

SOFTWARE LIBRE Y EDUCACIÓN:
SERVICIOS DE RED, GESTORES DE
CONTENIDOS Y SEGURIDAD

Linux como servidor



José Ángel Bernal, Fernando Gordillo, Hugo Santander y Paco Villegas

8 de marzo de 2005

Índice general

1. Demonios y Superdemonios	7
1.1. inetd	8
1.2. xinetd	10
1.3. Parando y arrancando demonios	13
1.3.1. Debian	13
1.3.2. Fedora	14
1.3.3. Algunos servicios de red usuales	16
1.4. TCP-Wrappers	17
1.4.1. Reglas de acceso	18
2. Terminal remoto. Telnet y SSH	21
2.1. Visión general	21
2.1.1. Acceso remoto: telnet	21
2.1.2. Copia remota: ftp	23
2.1.3. Una solución más segura	27
2.2. La Criptografía llega en nuestra ayuda	29
2.3. SSH como cliente	32
2.3.1. Sesiones remotas con SSH	32
2.3.2. Autenticación por clave criptográfica	34
2.3.3. Y ahora que tenemos las claves ... ¿qué hacemos con ellas?	36
2.3.4. El agente ssh	37
2.3.5. Uso del agente SSH en GNOME	38
2.4. Configuración del servidor SSH	39
2.4.1. Instalación	39
2.4.2. Configuración	40
3. Servidor de nombres DNS	43
3.1. ¿Qué necesito del DNS?	44
3.2. Recursos del Servidor de Nombres	44
3.3. Servidores de Nombres	48
4. Servicio de Directorio LDAP	53
4.1. Estructura del Directorio	54
4.2. Servidor OpenLDAP	56
4.2.1. Configuración del Servidor OpenLDAP	56
4.3. Clientes LDAP	61
4.4. Caso práctico - LDAP	64
4.4.1. Crear un directorio para autenticación	65
4.4.2. Configuración de Name Service Switch	66
4.4.3. Módulos PAM	68
4.4.4. nscd	69

5. Compartir impresoras:Cups	71
5.1. Introducción	71
5.2. Instalación	72
5.2.1. Fedora	73
5.2.2. GuadaLinux	73
5.3. Configuración de CUPS	74
5.3.1. client.conf	75
5.3.2. cupsd.conf	75
5.4. Interfaz Web	84
5.4.1. Añadir una impresora	85
5.4.2. Añadir una clase	89
5.5. Un poco de comandos	90
5.5.1. lpadmin	91
5.6. ➡ Para Practicar	92
6. Samba	95
6.1. ¿Qué es Samba?	95
6.2. Instalación	97
6.2.1. Fedora	97
6.2.2. Debian	98
6.2.3. Programas	98
6.3. Configuración	99
6.3.1. Configuración de las máquinas Güindows	99
6.3.2. Configuración de la máquina Linux	100
6.3.3. Swat	109
6.4. A “bailar” la Samba	110
6.4.1. Acceder desde una máquina Linux a una Windows	110
6.4.2. Acceder desde Windows a la máquina Linux	116
7. Servicio de compartición de ficheros NFS	121
7.1. Servidor NFS	121
7.1.1. Fichero /etc/exports	123
7.1.2. RPC y portmap	124
7.2. Cliente NFS	125
7.2.1. Montar sistemas de archivos NFS usando /etc/fstab	125
8. Servicio de Proxy-caché	127
8.1. ¿Qué es un proxy caché?	127
8.2. Squid, un proxy caché para Linux	127
8.2.1. Visión general	127
8.2.2. Conceptos sobre cachés	128
8.2.3. Instalación	128
8.3. Configuración de Squid	129
8.3.1. Configuración básica	129
8.3.2. Configuración de jerarquía de caché	131
8.3.3. Control de acceso	131
8.4. Configuración de los clientes	134
8.5. Acceso a internet autenticado contra ldap	136
8.5.1. Métodos de autenticación de Squid	136
8.5.2. Analizador de logs SARG	138
8.6. ➡ Para practicar	147
8.6.1. Castellanizar los errores de Squid	147
8.6.2. Limitar ancho de banda para determinadas extensiones	147
8.6.3. Proxy transparente	149



8.7. DansGuardian	149
8.7.1. Funcionamiento	150
8.7.2. Instalación	150
8.8. Configuración	150

Capítulo 1

Demonios y Superdemonios

Par Dios, señor -replicó Sancho-, ya yo los he tocado; y este diablo que aquí anda tan solícito es rollizo de carnes, y tiene otra propiedad muy diferente de la que yo he oído decir que tienen los demonios; porque, según se dice, todos huelen a piedra azufre y a otros malos olores; pero éste huele a ámbar de media legua. (*El ingenioso hidalgo Don Quijote de la Mancha*. MIGUEL DE CERVANTES SAAVEDRA).

Como vimos en la primera entrega, cuando se ejecuta un proceso servidor, éste abre un socket y permanece escuchando en un puerto de nuestra máquina Linux, en espera de que se conecten los clientes. Estos procesos reciben el nombre de *demonios*¹. Muchos procesos se ejecutan de esta forma, como *apache* o *sendmail*. Normalmente suelen ser procesos servidores que por su importancia, merecen un trato diferenciado. Éstos tienen sus propios ficheros de configuración y sus medidas de seguridad ya incluidas.

Algunos de estos demonios funcionan de forma diferente. Hay procesos servidores que atienden ellos mismos a todos los requerimientos de los clientes, y otros procesos servidores que son más comodones y lanzan un proceso hijo para que atienda las peticiones de los clientes y ellos seguir tranquilamente “durmiendo”². Ejemplo del primer caso puede ser el demonio *nscd*³, que hace él solo la tarea y el servidor *httpd* (*apache*), ejemplo del segundo (lanza varios hijos para servir las peticiones).

```
#ps aux
USER PID%CPU%MEM VSZ RSS TTY STAT START TIME COMMAND
nscd 1870 0.0 0.6 56244 836 ? S 20:28 0:00 /usr/sbin/nscd
root 1886 0.0 0.4 2724 576 ? S 20:29 0:00 /usr/sbin/smartd
root 1901 0.0 0.5 4580 644 ? S 20:29 0:00 /usr/sbin/sshd
apache 2301 0.0 7.2 21468 9108 ? S 21:02 0:00 /usr/sbin/httpd
apache 2302 0.0 7.2 21468 9104 ? S 21:02 0:00 /usr/sbin/httpd
apache 2303 0.0 7.2 21468 9104 ? S 21:02 0:00 /usr/sbin/httpd
apache 2304 0.0 7.2 21468 9104 ? S 21:02 0:00 /usr/sbin/httpd
apache 2305 0.0 7.2 21468 9104 ? S 21:02 0:00 /usr/sbin/httpd
apache 2306 0.0 7.2 21468 9104 ? S 21:02 0:00 /usr/sbin/httpd
apache 2307 0.0 7.2 21468 9104 ? S 21:02 0:00 /usr/sbin/httpd
apache 2308 0.0 7.2 21468 9104 ? S 21:02 0:00 /usr/sbin/httpd
```

Sin embargo, si todos los procesos se ejecutaran así, nuestro sistema se volvería inmanejable y el consumo de recursos por parte de estos procesos sería muy elevado. Cientos de demonios

¹En el contexto de los sistemas Unix, demonios son procesos que trabajan en segundo plano atendiendo a varias tareas sin intervención humana. El símbolo del demonio de la rama Unix BSD viene de ahí. Ver: <http://www.freebsd.org/es/copyright/daemon.html>

²Suele ser más eficiente para procesos con mucha carga.

³Fijaos que suelen terminar en la letra “d”, de demonio. Éste es el *Name Server Caché Daemon*, demonio que realiza una caché de las peticiones de resolución de nombres para mejorar los tiempos de respuesta.

escuchando en los puertos de nuestra máquina y sin control. Espeluznante. Aquí es donde el superdemonio⁴ *inetd* (o *xinetd* en sistemas más modernos) viene en nuestra ayuda.



Así aparecen dos modos de operación para los demonios de red. Éstos son:

autónomo (standalone) el programa demonio de red escucha en el puerto de red asignado y, cuando llega una conexión, se ocupa él mismo de dar el servicio de red (o bien uno de sus hijos). En este modo suele trabajar por ejemplo el servidor web (en nuestro caso Apache).

esclavo del servidor xinetd (o inetd) se trata de un superdemonio⁵ (siempre están en ejecución) cuya finalidad es estar a la espera de que se produzca alguna solicitud de conexión del exterior. Si esto pasa, *xinetd* analiza esa solicitud determinando qué servicio le están solicitando y le pasa el control a dicho servicio.

1.1. inetd

Aunque más antiguo y con menos posibilidades que *xinetd*, el “anciano” *inetd* aún sigue utilizándose en algunos sistemas, entre ellos Guadalinex.

`/etc/inetd.conf` es el fichero de configuración para el demonio servidor *inetd*. Su función es la de almacenar la información relativa a lo que *inetd* debe hacer cuando recibe una petición de conexión a un servicio en particular. Para cada servicio que deseemos que acepte conexiones de red, debemos decirle a *inetd* qué demonio servidor ejecutar, y cómo ha de hacerlo.

Es un fichero de texto en el que cada línea describe un servicio. Cualquier texto en una línea que siga al carácter `#` es ignorado y se considera un comentario. Cada línea contiene siete campos separados por cualquier número de espacios en blanco (espacio o tabulador). El formato general es el siguiente:

```
<servicio><tipo_socket><proto><flags><usuario><servidor><argumentos>
```

servicio es el servicio correspondiente a esta configuración, tomado del fichero `/etc/services` y se corresponde con un número de puerto en el que escuchar.

tipo_socket describe el tipo de socket para esta entrada. Los valores permitidos son: **stream**, **dgram**, **raw**, **rdm** o **seqpacket**. Simplificando, los servicios basados en **tcp** usan **stream**, y casi todos los basados en **udp** usan **dgram**. Sólo algunos demonios servidores muy particulares usarán otros valores.

proto el protocolo considerado válido para este servicio. Debería corresponder con la entrada apropiada en el fichero `/etc/services` y suele ser **tcp** o **udp**. Los servidores basados en RPC (*Remote Procedure Call*) usarán **rpc/tcp** o **rpc/udp**.

flags sólo hay dos valores posibles: **wait** y **nowait**. Este campo le dice a *inetd* si el programa servidor de red libera el socket después de comenzar la ejecución, y si por tanto *inetd* podrá ejecutar otro servidor para la siguiente petición de conexión. Si no se libera, *inetd* deberá esperar y asumir que el demonio servidor que esté ejecutándose controlará las nuevas peticiones de conexión. Por norma general todos los servidores **tcp** deberían tener esta entrada con el valor **nowait** y la mayoría de servidores **udp** deberían tener **wait**.

usuario este campo indica qué cuenta de usuario de `/etc/passwd` será asignada para ejecutar el servidor cuando se lance. Esto es a menudo útil si quiere protegerse ante riesgos de seguridad. Puedes asignar el usuario **nobody** a una entrada, por lo que si la seguridad del servidor de red se ve comprometida, el posible daño queda minimizado. Habitualmente, sin embargo, este campo está asignado a **root**, porque muchos servidores requieren privilegios de administrador para funcionar correctamente.

⁴También llamado Superservidor

⁵Red Hat/Fedora utiliza el sistema *xinetd*, que es una mejora del servidor *inetd* (utilizado aún por Guadalinex).

servidor este campo es el camino completo hasta el programa servidor a ejecutar para esta entrada.

argumentos este campo comprende el resto de la línea de órdenes y es opcional. Es en donde se pone cualquier argumento de línea de comandos que se desee pasar al programa demonio servidor cuando es ejecutado.

↪ Un ejemplo de una línea del fichero `/etc/inetd.conf`

```
telnet stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.telnetd
```

La línea anterior significa que para el servicio telnet (puerto 23, que recogerá del fichero `/etc/services`) habrá un servicio `tcp` de tipo `stream`. El modo es `nowait`, que quiere decir que el demonio `inetd` escucha en el puerto 23. Cuando tiene una petición, la atiende generando un servicio de telnet, pero tras atenderla con un proceso hijo, se pone otra vez a escuchar, sin esperar a que la conexión que ha lanzado anteriormente haya terminado.

El proceso `/usr/sbin/tcpd` realizará un control de seguridad⁶ y si es un acceso permitido, lanzará el servidor `/usr/sbin/in.telnetd`. Si no queremos que nuestra máquina proporcione un servicio determinado, debemos comentar la línea en este fichero, precediéndola del carácter `#`. Para que se active, podemos reiniciar la máquina, pero no es estrictamente necesario. Otras formas son:

1. Decirle que vuelva a cargar el fichero de configuración: `/etc/init.d/inetd reload`⁷
2. Localizar el número de proceso del servidor `inetd`.

```
root@guadalinux:~# ps aux|grep inetd
root 2656 0.0 0.1 2264 204 ? Ss 17:26 0:00 /usr/sbin/inetd
root 8017 0.0 0.5 2392 704 pts/1 R+ 21:05 0:00 grep inetd
```

Vemos que el número de proceso es el 2656. La línea con el `grep` es resultado de nuestro comando anterior.

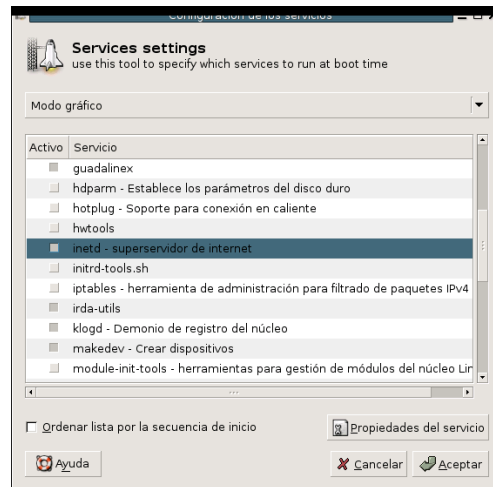
```
#kill -HUP 2656
```

Mandamos la señal `-HUP` al proceso `inetd`. Al recibirla, este proceso sabe que tiene que volver a leer su fichero de configuración (`/etc/inetd.conf`) y actualizar su modo de funcionamiento ante los cambios.

Por defecto, en Guadalinux `inetd` viene deshabilitado. Para habilitarlo, lanzamos **Aplicaciones** → **Configuración** → **Servicios** (o desde la línea de comandos `#services-admin`) y marcar la casilla de activación.

⁶Más adelante hablaremos de `Tcp-Wrappers`.

⁷También podríamos haber reiniciado `/etc/init.d/inetd restart`



1.2. xinetd

El superdemonio xinetd aparece en los sistemas RedHat/Fedora. Al igual que inetd, es un proceso especial que se queda a la escucha de conexiones TCP en unos puertos determinados. Cuando viene una solicitud de conexión, realiza una serie de comprobaciones y ejecuta el proceso servidor correspondiente.

Su funcionamiento es el siguiente: xinetd al arrancar, lee sus ficheros de configuración, que básicamente le dicen en qué puertos tiene que escuchar y cuando recibe una petición de conexión a uno de esos puertos⁸, qué servidor ejecuta y en qué condiciones debe ejecutarlo.

Lo más normal es que xinetd esté ya instalado en nuestro sistema, al ser un soporte básico de red. Su configuración se encuentra en un fichero principal (`/etc/xinetd.conf`) y un directorio (`/etc/xinetd.d`) en donde se encuentran ficheros para cada uno de los servicios que controla.

Veamos el fichero `/etc/xinetd.conf`

```
# Simple configuration file for xinetd
#
# Some defaults, and include /etc/xinetd.d/
defaults
{
    instances                = 60
    log_type                  = SYSLOG authpriv
    log_on_success            = HOST PID
    log_on_failure            = HOST RECORD
    cps                       = 25 30
}
```

`includedir /etc/xinetd.d`

Veamos qué nos dice el fichero. Dentro de la sección `defaults`, que se aplica a todos los servicios bajo el control de xinetd por defecto.

instances = 60 Se limita el número de conexiones simultáneas a cada servicio a un máximo de 60. Es una forma de evitar que alguien nos colapse la máquina intentando múltiples conexiones, o prevenir bloqueos por algún fallo.

log_type = SYSLOG authpriv Se utiliza el servicio de *syslog* en su categoría *authpriv* para guardar el registro de actividad (log⁹).

⁸Si es el 23, llama al servidor de `telnet`; si es al 22, llama al servidor de `ssh`. Por ejemplo, si la máquina remota solicita una transferencia de ficheros por el puerto 21, le pasará la solicitud a `wu-ftpd` (proceso del servidor de ftp)

⁹Se determina en `/etc/syslog.conf` con `authpriv: /var/log/secure`

log_on_success = HOST PID Si la conexión se lleva a cabo con éxito, guarda la máquina desde la que se realiza y el identificador de proceso.

log_on_failure = HOST Si la conexión no se realiza, guarda la máquina desde la que se realizó el intento de conexión.

cps= 25 30 Limita el número de conexiones por segundo que permite. En este caso, se establece a 25. El segundo valor (30) indica el número de segundos que se espera antes de continuar la actividad, en caso de que el valor de conexiones por segundo se haya sobrepasado.

includedir/etc/xinetd.d Incluye los servicios que se encuentran en el directorio `/etc/xinetd.d`

El formato para cada uno de los ficheros que permiten configurar los distintos servicios, es de la forma¹⁰:

```
service "nombre_servicio"
{
...
    atributo = valor
    serie_valores -= elimina_valor
    serie_valores += añade_valor
...
}
```

Lo normal es que sólo se use “=” para asignar una valor a un atributo. Si el atributo es una serie de valores podemos eliminar un elemento de la serie con “-=” o añadirlo con “+=". Normalmente son:

disable toma los valores “yes” o “no”. Es donde se activa (`disable=no`) o desactiva (`disable=yes`) el servicio.

type toma los valores `RPC`, `INTERNAL` (servicio que ya provee el propio `xinetd` de forma interna, puede ser: `echo`, `time`, `daytime`, `chargen` y `discard`) o `UNLISTED` (no aparece en `/etc/services`)

id nombre unívoco para identificar este servicio

socket_type describe el tipo de socket que esta entrada considerará relevante. Los valores permitidos son: `stream`, `dgram`, `raw` o `seqpacket`. Por regla general casi todos los servicios basados en tcp usan `stream`, y casi todos los basados en udp usan `dgram`. Sólo algunos demonios servidores muy particulares usarán otros valores

protocol el protocolo considerado válido para este servicio obtenido a partir del fichero `/etc/protocols` (suele ser `tcp` o `udp`). Si no se especifica, se usa el protocolo por defecto para ese servicio.

wait puede ser `yes` o `no`. Con este atributo indicamos a `xinetd` si el programa servidor de red libera el socket después de comenzar la ejecución (`wait=no`), y si por tanto `xinetd` podrá ejecutar otro servidor para la siguiente petición de conexión o deberá esperar (`wait=yes`) y asumir que el demonio servidor que esté ejecutándose controlará las nuevas peticiones de conexión. Por norma general todos los servidores `tcp` deberían tener esta entrada con el valor `no` y la mayoría de servidores `udp` deberían tener `yes`.

user indica qué cuenta de usuario de `/etc/passwd` será asignada para ejecutar el demonio de red al activarse.

group por si queremos especificar el grupo con que se ejecuta el servicio. Tiene que tener una entrada en `/etc/group`.

¹⁰Para conocer todas las posibilidades
\$ man xinetd.conf

instances número máximo de peticiones que este servicio puede administrar.

server camino completo hasta el programa servidor a ejecutar para esta entrada.

server_args argumentos pasados al servidor.

no_access IP de máquinas que no podrán acceder a este servicio.

Si vamos al directorio `/etc/xinetd.d`, veremos que existen varios ficheros. Veamos alguno de ellos:

```
[root@linux xinetd.d]# more telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#             unencrypted username/password pairs for authentication.
service telnet
{
    disable          = no
    flags             = REUSE
    socket_type      = stream
    wait             = no
    user             = root
    server           = /usr/sbin/in.telnetd
    log_on_failure   += USERID
}
```

Veamos qué significan cada uno de los valores para el servicio telnet.

service telnet Nos indica a qué servicio (y puerto en el que escucha¹¹) es aplicable lo siguiente.

disable = no Una forma un poco confusa de decir que está activado. Se le dice que *NO* está deshabilitado.

flags = REUSE Reutiliza el socket abierto para próximas conexiones

socket_type = stream El tipo de socket es stream (TCP)

wait = no No tiene porqué haber finalizado la ejecución anterior para lanzar otro servicio nuevo

user = root Usuario que ejecutará el proceso

server = /usr/sbin/in.telnetd El servidor que arranca será `/usr/sbin/in.telnetd`

log_on_failure += USERID En caso de fallo en la autenticación, registra el usuario que ha intentado entrar.

El del servicio **swat** (interfaz vía web para configuración de SAMBA, véase 6.3.3), será el siguiente.

```
# default: off
# description: SWAT is the Samba Web Admin Tool. Use swat \
#             to configure your Samba server. To use SWAT, \
#             connect to port 901 with your favorite web browser.
service swat
{
    disable = yes
    port    = 901
    socket_type = stream
    wait    = no
    only_from = localhost
    user    = root
}
```

¹¹En caso de no encontrar un puerto en alguna de las líneas siguientes, cogerá el que tenga este nombre del fichero `/etc/services`.



```
server = /usr/sbin/swat
log_on_failure += USERID
}
```

En este servicio vemos que por defecto está deshabilitado. En caso de querer activarlo, debemos poner `disable=no` y decirle a `xinetd` que vuelva a releer su configuración. En este caso ejecutando:

```
# /etc/init.d/xinetd reload
```

El control sobre los accesos que veremos después con el complemento `tcpwrappers`, puede ser realizado de otra forma aquí, mediante las opciones `only_from` (para indicar direcciones o nombres de host o dominios desde los que se puede acceder al servicio) y `no_access` (para excluir direcciones o nombres y desde los cuales no se podrá acceder). Por ejemplo, una entrada para el servicio `telnet` anterior modificada así, quedaría:

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#             unencrypted username/password pairs for authentication.
service telnet
{
    disable          = yes
    flags            = REUSE
    socket_type      = stream
    wait            = no
    user            = root
    server          = /usr/sbin/in.telnetd
    log_on_failure   += USERID
    only_from       = 172.26.0.0/16
    no_access       = 172.26.1.0/24
    access_time     = 08:30-14:45
}
```

permite el acceso sólo a las máquinas de la red `172.26.0.0/16` del atributo `only_from`, pero impide que se conecten las máquinas de la subred `172.26.1.0/24` y además limita la posibilidad de conexión vía `telnet` al horario establecido en el atributo `access_time`. Pero que ... ¡si este servicio está desactivado!

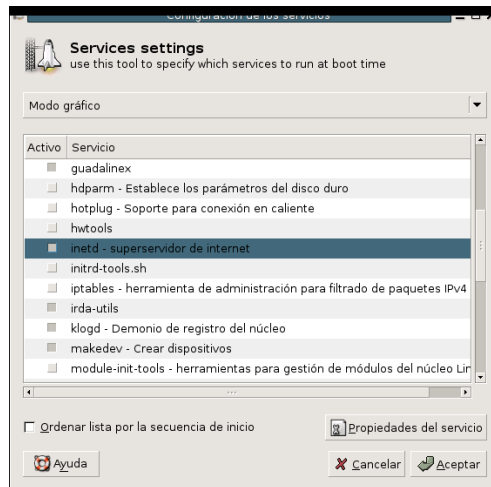
1.3. Parando y arrancando demonios

Podemos diferenciar dos formas de arrancar los demonios, ya sean independientes o controlados por `inetd/xinetd`. La primera forma de hacerlo es configurando el sistema para que los active automáticamente al arrancar. Será la más normal para los servicios que ofrezcamos desde nuestro servidor. Otra forma es arrancarlos manualmente, recordando que deberemos arrancarlos cuando los necesitemos y pararlos cuando no nos hagan falta o vayamos a apagar el servidor.

Antes de entrar de lleno en cómo instalar y configurar determinados servicios de red, vamos a recordar las herramientas de que disponemos para poder activar o desactivar determinados servicios según los distintos niveles de ejecución.

1.3.1. Debian

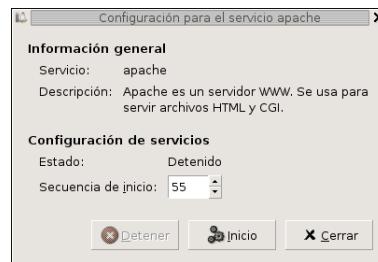
En Guadalinex contamos con la herramienta `services-admin`. Al ejecutarla, nos aparece la siguiente ventana.



El primer desplegable nos indica en qué nivel de ejecución del sistema será aplicable. Sus valores pueden ser Deteniendo el sistema (nivel 0), Modo gráfico (nivel 2), Modo texto (nivel 3) o Reiniciando el sistema (nivel 6).

Si seleccionamos el servicio, se ejecutará al pasar el sistema a dicho nivel de ejecución.

Las propiedades del servicio nos indican el nombre y descripción del servicio, el estado actual y en qué lugar se encuentra dentro del orden de arranque o parada de todos los servicios.



`/usr/sbin/update-rc.d` utilidad en línea de comandos para activar/desactivar servicios. En general es más fácil trabajar con la anterior.

Usando este comando podemos configurar los enlaces simbólicos de los directorios `/etc/rc?.d` y el script situado en `/etc/init.d/`. Si por ejemplo deseamos que el servicio de nombre `service` se ejecute en el arranque


1. Se pone en el directorio `/etc/init.d/`. En general los programas que instalemos y que sean necesarios en el arranque sitúan sus scripts de forma automática aquí.
2. Después creamos los enlaces simbólicos mediante el comando

```
update-rc.d servicio defaults 35
```

Al pasarle el parámetro `defaults` forzamos a que lo cree para los niveles de ejecución que van del 2 al 5. Con el 35 obligamos a que service se arranque antes de cualquier script que contenga un número mayor de 36.

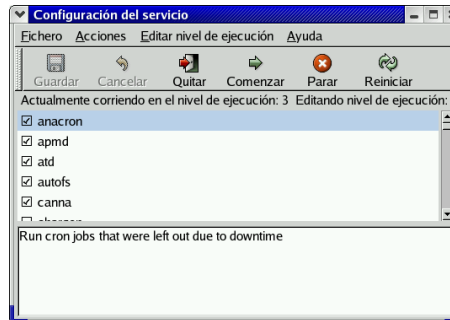
1.3.2. Fedora

Para los sistema Fedora disponemos de varias posibilidades.

`system-config-services` (desde Gnome:  → Configuración del Sistema → Configuración de servidores → Servicios). Utilidad gráfica que permite seleccionar qué servicios están en

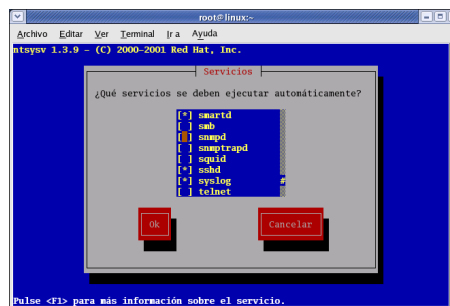


activo para los niveles 3, 4 y 5. Permite reiniciarlos, pararlos y activarlos desde el propio programa.



También se puede acceder al programa escribiendo el comando `serviceconf`.

`/usr/sbin/ntsysv` (se puede acceder a él desde el programa `setup`, opción **System services**). Los cambios no se activan en el momento. Además afecta sólo al nivel desde el que se ejecuta.¹²



`/sbin/chkconfig` utilidad en línea de comandos para activar o desactivar servicios. En general es más fácil trabajar con las dos anteriores.

Mediante estas interfaces podemos seleccionar los demonios que arrancarán automáticamente al iniciar el sistema. También se ocuparán de pararlos de forma ordenada cuando apaguemos el sistema.

Pero el haberlos activado aquí, no los activará en el momento, sino que deberemos esperar al próximo arranque del sistema.

Si lo que necesitamos es arrancarlos en el momento, distinguiremos los que se encuentran bajo el control de `xinetd` y los que son independientes.

Los que se encuentran bajo el control de `xinetd` podemos activarlos de la siguiente manera. Mediante el comando `setup` visto anteriormente, lo que hacemos es poner el valor de `disable=no` en su fichero de configuración¹³. Posteriormente, debemos reiniciar el servidor `xinetd` para que vuelva a leer los ficheros de configuración y perciba el cambio.

Mediante el comando

`#/etc/init.d/xinetd reload`¹⁴. O de forma equivalente, mediante

¹²Para conseguir que "afecte" a los niveles 3 y 5 por ejemplo, usar:

```
# ntsysv --levels 35
```

¹³También podríamos editar el fichero, pero es más cómodo así.

¹⁴o `restart` si preferimos parar y arrancar.



```
#service xinetd restart.
```

Respecto a los demonios independientes, podemos ver su fichero de arranque en el directorio `/etc/init.d` y arrancarlos mediante la llamada al script correspondiente. Por ejemplo

```
#/etc/init.d/httpd start15
```

o

```
#service httpd start
```

1.3.3. Algunos servicios de red usuales

autofs activa el proceso de administración de montaje automático de sistemas de ficheros o unidades (NFS, CD...)	postgresql activa el servidor de bases de datos postgresql.
dhcpcd inicia un servidor DHCP local que permite asignar direcciones IP de forma dinámica.	sendmail activa el Agente de transporte de correo (MTA) sendmail
httpd (o apache) activa el servidor web Apache	samba (o smb) activa el servicio Samba (para compartir archivos e impresoras con redes Windows)
iptables reglas de cortafuegos del kernel	squid permite disponer del proxy HTTP squid
cupsys (o lpd) servidor de impresión.	sshd habilita servicios de red seguros (Secure SHell)
mysqld para disponer del servidor de bases de datos MySQL	syslog demonio para registrar los log (o archivos de auditoría y trazas) del sistema
netfs activo permite montar sistemas de archivos de red: NFS, Samba y NetWare.	wu-ftpd activa los servicios ftp
network para activar interfaces de red de nuestra máquina.	xfs servidor de fuentes para las X
nfs activa servicios NFS	xinetd (o inetd) permite activar múltiples servicios de red
portmap este demonio administra conexiones a servicios basados en RPC.	



Si tenemos un servicio en nuestra máquina, con:

```
$/etc/init.d/servicio
```

podemos comprobar qué parámetros admite. Por ejemplo, con el servidor de impresión obtendríamos:

```
root@guadalinux:~# /etc/init.d/cupsys
```

```
Usage: /etc/init.d/cupsd {start|stop|restart|force-reload}
```

O sea, que si queremos pararlo sólo hay que ejecutar:

```
root@guadalinux:~# /etc/init.d/cupsys stop
```

```
Stopping printing system service: cupsd.
```

Y si queremos arrancarlo, hay que ejecutar:

```
root@guadalinux:~# /etc/init.d/cupsys start
```

```
Starting printing system service: cupsd.
```

¹⁵`/etc/init.d/apache` en Guadalinux

1.4. TCP-Wrappers

Además del cortafuegos personal con iptables y de las posibilidades de control y registro de `xinetd`¹⁶, nuestra máquina Linux posee otra herramienta para defenderse de los ataques y no permitir nada más que los accesos autorizados.

El sistema TCP-Wrappers añade una capa adicional de protección a los servicios de red, en la que indicamos a qué máquinas permitimos el acceso y a cuáles no.

Unos de los servicios de red que incorporan protección mediante TCP-Wrappers son los super-servidores `inetd` y `xinetd`. Para ver qué servicios de red tienen control de TCP-Wrappers, podemos ver si en su binario se encuentra la cadena `hosts_access`, indicando que se ha compilado incluyendo su soporte.

```
#strings /usr/sbin/sshd | grep hosts_access
@(#) hosts_access.c 1.21 97/02/12 02:13:22
```

El comando anterior mira si en el ejecutable `/usr/sbin/sshd` existe la cadena `hosts_access`, prueba de que incluye el soporte para `tcp_wrappers`.

El componente más importante dentro del paquete es la librería `/lib/libwrap.so`. Un servicio que incorpora control de acceso basado en TCP-Wrappers ha sido compilado con dicha librería.

Cuando se realiza un intento de conexión a un servicio con soporte de TCP-Wrappers, se comprueban los ficheros de control de acceso de TCP-Wrappers (`/etc/hosts.allow` y `/etc/hosts.deny`) para determinar si el cliente tiene permitida la conexión.

Si un cliente tras el control de dichos ficheros, tiene permitido el acceso, TCP-Wrappers pasa la conexión (socket) al servicio solicitado y no interfiere más con la comunicación entre el cliente y el servidor.

Además del control de acceso y registro, los TCP-Wrappers pueden activar comandos para interactuar con el cliente antes de negar o liberar el control de la conexión al servicio solicitado.

Los wrappers TCP ofrecen una serie de ventajas:

- Transparencia tanto para el cliente de la red como para el servicio de red controlado por TCP-Wrappers. El cliente que se está conectando no nota que está siendo controlado por TCP-Wrappers, excepto en el caso de que el acceso no se permita y se le cierre la conexión.
- Administración centralizada en los ficheros `/etc/hosts.allow` y `/etc/hosts.deny` del control de acceso para múltiples servicios de red.

Como hemos comentado, para determinar si una máquina cliente tiene permitido conectarse a un servicio, los TCP-Wrappers utilizan los archivos : `/etc/hosts.allow` y `/etc/hosts.deny` (conocidos como archivos de acceso a host, `host_access`).

Cuando una solicitud de un cliente es recibida por un servicio controlado por TCP-Wrappers, sigue los pasos siguientes:

1. TCP-Wrappers analiza secuencialmente el archivo `/etc/hosts.allow` y aplica la primera regla especificada para ese servicio. Si encuentra una regla que coincide, permite la conexión. Si no, se mira el archivo `/etc/hosts.deny`.
2. Se analiza secuencialmente el archivo `/etc/hosts.deny`. Si encuentra una regla que coincide, rechaza la conexión. Si no, se concede acceso al servicio.

Los siguientes son puntos importantes a considerar cuando se usen wrappers TCP para proteger servicios de red:

- Puesto que las reglas de acceso en `hosts.allow` son aplicadas primero, toman precedencia sobre las reglas en `hosts.deny`. Por lo tanto, si se permite el acceso a un servicio en `hosts.allow`, una regla negando el acceso al mismo servicio en `hosts.deny` es ignorada.

¹⁶No disponible en los sistemas Guadalinex

- Las reglas en cada archivo son leídas de arriba hacia abajo y la primera regla que coincida para un servicio dado es la única aplicada, siendo el orden de las reglas muy importante.
- Si no se encuentra ninguna regla para el servicio en ninguno de los archivos, o si no existe ninguno de los archivos, se concede el acceso al servicio.

1.4.1. Reglas de acceso

Los formatos para `/etc/hosts.allow` y `/etc/hosts.deny` son los mismos.

Las reglas se tienen que formatear de la siguiente manera:

```
<demonios>: <clientes>[: <opcion>: <opcion>: ...]
```

donde,

`<demonios>` es una lista separada por comas de los nombres de procesos (no de los nombres de servicios asociados a un número de puerto) o el comodín ALL para todos los servicios. Ejemplos de nombre de procesos demonios son: `ftpd`, `telnetd`, `sshd` o `fingerd`.

`<clientes>` es una lista separada por comas de nombres de host, direcciones IP o comodines, que identifica los hosts afectados por la regla. Por ejemplo: `ciencias.iesmurgi.org` para una máquina específica, `.juntadeandalucia.es` para cualquier nombre de máquina que acabe en esa cadena, ó `80.32.` para cualquier dirección IP que comience con esos dígitos.

`<opcion>` es una acción o una lista separada con puntos y comas de acciones a realizar cuando la regla es activada. Podría, por ejemplo, ejecutar una instrucción que intentase identificar quién está autenticado en el host que se conecta, o generar un mensaje de correo u otro tipo de alerta a un administrador de sistema avisando de que alguien intenta conectar.

Hay cierto número de expansiones que podemos incluir, ejemplos comunes son:

`%h` se expande al nombre de la máquina que se conecta o a su dirección si no tiene un nombre,

`%d` es el demonio que está siendo llamado.

↔ Por ejemplo:

```
sshd : .cica.es
```

Esta regla le dice a TCP-Wrappers que controle las conexiones al demonio SSH (`sshd`) y las aplique a cualquier máquina del dominio `.cica.es`.

Si esta regla aparece en `hosts.allow`, la conexión será aceptada y si aparece en `hosts.deny`, la conexión será rechazada (si antes no ha sido aceptada en `/etc/hosts.allow`).

Este otro ejemplo de regla de acceso es más compleja y utiliza dos campos de opciones:

```
sshd : .cica.es : spawn /bin/echo '/bin/date' access denied>>/var/log/sshd.log
: deny
```

Los comodines permiten a los wrappers TCP coincidir más fácilmente con grupos de demonios o hosts. Se pueden utilizar entre otros, los siguientes comodines:

ALL hace corresponder todo. Se puede usar para la lista de demonios o en la lista de clientes.

LOCAL hace corresponder todos los nombres de máquinas que no contengan un punto (`.`), tal como `localhost`. Es decir, las que pertenecen a nuestro dominio.

KNOWN se corresponde con cualquier máquina/usuario de IP/nombre conocido

UNKNOWN se corresponde con cualquier máquina/usuario de IP/nombre desconocido

PARANOID se corresponde con cualquier nombre que no se corresponda con su dirección IP (por si se intenta camuflar la identidad real de la máquina que solicita la conexión). Hay una última palabra que también es útil. La palabra

EXCEPT permite proporcionar una lista con excepciones.

Patrones

Los patrones se pueden utilizar en el campo de lista de cliente de las reglas de acceso para especificar de forma más precisa grupos de host clientes. La siguiente es una lista de los patrones más comunes:

- Nombre de host comenzando con un punto (.). Al colocar un punto al comienzo de un nombre de host, se hace coincidir todos los hosts compartiendo los componentes listados del nombre. El ejemplo siguiente se aplicará a cualquier host dentro del dominio cica.es:

```
ALL : .cica.es
```

- Dirección IP que termina con un punto (.). Al colocar un punto al final de una dirección IP hace corresponder todos los hosts compartiendo el grupo numérico inicial de la dirección IP. El ejemplo siguiente se aplicará a cualquier host dentro de la red 192.168.x.x:

```
ALL : 192.168.
```

- Dirección IP/máscara de red.

```
ALL : 192.168.0.0/255.255.254.0
```

- El asterisco (*). Los asteriscos pueden ser usados para coincidir grupos completos de nombres de host o direcciones IP, siempre y cuando no se mezclen en la lista de clientes conteniendo otros tipos de patrones. El ejemplo siguiente aplica a cualquier host dentro del dominio cica.es:

```
ALL : *.cica.es
```

Las reglas de control de acceso aceptan además el operador, EXCEPT. Se puede usar tanto en la lista de demonios como en la lista de clientes de una regla.

El operador EXCEPT permite excepciones específicas a coincidencias más amplias dentro de la misma regla.

↔ En el ejemplo siguiente desde un archivo `hosts.allow`, todos los hosts de `cica.es` pueden conectarse a todos los servicios excepto `invitado.cica.es`:

```
ALL: .cica.es EXCEPT invitado.cica.es
```

En el otro ejemplo, situado en un archivo `hosts.allow`, clientes desde la red 192.168.0.x pueden usar todos los servicios excepto el servicio de FTP, `vsftpd`:

```
ALL EXCEPT vsftpd: 192.168.0.
```

¿Cómo y por qué actúa TCP-Wrappers? Pues muy fácil, actúa porque el demonio así lo requiere. Si el demonio ha sido compilado con soporte de TCP-Wrappers, él solito al ser ejecutado, mirará si existen los ficheros `/etc/hosts.allow` y `/etc/hosts.deny` y comprobará si le es aplicable alguna regla.

Los demonios que actúan bajo `inetd`, también pueden incluirse con control TCP-Wrappers. Se les especifica en el fichero `/etc/inetd.conf`.

```
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

Para el anterior ejemplo, cuando se recibe una petición en el puerto 23 (`telnet`), de tipo `stream` y `tcp` por parte de un cliente, el superservidor `inetd` arranca un nuevo proceso por parte del superusuario. Este proceso es `/usr/sbin/tcpd`, que es el ejecutable de TCP-Wrappers. Después comprueba los ficheros de control de acceso y si se permite, lanza el demonio `in.telnetd`, el servidor original que presta el servicio.

↔Ejemplos:

```
# /etc/hosts.allow
#
# Permitir correo de salida a todo el mundo
in.smtpd: ALL
# telnet y FTP sólo a hosts dentro de mi dominio y al host de
# thales
telnetd, ftpd: LOCAL, thales.cica.es
# Permitir finger a cualquiera pero mantener un registro de quién es.
fingerd: ALL: (finger @%h | mail -s "finger desde%h" root)

# /etc/hosts.deny
#
# Desautorizar a todos los host con nombre sospechoso
ALL: PARANOID
#
# Desautorizar a todos los host.
ALL: ALL
```

La entrada PARANOID es redundante porque la otra entrada abarca todo en cualquier caso. Ambas entradas serían razonables por defecto dependiendo de sus requisitos particulares.

La configuración más segura es tener ALL: ALL por defecto en /etc/hosts.deny para después dar permiso específicamente a aquellos servicios y hosts que se desee en /etc/hosts.allow.

Capítulo 2

Terminal remoto. Telnet y SSH

Si has seguido las secciones con atención, te habrás dado cuenta de que hemos planteado remedios contradictorios para solucionar los problemas del protocolo FTP:

No utilices el modo activo del FTP. ...

No utilices el modo pasivo del FTP. ...

Por lo tanto, si sólo existen dos métodos para realizar transferencias de datos mediante FTP, y me aconsejáis que no utilice ninguna, ¿qué hago? Muy sencillo:

Medida de protección: No utilices FTP

(*Hackers en Linux*, BRIAN HATCH, JAMES LEE y GEORGE KURTZ)

2.1. Visión general

Dentro de las labores de un administrador de sistemas está el acceso remoto a los mismos, ya sea para buscar información en algún fichero del sistema, para copiar información o ejecutando en remoto algún comando. Existen varias utilidades que realizan estas funciones como telnet, ftp, rsh o rlogin.

2.1.1. Acceso remoto: telnet

El protocolo telnet¹ es una herramienta muy útil a la hora de administrar sistemas basados en Unix/Linux en cualquiera de sus sabores. Permite acceder a una máquina remotamente, de la misma forma que lo haríamos si estuviéramos sentados delante de la consola y utilizásemos su teclado para introducir los comandos. Nos proporcionará un mecanismo para conectarnos a un servidor remoto y ejecutar comandos en él de forma totalmente interactiva. De esta forma el sistema local es transparente al usuario, el cual tiene la sensación de estar conectado directamente a la máquina remota.

Los comandos que se teclean por parte del usuario son transmitidos directamente a la máquina remota y la respuesta de ésta es mostrada en la pantalla del usuario. Una conexión interactiva por telnet se conoce también como *login* remoto.

Para ello, el ordenador del usuario debe tener la habilidad de:

- Establecer una conexión con otra máquina
- Emular un terminal compatible con la máquina remota
- Regular el flujo de datos desde el terminal del usuario a la máquina remota y viceversa

¹El término telnet proviene de *TELEcommunication NETwork*

Esto es posible mediante el uso del protocolo TCP, el cual permite transmitir datos entre dos máquinas de forma coherente, y del protocolo IP, que proporciona una dirección única de 32 bits para cada máquina conectada a la red. Sobre estas bases está construido telnet, proporcionando así una emulación local de un terminal compatible con el servidor remoto.

La conexión telnet sobre TCP se establece entre el puerto de la máquina del usuario *U* y el puerto del servidor remoto *S*. El servidor remoto escucha en el puerto 23 a la espera de nuevas conexiones. Al ser la conexión TCP *full-duplex* e identificada por el par de puertos anteriores y el par de direcciones origen y destino, el servidor puede mantener tantas conexiones simultáneas que utilizan el puerto 23 y diferentes puertos clientes *U* como sean necesarias.

El punto débil de este protocolo, tal como hemos visto en los ejemplos del programa Ethereal, es que todos los datos se transmitirán en claro en la red. Si un usuario captura los datos que viajan en la red con programas como `tcpdump` o Ethereal podemos poner en compromiso la seguridad de nuestro sistema.

Para acceder al sistema remoto se nos solicitará la identificación para poder entrar al sistema. Por ejemplo² para acceder a la máquina MILETO escribiremos:

```
$telnet mileto.cica.es
```

Con Linux podemos acceder vía el cliente `telnet` a cualquier máquina remota, pero para ser servidor tenemos que cargar el paquete adecuado, éste es:

Fedora: `apt-get install telnet-server`³

Además hay que habilitar el servicio (con `ntsysv`, ...), por ejemplo, editando el fichero `/etc/xinetd.d/telnet` y cambiando `disable=yes` por `disable=no`. Después hay que decir a `xinetd` que recargue la configuración mediante:

```
#!/etc/init.d/xinetd reload
```

Debian: `apt-get install telnetd`⁴

y reiniciamos el demonio `inetd`

```
#!/etc/init.d/inetd
```

Como ya hemos comentado, telnet “da un paseo” a las contraseñas en texto plano por Internet. Una forma segura de telnet en nuestra intranet puede ser la de usar la seguridad adicional de `tcpwrappers` de la forma⁵:

```
$cat /etc/hosts.allow
in.telnetd: 172.26.0. 127.0.0.1
```

```
$cat /etc/hosts.deny
in.telnetd: ALL
```

O bien usando⁶ el propio `xinetd`, añadiendo al fichero `/etc/xinetd.d/telnet` la línea:

```
only_from = 172.26.0.0/24 127.0.0.1
```

y después:

```
#!/etc/init.d/xinetd reload
```

➔ Para practicar

1. Cargar un servidor de telnet, modificar el texto de bienvenida (`/etc/issue.net`) y conseguir que los accesos estén limitados a la máquina local y a la subred 172.26.0.0/24.

²Previamente debemos haber establecido la conexión con nuestro proveedor de Internet.

³Se encuentra en el CD3 de la distribución, es el paquete: `telnet-server-0.17-30`

⁴Podemos optar por instalar mejor el paquete:

```
apt-get install telnetd-ssl
```

“`telnet(d)-ssl` reemplaza al normal `telnet(d)` empleando autenticación SSL y cifrado. Puede interoperar con el `telnet(d)` normal en ambas direcciones. Comprueba si el otro extremo también usa SSL, y si no es posible, emplea el protocolo telnet estándar.”

⁵Que no se olvide pulsar **INTRO** después de escribir las líneas

⁶Sólo para Fedora.

2. Es interesante probar la posibilidad que nos ofrece Linux de trabajar en modo gráfico con programas situados en otro equipo, para esto tendremos que:
Desde un Xterm de la máquina local ejecutaremos⁷

```
$ xhost +máquina_remota
```

después haremos un telnet a la máquina remota y una vez conectados escribiremos

```
$ export DISPLAY=máquina_local:0
```

por último ya sólo tenemos que ejecutar el comando que deseemos, por ejemplo, podéis probar con

```
$ mozilla & ■
```

Sumarizando, el servicio telnet es inseguro y, aunque las extensiones ssl le puedan aportar seguridad, es mejor utilizar el servicio ssh.

2.1.2. Copia remota: ftp

El servicio ftp se utiliza para cargar y descargar archivos de la red. Este servicio puede verse dividido en dos partes:

- Los usuarios con cuenta en el sistema pueden acceder a su propio sistema de archivos y cargar y descargar información.
- Utilización anónima, en la que se permite que cualquiera (sea o no usuario del sistema) se conecte a una sección del sistema de archivos del servidor para cargar y descargar información.

Las cuestiones relacionadas con la configuración del ftp anónimo por parte de los administradores son numerosas. Si el sistema de archivos y la información de usuario de ftp para el acceso público no se crean con los permisos adecuados, podemos llegar a situaciones en las que usuarios sin cuenta en el sistema pasen del espacio público al privado del servidor.

Existen varios servidores de ftp que pueden instalarse en un sistema linux⁸, aunque normalmente el incluido en las distribuciones suele ser `wu-ftpd`. Independientemente del servidor que se decida utilizar, es necesario dedicar un tiempo a diseñar las formas de acceso y a qué partes del sistema de archivos, para los distintos usuarios.

Es posible regular el acceso al usuario así como permitir sólo a determinadas direcciones IP acceder a nuestro servidor por ftp. Sin embargo, tal como ocurre con el protocolo telnet, no es posible evitar⁹ que los datos que viajan por la red entre el servidor y el cliente viajen en claro. Así proponemos realizar una captura de una sesión telnet y ftp con Ethereal, tal como se vió en la entrega anterior, para tomar conciencia de qué datos pueden “robarnos” por la red.

wu-ftp

Sólo para que se tenga una referencia, veamos una pocas pinceladas sobre uno de los históricos e inseguros servidores de ftp: `wu-ftp`. Para instalarlo:

Fedora:

Descargamos el paquete `wu-ftpd`¹⁰. Para un acceso con clave de usuario, se nos ubicará en el directorio `$HOME` del usuario y para un acceso anónimo (sin usuario del sistema) en el directorio `/var/ftp/`. En el directorio `/var/ftp/pub` podemos dejar archivos listos para ser recogidos a través de accesos anónimos¹¹.

⁷Donde `máquina_remota` es o bien la dirección IP de la máquina remota, o bien, el nombre de esa máquina

⁸Algunos de los más comunes son `ftpd`, `glFtpD`, `lukemftpd`, `vsftpd` o `ProFTPD`

⁹Las nuevas versiones van incorporando mecanismos para cifrar los datos en su tránsito por la red.

¹⁰<http://rpmfind.net/linux/RPM/redhat/updates/8.0/i386/wu-ftpd-2.6.2-12.i386.html>

¹¹Para permitir el acceso anónimo (*anonymous*) hay que instalar el paquete `anonftp` <http://rpmfind.net/linux/redhat/8.0/en/os/i386/RedHat/RPMS/anonftp-4.0-12.i386.rpm>. Sólo deberíamos instalarlo si estamos seguros de que eso es lo que queremos, un acceso anónimo a ficheros, y la seguridad no se va a resentir.

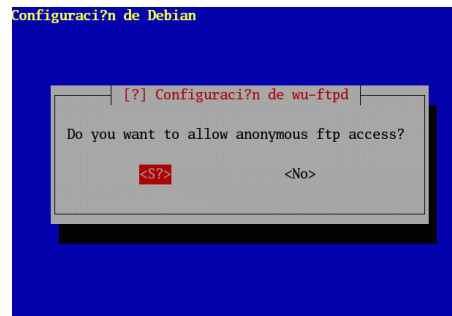
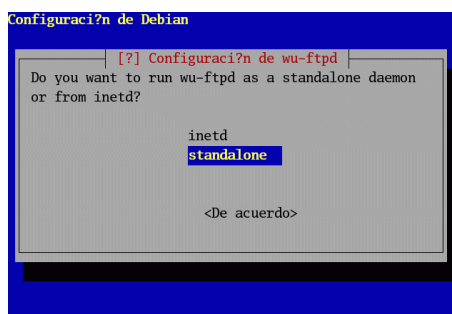
Debian: apt-get install wu-ftp

```

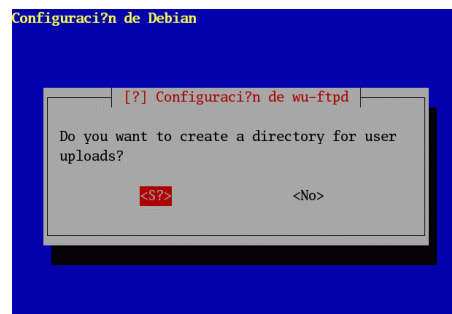
root@guadalinux:~/curso-linux/entrega2/entrega05-2# apt-get install wu-ftp
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes NUEVOS:
wu-ftp
0 actualizados, 1 se instalarán, 0 para eliminar y 123 no actualizados.
Necesito descargar 275kB de archivos.
Se utilizarán 770kB de espacio de disco adicional después de desempaquetar.
Des:1 http://http.guadalinux.org sarge/main wu-ftp 2.6.2-17.2 [275kB]
Descargados 275kB en 6s (44,0kB/s)
Preconfiguring packages ...
Seleccionando el paquete wu-ftp previamente no seleccionado.
(Leyendo la base de datos ...
106874 ficheros y directorios instalados actualmente.)
Desempaquetando wu-ftp (de .../wu-ftp_2.6.2-17.2_i386.deb) ...
Configurando wu-ftp (2.6.2-17.2) ...
Disabling other FTP services in /etc/inetd.conf
Añadiendo usuario del sistema ftp...
Adding new group 'ftp' (109).
Adding new user 'ftp' (109) with group 'ftp'.
Creando el directorio home /home/ftp.
The anonymous FTP user has been successfully set up.
Starting FTP server: wu-ftp.

```

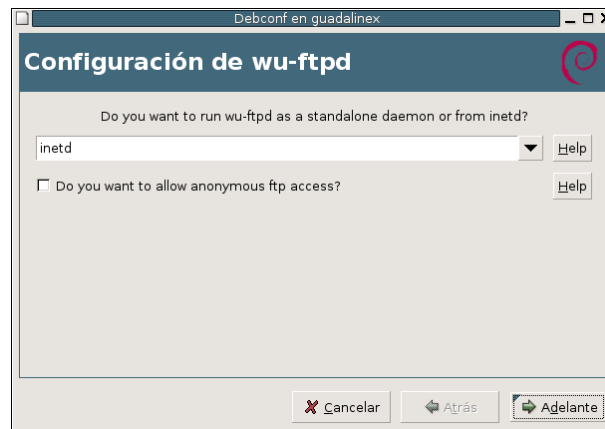
Se nos preguntará si deseamos que el servicio esté controlado por `inetd` o `standalone`. Optaremos por dejar el valor por defecto (`standalone`), es decir, que funcione como servidor independiente, ya que de esa manera podemos optimizar la velocidad de respuesta. Después pregunta si vamos a permitir accesos anónimos (login: `anonymous` y password: dirección de correo-e por convención).



Sólo en el caso de que optemos que `<sí>` (¡no deberíamos ser tan atrevidos!) tendremos que optar por crear la cuenta `ftp` (`/home/ftp`) y después la zona del disco (`/home/ftp/pub/incoming`) de accesos anónimos



En modo gráfico, sería equivalente al proceso anterior.



Configuración y clientes Para ver cómo funciona podemos ejecutar¹²:

```
$ftp localhost
Connected to guadalinux.
220 guadalinux FTP server (Version wu-
2.6.2(1) Sat Aug 21 20:26:00 UTC 2004) ready.
Name (localhost:root): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password:
230-Welcome, archive user anonymous@guadalinux !
230-
230-The local time is: Sat Mar 5 14:06:13 2005
230-
230-This is an experimental FTP server. If have any unusual problems,
230-please report them via e-mail to <root@guadalinux>.
230-
230-If you do have problems, please try using a dash (-) as the first
230-character of your password -- this will turn off the continuation
230-messages that may be confusing your FTP client.
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 24
d--x--x--x 2 0 4096 Mar 5 13:03 bin
d--x--x--x 2 0 4096 Mar 5 13:03 dev
d--x--x--x 2 0 4096 Mar 5 13:03 etc
d--x--x--x 2 0 4096 Mar 5 13:03 lib
dr-xr-xr-x 3 0 4096 Mar 5 13:03 pub
-rw-r--r-- 1 0 346 Mar 5 13:03 welcome.msg
226 Transfer complete.
ftp> bye
221-You have transferred 0 bytes in 0 files.
221-Total traffic for this session was 1104 bytes in 1 transfers.
```

¹²Si no nos sentimos cómodos desde la línea de comandos, es mejor usar los programas: mc, gftp.

```
221-Thank you for using the FTP service on guadalinux.  
221 Goodbye.
```

El demonio FTP se configura en Fedora¹³ a través del fichero `/etc/xinetd.d/wu-ftp`¹⁴ y del fichero¹⁵ `/etc/ftppaccess`, analicemos algunas de las entradas de este fichero:

```
# Denegar el acceso a los usuarios/grupos indicados:  
# con uid/gid menor que 99 o mayor que 65534  
# al resto se les permite el acceso  
deny-uid%-99%65534-  
deny-gid%-99%65534-  
allow-uid ftp  
allow-gid ftp  
  
# Se define la clase all como aquella formada por los usuarios  
# del sistema, los usuarios reales con cuentas de invitados  
# y el resto para todas las IP posibles (*)  
class all real,guest,anonymous *  
  
# Correo del administrador  
email root@localhost  
  
# Si hay 5 intentos fallidos se cierra la conexión  
loginfails 5  
  
# Si un usuario se registra o cambia de directorio se le  
# avisa (si existen) de ficheros README  
readme README* login  
readme README* cwd=*  
  
# Mensaje de bienvenida (hay que crearlos)  
message /welcome.msg login  
message .message cwd=*  
  
# Permite que se compriman/empaqueten ficheros a todos los usuarios  
compress yes all  
tar yes all  
  
# Opciones no permitidas a los usuarios que se listan  
  
chmod no guest,anonymous  
delete no anonymous  
overwrite no anonymous  
rename no anonymous  
  
# Se registran en /var/log/xferlog las transferencias de los  
# usuarios (separados por comas), tanto de carga como de descarga.  
log transfers anonymous,guest,real inbound,outbound  
  
# Si se va a cerrar el servidor se avisa a los usuarios conectados de ello  
# según las directrices del fichero especificado (man ftpshut)  
  
shutdown /etc/shutmsg
```

¹³En Debian se usa `/etc/inetd.conf` si está controlado por `inetd`

¹⁴Hay que comprobar si está inactivo (`disable=yes`), en ese caso poner `disable=no` y hacer que `xinetd` relea la configuración.

`/etc/init.d/xinetd reload`

¹⁵`/etc/wu-ftpd/ftppaces` en Debian.

```
# Si se conecta un usuario anónimo (anonymous) tendrá que introducir
# como contraseña una dirección de correo según esa norma
# es decir, nombre@host.dominio
passwd-check rfc822 warn
```

2.1.3. Una solución más segura

Como ya hemos visto, estas utilidades tienen un problema muy serio, su falta de seguridad. Si no se transmiten los datos por la red de forma segura, cualquier intruso puede interceptar nuestros datos y utilizarlos de forma fraudulenta. Incluso en el caso de telnet, lo que puede obtener el intruso es nuestro usuario y password del sistema además del contenido de la comunicación y a partir de ahí, intentar obtener la clave de root y el control total de nuestra máquina..

La solución a este problema es utilizar un protocolo alternativo denominado SSH¹⁶. Este protocolo cifrará los datos antes de pasarlos a la red, descifrándolos cuando lleguen a su destino. El procedimiento de cifrado asegura que el intruso que capture los datos será incapaz de descifrarlos y verlos. El resultado es un cifrado transparente, ya que el usuario no tiene que realizar ningún proceso previo con los datos que va a transmitir.

El protocolo SSH tiene una arquitectura de cliente/servidor. El servidor SSH es un programa, ejecutado normalmente con permisos de administrador, encargado de aceptar y rechazar las conexiones entrantes a la máquina. Por otro lado, el programa cliente SSH permite a un usuario hacer peticiones desde una máquina remota a la que tiene el servidor SSH activo.



A pesar del concepto shell que aparece en el nombre del protocolo, no hay que confundir SSH con una shell o intérprete de comandos. El funcionamiento de SSH se basa en establecer un canal de comunicación seguro con un ordenador remoto. A partir de ahí se pueden ejecutar aplicaciones, entre ellas una shell.

Nos centraremos en el uso de la implementación libre OpenSSH, siendo los ficheros y comandos que utilicemos los correspondientes a esta implementación. Dentro de los clientes que proporciona esta implementación tenemos `ssh` (conexión segura), `scp` (copia de ficheros segura) y `slogin` (login seguro), que sustituyen a `rsh`, `rcp` y `rlogin`:

Login remoto. Supongamos que tenemos cuentas de usuario en distintas máquinas que se encuentran en internet. Normalmente realizamos la conexión con telnet desde nuestro ordenador personal. Además de transmitir el usuario/clave por la red en formato texto plano, toda la información referente a la sesión que hayamos abierto será legible por cualquier intruso que se encuentre escuchando en la red. Al utilizar un cliente SSH nos autenticaremos en el sistema remoto utilizando una conexión cifrada, el servidor SSH nos permitirá el acceso y toda la sesión es cifrada. Los datos viajarán por la red encriptados entre el cliente y el servidor. Al realizarse este proceso de forma transparente, no existirán diferencias entre el uso de telnet y el de un cliente SSH¹⁷.

```
ssh -l legolas 172.26.0.40
```

Soporta de forma completa la ejecución de aplicaciones en el entorno X. Cuando un cliente se conecta a un servidor SSH intercambian claves de cifrado y a partir de ahí el servidor autentica al cliente, bien con RSA o bien mediante contraseña. Cuando se ha iniciado la conexión, el servidor lanza un servidor X ficticio de forma que las aplicaciones que se ejecutan en el servidor SSH se conectan al servidor X ficticio y de "forma segura" SSH reenvía los datos

¹⁶Existen varias versiones de SSH (SSH-1 y SSH-2), pero para simplificar nos centraremos en el uso de OpenSSH que es una implementación libre que contempla las dos versiones. Aparte, telnet y ftp están incorporando SSL a su funcionamiento.

¹⁷También podemos escribir
`ssh legolas@172.126.0.40`

al servidor X real. Es más seguro y fácil de usar que telnet y para ejecutar una aplicación gráfica (una vez realizada la conexión usando ssh) sólo hay que ejecutar el programa desde la xterm que estemos usando para la conexión.

Ejecución remota de comandos. Supongamos que queremos ejecutar un comando de forma remota en una máquina. En caso de utilizar rsh los datos resultantes de la ejecución del comando viajarán en claro por la red. Mediante SSH la salida que obtenemos es idéntica a si utilizásemos rsh, pero con la diferencia que estos datos estarán ocultos a cualquier intruso que escucha en la red.

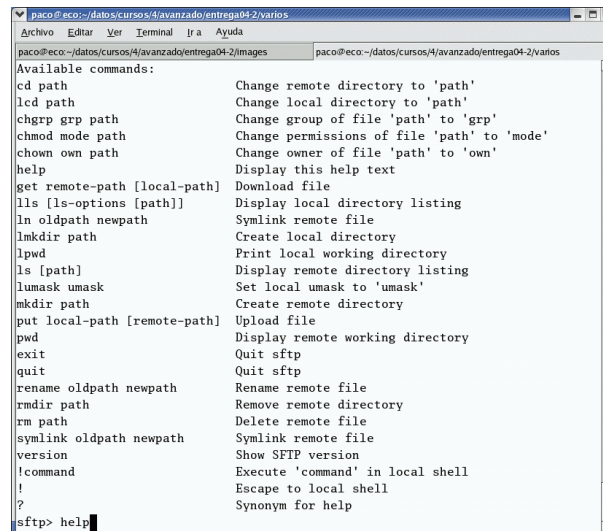
```
ssh 172.26.0.40 /usr/bin/who
```

Transferencia de ficheros. Supongamos que queremos transferir un fichero entre la máquina en la que nos encontramos y un servidor remoto, conteniendo dicho fichero información confidencial que no deseamos que nadie obtenga. Tradicionalmente utilizaríamos las utilidades ftp o rcp, pero cualquier intruso escuchando en nuestra red podría obtener estos datos. Mediante SSH el fichero se puede transmitir con total seguridad, de forma que se cifra antes de salir del origen y se descifra una vez llegado a su destino.

```
scp confidencial.txt legolas@172.26.0.40
```

En general, el comando que usaremos es

```
sftp legolas@172.26.0.1
```



```

paco@eco:~/datos/cursos/4/avanzado/entrega04-2/varios
Archivo Editar Ver Terminal Ir a Ayuda
paco@eco:~/datos/cursos/4/avanzado/entrega04-2/images paco@eco:~/datos/cursos/4/avanzado/entrega04-2/varios
Available commands:
cd path                Change remote directory to 'path'
lcd path               Change local directory to 'path'
chgrp grp path        Change group of file 'path' to 'grp'
chmod mode path       Change permissions of file 'path' to 'mode'
chown own path        Change owner of file 'path' to 'own'
help                  Display this help text
get remote-path [local-path] Download file
lls [ls-options [path]] Display local directory listing
ln oldpath newpath    Symlink remote file
mkdir path            Create local directory
lpwd                  Print local working directory
ls [path]             Display remote directory listing
lumask umask          Set local umask to 'umask'
mkdir path            Create remote directory
put local-path [remote-path] Upload file
pwd                   Display remote working directory
exit                  Quit sftp
quit                  Quit sftp
rename oldpath newpath Rename remote file
rmdir path            Remove remote directory
rm path               Delete remote file
symlink oldpath newpath Symlink remote file
version               Show SFTP version
!command              Execute 'command' in local shell
!                      Escape to local shell
?                      Synonym for help
sftp> help

```

e iniciaremos una comunicación, tipo ftp segura¹⁸ (*secure ftp*). Podemos realizar conexiones sftp en modo gráfico con el programa **gftp**:

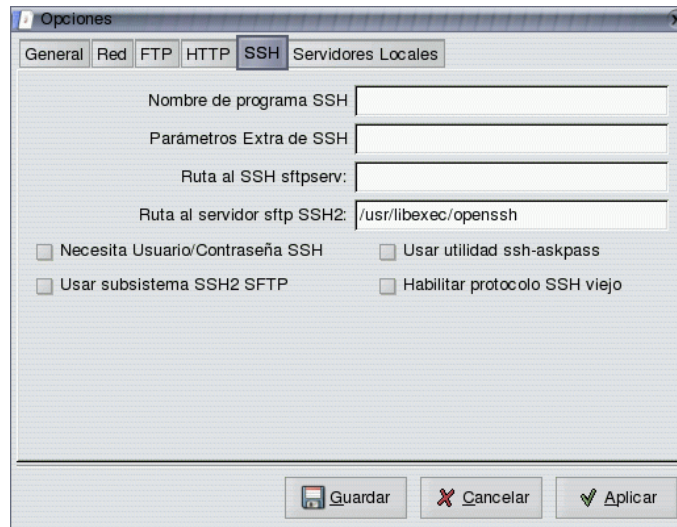
- Para que funcione el cliente gftp en modo SSH, podemos instalar el paquete **ssh-askpass**, que gestionará la autenticación.
- Otra forma sería, para trabajar con SFTP, cambiar la configuración del programa, para eso pulsamos en el menú principal sobre **F**TP y en la ventana que aparece sobre **Opciones**



¹⁸Los comandos disponibles son similares a los del ftp.

Podemos optar por una de la opciones

1. marcar la casilla **Usar subsistema SSH2 SFTP**, o bien
2. pestaña **SSH** y en el último campo escribimos `/usr/libexec/openssh`



- Si deseamos conectar con un servidor con Guadalinex se escribe `/usr/lib`

2.2. La Criptografía llega en nuestra ayuda

Con el uso de Internet y la necesidad de proteger las comunicaciones a través de redes de comunicación inseguras, la protección de la información se transforma en una necesidad y con ello se populariza la criptografía. Es necesario manejar herramientas que nos proporcionen un elevado nivel de seguridad cuando utilizamos Internet. Se vuelven comunes conceptos matemáticos como cifrado, descifrado, criptoanálisis, firma digital, Autoridades de Certificación, que aunque complejos en sus fundamentos, deben hacerse asequibles en su uso.

Definimos la criptografía como una rama inicial de las Matemáticas y en la actualidad muy difundida en la Informática y la Telemática, que utiliza métodos y técnicas con el objeto principal de cifrar y/o proteger un mensaje o archivo por medio de un algoritmo, usando una o más claves¹⁹. Esto da lugar a diferentes tipos de sistemas de cifrado que permiten asegurar cuatro aspectos de la seguridad informática: la confidencialidad, la integridad, la disponibilidad y el no repudio de emisor y receptor.

Confidencialidad: Los componentes del sistema serán accesibles sólo por aquellos usuarios autorizados.

Integridad: Los componentes del sistema sólo pueden ser creados y modificados por los usuarios autorizados.

Disponibilidad: Los usuarios deben tener disponibles todos los componentes del sistema cuando así lo deseen.

No Repudio: Asociado a la aceptación de un protocolo de comunicación entre emisor y receptor (cliente y servidor) normalmente a través del intercambio de sendos certificados digitales de autenticación. Se habla entonces de No Repudio de Origen y No Repudio de Destino, forzando a que se cumplan todas las operaciones por ambas partes en una comunicación.

¹⁹Para proteger un mensaje M se cifra con la función f , dando lugar al mensaje cifrado C $f(M) = C$. El descifrado consiste en aplicar la función inversa al mensaje cifrado para obtener el mensaje original $f^{-1}(C) = M$



Los sistemas criptográficos modernos²⁰ se pueden clasificar en simétricos (o de clave privada) y asimétricos (o de clave pública).

- Criptosistemas simétricos: Existirá una única clave secreta que deben compartir emisor y receptor. Con la misma clave se cifra y se descifra por lo que la seguridad reside sólo en mantener dicha clave en secreto.
- Criptosistemas asimétricos: Cada usuario crea un par de claves, una privada y otra pública, inversas dentro de un cuerpo finito. Lo que se cifra en emisión con una clave, se descifra en recepción con la clave inversa. La seguridad del sistema reside en la dificultad computacional de descubrir la clave privada a partir de la pública. Para ello, usan funciones matemáticas de un solo sentido con trampa.

Son funciones matemáticas de un solo sentido²¹ y que nos permiten usar la función en sentido directo o de cálculo fácil para cifrar y descifrar (usuarios legítimos) y fuerza el sentido inverso o de cálculo difícil para aquellos impostores, hackers, etc. que lo que desean es atacar o criptoanalizar el mensaje cifrado.

Pongamos como ejemplo el problema de la factorización de números grandes. El cálculo directo es fácil. Dados dos números p y q , por muy grandes que sean, hallar $p \cdot q = n$ es fácil, sobre todo para un ordenador. Sin embargo, el cálculo inverso, que es la descomposición en factores de un número grande $n = p \cdot q$, para el caso de p y q primos entre sí, es computacionalmente complejo.

¿Por qué utilizamos dos sistemas criptográficos distintos? Porque no existe el sistema perfecto para todos los usos. Los sistemas de clave pública son muy lentos cifrando, pero tienen firma digital y la gestión de claves se puede utilizar cuando podemos tener miles o millones de usuarios. Los sistemas de clave secreta son muy rápidos cifrando, pero no tienen firma digital y mantener la clave secreta es complicado cuando tenemos más de unos cuantos usuarios.

La solución es utilizar cada sistema para lo que es adecuado. Así, para el cifrado de la información se utilizarán sistemas de clave secreta y para la firma digital y el intercambio de claves de sesión, se utilizarán sistemas de clave pública.

➔ Para practicar:

Utilizaremos la herramienta criptográfica OpenSSL, que es algo así como una “navaja suiza” que hace de todo con respecto a los procedimientos criptográficos.

- Veamos un ejemplo de criptografía simétrica. Necesitamos un mensaje que cifrar, para ello generamos un fichero de texto, que llamamos `fichero.txt`, con el siguiente contenido:

```
En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que...
```

```
...y os hacen merecedora del merecimiento que merece la vuestra grandeza.
```

Utilizaremos el algoritmo CAST (otros serían el 3DES, Blowfish...) para cifrar el fichero. Para ello nos pedirá una clave²², que le proporcionaremos.

```
[root@unit3 cep]# openssl cast -in fichero.txt -out fichrc5.bin
enter cast5-cbc encryption password:
Verifying - enter cast5-cbc encryption password:
```

²⁰Para distinguirlos de los sistemas utilizados en la antigüedad basados en rotaciones, permutaciones... del mensaje y basados principalmente en el secreto del algoritmo de cifrado

²¹Se llaman así porque la ejecución de la función $f(M) = C$ es siempre fácil, pero la ejecución de la función inversa $f^{-1}(C) = M$ es difícil salvo que se tenga la trampa.

²²Con ella formará la clave privada



Si deseamos ver el contenido descifrado de nuevo, tendremos que aplicar el algoritmo al fichero cifrado y proporcionar la clave. Vemos que la misma clave que sirve para cifrar, es utilizada para descifrar.

```
[root@unit3 cep]# openssl cast -d -in fichrc5.bin
enter cast5-cbc decryption password:
En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha
mucho tiempo que...
...y os hacen merecedora del merecimiento que merece la vuestra grandeza
.
```

- Ejemplo de criptografía asimétrica:

Generaremos la pareja de claves (clave privada y clave pública) mediante el algoritmo asimétrico RSA. En este caso existen dos claves, una pública y otra privada, con la propiedad de que lo cifrado con una de ellas sólo puede ser descifrado con la otra.

```
[root@unit3 cep]# openssl genrsa -out clavesrsa.pem 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
```

Visualizamos las claves generadas. Se encuentran en formato PEM, que es una codificación en base64, formada solamente por caracteres ASCII, del formato binario ASN.1.

```
[root@unit3 cep]# more clavesrsa.pem
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC+9Z8yUuIrOMfIB9RUUuu4KoKZCinUfIU5MJsdKHmQvVuRJ+oq
A943VYGYIZOEoEv0EWntkmsz0bpB53nY48bDHbSK1DbqrpY3AytrfdpJbJtwl6L
E8IudlFe5atX0u7mQBzi2UhDoH3anT8xk3YAvSpMyghi9JQtOYW3V/JdLwIDAQAB
AoGAMATT4300K+o7WTteyVWQsN7/uDw2CJk8FQWD+/ERoLP9MqM23xnZ51A5JmL
EC6+6sD2QidvjbhPfgMXQNMGR9rJjqF545baG7at35fkKfllBvnN1CZUBjqRv94P
T3GhbTBMIZP+ozBY1XLLA52MBK+M5fct6Hv/n/ic2GbprQECQQDuw6NKgb6qee0W
I+bhYswOYkLBQsHRjd8oIiDDr8OUtrf7kkaUdVBX/MzslCYfh7iYiHps8ZNI5cfQ
MsQBKAHNAkEAzL6NZMp3z3rHOJZpQdW0G43p528UgLXawqCqQJD+G6iKc18WDaTb
r/XHa5DZHZNZ28jB5Xq7IdsyVP4ubx2O6wJAasg4OVQ5b4jEDcjUzyw6UpyDemde
w1eN3CcXPC1ZbSMiuXI7+p1U52T6STwgqH7JUf6Hsj2AP+ZyLJznqBS6aQJAY3iw
eSdzgh4gaWRvcp1lm18FISBQYcYoTYtgPDwg79+hE7OCBLwKKzgFgJpeVgl4AHJ
MhROlKRIT8KuDI3vwQJBAOk0VQPN0eLXEPQ9Zvvr5sO3YTIA3iChWuA+yUt97ZRC
/5PZNS/EY7GRpEECN3yHOy1YPVe2R6hp2/Pndb9ZQA8=
-----END RSA PRIVATE KEY-----
```

Si queremos mostrar las claves generadas en formato legible.

```
[root@unit3 cep]# openssl rsa -in clavesrsa.pem -noout -text
Private-Key: (1024 bit)
modulus:
00:be:f5:9f:32:52:e2:2b:38:c7:e2:07:d4:54:52:
eb:b8:2a:82:99:0a:29:d4:7e:55:39:30:9b:1d:28:
79:90:bd:5b:91:27:ea:2a:03:de:37:55:81:98:21:
93:84:a0:4b:f4:11:69:ed:92:6b:33:d1:ba:41:e7:
79:d8:e3:c6:c3:1d:b4:8a:d4:36:ea:ae:9c:82:dc:
0c:ad:ad:f7:69:25:b2:6d:c2:5e:8b:13:c2:2e:76:
51:5e:e5:ab:57:d2:ee:e6:40:1c:e2:d9:48:43:a0:
7d:da:9d:3f:31:93:76:00:bd:2a:4c:ca:08:62:f4:
94:2d:39:85:b7:57:f2:5d:2f
```

```

publicExponent: 65537 (0x10001)
privateExponent:
30:04:d3:e3:73:b4:2b:ea:3b:59:3b:5e:c9:55:90:
b0:de:ff:b8:3c:36:08:99:3c:15:05:83:fb:f1:11:
a0:b3:fd:32:a3:36:df:19:d9:e7:50:39:26:68:4b:
10:2e:be:ea:c0:f6:42:27:6f:8d:b8:4f:7e:03:17:
40:d3:06:47:da:c9:8e:a1:79:e3:96:da:1b:b6:ad:
df:97:e4:29:f2:25:06:f9:cd:d4:26:54:06:3a:91:
bf:de:0f:4f:71:a1:6d:30:4c:21:93:fe:a3:30:58:
d5:72:cb:03:9d:8c:04:af:8c:e5:f7:2d:e8:7b:ff:
9f:f8:9c:d8:66:e9:ad:01
prime1:
00:ee:c3:a3:4a:81:be:aa:79:ed:16:23:e6:e1:62:
cc:0e:62:42:c1:42:c1:d1:8d:df:28:22:20:c3:af:
c3:94:b6:b7:fb:92:46:94:75:50:57:fc:cc:ec:94:
26:1f:1f:b8:98:88:7a:6c:f1:93:49:e5:c7:d0:32:
c4:01:28:01:cd
prime2:
00:cc:be:8d:64:ca:77:cf:7a:c7:38:96:69:41:d5:
b4:1b:8d:e9:e7:6f:14:80:b5:da:c2:a0:aa:40:90:
fe:1b:a8:8a:73:5f:16:0d:a4:db:af:f5:c7:6b:90:
d9:1d:99:d9:db:c8:c1:e5:7a:bb:21:db:32:54:fe:
2e:6f:1d:8e:eb
exponent1:
6a:c8:38:39:54:39:6f:88:c4:0d:c8:d4:b3:2c:3a:
52:9c:83:7a:67:5e:c3:57:8d:dc:27:17:3c:2d:59:
6d:23:22:b9:72:3b:fa:9d:54:e7:64:fa:49:3c:20:
a8:7e:c9:51:fe:87:b2:3d:80:3f:e6:72:2c:9c:e7:
a8:14:ba:69
exponent2:
63:78:b0:79:27:64:ce:08:78:81:a5:91:bd:ca:75:
96:6d:7c:16:54:81:41:87:18:a1:36:2d:80:f0:f0:
83:bf:7e:84:4e:ce:08:12:f0:28:ac:e0:16:02:69:
79:58:25:e0:01:c9:32:14:4e:96:44:48:4f:c2:ae:
0c:8d:ef:c1
coefficient:
00:e9:34:55:03:cd:d1:e2:d7:10:f4:3d:66:fc:6b:
e6:c3:b7:61:32:00:de:20:a1:5a:e0:3e:c9:4b:7d:
ed:94:42:ff:93:d9:35:2f:c4:63:b1:91:a4:41:02:
37:7c:87:3b:2d:58:3d:57:b6:47:a8:69:db:f3:e7:
75:bf:59:40:0f}

```

Destacar que la clave pública está formada por el módulo (modulus) y el exponente público (publicExponent), y que la clave privada es el exponente privado (privateExponent).■

2.3. SSH como cliente

Tal como acabamos de ver, SSH parte de una idea simple, aunque los componentes que nos ayudan a llegar al objetivo son complejos, como veremos más adelante. Para empezar partiremos de la parte cliente para empezar a profundizar en lo que SSH nos aporta en la seguridad de nuestras comunicaciones.

2.3.1. Sesiones remotas con SSH

Uno de los usos más extendidos es en el uso de sesiones remotas. Veamos con más atención cómo nos conectamos a un servidor remoto con SSH.


```
The authenticity of host 'fedora2 (172.26.0.41)' can't be established.
RSA key fingerprint is a4:78:71:06:60:ea:a9:1b:ed:0a:5e:80:c7:bc:d7:3b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'fedora2' (RSA) to the list of known hosts.
legolas@fedora2's password:
[legolas@fedora2 legolas]$
```

La primera vez que nos conectamos a una máquina remota aparecerá el mensaje anterior. Está relacionado con una función de seguridad de SSH denominada *known hosts* (hosts conocidos). Supongamos que un intruso quiere obtener la clave de entrada a una máquina y sabe que estamos utilizando SSH, por lo que no puede obtener este dato escuchando en la red. En lugar de eso, suplantaré el nombre de la máquina por otra que él controle en la cual hay instalada una versión modificada de SSH. Le bastará con esperar a que nos conectemos a la máquina para obtener la clave que busca.

El mecanismo SSH *known hosts* previene estos ataques. Cuando un cliente SSH y un servidor establecen una conexión, cada uno de ellos comprueba la identidad del otro. No sólo el servidor autentica al cliente mediante la clave, el cliente también autentica al servidor mediante una clave pública. Básicamente, cada servidor SSH tiene un identificador único y secreto, llamado *host key*, para identificarse frente a los clientes que se conectan. La primera vez que nos conectamos a un servidor, la parte pública de la *host key* se copia en nuestra cuenta local (asumiendo que respondemos *yes*). Cada vez que nos conectemos a este servidor, el cliente SSH comprobará la identidad del servidor remoto con esta clave pública. Dicha clave pública, así como la del resto de máquinas con las que nos vayamos conectando se encuentra guardada en `$HOME/.ssh/known_hosts`

```
fedora2 ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAxT0bamLgJbuF8cMRoLU1
HPehrK5iPNyiEktZ2ATh85Tq/+7pIGHwYcmiZcS13X4ppR41G1uTAnujB0
/PuX3JNDql0qFlrzN1857DHQuVI2+bfEjNSsjZ2z/u7BQy188Sqyfn3gpd
nm5fgwtMkLBb+MifWZi04xc+OhBFWoKFNj0=
```

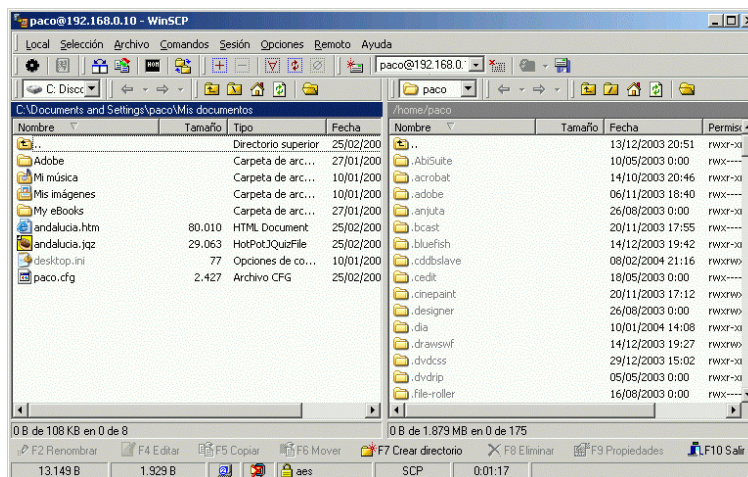
En caso de que el servidor al que nos conectemos no tenga una clave pública coincidente con la que tenemos almacenada en nuestra cuenta, aparecerá el siguiente mensaje:

```
[hugo@fedora hugo]$ ssh -l legolas fedora2
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@           WARNING: POSSIBLE DNS SPOOFING DETECTED!           @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The RSA host key for fedora2 has changed,
and the key for the according IP address 172.26.0.41
is unchanged. This could either mean that
DNS SPOOFING is happening or the IP address for the host
and its host key have changed at the same time.
Offending key for IP in /home/hugo/.ssh/known_hosts:1
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@           WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!           @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
a4:78:71:06:60:ea:a9:1b:ed:0a:5e:80:c7:bc:d7:3b.
Please contact your system administrator.
Add correct host key in /home/hugo/.ssh/known_hosts to get rid of this
message.
Offending key in /home/hugo/.ssh/known_hosts:3
RSA host key for fedora2 has changed and you have requested strict checking.
Host key verification failed.
```

El hecho de que la clave pública que proporciona el servidor no coincida con la que tenemos almacenada, puede deberse a varias causas. Por ejemplo, el servidor remoto puede haber cambiado la *host key* por algún motivo especial. Así, el ver esta advertencia no quiere decir necesariamente que el sistema ha sido “hackeado”²³. El administrador del sistema al que vamos a acceder será el que mejor nos pueda informar al respecto.

Una información añadida, clientes para Windows (gratuitos):

- El “genuino” para ssh, sftp, scp en modo comando:
[putty http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)
- Para sftp en modo gráfico, WinSCP:
<http://winscp.sourceforge.net/eng/>



2.3.2. Autenticación por clave criptográfica

Hasta ahora, los ejemplos que hemos visto se basaban en la autenticación del cliente por medio de usuario/clave. Sin embargo esto tiene algunos inconvenientes, ya que las claves más seguras son las más difíciles de recordar. Además, el sistema operativo únicamente permite una clave por usuario, lo que en el caso de cuentas compartidas (por ejemplo root) presenta dificultades en el cambio de clave ya que debe comunicarse a todo el que utilice esta cuenta.

Para solucionar este problema, SSH proporciona un mecanismo de autenticación basado en clave pública.

Introducción a las claves

Cuando estamos hablando de clave, nos referimos a ella como una identidad digital. Es una cadena de datos binarios que nos identifica de forma unívoca, en el caso de SSH nos identificará ante un servidor.

Una clave de identidad tendrá dos partes:

Clave privada. Es la parte que sólo el usuario al que identifica debe tener. Es utilizada por SSH para probar la identidad ante un servidor.

Clave pública. Como su propio nombre indica, es pública y podemos distribuirla a todos aquellos sistemas que queramos que nos identifiquen. SSH la utiliza durante la autenticación para identificarnos ante un servidor remoto.

²³Atacado por hackers (o por crackers, que son aún más malvados) que sustituyen los programas por otros con puertas traseras y barrido de pistas.

Veamos cómo sería la secuencia entre un cliente y un servidor SSH para comprender un poco mejor estos conceptos:

1. El cliente dice “Hola servidor, me gustaría conectarme por SSH a una de tus cuentas, la del usuario legolas”.
2. El servidor dice “Bien, pero primero te desafiaré a que pruebes tu identidad” y el servidor envía datos conocidos como desafío²⁴ al cliente.
3. El cliente dice “Acepto tu desafío. Aquí tienes una prueba de mi identidad. La hice yo mismo mediante algoritmos matemáticos usando tu desafío y mi clave privada”. Esta respuesta al servidor se conoce como *authenticator*.
4. El servidor dice “Gracias por el autenticador. Ahora examinaré la cuenta legolas para ver si puedes utilizarla”. Lo que realmente hace el servidor es comprobar las claves públicas de la cuenta legolas para ver si el autenticador concuerda con alguna de ellas. En caso de que concuerden, el servidor dará su consentimiento para el acceso al sistema. En otro caso, la autenticación falla.

A continuación veremos con más detalle los comandos y ficheros implicados en este proceso.

Generación de claves

Para usar la autenticación criptográfica es condición necesaria e indispensable la generación de la identidad digital, es decir, la pareja clave pública/clave privada. Para generar esta pareja de claves utilizaremos la utilidad `/usr/bin/ssh-keygen`.

```
ssh-keygen [options]
```

- `-b bits` Número de bits en la clave que se crea
- `-f filename` Nombre del fichero donde se almacenará la clave
- `-l` Enseña la marca del fichero de clave
- `-p` Cambia la palabra de paso del fichero de clave privada
- `-y` Lee la clave privada e imprime la pública
- `-t type` Tipo de clave que se creará (dsa o rsa)
- `-N phrase` Proporciona una nueva palabra de paso

Empezaremos viendo cómo generar una pareja de claves pública y privada:

```
[legolas@fedora legolas]$ ssh-keygen -t dsa -b 2048
Generating public/private dsa key pair.
Enter file in which to save the key (/home/legolas/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/legolas/.ssh/id_dsa.
Your public key has been saved in /home/legolas/.ssh/id_dsa.pub.
The key fingerprint is:
c0:cb:fe:5f:2b:d2:05:06:90:69:a7:1d:89:71:75:46 legolas@fedora.elpiso.es
```

Una vez creada, podemos realizar varias operaciones. Empezaremos mostrando la marca del fichero de clave:

²⁴Proviene del término inglés *challenge*

```
[legolas@fedora legolas]$ ssh-keygen -t dsa -l
Enter file in which the key is (/home/legolas/.ssh/id_dsa):
2048 c0:cb:fe:5f:2b:d2:05:06:90:69:a7:1d:89:71:75:46 /home/legolas/.ssh/
id_dsa.pub
```

En caso de que queramos cambiar la palabra de paso:

```
[legolas@fedora legolas]$ ssh-keygen -t dsa -p
Enter file in which the key is (/home/legolas/.ssh/id_dsa):
Enter old passphrase:
Key has comment '/home/legolas/.ssh/id_dsa'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

Como acabamos de ver, es necesario conocer la antigua palabra de paso. Sin embargo, existe otra forma de cambiarla sin necesidad de proporcionarla, lo cual es especialmente útil en el caso que se nos haya olvidado:

```
[legolas@fedora legolas]$ ssh-keygen -t dsa -N "elfo del bosque"
Generating public/private dsa key pair.
Enter file in which to save the key (/home/legolas/.ssh/id_dsa):
/home/legolas/.ssh/id_dsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/legolas/.ssh/id_dsa.
Your public key has been saved in /home/legolas/.ssh/id_dsa.pub.
The key fingerprint is:
72:ee:2e:0b:45:e5:c8:a4:10:5f:e8:d3:e6:59:94:c2 legolas@fedora.elpiso.es
```

2.3.3. Y ahora que tenemos las claves ... ¿qué hacemos con ellas?

Acabamos de ver cómo crear la pareja de claves y cómo obtener o modificar su información, sin embargo, ¿qué utilidad tiene esto?

Cuando las claves son utilizadas para la autenticación, el sistema operativo de la máquina mantiene la asociación entre el usuario y la contraseña. Para las claves criptográficas podemos establecer una asociación similar de forma manual. Después de crear el par de claves con `ssh-keygen` en el nodo local, deberemos instalar la clave pública en la cuenta del servidor remoto.

Volvamos al ejemplo anterior. Deberemos instalar la clave pública dentro de la cuenta `legolas` en el servidor remoto. Esto se hace editando el fichero `$HOME/.ssh/authorized_keys2` y añadiéndole la clave pública.

```
[legolas@fedora .ssh]$ ssh-keygen -t dsa -N "elfo del bosque"
Generating public/private dsa key pair.
Enter file in which to save the key (/home/legolas/.ssh/id_dsa):
/home/legolas/.ssh/id_dsa already exists.
Overwrite (y/n)? y
Your identification has been saved in /home/legolas/.ssh/id_dsa.
Your public key has been saved in /home/legolas/.ssh/id_dsa.pub.
The key fingerprint is:
28:83:99:dc:da:ed:e3:45:be:d9:c9:bb:0a:8d:43:1c legolas@fedora.elpiso.es
[legolas@fedora .ssh]$ scp id_dsa.pub fedora2:.ssh/fedora.pub
legolas@fedora2's password:
id_dsa.pub                               100% 614    342.2KB/s   00:00
```

Una vez copiada la clave pública en la máquina remota, en nuestro caso la máquina **fedora2**, creamos el fichero **\$HOME/.ssh/authorized_keys2** y le damos los permisos necesarios²⁵:

```
[legolas@fedora2 .ssh]$ cat fedora.pub >> authorized_keys2
[legolas@fedora2 .ssh]$ ls -l
total 16
-rw-rw-r-- 1 legolas legolas 614 feb 14 06:57 authorized_keys2
-rw-r--r-- 1 legolas legolas 614 feb 14 06:57 fedora.pub
-rw----- 1 legolas legolas 668 feb 14 06:54 id_dsa
-rw-r--r-- 1 legolas legolas 615 feb 14 06:54 id_dsa.pub
[legolas@fedora2 .ssh]$ chmod 644 authorized_keys2
[legolas@fedora2 .ssh]$ ls -l
total 16
-rw-r--r-- 1 legolas legolas 614 feb 14 06:57 authorized_keys2
-rw-r--r-- 1 legolas legolas 614 feb 14 06:58 fedora.pub
-rw----- 1 legolas legolas 668 feb 14 06:54 id_dsa
-rw-r--r-- 1 legolas legolas 615 feb 14 06:54 id_dsa.pub
```

A partir de ahora, una conexión que se realiza con la cuenta de usuario **legolas** desde la máquina **fedora** a la máquina **fedora2** utilizará el método de autenticación por clave criptográfica.

```
[legolas@fedora .ssh]$ ssh fedora2
Enter passphrase for key '/home/legolas/.ssh/id_dsa': elfo del bosque
[legolas@fedora2 legolas]$
```

Como acabamos de ver, el proceso resultante es similar a cuando utilizamos autenticación por usuario/clave. La única diferencia, a simple vista, es que hemos sustituido la clave por la palabra de paso de la clave criptográfica. Sin embargo, si examinamos el proceso con mayor profundidad vemos que en el caso de la clave criptográfica lo único que viaja por la red es la palabra de paso. Con la autenticación criptográfica, la palabra de paso sirve únicamente para descryptar la clave pública y crear el autenticador.

De esta forma, la autenticación mediante clave pública es más segura que la autenticación por clave. Serán necesarios dos componentes secretos (fichero con clave criptográfica y palabra de paso) y será necesario capturar las dos para poder tener acceso al sistema. En el caso del fichero con la clave criptográfica, el intruso tendría que tener acceso físico al mismo.

Podemos seguir sacando partido a este tipo de autenticación. Como hemos visto, al generar las claves criptográficas hemos introducido una palabra de paso, sin embargo no es obligatorio. Dejándola en blanco lograremos acceder a la máquina remota sin necesidad de introducir ninguna palabra de paso. Esto sin embargo presenta un grave problema en lo referente a la seguridad. Si otra persona obtiene nuestra clave privada podrá acceder sin ningún control a los servidores donde hayamos configurado el acceso por clave pública.

2.3.4. El agente ssh

Como acabamos de ver, cada vez que utilizamos la clave pública para acceder a un sistema por ssh o scp es necesario teclear de nuevo la frase de paso. Este proceso puede hacernos pensar que no hay diferencia y que es igual de "cómodo" que el uso de la autenticación por usuario/clave. Sería muy bonito tener un mecanismo por el cual sólo sería necesario introducir la frase de paso una vez y quedaría almacenada para posteriores conexiones. Esta labor la realiza el agente ssh, implementado en la utilidad **ssh-agent**.

El agente almacenará en memoria las claves privadas y proporciona servicios de autenticación a los clientes ssh. Así, cuando estemos en nuestro sistema y deseemos conectarnos a otros, utilizaremos la frase de paso para descryptar la clave privada y a partir de este momento será el agente el que recibe las peticiones de los clientes.

²⁵En caso de que no cambiemos los permisos al fichero **authorized_keys2** con la máscara 644, no lograremos que el proceso de autenticación utilice la clave criptográfica.

En OpenSSH el nombre de este agente es `ssh-agent`. Normalmente ejecutaremos un único `ssh-agent` en nuestra sesión local, antes de ejecutar cualquier cliente `ssh`. Estos clientes se comunican con el agente a través del entorno de procesos, por lo que todos los clientes y procesos de la sesión tendrán acceso al agente. Puede arrancarse el agente desde una sesión como puede verse a continuación:

```
ssh-agent $SHELL
```

donde `SHELL` es la variable de entorno que almacena la shell de login. De esta forma el agente se ejecuta e invoca a un shell definido como proceso hijo. El efecto visual es que aparece otro prompt de shell, pero esta shell tiene acceso al agente.

Una vez que el agente está arrancado podemos empezar a cargar las claves privadas con las que accederemos a otros sistemas. La utilidad que proporciona OpenSSH para este fin es `ssh-add`.

```
ssh-agent $SHELL
sh-2.05b$ ssh-add
Enter passphrase for /home/legolas/.ssh/id_dsa: elfo del bosque
Identity added: /home/legolas/.ssh/id_dsa (/home/legolas/.ssh/id_dsa)
```

Ahora los clientes `ssh` pueden conectarse a los hosts remotos sin la necesidad de teclear de nuevo la frase de paso.

La utilidad `ssh-add` leerá, por defecto, la palabra de paso desde el terminal. Sin embargo, es posible utilizar la entrada estándar para realizar esta entrada de forma no interactiva.

Tal como hemos ejecutado anteriormente `ssh-add` (sin ningún parámetro) añade la clave privada que se encuentra en la localización estándar (`$HOME/.ssh/id_dsa`). En caso de utilizar otro nombre:

```
ssh-add fichero_con_clave_privada
```

Podemos añadir tantas claves como queramos, pudiendo obtener un listado de las mismas con esta misma utilidad:

```
ssh-add -l
```

Igualmente, podemos eliminar alguna de las claves que tiene almacenadas el agente, incluso borrarlas todas:

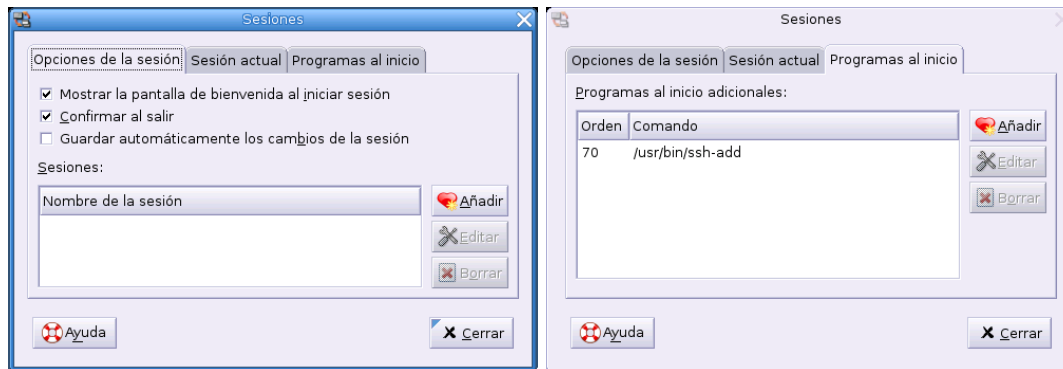
```
ssh-add -d fichero_con_clave_privada
ssh-add -D
```

2.3.5. Uso del agente SSH en GNOME

El procedimiento anterior puede automatizarse. Tanto Fedora como Guadalinex arrancan de forma automática `ssh-agent` al arrancar una sesión X. Solo quedaría que se cargue también en el inicio de sesión la clave privada mediante `ssh-add`.

Lo anterior es posible si modificamos la configuración de la sesión para que `ssh-add` sea uno de los programas que se arranca al inicio de la sesión X del usuario. Para cambiar la configuración de una sesión en Gnome utilizaremos `gnome-session-properties`.

Figura 2.1: Configuración de sesión



Crearemos una nueva entrada dentro de la configuración de la sesión para que se ejecute `ssh-add`. Estableceremos un número de prioridad más alto que cualquiera de los comandos existentes para asegurarnos que se ejecute el último. Mientras más alto el número, más baja será la prioridad. Si tenemos otros programas listados, éste debería tener la prioridad más baja. Para asegurarnos que se ejecute en último lugar le pondremos un número como 70 o superior.

Junto con este cambio en la configuración es necesario instalar también el paquete `ssh-askpass-gnome` que proporcionará una interfaz en la que indicar la frase con la que encriptamos la clave privada.

Guadalinex

```
apt-get install ssh-askpass-gnome
```

Fedora

```
rpm -Uvh openssh-askpass-gnome-3.9p1-7.i386.rpm
```

Quedará almacenada durante toda la sesión Gnome, a menos que la borremos de memoria mediante `ssh-add`, como vimos anteriormente.

2.4. Configuración del servidor SSH

Hasta ahora hemos estado viendo la funcionalidad de SSH desde el punto de vista del cliente, sin preocuparnos de la configuración existente en el servidor remoto. Evidentemente, será necesario tener el servicio de SSH activo en un servidor para poder conectarnos a él mediante SSH.

2.4.1. Instalación

Para disponer de la última versión:²⁶

- En Guadalinex²⁷

```
#apt-get install ssh ssh-askpass
```

²⁶Para activarlo: `/etc/init.d/ssh start`

²⁷El paquete `ssh` está ya instalado, para activar el demonio `sshd` podemos usar

```
#dpkg-reconfigure ssh
```



- Con Fedora²⁸

```
openssh-3.9p1-7
openssh-clients-3.9p1-7
openssh-server-3.9p1-7
```

El último es el que nos permite disponer de un servicio ssh.

2.4.2. Configuración

La configuración del servicio se realiza con el fichero `/etc/ssh/sshd_config`. Para habilitar el uso del servicio SSH para acceder a nuestro servidor no sería necesario hacer ninguna modificación a los valores que vienen definidos por defecto. La mayoría de ellos están comentados, indicando el valor que se está tomando por defecto.

```
#      $OpenBSD: sshd_config,v 1.59 2002/09/25 11:17:16 markus Exp $
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.
# This sshd was compiled with PATH=/usr/local/bin:/bin:/usr/bin
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.
#Port 22
#Protocol 2,1
#ListenAddress 0.0.0.0
#ListenAddress ::
# HostKey for protocol version 1
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key
# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 3600
#ServerKeyBits 768
# Logging
#obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
#SyslogFacility AUTHPRIV
#LogLevel INFO
# Authentication:
#LoginGraceTime 120
#PermitRootLogin yes
#StrictModes yes
#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile      .ssh/authorized_keys
# rhosts authentication should not be used
#RhostsAuthentication no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
```

²⁸Se instalan por defecto, además de los comentados se instalan también

```
openssh-askpass-3.9p1-7
openssh-askpass-gnome-3.9p1-7
```




```
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes
# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#AFSTokenPassing no
# Kerberos TGT Passing only works with the AFS kaserver
#KerberosTgtPassing no
# Set this to 'yes' to enable PAM keyboard-interactive authentication
# Warning: enabling this may bypass the setting of 'PasswordAuthentication'
#PAMAuthenticationViaKbdInt no
#X11Forwarding no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
#KeepAlive yes
#UseLogin no
#UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression yes
#MaxStartups 10
# no default banner path
#Banner /some/path
#VerifyReverseMapping no
# override default of no subsystems
Subsystem      sftp      /usr/libexec/openssh/sftp-server
```

Veamos ahora brevemente el significado de algunos de estos parámetros:

Port Especifica en qué número de puerto el demonio SSH va a escuchar las conexiones entrantes.

ListenAddress Especifica la dirección IP del interfaz de red en el que va a escuchar el demonio SSH.

HostKey Especifica la localización del fichero con la clave privada del nodo

ServerKeyBits Especifica cuántos bits se van a utilizar en la clave del servidor. Estos bits se utilizan cuando el demonio empieza a generar su clave RSA.

LoginGraceTime Especifica el tiempo en segundos que transcurre entre que se realiza una petición al servidor y no se ha conseguido el login satisfactorio al mismo.

KeyRegenerationInterval Especifica un intervalo en segundos que el servidor debe esperar antes de regenerar su clave automáticamente. Es una medida de seguridad para prevenir la descriptación de sesiones capturadas.

PermitRootLogin Especifica si el usuario root puede logarse utilizando ssh.

IgnoreRhosts Especifica si los ficheros rhosts o shosts no deben ser usados en la autenticación.



IgnoreUserKnownHosts Especifica si el demonio SSH debe ignorar el contenido del fichero de usuario `$HOME/.ssh/known_hosts` durante la autenticación `RhostsRSAAuthentication`.

StrictModes Especifica si SSH debe chequear los permisos de usuario en el directorio `$HOME` y el fichero `rhosts` antes de aceptar su entrada en el sistema.

QuietMode Si es sí no hace log de nada.

X11Forwarding Especifica si se permite X11 *forwarding* en el servidor. Tendremos que ponerlo a **yes** si deseamos ejecutar aplicaciones gráficas en una conexión ssh.

PrintMotd Especifica si el demonio SSH debe imprimir el contenido del fichero `/etc/motd` cuando un usuario accede al sistema de forma interactiva.

SyslogFacility Especifica el tipo de log de sistema que va a producir cuando se generen mensajes al sistema desde el demonio SSH.

LogLevel Especifica el nivel de log de sistema que es usado cuando el demonio SSH genera mensajes al sistema.

RhostsAuthentication Especifica si el demonio SSH puede usar la autenticación basada en `rhosts`.

RhostsRSAAuthentication Especifica si el demonio SSH puede usar la autenticación relacionada con autenticación de nodos RSA.

RSAAuthentication Especifica si se permite autenticación RSA.

PasswordAuthentication Especifica si puede utilizarse autenticación basada en usuario/clave para acceder al sistema.

PermitEmptyPasswords Especifica si el servidor permite la entrada en el sistema a cuentas que tienen una clave nula.

AllowUsers Especifica y controla qué usuarios pueden acceder a servicios SSH.

Es recomendable modificar con cuidado este fichero ya que si no tenemos acceso físico a la máquina, una configuración errónea nos dejaría sin acceso remoto a la misma. Igualmente, recordamos también que para que las modificaciones en este fichero tengan efecto será necesario reiniciar el servicio. La forma más cómoda de realizar esto es mediante la utilización del script de arranque del servicio que proporciona la instalación de los paquetes mencionados anteriormente²⁹:

```
/etc/init.d/sshd restart
```

La parada del servicio deja activas las conexiones existentes, por lo que no tenemos que preocuparnos de dejar sin conexión a los usuarios ya conectados en el proceso de parada/arranque del servicio.

²⁹En Guadalinex sustituir `sshd` por `ssh`

Capítulo 3

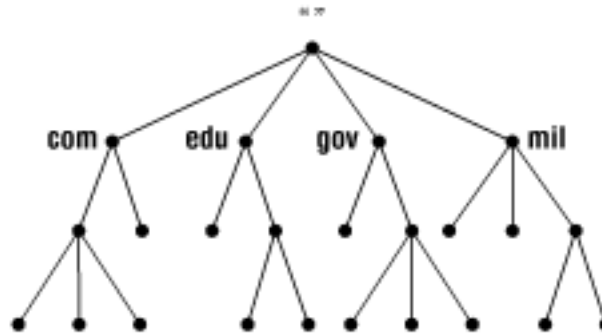
Servidor de nombres DNS

Puede que no conozcas mucho sobre el Sistema de Nombres de Dominio (DNS) –todavía– pero siempre que usas Internet, estás usando el DNS. Cada vez que envías un correo electrónico o navegas por la Web, dependes del DNS. (*DNS and Bind*, PAUL ALBITZ y CRICKET LIU)

Llegó la hora de las direcciones simbólicas. Las direcciones IP han campado a sus anchas y la verdad es que para nosotros son difíciles de recordar y propensas a errores. Donde esté un nombre simple y descriptivo como `thales.cica.es`, que se quiten todas las direcciones IP como su equivalente 172.26.0.2 ¿o era 150.214.22.12? ¡Ah! no, es 150.214.5.10. Véis, nuestra capacidad simbólica es superior a nuestra capacidad de recordar números.

El sistema DNS es una base de datos distribuida. Presenta una jerarquía en la que su parte más alta es el “punto” o raíz y de él cuelgan los dominios de primer nivel (.com, .edu, .es, etc).

DNS database



Su lectura en el orden jerárquico se realiza de derecha a izquierda. Por ejemplo, para la máquina `thales.cica.es`, primero en la jerarquía se encuentra el dominio de primer nivel¹ (.es), luego va el subdominio o subdominios (en este caso, `cica`) y por último el nombre de la máquina (`thales`).

En la figura de la página siguiente, podemos ver cómo sería la estructura jerárquica para la máquina `winnie.corp.hp.com`.

Los dominios genéricos de primer nivel son los .com, .edu, .org... más los correspondientes a los países (.es, .it, .uk, .pt,...). En Noviembre de 2000, ICANN (*Internet Corporation for Assigned Names and Numbers* www.icann.org) anunció la aparición de 7 nuevos dominios de primer nivel: .biz, .info, .name, .pro, .aero, .coop y .museum.

Además de estar jerarquizada, esta estructura se encuentra delegada. Veamos qué significa esto aplicándolo a nuestra dirección `thales.cica.es`.

¹En inglés, *Top Level Domain*



ICANN es una organización sin fines de lucro que opera a nivel internacional, responsable de asignar espacio de direcciones numéricas de protocolo de Internet (IP), identificadores de protocolo y de las funciones de gestión del sistema de nombres de dominio de primer nivel genéricos (gTLD) y de códigos de países (ccTLD).

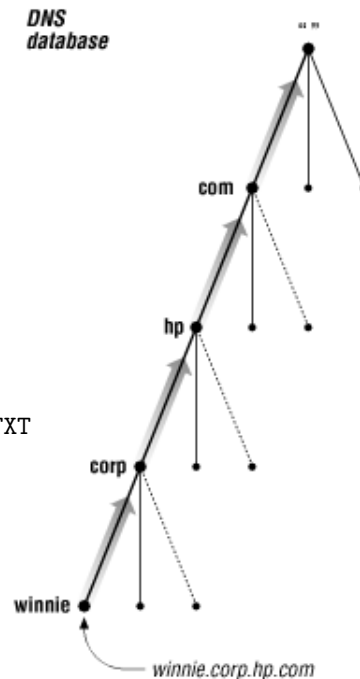
El dominio de primer nivel .es se encuentra delegado por ICANN a España, más concretamente al Organismo Red.es². A su vez, Red.es delega la administración del subdominio cica al Centro Informático Científico de Andalucía, que se convierte en responsable de todo lo que cuelgue de él, y así por ejemplo, puede darle nombre (y apellidos) a la máquina thales como thales.cica.es.

Este sistema hace que a pesar de la distribución y delegación de responsabilidades, todo funcione con la necesaria coordinación a nivel regional y mundial.

Para profundizar en el tema y conocer más sobre el dominio .es, podéis consultar en

<http://plugindoc.mozdev.org/linux.html>

Al principio, con pocas máquinas en Internet, bastaba para mantener este sistema con unos ficheros de nombre HOSTS.TXT o /etc/hosts, en los que se encontraban los nombres de las máquinas uno a uno. A medida que el sistema fue creciendo, se hacía necesario el soporte de un sistema más potente, que es el basado en *Servidores de Nombres*.



3.1. ¿Qué necesito del DNS?

Ésta es una de las principales cuestiones a las que deberemos responder a la hora de configurar y gestionar nuestros sistemas.

La gran mayoría de vosotros³, no necesitará montar y configurar un servidor de nombres, pero sí que los utilizaréis prácticamente en cada momento. Por ello, el comprender su funcionamiento y los recursos que ofrece es de gran ayuda.

Como vimos en la primera entrega, nuestra máquina Linux⁴ necesita saber cómo resolver las direcciones simbólicas a numéricas. Ello se hacía mediante los ficheros /etc/hosts, /etc/nsswitch.conf y /etc/resolv.conf, o los correspondientes interfaces gráficos.

Debemos diferenciar la utilización que hacemos de los servidores de nombres del hecho de montar un servidor de nombres propio. Es algo así como la diferencia entre utilizar un procesador de textos para nuestro trabajo diario y el desarrollar un procesador de textos nosotros mismos.

3.2. Recursos del Servidor de Nombres

Para ver qué nos ofrece un servidor de nombres utilizaremos la herramienta dig⁵. En su forma más simple, le preguntamos como argumento con un nombre de host para conocer la dirección que le corresponde.

```
root@guadalinux:~# dig thales.cica.es
; <<>> DiG 9.2.4 rc5 <<>> thales.cica.es
;; global options: printcmd
;; Got answer:
```

²Anteriormente era Rediris la encargada, a través del ES-NIC.

³Y la gran mayoría de los mortales

⁴Y las windows también.

⁵Domain Information Gropser, aunque puede significar "excavar". Esta herramienta sustituye a otra anterior que se llama nslookup.



```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49051
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4
;; QUESTION SECTION:
;thales.cica.es.                IN      A
;; ANSWER SECTION:
thales.cica.es.                172800 IN      A      150.214.5.10
;; AUTHORITY SECTION:
cica.es.                       172800 IN      NS     chico.rediris.es.
cica.es.                       172800 IN      NS     sun.rediris.es.
cica.es.                       172800 IN      NS     dns1.cica.es.
cica.es.                       172800 IN      NS     dns2.cica.es.
;; ADDITIONAL SECTION:
sun.rediris.es.                13337  IN      A      130.206.1.2
dns1.cica.es.                 172800 IN      A      150.214.5.83
dns2.cica.es.                 172800 IN      A      150.214.4.35
chico.rediris.es.             10872  IN      A      130.206.1.3
;; Query time: 80 msec
;; SERVER: 150.214.4.35#53(150.214.4.35)
;; WHEN: Sat Mar  5 18:54:39 2005
;; MSG SIZE  rcvd: 196
```

↪ Ésta es la salida del comando `dig`, bastante parlanchina, por cierto. La respuesta principal es la línea:

```
thales.cica.es. 172800 IN A 150.214.5.10
```

que nos dice que la máquina `thales.cica.es` tiene la dirección IP `150.214.5.10`. Además, nos dice que es una dirección de tipo INternet (IN) y es un recurso de tipo A (Address). El valor `172800` es un valor de tiempo de vida (ttl) del servidor de nombres.

Además, dentro de su cortesía nos regala información adicional, como las líneas

```
cica.es. 172800 IN NS sun.rediris.es.
```

que nos indican cuáles son los servidores de nombres “oficiales” para la zona `cica.es`, que son cuatro, con el tipo de recurso NS (Name Server), también nos ofrece sus direcciones

```
sun.rediris.es. 13337 IN A 130.206.1.2
```

y añade el tiempo que ha tardado la consulta, a quién y cuándo. La siguiente línea

```
;; SERVER: 150.214.4.35#53(150.214.4.35)
```

nos dice que la consulta ha sido realizada al servidor con dirección IP `150.214.4.35` por el puerto `53`, que es el que utiliza el servicio DNS. Como curiosidad, comentar que las consultas a los servidores DNS pueden realizarse tanto por TCP como por UDP.

El comando `dig` nos será de gran ayuda para consultar a los servidores de nombres. Una llamada típica al comando `dig` es de la forma:

```
dig @servidor_de_nombres recurso tipo_del_recurso
donde:
```

servidor_de_nombres es el servidor de nombres al que vamos a preguntar. En caso de que no lo especifiquemos, preguntará a los servidores de nombres que estén en el fichero `/etc/resolv.conf`

recurso es el nombre o dirección del que queremos consultar información

tipo_del_recurso es el tipo del recurso que buscamos. Si no especificamos ninguno, buscará el tipo A por defecto.



Si el puerto del servicio de nombres (53 o domain) está cortado por nuestro proveedor de acceso o red interna, podemos utilizar un interfaz web en <http://us.mirror.menandmice.com/cgibin/DoDig>.

Un servidor de nombres nos ofrece varios tipos de recursos. Veremos a continuación los más importantes.

A (*Address*) Nos da la correspondencia de dirección simbólica a dirección IP

CNAME (*canonical name*) Nos especifica un alias o apodo para una dirección simbólica

MX (*mail exchanger*) Indica la máquina o las máquinas que recibirán el correo

NS (*name server*) Indica los servidores de nombres oficiales para el dominio

PTR (*pointer*) Nos da la resolución inversa de una dirección IP a una dirección simbólica

SOA (*start of authority*) Autoridad sobre el Dominio de nombres.

Exprimamos un poco más el comando `dig`. Le preguntaremos al servidor de nombres 150.214.5.83⁶, que como vimos en el anterior comando, es un servidor de nombres oficial⁷ para el dominio cica.es.

```

root@guadalinux:~/curso-linux# dig @150.214.4.35 ANY cica.e s
; <<>> DiG 9.2.4rc5 <<>> @150.214.4.35 ANY cica.es
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51712
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 0, ADDITIONAL: 6
;; QUESTION SECTION:
;cica.es.                IN      ANY
;; ANSWER SECTION:
cica.es.                172800 IN      SOA     dns1.cica.es. hostmaster.
      cica.es . 2005022401 86400 7200 2592000 172800
cica.es.                300    IN      MX      15 smtp2.cica.es.
cica.es.                300    IN      MX      10 smtp.cica.es.
cica.es.                172800 IN      NS      sun.rediris.es.
cica.es.                172800 IN      NS      dns1.cica.es.
cica.es.                172800 IN      NS      dns2.cica.es.
cica.es.                172800 IN      NS      chico.rediris.es.
;; ADDITIONAL SECTION:
smtp.cica.es.          172800 IN      A       150.214.5.84
smtp2.cica.es.         172800 IN      A       150.214.5.100
sun.rediris.es.       12959  IN      A       130.206.1.2
dns1.cica.es.         172800 IN      A       150.214.5.83
dns2.cica.es.         172800 IN      A       150.214.4.35
chico.rediris.es.     10494  IN      A       130.206.1.3
;; Query time: 129 msec
;; SERVER: 150.214.4.35#53(150.214.4.35)
;; WHEN: Sat Mar 5 19:00:57 2005
;; MSG SIZE rcvd: 295

```

Los registros A y NS ya nos son conocidos. Aparece el registro SOA

```
cica.es. 172800 IN SOA dns1.cica.es. hostmaster.cica.es . 2005022401 86400 7200
2592000 172800
```

que indica quién es la autoridad para el dominio cica.es y parámetros para los servidores de nombres que veremos más adelante.

⁶Podríamos haber puesto `dns1.cica.es`

⁷El nombre en inglés es *authoritative*



También nos encontramos con registros MX, que a pesar de tener una gran importancia no son muy conocidos⁸.

```
cica.es. 300 IN MX 15 smtp2.cica.es.
```

```
cica.es. 300 IN MX 10 smtp.cica.es.
```

¿Por qué dijimos que eran muy importantes?, pues sencillamente porque dirigen los correos electrónicos. ¿Quién hoy día si le quitan el correo electrónico se quedaría igual?. Pues estos registros dicen que para todas las direcciones de correo electrónico del dominio `cica.es`⁹, como por ejemplo `jperez@cica.es`, deben dirigirse a los “intercambiadores de correo”¹⁰. Como es algo muy crítico, se suelen poner varios con una preferencia y en caso de fallo de alguno, los correos van al siguiente. En este caso irían preferentemente a `smtp.cica.es` y en caso de fallo de éste a `smtp2.cica.es`.

Preguntemos por un registro CNAME. El registro CNAME se suele utilizar como un alias o pseudónimo de otra u otras máquinas. ¿Qué utilidad puede tener esto? Por ejemplo, los servicios de Internet suelen prestarse en direcciones estandarizadas. Si queremos ver el Boletín Oficial del Estado y no sabemos con certeza la dirección, una de las primeras que probaremos si tenemos cierta experiencia con internet será `www.boe.es`. Nuestra máquina con el servidor web, no tiene porqué llamarse `www`¹¹ y además nos permite cambiar rápidamente a otra máquina sin demasiados problemas en nuestra red. Veamos lo que hace el CICA.

```
root@guadalinux:~# dig CNAME www.cica.es
; <<>> DiG 9.2.4rc5 <<>> CNAME www.cica.es
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31238
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 4
;; QUESTION SECTION:
;www.cica.es.                IN      CNAME
;; ANSWER SECTION:
www.cica.es.                3600    IN      CNAME   ataman.cica.es.
;; AUTHORITY SECTION:
cica.es.                    172800  IN      NS      chico.rediris.es.
cica.es.                    172800  IN      NS      sun.rediris.es.
cica.es.                    172800  IN      NS      dns1.cica.es.
cica.es.                    172800  IN      NS      dns2.cica.es.
;; ADDITIONAL SECTION:
sun.rediris.es.            12495   IN      A       130.206.1.2
dns1.cica.es.              172800  IN      A       150.214.5.83
dns2.cica.es.              172800  IN      A       150.214.4.35
chico.rediris.es.         10030   IN      A       130.206.1.3
;; Query time: 80 msec
;; SERVER: 150.214.4.35#53(150.214.4.35)
;; WHEN: Sat Mar 5 19:08:41 2005
;; MSG SIZE rcvd: 198
```

La línea importante en esta consulta es la que nos dice que `www.cica.es` es un apodo (CNAME) de la máquina `ataman.cica.es`. Si esa máquina se cae, una posible solución es cambiar el registro CNAME de `www.cica.es` a `atamon.cica.es`, que es una máquina que tenemos preparada para ello. El resto de usuarios (de todo el mundo) seguirán apuntando sus navegadores a `www.cica.es` sin enterarse del problema.

El recurso PTR es un poco más complicado. Veamos. Para que el mismo sistema funcione tanto para pedir conversiones de direcciones simbólicas a direcciones IP, como al revés, de direcciones IP a direcciones simbólicas se crea el recurso PTR y un dominio especial de nombre `in-addr.arpa`.

⁸Bueno, tú ya sé que eres un experto y sí los conoces ;-)

⁹Y de sus subdominios en caso de que no tengan especificados los suyos propios.

¹⁰Que eso es *Mail eXchanger*, de donde viene MX.

¹¹Sería un nombre bastante feo



Un comando sencillo para saber el nombre que le corresponde a una dirección IP es el comando `host`

```
[root@linux images]# host 150.214.5.10
10.5.214.150.in-addr.arpa domain name pointer thales.cica.es.
```

Vemos que nos devuelve que se corresponde con la dirección simbólica `thales.cica.es`, pero antes da una información un poco rara. Como en las direcciones simbólicas la jerarquía va de derecha a izquierda y en las direcciones IP de izquierda a derecha, se emplea un truco. Todas las direcciones IP se colocan bajo el dominio `in-addr.arpa` y se va poniendo cada uno de los bytes de la dirección IP de derecha a izquierda. Así `150.214.5.10` queda como `10.5.214.150.in-addr.arpa`. Veamos qué dice nuestro amigo `dig` sobre esto:

```
root@guadalinux:~/curso-linux# dig PTR 10.5.214.150.in-addr.arpa
; <<>> DiG 9.2.4 rc5 <<>> PTR 10.5.214.150.in-addr.arpa
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17607
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2
;; QUESTION SECTION:
;10.5.214.150.in-addr.arpa.      IN      PTR
;; ANSWER SECTION:
10.5.214.150.in-addr.arpa. 3600 IN      PTR      thales.cica.es.
;; AUTHORITY SECTION:
5.214.150.in-addr.arpa. 3600 IN      NS       dns2.cica.es.
5.214.150.in-addr.arpa. 3600 IN      NS       dns1.cica.es.
;; ADDITIONAL SECTION:
dns1.cica.es.          172800 IN      A        150.214.5.83
dns2.cica.es.          172800 IN      A        150.214.4.35
;; Query time: 155 msec
;; SERVER: 150.214.4.35#53(150.214.4.35)
;; WHEN: Sat Mar 5 19:15:56 2005
;; MSG SIZE rcvd: 141
```

Correcto, es un hacha este `dig`. Nos dice que estamos hablando de `thales.cica.es` y es un registro de tipo PTR (*Poin Te R*).

3.3. Servidores de Nombres

Seguro que el DNS os ha deparado muchas sorpresas. Pues aún hay más. El hecho de configurar un Servidor de Nombres es una auténtica odisea.

El servidor de nombres por excelencia es el demonio `named`, que es parte del paquete BIND, preparado y coordinado por el *Internet Software Consortium*.

Un servidor de nombres puede estar configurado de alguna de estas formas:

master Es el “dueño” del dominio¹², en el que se hacen las modificaciones para ese dominio, responde las consultas que se le hagan y se encarga de propagarlo al resto.

slave Son servidores de nombres del dominio y así se encargan de resolver las preguntas que se les hagan. Pero cada cierto tiempo le preguntan al “master” del que dependen para actualizar su información.

caching-only Solamente constituyen un caché de datos para optimizar las respuestas¹³. Por ejemplo, podemos montar uno de este tipo en nuestro equipo u organización para que todos los puestos clientes le pregunten a él. Sirve para optimizar las respuestas y el uso de la línea

¹²Zona es el término empleado.

¹³En algunos sistemas (por ejemplo, fedora) se incluye una caché local mediante el demonio `nscd` (*name server cache daemon*).

de comunicaciones, pero además simplifica la política de seguridad. Para las peticiones de resolución DNS, los clientes no pueden atravesar el cortafuegos y sí esta única máquina.

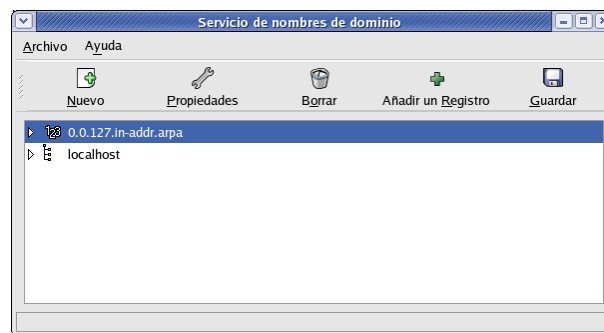
forwarding Redirige las peticiones a otros servidores de nombres. Es poca la diferencia con el de caché.

En el terreno árido, BIND guarda su configuración en los siguientes sitios:

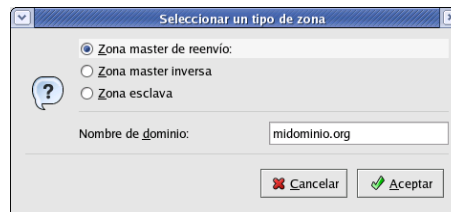
`/etc/named.conf` Fichero de configuración del demonio named.

`/var/named/` Directorio en el que almacena el resto de ficheros.

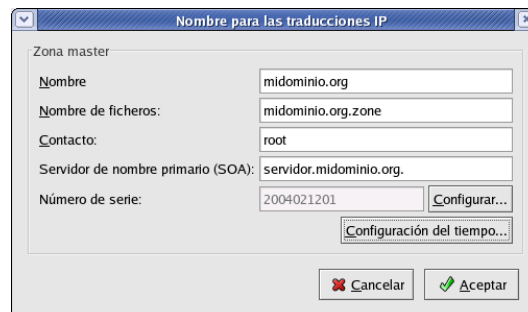
Afortunadamente, las herramientas gráficas de configuración nos son de mucha ayuda. Por ejemplo, **system-config-bind** es la herramienta gráfica de los sistemas tipo redhat y fedora.



Vemos que ya nos presenta una zona master para localhost y una zona master inversa para los registros PTR. Si deseamos crear una nueva zona, nos presenta tres opciones: una zona master, una zona master inversa o una zona esclava. Veremos el caso más completo que es el de una zona master. Crearemos el especial dominio *midominio.org*.

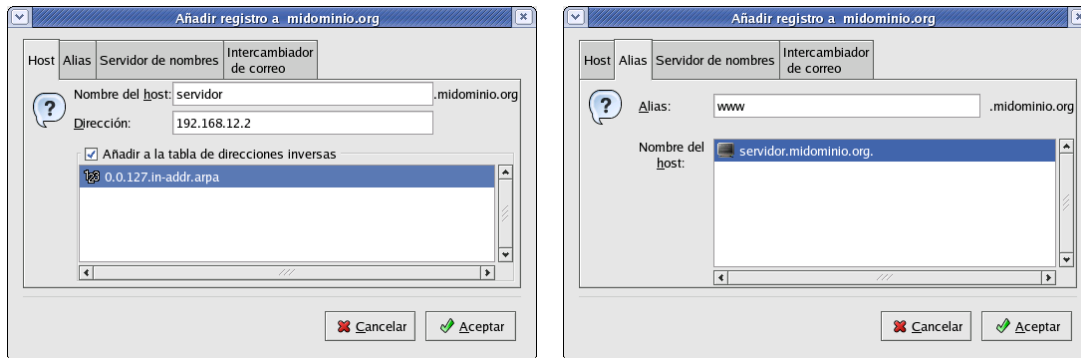


La herramienta ya por sí sola generará los ficheros necesarios y solamente tendremos que decirle el servidor dueño de la zona (`servidor.midominio.org`¹⁴). El número de serie es un número que se utiliza cuando un master traspassa información a un servidor de nombres esclavo. El esclavo si ve que el número de serie ha cambiado, pedirá la información. Si no ha cambiado el número de serie, se supone que la información tampoco ha cambiado.



¹⁴Fijaos que termina en un punto. Es una forma de indicarle dónde está la raíz principal y evitar fallos como `midominio.org.midominio.org`

Una vez que hemos creado la zona para nuestro dominio, le añadiremos registros, que pueden ser de los tipos vistos anteriormente (A, CNAME, NS o mX).



En la primera figura estamos creando un registro de tipo A. La dirección simbólica `servidor.midominio.org` la asignamos a la dirección IP `192.168.12.2`. Para la resolución inversa (PTR) tendremos que crear el dominio inverso `12.168.192.in-addr.arpa`.

En el segundo gráfico, añadimos un registro CNAME y creamos un alias entre las direcciones simbólicas `www.midominio.org` y `servidor.midominio.org`.

Pasemos a ver qué ha hecho la configuración gráfica sobre los ficheros. Empezamos con `/etc/named.conf`

```
[root@linux images]# more /etc/named.conf
// generated by named-bootconf.pl
//
// a caching only nameserver config
//
options {
/*
* If there is a firewall between you and nameservers you want
* to talk to, you might need to uncomment the query-source
* directive below. Previous versions of BIND always asked
* questions using port 53, but BIND 8.1 uses an unprivileged
* port by default.
*/
// query-source address * port 53;
directory "/var/named";
};
controls {
inet 127.0.0.1 allow { localhost; } keys { rndckey; };
};
zone "." {
type hint;
file "named.ca";
};
zone "localhost" {
allow-update { none; };
type master;
file "localhost.zone";
};
zone "0.0.127.in-addr.arpa" {
allow-update { none; };
type master;
file "named.local";
};
zone "midominio.org" {
```

```
type master;
file "midominio.org.zone";
};
include "/etc/rndc.key";
```

Especifica que los ficheros de zonas y configuración adicional estará en el directorio `/var/named`. La configuración que viene por defecto crea un servidor de nombres que funciona como caché. De ahí provienen las zonas `."`, `localhost` y `0.0.127.in-addr.arpa`.

Para la zona que hemos creado, `midominio.org`, especifica que es de tipo master y que el resto de la configuración se encuentra en el fichero `midominio.org.zone`, que se encontrará en el directorio... bien, has acertado: `/var/named`. Veámoslo

```
[root@linux images]# more /var/named/midominio.org.zone
$TTL 86400
servidor.midominio.org. IN SOA localhost root (
2004021207 ; serial
28800 ; refresh
14400 ; retry
3600000 ; expire
86400 ; ttl
)
servidor IN A 192.168.12.2
.midominio.org IN MX 1 servidor.midominio.org.
```

La autoridad para el dominio (SOA) es `servidor.midominio.org`, vemos que el serial es 2004021207. Normalmente, por convención se pone en formato año, mes, día y modificación dentro del día. El resto de valores son el tiempo en segundos, en que se refresca la información a los esclavos, que se reintenta en caso de no poder conectar, tiempo de expiración y máximo tiempo que lo pueden tener las cachés.

Hemos creado un registro tipo A que une las direcciones `servidor.midominio.org` y `192.168.12.2` y también un registro MX que indica que el correo dirigido al dominio `midominio.org`, será recogido por el servidor `servidor.midominio.org`.

En el fichero `named.local` podemos observar una típica zona de registros inversos tipo PTR.

```
[root@linux images]# more /var/named/named.local
$TTL 86400
@ IN SOA localhost. root.localhost. (
1997022703 ; serial
28800 ; refresh
14400 ; retry
3600000 ; expire
86400 ; ttl
)
@ IN NS localhost
1 IN PTR localhost.
2 IN PTR servidor.midominio.org.
```



Capítulo 4

Servicio de Directorio LDAP

El problema de nombrar y direccionar entidades no admite una única solución. En el fondo, el problema es éste: la entidad 1 debe conocer el nombre de la entidad 2 para intercambiar datos con ella, ¿pero cómo puede obtener ese nombre a menos que ya lo conozca? (WILLIAM STALLINGS, *Data and Computer Communications*)

En los albores de Internet, la ISO (*International Standards Organization*) comprendió la necesidad de tener un sistema de directorio en el que poder tener de forma ordenada y organizada información sobre una organización, ya fuera esta información sobre personas, máquinas o servicios. Así surgió X.500, un sistema de Directorio potente y muy completo. Pero que adolecía de los problemas de la mayoría de los protocolos ISO: su complejidad y dificultad de implantación.

Hasta hace poco tiempo, los sistemas de Internet han carecido de sistemas de directorio, pero en el momento actual, en organizaciones de tamaño mediano y grande se necesita de las facilidades que incorporan estos sistemas. Incluso en redes pequeñas también aportan muchas ventajas. Por ejemplo, los nombres de usuario y palabras de paso se pueden controlar de forma centralizada en un directorio y ser utilizadas por todos los sistemas. ¿No os gustaría como usuarios tener una sola password para todos los sistemas? ¿Y como administradores de una red por pequeña que sea, no os gustaría controlar las passwords desde un solo sitio para todos los usuarios?. Pues añadámosle a la coctelera las direcciones y nombres de las máquinas, los certificados digitales de los usuarios, teléfonos y direcciones electrónicas, etc. Resultado: ¿cómo hemos podido vivir hasta ahora sin los directorios?. Bueno, es un poco exagerado, pero los directorios han venido para quedarse e integrarse en nuestras aplicaciones.

Un sistema de directorio puede ser utilizado para almacenar un amplio rango de datos: dirección de correo electrónico, números de teléfono, ubicación, claves públicas de seguridad, listas de contactos, y prácticamente lo que se nos ocurra o necesitemos. El directorio LDAP se convierte en un punto de integración de información del organismo o empresa.

El Protocolo de Acceso Ligero a Directorio, más conocido como LDAP¹, está basado en el estándar X.500, pero es significativamente más simple y adaptado a TCP/IP.

Hasta aquí, alguien podría pensar que para esto ya tenemos las bases de datos. En cierto modo un directorio es una base de datos, pero con unas características especiales:

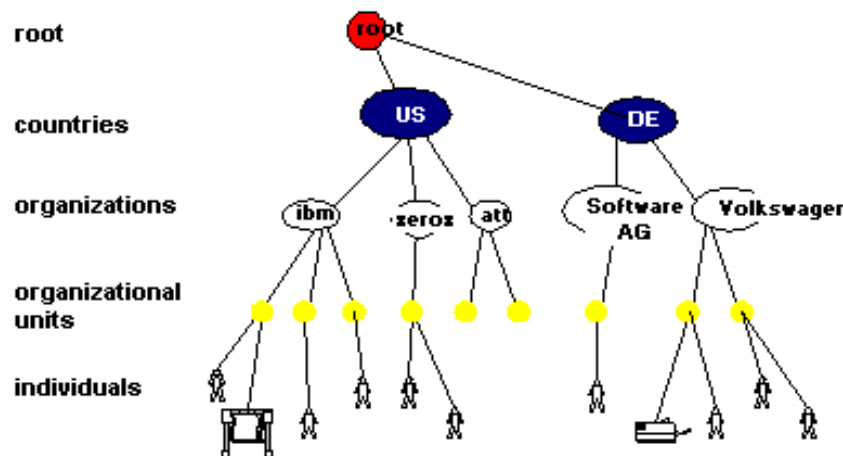
- Datos relativamente estáticos. Los datos son modificados con poca frecuencia. ¿Cada cuánto tiempo cambiamos de número de teléfono o de correo electrónico?
- Las operaciones de lectura (recuperación) deben ser muy rápidas. El directorio está optimizado para lecturas frecuentes y concurrentes.
- Distribuido. Los datos se ubican físicamente en varios sistemas de la red aportando redundancia, altas prestaciones y escalabilidad

¹Siglas en inglés de *Lightweight Directory Access Protocol*

- Jerárquico. Asegura una gestión descentralizada y delegada de la información, de forma similar a como vimos con el sistema DNS.
- Orientado a Objetos. El directorio representa objetos que pertenecen a clases. Las clases son una colección de atributos, no necesariamente normalizados como en una base de datos tradicional.
- Esquema estándar, disponible para todas las aplicaciones que usan el directorio.
- Atributos multivaluados. Los atributos del directorio pueden tener un único o múltiples valores.
- Replicación Multi-master. A diferencia de las bases de datos tradicionales, los directorios se distribuyen por la red. Si un sistema no está disponible, el cliente accede a otra réplica de la información.

4.1. Estructura del Directorio

En un directorio LDAP, las entradas se disponen en una estructura jerárquica en forma de árbol. Por ejemplo, esta estructura puede reflejar componentes geográficos u organizacionales. Los países pueden estar en la parte superior del árbol bajo la raíz. Debajo de cada país los estados o comunidades autónomas. Bajo cada organización puede haber más unidades organizacionales y bajo ellas, personas, impresoras, documentos o lo que podamos necesitar.



La forma de almacenar información en un directorio es en base a entradas (*entries*). Una *entrada* es una colección de atributos que tiene un identificador único llamado *Distinguished Name* (DN). Mediante el DN podemos referirnos a la entrada de forma no ambigua.

Una entrada pertenece a una clase de objeto (*objectclass*), que define el conjunto de atributos que puede tener. Cada uno de los atributos de una entrada es de un tipo² y puede poseer cero, uno o más valores. Los tipos se denominan con nombres que nos dan una idea de su contenido. Por ejemplo, *cn* para nombre común³, *o* para Organización, *c* para país⁴ o *mail* para dirección de correo electrónico.

Algunos atributos son obligatorios para la clase, mientras que otros son opcionales. Una definición de la clase de objeto (*objectclass*) determina qué atributos son obligatorios y cuáles no para cada una de las entradas que pertenecen a esa clase. Las definiciones de clases de objetos se

²O clase

³*Common Name* en inglés

⁴*Country*

encuentran en varios ficheros de esquema, que se encuentran en el directorio `/etc/openldap/schema/`⁵.

El valor de un atributo depende del tipo que posea. Un atributo de tipo `cn` puede tener el valor “Manuel Pérez Pérez”, y un atributo de tipo `mail` puede tener el valor “manuel.perez@midominio.org”.

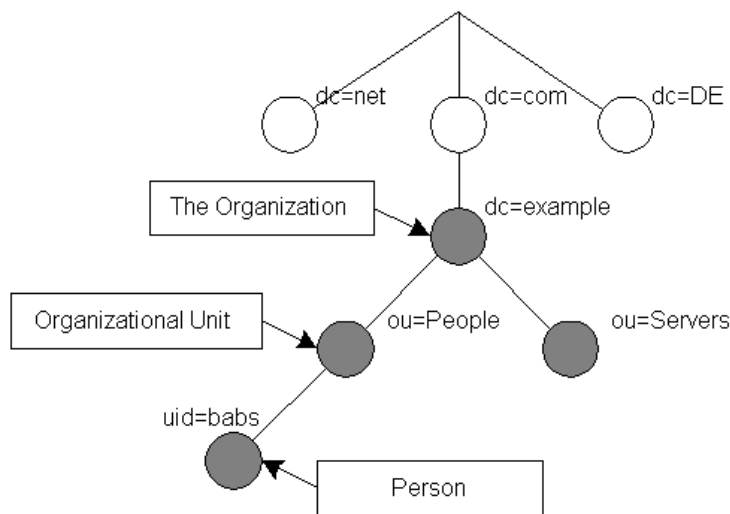
A continuación presentamos el contenido de una entrada:

```
dn: o=Sociedad Matemática Thales, c=ES
o: Sociedad Matemática Thales
objectclass: organization
```

El `dn` es el identificador de la entrada, que puede ser visto como su clave o su nombre completo, que debe ser único en el directorio. En este caso nos indica que es una organización de nombre “Sociedad Matemática Thales”, que pertenece al país España.

La entrada `objectclass` nos dice que es de un tipo denominado `organization`, que permitirá, según su definición, que la entrada pueda tener unos determinados atributos. Esta definición de los tipos se realiza en el esquema (*schema*).

La tendencia actual es a nombrar la estructura basándose en los nombres de dominio de Internet. En el árbol siguiente podemos ver un ejemplo.



El DN de una entrada se construye cogiendo el nombre de la entrada en el árbol⁶ y concatenando los nombres de cada entrada superior hasta llegar a la raíz del árbol. En la figura anterior, la persona con `uid=babs`, que es su RDN, tendrá un DN de `uid=babs, ou=People, dc=example, dc=com`. Significa que el identificador de usuario `babs`, es de una persona porque es su tipo, perteneciente a la unidad organizativa⁷ *Personas*⁸, de la Organización *example*, dentro del dominio de primer nivel *.com*.

El servicio de directorio LDAP se basa en un modelo cliente-servidor. Uno o más servidores LDAP contienen los datos que forman el Árbol de Información del Directorio (DIT⁹). Los clientes se conectan a un servidor y le realizan una consulta. El servidor responde con la respuesta o un puntero a dónde encontrarla si él no la tiene.

⁵En un sistema RedHat o Fedora.

⁶llamado el RDN, de *Relative Distinguished Name*

⁷*Organizational Unit*

⁸La otra rama indica los servidores

⁹*directory information tree*

4.2. Servidor OpenLDAP

El servidor OpenLDAP es una implementación en Software Libre del protocolo LDAP, proporcionando tanto los servidores, como clientes básicos para trabajar con ellos y librerías para enlazar con otras aplicaciones.

En un sistema Fedora, nos encontramos con los siguientes paquetes:

openldap Contiene las librerías que necesitan tanto el servidor OpenLDAP como las aplicaciones clientes.

openldap-clients Contiene clientes básicos de línea de comandos para consultar y modificar la información que almacena el servidor LDAP.

openldap-servers Contiene los servidores OpenLDAP y utilidades para su configuración.

En el paquete de servidores realmente vienen dos servidores, el propiamente dicho demonio de LDAP (`/usr/sbin/slapd`) y el demonio de replicación (`/usr/sbin/slurpd`). El demonio de sincronización `slurpd` se utiliza para sincronizar cambios de un servidor LDAP a otros que hayamos definido. Para nuestros propósitos no necesitaremos el demonio de replicación.

En el caso de tener una distribución basada en debian como el caso de Guadalinux2004 debemos realizar la instalación de los paquetes

slapd Contiene los servidores ldap (tanto slapd como slurpd)

ldap-utils Contiene herramientas de cliente básicas.

A la hora de realizar la instalación nos aparece una serie de pantallas de configuración que podemos rellenar o cancelar y definirlo posteriormente en el fichero de configuración `/etc/ldap/slapd.conf`



Por defecto el servidor ldap se ejecuta como root, si deseamos cambiar el usuario debemos modificar los parámetros `SLAPD_USER` y `SLAPD_GROUP` del fichero de configuración `/etc/default/slapd` para que se ejecute el demonio con la opción `-u usuario_definido`

4.2.1. Configuración del Servidor OpenLDAP

La configuración se encuentra en el fichero `/etc/openldap/slapd.conf` o en `/etc/ldap/slapd.conf` dependiendo de la distribución. Dentro de la configuración existen múltiples opciones, muchas de ellas dependerán de los tipos de clientes y otras de la finalidad que tenga el ldap. En principio recomendamos evitar la versión 2 del protocolo por facilitar la configuración, así como el *backend* `ldbm`. Trabajaremos con los ficheros de fedora, que son similares a los de Guadalinux salvo en la ubicación.

Veamos su contenido más interesante.



```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/inetorgperson.schema
```

Incluye los esquemas que podemos utilizar para los tipos de entradas.

```
database ldbm
```

Tipo de base de datos que utiliza para almacenar su contenido.

```
suffix "dc=midominio,dc=org"
```

Definimos la estructura principal de nuestro directorio. En este caso para una organización cuyo dominio en Internet es midominio.org.

```
rootdn "cn=Manager,dc=midominio,dc=org"
```

Definimos quién será el administrador del directorio

```
rootpw {SSHA}X1XTJTGJvKseb+AXnX/XY8iHqxq03EPV
```

La contraseña del administrador del directorio. Podemos ponerla aquí en texto claro (`rootpw secreto`) pero es preferible ponerla cifrada¹⁰.

```
directory /var/lib/ldap
```

Directorio donde se va a almacenar la información del directorio.

Ya estamos listos para arrancar el servidor de directorio.

```
Debian #/etc/init.d/sldap restart
```

```
Fedora #service ldap restart
```

Comprobamos que ya funciona, aunque todavía no tengamos datos en el directorio.

```
#ldapsearch -x -b " -s base '(objectclass=*)' namingContexts
# extended LDIF
# LDAPv3
# base <> with scope base
# filter: (objectclass=*)
# requesting: namingContexts
dn:
namingContexts: dc=midominio,dc=org
# search result
search: 2
result: 0 Success
# numResponses: 2
# numEntries: 1
```

La consulta no es normal, ya que estamos preguntando a un directorio vacío. La opción

`-x` es utilizar autenticación normal,

`-b` indica la base de búsqueda,

`-s` indica el ámbito de la búsqueda y nos contentamos con cualquier cosa que nos devuelva (`objectclass=*`).

Al menos vemos que el servidor está funcionando y nos devuelve algo.

Ya tenemos nuestro servidor ldap funcionando con una estructura organizativa `dc=midominio, dc=org`. Pero el directorio está vacío. Tenemos que alimentarlo de entradas.

El formato **LDIF** es el estándar para representar entradas del directorio en formato texto. Una entrada del directorio consiste en dos partes. El DN o nombre distinguido, que debe figurar en la

¹⁰Para obtener la clave cifrada, ejecutamos:

```
# slappasswd
New password:
Re-enter new password:
{SSHA}X1XTJTGJvKseb+AXnX/XY8iHqxq03EPV
```



primera línea de la entrada y que se compone de la cadena dn: seguida del DN de la entrada. La segunda parte son los atributos de la entrada. Cada atributo se compone de un nombre de atributo, seguido del carácter dos puntos, :, y el valor del atributo. Si hay atributos multievaluados, deben ponerse seguidos.

No hay ningún orden preestablecido para la colocación de los atributos, pero es conveniente listar primero el atributo `objectclass`, para mejorar la legibilidad de la entrada.

El DN debe estar en ASCII (el código ASCII puro es de 128 caracteres). Para conseguirlo seguimos estos criterios: cambiamos una vocal acentuada por la misma vocal no acentuada sustituimos la letra ñ por n y transformamos la letra ü en u.

```
dn: uid=Fernando G.,ou=AIT,o=midominio,c=org
objectclass:person
objectclass:organizationalPerson
objectclass:inetOrgPerson
objectclass:posixAccount
objectclass:top
uid: Fernando G.
sn: Gordillo
cn: Fernando Gordillo
description: usuario de ejemplo
loginshell: /bin/sh
uidnumber:502
gidnumber:1000
mail: fernandogs@midominio.org
telephonenumber: 55555555
homedirectory: /home/fernando
```

Las líneas excesivamente largas pueden partirse con un retorno de carro y añadiendo un espacio al principio de la siguiente línea.

Si un atributo contiene valores no ASCII, como por ejemplo una imagen JPEG, se codifica en formato Base64

El formato LDIF también puede ser utilizado para realizar actualizaciones y/o borrar entradas del directorio. El formato en este caso contiene en la primera línea el DN de la entrada sobre la que se aplica el cambio. La segunda línea indica el cambio a realizar y las siguientes líneas contienen los pares atributo-valor que componen el cambio.

Para añadir una entrada

```
Dn: nombre distinguido
Changetype: add
Tipo_atributo: valor
```

Para borrar una entrada basta indicar el cambio delete

```
Dn: nombre distinguido Changetype: delete
```

Para modificar una entrada

```
Dn: nombre distinguido Changetype: modify
TipoCambio: atributo atributo: valor
```

Varias operaciones se pueden combinar en un único fichero si las separamos por un guión. Veamos un fichero en este formato que nos servirá para iniciar nuestro servidor ldap en un primer momento:

```
dn: dc=midominio,dc=org
objectclass: dcObject
objectclass: organization
```

```
o: Organismo Ejemplo
dc: midominio

dn: cn=Manager,dc=midominio,dc=org
objectclass: organizationalRole
cn: Manager
```

Es importante la línea en blanco porque separa dos entradas del directorio. La primera es la de la organización en sí, con DN `dc=midominio,dc=org`. Las dos entradas `objectclass` nos indican a qué tipos pertenece la entrada: pertenece al tipo `organization` y al tipo `dcObject`. En este caso incluye un atributo que es `o` (organización), para introducir el nombre de ésta. Si queremos ver qué atributos permiten dichas clases, deberemos consultarlos en el directorio de esquemas `/etc/openldap/schema`.

La otra entrada es para el administrador del directorio, el *Manager*.

Insertemos en el directorio el contenido de este fichero. Utilizaremos el comando `ldapadd`.

```
#ldapadd -x -D "cn=Manager,dc=midominio,dc=org" -W -f entrada1.ldif
Enter LDAP Password:
adding new entry "dc=midominio,dc=org"
adding new entry "cn=Manager,dc=midominio,dc=org"
```

Las opciones con las que lo invocamos son:

`-x` para utilizar autenticación simple

`-D` más el DN del usuario con el que nos conectamos, que es el `Manager`

`-W` para que nos pregunte la clave, en vez de ponerla en la línea de comandos y que alguien pueda verla

`-f` para indicarle el fichero, que será `entrada1.ldif`

Si todo ha funcionado correctamente, como es el caso del comando anterior, ya tenemos dos entradas en nuestro directorio. Podemos utilizar cualquier cliente LDAP para consultarlas.

Un cliente básico de línea de comandos es `ldapsearch`. Lo utilizaremos para hacer esta consulta inicial. Con la opción `-b` le indicamos la base de búsqueda. Si no se la indicamos, no sabe por dónde hacer la búsqueda y no nos devolvería resultados.

Además no nos ponemos muy exigentes y le decimos que nos devuelva cualquier cosa que encuentre.

```
#ldapsearch -x -b 'dc=midominio,dc=org' '(objectclass=*)'
# extended LDIF
# LDAPv3
# base <dc=midominio,dc=org> with scope sub
# filter: (objectclass=*)
# requesting: ALL
# midominio.org
dn: dc=midominio,dc=org
objectClass: dcObject
objectClass: organization
o: Organismo Ejemplo
dc: midominio
# Manager, midominio.org
dn: cn=Manager,dc=midominio,dc=org
objectClass: organizationalRole
cn: Manager
```

Ha encontrado las dos entradas que existen en el directorio.

Hay una forma de especificar por defecto la base de búsqueda para los clientes, que es en el fichero `/etc/openldap/ldap.conf` o `/etc/ldap/ldap.conf`

```
HOST 127.0.0.1
BASE dc=midominio,dc=org
o
BASE dc=midominio,dc=org
URI ldap://ldap.example.com:389
```

Le estamos diciendo a qué servidor se tiene que conectar el cliente si no le especificamos otro y cuál será la base de búsqueda por defecto.

Seamos un poco más exigentes y alimentemos un fichero LDIF con empleados o funcionarios de nuestra organización.

```
root@guada04:~# more entrada3.ldif
dn: cn=Juan Lopez Perez, dc=midominio,dc=org
objectClass: person
objectClass: inetOrgPerson
mail: juan.lopez@midominio.org
telephoneNumber: +34-954-55-55-55
sn: Lopez
cn: Juan Lopez Perez
cn: Juan Lopez

dn: cn=Laura Jimenez Lora, dc=midominio,dc=org
objectClass: person
objectClass: inetOrgPerson
mail: laura.jimenez@midominio.org
telephoneNumber: +34-959-59-59-59
sn: Jimenez
cn: Laura Jimenez Lora
cn: Laura
```

Si realizamos una nueva búsqueda, ahora será mucho más rica:

```
root@guada04:~# ldapsearch -x -
b 'dc=midominio,dc=org' '(objectclass=*)'
# extended LDIF
# LDAPv3
# base <> with scope sub
# filter: (objectclass=*)
# requesting: ALL
# midominio.org
dn: dc=midominio,dc=org
dc: midominio
objectClass: dcObject
objectClass: organization
o: Organismo Ejemplo
# Manager, midominio.org
dn: cn=Manager,dc=midominio,dc=org
objectClass: organizationalRole
cn: Manager
# Juan Perez Perez, midominio.org
dn: cn=Juan Perez Perez,dc=midominio,dc=org
telephoneNumber: +34-950-50-50-50
```

```
mail: juan.perez@midominio.org
objectClass: person
objectClass: inetOrgPerson
sn: Lopez
cn: Juan Perez Perez
cn: Juan Perez
```

Las herramientas administrativas con que nos encontramos son:

slapadd Añade entradas de un fichero LDIF a un directorio LDAP directory.

slapcat Recupera entradas de un directorio LDAP y las almacena en un formato LDIF.

slapindex Reindexa el contenido del directorio.

slappasswd Genera una palabra de paso cifrada.

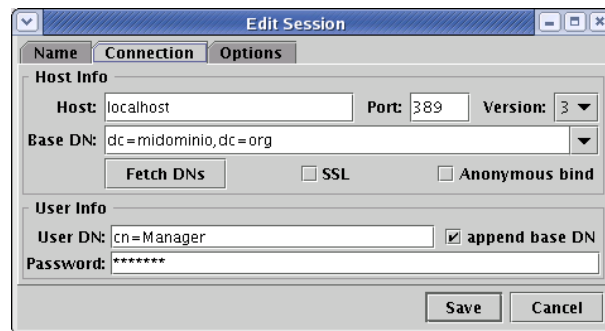
Sin embargo, recomendamos el cliente **ldapbrowser** que veremos a continuación para la modificación, importación y exportación de los datos del directorio.

4.3. Clientes LDAP

Tenemos una gran cantidad de clientes que pueden conectarse a nuestro servidor de directorio y además, cada uno puede utilizarlo para un propósito distinto. Un cliente de correo electrónico puede utilizarlo para buscar direcciones o el certificado del destinatario para enviarle el correo cifrado. Un sistema de proxy-caché puede autenticar al usuario basándose en el directorio para permitirle navegar por Internet. Además, estos clientes pueden ser clientes windows, Linux o de cualquier otro sistema. Solamente tienen que cumplir las reglas del protocolo LDAP.

Un cliente muy versátil que nos permitirá consultar y modificar el directorio es LDAP Browser/Editor. Es una herramienta construida en Java que podemos obtener de <http://www.iit.edu/~gawojar/ldap>.

Nos descargamos el fichero `Browser282b2.tar.gz`¹¹, lo descomprimos y ejecutamos en el directorio `ldapbrowser` el fichero `lbe.sh`. Es necesario que tengamos una máquina virtual de Java instalada en nuestro sistema. Si no es así podemos descargarnos de <http://java.sun.com> el *Java Runtime Environment*, por ejemplo el paquete `j2re-1_4_2_03-linux-i586.rpm`.

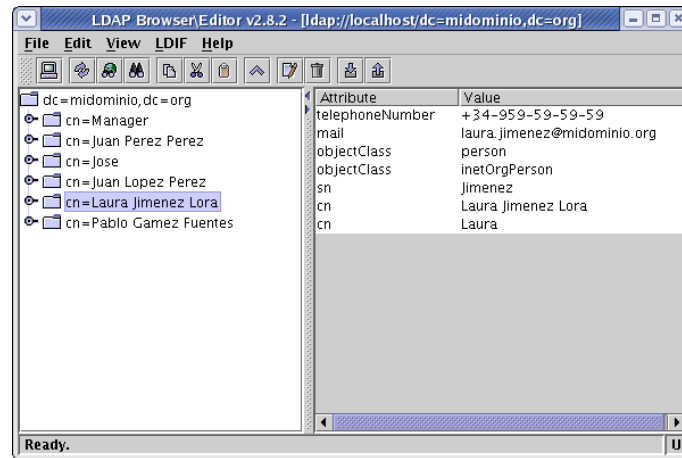


Para definir nuestra nueva conexión, debemos darle el host (localhost en este caso), el puerto (el 389 es el estándar del protocolo), la versión de protocolo (la 3 es la actual), la base de búsqueda (`dc=midominio,dc=org`) y si hacemos un acceso anónimo marcamos el cuadro [**Anonymous bind**].

Si queremos entrar como un usuario del directorio, en este caso utilizaremos el Manager, y podremos modificar entradas y atributos. Marcamos la opción de que le añada la base de búsqueda. Quedaría

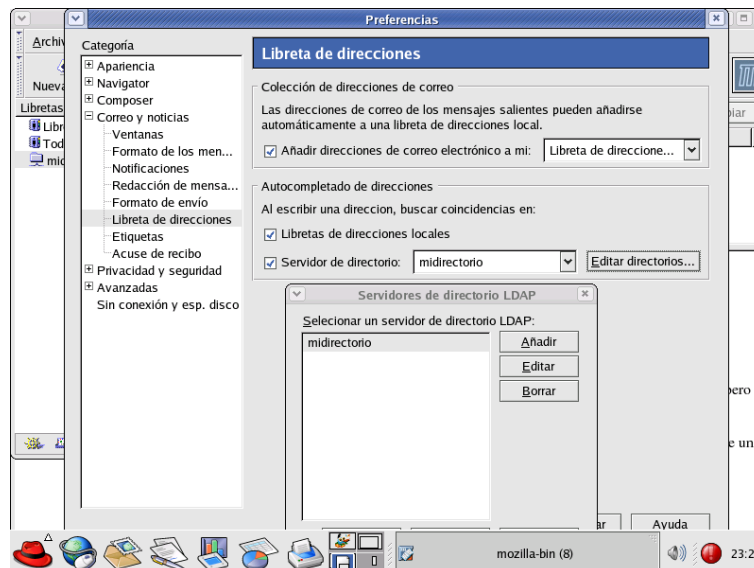
`cn=Manager,dc=midominio,dc=org`, pero hemos escrito menos ;-).

¹¹Para cualquier sistema operativo, ya que está hecho en Java

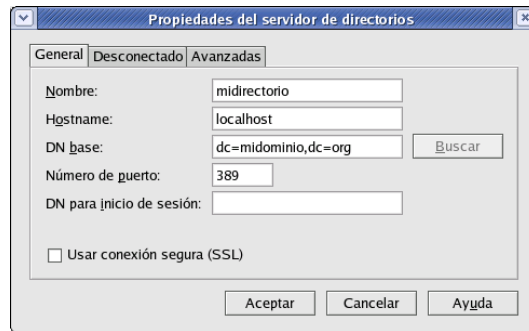


¿Queda bonito verdad? Pues además es fácil de manejar y es gratis. Se cumple lo de bueno, bonito y barato. Es una estupenda herramienta de consulta y modificación de directorios LDAP.

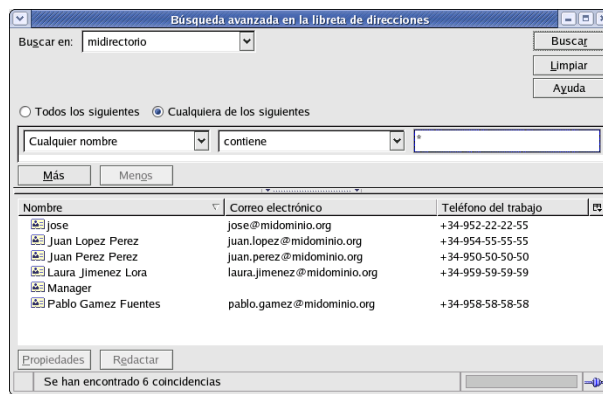
Veamos además un cliente de correo electrónico. Utilizaremos el correo electrónico de Mozilla. Para decirle a la libreta de direcciones que utilice el directorio, seleccionamos **Editar**→**Preferencias**→**Correo y Noticias**→**Libreta de Direcciones**. En la sección de Autocompletado de direcciones, seleccionamos **Servidor de Directorio** y editamos para crear uno nuevo.



En la ventana de **Propiedades del servidor de directorios**, especificamos un nombre, el nombre del servidor al que nos tenemos que conectar (hostname), la base de búsqueda (`dc=midominio,dc=org`) y el puerto (389 por defecto). Tendremos opción de entrar como usuario autenticado o de usar conexión segura mediante SSL.



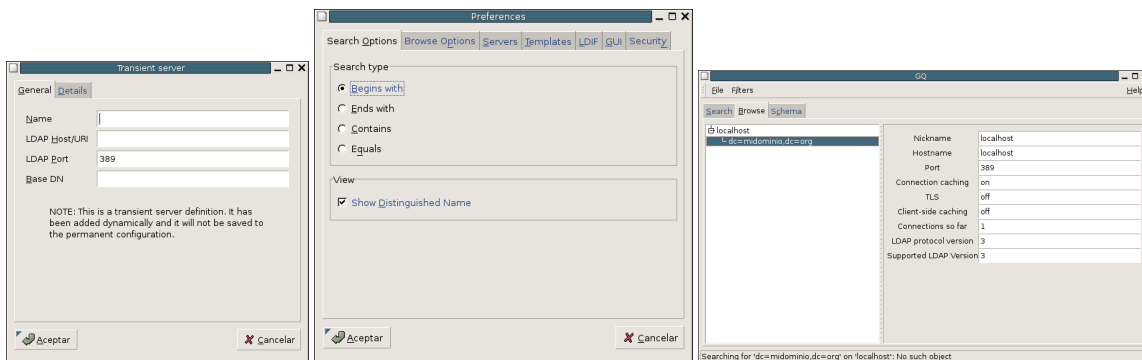
Una vez configurado, podemos utilizar el directorio para buscar personas, sus direcciones, teléfonos o los atributos que hayamos alimentado en el directorio.



Los servicios de directorio tenderán a concentrar cada vez más funciones y a convertirse en el alma de la mayoría de las organizaciones.

Otro cliente que puede utilizarse de forma rápida en una máquina Linux es **GQ**. Este paquete se instala de forma sencilla con `apt-get install gq` o `rpm -Uhv gq-version.sistema.rpm` y nos permite conectarnos a un directorio ldap y gestionarlo de forma gráfica.

Desde este programa una vez configuradas la preferencias y dentro de ellas los parámetros de conexión al servidor, podemos realizar búsquedas, navegar en el directorio y ver el schema.



El formato básico de los filtros de búsqueda de LDAP es el siguiente:

Atributo operador valor

El atributo se refiere al atributo sobre el que vamos a realizar la operación de comparación. El operador puede ser uno de los siguientes:

- = Devuelve las entradas cuyo atributo tiene el valor especificado.
- >= Devuelve las entradas cuyo atributo sea mayor o igual que el valor especificado
- <= Devuelve las entradas cuyo atributo sea menor o igual que el valor especificado
- =* Devuelve las entradas que tienen valor asignado en el atributo especificado
- ~= Devuelve las entradas cuyo atributo tenga un valor similar al especificado

Operadores de filtros

El carácter * tiene el significado de cualquier valor y puede ser empleado con el operador =. Pero además, los operadores de búsquedas pueden combinarse utilizando los operadores booleanos, dando lugar a expresiones de búsqueda más complejas. La sintaxis para combinar filtros de búsquedas es la siguiente:

```
(operador (filtro1) (filtro2) (filtro3) ? )  
(operador (filtro))
```

Los operadores son:

- & : AND lógico de los filtros.
- | : OR lógico de los filtros.
- ! : NOT lógico del filtro.

Algunos ejemplos de filtros:

- (& (uid=*2202) (cn=Fernando*)) Busca entradas cuyo campo uid termine en 2202 y cuyo campo cn empiece por Fernando.
- (| (cn=Fernando) (cn=Irene)) Busca entradas cuyo campo cn sea Fernando
- (! (cn=Emilio)) Busca todas las entradas cuyo campo cn no sea Emilio.

Para realizar búsquedas sobre atributos cuyos valores contienen alguno de los caracteres reservados para la construcción de los filtros, deben utilizarse secuencias de escape.

4.4. Caso práctico: Autenticación mediante directorio LDAP

Acabamos de ver cómo podemos tener una base de datos para almacenar información acerca de los usuarios de nuestro sistema. Ya comentamos en el comienzo de este capítulo que es posible utilizar el servicio de directorio LDAP para almacenar de forma centralizada la clave de acceso de los usuarios.

Usando una base de datos centralizada de información de usuarios se simplifica enormemente la gestión de claves. Supongamos el siguiente escenario de trabajo, que no tiene nada de extraordinario, en el que gestionamos un servidor de correo y un servidor web. Cada usuario tendrá una cuenta de correo y un espacio, respectivamente, en los servidores anteriores. Sin otra infraestructura adicional, cada vez que un usuario quiera cambiar la contraseña de un sistema deberá cambiarla en el otro. Si queremos que se utilice la misma contraseña en los dos sistemas tendremos que crear un mecanismo de replicación, con el consiguiente trabajo adicional.

Mediante un directorio LDAP podemos tener un repositorio común al que accederán los sistemas de autenticación de los servidores que sean configurados de esta forma. En esta sección, basándonos en lo visto anteriormente, daremos contenido a un directorio LDAP y posteriormente configuraremos el sistema para que lo utilice como base de datos de autenticación.

4.4.1. Crear un directorio para autenticación

En primer lugar vamos a modificar un poco el esquema anterior. Los siguientes ficheros en formato `ldif` nos definirán el esqueleto de los datos que utilizaremos. Primero crearemos la raíz de nuestro árbol LDAP:

```
dn: dc=midominio,dc=org
objectclass: dcObject
objectclass: organization
o: Organismo Ejemplo
dc: midominio
dn: cn=Manager,dc=midominio,dc=org
objectclass: organizationalRole
cn: Manager
```

A continuación crearemos dentro de nuestro árbol una serie de grupos que nos permiten tener una estructura más organizada dentro del directorio.

```
dn: ou=grupos,dc=midominio,dc=org
objectclass: top
objectclass: organizationalUnit
ou: grupos
dn: ou=personas,dc=midominio,dc=org
objectclass: top
objectclass: organizationalUnit
ou: personas
dn: cn=usuarios,ou=grupos,dc=midominio,dc=org
objectclass: top
objectclass: posixGroup
cn: usuarios
gidnumber: 2000
```

Aparecen otros objetos que no aparecieron anteriormente como son `organizationalUnit` y `posixGroup`. Su función es crear ramas hijas de la raíz que nos permitirán distribuir nuestros usuarios de una forma más ordenada. Inicialmente nuestros usuarios colgarán de la rama *personas*.

```
dn: cn=Juan Lopez Perez,ou=personas,dc=midominio,dc=org
objectclass: person
objectclass: inetOrgPerson
mail: juan.lopez@midominio.org
telephoneNumber: +34-954-55-55-55
sn: Lopez
cn: Juan Lopez Perez
cn: Juan Lopez
dn: cn=Laura Jimenez Lora,ou=personas,dc=midominio,dc=org
objectclass: person
objectclass: inetOrgPerson
mail: laura.jimenez@midominio.org
telephonenumber: +34-954-59-59-59
sn: Jimenez
cn: Laura Jimenez Lora
cn: Laura
dn: cn=Jose Fernandez Diaz,ou=personas,dc=midominio,dc=org
objectclass: person
objectclass: inetOrgPerson
objectclass: posixAccount
mail: jose.fernandez@midominio.org
telephoneNumber: +34-954-56-56-56
sn: Diaz
```

```

cn: Jose Fernandez Diaz
cn: Jose Fernandez
uid: jose.fernandez
userPassword: hola
gecos: Jose Fernandez Lopez
uidNumber: 2000
gidNumber: 2000
homeDirectory: /home/jose.fernandez
loginShell: /bin/bash

```

Los usuarios que definimos anteriormente no presentan variación, únicamente los hemos situado dentro de la rama *personas*, pero el nuevo usuario que introducimos tiene en su definición una nueva clase de objeto, fundamental para nuestro objetivo. La clase `posixAccount` tiene como atributos los necesarios para almacenar los datos de una cuenta de sistema. Los campos que utilizamos en este usuario nos recuerdan a los que se almacenan en el fichero `/etc/passwd` y serán los que utilizaremos para la definición de las cuentas de nuestros sistemas en el directorio LDAP.

`uid` Define el identificador de usuario en el sistema

`userPassword` Define la clave correspondiente al uid

`gecos` Almacena la descripción de la cuenta de usuario

`uidNumber` Almacena el número de identificación del usuario

`gidNumber` Almacena el número de grupo del usuario

`homeDirectory` Almacena la ruta del directorio \$HOME

`loginShell` Almacena la ruta completa de la shell del usuario

4.4.2. Configuración de Name Service Switch

La gestión de los permisos de los ficheros del sistema se hace a partir los número que definen el uid (*user identification*) y gid (*group identification*). Para simplificar su identificación se le asignan nombres a los usuarios y a los grupos de forma que podamos identificarlos de forma más sencilla¹².

Para que el resto de utilidades del sistema sepan a qué nombres de usuarios y grupos pertenecen los ficheros se realizan una serie de llamadas a las funciones de la librería `glibc`. Estas llamadas darán como resultado la relación uid/gid nombre identificativo¹³.

En el fichero `/etc/nsswitch.conf` se indica al sistema dónde debe buscar el propietario de un fichero o directorio a partir de su uid. Las fuentes para las "bases de datos" y su orden de búsqueda se especificarán en este fichero.

```

# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the 'glibc-doc' and 'info' packages installed, try:
# 'info libc "Name Service Switch"' for information about this file.
passwd:          compat
group:           compat
shadow:         compat
hosts:          files dns
networks:       files
protocols:      db files
services:       db files

```

¹²Es más fácil identificar un fichero o directorio como perteneciente al usuario `legolas` que al uid `2546`. Lo mismo ocurre con los grupos.

¹³También pueden gestionarse otras relaciones como las referidas a dirección IP y nombre de `host`

```
ethers :      db files
rpc :        db files
netgroup :   nis
```

Modificaremos las primeras entradas de este fichero de la siguiente forma:

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

De esta forma cuando un programa solicite "el nombre de usuario con uid 2345" se buscará primero en el fichero `/etc/passwd` y posteriormente en el directorio LDAP definido en `/etc/ldap/ldap.conf`.

En otras distribuciones podremos tener los parámetros **files ldap**

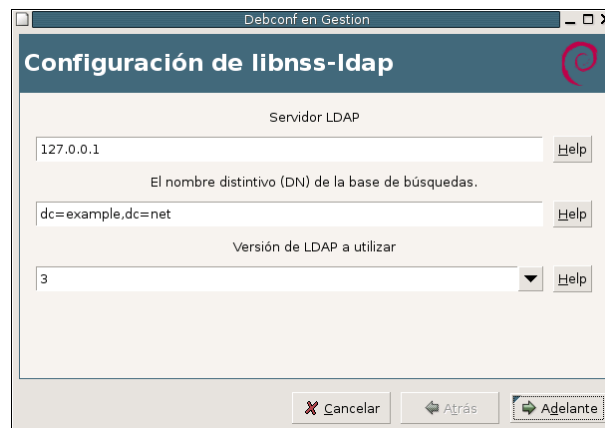
Para realizar esta función es necesario tener instalada la librería `libnss-ldap` que ejercerá de cliente para dar soporte LDAP al servicio NSS (*Name Service Switch*).

Con Guadalinex:

libnss-ldap

El fichero de configuración es `/etc/libnss-ldap.conf`, donde estableceremos los parámetros necesarios para realizar la autenticación

A la hora de la instalación en Guadalinex nos aparecerá un gestor de configuración en el que debemos introducir los datos del directorio, decir si queremos que se exija contraseña para consultar, si sólo el propietario podrá escribir y los datos del usuario de consulta que deberá tener los suficientes permisos.



Una vez terminada la preconfiguración podemos encontrar un ejemplo de `nsswitch.conf` en `/usr/share/doc/libnss-ldap/examples/nsswitch.ldap`. Además todos los parámetros del asistente pueden modificarse y deben repasarse en `/etc/libnss-ldap.conf`. Donde también se definen las ramas del directorio donde consultar cada uno de los elementos (`passwd`, `shadow`, ...) y los parámetros de pam para la autenticación.

```
uri ldap://ldap.midominio.org/ ldap://ldap-copia.midominio.org/
base dc=midominio, dc=org
```

`nss-ldap` espera que las cuentas sean objetos con los siguientes atributos: `uid`, `uidNumber`, `gidNumber`, `homeDirectory`, y `loginShell`. Estos atributos son permitidas por el `objectClass posixAccount`.

4.4.3. Módulos PAM

Lo siguiente será configurar los módulos PAM (*Pluggable Authentication Modules*). Para poder configurar el cliente PAM tendremos que instalar el paquete `libpam-ldap`

Guadalinex

`libpam-ldap`

En esta instalación de la misma forma que las anteriores se utiliza el asistente de configuración para los parámetros más importantes (administrador, password, tipo de cifrado,...)



Este asistente lo que configura son los ficheros `/etc/pam_ldap.conf` y `/etc/ldap.secret`.

Hay varios programas que pueden usar un método de autenticación "centralizado" usando los módulos PAM. Son unas librerías que sirven de interfaz contra varios métodos de autenticación (en nuestro caso LDAP).

Estos módulos son unas librerías que sirven como interfaz entre los programas y distintos métodos de autenticación. En nuestro caso los configuraremos para la autenticación contra LDAP.

La configuración de estos módulos en Guadalinex se realiza en el directorio `/etc/pam.d` y tenemos un fichero de configuración por cada servicio.

```
uri ldap://ldap.midominio.org/
base dc=midominio,dc=org
pam_password [exop/crypt]
```

Las directivas `uri` y `base` trabajan del mismo modo que en `/etc/libnss_ldap.conf` y `/etc/ldap/ldap.conf`.

Si es necesario que la conexión sea con privilegios, se usará la contraseña almacenada en `/etc/ldap.secret`.

Vemos algunos ejemplos.

Consideremos primeramente una distribución genérica con el programa `login`, el cual maneja el `login` desde una consola de texto. Una pila típica de PAM, que chequea las contraseñas, tanto en `/etc/passwd` como en la base de datos LDAP. En el fichero `/etc/pam.d/login`:

```
auth required pam_nologin.so
auth sufficient pam_ldap.so
auth sufficient pam_unix.so shadow use_first_pass
auth required pam_deny.so
```

Para aquellas aplicaciones que utilicen las contraseñas en el fichero `/etc/pam.d/passwd`

```
password required pam_cracklib.so
password sufficient pam_ldap.so
password sufficient pam_unix.so
password required pam_deny.so
```

En el caso de una distribución Debian como Guadalinex:

`/etc/pam.d/common-auth`:

```
auth sufficient pam_ldap.so
auth required pam_unix.so nullok_secure try_first_pass
```

/etc/pam.d/common-account:

```
account sufficient pam_ldap.so
account required pam_unix.so
```

/etc/pam.d/common-password:

```
password sufficient pam_ldap.so
password required pam_unix.so nullok obscure min=4 max=8 md5
```

Una herramienta importante para utilizar ldap para autenticación es el paquete de Migrationtools. Este paquete, instalable mediante `apt-get`, contiene una serie de script en perl que nos permiten una vez configurado el fichero `/etc/migrationtools/migrate_common.ph` convertir nuestros ficheros de usuarios a ficheros en formatos ldif para cargarlos en el ldap.

```
#> cd /usr/share/migrationtools
#> ./migrate_group.pl /etc/group /tmp/group.ldif
#> ./migrate_passwd.pl /etc/passwd |grep -
v 'objectClass: account' > /tmp/passwd.ldif
```

4.4.4. nscd



Aunque a continuación se explique el demonio `nscd`, no se recomienda su utilización, ya que puede haber problemas al autenticarse porque dicho servicio deje bloqueada la máquina. Es conveniente, probar en la distribución y versiones que estemos utilizando antes de ponerlo en producción.

Para evitar que sea consultado el servidor LDAP cada vez que es ejecutado un comando como `ls -l` dentro de nuestra organización, es una buena idea configurar en nuestras estaciones de trabajo un sistema de cache para algunos datos de usuario. Mientras los datos en la cache sean lo suficiente recientes, las estaciones de trabajo utilizarán estos en vez de preguntar al servidor LDAP otra vez. El demonio servidor de caché de nombres (`nscd`) cumple exactamente esta tarea.

Para instalar `nscd`:

```
# apt-get install nscd
```

El fichero de configuración de `nscd` es `/etc/nscd.conf`.

```
/etc/nscd.conf
enable-cache passwd yes
positive-time-to-live passwd 600
negative-time-to-live passwd 20
suggested-size passwd 211
check-files passwd yes
```



Capítulo 5

Compartir impresoras:Cups

Durante mucho tiempo, la configuración de impresoras ha sido uno de los dolores de cabeza para los administradores de linux y UNIX. (*El Libro Oficial de Red Hat Linux: Guía del administrador*)

5.1. Introducción

En Linux cuando queremos imprimir un trabajo el sistema lo envía a la impresora en lenguaje Postscript, se trata de un lenguaje que le dice a la impresora cómo tiene que imprimir la página. Aquí es donde pueden surgir problemas dependiendo del tipo de impresora que tengamos:

- impresoras Postscript: la impresora interpreta directamente las páginas enviadas por el sistema y no vamos a tener ningún problema con ellas. La única “pega” suele ser su precio, ya que la mayoría de ellas son de gama alta.
- impresoras que tienen su propio lenguaje: como no interpretan el lenguaje Postscript, es el sistema el que tiene que traducir la salida Postscript al lenguaje propio de la impresora. Si los fabricantes lo han dado a conocer no suele haber problemas con ellas (algunas empresas proporcionan controladores libres para sus modelos) y están soportadas bajo Linux. Este tipo de impresoras suelen ser las más frecuentes por su bajo precio.
- Winimpresoras: utilizan drivers específicos de Windows para interpretar el lenguaje Postscript, son las más difíciles (¡o imposibles!) de configurar bajo Linux.

CUPS (*Common Unix Printing System*) es un servidor de impresión pensado para gestionar una red en la que una o varias impresoras tienen que dar servicio a todos los equipos interconectados por esa red. Con él disponemos de un sistema de impresión portable y estándar (IPP/1.1¹) dentro del mundo GNU/Linux.

IPP define un protocolo estándar para impresoras y para el control de los servicios de impresión. Como todos los protocolos basados en IP, IPP se puede usar tanto en una red local como en una intranet. A diferencia de otros protocolos, soporta control de acceso, autenticación y encriptación, proporcionando soluciones más idóneas y seguras que con los antiguos protocolos.

Características más destacadas de CUPS:

- Posee un interfaz web para poder configurar el servidor
- Cuotas y administración de trabajos y páginas.
- Detección automática de impresoras de red.

¹*Internet Printing Protocol.*

- Servicios de directorio de impresoras en red.
- Soporta control de acceso, autenticación y encriptación.

Sitios Web de utilidad

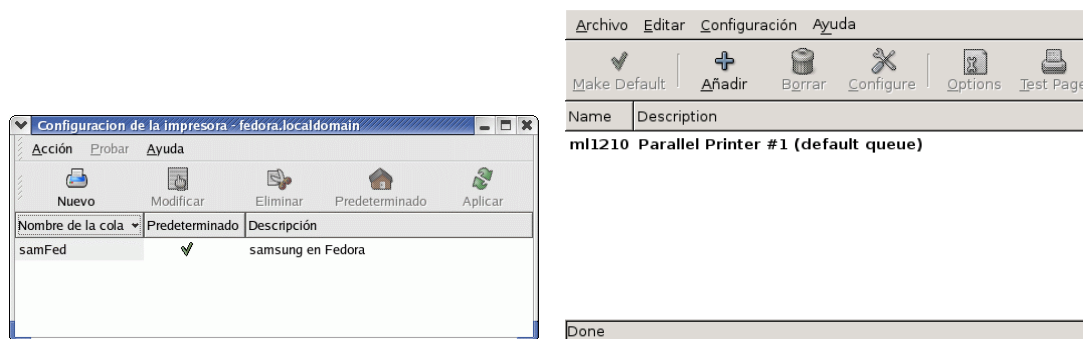
- GNU/Linux Printing <http://www.linuxprinting.org> todo lo que necesitamos conocer sobre la posibilidad de configurar nuestra impresora con Linux
- <http://www.cups.org/> Web de CUPS.



Tanto Fedora como Guadalinex disponen de herramientas gráficas de configuración de la impresora. No vamos a trabajar con ninguna de ellas ya que no suponen una ventaja sustancial sobre el método general que vamos a estudiar. Las herramientas gráficas son:

Fedora **system-config-printer**². Si bien su uso es simple y es una herramienta potente, tenemos que tener en cuenta que sobrescribe los cambios que realicemos “a mano” en el fichero de configuración de CUPS.

Debian **foomatic-gui**³



(a) system-config-printer

(b) foomatic-gui

Figura 5.1: Herramientas gráficas de configuración

Hemos optado por el método más general, consistente en el interfaz Web de CUPS y el fichero de configuración del servidor.

5.2. Instalación



En ambos sistemas supondremos que la instalación se hace con conexión a Internet. Si no es así, la única diferencia estriba en bajar los paquetes y, una vez que están en nuestra máquina, usar `rpm -ivh paquete` (para Fedora) o `dpkg -i paquete` (para Debian).

²En modo texto `system-config-printer-tui`

³La aplicación gráfica por excelencia para configurar las impresoras, `gnome-cups-manager`, tiene un “bug” que impide añadir impresoras. Ese problema no se resuelve (por ahora) aunque actualicemos el sistema.

5.2.1. Fedora

Se instala por defecto, aunque siempre es recomendable que comprobemos si tenemos instalada la última versión del paquete con⁴:

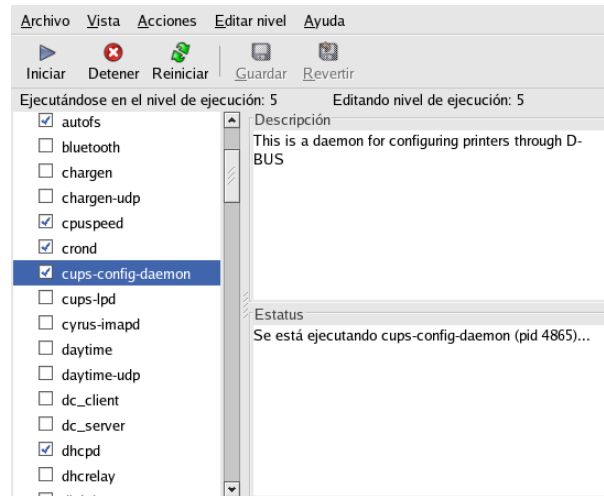
```
# apt-get update; apt-get install cups cups-libs
```

Para reiniciar el servicio podemos usar:

```
#service cups restart
```

O en modo gráfico:

```
#system-config-services
```



Una vez instalado, además de las páginas man del programa es obligado visitar el directorio:

```
/usr/share/doc/cups-x.x.x/
```

5.2.2. GuadaLinux

Si bien debe de estar instalado, lo mejor es que actualicemos a la última versión disponible:

```
# apt-get install cupsys
```

y que instalemos también el paquete⁵

```
# apt-get install cupsys-client
```

cupsys Servidor de impresión CUPS

cupsys-bsd comandos BSD para CUPS

cupsys-client programa cliente de CUPS

Para reiniciar el servicio

```
# /etc/init.d/cupsys start
```

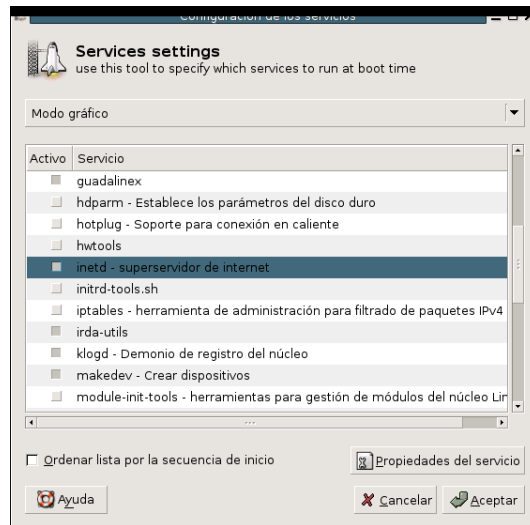
o en modo gráfico lo podemos hacer con

```
# services-admin
```

⁴O con

```
# yum install cups
```

⁵No es imprescindible. Además, deberíamos garantizarnos de que los paquetes **foomatic-filter** (filtros usados por las colas de impresión para convertir los datos de entrada PostScript en el formato nativo que usa una impresora en concreto) y **foomatic-db** (base de datos con filtros de impresora) estuviesen también actualizados.



- Una vez instalado, además de las páginas man del programa es obligado visitar los subdirectorios:
/usr/share/doc/cups*
- Por defecto sólo permite que imprimamos desde la propia máquina. Tendremos que adecuar el fichero de configuración (lo vemos en un par de páginas) para que nos permita imprimir desde la red.

5.3. Configuración de CUPS

Para entender cómo funciona CUPS necesitamos introducir algunos conceptos:

Cola de impresión lista de trabajos pendientes de imprimir. Se imprime el primero que llega (como en una cola de un cine, el primero que se pone en la fila es el primero que saca la entrada)

Clases se trata de una abstracción sobre la idea de impresora. Una clase (Dpto de matemáticas, Dpto de Inglés, ...) puede estar compuesta por varias impresoras. Si mandamos un trabajo a una clase, CUPS se encarga de repartir el trabajo de forma óptima entre las impresoras que componen esa clase.

Filtros reglas que usa CUPS para traducir los trabajos a imprimir al modelo concreto de impresora.

Siguiendo un orden cronológico, para imprimir un trabajo lo haremos con `lpr6`, que lo manda a un directorio de spool. Del directorio de spool lo coge el demonio `cupsd` que lo enviará a la impresora física correspondiente, pasándole el filtro adecuado. Si no lo puede mandar inmediatamente a la impresora, lo dejará en el directorio de spool en espera de que llegue su turno o la impresora esté preparada.

Los ficheros de configuración de CUPS se localizan en `/etc/cups`. Los ficheros estándar de configuración son:

client.conf opciones de configuración para los clientes de impresión.

⁶O los comandos o iconos gráficos que a su vez llaman a éste

cupsd.conf permite configurar el demonio de impresión de CUPS (`/usr/sbin/cupsd`)

Además de los dos ficheros anteriores hay otros. En general estos últimos no se modifican “a mano” ya que su contenido se obtiene a partir del interfaz Web (o del comando `lpadmin`). Los más importantes y que aparecen de forma estándar en ambas distribuciones son:

classes.conf en él se almacena la información sobre las clases de impresión.

mime.convs lista de programas conversores a usar para convertir de un tipo MIME a otro

mime.types le dice a CUPS cómo reconocer un tipo de dato a partir de números mágicos dentro del archivo

printers.conf archivo de configuración de las impresoras

5.3.1. client.conf

En este fichero configuramos las opciones de las máquinas clientes. Las opciones en este fichero en general, no deberíamos cambiarlas.

```
#ServerName myhost.domain.com
```

CUPS puede configurarse para que los trabajos pendientes de imprimir se almacenen en la máquina servidor y que no se use el directorio de spool local. Si lo que deseamos es que los trabajos pendientes se almacenen sólo en el servidor, debemos activar la directiva nombre del servidor, en ella pondríamos el nombre de nuestro servidor de impresión (si es una red local lo normal es que aquí esté la IP local del servidor de impresión). En general, es mejor dejarlo todo como está ya que una caída del servidor acarrearía la pérdida de los trabajos de impresión.

Con la directiva **Encryption**, podemos optar por el tipo de encriptación a usar, los valores posibles son:

Always usar siempre encriptación (SSL)

Never no usar nunca encriptación

Required usa actualización de encriptación TLS

IfRequested usa encriptación si el servidor lo pide, es la directiva por defecto.

```
#Encryption Always
#Encryption Never
#Encryption Required
#Encryption IfRequested
```

5.3.2. cupsd.conf

Las entradas de este fichero son sencillas de manipular⁷ y entender (se parecen a las del fichero de configuración de Apache).

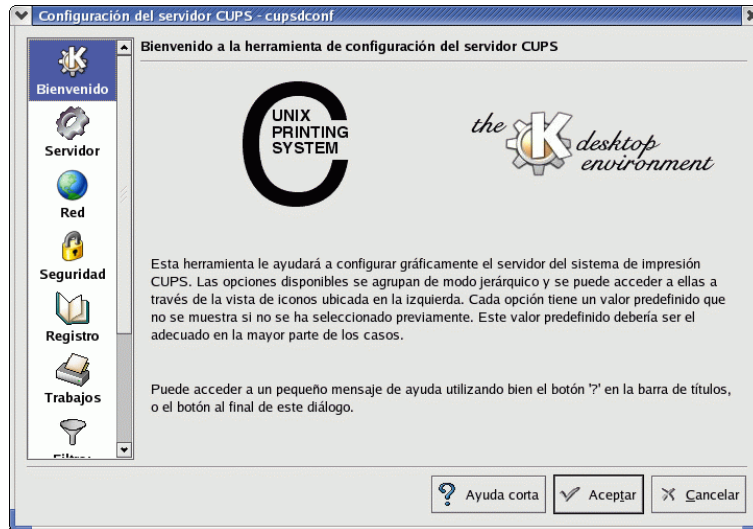
Con ellas podemos configurar, además de aspectos generales del servidor, el registro, las rutas de los directorios del servidor, el cifrado y soporte de certificados, ... En general no hay que modificarlas para disponer del servicio de impresión. Sólo puede ser necesario hacerlo para permitir accesos vía red y en este caso nos limitaremos a añadir las entradas adecuadas al final del fichero (véanse los ejemplos en la página 82)

⁷Si se va a modificar es conveniente crear antes una copia del original por si “metemos la pata” en algo.

```
#cp cups.conf cups.conf.original
```

cupsdconf

Si hemos instalado el KDE, podemos usar la herramienta gráfica de configuración de CUPS⁸ #cupsdconf



para ajustar nuestro fichero `/etc/cups/cupsd.conf`. Si bien no soporta todas las opciones de configuración⁹ sí que permite configurar las más usuales. La ayuda corta que acompaña al programa es muy buena y hace que usar esta herramienta sea un “juego de niños”.



Desde ella podemos configurar fácilmente el demonio de impresión, pero para explicar las distintas opciones hemos optado por el método más general, es decir, usar el fichero de configuración. Es un buen ejercicio usar la herramienta gráfica anterior y después revisar qué directivas de configuración son las que hemos escrito en el fichero `/etc/cups/cupsd.conf`

El fichero

La mayoría de las directivas están muy bien comentadas en el propio fichero. Estudiemos el contenido del archivo¹⁰, hemos mantenido el # en aquellas directivas que aparecen comentadas

⁸En general debe estar a nuestra disposición en ambas distribuciones, si no es así, los instalamos con:

```
Debian # apt-get install kdelibs-bin
Fedora # apt-get install kdelibs
```

⁹Si detecta una directiva no soportada, nos avisa de ello y de que no permitirá su modificación.

¹⁰

- Para Fedora, en el de Debian cambian un poco los valores por defecto, no las directivas de configuración.
- Está claro que no hay que conocerlas ni todas ni de memoria -). El incluirlo completo es con la idea de que dispongáis de una referencia rápida de las posibilidades de configuración de CUPS



por defecto, los valores por defecto los escribiremos entre paréntesis y además, para simplificar su lectura, mantendremos la organización por secciones:

Server Identity En esta sección podemos configurar el nombre del servidor, por defecto se usará el nombre del sistema. También podemos determinar la cuenta de correo a la que se envíen las quejas cuando los clientes tengan problemas con la impresión.

```
#ServerName myhost.domain.com
#ServerAdmin root@your.domain.com
```

Server Options En primer lugar configuramos el fichero en que se almacenarán los logs del sistema, si bien aparece comentado se toma como valor por defecto. En él se almacenan los datos en el llamado "Formato de registro común", esto significa que podemos usar cualquier herramienta de informe de registro de acceso para generar informes de la actividad del servidor (por ejemplo Webalizer¹¹).

```
#AccessLog /var/log/cups/access_log
```

Las directivas que siguen (por defecto están inhabilitadas) permiten que en los trabajos de impresión aparezca el texto que sigue a **Classification** junto con páginas informativas sobre el trabajo de impresión (identidad del trabajo, usuario, etc).

```
#Classification classified
#Classification confidential
#Classification secret
#Classification topsecret
#Classification unclassified
```

Si deseamos permitir que los usuarios puedan modificar los parámetros de la directiva anterior (**Classification**) pondremos la directiva que sigue en **on** (**off**). En este caso podrán limitar la primera y última página que se añade al trabajo, así como cambiar el tipo de clasificación.

```
#ClassifyOverride off
```

Directorio de datos de CUPS, codificación (por defecto UTF-8) y lenguaje por defecto.

```
#DataDir /usr/share/cups
#DefaultCharset utf-8
#DefaultLanguage en
```

Está claro que una primera modificación consiste en adecuar el lenguaje por defecto a **es**. Directorio donde se localiza la ayuda del programa¹².

```
#DocumentRoot /usr/share/doc/cups-1.1.22
```

Archivo de registro de errores y permisos del fichero en donde se almacenan, tal cual está sólo el root podrá ver su contenido.

```
#ErrorLog /var/log/cups/error_log
LogFilePerm 0600
```

Para permitir imprimir sobre un archivo pondremos la directiva que sigue en **Yes** (por defecto es **No** para limitar problemas de seguridad), de esa forma se admitirán URIs de la forma **file:/tmp/print.prn**. Si no se tiene claro es mejor dejarlo en **No**.

¹¹Se estudia en entregas posteriores.

¹²En Guadalinex es **/usr/share/cups/doc-root**

```
#FileDevice No
```

Especifica la ruta por defecto de los archivos de fuentes.

```
#FontPath /usr/share/cups/fonts
```

Nivel de registro de errores, por defecto `info`. Puede ser:

<code>debug2</code>	Registra todo.	<code>warn</code>	Registro de errores y avisos.
<code>debug</code>	Registra casi todo.	<code>error</code>	Sólo registra errores.
<code>info</code>	Registra las peticiones y los cambios de estado.	<code>none</code>	No registra nada.

En general está bien así y sólo en el caso de que tengamos problemas que no sabemos resolver podemos optar por `debug2`. En ese caso el fichero de registro de errores se puede hacer enorme¹³. Con `MaxLogSize` establecemos el límite del tamaño, en bytes, de los archivos de registro de errores (si se comenta toma de valor por defecto `1048576=1MB`) antes de que sean rotados (si optamos por poner `0` desactivamos la rotación).

```
LogLevel info
MaxLogSize 200000000
```

Archivo en dónde almacenar el registro de páginas¹⁴, en cada línea de ese fichero se almacena el nombre de la impresora, el de usuario, los IDs de los trabajos, fecha de impresión, número de páginas del trabajo, y número de copias de cada página.

↔ Por ejemplo, el trabajo con ID 284 tenía 4 páginas y 1 copia impresa, el trabajo con ID 285 tenía 1 copia de sólo 1 página.

```
# tail -f /var/log/cups/page_log
lp0 paco 284 [21/Feb/2004:16:11:14 +0100] 3 1
lp0 paco 284 [21/Feb/2004:16:11:14 +0100] 4 1
lp0 paco 285 [21/Feb/2004:17:09:11 +0100] 1 1
```

```
#PageLog /var/log/cups/page_log
```

Las tres opciones que siguen nos permiten:

- preservar el historial de un trabajo (**Yes**) después de finalizar
- preservar los archivos de un trabajo (**No**)
- purgar (con el sistema de cuotas activo) automáticamente los trabajos completados que no se necesitan más (**No**)

```
#PreserveJobHistory Yes
#PreserveJobFiles No
#AutoPurgeJobs No
```

Las directivas que siguen nos permiten:

¹³Parar poder ver qué está pasando podemos ejecutar:

```
#tail -f /var/log/cups/error_log
```

¹⁴Para que se calcule correctamente el número de páginas el trabajo ha de pasar a través del filtro `pstops`. Si el trabajo se ha pasado a través de una cola "en bruto", el número de páginas no se contabilizará bien (en general, aparecen como 1 trabajo de 1 página con múltiples copias).



- Número máximo de copias que un usuario puede solicitar (100)
- Número máximo de trabajos que se mantienen almacenados (500), una vez que el número de trabajos alcanza el límite se borran los más antiguos. Si los trabajos están aún sin finalizar se rechaza el trabajo nuevo. Si optamos por el valor 0 no hay límite.
- Número máximo de trabajos activos para cada impresora o clase (por defecto 0).
- Número máximo de trabajos por usuario (0)
- Si se pone a cero se desactiva. Controla el número máximo de colecciones de históricos del atributo `printer-state-history`

```
#MaxCopies 100
#MaxJobs 500
#MaxJobsPerPrinter 0
#MaxJobsPerUser 0
#MaxPrinterHistory 10
```

El nombre del archivo *printcap*. Es necesario mantenerlo activo para que funcionen bien ciertas aplicaciones antiguas que precisan un archivo de ese tipo. Con la directiva siguiente optamos por seleccionar el formato de este archivo, por defecto BSD

```
Printcap /etc/printcap
#PrintcapFormat BSD
#PrintcapFormat Solaris
```

Con las 4 directivas que siguen determinamos:

- Directorio donde se almacenan las solicitudes de impresión
- Nombre de usuario que se asigna para accesos no autenticados desde sistemas remotos. Por defecto es `remroot`. Este nombre aparecerá en archivos de registro y solicitudes para todos aquellos recursos y direcciones del servidor que permiten acceso sin autenticación. Las entradas autenticadas contendrán los nombres autenticados.
- Directorio para los ejecutables del servidor de impresión, por defecto `/usr/lib/cups`
- Directorio que contiene los archivos de configuración (`/etc/cups`)

```
#RequestRoot /var/spool/cups
#RemoteRoot remroot
#ServerBin /usr/lib/cups
#ServerRoot /etc/cups
```

Con la directiva `ServerTokens` especificamos qué información se devuelve en el encabezado de las peticiones HTTP, por defecto es `Minor`.

```
# ServerTokens None
# ServerTokens ProductOnly CUPS
# ServerTokens Major CUPS/1
# ServerTokens Minor CUPS/1.1
# ServerTokens Minimal CUPS/1.1.22rc1
# ServerTokens OS CUPS/1.1.22rc1 (uname)
# ServerTokens Full CUPS/1.1.22rc1 (uname) IPP/1.1
```

Fax Support Con las directivas de este apartado establecemos el número de veces que se reintenta un trabajo de fax (5) así como el intervalo de tiempo (300 segundos) entre reintentos.

```
#FaxRetryLimit 5
#FaxRetryInterval 300
```

Encryption Support Si usamos comunicaciones seguras, ficheros que contienen el certificado y la clave del servidor (respectivamente)

```
#ServerCertificate /etc/cups/ssl/server.crt
#ServerKey /etc/cups/ssl/server.key
```

Filter Options Establecemos el usuario y grupo bajo los que se ejecuta el servidor de impresión (lo usual es dejar los valores por defecto: lp y sys respectivamente).

```
#User lp
#Group sys
```

Cantidad de memoria que cada RIP (*Raster Image Processor*) debe utilizar para usar el caché de mapa de bits (por defecto 8m). El valor puede ser un número seguido de

- k para kilobytes
- m para megabytes
- g para gigabytes
- t para “baldosas”, donde 1t=256 x 256 pixels.

```
#RIPCache 8m
```

Directorio para archivos temporales

```
#TempDir /var/spool/cups/tmp
```

Establece el coste máximo de todos los filtros de los trabajos que se están ejecutando al mismo tiempo. Un trabajo medio de una impresora no PostScript necesita un límite de filtro de al menos 200 (la mitad si es PostScript). Límites inferiores al mínimo requerido por un trabajo hacen que sólo se imprima un trabajo cada vez. El valor por defecto es 0 (ilimitado).

```
#FilterLimit 0
```

Network Options Las directivas de esta sección nos permiten configurar las opciones de red del servidor. Con **Port** establecemos el puerto en que escucha el servidor (631). La directiva **Listen** nos permite establecer la dirección en la que escuchar. Podemos tener varias entradas para cada una de ellas.

```
# Port 80
# Port 631
# Listen hostname
# Listen hostname:80
# Listen hostname:631
# Listen 1.2.3.4
# Listen 1.2.3.4:631
#Port 80
#Port 443
Listen 127.0.0.1:631
```


Nos permite optar por trabajar con nombres de máquina totalmente cualificados. Para mejorar las prestaciones por defecto está en `Off`

```
#HostNameLookups On
```

Con esta directiva y la siguiente obligamos a que el servidor se mantenga activo 60 segundos en espera de servir nuevos clientes desde la misma conexión.

```
#KeepAlive On
#KeepAliveTimeout 60
```

Las cuatro directivas que siguen permiten determinar:

- Número máximo de clientes a la vez (100)
- Número máximo de clientes para un host determinado, por defecto 0: significa 1/10 del número máximo de clientes.
- Tamaño máximo del trabajo de impresión (0, significa ilimitado)
- Tiempo de espera (300 segundos) antes de que expiren las peticiones

```
#MaxClients 100
#MaxClientsPerHost 0
#MaxRequestSize 0
#Timeout 300
```

Browsing Options Por defecto se difunden las impresoras a las máquinas de la red, usando el protocolo CUPS. Con `BrowseAddress` podemos establecer la dirección de multidifusión.

```
Browsing On
BrowseProtocols cups
#BrowseAddress x.y.z.255
#BrowseAddress x.y.255.255
#BrowseAddress x.255.255.255
#BrowseAddress 255.255.255.255
#BrowseAddress @LOCAL
#BrowseAddress @IF(name)
```

Con la opción siguiente (`Yes`) optamos por usar nombres cortos para impresoras remotas (`impresora` en lugar de `impresora@host`)

```
#BrowseShortNames Yes
```

Podemos permitir o denegar la exploración usando las dos directivas que siguen. Por defecto se permite la entrada a paquetes de cualquier dirección local y no se deniega ningún paquete.

```
BrowseAllow from @LOCAL
#BrowseDeny address
```

Los parámetros permitidos son¹⁵:

¹⁵Para que podamos usar nombres de máquina o dominios es necesario que esté activada la resolución de nombres.



All	nnn.nnn.nnn.*
None	nnn.nnn.nnn.nnn
*.domain.com	nnn.nnn.nnn.nnn/mm
.domain.com	nnn.nnn.nnn.nnn/mmm.mmm.mmm.mmm
host.domain.com	@LOCAL
nnn.*	@IF(name)
nnn.nnn.*	

A continuación podemos configurar el tiempo entre las actualizaciones de la exploración en segundos (30), si lo establecemos a 0 se desactivan las difusiones. También podemos determinar el orden en que se evalúa la directiva `BrowseOrder`, por defecto `deny,allow`: se deniega todo lo que no se autorice explícitamente.

```
#BrowseInterval 30
#BrowseOrder allow,deny
BrowseOrder deny,allow
```

Puerto utilizado para las difusiones UDP (por defecto IPP)

```
#BrowsePoll address:port
#BrowsePort 631
```

Si deseamos pasar la información desde una red a otra usaremos `BrowseRelay`, ejemplos de uso:

```
#BrowseRelay 172.26.0.2 192.168.0.255
#BrowseRelay 172.26.0.0/24 192.168.0.255
#BrowseRelay source-address destination-address
#BrowseRelay @IF(src) @IF(dst)
```

Caducidad en segundos de las impresoras de red (300 segundos), si en este tiempo no se obtiene una actualización se elimina de la lista de impresoras.

```
#BrowseTimeout 300
```

Permite usar clases implícitas, de esta forma las impresoras de la red que tienen el mismo nombre se ponen en una clase de igual nombre que la impresora¹⁶. Por defecto no creamos una clase implícita para cada una de las impresoras (`ImplicitAnyClasses Off`) y no mostramos las impresoras que componen las clases implícitas.

```
#ImplicitClasses On
#ImplicitAnyClasses Off
#HideImplicitMembers On
```

`Security Options` Grupo de administración del sistema¹⁷ y frecuencia con que se regenera el certificado de autenticación (300)

```
#SystemGroup sys
#RootCertDuration 300
```

Con la directiva `location` podemos establecer controles de autenticación y acceso a determinados recursos del servidor¹⁸:

¹⁶ Así disponemos de colas redundantes múltiples en nuestra red configuradas de forma automática. Si un usuario envía un trabajo a la clase implícita, el trabajo irá a la primera impresora disponible que compone la clase.

¹⁷ En Debian `lpadmin`

¹⁸ No están todas las opciones, para conocer todas las posibilidades os remitimos a la documentación del programa

classes clases de impresoras

classes/name clase *name* del servidor

jobs todos los trabajos

printers todas las impresoras

printers/name la impresora de nombre *name*

admin todas las cuestiones relacionadas con la administración

/ todas las operaciones de configuración del servidor de impresión

Los valores permitidos son:

AuthType tipo de autenticación, puede ser

none no se establece ningún método de autenticación

basic en este modelo es necesario un nombre de usuario y contraseña (que se pasan en texto plano) para autenticarse ante el servidor

digest más segura que la anterior¹⁹.

AuthClass nivel de autenticación, puede ser

Anonymous valor por defecto, no se necesita autenticación

User es necesario un nombre de usuario y contraseña

System es necesario un nombre de usuario y contraseña, el usuario ha de pertenecer al grupo del sistema (**sys** y **lpadmin** para Fedora o Guadalinex respectivamente)

Group es necesario un nombre de usuario y contraseña, el usuario ha de pertenecer al grupo definido en la directiva **AuthGroupName**

AuthGroupName nombre del grupo para la autorización comentada antes

Order orden en que se analizan las directivas **allow** y **deny**

Allow From permite el acceso desde determinados máquinas, dominios, ...

Deny From deniega el acceso desde determinados máquinas, dominios, ...

La notación permitida para las dos directivas anteriores se ha comentado ya. Se puede consultar en la 5.3.2 en la página 81

```
#<Location /classes>                                #<Location /printers>
#</Location>                                         #</Location>

#<Location /classes/name>                            #<Location /printers/name>
#</Location>                                         #AuthType None
                                                       #AuthType Basic
#<Location /jobs>                                    #AuthClass User
#</Location>                                         #AuthType Digest
                                                       #AuthClass User
```

¹⁹Para:

- Conocer mejor las diferencias <http://www.faqs.org/rfcs/rfc2617.html>.
- Saber cómo rabajar con ella: <http://www.cups.org/doc-1.1/sam.html>

```

#Order Deny,Allow           Allow From 127.0.0.1
#Deny From All              #Encryption Required
#Allow From .mydomain.com   </Location>
#</Location>

<Location /admin>           <Location />
  AuthType Basic             Order Deny,Allow
  AuthClass System           Deny From All
  Order Deny,Allow           Allow From 127.0.0.1
  Deny From All              </Location>

```

↔ Veamos un ejemplo de configuración:

```

<Location />
  # Directorio raíz del servidor CUPS
  # Permitimos sólo acceso desde el bucle local y la red local.
  Order Deny,Allow
  Deny From All
  Allow From 127.0.0.1
  Allow From 172.26.0.0/255.255.255.0
  # Además permitimos la administración a la IP remota20
  Allow 150.214.5.11
</Location>

<Location /admin>
  # Configuremos el acceso a la interfaz Web.
  # Autenticación básica y sólo desde la red local
  AuthType Basic
  AuthClass System
  Order Deny,Allow
  Deny From All
  Allow From 127.0.0.1
  Allow From 172.26.0.0./255.255.255.0
</Location>

```

5.4. Interfaz Web

Para acceder a la interfaz web de configuración del programa, abrimos nuestro navegador web y escribimos:

```
http://localhost:631
```

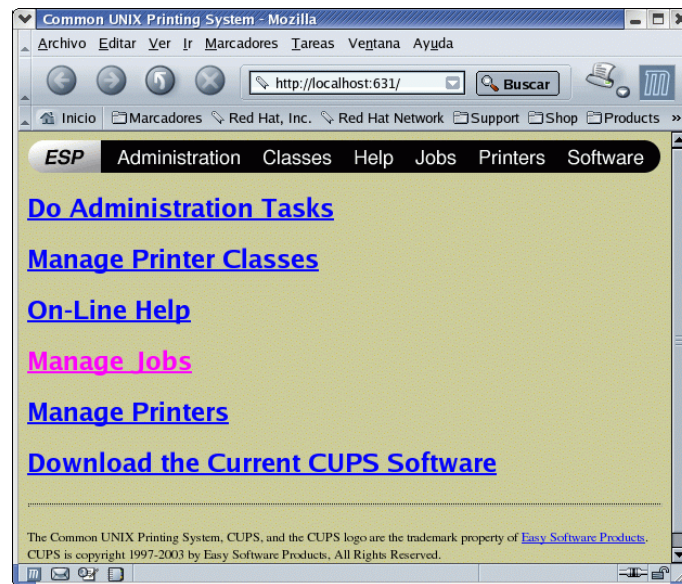
²⁰Si mantenemos sólo la directiva

```
Listen 127.0.0.1:631
```

no podremos acceder desde otras máquinas, es necesario ajustarla añadiendo, por ejemplo:

```
Listen 150.214.5.11:631
```

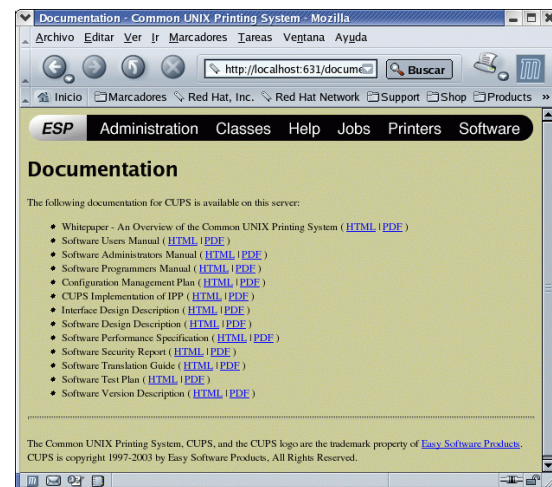
o no limitando el acceso



Para poder cambiar algo se nos pedirá mediante una ventana de autenticación la contraseña del root.

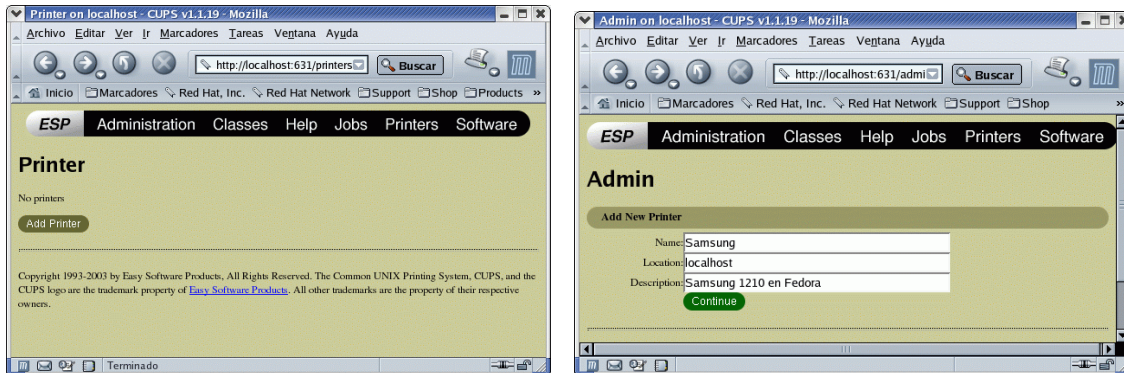
Las opciones disponibles son:

- Tareas administrativas: desde este apartado podemos gestionar las clases, los trabajos de impresión y las impresoras.
- Administrar clases de impresoras
- Ayuda en línea
- Administrar los trabajos de impresión
- Administrar las impresoras
- Acceder a la Web de CUPS por si deseamos bajarnos la última versión del programa.



5.4.1. Añadir una impresora

Pulsamos sobre **Printers** y optamos por añadir una impresora (**Add Printer**). Introducimos el nombre, la localización y una breve descripción sobre la impresora que estamos configurando. Sólo es obligatorio introducir el nombre (es el único campo que no podremos modificar después).



A continuación hemos de optar por el interfaz al que está conectada. Si nuestra impresora es local, sólo hemos de optar por el puerto adecuado (serie, usb, paralelo). Algunas de las opciones son:

- Puerto paralelo
- Puertos USB
- Puertos serie
- Impresora en red compartida mediante el sistema LPD
- Impresora compartida mediante IPP (con otro CUPS o windows 2000)
- Impresora compartida mediante SMB (protocolo de red de windows)



Además de impresoras conectadas de forma local, CUPS soporta los protocolos socket, LPD (“*Line Printer Daemon*”), IPP y smb²¹. En CUPS a cada cola de impresión se le asocia un nombre y un dispositivo. La sintaxis URI con que se especifica cada dispositivo es por ejemplo:

- `parallel:/dev/lp0` para una impresora local conectada al puerto paralelo.
- `lpd://servidor/lp` para una impresora de nombre lp conectada a un “servidor” de impresión UNIX en el que corre un sistema LPD
- `ipp://servidor/impresora` o `ipp://servidor/printers/impresora` en este caso, nuestro servidor utiliza CUPS y deseamos imprimir en la impresora de nombre **impresora**
- `smb://servidor/impresora`, `smb://user:password@workgroup/servidor/impresora` o `smb://user:password@servidor/impresora` si usamos un servidor de impresión basado en Windows.

Podemos conocer los *back-ends* disponible es nuestra máquina listando `/usr/lib/cups/backend22`

Si optamos por seleccionar una impresora de red tendremos que introducir la ruta completa al dispositivo. CUPS nos pone el protocolo que seleccionemos (podemos modificarlo al cambiar la URI). Por ejemplo:

file:/tmp/impresora.prn Imprimirá en el archivo especificado

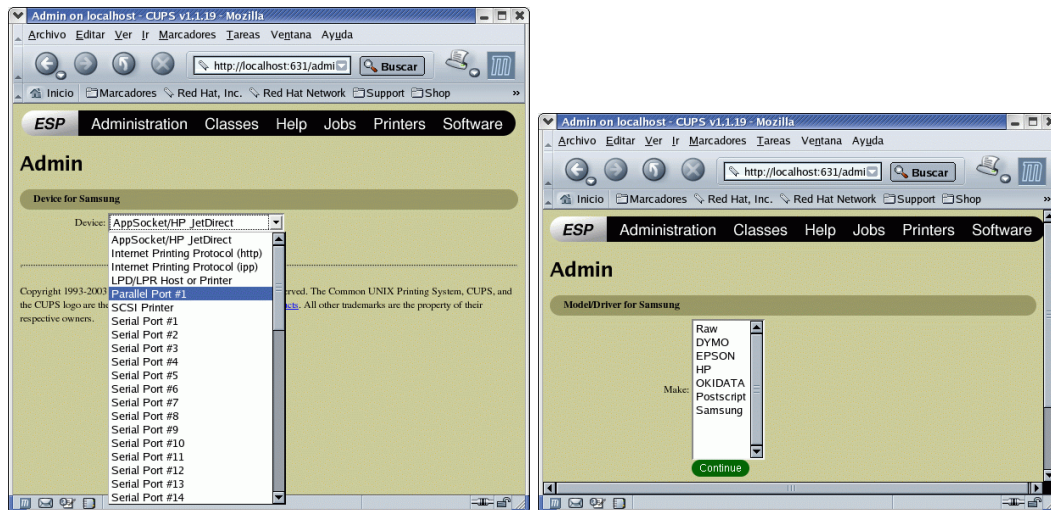
²¹impresión en una impresora compartida de Windows

²²O usar:

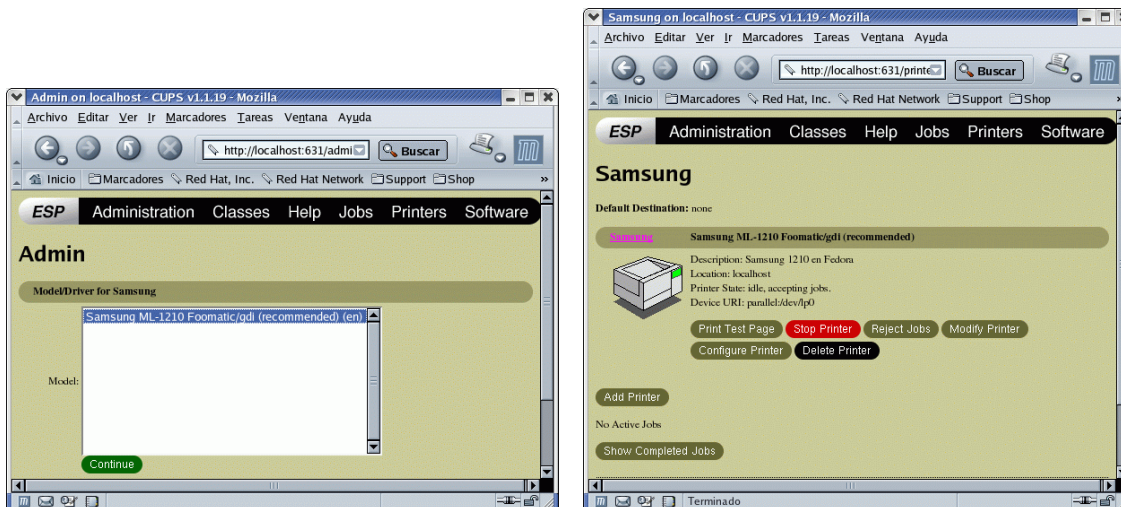
`$/usr/sbin/lpinfo -v`

`ipp://mileto.cica.es/printers/lp0` Imprimirá en la cola lp0 de la máquina mileto
`smb://murgi/tux/epson` Imprimirá en una impresora compartida de una red windows
 con:

- grupo de trabajo murgi
- nombre del servidor tux
- recurso compartido epson



Llega el momento de seleccionar la marca y el modelo de impresora (filtro a usar para nuestra impresora, en el ejemplo capturado se trata de una Samsung ML1210). Una vez que todo está bien ya tenemos nuestra impresora lista para usar. Antes de dar por finalizada la configuración es conveniente imprimir una página de prueba (**Print Test Page**) y, en su caso, ajustar los parámetros de impresión (tipo de papel, resolución, ...)



Si accedemos a la impresora instalada, veremos que disponemos de las opciones:

Print test page imprimir una página de prueba

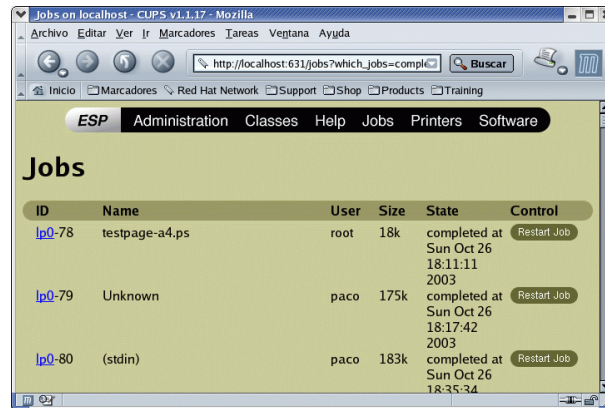
Stop printer para detener los trabajos en curso

Rejects Jobs rechazar trabajos de impresión

Modify Printer cambiar la configuración de la impresora

Configure Printer para ajustar los parámetros finales de impresión: tipo de papel, resolución, añadir un *banner*, ... Dependien del modelo de impresora.

Desde la sección **Jobs** podemos acceder a los trabajos pendientes de imprimir (**Show Active Jobs**) y los trabajos completados (**Show Completed Jobs**)



The screenshot shows a web browser window displaying the CUPS Jobs page. The page has a navigation menu with 'Administration', 'Classes', 'Help', 'Jobs', 'Printers', and 'Software'. The 'Jobs' section is active, showing a table with columns: ID, Name, User, Size, State, and Control. Three jobs are listed, all with a state of 'completed at'.

ID	Name	User	Size	State	Control
lp0-78	testpage-a4.ps	root	18k	completed at Sun Oct 26 18:11:11 2003	Restart Job
lp0-79	Unknown	paco	175k	completed at Sun Oct 26 18:17:42 2003	Restart Job
lp0-80	(stdin)	paco	183k	completed at Sun Oct 26 18:35:34	Restart Job

En esta ventana se nos informa de:

ID impresora responsable del trabajo

Name nombre del fichero impreso

User usuario que ha mandado el trabajo

Size tamaño del fichero

State estado en que se encuentra: activo, cancelado, en espera.

Control si el trabajo está completado podemos imprimirlo de nuevo pulsando sobre **Restart Jobs**.

Si el trabajo está aún activo, nos aparecen las opciones:

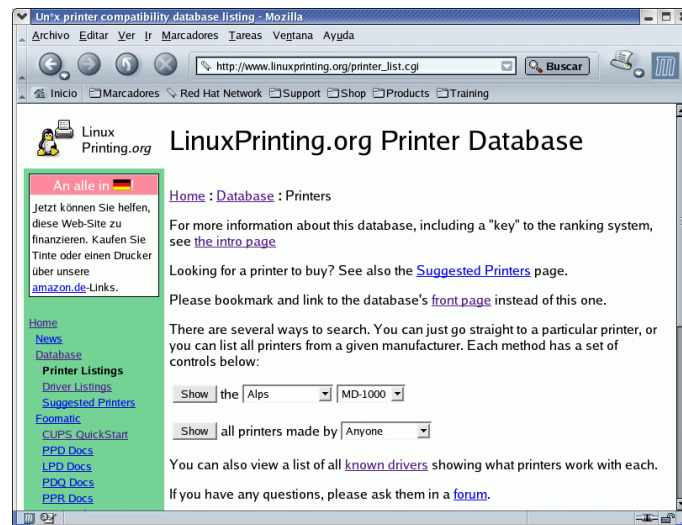
Hold Jobs para detener el trabajo de impresión

Cancel Jobs para cancelar el trabajo

➤ Si el filtro no está instalado

En este caso la página de obligada visita es

http://www.linuxprinting.org/printer_list.cgi



Veamos un par de ejemplos²³::

↪ **Samsung ML1210** Tras consultar en la web anterior nos informan que:

Recommended driver: gdi (Home page, view PPD, download PPD)

Bajamos el driver PPD y lo ponemos en el lugar adecuado

```
# cp Samsung-ML-1210-gdi.ppd /usr/share/cups/model/
```

Tras esto, es fundamental reiniciar el servicio, si no, no lee los cambios.

- Fedora

```
#service cups restart
```

- Debian

```
#/etc/init.d/cupsys restart
```

↪ **Epson C62** Para esta impresora (y tras revisar la página anterior) obtenemos que:

Recommended driver: gimp-print (Home page)

Instalamos el paquete adecuado `gimp-print-cups`

```
#apt-get install gimp-print-cups
```

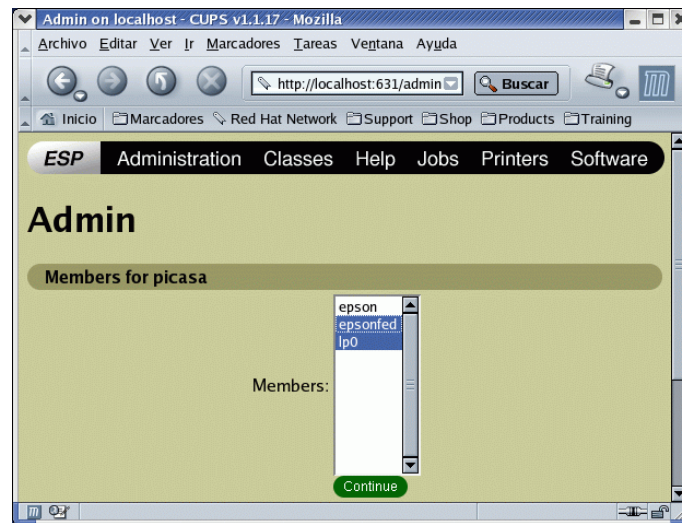
y tras reiniciar el servicio todo solucionado.

5.4.2. Añadir una clase

Una de las características que hacen de CUPS un gran servidor de impresión es la idea de clase. Una clase está formada por una serie de impresoras y es CUPS, cuando imprimimos sobre esa clase, el que se encarga de gestionar la carga: si una impresora está ocupada se envía el trabajo a otra de la misma clase, de esta forma podemos minimizar tiempos de espera.

Para crear una clase optaremos por pulsar sobre **Classes** y tras rellenar el nombre, localización y breve descripción pasamos a seleccionar las impresoras que van a constituir la clase.

²³Se trata de dos ejemplos “ficticios” ya que el filtro para ambos modelos, en general, debería estar ya instalado.



Las posibilidades de administración de una clase son similares a las comentadas sobre impresoras.

5.5. Un poco de comandos

CUPS proporciona los comandos²⁴

/usr/bin/cancel cancela los trabajos de impresión existentes

/usr/bin/disable para la impresora o clase pasada como argumento

/usr/bin/enable arranca la impresora o clase pasada como argumento

/usr/bin/lp imprime el fichero o altera un trabajo pendiente

/usr/bin/lpoptions muestra o establece las opciones de las impresoras

/usr/bin/lppasswd para añadir, cambiar o borrar contraseñas en el fichero de contraseñas de CUPS (`passwd.md5`)

/usr/bin/lpq muestra, para la impresora pasado como parámetro, el estado actual de la cola de impresión

/usr/bin/lpr imprime el fichero pasado como parámetro

/usr/bin/lprm cancela trabajos de la cola de impresión

/usr/bin/lpstat muestra información del estado de las clases, trabajos e impresoras actuales

/usr/sbin/accept indica al sistema de impresión que acepte trabajos para el destino indicado

/usr/sbin/cupsaddsmb exporta impresoras para usarlas con clientes Windows mediante SAMBA

/usr/sbin/cupsd demonio de impresión

/usr/sbin/lpadmin para configurar las impresoras y clases

²⁴En general, en Fedora a algunos se les puede añadir el “apellido” `.cups` (en Debian no aparece). Así, para imprimir un trabajo de nombre `fichero.ps` escribiremos:

```
$lp fichero.ps
```

`/usr/sbin/lpc` permite controlar las impresoras y clases

`/usr/sbin/lpinfo` lista los dispositivos disponibles o los controladores conocidos por el servidor CUPS

`/usr/sbin/lpmove` mueve el trabajo especificado a otro destino

`/usr/sbin/reject` indica al sistema que rechace trabajos de impresión para el destino especificado

Para conocer las opciones de estos comandos os recomendamos usar las páginas man de cada uno de ellos.

El uso de `lpr`, `lprm` y `lpc` es inmediato.

Todo lo que hemos visto usando el interfaz Web (y más) se puede hacer con los comandos anteriores. Uno de los más importantes es:

5.5.1. `lpadmin`

permite añadir impresoras y colas de impresión

Su sintaxis básica es

```
#lpadmin -p mi_impresora -E dispositivo -m driver.ppd
```

con este comando añadimos la impresora de nombre `mi_impresora`

↪ Por ejemplo, para añadir la impresora Samsung de ejemplo, conectada al puerto paralelo, escribiremos

```
#lpadmin -p samsung -E -v parallel:/dev/lp0 -m Samsung-ML-1210-gdi.ppd
```

Para modificar una impresora escribiremos

```
#lpadmin -p mi_impresora -m driver1.ppd
```

si lo que deseamos es trabajar con otro filtro de impresión

Para poner `mi_impresora` como impresora por defecto

```
#lpadmin -d mi_impresora
```

Si queremos eliminar la impresora anterior usaremos

```
#lpadmin -x mi_impresora
```

Para añadir la impresora a la clase pasada de nombre `mi_clase`

```
#lpadmin -p mi_impresora -c clase
```

y, para eliminar la impresora de la clase `mi_clase`

```
#lpadmin -p mi_impresora -r mi_clase
```

Si lo que deseamos es eliminar una clase escribiremos

```
#lpadmin -x mi_clase
```

Si no queremos que dos usuarios de nombre `usuario1` y `usuario2`²⁵ puedan imprimir en la impresora pasada como parámetro escribiremos

```
#lpadmin -p mi_impresora -u deny:usuario1,usuario2
```

y, para que `usuario1` pueda imprimir de nuevo

```
#lpadmin -p mi_impresora -u allow:usuario1
```

`lpstat`

Para conocer el estado de la impresoras usaremos

```
$ lpstat -v
```

²⁵Para conseguir esto mismo con un grupo escribiremos `@grupo`

```
device for epson: smb://MURGI/NOVO/epson
device for epsonfed: ipp://172.26.0.2:631/printers/epson
device for lp0: parallel:/dev/lp0
```

en esta salida de ejemplo podemos ver que en la máquina donde se ha ejecutado el comando se dispone de:

- Una impresora en red que trabaja sobre una máquina Windows
- Una impresora en red sobre una máquina Linux usando CUPS
- Una impresora local conectada al puerto paralelo

5.6. ➔ Para Practicar

Veamos un par de ejemplos sobre las posibilidades que nos brinda CUPS. En ambos sólo hemos puesto aquellas directivas necesarias para conseguir que todo funcione, el resto del fichero ha quedado igual en el fichero de configuración.



Dos notas que nos pueden facilitar la realización de las prácticas:

- Si cambiamos el fichero de configuración de CUPS y deseamos que se active, antes hay que reiniciar el servicio.
 - Para comprobar los cambios al trabajar con autenticación de usuarios hemos de reiniciar el navegador web.
1. Con este ejemplo de configuración sólo permitimos imprimir en nuestra impresora local (lp) al usuario “matematicas”. Lo hacemos en dos pasos:
 - a) Modificamos el fichero de configuración de CUPS para que permita autenticación básica y que sea accesible desde la red local.

```
#####
#####_Browsing_Options
#####

5 #Descomentamos_para_que_se_anuncie_a_la_red_local
  BrowseAddress_@LOCAL

<Location_/>
  ----->Order_Deny, Allow
10 ----->Deny_From_All
  ----->Allow_From_127.0.0.1
  ----->#Permitimos_el_acceso_a_la_red_local
  ----->Allow_From_@LOCAL
</Location>

15 <Location_/jobs>
  #
  #_You_may_wish_to_limit_access_to_job_operations, _either_with_Allow
  #_and_Deny_lines, _or_by_requiring_a_username_and_password.
20 #
  ----->AuthType_Basic
  ----->AuthClass_User
</Location>

25 #añadimos_la_sección
```

```

<Location_/printers/Samsung>
  →#Usamos_autenticación_básica
  →AuthType_Basic
  →#Permitimos_el_acceso_a_todos_los_usuarios_del_sistema
30 →AuthClass_User

  →#Descomentamos_para_restringir_el_acceso_a_la_red_local_
    192.168.1.0
  →#También_es_posible_hacerlo_usando_la_línea_comentada_al_
    final
  →Order_Deny, Allow
35 →Deny_From_All
  →Allow_From_192.168.1.0/255.255.255.0
  →#Allow_From_@LOCAL
</Location>

```

Listado 5.1: Cups Matemáticas

Así cuando deseamos imprimir un trabajo, nos pedirá el nombre de usuario y la contraseña.

- b) Sólo permitimos que imprima el usuario del sistema de nombre matemáticas, para eso hemos de usar la directiva AllowUser en el fichero de configuración de las impresoras (/etc/cups/printers.conf). Para el modelo de ejemplo quedaría

```

1 #_Printer_configuration_file_for_CUPS_v1.1.21_rc1
  #_Written_by_cupsd_on_Thu_Mar_3_22:33:50_2005
<DefaultPrinter_Samsung>
  →Info
  →Location
6  →DeviceURI_parallel:/dev/lp0
  →State_Idle
  →Accepting_Yes
  →JobSheets_none_none
  →QuotaPeriod_0
11 →PageLimit_0
  →KLimit_0
  →#Añadimos_esta_línea_para_permitir_que_solo_pueda_imprimir_
    el_usuario_matematicas
  →AllowUser_matematicas
</Printer>

```

Listado 5.2: Printers Cups Matemáticas

2. Con el segundo ejemplo, veamos algunas de las posibilidades de CUPS usando Internet. Se trata de conseguir que:
- Podamos configurar²⁶ la impresora del instituto desde nuestra casa (80.32.193.107) en la impresora (de nombre imprenta) que está en una máquina del instituto (IP 80.32.184.162).

```

#Si_deseamos_que_sea_accesible_desde_la_IP_especificada_hay_que_
  permitirlo
Listen_127.0.0.1:631
#IP_pública_de_la_máquina_del_centro
Listen_80.32.184.162:631
5
<Location_/admin>
  →AuthType_Basic
  →AuthClass_System

```

²⁶Tal cual está planteado tendremos que conocer la password del root de la máquina del instituto

```
10 → ##_Restringimos_el_acceso
    → Order_Deny , Allow
    → Deny_From_All
    → Allow_From_127.0.0.1
    → #Permite_la_administración_desde_la_IP_de_casa
    → Allow_From_80.32.193.107
15 → #Encryption_Required
    </Location>

<Location_/>
    → Order_Deny , Allow
20 → Deny_From_All
    → Allow_From_127.0.0.1
    → #Permite_acceder_desde_la_IP_de_casa
    → Allow_From_80.32.193.107
    </Location>
```

Listado 5.3: cups-2.1

- Imprimir desde nuestra casa en la impresora que hay conectada a la máquina del instituto. Para poder hacerlo, además de tener que adecuar el fichero de configuración de CUPS para que nos permita imprimir:

1. Tendremos que permitir la posibilidad de imprimir

```
<Location_/printers/imprenta>
    → Order_Deny , Allow
    → Deny_From_All
4 → Allow_From_127.0.0.1
    → AuthType_None
    → #Para_que_nos_permita_imprimir_desde_nuestra_casa_en_esta_
      impresora
    → Allow_From_80.32.193.107
    </Location>
```

Listado 5.4: cups-2.2

2. Por último, añadiremos en nuestro sistema la impresora remota, por ejemplo con
`ipp://80.32.184.162:631/printers/imprenta`
y seleccionar después el filtro adecuado

Capítulo 6

Samba

Samba es la idea de Andrew Tridgell, ... Unos cuantos años después, él lo expandió como su servidor SMB particular y comenzó a distribuirlo como producto por Internet bajo el nombre de servidor SMB. Sin embargo, Andrew no pudo mantener ese nombre -ya pertenecía como nombre de producto de otra compañía-, así que intentó lo siguiente para buscarle un nuevo nombre desde Unix:

```
grep -i 's.*m.*b' /usr/dict/words
```

y la respuesta fue:

```
salmonberry samba sawtimber scramble
```

De ésta manera nació el nombre de Samba.

(*Usando Samba*, ROBERT ECKSTEIN y otros)

6.1. ¿Qué es Samba?

La página principal de Samba es:

<http://www.samba.org>

Mediante Samba y a través del protocolo TCP/IP podemos compartir y utilizar recursos de sistemas de ficheros Unix e impresoras con otros sistemas operativos¹ (discos duros e impresoras) que “hablen” el protocolo SMB (*Session Message Block*, Bloque de mensajes de sesión). Samba es rápido y sencillo de configurar. Linux con Samba puede trabajar como servidor y como cliente. Como servidor ofrece recursos (discos e impresoras) para que los utilicen las máquinas windows. Como cliente utiliza los servicios ofrecidos por las máquinas windows²



Una noticia interesante que apareció hace poco en Internet exponía: “Segun los testeos realizados por IT Week Labs (<http://www.itweek.co.uk/News/1144312>), confirman que la nueva versión de Samba , es casi tres veces más rápida que su contraparte comercial de Microsoft. ” Con Samba, además, disponemos también de servicios de dominios NT³.

¹Sistemas Windows 3.11, 9x, NT, 200x, XP y OS/2

²Os remitimos a la documentación comentada más adelante para esto

³Si proporciona servicios de archivo y de impresión, soporte de *Active Directory*, ..., es mejor como NT que un NT y es gratis, la pregunta es obvia (aunque Microsoft y su propaganda intente demostrar lo contrario).

Samba debe sus “orígenes” a Andrew Tridgell. Necesitaba poder compartir archivos desde el DOS a su servidor UNIX y consiguió el primer programa sobre 1992, que si bien funcionó dejaba bastante que desear. En 1994 y tras retomar el proyecto inicial pero ahora con la idea de interconectar en la misma red Windows y Linux apareció de forma “oficial” Samba. Con Samba disponemos de los servicios:

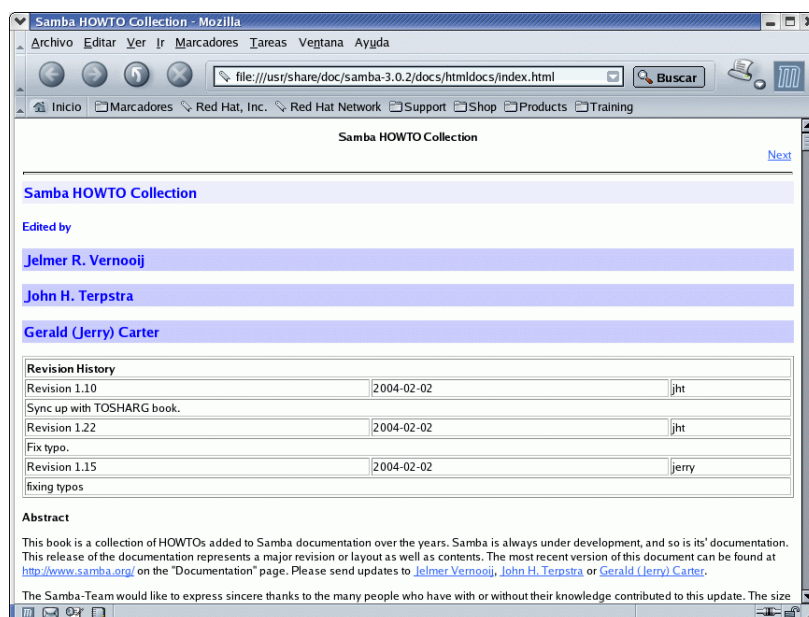
- Servicios de archivos y de impresión.
- Autenticación y autorización.
- Servicios de resolución de nombres.
- Servicios de exploración.

La versión de SAMBA con la que vamos a trabajar es la 3.0. Entre otras mejoras sobre versiones anteriores, destacamos:

- Soporte de *Active Directory*
- Soporte Unicode
- Nuevo sistema de autenticación
- Nuevos comandos net
- Mejor soporte de impresión para Win 2000/XP/2003 incluyendo la publicación de los atributos de impresora en el *Active Directory*

Para afinar mejor la configuración y ampliar sobre el tema os remitimos a

- la extensa documentación que acompaña al programa⁴
`/usr/share/doc/samba-x.x.x`



⁴En Guadalinex hay que instalar el paquete `samba-doc`

- la traducción del libro de O'Reilly *Usando SAMBA* (imprescindible) disponible con la documentación que acompaña al programa y traducido en⁵:
<http://es.tldp.org/Manuales-LuCAS/USANDO-SAMBA/>
<http://www.sobl.org/modules.php?name=Downloads>
- al cómo *Samba-Como*
<http://lucas.hispalinux.es/COMO-INSFLUG/COMOs/Samba-Como/Samba-Como.html>
- un documento traducido por PEDRO P. FÁBREGA llamado *Configuración y Uso de Samba* que podéis conseguir en:
<http://www.arrakis.es/~pfabrega>
en él se explican todos los parámetros del archivo de configuración de SAMBA.
- a una página personal
<http://teleline.terra.es/personal/garzones/garzalin.html>
- además, en Inglés, tenemos los Howtos (están más actualizados que los que hay en castellano):
<http://www.tldp.org/HOWTO/Samba-Authenticated-Gateway-HOWTO.html>
<http://www.tldp.org/HOWTO/SMB-HOWTO.html>
<http://www.tldp.org/HOWTO/Windows-LAN-Server-HOWTO.html>
- y cómo no, a las manpages de los programas.

6.2. Instalación

6.2.1. Fedora

Para garantizarnos que disponemos de la última versión del programa ejecutemos⁶

```
# apt-get install samba samba-common samba-client
```

¿Qué contiene cada paquete?

- Paquete “principal” de la aplicación. Al instalarlo dispondremos de un servidor SMB:
`samba-x.x.x.i386.rpm`
- Si está instalado dispondremos de varios clientes SMB que complementan el sistema de ficheros SMB. Permiten acceder a recursos compartidos SMB (archivos e impresoras)
`samba-client-x.x.x.i386.rpm`
- Proporciona ficheros necesarios para los paquetes anteriores
`samba-common-x.x.x.i386.rpm`

Si instalamos sólo los dos últimos ¿qué disponibilidad tenemos? Muchísima, podemos ser clientes de máquinas windows. Dispondríamos ya (entre otros ficheros) de las utilidades:

⁵Una versión anterior

⁶

- De nuevo comentar que también se puede usar el comando `yum`.
- También podemos optar por usar el comando `rpm` instalando la versión que viene en los CDs de la distribución o bajarlos “a mano” de http://us2.samba.org/samba/ftp/Binary_Packages/Fedora/RPMS/i386/core/3/

```

$rpm -ql samba-client | grep bin/
/sbin/mount.cifs
/sbin/mount.smb
/sbin/mount.smbfs
/usr/bin/findsmb
/usr/bin/nmblookup
/usr/bin/rpcclient
/usr/bin/smbcacls
/usr/bin/smbclient
/usr/bin/smbmnt
/usr/bin/smbmount
/usr/bin/smbprint
/usr/bin/smbpool
/usr/bin/smbtar
/usr/bin/smbtree
/usr/bin/smbumount

$rpm -ql samba-common | grep bin/
/usr/bin/net
/usr/bin/ntlm_auth
/usr/bin/pdbedit
/usr/bin/profiles
/usr/bin/smbcquotas
/usr/bin/smbpasswd
/usr/bin/testparm
/usr/bin/testprns
/usr/bin/wbinfo
/usr/sbin/winbindd
    
```

y del fichero de configuración de Samba
 /etc/samba/smb.conf

6.2.2. Debian

Si bien se instala por defecto, lo mejor es actualizar los paquetes a la última versión⁷:

```
# apt-get install samba samba-common smbclient samba-doc smbfs
```

6.2.3. Programas

Una vez instalados los programas, tenemos a nuestra disposición las utilidades⁸:

smbd demonio SMB, se encarga de los servicios de archivos, de impresión y autenticación y autorización

nmbd demonio de servidor de nombres NetBIOS.

winbindd demonio para resolver nombres con servidores NT.

Además de estos demonios, en los paquetes que componen el programa Samba tenemos entre otros:

findsmb nos muestra información sobre las máquinas SMB.

net utilidad similar a la del Windows o DOS

nmblookup se usa para consultar nombres de NetBIOS y mapearlos a direcciones IP.

smbclient cliente tipo ftp .

smbmount para montar sistemas compartidos SMB en nuestra máquina Linux.

smbumount para desmontar un sistema de archivos SMB ya montado.

⁷

- Los dos últimos paquetes no son “indispensables”, se trata de la documentación de SAMBA y de poder disponer de los comandos `mount` y `umount` para el sistema de ficheros `smb`.
- También podemos bajarlos de
http://us2.samba.org/samba/ftp/Binary_Packages/Debian/samba3/dists/stable/main/binary-i386/
 y usar `dpkg`.
- Si deseamos disponer del demonio `winbindd` habrá que instalarlo:

```
#apt-get install winbind
```

⁸No están todas las utilidades que contienen los paquetes que componen samba.

smbadduser para añadir usuarios.

smbpasswd para cambiar la contraseña de acceso SMB tanto local como remota.

smbprint smbclient reducido que permite imprimir en los recursos de impresión compartidos SMB.

smbstatus utilidad para mostrar las conexiones SMB activas.

smbtar para hacer copia de seguridad de los sistemas de archivos compartidos SMB en una unidad de cinta de nuestra máquina Linux.

smbtree un buscador en modo texto de máquinas que hablan el protocolo SMB

swat utilidad para configurar SAMBA usando un navegador⁹.

testparm revisa/prueba los archivos de configuración de SAMBA.

testprns para revisar el nombre de impresora para usarlo con SAMBA.

6.3. Configuración

Una vez instalados los paquetes estamos casi listos para funcionar, ya que los demonios que requiere se ponen en marcha por defecto al arrancar el sistema operativo¹⁰. Antes de activarlos tendremos que configurar la máquina Linux y la máquina Windows.

Partiremos de una red privada con un grupo de trabajo THALES con la siguiente configuración:

Sistema Operativo	Nombre	IP
Linux	eco	172.26.0.2
Windows 98	bag	172.26.0.11
Windows XP	compa	172.26.0.12

El proceso consiste:

6.3.1. Configuración de las máquinas Windows

Para trabajar con Samba tendremos que tener cargados los protocolos TCP/IP, por tanto, en Red comprobaremos que tenemos instalados¹¹ esos protocolos

⁹Hay que instalarlo

¹⁰Si deseamos activarlos sin reiniciar el sistema escribiremos:

Fedora `#/etc/rc.d/init.d/smb start`

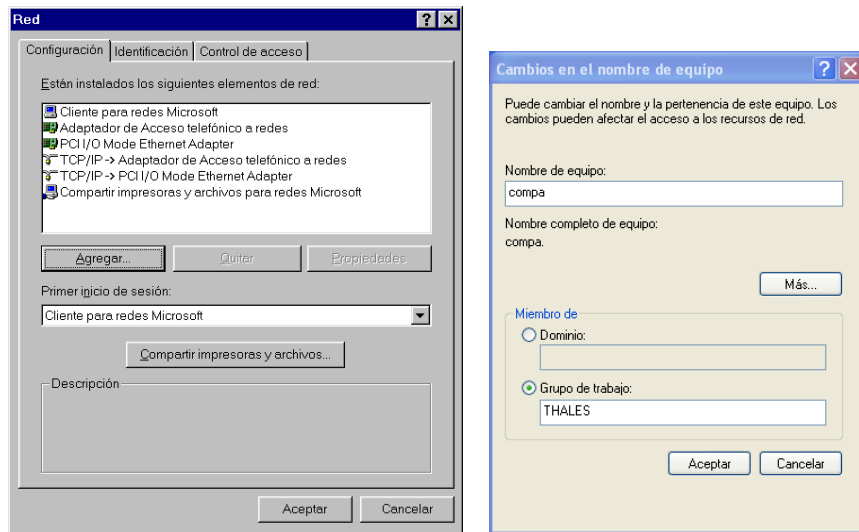
Debian `#/etc/init.d/samba start`

Si optamos por cambiar las opciones de arranque podemos usar las herramientas gráficas:

Fedora `#system-config-services`

Debian `#services-admin`

¹¹Notar que las capturas son un poco viejas, pero hemos preferido mantenerlas porque en Windows 95 no se instalaba por defecto el protocolo TCP/IP. No es así en los Windows posteriores.



y asignaremos una dirección IP a nuestras máquinas (por ejemplo 172.26.0.11), en máscara de Red pondremos 255.255.255.0

Asignaremos un nombre a nuestra máquina (BAG) y pondremos el grupo de trabajo en el caso de que no estuviese ya definido (THALES). Repetiremos este proceso con la máquina COMPA.

Problemas con las contraseñas.

Si todos nuestros equipos con Windows son posteriores a Win95 OSR2 (de esa premisa partimos en la configuración de ejemplo), no tendremos que hacer nada de esto y lo mejor es trabajar con contraseñas encriptadas. Incluso si no estamos seguros lo mejor es dejar este tema aparcado. Sólo en el caso de que tengamos una máquina con un Windows antiguo con la que no conseguimos acceder al servidor SMB y sin embargo sí podemos usar el resto de servicios es cuando deberíamos plantearnos si el problema puede estar aquí. Para saber cómo conseguirlo véase en

Fedora /usr/share/doc/samba-x.x.x/docs/Registry


Debian /usr/share/doc/samba-doc/registry

y la página ??.

6.3.2. Configuración de la máquina Linux

Análisis del archivo de configuración de Samba

El archivo de configuración de Samba es `/etc/samba/smb.conf`¹². Mediante este archivo podemos controlar la gran cantidad de opciones disponibles del programa, aunque tratar a fondo todas y cada una de ellas está más allá del objetivo de esta entrega.

 Nos centraremos para su estudio en el fichero de Fedora¹³, trasladar lo aquí expuesto a máquinas con Debian no supone ninguna dificultad.

Se divide en varias secciones, cada una de ellas determinada por un nombre entre corchetes: `[impresoras]`, `[global]`, etc. En cada sección encontramos una serie de parámetros compuestos por pares de clave/valor. Los comentarios comienzan con punto y coma (;) o mediante una almohadilla

¹² Antes de tocar este archivo, como viene siendo habitual, deberíamos hacer una copia por si acaso.

¹³ Es más completo y se presta más a su estudio que el que se instala por defecto con Guadalinex

(#).¹⁴ Ejemplifiquemos esto con una parte de la sección `[global]`:

```
[global]
# La almohadilla indica que estamos en un comentario
  remote announce = 172.26.0.255 172.26.2.44
# El siguiente parámetro está marcado como comentario y
# el segundo no, es decir, está activo
; local master = no
  os level = 33
```

Si modificamos el fichero de configuración tendremos que reiniciar el servidor¹⁵ con:

Fedora `#/etc/rc.d/init.d/smb restart`

Debian `#/etc/init.d/samba restart`

Para comenzar a trabajar y conocer las posibilidades que nos ofrece el programa sólo vamos a modificar las opciones más usuales de este fichero.

Sección `[global]` En esta sección configuraremos parámetros para todo el servidor SAMBA así como algunos valores predeterminados a las otras secciones. Veamos algunas de las opciones más usuales¹⁶.

Comencemos por ajustar el grupo de trabajo¹⁷ en el que nos encontramos. Por ejemplo, si nuestro grupo de trabajo es THALES, escribiremos:

```
workgroup=THALES
```

Con el parámetro

```
server string = Samba Server
```

indicamos el nombre que identificará al servidor cuando lo consultan los clientes SAMBA. Con la directiva

```
netbios name = bag
```

establecemos el nombre NetBIOS de la máquina.



- Si no se proporciona el nombre del grupo de trabajo tomará como grupo predeterminado `WORKGROUP`.
- Si no establecemos el nombre NetBIOS de la máquina tomará el que se obtenga de ejecutar el comando `hostname`

La línea que sigue aparece marcada como comentario. Si la activamos con el valor

```
hosts allow = 172.26.0. 127.
```

conseguimos que el acceso al servidor de Samba esté restringido a los hosts o redes especificados.

En este caso el acceso está limitado a la red 172.26.0.0/24 y al host local. Si deseamos limitar el acceso a una red de clase B escribiríamos 172.26. y si es de clase A, 172.

Con `hosts deny` podemos negar el acceso a determinadas máquinas o subredes (en caso de duda “gana” `hosts allow`). Por ejemplo, con

¹⁴Normalmente aparecerán marcados los parámetros como comentarios utilizando el punto y coma, y se deja la almohadilla para comentarios normales.

¹⁵Es deseable actualizar el fichero `/etc/hosts` con las direcciones IP y el nombre de cada máquina a la que vamos a acceder. De esta forma con sólo escribir el nombre del equipo al que queremos acceder, el servidor buscará en ese fichero el nombre del equipo y dirección IP correspondiente.

¹⁶Para conocer todas las opciones véase el libro *Usando Samba* ya comentado.

¹⁷Deberá estar limitado como máximo a nueve caracteres, sin espacios y todos en mayúsculas.

```
hosts allow = 172.26 EXCEPT 172.26.0
permitimos el acceso a la red de clase B 172.26. pero denegamos el acceso a la subred de clase
C 172.26.0
```

Por defecto el archivo `smb.conf` permite que estén disponibles todas las impresoras definidas en `/etc/printcap` como recursos compartidos.

```
# si se quiere cargar automáticamente la lista de impresoras en lugar
# de configurarlas por separado, será necesario esto
printcap name = /etc/printcap
load printers = yes

# No será necesario especificar el tipo de sistema de impresión a menos
# que no sea estándar. Los sistemas con soporte en la actualidad incluyen:
# BSD, sysv, pip, lprng, aix, hpux, qnx
;printing = cups

# Así le decimos a cups que los datos han sido tratados
cups options = raw
```



Para poder usar CUPS como servidor de impresión hemos de introducir aquí algunos cambios. Para que las distintas modificaciones no queden “dispersas” por la entrega hemos optado por concentrarlos todos en 6.4.2 en la página 117

Si se quiere especificar una cuenta de usuario *guest* (invitado) para el acceso anónimo a los servicios en un servidor Samba descomentaremos esta línea. Esto no es imprescindible, por omisión se permite el acceso de un usuario *guest* mediante la cuenta *nobody*¹⁸.

```
# Quitar la marca de comentario aquí si se desea una cuenta de usuario guest,
# se debe añadir esto a /etc/passwd, de no ser así, se utiliza el usuario
# nobody
; guest account = pcguest
```

La líneas que siguen nos indican el lugar en dónde se almacenarán los logs del sistema SMB así como el tamaño máximo (en Kb) que pueden tener¹⁹:

```
log file = /var/log/samba/%m.log
max log size = 50
Si deseamos que se use un sólo fichero escribiremos log file = /var/log/samba/smbd.log
```

Con la entrada que sigue nos garantizamos que sólo los usuarios de la máquina Linux tienen acceso vía SMB

```
# Modo de seguridad. La mayoría querrá nivel de seguridad de usuario.
# Ver security_level.txt para más detalles.
security=user
# Sólo se debe descomentar esta directiva si security = server
; password server = <NT-Server-Name>
```

Si en vez de `user` escribimos `share` podrán acceder al servidor usuarios no registrados en el sistema. Es mejor si se desea un acceso menos restringido al servicio de impresión o crear una zona pública limitada por la seguridad del recurso.

Con las líneas que siguen (están comentadas por defecto y no deberían descomentarse salvo que tengamos problemas) establecemos el número de caracteres (no distinguen entre mayúsculas y minúsculas) que comprobaremos de los nombres de usuario así como de su contraseña.

```
# El nivel de password permite que concuerden _n_ caracteres de la contraseña
# para todas las combinaciones de mayúsculas y minúsculas.
```

¹⁸Sobre la forma de dar de alta un usuario véase la orden `smbpassword` 6.4.2 en la página 116

¹⁹Con un valor de 0 no ponemos ningún límite en cuanto al tamaño de los ficheros.



```
; password level = 8
; username level = 8
```

Aunque las líneas que siguen estén comentadas, no importa ya que SAMBA trabaja con contraseñas encriptadas por defecto: es la forma de pasar las contraseñas de los Windows 95 OSR2 y siguientes.

```
# Si se quiere usar encriptación de la password, consúltese, por favor
# ENCRYPTION.txt, Win95.txt y WinNT.txt en la documentación de Samba.
# NO activar esta opción a menos que se hayan consultado dichos documentos
;encrypt passwords = yes
;smb passwd file = /etc/smbpasswd
```

Permite a Samba actualizar el fichero de contraseñas de Linux cuando un usuario cambia su contraseña encriptada (fichero `/etc/samba/smbpasswd`).

```
# Lo que sigue se necesita para permitir cambiar las contraseñas desde Windows
# y que se actualicen las de Linux.
# NOTE: Úselo con 'encrypt passwords' y 'smb passwd file'.
# NOTE2: No necesita esto para permitir a las estaciones de trabajo cambiar solamente
# las contraseñas encriptadas SMB. Permite que las contraseñas Unix
# se mantengan sincronizadas con las password SMB.
;unix password sync = Yes
;passwd program = /usr/bin/passwd%u
;passwd chat = *New*password* %n\n *Retye*new*password* %n\n ...
```

Permite especificar un fichero que contiene un mapa de nombres de usuarios de los clientes del servidor²⁰

```
# Los usuarios de Unix pueden mapear a distintos nombres de usuario SMB
; username map = /etc/samba/smbusers
```

Si la descomentamos, copiamos el fichero objetivo en el actual fichero de configuración a partir de este punto. Si el fichero referenciado no existe se ignora esta directiva. La variable `%m` almacena el nombre NetBios del cliente

```
# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
; include = /etc/samba/smb.conf.%m
```

Con `socket options`²¹ podemos poner las opciones de *socket* para usarlas cuando hable con el cliente. Se usa para ajustar Samba a bajo nivel y conseguir mejorar las prestaciones de la red local: “Puesto que cada red es diferente (cableado, interruptores, ruido, etc), no hay una fórmula mágica que funcione para todo el mundo. Como consecuencia, si quiere un ajuste fino del rendimiento de Samba para su red específica, tendrá que hacer algunos experimentos. Para los puristas (y un gran remedio contra el insomnio), pueden documentarse sobre sockets en la página `man socket`” (*The Unofficial Samba HOWTO*²²).

```
# Mucha gente encontrará que esta opción mejora el rendimiento del servidor
# Para más detalles, vea speed.txt y las páginas del manual
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

²⁰Cada línea de este fichero tiene de formato:

usuario Unix= uno o más nombres de usuario cliente separados por espacios.

Por ejemplo:

```
thales = PepeLinux
```

```
mileto = Griego
```

Véase la sección 6.2.3.1 del Libro *Usando Samba*

²¹Para entender mejor esto, una buena frase de LINUS TORVALDS:

He intentado obtener alguna documentación fuera de Digital sobre esto, pero hasta donde puedo decir incluso ellos no la tienen;-)

²²Para conocer más sobre la directiva que sigue se puede consultar este documento.



Con `interfaces` configuramos las direcciones de red a las cuales reconoce el servidor Samba. Si nuestra máquina tiene varias tarjetas de red tendremos que ponerlas aquí, si no lo hacemos sólo reconocerá el primer interfaz de red al arrancar y trabajará sólo con esa subred.

```
# Configura Samba para multiples interfaces de red
# Si se tienen varias tarjetas de red hay que listarlas aquí
#Véase las páginas man para más detalles
; interfaces = 192.168.12.2/24 192.168.13.2/24
```

`browse sync` lista los servidores Samba para sincronizar sus listas de visualización con los demás visualizadores maestros en otras subredes. Si se descomenta la línea de ejemplo, el servidor Samba contactará con la máquina de IP 192.168.3.25 para sincronizar listas de visualización. Con 192.168.5.255 forzamos a Samba a hacer un *broadcast* de peticiones para determinar las direcciones IP de los visualizadores maestros locales en la red 192.168.5.0 para después realizar la sincronización.

Con `remote announce` hacemos que Samba proporcione listas de visualización al visualizador maestro local de una red externa. El comentario anterior se puede extender a este caso para entender la diferencia de escribir 192.168.1.255 y 192.168.2.44 en la línea de ejemplo comentada.

```
# Para sincronizar las listas de visualización
# y los anuncios de peticiones hacia o desde los visualizadores maestros:
# desde una máquina concreta o hacia una subred completa
; remote browse sync = 192.168.3.25 192.168.5.255
# El host mandará anuncios a todas las subredes locales especificadas aquí
; remote announce = 192.168.1.255 192.168.2.44

# Browser Control Options:
# ponga local master en no si no quiere que samba sea
# un visualizador maestro en su red. Si no, se aplicarán las reglas normales de elección
; local master = no
# OS Level determina la prioridad de este servidor en las elecciones
# del visualizador maestro. Por defecto debería tener un valor razonable
; os level = 33
# Domain Master especifica que Samba sea el Visualizar Maestro de Dominio. Esto
# le permite a Samba comparar listas de visualización entre subredes
# no lo use si ya tiene un controlador de dominio de Windows NT haciendo esto
; domain master = yes
# Preferred Master hace que Samba establezca el bit de maestro preferido en la elección
de visualizador maestro
# y le da una posibilidad ligeramente mayor de ganar en la elección
; preferred master = yes
```

Podemos conseguir que Linux autentifique a los clientes Windows en la red. Por defecto, estas líneas están comentadas:

```
# Actívese esto si se desea que Samba sea un servidor de inicio de sesión de
# dominio para estaciones de trabajo de Windows95.
; domain logons = yes

# Si se activan los inicios de sesión de dominio puede ser necesario un
# script de inicio de sesión por máquina o por usuario ejecútase un archivo
# específico de procesamiento por lotes de inicio de sesión por máquina
; logon script = %m.bat

# ejecútase un archivo específico (de procesamiento por lotes de inicio de
# sesión por cada usuario
; logon script = %U.bat
```




```
# Dónde almacenar perfiles itinerantes (sólo para Win95 and WinNT)23
#%L se sustituye por el nombre del servidor netbios,%U es el nombre de usuario
# debe descomentar [Profiles] que aparece más abajo
; logon path = \\%L\Profiles\%U
```

La directiva `name resolve order` especifica el orden de los servicios que usará SAMBA para resolver nombres. Si descomentamos la línea no sigue el método por defecto²⁴: primero opta por un servidor Wims, después el fichero `lmhosts` y por último usa *broadcasting* para determinar la dirección de un nombre NetBIOS.

```
; name resolve order = wins lmhosts bcast
```

```
# Windows Internet Name Serving Support Section:
# WINS Support - Le dice al componente NMBD de Samba que habilite su servidor WINS
; wins support = yes
# WINS Server - Le dice a su componente NMBD que sea un cliente WIMS
# Nota: Samba puede ser un servidor o un cliente WIMS, pero no ambos
; wins server = w.x.y.z
# WINS Proxy - Permite a SAMBA responder a las peticiones de resolución de nombres
# en nombre de un cliente, para que funcione debe de haber al menos
#un servidor WIMS en la red. Por defecto está en No
; wins proxy = yes
# DNS Proxy - si Samba debe intentar o no resolver nombres NetBIOS
# vía DNS nslookups. Por defecto en las versiones 1.9.17 es sí,
# y ha cambiado en la versión 1.9.18 a no.
dns proxy = no
#
```

Sección [homes] Con esta sección podemos controlar de qué forma accederán los clientes al directorio principal de usuario en el servidor Linux. La configuración aquí establecida permite que los parámetros sean válidos para todos los usuarios y no hay que especificar una configuración para cada usuario por separado.

```
comment = Home Directories
browseable = no
writable = yes
valid users = %S
create mode = 0664
directory mode = 0775
```

Analizamos cada una de las líneas anteriores²⁵:

comment cadena de identificación que se muestra a los clientes que examinan el servidor Samba.

browseable al establecerlo a `no` conseguimos que el explorador de Windows no nos muestre los `$HOME` de otros usuarios del sistema Linux. Si no somos ningún usuario registrado del sistema no veríamos ningún `$HOME`. Sin embargo, si nos conectamos como un usuario del sistema Linux aparecerá una carpeta de nombre ese usuario a la que podremos acceder con nuestra contraseña. Si lo establecemos a `yes` aparecerán todas las carpetas de los `$HOME` de usuario, aunque no podríamos acceder a ellas salvo que la validación sea la correcta (nombre de usuario y contraseña)

²³Véase el libro *Usando Samba*, sección 6.7.1

²⁴`lmhosts`, después los métodos de resolución Linux estándar (`/etc/hosts`, `DNS`, y `NIS`), a continuación interroga a un servidor WINS, y por último usa *broadcasting*

²⁵Las tres últimas no aparecen en el fichero de configuración por defecto.

writable (sinónimo de **writeable**) permite que un usuario pueda crear y modificar archivos en su directorio `$HOME` cuando inicia una conexión SAMBA. Podemos conseguir esto mismo sustituyendo esta línea por

```
read only = no
write ok = yes
```

valid users lista de usuarios que pueden conectarse a un recurso (en este caso, la variable `%S` contiene el nombre actual del recurso).

create mode determinamos que los permisos de archivo para todos los archivos creados en el directorio compartido sean 0664.

directory mode los permisos para todos los directorios creados en el recurso compartido serán 0775.

Secciones `[netlogon]` y `[Profiles]`

Por defecto están comentadas. Samba soporta la ejecución de script de entrada que permiten configurar las opciones de red cuando los usuarios se conectan. También permite (`[Profiles]`) que cada usuario almacene su perfil, accesible vía red.

Para poder usarlas tenemos que configurar las directivas adecuadas de la sección `[Global]`. Para conocer mejor este tema se puede consultar el capítulo 6 (Usuarios, seguridad y dominios), sección Scripts de Entrada, del libro *Usando Samba*



La recomendación de *The Unofficial Samba HOWTO* sobre la posibilidad de activar la sección `[Profiles]` es: *Si desea habilitar roaming profiles para Windows 2000/XP haga los cambios siguientes en su archivo `smb.conf`. Nota: ¡es absolutamente desaconsejable a menos que sepa lo que está haciendo (prevenga quebraderos de cabeza)!.* Así que para no tener que tomar una pastilla mejor lo dejamos como está.

Sección `[printers]`

Con SAMBA podemos configurar de dos formas distintas la forma en que nuestras impresoras están disponibles para la red:

1. Creando una sección específica para compartir en `/etc/printcap` para cada impresora que se quiera compartir. No es el método que vamos a usar.
2. Usar la sección especial `[printers]` para compartir todas las impresoras definidas en el archivo `/etc/printcap`.

Con esta sección definimos cómo se controlan los servicios de impresión en el caso de que no haya entradas específicas en el archivo de configuración de SAMBA: si disponemos de un servicio, SAMBA buscará en primer lugar un servicio con ese nombre, si no hay ninguno con ese nombre se usa esta sección para permitir al usuario conectarse con cualquier impresora definida en `/etc/printcap`.

Con la sección `[printers]` que se lista a continuación damos a compartir todas las impresoras del sistema y además permitimos imprimir a cualquiera, excepto a la cuenta *guest*.

```
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
# Set public = yes para permitir al usuario guest imprimir
# es equivalente a guest ok = yes
guest ok = no
writable = no
```

```
printable = yes
```

Los parámetros `comment` y `browseable` ya se han visto en la sección `[homes]`.

Con el parámetro `path` determinamos en qué directorio temporal²⁶ se copian los archivos antes de imprimirlos, en este caso `/var/spool/samba`.

Con el parámetro `printable` en `yes` permitimos la escritura en el fichero de `spool` de impresión (si los permisos lo permiten). Con el parámetro `writable` a `no` restringimos a que sólo se permita la escritura de trabajos de impresión en el directorio de `spool` definido con el parámetro `path`. Es decir, definimos el servicio como no “escribible” pero sí como “imprimible”.



Para poder usar CUPS hay que cambiar esta sección, véase 6.4.2 en la página 117

Secciones personalizadas Usando las secciones personalizadas podemos compartir impresoras o directorios de una manera no genérica. La idea que subyace en estas secciones consiste en poder compartir directorios para grupos de usuarios o permitir que determinados directorios sean de acceso público. A su vez, si tenemos varias impresoras conectadas a nuestra máquina y no queremos darlas todas a compartir, podemos usar esta sección para dar a compartir sólo una impresora en concreto.

Con el ejemplo que se lista a continuación permitimos acceso al servicio `trabajos` a los usuarios del grupo llamado `clase`:

```
[trabajos]
comment = Directorio compartido de la clase de informática
path = /home/trabajos
browseable = yes
writable = yes
printable = no
valid users = @clase
```

Las líneas de este ejemplo significan:

- damos a compartir el directorio `/home/trabajos`
- La identificación de este servicio es: `Directorio compartido de la clase de informática`
- Al estar `browseable` en `yes`, se mostrará la carpeta del recurso compartido en el explorador de Windows siempre que accedamos al servidor SMB.
- Permitimos que se pueda escribir en él.
- Con `printable = no` indicamos que no es un servicio de impresión.
- Con el parámetro `valid users = @clase` restringimos el acceso a este directorio a miembros del grupo `clase`.

Veamos cómo compartir una impresora dedicada en la que sólo puede imprimir el usuario `CURSO-LINUX`:

```
[Impresora]
comment = Impresora a compartir
# con print ok = yes se consigue el mismo efecto
```

²⁶Los permisos de este directorio están configurados para permitir la lectura y la escritura, lo mismo que el directorio `/tmp`.

```
drwxrwxrwt 2 root root 4096 jun 9 20:05 /var/spool/samba
```

```
# que con la línea que sigue
printable= yes
printer = lp
path = /var/tmp
public = no
writable = no
valid users = cursolinux
```

Para finalizar, ¿por qué no dar a compartir nuestra unidad de CD a los usuarios **THALES** y **MILETO** de nuestra máquina:

```
[Cdrom]
#En Guadalinux será /cdrom
path =/mnt/cdrom
writable = no
printable = no
valid users = thales, mileto
public = no
```

Cuando terminemos de configurar nuestro sistema podemos usar:

```
$ testparm
```

y comprobar que todo está perfectamente.

En Guadalinux

El fichero que se instala en un sistema actualizado de Guadalinux 2004 es de la forma

```
[ global ]
2  netbios_name = G2004_1108897514
   server_string = Guadalinux_2004
   workgroup = GUADALINUX
   wins_support = no
   encrypt_passwords = true
7
#_Do_something_sensible_when_Samba_crashes: _mail_the_admin_a_backtrace
#_panic_action = /usr/share/samba/panic-action_%d

[ compartido ]
12 path = /home/compartido
   comment = Directorio_compartido_en_Guadalinux_2004
   writeable = no
   guest_ok = yes
   guest_only = yes
17 browseable = yes
```

Listado 6.1: /etc/samba/smb.conf

No debería presentar problema comprender su significado y deberíamos adecuarlo a nuestros intereses.

Respecto a lo estudiado sólo hay que comentar varias cuestiones:

- Si instalamos varios Guadalinux y no actualizamos SAMBA tendremos que ajustar el nombre NetBIOS ya que por defecto pone en todos ellos el mismo. Si actualizamos SAMBA este problema desaparece.
- En Guadalinux han creado un directorio de uso compartido y visible por todos (**browseable = yes**) que no está exento de riesgos de seguridad ya que:

- no es necesario nombre de usuario ni contraseña para acceder al recurso (`guest ok = yes`), y
- aunque la conexión sea autenticada, en este recurso se accede como anónimo (`guest only = yes`)
- al final de esta sección (6.4.2 en la página 119) hemos añadido un fichero de configuración básico de ejemplo que recoge los aspectos más importantes tratados en este tema, permite que los usuarios accedan a sus `$HOME` desde máquinas Windows y que puedan imprimir desde los windows sobre la máquina Linux.

6.3.3. Swat

Vamos a comentar una herramienta que permite configurar Samba usando el navegador Web, se trata de SWAT (*Samba Web Administration Tool*). SWAT se incluye como parte del paquete estándar de Samba. La idea de este programa (hay más con esta misma filosofía) consiste en facilitar la configuración del servidor.



Tiene una “pega”, y es que cuando se usa escribe un archivo de configuración sin comentarios.

Instalación

Fedora # `apt-get install samba-swat`

Antes de poder usar esta utilidad tenemos que configurar nuestro sistema para que permita acceder a ella, para eso hemos de decirle a `xinetd` que nos permita trabajar con él. En el fichero `/etc/xinetd.d/swat` sustituiremos la línea

```
disable = yes
```

por

```
disable = no
```

Después, tendremos que reiniciar `xinetd` para activar los cambios:

```
# /etc/rc.d/init.d/xinetd restart
```

Debian: # `apt-get install swat`

Una vez instalado, hemos de descomentar la línea adecuada (una que comienza por `swat`) del fichero `/etc/inetd.conf` y releer la nueva configuración

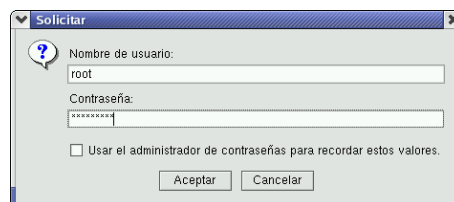
```
#/etc/init.d/inet reload
```

Uso

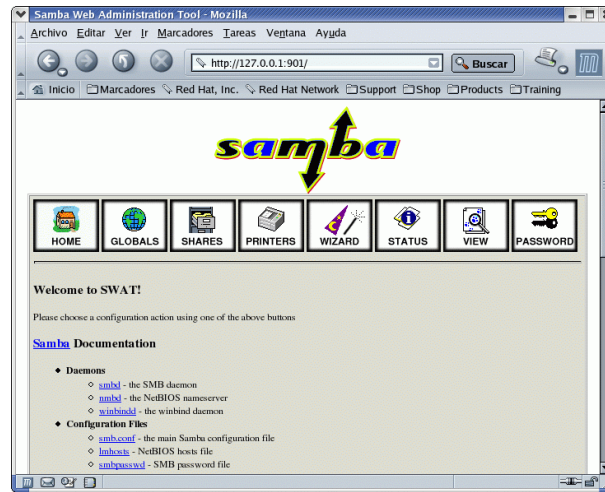
Para usar el programa SWAT tendremos que iniciar un navegador y nos conectaremos a la URL

```
http://localhost:901/
```

Tras la ventana que nos pide los datos del root



accederemos a la página principal de la aplicación:



Los iconos de la parte superior de la pantalla nos permiten acceder a diferentes páginas de SWAT:



página principal, en ella entre otras cosas tenemos enlaces a la documentación del paquete Samba. Entre otros el libro de O'Reilly (pero en Inglés) comentado en esta misma sección.



desde aquí podemos manipular la sección globals del archivo `/etc/samba/smb.conf`. Podemos modificar los valores de los distintos parámetros, obtener ayuda sobre ellos y/o mantener el valor predeterminado. Para grabar los cambios pulsaremos sobre [**Commit Changes**].



usando esta página podremos añadir, modificar o borrar recursos compartidos. Por defecto la pantalla inicial muestra sólo los parámetros de uso más frecuente del archivo `smb.conf`. Si pulsamos sobre [**Advanced View**] tendremos la posibilidad de configurar alguno de los parámetros menos usados.



para configurar las impresoras.



desde esta página podemos limpiar el archivo `smb.conf` de todos los comentarios y valores por defecto.



desde aquí podemos comprobar el estado de Samba. Además de ver cómo están las “cosas” podemos arrancar y parar los demonios de Samba y ver las conexiones activas de nuestro servidor Samba.



para examinar el contenido del archivo `smb.conf`. Si deseamos ver todas las variables disponibles y sus valores hay que pulsar en **Full View**.



con este enlace aparecerá la pantalla mediante la cual podremos cambiar cuentas de usuarios locales y cuentas del controlador del dominio primario.

6.4. A “bailar” la Samba

6.4.1. Acceder desde una máquina Linux a una Windows

Como es evidente sólo podremos acceder a aquellos recursos autorizados en la máquina Windows.

Si optamos sólo por usar esta posibilidad no necesitamos tener instalado el paquete “principal” de la aplicación²⁷. Trabajando de esta forma Samba no comparte ningún recurso con otro sistema, se limita a acceder a los recursos compartidos en los servidores de recursos la red. Si trabajamos sólo como cliente no tenemos que tener activos los demonios `smbd` o `nmbd` aunque sí debemos ajustar a nuestro grupo de trabajo el archivo `smb.conf`, la única línea necesaria en ese fichero sería:

```
workgroup=THALES
```

Acceder en modo gráfico es fácil, sólo hemos de pulsar sobre el escritorio en **Equipo→Red**²⁸



y acceder a la **Red de Windows**. Pero, además de la posibilidad de acceder en modo gráfico, veamos algunas de las posibilidades de trabajo en modo texto:

smbclient

Una de las utilidades más interesantes de las que acompañan a Samba es `smbclient`. Con él podemos acceder desde Linux a los recursos compartidos en máquinas Windows con métodos que incluirían FTP, NFS y los “comandos r”, como `rep`.

`smbclient` nos permite disponer de un interfaz similar a un FTP, por tanto, es una utilidad cuyo objetivo es accesos temporales a un recurso compartido.²⁹

Veamos algunos ejemplos sobre su uso. Partiremos de la base de que estamos conectados en red, que nuestra máquina Linux se denomina ECO y que podemos acceder a varias máquinas con Windos (máquinas BAG y COMPA).

Antes de nada y como no sabemos qué máquinas están a nuestra disposición ejecutemos:

```
# nmblookup -d 2 thales
added interface ip=172.26.0.2 bcast=172.26.0.255 nmask=255.255.255.0
querying thales on 172.26.0.255
Got a positive name query response from 172.26.0.2 ( 172.26.0.2 )
Got a positive name query response from 172.26.0.11 ( 172.26.0.11 )
Got a positive name query response from 172.26.0.12 ( 172.26.0.12 )
172.26.0.2 thales<00>
172.26.0.11 thales<00>
172.26.0.12 thales<00>
```

O bien

```
$ findsmb30
```

IP ADDR	NETBIOS NAME	WORKGROUP/OS/VERSION
172.26.0.2	ECO	+ [THALES] [Unix] [Samba 3.0.11]
172.26.0.11	BAG	[THALES]
117.26.0.12	COMPA	[THALES]

Vemos qué máquinas del grupo de trabajo especificado responden positivamente. Pero el mejor se deja para el final, usamos ahora:

²⁷Sólo hay que instalar `samba-client` y `samba-common`.

²⁸Equivale a abrir **Nautilus** y escribir `network://`

²⁹Si lo que deseamos es mantener una conexión “permanente” es mejor usar `smbmount`.

³⁰Sólo para Fedora. Si no tenemos instalados y activos los demonios del paquete samba puede que nos dé algunos errores.

➔ **Para practicar:** comprobar que el comando que mejor nos informa de los equipos y recursos compartidos es:

```
$smbtree
```

■

Vemos que la máquina de IP 172.26.0.11 está encendida y, al ejecutar el comando anterior sabemos ya qué recursos tiene compartidos. Otra forma de listar los recursos compartidos es con el comando `smbclient` pasándole el parámetro `-L`. Por ejemplo, si nuestra máquina Windows se llama `bag` una posible salida es:

```
$ smbclient -L bag
added interface ip=172.26.0.2 bcast=172.26.0.255 nmask=255.255.255.0
Password:
  Sharename      Type      Comment
  -----      -
  ERASE          Disk
  ADMIN$         Disk
  PRINTER$       Disk
  EPSON          Printer
  D              Disk
  C              Disk
  IPC$           IPC       Comunicación remota entre procesos

  Server         Comment
  -----
  Workgroup      Master
```

nos muestra una lista con los recursos disponibles, las máquinas que comparten recursos y los grupos de trabajo. Comentar que la máquina Linux verá a la máquina Windows aunque el nombre del grupo de trabajo no sea el mismo.

Si usamos³¹:

```
$ smbclient -L fedora -U cursolinux
added interface ip=172.26.0.2 bcast=172.26.0.255 nmask=255.255.255.0

Anonymous login successful
Domain=[THALES] OS=[Unix] Server=[Samba 3.0.11]

  Sharename      Type      Comment
  -----      -
  IPC$           IPC       IPC Service (Samba Server)

  Server         Comment
  -----
```

³¹Mejor si le decimos que el nombre de usuario y las password están vacías:

```
$ smbclient -U%
```


ECO	Samba Server
Workgroup	Master
-----	-----
THALES	SECRE

veremos la lista de los recursos compartidos disponibles en el servidor SMB FEDORA para el usuario *cursorlinux*.

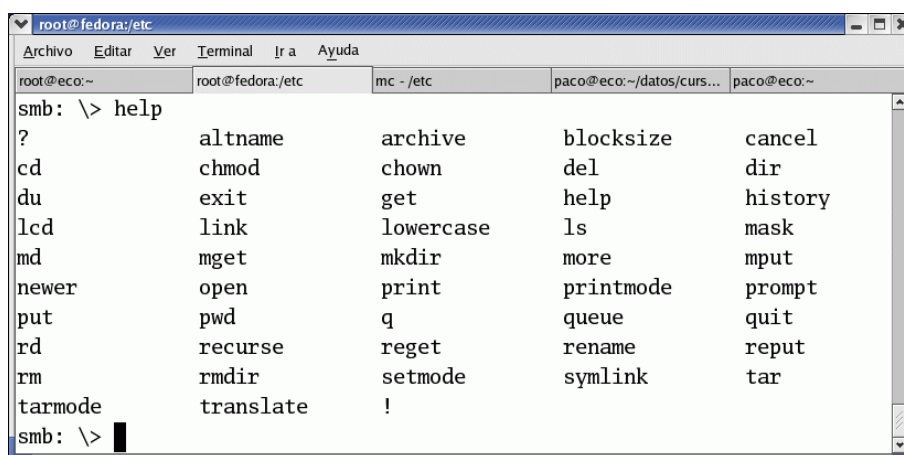
Con salida de `smbclient -L bag` hemos podido comprobar qué recursos compartidos tenemos a nuestra disposición en esa máquina, accedamos a alguno de ellos, para eso escribimos³²:

```
$ smbclient //bag/c
```

si deseamos acceder al disco C. Tras introducir la contraseña veremos:

```
smb: \>
```

Aquí podemos introducir comandos, para saber cuáles tenemos disponibles podemos usar la orden `help`³³



Para ampliar sobre el significado de cada uno de ellos sólo debemos usar de nuevo `help` seguido del comando del que queremos ayuda:

```
smb: \>help get
```

```
HELP get:
```

```
<remote name>[local name] get a file
```

Comentemos algunos de los más usuales:

? [comando] muestra ayuda sobre ese comando o lista de comandos

help [comando] igual que ?

! [comando de shell] ejecuta un comando de la shell

cd [directorio] cambia el directorio remoto

lcd [directorio] cambia el directorio local

md [directorio] crea un directorio

mkdir [directorio] igual que md

rd [directorio] borra el directorio

rmdir [directorio] igual que rm

del [archivos] borra el archivo

dir [archivos] lista los archivos

ls [archivos] igual que dir

get [rarchivo] [larchivo] copia el archivo remoto (rarchivo) en el archivo de nombre larchivo

³²Si el nombre del recurso compartido contiene espacios (por ejemplo “Mis Documentos”) tendremos que escribir:
`smbclient //bag/Mis\ Documentos`

³³Son muy parecidos a los del ftp.

mget [archivos] copia los archivos que que coincidan con el nombre especificado (normalmente se usan comodines)	nombre especificado (normalmente se usan comodines)
newer [archivo] sólo tomará los archivos posteriores al especificado.	printmode [modo] determina el modo de impresión (text o graphics)
put [larchivo] [rarchivo] copia desde la máquina local el archivo larchivo en la máquina remota con el nombre rarchive	print [archivo] imprime el archivo especificado en la máquina remota
mput [archivos] copia en el servidor los archivos de la máquina local que coinciden con el	queue muestra la cola de impresión.
	exit sale del programa
	quit igual que exit

Por ejemplo, si tras hacer un listado de la máquina remota:

```
smb: \>ls
```

comprobamos que hay un archivo de nombre `curso.txt`, podemos bajarlo a la máquina local usando:

```
smb: \>get curso.txt
```

para salir

```
smb: \>exit
```

smbmount y smbmount

Si deseamos tener la posibilidad de montar un sistema de archivos compartido en el árbol del sistema de archivos de Linux tenemos que usar `smbmount`³⁴

Para montar algún recurso de la máquina Windows usaremos el comando:

```
# smbmount //máquina_windows/recurso destino_montaje
```

Veamos su utilidad con un ejemplo. Recordemos que en la máquina BAG disponíamos del recurso `C`, montémoslo en nuestro sistema de archivos. Para eso creamos un directorio destino de montaje:

```
# mkdir /mnt/bag
```

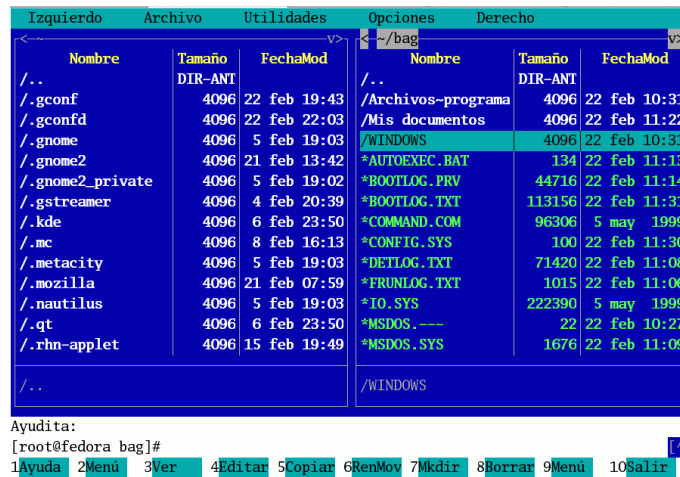
ya sí, usemos ahora

```
# smbmount //bag/C /mnt/bag
```

Password:

Al usar este comando, estamos consiguiendo que el recurso compartido sea montado en `/mnt/-bag` y que lo veamos como cualquier otra parte del sistema de archivos Linux.

³⁴Tanto `smbmount` como `smbumount` las podrán usar los usuarios “normales” si les activamos el bit `setuid`. Para ampliar sobre su uso os remitimos a las `manpages` de ambos programas.



- Podemos acceder a máquinas Windows sin necesidad de usar el paquete Samba. Esto es posible gracias al soporte del sistema de archivos `smbfs` del núcleo. Si nuestro núcleo está compilado con esta opción (los de RedHat lo están) podemos montar recursos compartidos de máquinas Windows sin tener activos los demonios del paquete Samba y sólo necesitamos el programa `mount`. Si optamos por esta opción, para montar el recurso `C` de la máquina `BAG` escribiremos:

```
#mount -t smbfs //bag/c /mnt/bag
```

- Debemos tener cuidado porque los archivos de texto de Linux y de Windows son diferentes. Si creamos archivos de texto en un recurso montado vía `SAMBA` desde Linux, éste utilizará el formato de fin de línea del sistema Linux y después tendremos problemas para leerlos desde windows.

Para desmontarlo usaremos el comando `smbumount`

```
#smbumount /mnt/bag
```

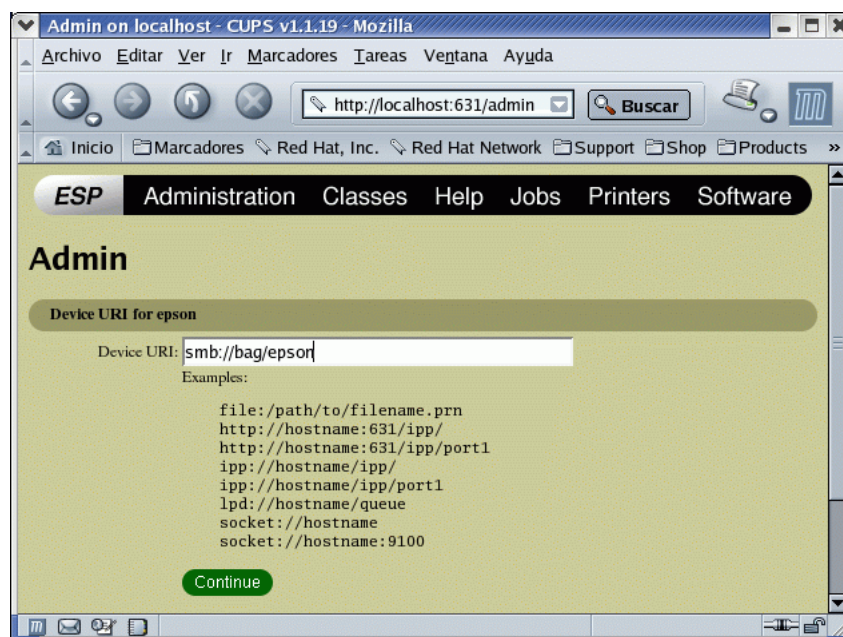
o sólo `umount`

```
# umount /mnt/bag
```

Agregar una impresora. Veamos cómo imprimir vía red usando una impresora conectada a una máquina Windows (notar que previo debe de estar dada a compartir). Supongamos que disponemos de la impresora `EPSON` en la máquina `BAG`, añadámosla a Linux. Iniciamos `CUPS`, optamos por añadir una impresora (véase 6.4.2 en la página 117) y en Device URI escribimos los datos adecuados a nuestro sistema³⁵:

³⁵Si no existe el enlace simbólico lo podemos crear con:

```
#ln -s /usr/bin/smbpool /usr/lib/cups/backend/smb
```



- Si la máquina está en el mismo grupo de trabajo y no se requieren usuario y password:
smb://servidor/recursocompartido
- Si la máquina está en el mismo grupo de trabajo y se requieren usuario y password:
smb://nombreusuario:password@servidor/recursocompartido
- Si la máquina está en otro grupo de trabajo y no se requieren usuario y password:
smb://grupotrabajo/servidor/recursocompartido
- Si la máquina está en otro grupo de trabajo y se requieren usuario y password:
smb://nombreusuario:password@grupotrabajo/servidor/recursocompartido

Una vez especificada la URI correctamente sólo nos falta seleccionar el filtro adecuado.

6.4.2. Acceder desde Windows a la máquina Linux

Un papel más interesante para Linux y Samba en una red Windows es el de servidor de recursos. Para conseguir esto debemos tener instalado el paquete `samba` y tener correctamente configurado el fichero `/etc/samba/smb.conf`.

Para acceder (sin hacer más cambios que los ya comentados³⁶) desde una máquina Windows a una Linux sólo lo podremos hacer si somos usuarios registrados en el sistema Linux. Además, en Linux debemos dar de alta a ese usuario y definir con qué contraseña puede acceder a nuestro sistema. Si queremos dar de alta al usuario `cursolinux` de la máquina Linux (con igual nombre en la máquina Windows) usaremos:

```
#smbpasswd -a cursolinux
ENTER password for cursolinux
New SMB password:
Retype new SMB password:
Added user cursolinux.
```

³⁶Puede haber también directorios públicos, para los cuales no hace falta que el usuario esté dado de alta como usuario del linux. A los usuarios registrados en Linux además les muestra su directorio `home`.

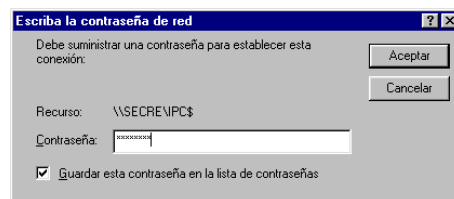
También, si existe un NT que valida los usuarios en la red, se puede poner ese NT como *password server* y es el que valida los usuarios.

Podremos cambiar la contraseña después usando el comando `smbpasswd` (sin el parámetro `-a`). Si lo que deseamos es borrar un usuario le pasaremos como parámetro `-x` seguido del nombre de ese usuario.

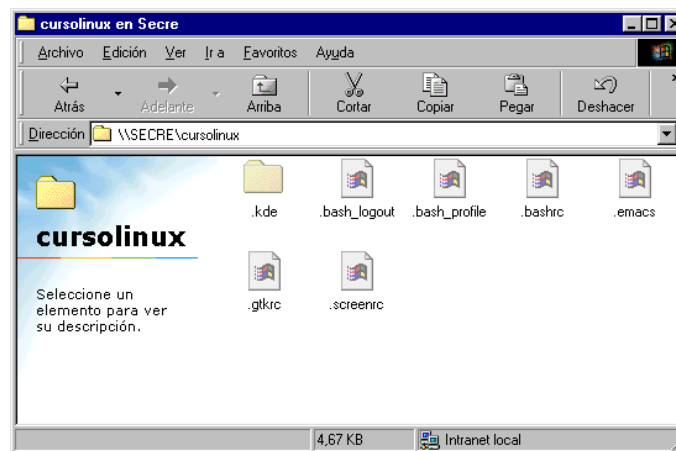
Si nuestro usuario `cursolinux` desea tener otro alias en el sistema tenemos que usar el fichero `/etc/samba/smbusers`. Por ejemplo:

```
root = administrator admin
cursolinux = thales mileto
```

En la máquina Windows 9x tendremos que entrar **como ese usuario** (si se trata de un XP esto no es necesario) y si en el inicio no ponemos la contraseña, ésta se nos pedirá cuando intentemos acceder a los servicios de Red:



Una vez validada la contraseña (interesa desmarcar la casilla de guardar contraseña ya que si no, nos arriesgamos a que nos fastidien el Linux desde Windows), los directorios a los que tengamos acceso en nuestro sistema Linux se nos mostrarán como carpetas de Windows y podremos trabajar con ellas de la forma habitual.



Es importante resaltar que si el grupo de trabajo no está bien configurado en el fichero `/etc/smb.conf` no veremos a la máquina Linux cuando accedamos a la red.

Agregar una impresora.



Vamos a partir de la base de que los *drivers* se instalan de lado del cliente (en la máquina Windows). Si se opta por usar el servidor SAMBA como depósito de los *drivers*, véase la documentación de SAMBA.

Para poder usar una impresora configurada con CUPS desde una máquina Windows hay que realizar una serie de cambios en los ficheros de configuración de samba y de CUPS. Comencemos por `/etc/smb.conf`, tenemos que añadir/modificar

```
load printers = yes
printing = cups
printcap name = cups
```

en la sección [Global]. Además, la sección [Printers] ha de quedar como sigue

```
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
public=yes
guest ok = yes
writable = no
printable = yes
```



En Guadalinex no existe el directorio `/var/spool/samba` así que tenemos dos opciones:

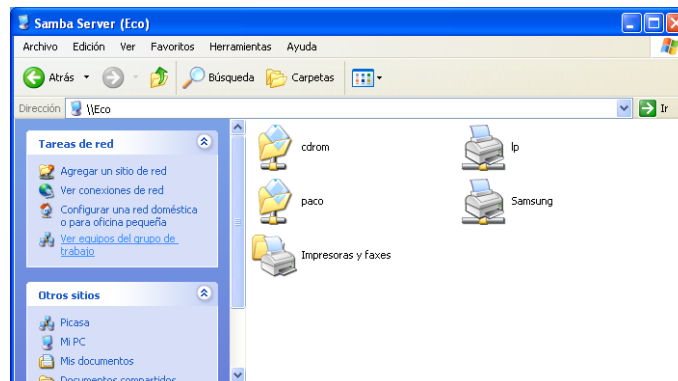
- Usar `/var/tmp`
- Crearlo con los permisos adecuados

```
#mkdir /var/spool/samba
#chmod 1777 /var/spool/samba
```

En cuanto a los ficheros de configuración de CUPS, hemos de permitir que los trabajos de impresión se manden en “bruto”, para eso, descomentaremos las líneas:

```
application/octet-stream
del fichero /etc/cups/mime.types, y
application/octet-stream application/vnd.cups-raw 0 -
del fichero /etc/cups/mime.convs
```

Una vez en el Windows y tras acceder como un usuario registrado de la máquina Linux, instalaremos la impresora sin más dificultad, ya que es igual que si la red fuese sólo de equipos Windows.



O sea que: se puede montar una unidad de Red en Windows usando Linux, instalar programas sobre esa unidad, cortar, copiar, pegar, imprimir, etc sin que el cliente Windows se entere; usando la seguridad, precio y potencia de un Linux.

➔ Para practicar

Vamos a partir de que en la máquina Linux hay un usuario genérico de nombre INVITADO, ajustarlo a vuestro caso particular.

1. Instalar el servidor samba y conseguir que:
 - a) El usuario INVITADO pueda acceder desde Windows al equipo Linux.
 - b) El acceso esté limitado a la red local.
 - c) Haya un recurso compartido público de sólo lectura (`/home/samba`) que sea accesible por todos los equipos de la red. Para eso:


```
# mkdir /home/samba
# chmod 755 /home/samba
y en /etc/samba/smb.conf:
security=share
y además
[public]
    comment = Directorio Público
    path = /home/samba
    public =yes
    writable = no
    printable = no
```
2. Construir una sección personalizada de nombre `[Practicas]` para el fichero `smb.conf` de manera que el directorio `/home/practicas` sea un recurso compartido de sólo lectura para el grupo `clase`. Además, deberá poder verse (la carpeta que da acceso a él) desde el navegador de windows por todos los usuarios del sistema si bien no podrán acceder a él.
3. En esta práctica vamos a conseguir que un CD se monte (y se desmonte) de forma automática cuando se accede a él desde un cliente Windows, además de ser un recurso accesible para toda la red. Sólo comentaremos las directivas nuevas:

```
[cdrom]
browseable = yes
#Activamos el soporte de bloqueos oportunistas por la-
do del cliente
oplocks = yes
guest ok = yes
#Hay que adecuar a nuestro sistema el punto de montaje
#En Guadalinux será /cdrom
path = /mnt/cdrom
#Comando a ejecutar antes de conectarse al recurso.
root preexec = /bin/mount -t iso9660 /dev/cdrom /mnt/cdrom
#Comando ejecutado al desconectarse del recurso
root postexec = /bin/umount /dev/cdrom
```

smb.conf de ejemplo

```
[global]
workgroup =_MYGROUP
3 server_string =_Samba_Server
netbios_name =_Linux
; hosts_allow =_192.168.1._127.

load_printers =_yes
8 printing =_cups
cups_options =_raw
printcap_name =_cups

[homes]
```

```
13  ___comment__= Directorios_de_usuario
    ___browseable__=no
    ___writable__=yes

[ printers ]
18  ___comment__= All_Printers
    ___path__= /var/spool/samba
    ___browseable__=no
    ___guest_ok__=no
    ___writable__=no
23  ___printable__=yes

[ compartido ]
    ___path__= /home/compartido
    ___comment__= Directorio_compartido_en_Guadalinux_2004
28  ___writeable__=no
    ___guest_ok__=yes
    ___guest_only__=yes
    ___browseable__=yes
```

Listado 6.2: /etc/samba/smb.conf

Capítulo 7

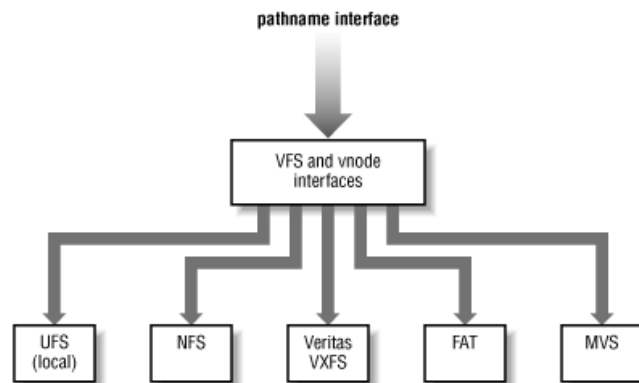
Servicio de compartición de ficheros NFS

Cuando trabajas con Linux estás ante un sistema operativo orientado al trabajo con redes de ordenadores. ¿Qué nos empuja a poder afirmar tan categóricamente? Ya te darás cuenta poco a poco. (*Manual Avanzado de Linux* de RAÚL MONTERO RIVERO, Ed. Anaya)

Un Sistema de Ficheros en Red (NFS¹) es un método de compartir archivos entre máquinas de una red, de tal forma que tenemos la impresión de trabajar en nuestro disco duro local, cuando en realidad están en otro lugar de la red. En los sistemas Unix era la forma tradicional de compartir ficheros en red, pero SAMBA ha ido ocupando gran parte de su terreno, debido a la necesidad de comunicarse con máquinas Windows. La tendencia es a que surjan sistemas de ficheros accesibles a través de la red, como WebDAV o el reciente ZFS.

Un *servidor NFS* es el que ofrece uno o varios de sus sistemas de ficheros para que otros sistemas los puedan utilizar. El *cliente NFS* es el que monta un sistema de ficheros de un sistema remoto sobre su sistema de ficheros local, accediendo a él como si estuviera en sus discos.

El sistema de ficheros remoto se integra en la jerarquía local y las aplicaciones accederán a él de forma transparente, sin darse cuenta de que están en una ubicación remota. Esa independencia le hace acceder a distintos sistemas de ficheros, ya sean ext3, FAT, NTFS, UFS o VXFS de la misma manera, a través de una entrada en el árbol de directorios.




7.1. Servidor NFS

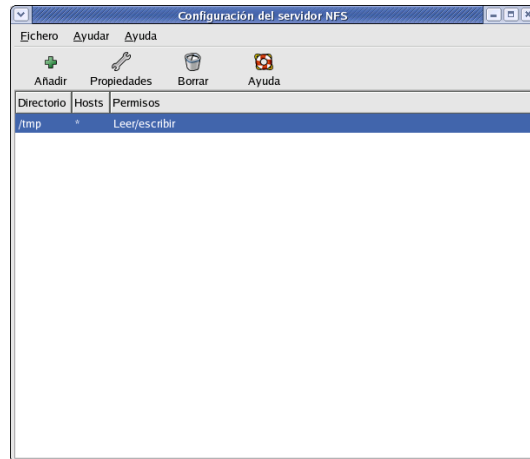
Compartir archivos desde un servidor NFS es conocido como exportar directorios. Para disponer de los servicios NFS tenemos que tener activos (en Fedora) los demonios `nfsd` y `mountd` que forman

¹Network File System

parte del paquete `nfs-utils`².

La herramienta de configuración del servidor NFS se puede usar para configurar un sistema como servidor NFS.

Para usar la Herramienta de configuración del servidor NFS, desde el entorno gráfico, debe tener el paquete RPM `system-config-nfs` instalado. Para iniciar la aplicación, seleccione Botón de menú principal ( → **Configuración del sistema** → **Configuración de servidores** → **Servidor NFS**, o escriba el comando `system-config-nfs`.



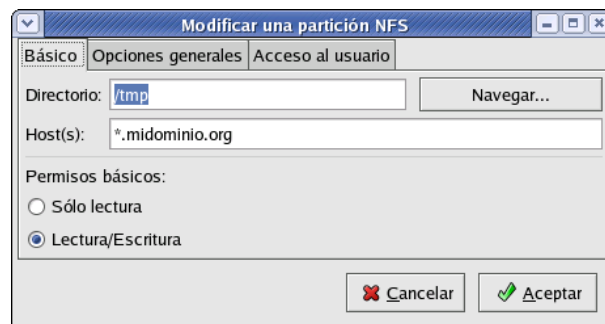
Para añadir una partición NFS, pulse el botón Añadir. Aparecerá un cuadro de diálogo con tres pestañas: **Básico**, **Opciones generales** y **Acceso al usuario**.

La pestaña Básico necesita que le aportemos la siguiente información:

Directorio Especifique el directorio a compartir, por ejemplo `/tmp`.

Host(s) Especifique el o los hosts con los que compartir el directorio.

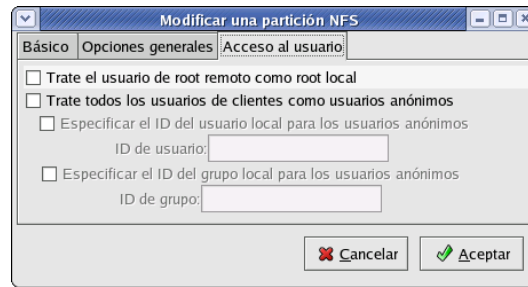
Permisos básicos. Especifique si el directorio deberá tener permisos de sólo lectura o de lectura y escritura.



Las opciones de las siguientes pestañas normalmente no serán utilizadas. La siguiente figura muestra la pestaña de *Acceso al usuario*.

²En Debian

```
#apt-get install nfs-common portmap nfs-kernel-server
```



Después de **Aceptar** la opción de añadir, modificar o eliminar un directorio compartido³ mediante NFS desde la lista, los cambios tendrán efecto inmediatamente. El demonio del servidor es reiniciado por la interfaz gráfica y el archivo de configuración antiguo es guardado como `/etc/exports.bak`. La nueva configuración es escrita a `/etc/exports`. También podemos modificar el archivo de forma manual.

7.1.1. Fichero `/etc/exports`

El archivo `/etc/exports` controla qué directorios exporta el servidor NFS. Su formato es como se muestra a continuación:

```
# more /etc/exports
/tmp *.midominio.org(rw, sync)
/usr/local thales(ro) mileto(ro)
/ pitagoras(rw, no_root_squash)
```

Pasemos a comentarlo. Se permite montar el directorio `/tmp` a los hosts de `midominio.org` con permisos de lectura/escritura. El directorio `/usr/local` lo podrán montar las máquinas `thales` y `mileto` con permisos de sólo lectura y el directorio raíz, se podrá montar desde la máquina `pitágoras` con permisos de lectura y escritura. No muy recomendale esto último.

Formato del nombre de host

El nombre de host puede ser de alguna de las siguientes maneras:

Máquina única Nombre de dominio completamente cualificado (`thales.cica.es`), nombre del host que puede ser resuelto por el servidor (`thales`, sabiendo que se encuentra en el dominio `cica.es`) o dirección IP.

Series de máquinas especificadas con comodines. Se usa el caracter `*` o `?` para especificar una cadena de caracteres que coincida. Por ejemplo, `192.168.100.*` especifica cualquier dirección IP que comience con `192.168.100`. Cuando se usan comodines en nombres de dominio completos, los puntos (`.`) no son incluidos en el comodín. Por ejemplo, `*.cica.es` incluye `thales.cica.es` pero no incluye `sirio.sistemas.cica.es`.

Redes IP. Se usa el formato `a.b.c.d/z`, donde `a.b.c.d` es la red y `z` es el número de bits en la máscara de red (por ejemplo `192.168.0.0/24`). Otro formato aceptable es `a.b.c.d/netmask`, donde `a.b.c.d` es la red y `netmask` es la máscara de red (por ejemplo, `192.168.100.8/255.255.255.0`).

Flags

Sin ser exhaustivos, comentaremos algunos valores de flags que podemos tener tras el nombre de host.

ro El sistema de ficheros se montará en modo de sólo lectura.

rw El sistema de ficheros se montará en modo de lectura/escritura.

³En inglés *share*

root_squash El superusuario del sistema cliente (el que monta el sistema de ficheros) no tendrá privilegios especiales sobre el sistema de ficheros que se monte.

no_root_squash Lo contrario del anterior, el superusuario sigue siendo el “jefe” incluso en los ficheros remotos.

Para ver el estado del demonio NFS desde Fedora utilizamos el siguiente comando:

```
/sbin/service nfs status
```

Lo arrancamos como de costumbre:

```
/sbin/service nfs start
```

En Debian, para reiniciar el servicio usaremos:

```
/etc/init.d/nfs-kernel-server restart
```

7.1.2. RPC y portmap

NFS se apoya en las llamadas de procedimientos remotos (RPC⁴) para funcionar. Se requiere del demonio **portmap** para enlazar las peticiones RPC a los servicios concretos. Los procesos RPC notifican a **portmap** cuándo comienzan, indicando el número de puerto en el que están esperando y el número de programas RPC que esperan servir. En definitiva, **portmap** es un demonio que se encarga de controlar puertos y de asignar programas RPC a dichos puertos.

El sistema cliente contacta con el **portmap** del servidor con un número de programa RPC particular. Entonces **portmap** redirecciona al cliente al número del puerto apropiado para que se comunique con el servicio adecuado.

Como **portmap** proporciona la coordinación entre servicios RPC y los números de puertos usados para comunicarlos, es útil poder visualizar el estado de los servicios RPC actuales usando **portmap** cuando estamos resolviendo algún problema. El comando **rpcinfo** muestra cada servicio basado en RPC con su número de puerto, número de programa RPC, versión y tipo de protocolo (TCP o UDP).

Para asegurarnos de que los servicios NFS basados en RPC están activos para **portmap**, podemos utilizar el comando **rpcinfo -p**:

```
# rpcinfo -p
programa vers proto puerto
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 1024 status
100024 1 tcp 1026 status
391002 2 tcp 1027 sgi_fam
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100021 1 udp 1046 nlockmgr
100021 3 udp 1046 nlockmgr
100021 4 udp 1046 nlockmgr
100021 1 tcp 1856 nlockmgr
100021 3 tcp 1856 nlockmgr
100021 4 tcp 1856 nlockmgr
100011 1 udp 784 rquotad
100011 2 udp 784 rquotad
100011 1 tcp 787 rquotad
100011 2 tcp 787 rquotad
```

⁴Remote Procedure Call

```
100005 1 udp 792 mountd
100005 1 tcp 795 mountd
100005 2 udp 792 mountd
100005 2 tcp 795 mountd
100005 3 udp 792 mountd
100005 3 tcp 795 mountd
```

7.2. Cliente NFS

El comando `mount` es el utilizado para montar directorios de NFS compartidos desde otra máquina, al igual que se montaría un sistema de ficheros local:

```
#mount thales:/tmp /mnt/tmp
```

Donde `thales:/tmp` significa el directorio `/tmp` de la máquina `thales` que montaremos bajo el directorio `/mnt/tmp` de la máquina local. El directorio `/mnt/tmp` de la máquina local debe existir y tener los permisos adecuados para el usuario que intenta montarlo.

Una vez hayamos ejecutado el comando `mount` (siempre que tengamos los permisos adecuados en el servidor `thales`), podremos teclear `#ls /mnt/tmp` y obtener un listado de los archivos que se encuentran en el directorio `/tmp` de la máquina `thales`.

Para que un cliente pueda montar sistemas de fichero remotos mediante NFS, debe soportar este tipo de sistema de ficheros en el kernel. Podemos comprobarlo mediante:

```
$cat /proc/filesystems
nodev rootfs
nodev bdev
nodev proc
nodev sockfs
nodev tmpfs
nodev shm
nodev pipefs
ext2
nodev ramfs
iso9660
nodev devpts
ext3
nodev usbdevfs
nodev usbfs
nodev autofs
vfat
nodev nfs
```

Las siguientes órdenes nos muestran cómo se montaría un sistema de ficheros NFS.

```
#mount -t nfs linux:/tmp /mnt/tmp
```

La apariencia del sistema de ficheros montado junto a los sistemas locales.

```
[root@linux images]# df
S.ficheros Bloques de 1K Usado Dispon Uso% Montado en
/dev/hda2 2276952 2048220 113064 95% /
none 62988 0 62988 0% /dev/shm
linux:/tmp 2276952 2048220 113064 95% /mnt/tmp
Desmontamos el sistema de ficheros
[root@linux images]# umount /mnt/tmp
```

7.2.1. Montar sistemas de archivos NFS usando `/etc/fstab`

Un método alternativo para montar datos compartidos mediante NFS es añadir una línea en el archivo `/etc/fstab`. La línea debe incluir el nombre del servidor NFS, el directorio que el servidor está exportando y el directorio de nuestra máquina local donde queremos montar el sistema de

archivos. Recordad que debéis tener permisos de superusuario para poder modificar el archivo `/etc/fstab`.

La sintaxis general de esta línea del archivo `/etc/fstab` es la siguiente:

```
server:/tmp /mnt/tmp nfs rsize=8192,timeo=14,intr
```

Los valores de opciones que podemos utilizar son:

rsize=n, wsize=n Especifican el tamaño del datagrama⁵ utilizado por los clientes en las peticiones de lectura y escritura, respectivamente.

timeo=n Especifica el límite en décimas de segundo que el cliente esperará que una petición se complete. Lo que ocurre después de un timeout (agotamiento del tiempo) depende de si hemos utilizado la opción **hard** o la opción **soft**.

hard Es la opción por defecto. El cliente, si no puede contactar con el servidor, presenta un mensaje por pantalla y continúa intentando indefinidamente la operación.

soft Causa que al ocurrir un timeout, la operación falle con un error de entrada/salida y no se reintente más.

intr Permite mandar señales de interrupción a la llamada NFS. Es útil para abortar la operación cuando el servidor no responde.

⁵Sí, NFS utiliza UDP, por eso son datagramas. Las nuevas versiones empiezan a incorporar TCP.

Capítulo 8

Servicio de Proxy-caché

-Como me quieres bien, Sancho, hablas desa manera -dijo don Quijote-; y, como no estás experimentado en las cosas del mundo, todas las cosas que tienen algo de dificultad te parecen imposibles; pero andará el tiempo, como otra vez he dicho, y yo te contaré algunas de las que allá abajo he visto, que te harán creer las que aquí he contado, cuya verdad ni admite réplica ni disputa. (*El ingenioso hidalgo Don Quijote de la Mancha*. MIGUEL DE CERVANTES SAAVEDRA).

8.1. ¿Qué es un proxy caché?

Existen dos métodos para que los usuarios de nuestra red naveguen por internet. El primer método implica que los usuarios tengan sus ordenadores conectados directamente a internet. Los navegadores solicitan directamente las páginas a los servidores web remotos. Este método es el utilizado cuando navegamos por internet desde un ordenador que está conectado por módem, cable-módem o ADSL directamente.

El otro método implica un almacenamiento (caché) de las páginas que se visitan. Siempre que un usuario quiera visitar una página web, el navegador se conectará a un servidor de caché que le suministrará la página (caso de tenerla almacenada) o la solicitará al servidor web remoto (caso de no disponer de ella).

En caso de optar por la segunda opción, puede reducirse considerablemente el tiempo de descarga de una página. Las páginas que se encuentran en caché no requieren acceso al servidor remoto y las páginas que no están en caché no introducen un tiempo extra en la descarga, al disponer el servidor de todo el ancho de banda para descargar las páginas que no están en caché.

8.2. Squid, un proxy caché para Linux

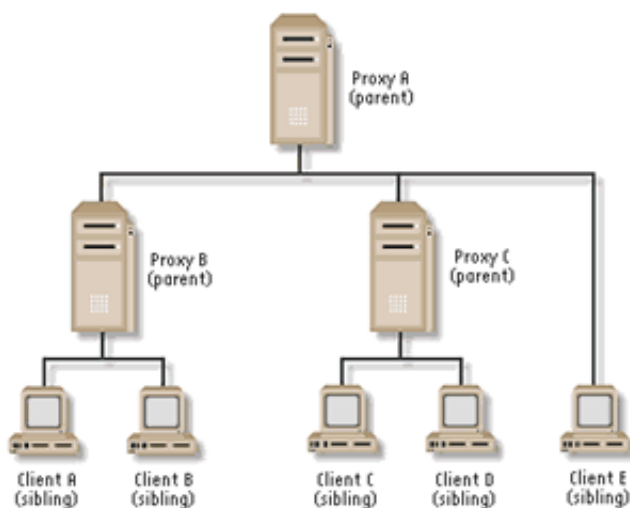
8.2.1. Visión general

Como alternativa al software comercial existente, apareció Squid. Su funcionamiento se basa en guardar las peticiones que hacen los usuarios a servidores web remotos. Cuando un usuario quiere acceder a una página la solicita a Squid, que se encarga de acceder al servidor web remoto. Una vez obtenida, la reenvía al usuario, guardando una copia. En el caso que otro usuario solicite de nuevo esa página, únicamente tendrá que recuperarla de su disco local y servirla.

Otra función que realiza Squid es la de proporcionar un servicio de proxy a ordenadores que necesiten acceder a internet a través de algún tipo de cortafuegos. Por eso es común denominar a Squid como un proxy caché, al unir las dos funcionalidades que presenta.

Squid puede almacenar datos de los protocolos HTTP, FTP, Gopher y DNS. El tener un servidor de caché especializado puede reducir considerablemente el uso que se haga del ancho de banda disponible. En lugar de descargar páginas repetidamente, se comprobará si la página del

Figura 8.1: Jerarquía de proxy



servidor remoto es más nueva que la que tiene almacenada en disco. De no ser así, no se molestará en descargarla.

8.2.2. Conceptos sobre cachés

Los servidores que actúan de proxy-caché se pueden configurar de varias formas. La forma más simple es un solo servidor proxy-caché en la red en el que todos los ordenadores pertenecientes a esa red accederán a este servidor, que será el que almacenará todos los datos. Cuando un usuario solicita al servidor una página, éste comprueba si fue actualizada desde que fue almacenada. Si tiene la versión actualizada ahorra al usuario final la descarga de la misma proporcionándosela directamente.

Otro método de configurar la salida a internet de una red de ordenadores es creando una jerarquía de servidores proxy-caché. Los servidores en un nivel superior a un servidor son denominados padres (*parent*) y los que se encuentran al mismo nivel son hermanos o iguales (*siblings*, *neighbor* o *peer*).

Cuando Squid obtiene una petición de un cliente, comprueba si el objeto solicitado (página, gráfico o fichero) está en el disco del servidor. Si está, comprueba que el objeto no está caducado y procede a enviarlo al cliente. Si, por el contrario, el objeto no está o ha caducado, comprueba que otras cachés (padres o hermanas) lo tengan. Lo hace a su vez enviando paquetes UDP a esas máquinas con la URL.

La otra caché comprueba, a continuación, si tiene dicho objeto en el disco duro y envía un mensaje indicando si lo posee o no. La máquina original espera las respuestas y después decide si debe obtener el objeto de la otra caché o debe ir directamente a por él.

Cuando existe una máquina hermana el servidor le solicitará la información que no tiene y en caso de no tenerla estos servidores accederá directamente al servidor web remoto. En caso que existan también servidores padre en la configuración, la información que no tengan los hermanos la solicitará al servidor padre, que a su vez la solicitará directamente al servidor web remoto.

8.2.3. Instalación

Si no estuviera instalado, lo instalamos a partir del paquete RPM o DEB correspondiente.

Fedora: `apt-get install squid`

y una vez instalado optar porque se inicie en el arranque

```
# ntsysv
```

Guadalinux: `apt-get install squid`

Los ficheros y directorios más importantes son:

- En el directorio `/etc/squid` se guardan los ficheros de configuración. Específicamente en el fichero `squid.conf` se encuentra la mayor parte de ella.
- Una parte importante de ficheros se encuentran en `/usr/lib/squid`, pero no tendremos que preocuparnos de ellos por ahora.
- La documentación se encuentra en `/usr/share/doc/squid-x.x.x/`
- En `/var/spool/squid` se van a encontrar las páginas “cacheadas”, es decir, las traídas desde Internet y que se almacenan para la próxima vez que las solicite alguien y no hayan cambiado.
- En `/var/log/squid` se guardan los accesos de nuestros usuarios a Internet a través del proxy, así como los posibles errores que hayan ocurrido.

8.3. Configuración de Squid

8.3.1. Configuración básica

El archivo de configuración que utiliza Squid es `squid.conf`, como hemos dicho se encuentra situado en la ruta `/etc/squid/squid.conf`, pudiendo encontrarse en otras localizaciones dependiendo de la instalación. Este archivo está ampliamente comentado por lo que no lo analizaremos de forma detallada, sino que haremos un rápido recorrido por el fichero de configuración centrándonos en los aspectos que consideremos más importantes.

Una de las primeras directivas de configuración que aparece es:

```
http_port 3128
```

Indica el puerto en el que va a estar escuchando Squid.

A continuación podemos ver el parámetro que indica el puerto en el que Squid escucha las peticiones ICP¹.

```
icp_port 3130
```

Como ya hemos descrito, Squid es un proxy-caché y pueden existir elementos que no queramos almacenar. Esto puede conseguirse a través del fichero de configuración. Con la siguiente línea no almacenaríamos en caché ningún objeto que se encuentre en la ruta `cgi-bin`:

```
acl QUERY urlpath_regex cgi-bin \?  
no_cache deny QUERY
```

Posteriormente veremos con más detalle la sintaxis y usos de la directiva `acl` para la creación de clases.

Es posible también definir parámetros de uso de memoria. Si queremos definir la cantidad de memoria RAM que deseamos asignar a las funciones de Squid con un valor de 8 Mb²

```
cache_mem 8 MB
```

¹Protocolo utilizado para comunicaciones entre distintos cachés

²Con las configuraciones actuales y en función del uso de nuestra máquina, en general, debemos optar por un valor mayor.

Es posible también definir cuando se empieza a eliminar archivos de la caché. Cuando la caché llega al total del porcentaje de `cache_swap_high` Squid comienza a eliminar los elementos almacenados menos utilizados hasta que llega al total del porcentaje `cache_swap_low`.

```
cache_swap_low 90
cache_swap_high 95
```

Otra alternativa para regular el caché es configurarlo para que no almacene archivos que tengan un tamaño mayor que el indicado:

```
maximum_object_size 4096 KB
```

Ya hemos comentado que el caché de Squid es un espacio en disco reservado para almacenar los distintos objetos que se piden a través del proxy. Será necesario definir el lugar donde se va a almacenar el caché³.

```
cache_dir ufs /var/spool/squid 100 16 256
```

El formato genérico de esta directiva es:

```
cache_dir tipo directorio Mbytes L1 L2 [options]
```

- **tipo.** Tipo de sistema de almacenamiento a utilizar (ufs es el único que está definido por defecto en la instalación).
- **directorio.** Ruta del directorio que se va a utilizar para guardar los datos del caché.
- **Mbytes.** Cantidad de espacio en disco que se va a utilizar para el caché. Si queremos que utilice el disco entero es recomendable poner aquí un 20% menos del tamaño.
- **L1.** Número de subdirectorios de primer nivel que serán creados bajo directorio.
- **L2.** Número de subdirectorios de segundo nivel que serán creados bajo cada subdirectorio de primer nivel.

Una consideración a tener en cuenta es que el contenido de este directorio va a cambiar con frecuencia, siendo recomendable colocarlo en una partición separada por varias razones:

- La caché podría sobrepasar al resto del sistema de archivos o de la partición que comparte con otros procesos.
- Cuanto más cambie un sistema de archivos, mayores son también las posibilidades de que se encuentre dañado. Mantener la caché en una partición limita la parte de su sistema completo de archivos que resulta dañado.

También es posible configurar la localización de los archivos de log así como la información general de la caché⁴:

```
cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log /var/log/squid/store.log
```

³El valor de 100MB es escaso para los discos duros actuales, un valor de 1GB puede ser mejor si nuestro disco lo permite.

⁴Los ficheros de contabilidad que deja, pueden ser monitorizados para impedir accesos a Internet no deseados. Un ejemplo real es el de un organismo que manda semanalmente a los usuarios de Internet un fichero con los accesos en ese periodo. El usuario se siente controlado y es más responsable con sus accesos.

Una herramienta sencilla de configurar y muy útil para obtener estadísticas sobre las páginas visitadas es `sarg`, la podéis conseguir de <http://web.onda.com.br/orso/sarg.html>. La analizaremos mejor al final de este capítulo.

8.3.2. Configuración de jerarquía de caché

De los anteriores conceptos sobre jerarquía de caché podemos deducir que el empleo de este tipo de infraestructura puede ser complicado y a veces puede no merecer la pena. Dependiendo del tipo de instalación y del tráfico que manejemos puede ser interesante su consideración.

Como ya hemos comentado, si configuramos el caché para tener sólo hermanos, dicha caché enviará las peticiones UDP a la lista de hermanos y si no poseen dicho objeto, Squid se conectará directamente al servidor web remoto.

En el caso de realizar una configuración del caché para tener un padre, significa que si dicha caché no tiene el objeto, y ninguna de las hermanas lo posee, abrirá una conexión TCP a la caché padre para que ésta obtenga el objeto. Al ser una conexión TCP, esta caché padre posiblemente recorrerá la lista de cachés padres y hermanas para buscar el objeto (sólo les envía una petición UDP para comprobar si la poseen en el disco duro). Lo que complica las cosas es tener múltiples cachés padres y hermanas.

Obviamente, si sólo un padre tiene el objeto, lo descargará de allí, pero si ninguna lo posee, su caché dará la petición a la máquina que respondió más rápida, suponiendo que es la máquina menos saturada o que posee una conexión más directa.

Existe otra opción que es balancear las cachés, repartiendo la carga. Incluso se puede especificar que use una caché padre para descargar las peticiones que no haya podido obtener.

Veamos ahora cómo pueden configurarse estas opciones en Squid, lo que nos aclarará un poco los conceptos.

```
cache_host cache.mordor.com parent 3128 3130
```

En este ejemplo sólo existe una caché a la que Squid va a preguntar. Sin embargo, podemos ajustar un poco más la configuración de forma que se conectará al caché padre para todas las peticiones, a diferencia del ejemplo anterior que necesitaba saber si estaba o no activa.

```
cache_host cache.mordor.com parent 3128 3130 no-query default
```

Otra característica muy útil de este sistema es la capacidad de cachés hermanas. Supongamos que no desea gastar mucho dinero en una sola caché, y desea balancear la carga entre varias máquinas, pero no duplicando los objetos en cada máquina. Es posible configurar estas máquinas para que hablen entre ellas mediante las señales `proxy-only`, como este ejemplo:

Configuración en caché 1

```
cache_host cache2.gondor.com sibling 3128 3130 proxy-only
```

Configuración en caché 2

```
cache_host cache1.rohan.com sibling 3128 3130 proxy-only
```

En este caso, si una petición se dirige a la caché 1 y no se encuentra en el disco, esta caché enviará una petición ICP a la caché 2 (3, 4, etc) y la descargará de allí, pero no la salvará en su disco duro, tan solo la descargará de la otra caché cuando la necesite de nuevo. Este sistema duplicará la capacidad del disco duro sin necesidad de gastar grandes cantidades de dinero en sistemas raid para soportar muchos gigas.

8.3.3. Control de acceso

Otro aspecto importante en la configuración de Squid son las listas de control de acceso o ACL⁵. Una de sus principales funciones es la de permitir o denegar el acceso a la caché, aunque no se queda aquí. Las ACL pueden usarse también para definir las jerarquías de caché.

⁵ *Access Control List*



El procedimiento que se sigue es definir las distintas ACL y posteriormente se permite o deniega el acceso a una determinada función de la caché. La opción de configuración encargada es `http_access`, que permite o deniega al navegador web el acceso a Squid.

Es importante tener en cuenta que Squid lee las directivas de arriba a abajo para determinar qué regla aplicar.

Veamos un primer ejemplo de uso. Supongamos que disponemos de una red de clase C (con direcciones IP dentro de la red 172.26.0.0) y que solo queremos permitir el acceso a internet a través de Squid a estas máquinas.

```
acl hostpermitidos src 172.26.0.0/255.255.255.0
acl all src 0.0.0.0/0.0.0.0
http_access allow hostpermitidos
http_access deny all
```

Las dos primeras líneas de este ejemplo crean las ACL `hostpermitidos` y `all`. El formato general de esta directiva es:

```
acl nombreACL tipoACL cadena ...
acl nombreACL tipoACL "fichero" ...
```

donde:

- `nombreACL` es el nombre que corresponde a esta definición ACL
- `tipoACL` es el tipo de elemento contenido en esta definición
- `cadena` o `fichero` es el argumento apropiado al `tipoACL`, pudiendo haber más de un argumento en la lista

Cuando utilicemos un fichero como origen de los datos de la ACL deberá contener un elemento por línea. Algunos de los tipos de ACL que podemos utilizar en esta directiva son:

- Direcciones IP de los clientes. Especifica la dirección IP local, dirección de red o rango de direcciones a buscar

```
acl nombreACL src dirIP/máscara
acl nombreACL src dirIP1-dirIP2/máscara
```

- Dirección IP de la URL destino. Especifica la dirección IP de la máquina remota, dirección de red o rango de dirección a buscar

```
acl nombreACL dst dirIP/máscara
acl nombreACL dst dirIP1-dirIP2/máscara
```

- Dominio de la máquina cliente. Especifica el `host.dominio.extensión` o bien el `dominio.extensión` a buscar

```
acl nombreACL srcdomain nombreDominio
```

- Dominio de la máquina destino. Especifica el `host.dominio.extensión` o bien el `dominio.extensión` a buscar

```
acl nombreACL dstdomain nombreDominio
```

- Expresión regular que concuerda con el nombre del cliente

```
acl nombreACL srcdom_regex [-i] expresión
```

- Expresión regular que concuerda con el nombre del servidor

```
acl nombreACL dstdom_regex [-i] expresión
```

- Control por día y hora. Especifica la información de tiempo a buscar

```
acl nombreACL time [día] [h1:m1-h2:m2]
```

M - Lunes

T - Martes

W - Miércoles

H - Jueves

F - Viernes

A - Sábado

S - Domingo

h1:m1 < h2:m2

- Expresión regular que concuerda con la URL completa

```
acl aclname url_regex [-i] ^http://expresión
```

- Expresión regular que concuerda con la ruta de la URL

```
acl aclname urlpath_regex [-i] \.gif$
```

La primera de las opciones permite decidir en qué ACL se encuentra la dirección IP del usuario. También podemos decidir sobre aspectos como el tiempo actual o el sitio al que se dirigen. Para más información sobre estos y otros tipos de ACL recomendamos acceder al fichero `/etc/squid/squid.conf`.

Una vez definidas las ACL entra en juego la directiva `http_access`, que se utiliza para permitir o denegar el acceso de una determinada ACL, siempre y cuando el cliente utilice el método HTTP para solicitar el objeto al servidor web remoto.

Hay que tener en cuenta que la lectura se realiza de arriba hacia abajo, y se para en la primera coincidencia para decidir si permitir o denegar la petición. En el ejemplo anterior, Squid verá que la primera línea `http_access` se cumple, y procederá a aplicarla. En este caso, permitirá el acceso y ejecutará la petición.

Pero tengamos en cuenta el siguiente ejemplo:

```
acl hostpermitidos src 172.26.0.0/255.255.255.0
acl all src 0.0.0.0/0.0.0.0
http_access deny all
http_access allow hostpermitidos
```

En este caso no funcionará, ya que Squid aplicará la primera coincidencia (la primera línea) y denegará el acceso.

Las ACL son especialmente útiles cuando queremos prohibir el acceso a una lista de sitios inapropiados (enlaces a páginas web con contenido pornográfico, ...). Squid no está optimizado para gestionar una larga lista de sitios, pero puede gestionar un número concreto de sitios sin problemas.

```
acl adultos dstdomain playboy.com sex.com
acl hostpermitidos src 172.26.0.0/255.255.255.0
acl all src 0.0.0.0/0.0.0.0
http_access deny adultos
http_access allow hostpermitidos
http_access deny all
```



En este caso, las direcciones que serán consideradas como inadecuadas son las que tienen los dominios `playboy.com` o `sex.com`. Estas URL tendrán filtrado el acceso y no será posible acceder a ellas, tal como indica la directiva `http_access deny adultos`. Si se piden otras URL, Squid pasará a evaluar las siguientes directivas. Por tanto, si el cliente se conecta dentro del rango permitido se cursará la petición. De lo contrario, la petición será rechazada.

La configuración que viene por defecto es denegar todos los accesos, mediante:

```
http_access deny all
```

Obviamente deberemos al menos incluir algún grupo que pueda acceder, porque si no, nuestro proxy sería innecesario.

Una última observación, este método considera exclusivamente los dominios. Para evitar conexiones especificando la IP de la máquina, utilizaremos la directiva `dst acl`.

Suponiendo que ya hemos definido nuestra política de acceso, arrancamos el servicio

```
#!/etc/init.d/squid start
```

La primera vez que lo ejecutemos tardará un ratito porque tiene que construir sus índices para el almacenamiento de páginas.

```
root@guadalinux:/usr/lib/squid# /etc/init.d/squid start
Starting proxy server: Creating squid spool directory structure
2005/02/15 14:18:23| Creating Swap Directories
squid.
root@guadalinux:/usr/lib/squid#
```



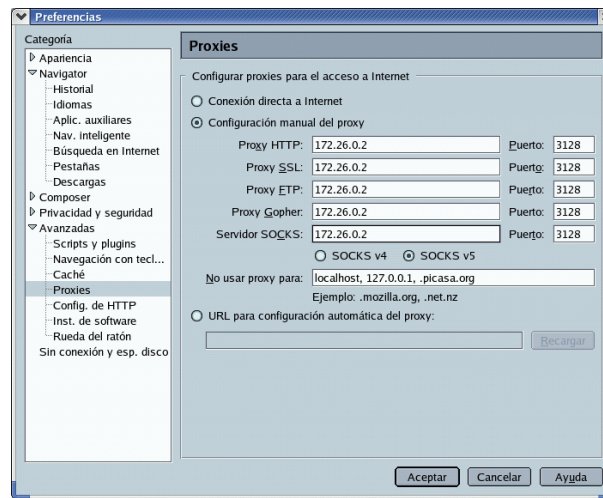
En función de la versión de squid puede que nos aparezca un error al ponerlo en marcha, en general se debe a que hay que configurar correctamente la directiva `visible_hostname` (aparece comentado por defecto).

8.4. Configuración de los clientes

El cliente lo configuraremos de la siguiente forma:

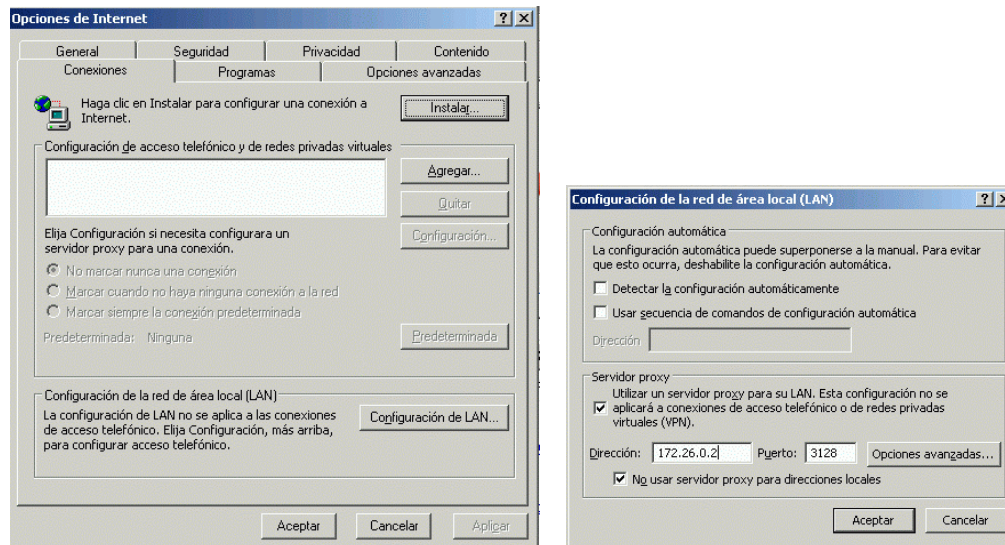
- Si es Mozilla, seleccionamos **Editar**→**Preferencias**→**Avanzadas**→**Proxy**→**Configuración Manual del Proxy**, pulsamos **Ver** y en los distintos protocolos ponemos el host⁶ `176.26.0.2` por el puerto `3128` que es por el que escucha el squid peticiones de sus clientes. Podemos ponerlo en todos los protocolos menos en el socks.

⁶Suponemos que el proxy-caché está a la escucha en la máquina de IP `176.26.0.2`



Veremos que nuestros accesos a Internet son mucho más rápidos y el control que podemos llegar a tener es muy importante.

- Si es Explorer optaremos por seleccionar **Configuración de LAN...** y una vez allí marcar la IP de la máquina con el proxy squid, el puerto (por defecto 3128) y mejor si no usamos el proxy para direcciones locales



- Si trabajamos en modo consola y deseamos definir la variable de entorno⁷ `http_proxy`, podemos usar:

```
export http_proxy="http://ip_proxy:3128"
```

Para ver que todo está bien:

```
lynx http://www.iesmurgi.org
```

Es útil, por ejemplo, para actualizar un sistema con `yum` o `apt-get` que accede a internet a través de un proxy

⁷Si añadimos la variable a algún script de arranque se tomará como valor por defecto. Desde GNOME o KDE también podemos configurarla.



8.5. Acceso a internet autenticado contra ldap

8.5.1. Métodos de autenticación de Squid

Squid tiene un potente conjunto de utilidades que permiten la autenticación del proxy. Mediante la autenticación, las peticiones HTTP de los clientes contienen una cabecera que incluye las credenciales de autenticación. Estas credenciales suelen consistir en una pareja usuario/clave. Squid decodifica esta información y posteriormente realiza una consulta a un proceso de autenticación externo, el cual se encarga de la verificación de las credenciales.

Squid soporta varias técnicas de autenticación:

- Basic
- Digest
- NTLM

Nos centraremos en la autenticación básica. Aunque es una técnica insegura debido a que la pareja usuario/clave va a viajar en claro por la red, nos servirá de aproximación al resto de métodos, algo más complicados de implementar.

Ya vimos en un apartado anterior como configurar ldap y aprovecharemos ahora esta base de datos de usuario para la autenticación. Utilizaremos los programas de ayuda que proporciona Squid, más concretamente `/usr/lib/squid/ldap_auth`⁸.

Las directivas de Squid encargadas de definir los parámetros del mecanismo de autenticación son `auth_param` y `proxy_auth`. Es muy importante el orden en que se ponen estas directivas en el fichero de configuración.

Será necesario definir al menos un método de autenticación con `auth_param` antes que ninguna ACL definida con `proxy_auth` haga referencia al mismo. En caso contrario Squid mostrará un mensaje de error e ignorará la ACL definida con `proxy_auth`, aunque la ejecución de Squid seguirá su curso. La directiva `proxy_auth` tomará los nombres de usuario como valores por defecto, aunque la mayoría de las veces únicamente se utilizará en su definición `REQUIRED`:

```
auth_param ...
acl autenticacion proxy_auth REQUIRED
```

En este caso, cualquier petición con credenciales válidas verifica la ACL. En el caso que necesitemos un control más fino es cuando usaremos los nombres de usuario:

```
auth_param ...
acl autenticacion1 proxy_auth usuario1 usuario2
acl autenticacion2 proxy_auth usuario3 usuario4 usuario5
```

Anteriormente configuramos nuestro propio LDAP y ahora es un buen momento para utilizarlo⁹. La configuración del módulo de autenticación dentro de Squid quedaría:

```
auth_param basic program /usr/lib/squid/ldap_auth -b ou=personas,dc=
midominio,dc=org -D cn=Manager,dc=midominio,dc=org -W /etc/ldap.secret -
f uid=%s -v 3 192.168.0.50
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
```

⁸En versiones anteriores de Squid este módulo se llamaba `squid_ldap_auth`. Cuando ejecutamos `ldap_auth` sin ningún argumento nos muestra una ayuda en la que aún se hace referencia a este programa.

⁹Utilizaremos el árbol LDAP creado para la autenticación mediante módulos PAM, siendo el `uid` el atributo que contendrá el identificador de usuario.

El fichero `/etc/ldap.secret` contendrá la contraseña que se estableció para el usuario `Manager` al crear nuestro LDAP¹⁰. El resto de opciones pueden obtenerse de la ejecución de `ldap_auth` sin ningún argumento:

```
Usage: squid_ldap_auth -b basedn [options] [ldap_server_name[:port]]...
  -b basedn (REQUIRED)      base dn under which to search
  -f filter                  search filter to locate user DN
  -u userattr               username DN attribute
  -s base|one|sub           search scope
  -D binddn                 DN to bind as to perform searches
  -w bindpasswd             password for binddn
  -W secretfile             read password for binddn from file
                             secretfile
  -H URI                    LDAPURI (defaults to ldap://localhost)
  -h server                 LDAP server (defaults to localhost)
  -p port                   LDAP server port
  -P                        persistent LDAP connection
  -c timeout                connect timeout
  -t timelimit              search time limit
  -R                        do not follow referrals
  -a never|always|search|find
                             when to dereference aliases
  -v 2|3                   LDAP version
  -Z                        TLS encrypt the LDAP connection, requires
                             LDAP version 3
  If no search filter is specified, then the dn <userattr>=user,basedn
  will be used (same as specifying a search filter of '<userattr>=',
  but quicker as as there is no need to search for the user DN)
  If you need to bind as a user to perform searches then use the
  -D binddn -w bindpasswd or -D binddn -W secretfile options
```

Una vez modificada la configuración de Squid es necesario reiniciar el servicio para que los cambios tengan efecto con `/etc/init.d/squid restart`¹¹.

Existe una forma de comprobar la correcta configuración del módulo mediante la ejecución del mismo desde la línea de comandos:

```
root@guadalinux:~# /usr/lib/squid/ldap_auth -b ou=personas,dc=midominio,dc=
org -D cn=Manager,dc=midominio,dc=org -W /etc/ldap.secret -f uid=%s -v 3
192.168.0.50
jose.fernandez hola
OK
jose.fernandez holaa
ERR
```

En el ejemplo anterior estamos comprobando la configuración usando al usuario `jose.fernandez` con clave `hola`. En caso de introducir mal la clave o que el usuario no exista la salida será `ERR`, por el contrario, en una autenticación correcta obtenemos `OK`.

Ya tenemos correctamente configurado el módulo, pero aún no hemos dicho como queremos utilizarlo. Ahora entra en juego la definición de la ACL siguiendo las indicaciones previas:

```
acl ldap proxy_auth REQUIRED
...
http_access allow ldap
```

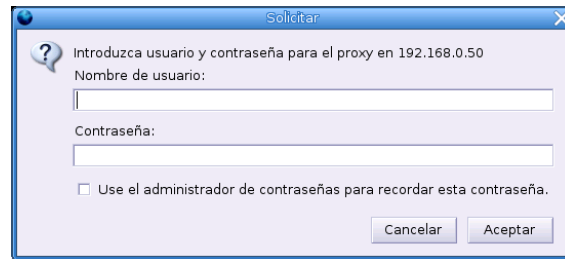
¹⁰Hay ocasiones en que la opción `-W` no funciona correctamente por lo que no obtendremos la clave almacenada en él. Bastará con sustituir este parámetro `-W <fichero_clave_manager>` por `-w <clave_manager>`.

¹¹Aunque el tiempo necesario es mínimo, si no queremos parar la instancia de Squid que se encuentra activa podemos utilizar `squid -k reconfigure` que actualiza la configuración sin parar Squid.



Con esta restricción cada vez que accedamos a una página de internet aparecerá la siguiente pantalla en la que deberemos introducir nuestro nombre de usuario y clave, según lo hayamos definido en LDAP.

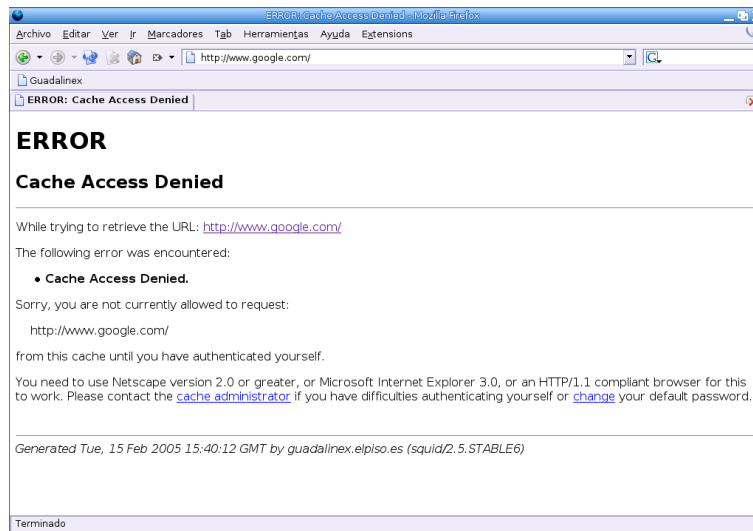
Figura 8.2: Autenticación de Squid



A partir de este momento, el proceso de autenticación se realizará únicamente para las nuevas instancias del navegador pero no para las que se creen a partir de la actual.

Cuando falle la autenticación se mostrará un mensaje de error informándonos al respecto.

Figura 8.3: Error en la autenticación



8.5.2. Analizador de logs SARG

El objetivo de restringir el acceso a internet es controlar que no se realice un uso ilegítimo del mismo. Mediante la configuración que acabamos de ver será necesario autenticarse antes de acceder a internet. Pero ¿cómo sabemos que el uso que se hace de este recurso es el correcto? Vamos ver qué información se está almacenando en los ficheros de log de Squid, más concretamente en `/var/log/squid/access.log`:

```
1108489147.817 471 192.168.0.50 TCP_MISS/200 3322 GET http://sarg.
sourceforge.net/sarg.README.txt hugo.santander DIRECT/66.35.250.209 text
/plain
```



Este log ya lo vimos anteriormente, pero ahora refleja un nuevo dato, el nombre de usuario que se autenticó. A partir de él podemos saber a qué direcciones han accedido los usuarios autenticados.

Podríamos generar a mano informes sobre el uso del acceso a internet a través de nuestro proxy. Sería necesario crear un script que formatease el fichero de log de Squid y generase un resumen. La herramienta Sarg nos permite obtener esta información.

Sarg es una herramienta que nos permite conocer las páginas que nuestros usuarios han visitado, así como otra información referente a la navegación de los mismos a través de Squid. Se obtiene con un paquete separado que puede descargarse de <http://sarg.sourceforge.net/sarg.php> o instalarse a partir de las utilidades que tienen Guadalinex y Fedora para instalar nuevos paquetes:

```
# apt-get install sarg
```

En el caso de Guadalinex los ficheros de configuración se sitúan en `/etc/squid`. El fichero de configuración más importante es `sarg.conf`.

```
# sarg.conf
#
# TAG: language
# Available languages:
# Bulgarian_windows1251
# Catalan
# Czech
# Dutch
# English
# French
# German
# Hungarian
# Indonesian
# Italian
# Japanese
# Latvian
# Polish
# Portuguese
# Romanian
# Russian_koi8
# Russian_windows1251
# Serbian
# Spanish
# Turkish
#
language English
# TAG: access_log file
# Where is the access.log file
# sarg -l file
#
#access_log /usr/local/squid/var/logs/access.log
access_log /var/log/squid/access.log
# TAG: title
# Specify the title for html page.
#
title "Squid User Access Reports"
# TAG: font_face
# Specify the font for html page.
#
font_face Arial
# TAG: header_color
# Specify the header color
#
```



```
header_color darkblue
# TAG: header_bgcolor
#     Especificy the header bgcolor
#
header_bgcolor blanchetalum
# TAG: font_size
# TAG: font_size
#     Especificy the font size
#
header_font_size -1
# TAG: background_color
# TAG: background_color
#     Html page background color
#
background_color white
# TAG: text_color
#     Html page text color
#
text_color black
# TAG: text_bgcolor
#     Html page text background color
#
text_bgcolor beige
# TAG: title_color
#     Html page title color
#
title_color green
# TAG: logo_image
#     Html page logo.
#
#logo_image none
# TAG: logo_text
#     Html page logo text.
#
#logo_text ""
# TAG: logo_text_color
#     Html page logo text color.
#
#logo_text_color black
# TAG: logo_image_size
#     Html page logo image size.
#     width height
#
#image_size 80 45
# TAG: background_image
#     Html page background image
#
#background_image none
# TAG: password
#     User password file used by authentication
#     If used here, reports will be generated only for that users.
#
#password none
# TAG: temporary_dir
#     Temporary directory name for work files
#     sarg -w dir
#
temporary_dir /tmp
```



```
# TAG: output_dir
# The reports will be saved in that directory
# sarg -o dir
#
#output_dir /var/www/html/squid-reports
output_dir /var/www/squid-reports
# TAG: output_email
# Email address to send the reports. If you use this tag, no html
# reports will be generated.
# sarg -e email
#
#output_email root@localhost
# TAG: resolve_ip yes/no
# Convert ip address to dns name
# sarg -n
resolve_ip
# TAG: user_ip yes/no
# Use Ip Address instead userid in reports.
# sarg -p
user_ip no
# TAG: topuser_sort_field field normal/reverse
# Sort field for the Topuser Report.
# Allowed fields: USER CONNECT BYTES TIME
#
topuser_sort_field BYTES reverse
# TAG: user_sort_field field normal/reverse
# Sort field for the User Report.
# Allowed fields: SITE CONNECT BYTES TIME
#
user_sort_field BYTES reverse
# TAG: exclude_users file
# users within the file will be excluded from reports.
# you can use indexonly to have only index.html file.
#
exclude_users /etc/squid/sarg.users
# TAG: exclude_hosts file
# Hosts, domains or subnets will be excluded from reports.
#
# Eg.: 192.168.10.10 - exclude ip address only
# 192.168.10.0 - exclude full C class
# s1.acme.foo - exclude hostname only
# acme.foo - exclude full domain name
#
exclude_hosts /etc/squid/sarg.hosts
# TAG: useragent_log file
# Put here where is useragent.log to nable useragent report.
#
#useragent_log none
# TAG: date_format
# Date format in reports: e (Europe=dd/mm/yy), u (USA=mm/dd/yy), w (
# Weekly=yy.ww)
date_format u
# TAG: per_user_limit file MB
# Save userid on file if download exceed n MB.
#
# This option can be used to disable user access if user exceed a
# download limit.
#per_user_limit none
```



```
# TAG: lastlog n
#   How many reports files must be kept in reports directory.
#   The oldest report file will be automatically removed.
#   0 - no limit.
#
lastlog 0
# TAG: remove_temp_files yes
#   Remove temporary files: geral, usuarios, top, periodo from root
#   report directory.
#
remove_temp_files yes
# TAG: index yes|no|only
#   Generate the main index.html.
#   only - generate only the main index.html
#
index yes
# TAG: overwrite_report yes|no
#   yes - if report date already exist then will be overwritten.
#   no - if report date already exist then will be renamed to filename.n
#   , filename.n+1
#
overwrite_report yes
# TAG: records_without_userid ignore|ip|everybody
#   What can I do with records without user id (no authentication) in
#   access.log file ?
#
#   ignore - This record will be ignored.
#   ip - Use ip address instead. (default)
#   everybody - Use "everybody" instead.
#
records_without_userid ip
# TAG: use_comma no|yes
#   Use comma instead point in reports.
#   Eg.: use_comma yes => 23,450,110
#   use_comma no => 23.450.110
#
use_comma yes
# TAG: mail_utility mail|mailx
#   Mail command to use to send reports via SMTP
#
mail_utility mailx
# TAG: topsites_num n
#   How many sites in topsites report.
#
topsites_num 100
# TAG: topsites_sort_order CONNECT|BYTES A|D
#   Sort for topsites report, where A=Ascendent, D=Descendent
#
topsites_sort_order CONNECT D
# TAG: index_sort_order A/D
#   Sort for index.html, where A=Ascendent, D=Descendent
#
index_sort_order D
# TAG: exclude_codes file
#   Ignore records with these codes. Eg.: NONE/400
#
exclude_codes /etc/squid/sarg.exclude_codes
# TAG: replace_index string
```



```
# Replace "index.html" in the main index file with this string
# If null "index.html" is used
#
#replace_index <?php echo str_replace(".", "_", $REMOTEADDR); echo ".html";
?>
# TAG: max_elapsed milliseconds
# If elapsed time is recorded in log is greater than max_elapsed use 0
for elapsed time.
# Use 0 for no checking
#
#max_elapsed 0
# 8 Hours
max_elapsed 28800000
# TAG: report_type type
# What kind of reports to generate.
# topsites - shows the site, connect and bytes
# sites_users - shows which users were accessing a site
# users_sites - shows sites accessed by the user
# date_time - shows the amount of bytes used by day and hour
# denied - shows all denied sites with full URL
# auth_failures - shows authentication failures
# site_user_time_date - shows sites, dates, times and bytes
#
# Eg.: report_type topsites denied
#
report_type topsites sites_users users_sites date_time denied auth_failures
site_user_time_date
# TAG: usertab filename
# You can change the "userid" or the "ip address" to be a real user
name on the rpeorts.
# Table syntax:
# userid name or ip address name
#
# Eg:
# SirIsaac Isaac Newton
# vinci Leonardo da Vinci
# 192.168.10.1 Karol Wojtyla
#
# Each line must be terminated with '\n'
#
usertab /etc/squid/sarg.usertab
# TAG: long_url yes|no
# If yes, the full url is showed in report.
# If no, only the site will be showed
#
# YES option generate very big sort files and reports.
#
long_url no
# TAG: date_time_by bytes|elap
# Date/Time reports will use bytes or elapsed time?
#
date_time_by bytes
# TAG: charset name
# ISO 8859 is a full series of 10 standardized multilingual single-byte
coded (8bit)
# graphic character sets for writing in alphabetic languages
# You can use the following charsets:
# Latin1 - West European
# Latin2 - East European
```



```
#          Latin3          - South European
#          Latin4          - North European
#          Cyrillic
#          Arabic
#          Greek
#          Hebrew
#          Latin5          - Turkish
#          Latin6
#          Windows-1251
#          Koi8-r
#
charset Latin1
# TAG: user_invalid_char "&/"
#     Records that contain invalid characters in userid will be ignored by
#     Sarg.
#
#user_invalid_char "&/"
# TAG: privacy yes|no
#     privacy_string "****.****.****.****"
#     privacy_string_color blue
#     In some countries the sysadm cannot see the visited sites by a
#     restrictive law.
#     Using privacy yes the visited url will be changes by privacy_string
#     and the link
#     will be removed from reports.
#
#privacy no
#privacy_string "****.****.****.****"
#privacy_string_color blue
# TAG: include_users "user1:user2:...:usern"
#     Reports will be generated only for listed users.
#
#include_users none
# TAG: exclude_string "string1:string2:...:stringn"
#     Records from access.log file that contain one of listed strings will
#     be ignored.
#
#exclude_string none
# TAG: show_successful_message yes|no
#     Shows "Successful report generated on dir" at end of process.
#
show_successful_message no
# TAG: show_read_statistics yes|no
#     Shows some reading statistics.
#
show_read_statistics no
# TAG: topuser_fields
#     Which fields must be in Topuser report.
#
topuser_fields NUM DATE.TIME USERID CONNECT BYTES %BYTES IN-CACHE-OUT
              USED_TIME MILLISEC %TIME TOTAL AVERAGE
# TAG: user_report_fields
#     Which fields must be in User report.
#
user_report_fields CONNECT BYTES %BYTES IN-CACHE-OUT USED_TIME MILLISEC %TIME
                  TOTAL AVERAGE
# TAG: topuser_num n
#     How many users in topsites report. 0 = no limit
```




```
#
topuser_num 0
# TAG: site_user_time_date_type list|table
#     generate reports for site_user_time_date in list or table format
#
site_user_time_date_type table
# TAG: datafile file
#     Save the report results in a file to populate some database
#
#datafile none
# TAG: datafile_delimiter ";"
#     ascii character to use as a field separator in datafile
#
#datafile_delimiter ";"
# TAG: datafile_fields all
#     Which data fields must be in datafile
#     user;date;time;url;connect;bytes;in_cache;out_cache;elapsed
#
#datafile_fields user;date;time;url;connect;bytes;in_cache;out_cache;elapsed
# TAG: weekdays
#     The weekdays to take account ( Sunday->0, Saturday->6 )
# Example:
#weekdays 1-3,5
# Default:
#weekdays 0-6
# TAG: hours
#     The hours to take account
# Example:
#hours 7-12,14,16,18-20
# Default:
#hours 0-23
# TAG: squidguard_log_path file
#     Generate reports from SquidGuard logs.
#
#squidguard_log_path none
# TAG: show_sarg_info yes|no
#     shows sarg information and site patch on each report bottom
#
#show_sarg_info yes
# TAG: parsed_output_log directory
#     Saves the processed log in a sarg format after parsing the squid log
#     file.
#     This is a way to dump all of the data structures out, after parsing
#     from
#     the logs (presumably this data will be much smaller than the log
#     files themselves),
#     and pull them back in for later processing and merging with data from
#     previous logs.
#
#parsed_output_log none
# TAG parsed_output_log_compress /bin/gzip//usr/bin/bzip2|nocompress
#     sarg logs compress util
#
#parsed_output_log_compress /bin/gzip
# TAG displayed_values bytes|abbreviation
#     how the values will be displayed in reports.
#     eg. bytes      - 209.526
#     abbreviation - 210K
```



```
#
displayed_values bytes
```

No entraremos a describir los parámetros de configuración de Sarg ya que el fichero de configuración generado por defecto es suficiente para comenzar a utilizar la herramienta. Únicamente indicar que algunos de estos parámetros pueden ser definidos también en tiempo de ejecución. De esta manera cuando ejecutemos Sarg podemos modificar el comportamiento definido en `sarg.conf` a partir de parámetros de ejecución. Para más información puede consultarse la página del manual referente a sarg (`man sarg`).

La configuración por defecto establece la localización de los ficheros de log de Squid así como el lugar donde queremos almacenar los informes generados.

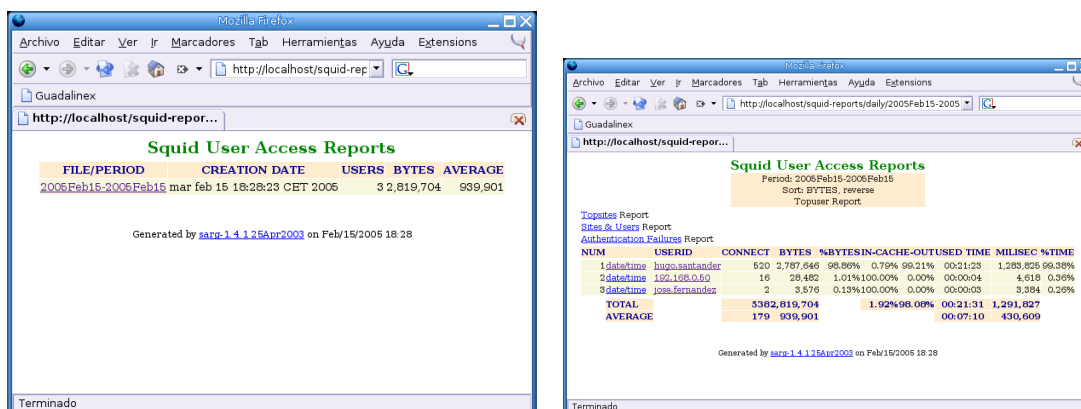
```
# TAG: access_log file
#       Where is the access.log file
#       sarg -l file
#
access_log /var/log/squid/access.log
....
# TAG: output_dir
#       The reports will be saved in that directory
#       sarg -o dir
#
output_dir /var/www/squid-reports
```

Estos parámetros de configuración pueden modificarse desde línea de comandos:

```
root@guadalinux:/etc/squid# /usr/bin/sarg -l /var/log/squid/access.log -o /
var/www/squid-reports/daily
```

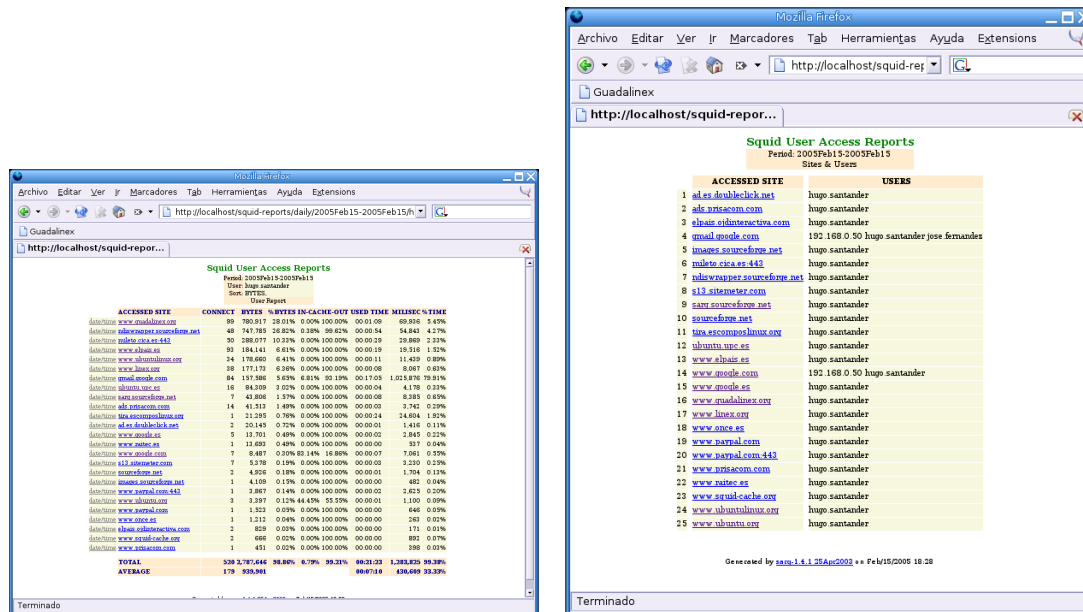
¿Y eso es todo? Pues vamos a ver que informes se han generado:

Figura 8.4: Página principal y listado de informes disponible



Puede obtenerse aún más detalle, mostrando un informe detallado de los sitios que ha accedido cada usuario o un listado de sitios con los usuarios que han accedido a los mismos:

Figura 8.5: Informes por usuario y por sitio visitado



8.6. ➔ Para practicar

8.6.1. Castellanizar los errores de Squid

1.

- Instalar y configurar squid en el equipo local para que no permita el acceso a más máquinas que la nuestra y que además limite el acceso a todas las páginas en cuya URL aparezcan las cadenas: `apa` y `sex`.
- Castellanizar los errores: sólo hay que modificar el enlace simbólico `/etc/squid/errors` para que apunte a `/usr/share/squid/errors/Spanish`. Después hacer que el demonio relea la configuración.

```
# ll /etc/squid/errors
```

```
lrwxrwxrwx 1 root root 31 nov 9 14:49 errors ->/usr/share/squid/errors/Spanish
```

- Comprobar que funciona configurando mozilla tal cual aparece en los apuntes.

8.6.2. Limitar ancho de banda para determinadas extensiones

Se trata de optar porque nuestros clientes no consuman todo el ancho de banda¹² disponible para bajarse determinados programas. Para eso se usan las *delay pools*. Tendremos que añadir al final (casi) del fichero de configuración de squid una serie de líneas. Las directrices que vamos a usar se pueden resumir en:

- Un par de ACL que nos van a permitir:

- Definir el ancho de banda para nuestra máquina
- Limitar el ancho de banda a partir de la extensión al resto de ordenadores

¹²Para conocer más sobre este tema Limitar el ancho de banda COMO <http://mural.uv.es/~joferna/doc/Limitar-ancho-de-banda-COMO/html/>

Podemos optar por no poner la primera regla y limitar también a nuestro ordenador.

```
acl regla_primera src 127.0.0.1/255.255.255.255
acl regla_segunda url_regex -i .exe .mp3 .zip .avi .mpeg .rar
```

- Definimos el nº de reglas de demora

```
delay_pools 2
```

- Podemos definir tres tipos de *delay pools*, pero sólo vamos a trabajar con el segundo tipo, eso lo indicamos con

```
delay_class número tipo
```

Cada tipo permite definirle una serie de parámetros, a las del tipo dos

```
delay_parameters número global máquina
```

donde lo único a comentar es el significado de *global* y *máquina*. Ambos de la forma *caudal_bytes/máximo_bytes* con el significado:

- *caudal_bytes* es el número de bytes por segundo de tasa de transferencia mantenida una vez que la descarga sobrepasa *máximo_bytes*¹³. Si establecemos los valores *-1/-1* el significado es sin límite.
- El primer par de números (*global*) establece los valores para toda la red, mientras que con *máquina* lo hacemos para una IP concreta. Por ejemplo con

```
delay_parameters 2 10000/20000 5000/15000
```

establecemos:

10000/20000 una vez que los archivos sean mayores de 20000 bytes, las descargas (para toda la red) usan como máximo 10000 bytes

5000/15000 para cada IP concreta, si el archivo sobrepasa los 15000 bytes, proseguirá la descarga a 5000 bytes por segundo

Con *delay_access* establecemos qué *delay pools* gestiona la petición.

- Todo junto quedaría

```
#Listas ACL y número de delay pools
acl regla_primera src 127.0.0.1/255.255.255.255
acl regla_segunda url_regex -i .exe .mp3 .zip .avi .mpeg .rar
delay_pools 2

#Pool 1 de tipo 2
#Permitimos que desde nuestra máquina no haya límite de bajada
delay_class 1 2
delay_parameters 1 -1/-1 -1/-1
delay_access 1 allow regla_primera

#Pool 2 de tipo 2
#Al resto de máquinas le limitamos la bajada de ficheros extensión
#exe .mp3 .zip .avi .mpeg .rar .vqf
#Establecemos un límite para toda la red a 10Kb/s
```

¹³Máximo guardado para toda la red o máquina en particular.

```
#pero cada máquina con un límite de 5Kb/s
delay_class 2 2
delay_parameters 2 10000/15000 5000/15000
delay_access 2 allow regla_segunda
```

8.6.3. Proxy transparente

Si deseamos disponer de un servicio de proxy en el que los clientes no tengan que modificar nada en la configuración del navegador necesitamos montar un proxy transparente (como el de Telefónica). En este caso, los usuarios no necesitan modificar nada en la configuración del navegador de sus equipos.

- Máquina servidor: Para disponer de esta funcionalidad descomentaremos en el fichero de configuración de squid las líneas¹⁴

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Además, hay que configurar bien las reglas de filtrado de paquetes (supondremos que el interfaz de red conectado a Internet es `eth1` y el de la red local `eth0`). Un posible script de configuración puede ser

```
echo 1 >/proc/sys/net/ipv4/ip_forward
iptables --flush
iptables --table nat --flush
#Activamos el NAT con enmascaramiento
iptables --table nat --append POSTROUTING -s 172.26.0.0/24 --out-interface eth1 -j MASQUERADE
iptables --append FORWARD -s 172.26.0.0/24 --in-interface eth0 -j ACCEPT
# Hacer que squid responda llamadas http
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 3128
#Política para la red local de todo permitido
iptables -A INPUT -s 172.26.0.0/24 -j ACCEPT
```

- Máquinas clientes: Es necesario poner la puerta de enlace y las IPs de los servidores de nombres (si optamos por usar un servicio de DHCP se puede hacer automáticamente). Pero ya no hay que optar por configurar nada en los navegadores.

8.7. DansGuardian

Si queremos proteger a los menores del contenido peligros que existe en Internet o la política de nuestra organización restringe la visita a determinados sitios, tenemos dos opciones. Una es aplicar reglas en squid para prohibir determinados sitios¹⁵, pero esta opción se convertirá en una tarea draconiana. O sólo permitimos el acceso a unos pocos sitios, o la lista de los sitios prohibidos será siempre una lista inacabada. Una mejor solución es analizar el contenido de las páginas que se sirven y si no cumplen con nuestra norma de lo que es un sitio seguro, no dejarla pasar. Esta es la tarea de un filtro de contenidos, y DansGuardian es uno de ellos.

¹⁴Funciona pero sin https ni ftp.

¹⁵Como playboy.com, por ejemplo.

8.7.1. Funcionamiento

DansGuardian funcionará entre el usuario y el servidor proxy-cache (por defecto squid). El usuario realizará las peticiones a dansGuardian que se las envía al proxy si no están filtradas y a las devueltas por el proxy, les analizará el contenido para aplicar de nuevo las restricciones.

Es por esta manera de funcionar, que si queremos un filtrado de contenidos fiable debemos configurar las acl de squid para que sólo permita conexiones desde dansguardian, por ejemplo limitando a la misma máquina, si ambos se ejecutan en ella.

Para realizar la configuración inicial modificaremos lo necesario en `dansguardian.conf` y `dansguardianfx.conf` donde `x` será el grupo de filtrado.

DansGuardian revisa primero las listas de excepciones, después comprobará las listas grises y por último las banned.

Comprueba si el usuario y la máquina están permitidos y el filtro a aplicar. Comprueba el dominio, la url y posteriormente el contenido. A la hora de filtrar el contenido con las palabras clave, podremos cambiar esas expresiones o podremos bloquear la página. Para bloquear la página a la hora de definir las listas de palabras se les añade un peso, de tal forma que si esa cadena de texto aparece, se suma ese peso. Si la palabra pertenece a la lista de excepciones se le quita ese peso de forma que al final del análisis de la página obtendremos un peso total que no debe sobrepasar el límite que se le haya fijado. El límite variará según el tipo de usuario y se establece por grupo de filtrado (por ejemplo para los niños se recomienda un límite de 50).

Siempre tendremos que tener en cuenta el compromiso entre el filtrado de contenidos y el tiempo de proceso para la página.

Esta herramienta permite un filtrado eficiente pero para ello tendremos que probar, ajustar y personalizar la configuración hasta obtener los resultados requeridos.

8.7.2. Instalación

Para la instalación de dansguardian, según la distribución que estemos utilizando se realizará de forma inmediata.

Mediante :

```
#apt-get install dansguardian
```

o bien

```
# rpm -Uhv dansguardian-version-sistema.rpm
```

A la hora de instalar dansguardian, debemos tener en cuenta que para un funcionamiento correcto tendremos que tener instalado squid como proxy-cache.

Para iniciar dansguardian lo haremos como otro servicio más mediante el script

```
#/etc/init.d/dansguardian [start/stop]
```

Por defecto se inicia la ejecución en el puerto 8080 y con el puerto 3128 de squid.

8.8. Configuración

En el directorio de configuración `/etc/dansguardian` nos encontramos los diferentes ficheros que permiten afinar la configuración inicial.

En el fichero `dansguardian.conf` encontramos los siguiente parámetros más importantes:

`reportinglevel=3` \mapsto el valor por defecto es el más completo

`languagedir = '/etc/dansguardian/languages'` \mapsto directorio donde se sitúan las diferentes plantillas para los lenguajes

`language = 'mxspanish'` \mapsto nombre del fichero de idioma

`logfileformat = 3` \mapsto nos puede interesar que esté en formato de squid para unificar las estadísticas

`filterip =` \mapsto dirección ip en la que escucha

`filterport = 8080` \mapsto puerto en el que escucha

`proxyip = 127.0.0.1` \mapsto dirección de la máquina donde esté squid

`proxyport = 3128` \mapsto puerto en el que escucha el proxy

`urlcachenumber = 1000` y `urlcacheage = 900` \mapsto para caché de páginas permitidas

`preservecase = 0` \mapsto para no distinguir mayúsculas y minúsculas, todo se pasa a minúsculas antes de analizarlo

El resto de parámetros hacen referencia a datos del proceso y al resto de ficheros de configuración.

En el fichero `dansguardianf1.conf` se definen el resto de ficheros, el peso de la página a filtrar y la posibilidad de definir una palabra de paso para desabilitar el filtrado temporalmente.

A continuación, describimos el resto de ficheros que nos van a permitir realizar el filtrado y control de contenidos. Tendremos que ir definiendo y afinando los diferentes tipos de filtrado teniendo en cuenta el retraso que puede introducirse en la navegación web del usuario. Es por esto, que tendremos que llegar a un compromiso del tipo de filtrado y la rapidez del mismo.

`bannedextensionlist` \mapsto extensiones de ficheros no deseados

`bannediplist` \mapsto ip de los clientes para denegarles el acceso

`bannedmimetypelist` \mapsto tipos de contenidos no permitidos

`bannedphraselist` \mapsto palabras o frases para filtrar o ficheros donde se hayan definido previamente dichas palabras mediante la directiva `.Include`, podemos encontrar ejemplos en `/etc/dansguardian/phraselists`

`bannedregexprlist` \mapsto expresiones que queremos filtrar cuando aparecen en la url

`bannedsitelist` \mapsto podemos incluir los dominios para filtrar, usar la directiva `.Include`, filtrar todo y permitir sólo las excepciones o forzar a que se filtre por ip

`bannedurllist` \mapsto bloquean una parte de un dominio mediante la definición de su url

`banneduserlist` \mapsto usuarios de la autenticación del proxy a los que se quiere bloquear el acceso

`contentregexplist` \mapsto mediante el formato “badword”->”expresion” podemos sustituir determinadas expresiones por otras

`filtergroupslist` \mapsto permiten asociar a usuarios grupos de filtros

`greysitelist` \mapsto similares a las “banned” pero las sobrescriben

`greyurllist` \mapsto similares a las “banned” pero las sobrescriben

`exceptioniplist`, `exceptionphraselist`, `exceptionsitelist`, `exceptionurllist`, `exceptionuserlist` \mapsto En los ficheros “exception” usando la misma forma de definición y la directiva `.Include` podremos definir los elementos que queremos asegurarnos que no sean filtrados.

`weightedphraselist` \mapsto instrucciones para definir la lista de filtrado `<palabra><peso>`

A partir de aquí debemos comenzar a utilizar las múltiples opciones de filtrado hasta obtener un resultado óptimo.

Para facilitar la gestión, existe un módulo para versiones superiores de 2.4.x para la herramienta webmin.

Si configuramos nuestro navegador para que acceda al filtro de contenido (por ejemplo en la máquina `guadalinux.midominio.org`, en el puerto 8080), e intentamos acceder a una página no recomendada a menores, obtenemos lo siguiente:



ACCESO DENEGADO -

El acceso a la página:

<http://www.playboy.com>

... ha sido denegado por la siguiente razón:

ICRA languagesexual Las etiquetas del sitio exceden el nivel PICS.

Usted esta viendo esta página de error porque el sitio que está tratando de ver o su contenido han sido catalogados como inapropiados.

Si requiere acceso a esta página por favor pongase en contacto con el Administrador de Sistemas o el Administrador de la Red.

Powered by [DansGuardian](#)

Existe también un conjunto de herramientas de testeo y ampliación (antivirus, x-forwarded,...) Para profundizar más, podéis visitar <http://dansguardian.org>

Bibliografía

- [1] Squid: The Definitive Guide. Duane Wessels. Editorial O'Reilly
- [2] Sarg: Squid Analysis Report Generator. <http://sarg.sourceforge.net/>
- [3] Guía de Administración de Redes con Linux. OLAF KIRCH Y TERRY DAWSON. Proyecto LuCAS, traducción al español. <http://es.tldp.org/Manuales-LuCAS/GARL2/gar12>
- [4] Introduction to Linux A Hands on Guide. MACHTELT GARRELS. <http://tille.soti.org/training/tldp>
- [5] TCP/IP Network Administration, 3rd Edition. CRAIG HUNT
- [6] SSH, The Secure Shell: The Definitive Guide. by Daniel J. Barrett and Richard E. Silverman
- [7] Managing NFS and NIS, 2nd Edition By Hal Stern, Mike Eisler and Ricardo Labiaga
- [8] DNS and Bind, 4th Edition By Paul Albitz and Cricket Liu
- [9] Usando SAMBA, ROBERT ECKSTEIN, DAVID COLLIER-BROWN, PETER KELLY