustar Name Service Switch:
Proporciona listas de usuarios y grupos, asocia números de ID de usuario (UID), identifica los directorios home de usuarios, etc.

El archivo de configuración se encuentra en "**/etc/nsswitch.conf"**.

Si añadimos LDAP para autenticación hay que ajustar la configuración de NSS, de la siguiente manera:

passwd: compat ldap
group: compat ldap
shadow: compat ldap

210.3 Uso del cliente LDAP

Hay que instalar el cliente de OpenLDAP y editar "/etc/openldap/ldap.conf", que especifica la base de datos de directorio LDAP, servidor, clave, etc. Ejemplo:

BASE dc=pangaea, dc=edu
URI ldaps://ldap.pangaea.edu
TLS_CACERT /etc/openldap/ssl/certs/slapd-cert.crt"

LDAP usa **LDIF** (LDAP Data Interface Format) como método de transferencia de datos. Consiste en un archivo con una serie de nombres y valores de atributos separados por dos puntos. Estos archivos deben contener información equivalente a las entradas en "**/etc/passdw**" y "**/etc/shadow**".

Añadir cuentas:
Ejemplo: "**ldapadd** – D cn=manager, dc=pangaea, dc=edu -W -f acct.ldif".

Opciones de ldapadd:

- **-c**: continua procesando aunque se detecten errores.
- **-S archivo**: el log de errores se escribe en ese archivo.
- **-n**: no modofica el directorio, pero muestra lo que haría.
- **-v**: salida detallada.
- **-d** nivel: nivel numérico de depuración.
- **-f archivo**: lee registros de ese archivo.
- **-x**: usa un método sencillo de autenticación, en lugar de SASL.
- **-D usuario**: se une al directorio usando el nombre distinguido especificado.
- **-W**: solicita autenticación.
- **-w contraseña**: usa esa contraseña (no recomendable).
- **-y archivodecontraseña**: lee la contraseña desde el archivo.
- **-H ldapuri**: accede a través de URI.
- **-P versión 2/3**: usa esa versión.
- **-Z**: intenta usar StartTLS.

***Nota**: hay scripts para migrar desde passwd a formato LDIF, pero no shadow, por lo que no tiene mucha utilidad.*

Modificar cuentas:
Se usa "**ldapmodify**" que contiene las mismas opciones que "ldapadd". Para cambiar la contraseña se usa "**slappasswd**" con las siguientes opciones:

- **-v**: salida detallada.
- **-s contraseña**: crea un encriptado de la contraseña especificada.
- **-T archivo**: crea un encriptado de los contenidos del archivo.
- **-h esquema**: usa el esquema especificado para encriptar. Puede ser: CRYPT, MD5, SMD5, SSHA Y SHA. El predeterminado es SSHA.

Podemos cambiar la pass directamente usando "**ldappasswd**" (combina ldapmodify y slappasswd), ejemplo: "ldapasswd -D cn=manager, dc=pangaea, dc=edu -W -S uid=maryam, ou=People, dc= pangaea..."

Borrar cuentas:
Usamos "**ldapdelete**" con las mismas opciones que "**ldapadd**".

Consultas sobre cuentas ldap:
Con el comando "**getent**" consultamos. Si añadimos "**-s ldap**" realiza la búsqueda sobre ldap.

Con la herramienta "**ldapsearch**" podemos buscae empleando cualquier campo de la base de datos. Ejemplo "ldapsearch uid=monino".

Tema 211: Servicios de correo

211.1 Usando servidores de correo

Configurar un dominio para que acepte correo:
Un dominio requiere una entrada DNS especial. Conocida como registro MX. Ejemplo "@ IN MX smtp.pangaea.edu".

Servidores de correo:
- **Sendmail**: fue el más importante. Configuración difícil.
- **Postfix**: tan popular como Sendmail, configuración media. Modular. Buena seguridad.
- **Exim**: diseño monolítico, configuración sencilla.
- **Qmail**: no permite la distribución de binario. Es modular. Poco popular.

Postfix:
Fácil de configurar, archivo de configuración "**/etc/postfix/main.cf**". El nombre del host lo adquiere automáticamente. Opciones:
- **myhostname nombrehost**: lo que devuelve el comando hostname.
- **mydomain $myhostname**: nombre de dominio, define el dominio para que Postfix entregue correo localmente.
- **mydestination**: indica la máquina local donde enviar.
- **masquerade_domains**: dominios a los que se le reduce la dirección (como quitar subdominios).
- **masquerade_classes envelope_sender/header_sender**: tipo de direcciones afectadas por masquerade_domains.
- **masquerade_exceptions**: nombres de usuario que no se enmascararán.
- **sender_canonical_map**: cambia la dirección del remitente usando una base de datos de búsqueda flexible.
- **sender_canonical**: cambia la dirección del remitente para emails salientes.

Aceptar correo entrante: Normalmente acepta correo local dirigido a "**$myhostname**" o "**localhost.$myhostname**". Se puede ampliar o reducir el rango de direcciones aceptadas cambiando el parámetro "**mydestination**". Ejemplo: "$mydestination=$myhostname, localhost, $mydomain". Se pueden añadir más dominios.

Otro parámetro a cambiar es "**inet_interfaces**" que establece desde que adaptador se escucha (si hay varios y se quiere escuchar desde todos, se pone "**all**").

Configuración de transmisión: La configuración de transmisión se construye sobre la base de la confianza, el servidor transmite el correo de las máquinas en las que confía. Opciones:
- **mynetworks_style host/subnet/class**: subnet significa la misma subred donde está el servidor, class la misma clase que la dirección IP del servidor y host implica confiar sólo en el propio pc.
- **mynetworks listadoderedes**: lista de redes confiables IP/Máscara.
- **relay_domains $mydestination**: máquinas y dominios listados de forma explícita por nombre.

***Nota**: el demonio puede enjaularse con el flag chroot en "**master.cf**".*

Sendmail:
El archivo de configuración principal es "**/etc/mail/sendmail.cf**" (largo y difícil de entender), no se edita, es mejor usar un archivo de configuración, que se usará para generar este "sendmail.cf". Este archivo de configuración está en lenguace "procesador de macros m4", para editarlo hay que instalar el lenguaje de macros mv (/usr/bin/m4) y los archivos de configuración de sendmail m4 (**paquete sendmail-cf)**.
El nombre de este archivo de configuración varía "**sendmail.mc, linux.smtp.mc, etc**" y podría residir en "**/etc/mail/, /usr/share/sendmail/cf, etc**".

Para **realizar cambios en sendmail**:

- Realizar **copia de seguridad** de "**/etc/mail/sendmail.cf**".
- Ir al direcotiro del archivo de config m4 y crear copia.
- **Editar el archivo m4**.
- Crear el nuevo sendmail.cf ejecutando "**m4 archivom4.mc /etc/mail/sendmail.cf**".

Otros archivos de config:

- **/etc/mail/access.db**: controla el acceso al servidor sendmail. Es una base de datos de binarios creada a partir del archivo de texto plano "**access**" usando el programa "**makemap**".
- **/etc/mail/aliases.db**: (puede estar en /etc) es una base bd creada a partir del archivo "**aliases**" empleando "**new aliases**".

Cambiar el nombre de host en Sendmail: añadimos al m4

MASQUERADE_AS (`dirección destino')
FEATURE (masquerade_envelope)
Nota: *1ª comilla es tilde inversa, la 2º es normal.*

Configurar para aceptar correo entrante: editar la línea

DAEMON_OPTIONS (`Port=smtp, Addr=127.0.0.1, Name=MTA') dn1
Nota: *1ª comilla es tilde inversa, la 2º es normal.*

Con **dn1** y un **espacio al principio** de la línea, esta se convierte en **comentario**. Una vez comentado, añadir una línea con "**FEATURE (use_cw_file)**", luego editar "**/etc/mail/local-host-names**" y añadir los nombres que sendmail quiere que reconozca como locales, ejemplo: pangaea.edu, mail.paganea.edu.

Configurar Sendmail para que transmita correo: en el m4 ponemos "**FEATURE (`access.db**')", editar "**/etc/mail/access**" y añadir la red o IPs seguido de:

- **OK**: acepta correo aunque otra regla lo rechace.
- **RELAY**: transmisión bidireccional, por defecto.
- **REJECT**: bloquea lo que venga de resa red, genera mensaje de devolución.
- **DISCARD**: como REJECT pero sin mensaje.
- **ERROR nº texto**: como REJECT, pero el mensaje lo proporcionamos nosotros

Luego hay que **convertir** "**/etc/mail/access**" en **binario con**: "**makemap hash /etc/mail/access.db < /etc/mail/access**".

Probar un SMTP:

telnet localhost 25
HELO localhost
MAIL FROM: <usuario@pangaea.com>
RCPT TO: <destino@pangaea.edu>
DATA es unmensaje de prueba
QUIT

Comprobar cola de correo:
Con el programa "**mailq**" se gestiona la cola y se mostrará los pendientes. Con "**sendmail -q**" o "**postqueue**" en Postfix limpiamos la cola. Esta cola se almacena en "**/var/spool/mail**".

Alias:
Modificando "**/etc/aliases**" podemos redireccionar, por ejemplo "**monino:postmaster**". Luego hay que recompilar el archivo con "**newaliases**".

Logs:
Se encuentra en "**/var/log/mail.log**" o "**/var/log/mail.err**".

Majordomo:
Es una aplicación para manejar listas, el programa corre cada vez que un correo llega a majordomo@hostname. Debe crearse el usuario majordomo, el archivo de configuración es "**.../majordomo/majordomo.cf**".

211.2 Manejo de colas

Para manejar las colas de mail más eficientemente se usa "**Procmail**".

Almacenamiento de correo:

- **Mbox**: usado por Sendmail, Postfix y Exim, los mensajes se almacenan en un sólo archivo, normalmente en "**/var/spool/mail/usuario**". Debería ser el más lento, pero es rápido. Acceso secuencial.
- **Maildir**: usado por Qmail, un directorio a cada carpeta de correo y coloca cada mensaje en su propio archivo. Maildir no se usa, Qmail se suele configurar para ser usado como Mbox.

***Nota**: Si se usa **IMAP** es recomendable usar **Maildir**.*

Reglas de Procmail:
El archivo de configuración es "**/etc/procmailrc**" y el archivo "**~/.procmailrc**" de los directorios home de los usuarios. Las fórmulas de Procmail tienen el siguiente formato:

:0[Marcas].[:[lockfile]]
[condiciones]
acción

Marcas:

- **-H**: se realiza la concordancia en la cabecera del mensaje, predeterminado.
- **-B**: concordancia en el cuerpo del mensaje.
- **-D**: concordancia distinguiendo entre mayúsculas y minúsculas.
- **-C**: la concordancia se realiza en una copia del mensaje.
- **-w**: espera a que se complete la acción, si no tiene éxito se hace coincidir con formulas posteriores.
- **-W**: igual que w, pero se suprime los mensajes de fallo del programa.

Condiciones: son expresiones regulares que comienzan siempre con un asterisco:

- ^: denota inicio de línea.
- **.**: coincide con cualquier carácter excepto una nueva línea.
- *****: denota una secuencia de cualquier longitud.
- |: comparación.
- \: deshace el formato especial del siguiente carácter.
- **!**: invierte el sentido de la comparación.

Acciones:

- **Una referencia de nombre de archivo**: Porcmail almacena en Mbox, para que almacene en Maildir hay que añadir "/" al final del nombre de archivo.
- **Un programa externo**: si la línea acción comienza con una barra vertical "|", Procmail trata esa línea como un programa a ejecutar.
- **Una dirección de correo**: una exclamación "**!**" al principio de la línea denota una dirección de correo electrónico, lo mandaría allí.
- **Un bloque de anidamiento**: comienza con llave de apertura "**{**" y denota una fórmula anidada. Se finaliza con "**}**".

211.3 Configuración POP e IMAP

Son del tipo **PULL**, hay varios servidores: UW IMAP, CyrusIMAP, Courier y Dovecot.

Configuración de Courier:
Archivos de configuración en "**/etc/courier**", el archivo "**authdaemonrc**" controla el demonio de autenticación e "**imapd**" controla los parámetros de servicio.

Parámetros de Courier IMAP:

- **ADDRESS**: establece la IP que escucha el servidor, si se pone "0", escucha todas.
- **PORT**: el nº de puerto, por defecto 143.
- **MAXDAEMONS**: limita el nº de demonios y con ello las conexiones simultáneas.
- **IMAP_CAPABILITY**: capacidad del server, no se suele modificar.
- **MAILDIRPATH**: nombre del directorio en el que se almacenarán los correos.

Configurar Dovecot:
Archivo de configuración en "**/etc/dovecot/doveco.conf**". **Parámetros**:

- **protocols**: indica los protocolos (imap, imaps, pop3, pop3s).
- **listen**: IP de escucha y si el nº de puerto. Si ponemos "*" escucha IPv4 y "::" IPv6.
- **login_process_per_conection Yes/No**: si cada registro inicia sus propios procesos. Por defecto Yes.
- **login_max_processes_count**: nº máximo de procesos de acceso Dovecot. Si login_process_per_conection está a Yes.
- **login_max_conections**: nº máximo de conexiones, si login_process_per_conection está a No.
- **mail_location**: ruta de los archivos Mbox. Usa variables:
  - **%u**: nombre de usuario.
  - **%d**: parte del dominio del correo.
  - **%h**: directorio home.

Tema 212: Seguridad del sistema

212.1 Configurar un router

Para usar el pc como router hay que configurarlo para que use dos o más interfaces de red.. Luego hay que vincular los interfaces de red escribiendo:
"**echo "1" > /proc/sys/net/ipv4/ip_forward**".

IPtables:
Hay 3 tipos de cadenas dentro de la tabla "**filter**":

- **INPUT**: paquetes destinados a programas locales.
- **FORWARD**: paquetes que el sistema va a emitir.
- **OUTPUT**: paquetes originados localmente y destinados a sistemas externos.

La tabla "**nat**" gestiona el natting y la tabla "**mangle**" modifica paquetes de forma especializada.
Para guardar la configuración actual de iptables en un archivo, usamos "**iptables-save > archivo**" y para restaurarla "**iptables-restore < archivo**".

***Nota**: iptables lee en forma secuencial.*

Opciones de IPtables:

- **tabla**: especifica la tabla sobre la que se va a operar, normalmente filter o nat.
- **-L [cadena]**: muestra la config de todos las cadenas o de la seleccionada.
- **-S [cadena]**: como -L pero más concisa.
- **-F [cadena]**: elimina todas las reglas o la indicada.
- **-Pcadena destino**: establece política para la cadena especificada.
- **-A cadena regla**: añade una nueva regla a la cadena.
- **-D cadena regla**: borra una regla de una cadena, indicada por el nº o descripción.
- **-I cadena [nº] regla**: inserta una nueva regla en una cadena, en la posición especificada (nº), si se omite el nº, la regla se inserta en la cabeza de la cadena.
- **-R cadena nº regla**: reemplaza la regla especificada en ese nº.

Reglas predeterminadas: hay 3 opciones y sólo puede predeterminarse una:

- **ACCEPT**: acepta el paquete, por ejemplo, si la política predeterminada de input es ACCEPT, entraría todo.
- **DROP**: no deja pasar.
- **REJECT**: como DROP, pero devuelve un código de rechazo a esa aplicación.

Con "**iptables -L**" se muestra la política predeterminada (la que actualmente está en uso). Para cambiar la política predeterminada, primero hay que borrar todas las reglas "**iptables -F**" y luego "**iptables -P política_predeterminada**".

Por seguridad se suene denegar todo con "**iptables -t filter -P INPUT DROP**".

Uso de IPtables: Para añadir una regla a una cadena:

Iptables -A CADENA selección_criterios -j OBJETIVOS

INPUT	***ACCEPT***
OUTPUT	***DROP***
FORWARD	***REJECT***

Criterios:

- **-m nombre**: añade reglas de coincidencia por el módulo nombrado.
- **-p protocolo**: tcp, udp, udplite, icmp, ???, ???, sctp o all.
- **--sport puerto [:puerto]**: puerto de origen (o rango) entre 1024 y 65535.
- **--dport puerto [:puerto]**: especifica el puerto de destino o rango,
- **-s dirección [/máscara]**: la dirección de origen o un bloque mediante la máscara.
- **-d dirección [/máscara]**: especifica la dirección de destino.
- **-i nombre**: especifica la interfaz de la regla.
- **-o nombre**: especifica la interfaz de salida.
- **--state estado**: especifica el estado: INVALID, NEW, STABLISHED, RELATED.
- **-j objetivo**: le indica que hacer si un paquete coincide.
- **-g cadena**: le indica que continúe procesando en una nueva cadena.

Ejemplos:

- Acepta cualquier IP al tcp por el puerto 80: "*iptables -A INPUT -s 0.0.0.0 -p tcp –dport 80 -j ACCEPT*".
- Permitir que una máquina haga ssh y denegar las demás: "*iptables -A INPUT -s 192.168.1.3 -p tcp –dport 22 -j ACCEPT*" e "*iptables -A INPUT -s 0.0.0.0 -p tcp –dport 22 -j DROP*".

Nat:
Primero hay que habilitar NAT "**iptables -t nat -A POSTROUTING -o interfaz -j MASQUERADE**". El reenvio de puertos se realiza con "**-j**", ejemplo de redirigir el puerto 22: "iptables -t nat -A PREROUTING -p tcp -i eth0 --dport 22 -j DNAT --to-destination 192.168.107.64:22".

Restaurar reglas de enrutamiento de forma ordenada:
Lo mejor es crear un script para establecer la configuración y añadirlo como inicio del SO.

Routed:
Para enrutar redes mediante un protocolo de enrutamiento como RIP. Basta con ejecutar "**routed**" y lo hará sólo.

212.2 Securizando servidores FTP

El problema de FTP es que envía las contraseñas en texto plano.
Algunos servidores FTP son:

- **Pure-FTPd**: muy seguro.
- **Vsftpd**: destaca en seguridad, estabilidad y velocidad.
- **ProFTPD**: difícil de configurar.

Usa el **puerto 20 para transferencia** de datos y el **21 para emitir comandos**. Hay dos modos de **funcionamiento**:

- **Activo**: inicia conexión y luego conexión inversa.
- **Pasivo**: el cliente inicia ambas conexiones.

Modos de **transmisión** de datos:

- **ASCII**: corrompe archivos binarios (gráficos, docs, etc).
- **Binary**: para archivos en texto plano.

Pure-FTP
El archivo de configuración se encuentra en "**/etc/default/pure-ftpd-common**" y "**/etc/pure-ftpd/conf**", en otros se encuentra en "**/etc/conf.d/pure-ftpd**".

Opciones de "**pure-ftpd**":

- **-4**: sólo acepta conexiones IPv4.
- **-6**: sólo acepta conexiones IPv6.
- **-a gid**: No usa chroot excepto en los directorios home.
- **-A**: usa chroot con todos excepto root, para cambiar al home.
- **-B**: inicia el servidor en segundo plano.
- **-c nº**: acepta un número máximo de conexiones de clientes simultáneas, por defecto 50.
- **-C nº**: igual que "**-c**" pero por IP del cliente.
- **-e**: sólo acepta conexiones anónimas.
- **-E**: sólo acepta acceso que no sean anónimos.
- **-i**: no deja subir a anónimos.
- **-M**: permite a los usuarios anónimos crear directorios.
- **-N**: usa el modo activo por defecto.
- **-u uid**: deshabilita el acceso a los usuarios con número UID menor.

El acceso **anónimo** implica que el ordenador tenga una **cuenta llamada** "**ftp**".

Vsftpd:
Se ejecuta desde un superservidor, pero se puede desde local. El archivo de configuración se encuentra en "**/etc/vsftpd.conf**" o en "/**etc/vsftpd/vsftpd**". Formado por "**opción=valor**" (sin espacios).

Opciones:

- **listen=Yes/No**: con Yes escucha conexiones IPv4, si se ejecuta como supervisor hay que poner NO.
- **listen_ipv6=Yes/No**: como listen, pero para IPv6.
- **ftpd_banner=secuencia**: mensaje de bienvenida.
- **nopriv_user=usuario**: nombre de usuario para operaciones no privilegiadas.
- **ftp_username=usuario**: usuario para acceso anónimo, por defecto "ftp".
- **local_enable=Yes/No**: indica si acepta acceso de usuarios locales autenticados.
- **anonymous_enable=Yes/No**: indica si acepta accesos anónimos.
- **anon_root=directorio**: establece el directorio que se usará como root para los accesos anónimos, por defecto "/home/ftp".
- **chroot_local_user=Yes/No**: le indica a vsftpd si usar chroot acuando acepte accesos de usuarios locales.
- **userlist_enable=Yes/No:** en Yes se comprueba el archivo "userlist_file" y deniega el acceso a usuarios antes de solicitar contraseña.
- **write_enable=Yes/No:** subir archivo.
- **anon_upload_enable=Yes/No:** subir para anónimo.

***Nota**: se puede usar con certificados.*

212.3 SSH

El más popular es **OpenSSH**. El archivo de configuración está en "**/etc/ssh/sshd_config**".
***Nota**: no confundir "**sshd_config**" (del **servidor**), con "**ssh_config**" (del **cliente**).*

Los comentarios se hacen con "**#**" o "**;**". **Opciones** importantes:

- **Protocol=1, o 2 o 1,2**: especifica versiones compatibles.
- **PasswordAuthentication=Yes/No**: especifica si se permite la autenticación a través de contraseña, por defecto Yes.
- **PubKeyAuthentication=Yes/No**: sólo en nivel2, permite la autenticación por clave pública.
- **UsePAM=Yes/No**: autentifica a través de PAM, por defecto No.
- **AllowUsers=usuarios**: grupo de usuarios separados por espacios, que se podrá conectar. También hay para filtrar grupos.
- **DenyUsers=usuarios**: inverso a AllowUsers.
- **PermitRootLogin=Yes/NO**: por defecto Yes.
- **X11Forwarding=Yes/No**: por defecto no. Permite tuneling de programas X.
- **AllowTcpForwarding=Yes/No**: por defecto yes, acepta tuneling.

Generar claves ssh:
Se encuentra en "**/etc/ssh**" y se llaman:

- **ssh_host_rsa_key** y **ssh_host_dsa_key**: privadas.
- **ssh_host_rsa_key.pub** y **ssh_host_dsa_key.pub**: públicas.

***Nota**: si en las claves aparece el número 1 antes de "-key" es para ssh nivel 1.*

Para generar las claves usamos "**ssh-keygen**":

ssh-keygen -q -t rsa -f ~/.ssh/id_rsa -C "-N"
ssh-keygen -q -t rsa1 -f ~/.ssh/id_rsa -C "-N"
ssh-keygen -q -t dsa -f ~/.ssh/id_rsa -C "-N"

Las **claves** deben de tener premisos **0600** y root como propietario, pero las **públicas "***.pub" deben de tener **0644** y dueño root.

El programa "**ssh**" registra las claves de host para cada usuario individual en "**~/.ssh/kown_hosts**", se pueden meter en el global "**/etc/ssh_know_hosts**".

Copiar archivos por ssh:
SSH incluye el comando "**SCP**" para copiar archivos, ejemplo "**scp archivo usuario@host_o_IP:**" Los "**:**" al final son imprescindible. Si desea renombrar el archivo, se pondría el nuevo nombre después de los dos puntos.

Acceso sin contraseña:
Hay que generar una clave en el sistema cliente usando "**ssh-keygen**" y copiar ese archivo de clave al servidor en "**~/.ssh/authorized_keys**".

Usar ssh-agent:
Recuerda la contraseña. Para ello usamos el "**ssh-keygen**" pero sin "**-N**", se solicita una frase de contraseña, que será la clave para todos los accesos gestionados. En el cliente ejecutamos "**ssh-agent /bin/bash**" y en la consola escribimos "**ssh-add ~/.ssh/id_rsa**".

Túneles SSH:
Encripta otro protocolo. Comprobar que "**/etc/ssh/sshd_config**" tiene a "**Yes**" la opción "**AllowTcpForwarding**". En el cliente hay que establecer una conexión especial, ejemplo "**ssh -N -f -L 142:servidor:143 usuario@servidor**".

***Nota**: si se usa un puerto <1024 hay que hacerlo como root.*

Túneles para X:
Hay que asgurarnos que "**sshd_config**" tenga el parámetro "**X11Forwarding**" esté a "**Yes**". En el lado del cliente "**ssh_config**" debe de tener "**ForwardingX11**" a "Yes".

Otros:
Cambiar el puerto por seguridad dentro de la configuración, el cliente conectaría con "**ssh -p puerto usuario@ip**".

212.4 TCPWrapper

El servicio se llama "**tdpd**", hay que configurar "**/etc/hosts.allow**" y "**/etc/host.deny**", "**allow**" tiene preferencia sobre "deny" si el host está en los dos archivos.
Lo más seguro sería añadir "**ALL:ALL**" en "**deny**" y después ir abriendo hosts.

Formato: "**servicio-nombres : cliente-lista [: consola-comando]**":

- **servicio-nombre**: puede ser "in.ftpd", se sacan usando "ps".
- **cliente-lista**: es para especificar clientes por:
  - **Dirección IP**.
  - **Rango de IPs** (con máscara o /).
  - **Nombre de host**.
  - **Nombre de Dominio**.
  - **Nombre de grupo de red NIS**.
  - **Comodiles**: ALL, LOCAL, UNKNOWN, KNOWN, PARANOID.
  - **Nombres de usuario**: como usuario@host.

Se puede usar "**EXCEPT**" para omitir una regla, ejemplo: denegar todos de un rango de IPs excepto una.
TCPWrapper **sólo protege** a los servicios que se inician a través de "**inetd**".

Libwrap: es la bibliteca que implementa la funcionalidad de TCPWrapper.

212.5 Tareas de seguridad

Telnet: se puede usar para un análisis rápido, "telnet web.com 80".

Netstat: con "**-l**" informa de los puertos abiertos.

Nc: envía info entre equipos, monitoriza puertos.

Nmap: escaneo de red, como "**nc**" pero más completo.

OpenVas: escaner de red.

Snort: esnifer de paquetes y puede funcionar como IDS (sistema de detección de intrusos).
Configuración en "**/etc/snort/snort.conf**".
Lo iniciamos con "**snort**" y va dejando registros en "**/var/log/snort**".

Fail2Ban: Fail2ban es una aplicación escrita en Python para la prevención de intrusos en un sistema, que se basa en la penalización de conexión (bloquear conexión) a los orígenes que intentan accesos por fuerza bruta.
Podemos descargar reglas desde su web, que iremos añadiendo. También podemos guardar logs en MySQL.

Se encarga de buscar en los registros (logs) de los programas que se especifiquen, las reglas que el usuario decida aplicarle una penalización. La penalización puede ser bloquear la aplicación que ha fallado en un puerto, bloquearla para todos los puertos, etc. Este veto se realiza actualizando el firewall (iptables). Por ejemplo lee: /var/log/pwdfail o /var/log/apache/error_log y veta todas aquellas ips que fallan un determinado número de veces.

Fail2ban se caracteriza por su simplicidad a la hora de configurar la aplicación, ya que sólo tiene un único fichero de configuración, que se encuentra en: "**/etc/fail2ban.conf**" o "**/etc/fail2ba/fail2ban.conf** y **jail.conf**".

Opciones de Fail2ban:

- **-b**: ejecuta fail2ban en background.
- **-d** ejecuta fail2ban en modo depuración.
- **-c archivo**: lee el archivo de configuración.
- **-p archivo**: crea PID lock en un archivo.
- **-h**: muestra la ayuda.
- **-i [IPs]**: IPs a ignorar.
- **-k**: mata el proceso Fail2Ban.
- **-r nº**: número máximo de fallos de contraseña.
- **-t tiempo**: tiempo de baneo para IP.
- **-v**: muestra más información.
- **-V**: muestra la versión de Fail2ban.

Actualmente fail2ban establece filtros para Apache, sshd, qmail, vsftpd, lighttpd, Postfix y Courier Mail Server. Los filtros son escritos con expresiones regulares de Python que establecen la regla que hará disparar una determinada acción sobre la IP que origina el hecho. La tupla (regla, acción) o (filtro-> carpeta filter.d, acción-> carpeta action.d) es llamado "Jail" o "prisión", y es lo que determina la penalización a un host maligno.

Tema 213: Solución de problemas

213.1 Identificación de las estapas del arranque y solución de problemas del bootloader

Identificar el cargador de arranque:

- **Grub** Legacy: El grub1, hasta el 2010, sólo BIOS.
- **Grub2**: BIOS y EFI.
- **Burg**: fork de Grub2, pero con temas gráficos.
- **LILO**: más antiguo que Grub Legacy, sólo con BIOS.
- **ELILO**: LILO pero con soporte EFI.
- **VEFIT**: de uso habitual de PCs con EFI y que arrancan PCs con MacOS y Linux.
- **Loadlin**: desde MSDOS.
- **Syslinux**: es una familia: isolinux, extlinux, pxlinux, etc.
- **El de Windows**.
- **EFI**: necesita Grub2 o Elilo.

Existe un script para averiguar que bootloader usa "**bootinfoscript**" desde sourceforge.

Localizar los archivos y el código del cargador de arranque:
Ubicaciones clave:

- **MBR**: en él reside parte de Grub y LILO.
- **Sectores posteriores a MBR sin asignar**: los 62 sectores siguientes al MBR están sin asignar, aunque algunos bootloaders lo usan para contenido adicional y puede acarrear problemas.
- **La partición de arranque BIOS**: Grub2 lo usa en sistemas con GPT.
- **Sector de arranque de una partición Linux**: un bootloader puede usar el primer sector de una partición Linux como MBR.
- **Archivos de apoyo al arranque**: como el disco RAM.
- **Archivos de configuraciones previas a la instalación**: lilo.conf, grub.d.
- **Archivos de configuración del tiempo de arranque**: grub.cfg, menu.lst, etc.
- **Binarios**.

Mensajes de error de los bootloaders:

- **lilo**: aparece en pantalla:
  - **Nada**: lilo no ha cargado, no está instalalo lilo o la partición está dañada.
  - **L##:** carga la primera fase, pero no es capaz de localizar la segunda que está en "**/boot/boot.b**".
  - **LI**: localiza la segunda fase, pero no la puede ejecutar.
  - **LI1010...**: indica que no es capaz de localizar la imagen del kernel.
  - **LIL**: carga y ejecuta la segunda fase, pero no se puede leer el archivo "**/boot/map**".
  - **LIL?**: ejecuta la segunda fase en dirección equivocada, ocurre si se mueve "**/boot/boot.b**" sin volver a instalar lilo.
  - **LIL-**: se detectan fallos en "**/boot/map**", ocurre si se mueve este archivo sin volver a instalar lilo.
  - **LILO**: error en Kernel o en disco RAM.
- **Grub Legacy**: tiene 3 **fases**: **1**-> arranca grub desde MBR, **1,5**-> código después de MBR y fase **2**-> carga desde archivos en la partición de arranque. Errores:

- **Hard Disk Error**: (fase 1) la geometría del HDD no se puedo determinar.
- **Floppy Error**: (fase 1) la geometría del disquete no se puedo determinar.
- **Read Error**: (fase 1) no se pudo leer el disco.
- **Geom Error**: (fase 1) el archivo que está siendo leido reside fuera del área incluida en la BIOS.
- **Error nº**: (fase 1.5) el nº de error corresponde con un error de fase 2.
- **Errores de fase 2**: hay 34 distintos.

- **Grub2**: aparece la consola de rescate, esto sucede cuando no se puede localizar "**/boot/grub/grub.cfg**". Normalmente ocurre por error en la variable interna "**prefix**".
  Desde consola buscaríamos donde está "/" y luego la asignaríamos con "**set prefix=(disco,partición) /grub**", y luego "**set root=(disco,partición)**":
  Luego podemos volver al menú normal con "insmod normal" y luego "normal".
  Todo es temporal y habrá que volver a instalar grub con "**sudo grub-install /dev/dispositivo**".
  Si en el menú pulsamos "**e**", podemos editar las entradas de SO.

***Nota**: podemos recuperar bootloaders grub y grub2 con el disco de rescate "**SuperGrub**", usando "**grub-install**".*

***Nota**: con el comando "**rdev**" modificamos el dispositivo "/", el intercambio, el disco RAM, o el modo de vídeo de una imagen. Sin parámetros muestra "/etc/mtab".*

213.2 Problemas generales

Comprobar el bufer circular del Kernel:
Usamos "**dmseg**", es preferible canalizarla con "dmesg | less" o redirigirla a un archivo.
Normalmente se encuentra estos datos en "/var/log/dmesg o /var/log/boot.log".

Identificar hardware:
- **lspci**: muestra información de los dispositivos PCI conectados. (-v más info y -s nº_bus para ver uno en concreto).
- **lsusb**: muestra información de los dispositivos USB conectados. (-v más info y -s nº_bus para ver uno en concreto).
- **lsdev**: no instalada por defecto (instalar procinfo), muestra las líneas DMA, IRQ y los puertos I/O-
- **lsmod**: para módulos con "insmod" y "modprobe".

Nota: "**uname**" indica arquitectura, Kernel, etc.

Archivos de log:
Depende de la distro, hay que mirar "**syslog**" y "**/var/log/**". Recomendable establecer "**tail -n nºdelineas archivo**" para que muestre las últimas nº de líneas.
Hay una herramienta para mirarlas como "**Logcheck**" del paquete "**Sentrytools**".

Seguimiento de las actividades de un programa:
En vez de gui, probar por terminal para ver los mensajes. Podemos usar "**strace programa archivo_salida**" y mirar las actividades, reflejándolas en el archivo, esto monitoriza las llamadas al sistema.

Existe también "**ltrace**", pero monitoriza las llamadas a bibliotecas dinámicas.

Identificar bibliotecas compartidas:
Para conocer las bibliotecas necesarias por un programa se usa el comando "**ldd"**,
ejemplo: "**ldd /usr/bin/vi"** y mostrará todas las bibliotecas que necesita. Si aparece "**not found"** es que hay un error y no encuentra esa biblioteca.

Localización de las bibliotecas:
El programa responsable de cargar la biblioteca y vincularla al programa es "**ld.so"** que es invocado cada vez que un programa necesita una función de biblioteca externa.
El ld.so consigue localizar con la ayuda del mapeo encontrado en "**/etc/ld.so.cache"**.
Las blibliotecas estandar del sistema suelen encontrarse en /lib y /usr/lib, si se añaden otros directorios se deben de incluir en "**/etc/ld.so.conf"** o en "**/etc/ld.so.conf.d/"** en algunas distros. El comando "**ldconfig"** actualiza esta caché con las configuraciones de ld.

Encontrar archivos abiertos:
Con "**lsof**" encontramos archivos abiertos, podemos filtrar con "**-u usuario**" y también por el programa con "**lsof | grep programa**".

Localiza secuencias en archivos binarios:
Con "**strings archivo_binario**" busca texto ASCII. Lo ideal es redirigirlo a un archivo o usar "**less**". **Opciones**:

- **-a**: escanea todo el archivo.
- **-f**: muestra el nombre del archivo antes de cada secuencia.

213.3 Solución de problemas relacionados con los recursos del sistema

Problemas de SysV:
Estaría relacionado con "**/etc/inittab**", hay que comprobar el runlevel por defecto "**id:3:initdefault**".

Modificar las opciones del Kernel:
Modificando el sistema de archivos "/**proc**" o usando "**sysctl**" podemos modificar parámetros.

Formato sysctl: "**sysctl [opciones] [clave] [clave=valor | nombrearchivo]**".
Opciones de **sysctl**:

- **-a**: muestra todas las claves y sus valores.
- **-A**: igual que "-a" pero formateado.
- **-n**: deshabilita la vista de nombres.
- **-N**: deshabilita la lista de valor.
- **-q**: como "-N", pero sólo afecta a stdout.
- **-e**: deshabilita errores al encontrar una clave desconocida.
- **-w clave=valor**: asigna valores.
- **-p archivo**: cambia parámetros especificadas en el archivo.

Ejemplo "sysctl Kernel.hostname=moninopc".
Las opciones se encuentra en "**/etc/sysctl.conf**".

Arreglar problemas de scripts de inicio:
Si al iniciar un servicio falla, hay que comprobar que el runlevel sea el adecuado. Comprobar vínculo simbólico en "**/etc/rc?.d**, **/etc/rc.d/rc** o **/etc/init.d/rc**".
Los nº de secuencia de inicio deben de sumar 99, ej: "K37servidor" pues el de inicio debe ser 99-37=62 "S62servicio". Otra forma de controlarlo es con "**rc-update** y **update-rc.d**".

Probelas con Upstart:
Upstart es más reciente que SysV. Hay que mirar "**/etc/init** o **/etc/event.d**" y ajustar los runlevels.

Scripts de inicio locales:
Se encuentran en "**/etc/rc.local** o **/etc/rd.rc.local** o **/etc/rc.d/boot.local**". Para añadir ponemos la ruta del binario con "**&**" al final para que sea en 2º plano.

213.4 Solución de problemas con variables de configuración

Ajustar variables de acceso: Los valores se encuentran en "**/etc/login.defs**".

Problemas con bash: Script de inicio de consola.

	Archivos de acceso	**Archivos de no acceso**
Archivos globales	/etc/profile y /etc/profile.d	/etc/bash.bashrc /etc/bash/bashrc /etc/basrc
Archivos de usuario	~/.bash_profile ~/.bash_login ~/.profile	~/.bashrc

Opciones de acceso:
Con el demonio "**syslog.d**" se pueden mandar logs a otros servidores.
El archivo de configuración es "**/etc/syslog.conf**".

Cron:
Cron es impulsado por un "***crond*"**, un archivo de configuración que especifica comando shell para ejecutarse periódicamente a una hora específica. Los usuarios habilitados para crear su fichero **crontab** se especifican en el fichero "**cron.allow**" y los que no en "**cron.deny**". Estos dos últimos ficheros se encuentran en "**/etc/cron.d/** o **/etc/**".

```
minuto (0-59)
|   hora (0-23)
|   |   día del mes (1-31)
|   |   |   mes (1-12)
|   |   |   |   día de la semana (0-6 donde 0=Domingo)
|   |   |   |   |          comandos
15 02   *   *   *
```

Ejemplo: "30 10 * * 1 /usr/bin/who >> /home/quien.tex".Ejecuta la orden who todos los lunes a las 10:30 y guarda la salida en el fichero quien.tex.

Para listar las tareas cron de un usuario "***sudo crontab -u usuario -l***".