TEMA 4

Contenido

1.- Servicio de transferencia de ficheros. 2
1.1.- ¿Cómo funciona? 3
1.2.- Cliente FTP. 4
1.3.- Tipos de usuarios.....5
1.4.- Modos de conexión del cliente. 6
1.5.- Tipos de transferencia de archivos. 7
1.6.- Establecer permisos en ftp. 8
1.7.- Servicio de transferencia de archivos en modo texto. 9
1.7.1.- Comandos ftp. 11
1.8.- Servicio de transferencia de archivos en modo gráfico. 12
1.9.- Servicio de transferencia de archivos desde el navegador. 14
1.10.- Asegurando el servicio de transferencia de archivos. 15
1.11.- El servicio de transferencia de archivos en el proceso de despliegue de una aplicación Web. 16
2.- Instalación del servidor proftpd. 18
2.1.- Configuración de proftpd. 19
2.2.- Configurar el servidor como ftp privado. .21
2.3.- Configurar el servidor como ftp privado y anónimo. 21
2.4.- Configurar el servidor como ftp anónimo. 22
2.5.- Configurar el servidor ftp con múltiples dominios. 23
2.6.- Virtualhosts basados en nombre. 24
2.7.- Virtualhosts basados en IP. 25
2.8.- Cuotas de disco para los usuarios (I). 26
2.8.1.- Cuotas de disco para los usuarios (II). 27
2.9.- Acceso seguro mediante TLS. 28
Anexo I - PAM. 33
¿Qué es PAM? 33
Grupos de gestión. 33
Arquitectura 34
Configuración 35
Enfoques de la organización de la configuración. 35
Reglas 35
Servicio. 35
Tipo 36
Control 36
Ruta. 37
Argumentos 37
Algunos módulos disponibles 37
pam console.so. 37
Grupo 37
pam cracklib.so. 38
Grupo 38
pam deny.so 38
Grupo 38
pam env.so. 38
Grupo 38
pam limits.so 39
Grupo 39
pam nologin.so 39
Grupo 39
pam permit.so 39
Grupo 39
pam rootok.so 40
Grupo 40
pam securetty.so 40
Grupo 40
pam stack.so 40
Grupo 40
pam wheel.so 40
Grupo 40
pam xauth.so 40
Grupo 40
Ejemplos de configuración 40
login. 41
passwd 42
su. 42
other 43
Valores devueltos por los módulos de PAM. 43
authentication 43
account 44
password. 44
session. 44
Anexo II - proftpd.conf. 45
Anexo III - tls.conf 48
Anexo IV - tls2.conf. 49

Instalación y administración de servidores FTP

Caso práctico

La empresa BK Programación se ha dedicado últimamente a la creación de entornos web en servidores dedicados y compartidos. Siempre se ha decantado para la instalación, administración y configuración de servidores de software libre. Y es por ello que varias empresas se han puesto en contacto con BK Programación para contratarles sus servicios. Una de estas empresas con las que trabaja BK Programación tuvo un problema subiendo un archivo a su dominio (Nombre por el cual se reconoce a un grupo de dispositivos o equipos conectados a la red. Éstos pueden ser nombres locales, no existentes enInternet, pero, en general, son utilizados para su uso en Internet, por ejemplo: debian.org), por lo cual se ha recibido una llamada dirigida a soporte técnico en BK Programación. En ésta se mantuvo la siguiente conversación:

- ✓ *Hola, buenos días. BK Programación, ¿en qué puedo ayudarle?*
- ✓ *Hola, buenos días. Tengo un problema en el servidor de nuestra página, no puedo subir un archivo de vídeo.*
- ✓ *¿No le aparece la opción de subida o, en medio del proceso, se le corta la conexión?*
- ✓ *Sí, soy capaz de empezar a subir el archivo pero, pasado un tiempo, el proceso se corta.*
- ✓ *¿Cuánto pesa el archivo? Es decir, ¿qué tamaño posee?*
- ✓ *No sé, espere un momento... —pasado un tiempo—, sí..., mire, el archivo ocupa 300 MB.*
- ✓ *Vale, parece claro, tal como está procediendo hasta ahora no podrá subir el archivo, debido a la limitación establecida en el servidor web para la subida de archivos, por lo tanto debe utilizar su cuenta ftp para la transferencia de archivos.*
- ✓ *Y eso, ¿cómo procedo?¿está estipulado en el contrato?*
- ✓ *Sí, no se preocupe. El contrato estándar ya establece una cuenta ftp por dominio, pero eso sí, dependiendo del contrato poseerá una cuota de disco u otra. En cuanto al método para proceder, aparece explicado en nuestra página web, paso por paso, en la documentación que podrá encontrar en la pestaña descargas.*
- ✓ *Pues, poseo el contrato estándar.*
- ✓ *Bien, entonces posee una cuota de 2 GB.*
- ✓ *Vale, gracias, entonces ¿cuándo podré contar con la cuenta ftp para subir el archivo?*
- ✓ *Ya la tiene operativa, solamente debe seguir los pasos del documento que le he comentado. Si tiene cualquier problema no dude en contactar de nuevo con nosotros.*
- ✓ *De acuerdo, muchas gracias. Hasta luego.*
- ✓ *Hasta luego.*

Este tipo de incidencias son típicas en BK Programación, por lo cual Ada, la directora de BK Programación ha establecido un protocolo de actuación según el tipo de incidencia. Las incidencias primero serán atendidas por una unidad de servicio telefónico, en caso de que no se encuentre la solución dentro de las posibles será escalada a su correspondiente área técnica, así las incidencias de servidores serán escaladas a María, las de programación a Juan y éstos derivarán la incidencia a un técnico de la empresa. En el caso de la incidencia por llamada telefónica anterior la solución ya existía dentro de las posibles con lo cual la incidencia no fue escalada.

Las posibles soluciones fueron proporcionadas por el personal responsable para cada área. Así como la incidencia que hablamos era sobre servidores fue propuesta la solución por María. Además, en este caso, María había estudiado varios servidores ftp y se decantó por la versatilidad, funcionalidad y seguridad del servidor ftp ProFTPD.

1.- Servicio de transferencia de ficheros.

Caso práctico

María, sabía que, llegado el momento, las empresas a las que darían soporte web necesitarían subir archivos a sus dominios, por lo cual necesitarían una alternativa a la aplicación web destinada para tal fin: un servicio ftp. Así realizó un estudio sobre servidores ftp y se decantó por la versatilidad, funcionalidad y seguridad del servidor ftp ***ProFTPD****. En ese estudio quería llegar a saber del servidor ftp lo siguiente:*

1. *¿Cómo funciona?*
2. *Posibilidades de autenticación y control de acceso.*
3. *Seguridad. ¿Es posible cifrar la transferencia de archivos?*
4. *¿Permite cuotas de disco?*
5. *¿Permite cuotas de subida y bajada de archivos?*
6. *¿Qué clientes ftp soporta?*

Pero antes de ponerlo en producción necesitaba probarlo, es por eso que construyó el siguiente escenario de pruebas, similar al escenario de producción para una empresa que ofrezca sus servicios web por medio de la infraestructura proporcionada por BK Programación y totalmente transparente al cliente final:

- ✓ *Sistema Operativo: Debian GNU/Linux 6.0*
- ✓ *Servidor FTP: ProFTPD*
- ✓ *Configuración de Red:*
  - ➔ *Servidor FTP: 192.168.200.250*
  - ➔ *Cliente de pruebas, desde donde se lanza el cliente ftp: 192.168.200.100*

Hoy en día encontramos muchísima información en Internet, es más, en muchas ocasiones cuando buscamos una determinada información es muy probable que tengamos que filtrarla, ya que encontramos demasiada. Pero, una vez encontrada ¿la podemos guardar? ¿y descargar? Y si es así ¿cómo fue subida?

Normalmente para subir archivos en Internet, ya sean de texto, imágenes, vídeo... hubo que emplear algún método de transferencia de archivos para ubicarlos.

Uno de los métodos más empleados como servicio de transferencia de archivos se realiza mediante el servicio ftp. Éste utiliza el protocolo FTP empleando la arquitectura cliente-servidor. Así el servidor ftp esperará peticiones para transferir los archivos y el cliente ftp, ya sea por terminal o de modo gráfico, realizará esas peticiones.

Uno de los principales problemas, a pesar de ser uno de los métodos más utilizados del protocolo FTP es la no seguridad de la información, esto es, la transferencia tiene lugar sin cifrar la información transferida. Este no sólo es un problema del protocolo FTP sino de muchos de los protocolos utilizados enInternet, puesto que en el comienzo de Internet no se preveía su expansión actual y no se pensaba en asegurar la información mediante cifrado, sino simplemente asegurar el buen funcionamiento. Hoy en día existen extensiones sobre el protocolo FTP que aseguran el cifrado en la transferencia, como FTPS, empleando el cifrado SSL/TLS (*protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet*).

No confundir FTPS con SFTP, ya que este último es implementado con otro servicio, el servicio SSH, y es utilizado para conexiones remotas seguras a través de un terminal de comandos.

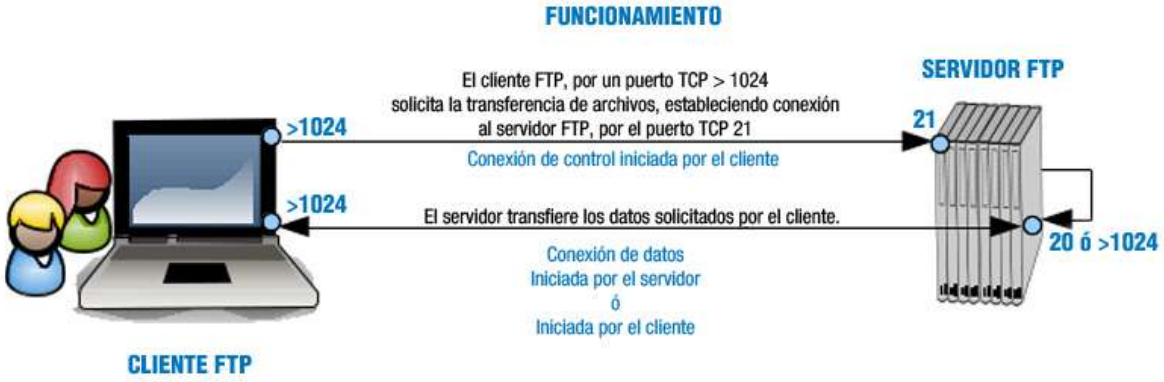
En el siguiente enlace, página web del RFC(*serie de documentos en los que se detalla prácticamente todo lo relacionado con la tecnología de la que se sirve Internet: protocolos, recomendaciones, comunicaciones…*) 959 sobre FTP, puedes encontrar traducido el estándar RFC sobre FTP.

http://www.rfc-es.org/rfc/rfc0959-es.txt

En el siguiente enlace, página web del RFC 4251 sobre SSH, puedes encontrar el estándar RFC sobre SSH.

http://www.ietf.org/rfc/rfc4251.txt

1.1.- ¿Cómo funciona?



El protocolo FTP emplea una arquitectura cliente/servidor, siendo el cliente ftp quien solicita la transferencia de archivos y el servidor ftp quien ofrece los archivos. Pertenece a la familia de protocolos de red TCP (*uno de los protocolos fundamentales en Internet. Garantiza que los datos serán entregados en su destino sin errores y, una vez recogidos, puestos en el mismo orden en que se transmitieron*) y por lo tanto es un protocolo orientado a conexión, esto es, el cliente ftp necesita establecer una conexión con el servidor para empezar la transferencia de ficheros. Si no se establece la conexión ésta no tiene lugar.

Puesto que FTP es un protocolo que no utiliza una autenticación de usuarios y contraseña cifrada, se considera un protocolo inseguro y no se debería utilizar a menos que sea absolutamente necesario. Verás que existen otras alternativas al FTP, como por ejemplo el protocolo FTPS, para mantener comunicaciones cifradas. Aún así, el protocolo FTP está muy extendido en Internet ya que a menudo los usuarios necesitan transferir archivos entre máquinas sin importar la seguridad.

El protocolo FTP requiere de dos puertos (*números utilizados en las comunicaciones cliente/servidor, en transmisiones TCP o UDP comprendidos entre 1 y 65535, que indican por donde tiene lugar la conexión con un servidor. Están estandarizados, esto es, un servidor suele estar activo siempre por definición en un puerto determinado, pero éste puede que sea modificado en la configuración del servidor. Por ejemplo un servidor web suele esperar en el puerto TCP 80*) TCP en el servidor para su funcionamiento, a diferencia de la mayoría de los protocolos utilizados en Internet que solamente requieren un puerto en el servidor. Un puerto es necesario para establecer el control de la conexión y otro se utiliza para el control de la transmisión, es decir, un puerto se utiliza para establecer la conexión entre el cliente y el servidor y otro para la transferencia de datos.

Los puertos TCP del servidor en cuestión, suelen ser el 21 para el control de la conexión y otro a determinar según el modo de conexión: podría ser el 20 o incluso uno mayor de 1024. Hay que tener en cuenta que estos puertos pueden ser modificados en la configuración del servidor, así no es obligatorio que los puertos 21 y 20 sean los asignados al servidor FTP, pero sí son los que éste maneja por defecto. El puerto 21 también es conocido como puerto de comandos y el puerto 20 como puerto de datos.

La ventaja que supone utilizar el protocolo FTP se basa en su alto rendimiento y sencillez, que lo hacen una opción conveniente para la transferencia de archivos a través de Internet.

A través de un cliente ftp descargas un archivo a tu equipo desde el servidor ftp: ¿cuáles de las siguientes afirmaciones son correctas teniendo en cuenta que el archivo puede descargarse sin problemas?

- [x] **El servidor ftp posee dos puertos TCP: uno para el control de la transmisión y otro para la transferencia de datos.**

- ☐ El servidor ftp posee siempre los puertos TCP 21 y 20: el 21 para el control de la transmisión y el 20 para la transferencia de datos.
- ☑ **El servidor ftp posee los puertos TCP 21 y 20: el 21 para el control de la transmisión y el 20 para la transferencia de datos.**
- ☐ El servidor ftp, configurado por defecto, posee el puerto TCP 21 válido para el control de la transmisión y para la transferencia de datos.

Siempre deben existir dos puertos TCP, uno para el control de la transmisión y otro para la transferencia de datos, éstos en un servidor ftpsuelen ser el 21 y el 20 respectivamente, pero pueden ser modificados.

1.2.- Cliente FTP.

Lo importante es no dejar de hacerse preguntas.

Albert Einstein

Para poder establecer una conexión con el protocolo FTP son necesarias dos partes: un servidor y un cliente.

Existen múltiples tipos de clientes ftp, desde clientes en terminal de comandos, como `ftp` o `lftp`, clientes gráficos como `gftp` o `FileZilla`, hasta un cliente ftp en los navegadores mediante `ftp://`
¿Cuál elegir? Pues, como todo, depende:

- ✓ ¿Conoces la consola ftp? Si te manejas con soltura en la consola ftp puedes pensar en un cliente ftp de comandos que permita utilizar la tecla "tabulador" después de escribir unos caracteres para complementar los nombres de archivos.
- ✓ ¿Cuál es el uso que necesitas? ¿Para qué lo vas a utilizar? A lo mejor solamente quieres visitar un servidor ftp y descargar un archivo sin tener que andar instalando nuevos programas. En este caso puedes utilizar el cliente ftp del navegador, `ftp://`
- ✓ ¿Quieres reanudar la conexión en caso de corte en la misma? En este caso mejor un cliente tipo gráfico.
- ✓ ¿Deseas facilidad de manejo? Un cliente terminal de comandos suele ser menos interactivo que uno gráfico, debes saber manejarte con comandos en la consola ftp, mientras que en un cliente gráfico puedes manejarte a través de clics del ratón. Los clientes gráficos suelen ser más amigables y por lo tanto más utilizados.
- ✓ ¿Qué tipo de conexión quieres establecer? ¿cifrada? ¿no cifrada? Dependiendo del tipo de conexión debes emplear un cliente u otro, ya que no todos los clientes ftp permiten conexiones cifradas.
- ✓ ¿Deseas recordar conexiones (sitios *(Plantilla de configuración para recordar perfiles de configuración a servidores FTP)*)? Pues lo mismo, no todos los clientes ftp lo permiten.

Un cliente ftp muy recomendable es el cliente gráfico ftp FileZilla, ya que posee las siguientes características:

- ✓ Fácil de usar.
- ✓ Soporta FTP, FTP sobre SSL/TLS (FTPS) y SFTP.
- ✓ Compatibilidad con múltiples plataformas: se ejecuta en Windows, Linux, BSD, Mac OS X y más.
- ✓ Soporte Ipv6.
- ✓ Disponible en varios idiomas.
- ✓ Soporta y reanuda la transferencia de archivos de gran tamaño (mayores de 4 GB).
- ✓ Interfaz de usuario con pestañas.
- ✓ Potente administrador de sitios y cola de transferencia.
- ✓ Marcadores.
- ✓ Arrastrar y soltar.
- ✓ Permite configurar límites de velocidad de transferencia.
- ✓ Nombre de filtros.
- ✓ Directorio de comparación.

- ✓ Asistente de configuración de la red.
- ✓ Edición de archivos remoto.
- ✓ Automantenimiento de la conexión.
- ✓ HTTP(*protocolo usado en cada transacción de la World Wide Web*)/1.1, SOCKS5 y soporte de FTP-Proxy.
- ✓ Fichero de registro.
- ✓ Sincronización de directorios de navegación.
- ✓ Búsqueda de archivos remoto.

En el siguiente enlace puedes acceder a la página web oficial de FileZilla donde puedes descargarlo y encontrar documentación sobre el mismo.
http://filezilla-project.org/

1.3.- Tipos de usuarios.

¿Qué usuarios se pueden conectar al servidor ftp? ¿cualquiera? ¿sólo los usuarios del sistema? Bien, típicamente existen dos tipos de usuarios:

- ✓ **Usuarios anónimos**: usuarios que tienen acceso y permisos limitados por el sistema de archivos. Al conectarse al servidor FTP sólo deben introducir una contraseña simbólica, normalmente cualquier dirección de correo -real o ficticia-, por ejemplo: `a@` .
- ✓ **Usuarios del sistema**: aquellos que disponen de una cuenta en la máquina que ofrece el servicio FTP. Al conectarse al servidor FTP deben introducir su contraseña de sistema.

Pero en ciertos servidores, como el servidor ProFTPD, existe una tercera posibilidad muy interesante: **usuarios virtuales**. Los usuarios virtuales poseen acceso y permisos al servidor FTP sin necesidad de ser usuarios del sistema, por lo tanto si un usuario virtual quisiera acceder al sistema operativo como si fuese un usuario del sistema, ya sea de forma local o remota no podría, pues su cuenta de usuario no existe en el sistema. Los usuarios virtuales tienen definida una contraseña propia y pueden estar definidos en ficheros de autenticación (de texto) con el mismo formato que los del sistema operativo GNU/Linux `/etc/passwd`, directorios `LDAP` (*protocolo de acceso unificado a un conjunto de información sobre una red*), bases de datos `SQL` (*lenguaje de consulta estructurado es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre ella*) y servidores `RADIUS` (*protocolo de autenticación y autorización para aplicaciones de acceso a la red o movilidad IP. Una de las características más importantes del protocolo RADIUS es su capacidad de manejar sesiones, notificando cuando comienza y termina una conexión*).

Dependiendo del servidor ftp, podrás tener unos métodos de autenticación de usuarios u otros, por ejemplo en el servidor ftp ProFTPD se permite los siguientes métodos:

- ✓ Ficheros de autenticación del sistema operativo: `/etc/passwd` y `/etc/shadow`: Para ello usa las directivas `AuthUserFile` y `AuthGroupFile`.
  http://www.proftpd.org/docs/howto/AuthFiles.html
- ✓ Usuarios virtuales definidos mediante ficheros de autenticación (de texto) propios, distintos de los del sistema operativo: para ello también usa las directivas `AuthUserFile` y `AuthGroupFile`.
- ✓ Autenticación PAM (*mecanismo de autenticación flexible que permite abstraer las aplicaciones y otro software del proceso de identificación. También permite construir políticas diferentes de autenticación para cada servicio*): Es necesario establecer la directiva `AuthPAMAuthorative` a '`on`'.
  http://www.proftpd.org/docs/directives/linked/config_ref_AuthPAM.html
- ✓ Bases de datos SQL, tales como MySQL o Postgres. Para ello emplea el módulo `mod sql`; más información sobre el uso de `mod_sql` lo puedes encontrar en el HowTo SQL
  http://www.proftpd.org/docs/howto/SQL.html
- ✓ LDAP: Para ello emplea el módulo `mod ldap`.

- ✓ RADIUS: Para ello emplea el módulo `mod_radius`.

Mediante la directiva `UserPassword` (*http://www.proftpd.org/docs/directives/linked/config_ref_UserPassword.html*)se puede crear una contraseña para un usuario particular que sobreescribe la contraseña del usuario en `/etc/passwd` (o `/etc/shadow`), esta contraseña es solamente efectiva dentro del contexto en el cual la directiva es aplicada, esto es, no se modifica el fichero `/etc/passwd` (o `/etc/shadow`) sino que se da la posibilidad de que el usuario emplee otra contraseña distinta de la definida en los ficheros del sistema operativo.

En el siguiente anexo encontrarás más información sobre PAM.
Información sobre PAM

1.4.- Modos de conexión del cliente.

Si en una transferencia de archivos mediante el protocolo FTP el cliente posee un cortafuegos configurado para impedir acceso local a puertos TCP menores de 1024, ¿es posible la transferencia?

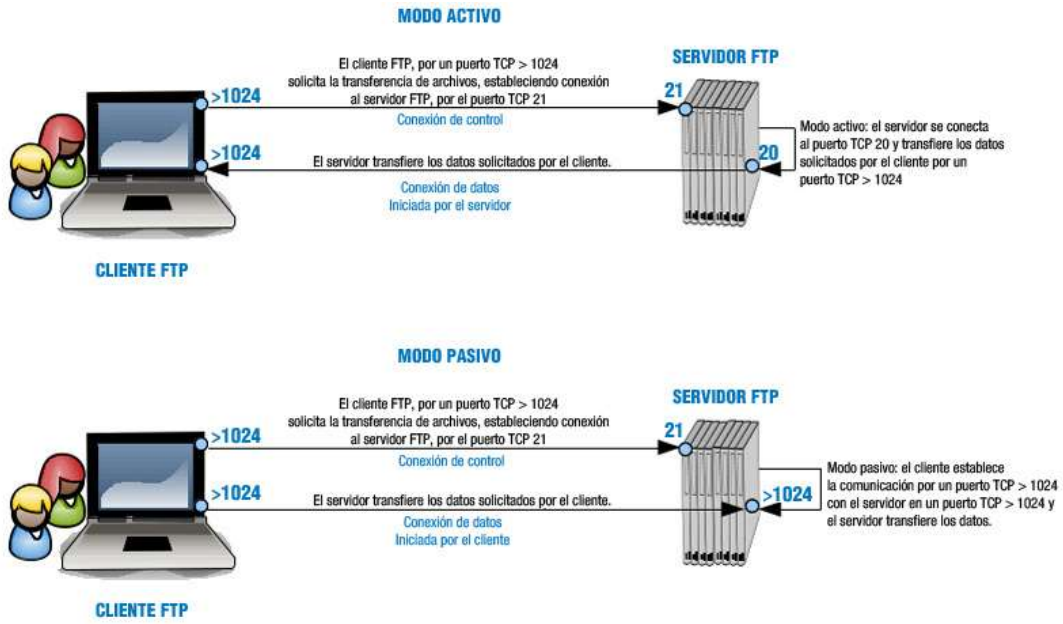
Ya se ha comentado que el servidor FTP a diferencia de otros servidores necesita dos puertos TCP para hacer posible la transferencia de archivos. Ahora bien, ¿son estos puertos siempre los mismos o no? ¿son independientes del tipo de cliente y servidor o no? Pues, básicamente depende de dos factores: del modo de conexión del cliente ftp y de la configuración del servidor ftp.

A priori, si no modificamos la configuración del servidor ftp esté otorgará siempre el puerto TCP 21 para el canal de conexión de control. Es el puerto del canal de transmisión de datos, el que varía, ¿cómo?, pues según el modo de conexión del cliente ftp, que puede ser activo o pasivo.

Cuando una aplicación cliente FTP inicia una conexión a un servidor FTP, abre el puerto 21 en el servidor. Se utiliza este puerto para arrojar todos los comandos al servidor. Cualquier petición de datos desde el servidor se devuelve al cliente a través de otro puerto TCP del servidor dependiendo del modo de conexión del cliente. Así:

- ✓ El modo activo es el método original utilizado por el protocolo FTP para la transferencia de datos a la aplicación cliente. Cuando el cliente FTP inicia una transferencia de datos, el servidor abre una conexión desde el puerto 20 en el servidor para la dirección IP y un puerto aleatorio sin privilegios (mayor que 1024) especificado por el cliente. Este arreglo implica que la máquina cliente debe poder aceptar conexiones en cualquier puerto superior al 1024. Con el crecimiento de las redes inseguras, tales como Internet, es muy común el uso de cortafuegos para proteger las máquinas cliente. Debido a que estos cortafuegos en el lado del cliente normalmente rechazan las conexiones entrantes desde servidores FTP en modo activo, se creó el modo pasivo.
- ✓ La aplicación FTP cliente es la que inicia el modo pasivo, de la misma forma que el modo activo. El cliente FTP indica que desea acceder a los datos en modo pasivo y el servidor proporciona la dirección IP y el puerto aleatorio, sin privilegios (mayor que 1024) en el servidor. Luego, el cliente se conecta al puerto en el servidor y descarga la información requerida.

A continuación puedes ver una imagen que muestra el funcionamiento de los dos modos: el activo y el pasivo.



En sistemas GNU/Linux es típico encontrar el archivo `/etc/services` que contiene una lista de puertos TCP/UDP relacionado con los servicios estándar que trabajan en los mismos. Ejecuta el comando `cat /etc/services | grep ftp` y encontrarás todos los puertos y servidores relacionados con la cadena `ftp`.

1.5.- Tipos de transferencia de archivos.

¿Es lo mismo descargar/subir mediante FTP un archivo de vídeo, que uno de texto o uno ejecutable?

Desde el punto de vista de FTP, los archivos se agrupan en dos tipos:

- ✓ **Archivos ASCII** (*código de caracteres basado en el alfabeto latino. ASCII es, en sentido estricto, un código de siete bits, lo que significa que usa cadenas de bits representables con siete dígitos binarios (que van de 0 a 127 en base decimal) para representar información de caracteres. El código ASCII define así una relación entre caracteres específicos y secuencias de bits*): son archivos de texto plano (.txt, .ps, .html...)
- ✓ **Archivos binarios**: todo lo que no son archivos de texto: ejecutables (.exe), imágenes (.jpg, .png ...), archivos de audio (.mp3, .wav ...), vídeo (.avi, .mov …) , etcétera.

Es muy importante saber con qué tipo de archivos estás trabajando en la transferencia ya que si no utilizas las opciones adecuadas puedes destruir la información del archivo. El servidor ftp permite configurar la transferencia de archivos según el tipo del mismo, es por eso que al ejecutar el cliente FTP, antes de transferir un archivo, debes utilizar uno de los siguientes comandos o poner la correspondiente opción en un programa con interfaz gráfica:

- ✓ `ascii` para tipos de archivos ascii.
- ✓ `binary` para tipos de archivos binarios.

Relaciona cada extensión de archivo con el tipo de transferencia ftp correspondiente, escribiendo el número del tipo de transferencia en el cuadro correspondiente:

Extensión de archivo	Relación	Tipo de Transferencia	Relación	Extensión de archivo
txt (texto).	1	1. ascii. 2. binario.	1	ps (poscript).
html (página web).	1		2	mp3 (audio).
doc (documento).	2		2	tar (comprimido).
zip (comprimido).	2		2	tgz (comprimido).
bz2 (comprimido).	2			

1.6.- Establecer permisos en ftp.

El protocolo FTP sigue los permisos establecidos en entornos de tipo UNIX y sus similares GNU/Linux, con lo cual existen tres grupos de permisos en el siguiente orden: propietario, grupo y otros:

- ✓ **Propietario(user=`u`):** El creador o el que ha subido el archivo al servidor FTP.
- ✓ **Grupo(group=`g`):** Se refiere a un grupo de usuarios que posee la propiedad del archivo, al que probablemente pertenece el propietario.
- ✓ **Otros(others=`o`):** Son el resto de usuarios no propietarios o que no pertenecen al grupo indicado. Son el resto del mundo.

Cada grupo a su vez puede tener tres permisos en el siguiente orden: lectura, escritura y ejecución, identificados respectivamente por una '`r`', una '`w`' y una '`x`'. La ausencia de permiso es identificada con el carácter '`-`'. Cada permiso tiene un equivalente numérico, así: `r=4`, `w=2`, `x=1` y `-=0`. Por ejemplo: `rw-` identifica permiso de lectura y escritura o lo que es lo mismo **4+2+0=6**

En un sistema operativo tipo GNU/Linux mediante el comando '`ls -l`' puedes ver los permisos asignados a ficheros y directorios, por ejemplo si la salida del anterior comando es:

```
-rw-r--r-- 1 alumno clase 0 jun 20 01:15 prueba1.txt
```

significa que,

- ✓ `prueba1.txt` es un fichero ya que `-rw-r--r--` comienza con '`-`', si fuese un directorio aparecería una '`d`'
- ✓ `rw-r--r--` identifica los permisos del fichero `prueba1.txt`, que divididos 3 a 3 representan de izquierda a derecha: propietario, grupo, otros.
- ✓ `rw-` identifican los permisos del usuario propietario, en este caso `alumno`. Por lo tanto alumno posee los permisos de **lectura** y **escritura** sobre el fichero `prueba1.txt` o lo que es lo mismo **4+2+0=6**
- ✓ `r--` identifican los permisos del grupo propietario, en este caso `clase`. Por lo tanto `clase` posee solamente el permiso de **lectura** o lo que es lo mismo**4+0+0=4**
- ✓ `r--` identifican los permisos de los `otros` (resto del mundo). Por lo tanto todos los usuarios que no son alumno y aquellos que no pertenecen al grupo clase poseen solamente el permiso de **lectura** o lo que es lo mismo **4+0+0=4**

Por lo tanto los permisos `rw-r- -r- -` equivalen a `644`.

Es conveniente que le des un vistazo al manual de `chmod` y `umask`: `man chmod` y `man umask`.

Por otro lado en un sistema GNU/Linux, en principio, no todos los usuarios del sistema tienen acceso por `ftp`, así existe un fichero `/etc/ftpusers` que contiene una lista de usuarios que no tienen permiso de acceso por FTP. Por razones de seguridad al menos los siguientes usuarios deberían estar listados en este fichero: `root`, `bin`, `uucp`, `news`. Ten en cuenta que las líneas en blanco y las líneas que comiencen por el carácter '`#`' serán ignoradas.

Ejecutas en una consola de comandos en la ruta /home/alumno el comando ls -l obteniendo la siguiente salida:

```
drwxr-x--- 1 alumno clase 0 jun 20 01:16 Documentos
```

Entonces, con esa información puedes deducir que:

- ☑ **Documentos es un directorio con permisos 750.**
- ☐ Documentos es un fichero con permisos 750.
- ☑ **Documentos pertenece al usuario propietario alumno y al grupo propietario clase. Además el usuario alumno posee permisos de modificación, mientrás que el grupo clase y los**

demás no poseen ese permiso.

Documentos pertenece al grupo propietario alumno y al usuario propietario clase. Además el usuario alumno posee permisos de modificación, mientrás que el grupo clase y los demás no poseen ese permiso.

*Documentos es un directorio ya que **drwxr-x---** comienza con '**d**', si fuese un fichero aparecería un '-'*

- ✓ ***rwxr-x---** identifica los permisos del directorio Documentos, que divididos 3 a 3 representan de izquierda a derecha: propietario, grupo, otros.*
- ✓ ***rwx** identifican los permisos del usuario propietario, en este caso **alumno**. Por lo tanto **alumno** posee los permisos de lectura, escritura y ejecución sobre el directorio **Documentos** o lo que es lo mismo **4+2+1=7***
- ✓ ***r-x** identifican los permisos del grupo propietario, en este caso **clase**. Por lo tanto clase posee los permisos de lectura y ejecución o lo que es lo mismo **4+0+1=5***
- ✓ ***---** identifican los permisos de los **otros** (resto del mundo). Por lo tanto todos los usuarios que no son **alumno** y aquellos que no pertenecen al grupo **clase** no poseen permisos o lo que es lo mismo **0+0+0=0***

*Por lo tanto los permisos **rwxr-x---** equivalen a **750**.*

1.7.- Servicio de transferencia de archivos en modo texto.

Como se comentó anteriormente, existen varios tipos de clientes ftp, entre los cuales los clientes en modo texto desde siempre estuvieron incorporados en las distribuciones GNU/Linux.

De entre los clientes tipo texto cabe destacar dos: el cliente en modo texto `ftp` y el cliente en modo texto `lftp`. En GNU/Linux Debian 6 se dispone del cliente modo `ftp` en una instalación básica. Para poder utilizarlo en el sistema simplemente hay que ejecutarlo como comando: el comando `ftp`.

Vamos a ver, a continuación, el comportamiento del cliente en modo texto `ftp` en la conexión al servidor `ftp` `ftp.rediris.es`:

1. Básicamente la sintaxis es la siguiente:

   ```
   ftp [-pinegvd] [host [port]]
   ```

   donde
   - ✓ `host` identifica el servidor ftp
   - ✓ `port` identifica el puerto, por defecto 21, por lo cual si conectas a un servidor ftp configurado en ese puerto no es necesario escribirlo, ya se considera.

   Puedes ver la ayuda del comando ftp mediante: **`man ftp`** ó **`info ftp`** .
2. Al ejecutar el comando se abrirá una consola propia de ftp en la cual puedes introducir comandos ftp para: abrir conexión, moverse por rutas, descargar archivos …
   Es muy típico ejecutarlo con el parámetro host, esto es, con el servidor ftp al cual quieres conectar:

   ```
   root@debian-servidor-fp:~# ftp ftp.rediris.es
   ```

   También puedes ejecutar el comando sin parámetros, de esta forma abrirás directamente la consola ftp y deberás actuar con ella a través de los comandos de la misma:

   ```
   root@debian-servidor-fp:~# ftp
   ftp> o
   (to) ftp.rediris.es
   ```
3. A continuación se pedirá usuario y contraseña para establecer la conexión. En el caso del servidor de rediris puedes conectar mediante un usuario cualquiera y una contraseña cualquiera. Es muy típico en servidores ftp que exista un usuario anónimo, cuya contraseña sea cualquier dirección de correo -real o ficticia-, por ejemplo: `a@` .
4. Ahora en la consola ftp puedes ejecutar comandos, ¿cuales? Pues los que estén habilitados, y ¿cuales están habilitados? Lo puedes saber ejecutando el comando help.

En la siguiente imagen puedes ver el ejemplo de conexión ftp mediante el cliente en modo texto ftp al servidor ftp.rediris.es:

```
root@debian-servidor-fp:~# ftp
ftp> o
(to) ftp.rediris.es
Connected to zeppo.rediris.es.
220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
220-You are user number 38 of 3000 allowed.
220-++
220-  Bienvenido al FTP anónimo de RedIRIS.
220-Welcome to the RedIRIS anonymous FTP server.
220->>
220-Local time is now 07:19. Server port: 21.
220-Only anonymous FTP is allowed here.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 5 minutes of inactivity.
Name (ftp.rediris.es:root):
331-           RedIRIS - Red Académica y de Investigación Española
331-               RedIRIS - Spanish National Research Network
331-
331-          ftp://ftp.rediris.es  -*-  http://ftp.rediris.es
331-
331-Debido a una incidencia hardware del almacenamiento que utiliza el servicio, éste
331-no estará disponible hasta nuevo aviso. Rogamos disculpen las molestias que este
331-fallo les pueda ocasionar. Estamos trabajando para resolver esta incidencia lo antes
331-posible.
331 Any password will work
Password:
230 Any password will work
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> help
Commands may be abbreviated.  Commands are:

!               debug           mdir            qc              send
$               dir             mget            sendport        site
account         disconnect      mkdir           put             size
append          exit            mls             pwd             status
ascii           form            mode            quit            struct
bell            get             modtime         quote           system
binary          glob            mput            recv            sunique
bye             hash            newer           reget           tenex
case            help            nmap            rstatus         tick
cd              idle            nlist           rhelp           trace
cdup            image           ntrans          rename          type
chmod           lcd             open            reset           user
close           ls              prompt          restart         umask
cr              macdef          passive         rmdir           verbose
delete          mdelete         proxy           runique         ?
ftp>
```

A continuación puedes ver el mismo ejemplo de conexión al servidor `ftp: ftp.rediris.es` utilizando el cliente en modo texto `lftp`:

```
1) Actualizar sistema operativo GNU/Linux Debian 6 Squeeze:
root@debian-servidor-fp:~# apt-get update
root@debian-servidor-fp:~# apt-get upgrade

2) Instalar cliente ftp en modo texto: lftp
root@debian-servidor-fp:~# apt-get install lftp

3) Ver ayuda comando lftp
root@debian-servidor-fp:~# lftp --help
Uso: lftp [OPCS] <servidor>
`lftp' es la primera orden ejecutada por lftp después de los archivos
de configuración.
 -f <archivo>           ejecuta órdenes del archivo y sale
 -c <ords>              ejecuta las órdenes y sale
 --help                 muestra esta ayuda y sale
 --version              muestra la versión de lftp y sale
Las demás opciones son las mismas que en la orden `open'
 -e <ord>               ejecuta la orden justo después de seleccionar
 -u <usuario>[,<clave>] usa usuario/clave para autenticación
 -p <puerto>            usa el puerto para una conexión
 <sitio>                servidor, URL o nombre de señalador

4) Establecer conexión con el servidor ftp: ftp.rediris.es
root@debian-servidor-fp:~# lftp ftp.rediris.es
lftp ftp.rediris.es:~>

Como puedes ver conecta directamente, a difencia del comando ftp, ya que no pide usuario y
contraseña. Esto es debido a que automaticamente comprueba si la
conexión mediante un usuario anónimo es posible, y si es posible introduce unos valores para
establecer la conexión.

5) Introducir comando help para saber que comandos dispones dentro de la consola ftp:
lftp ftp.rediris.es:~> help
    !<orden-de-shell> (órdenes) alias [<nombre> [<valor>]] bookmark [SUBORDEN]
cache [SUBORDEN]
    cat [-b] <archivos> cd <dir remoto> chmod [OPTS] modo archivo. close [-a]
    [re]cls [opts] [path/][pattern] debug [<nivel>|off] [-o <archivo>] du [options] <dirs>
 exit [<código>|bg]
    get [OPCS] <arch_r> [-o <arch_l>] glob [OPTS] <cmd> <args> help [<ord>]
```

```
history -w file|-r file|-c|-l [cnt]  jobs [-v]
    kill all|<númtarea> lcd <dirlocal> lftp [OPCS] <servidor> ls [<args>]
    mget [OPCS] <archivos> mirror [OPCS] [remoto [local]] mkdir [-p] <dirs> module nombre
[args] more <archivos>
    mput [OPCS] <archivos> mrm <archivos> mv <archivo1> <archivo2> [re]nlist [<args>]
    open [OPCS] <servidor> pget [OPCS] <arch_r> [-o <arch_l>]    put [OPCS] <arch_l> [-o
<arch_r>] pwd [-p] queue [OPTS] [<cmd>]
    quote <orden> repeat [OPTS] [delay] [command] rm [-r] [-f] <archivos> rmdir [-f] <dirs>
    scache [<núm_sesión>] set [OPT] [<var> [<val>]] site <orden_directa_al_sitio> source
<archivo>
    torrent [-O <dir>] <file> user <usuario|URL> [<clave>] version wait [<númtarea>] zcat
<archivos>
    zmore <archivos>
lftp ftp.rediris.es:~>
```

1.7.1.- Comandos ftp.

En la consola ftp pueden estar disponibles múltiples comandos, algunos de los más empleados son los recogidos en la siguiente tabla:

ABRIR/CERRAR CONEXIÓN	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**
open servidor	Inicia conexión remota con un servidor ftp.
close / disconnect	Finalizan la sesión ftp sin cerrar la consola ftp.
bye / quit / exit	Terminan la sesión ftp y salen de la consola ftp.
!	Sale a línea de comandos del sistema operativo temporalmente sin cortar la conexión. Para volver, teclea exit en la línea de comandos.
AYUDA	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**
? / help	Muestra una lista de los comandos disponibles.
? comando / help comando	Muestra la información relativa al comando.
TRABAJAR CON DIRECTORIOS	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**
cd directorio	Cambia de directorio en el servidor remoto.
lcd directorio	Cambiarse de directorio en el equipo local (cliente ftp).
dir directorio / ls directorio	Listan el contenido del directorio remoto actual.
pwd	Muestra el directorio activo en el servidor.
lpwd	Muestra el directorio activo en el equipo local (cliente ftp).
rmdir directorio	Elimina un directorio vacío en el servidor.
mkdir directorio	Crea un directorio en el servidor.Crea un directorio en el servidor.
TRABAJAR CON FICHEROS	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**
delete archivo	Borrar un archivo en el servidor remoto.
mdelete patrón	Borrar varios archivos según un patrón.
get archivo	Obtiene archivo en el equipo cliente desde el servidor remoto.
mget archivos	Obtiene varios archivos desde el servidor remoto.
put archivo	Envía un archivo al servidor remoto.
mput archivos	Envía varios archivos al servidor remoto.
rename archivo	Cambia el nombre a un archivo en el servidor.
ascii	Para configurar y transferir archivos tipo ascii.
binary	Para configurar y transferir archivos tipo binario.
less archivo	Leer contenido de archivo mediante el comando less.
TRABAJAR CON PERMISOS	
COMANDO/S Y ARGUMENTOS	**EXPLICACIÓN**

chmod	Cambio de permisos en el servidor remoto.
umask	Configura el sistema de permisos en el lado remoto.

Observa la siguiente tabla con más comandos ftp.

COMANDO	USO
$	Ejecuta macro
account	Envía comando a la cuenta del servidor remoto
append	Concatena un archivo
bell	Sonido de campanilla cuando el comando se ha completado
case	Mapeo de letras iguales
cdup	Cambiar al directorio padre en el servidor remoto
cr	Retorno de carro
debug	Configura modo de errores
form	Configurar formato de transferencia de archivos
glob	Transponer nombre de archivo local con un metacarácter
hash	Imprimir el metacarácter "#" por cada buffer transferido
idle	Configurar el tiempo disponible en el lado remoto
image	Para configurar y transferir archivos tipo binario
less	Ver el contenido de un archivo pudiendo subir y bajar por el mismo mediante las flechas
macdef	Define una macro
mdir	Lista contenido de varios directorios remotos
mls	Lista contenido de varios directorios remotos
mode	Configura el modo de transferencia
modtime	Modo de reloj
newer	Recibe el archivo remoto si es más nuevo que el de la máquina local
nlist	Lista el contenido de varios directorios remotos
nmap	Configura nombre de archivo de acuerdo a plantilla
ntrans	Configura tabla de traducción para mapeo de nombres de archivos
prompt	Fuerza la ejecución de múltiples comandos
proxy	Comando para conexión alternativa
sendport	Activa/desactiva use del comando PORT para cada conexión de de datos
set	
site	Envia un comando específico a la máquina remota
size	Muestra el tamaño de un archivo
struct	Configura la estructra de la transferencia de los archivos
sunique	Activa/desactiva almacenamiento único sobre la máquina remota
tenex	Transferencia de archivos de tipo tenex
trace	Activa/desactiva trazado de transferencia de paquetes
type	Configura el tipo de archivo a transferir
user	Envia información de usuario nuevo
verbose	Activa/Desactiva modo de entrega de información completa

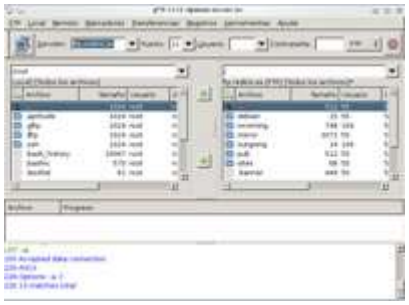
1.8.- Servicio de transferencia de archivos en modo gráfico.

El servicio de transferencia de ficheros está bien, pero obliga a entender el funcionamiento de un servidor ftp mediante el uso de sus comandos. La verdad, es que no es muy interactivo, ¿entonces..., no existe la posibilidad de trabajar de otro modo más interactivo? Pues si, mediante clientes ftp de modo gráfico o mediante el navegador, ya que éste incorpora su propio cliente ftp.

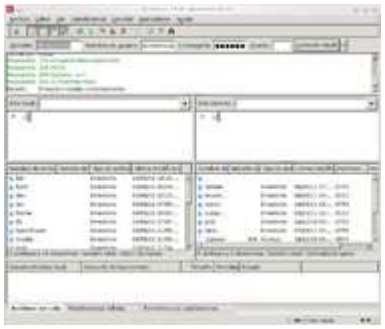
Típicamente los clientes gráficos se comportan todos igual, esto es, tienen una interfaz parecida, básicamente presentan una ventana partida en dos secciones: la de la izquierda suele representar el equipo cliente ftp (*desde donde se intenta establecer la conexión*) y la de la derecha suele representar el equipo servidor ftp (quién recibe la conexión). Luego suelen existir, en alguna zona determinada de la ventana: en el centro entre las dos secciones, arriba de las dos secciones, etc una serie de botones, usualmente representados como flechas que indican la posibilidad de subir o descargar archivos. Incluso dependiendo del cliente en modo gráfico es posible guardar los datos de las conexiones como plantillas, de tal forma que la próxima vez que intentes establecer la conexión con un mismo servidor ftp en vez de tener que rellenar los campos referentes a la conexión puedes hacerlo a través de la plantilla que ya posee el valor de esos campos.

Dentro de los clientes ftp en modo gráfico cabe destacar dos: `gftp` y `filezilla`. A continuación puedes ver un ejemplo de como utilizarlos para establecer una conexión con un servidor ftp, el servidor `ftp.rediris.es` :

1. Cliente en modo gráfico `gftp`.
   - ✓ **Servidor**: Escribe aquí el nombre o IP del servidor FTP: `ftp.rediris.es`
   - ✓ **Puerto**: Escribe aquí el puerto TCP de la conexión de control, por defecto: `21`. Puedes omitirlo siempre y cuando sea el 21.
   - ✓ **Usuario**: Escribe aquí el usuario con permisos de conexión en el servidor ftp. En la imagen puedes ver que no se ha escrito nada, esto es debido que el servidor `ftp.rediris.es` permite la entrada a cualquier usuario y el cliente gráfico `gftp` al intentar conectar te pedirá un usuario que tenga permisos para la conexión. Pulsas en cancelar y `gftp` cubrirá los campos usuario y contraseña, entrando al servidor ftp.



   - ✓ **Contraseña**: Escribe aquí la contraseña del usuario con permisos de conexión en el servidor ftp. En la imagen puedes ver que no se ha escrito nada, esto es debido a la misma causa que en el campo Usuario.
2. Cliente en modo gráfico `filezilla`.
   - ✓ **Servidor**: Escribe aquí el nombre o IP del servidor FTP: `ftp.rediris.es`
   - ✓ **Nombre de usuario**: Escribe aquí el usuario con permisos de conexión en el servidor `ftp.Filezilla`, al contrario que `gftp`, no cubre los datos usuario y contraseña si tú no escribes nada en los campos, entonces debes escribir un nombre de usuario, por ejemplo `anonymous`, y una contraseña (*cualquier secuencia de caracteres*).



   - ✓ **Contraseña**: Escribe aquí la contraseña del usuario con permisos de conexión en el servidor ftp. En la imagen puedes ver que se ha escrito una secuencia de caracteres punto, lo que significa que a la hora de escribir caracteres en ese campo no se muestra su valor por seguridad. Es necesario escribir una contraseña por lo comentado en el campo anterior: Nombre de usuario.
   - ✓ **Puerto**: Escribe aquí el puerto TCP de la conexión de control, por defecto: `21`. Puedes omitirlo siempre y cuando sea el 21.

A continuación puedes ver información de la instalación del cliente en modo gráfico gftp.

1. Actualizar sistema operativo GNU/Linux Debian 6 Squeeze:

```
root@debian-servidor-fp:~# apt-get update
root@debian-servidor-fp:~# apt-get upgrade
```

2. Instalar cliente ftp en modo gráfico: gftp

```
root@debian-servidor-fp:~# apt-get install gftp
```

Te proponemos el siguiente enlace de un vídeo práctico sobre la instalación y uso de FileZilla en una distribución GNU/Linux basada en Debian.

http://www.youtube.com/watch?feature=player_embedded&v=nBm-2rgkf5Y

Resumen:

Se ve una consola de comandos del usuario `root`, y en la misma el contenido del **script** `FileZilla Perfiles.sh`, a saber:

```
#!/bin/bash
# Actualizar repositorios
apt-get update
# Actualizar sistema operativo
apt-get upgrade
# Buscar paquete filezilla
apt-cache search filezilla
# Instalar paquete filezilla
apt-get install filezilla
# Arrancar filezilla en la consola del usuario del sistema: alumno
su -c filezilla alumno
# Dirigirse a Gestor de Sitios para crear los perfiles a conexiones ftp
```

Se ejecuta el script mediante el comando:

```
sh FileZilla_Perfiles.sh
```

Una vez acabada la ejecución del script aparece el cliente ftp gráfico **FileZilla**. En la interface arriba a la izquierda aparece el primer **icono Abrir el Gestor de Sitios**. Se hace click en el mismo y aparece un panel con dos secciones: la de la izquierda donde aparecen los Sitios configurados y la de la derecha donde aparecen las opciones configuradas de cada Sitio Seleccionado:

- ✓ En la sección de la izquierda no existe ningún sito configurado y se pulsa en el botón **Nuevo Sitio**, apareciendo una caja de texto donde se escribe el nombre del sitio a configurar (nombre del perfil a guardar). Se escribe **REDIRIS** y se activa la sección de la derecha.
- ✓ En la sección de la derecha, al tener seleccionado **REDIRIS**, se escribe `ftp.rediris.es` en la caja de texto **Servidor** y se pulsa el botón **Aceptar**, desapareciendo el panel de configuración.

Se vuelve a pulsar en el **icono Abrir el Gestor de Sitios** y aparece el nuevo sitio(perfil) configurado REDIRIS. Ahora se pulsa el botón Conectar en la sección de la derecha del panel y se establece la conexión con el servidor `ftp.rediris.es`.

1.9.- Servicio de transferencia de archivos desde el navegador.

El navegador web también puede ejercer de cliente ftp y, puesto que la mayoría de los sistemas operativos cuentan con un navegador en su instalación, es una de las herramientas más usadas para transferencia de archivos.

Para poder usar el navegador como cliente ftp solamente debes escribir en la barra de dirección una dirección URL tipo, como la siguiente:

```
ftp://nombre_servidor_ftp:puerto
```

donde,

- ✓ `ftp://` indica que el protocolo que deseas que interprete el navegador sea el ftp.
- ✓ `nombre_servidor_ftp` representa el nombre o la IP del servidor ftp.
- ✓ `puerto` indica el puerto TCP, por defecto 21. Puedes omitirlo siempre y cuando sea el 21.

Si el servidor ftp permite la conexión a un usuario anónimo, al ejecutar `ftp://nombre servidor ftp:puerto` entrarás directamente al servidor ftp, esto es, el navegador no preguntará qué usuario y contraseña necesitas para establecer la conexión.

En la siguiente imagen puedes ver como puedes acceder al servidor ftp de rediris utilizando el navegador:

Así, lo único que tienes que hacer es escribir en la dirección URL: `ftp://ftp.rediris.es` y pulsar **Enter**, con lo cual, automáticamente, conectas con el servidorftp, pudiendo visitar las carpetas y ver los ficheros como si de un explorador de archivos se tratará.

Para descargar las carpetas o archivos simplemente debes pulsar con el botón derecho del ratón sobre ellos y elegir la opción **Guardar enlace como…** -que aparece en Firefox y es similar en otros navegadores-.

Pero no todo van a ser ventajas al utilizar el navegador como cliente ftp, puesto que otros clientes tienen la posibilidad de continuar las descargas cuando estás sufrieron algún tipo de interrupción, cosa que no pasa con el cliente ftp del navegador, como por ejemplo el cliente gráfico FileZilla que soporta y reanuda la transferencia de archivos de gran tamaño(> 4 GB).

1.10.- Asegurando el servicio de transferencia de archivos.

Bien, pero ¿qué pasa con los datos en la transferencia? ¿viajan cifrados? ¿no? Pues empleando el protocolo ftp cualquiera que tenga acceso al canal de transmisión podrá ver en texto claro todo lo que se transmite, esto es, los datos no se cifran. Esto puede carecer de importancia, o no, según el contexto de la transmisión. Así, puede que a un organismo público no le importe compartir información a través de ftp y que los datos en la transferencia viajen sin cifrar y, sin embargo, a una empresa si le interese que los datos viajen cifrados.

Entonces, cuando interese asegurar el servicio de transferencia de archivos debes descartar el protocolo ftp y empezar a pensar en otras alternativas, como: `ftps` o `sftp`.

`FTPS` es una extensión del protocolo FTP que asegura el cifrado en la transferencia mediante los protocolos `SSL/TLS`. Permite tres tipos de funcionamiento:

- ✓ SSL Implícito:
  - ➔ Como conexiones HTTPS.
  - ➔ Usa los puertos 990 y 989.
- ✓ SSL Explícito
  - ➔ El cliente usa los mismos puertos estándar FTP: 20 y 21 pero se efectúa el cifrado en ellos.
  - ➔ Usa AUTH SSL.
- ✓ TLS Explícito:
  - ➔ Similar a SSL Explícito pero usa AUTH TLS.

El cifrado al que nos referimos es el cifrado de clave pública o asimétrico: **clave pública(`kpub`)** y **clave privada(`kpriv`)**. La `kpub` interesa publicarla para que llegue a ser conocida por cualquiera, la **kpriv** no interesa que nadie la posea, solo el propietario de la misma. Ambas son necesarias para que la comunicación sea posible, una sin la otra no tienen sentido, así una información cifrada mediante la `kpub` solamente puede ser descifrada mediante la `kpriv` y una información cifrada mediante la `kpriv` sólo puede ser descifrada mediante la `kpub`.

En el cifrado asimétrico podemos estar hablando de individuos o de máquinas, en nuestro caso hablamos de máquinas y de flujo de información entre el **cliente ftp(A)** y el **servidor ftp(B)**. Ver la siguiente tabla como ejemplo de funcionamiento del cifrado asimétrico:

Funcionamiento del cifrado asimétrico.

A(inf) → inf cifrada → B [descifrar inf] → B(inf) = A(inf)
A(inf) → inf cifrada = [(inf)]kpubB → B [inf. cifrada]kprivB → B(inf) = A(inf)

	Identificación
A	Cliente ftp.
inf cifrada = [(inf)]kpubB	Información cifrada mediante la clave pública de B obtenida a través de un certificado digital.
[inf. cifrada]kprivB	Información descifrada mediante la clave privada de B.
B	Servidor ftp.

En el siguiente enlace encontrarás más información sobre asegurar FTP con TLS.
http://tools.ietf.org/html/rfc4217
En este otro enlace encontrarás más información sobre el protocolo TLS.
http://tools.ietf.org/html/rfc4346

1.11.- El servicio de transferencia de archivos en el proceso de despliegue de una aplicación Web.

¿Cómo actualizas una página web de forma remota? ¿Qué servicios suelen ofrecer las empresas de alojamiento web para que puedas subir archivos a tus aplicaciones?¿Cuánto tiempo se mantiene la conexión subiendo un archivo?

Suele ser típico que cualquier aplicación web en Internet disponga de la posibilidad de subir archivos mediante una configuración del código fuente de la misma, una aplicación propia o una aplicación de terceros, como los paneles de administración web.

Si empleas una aplicación web para subir archivos debes tener en cuenta cuánto tiempo puedes mantener la conexión abierta con el servicio web y cuál es el tamaño máximo de subida de un archivo. Estas cuestiones suelen ser típicas de la configuración del servidor web. Por la contra, si empleas un servidor ftp dependerá de éste las cuestiones anteriores.

Se suele configurar el servidor web con unos parámetros: tiempo de conexión y tamaño máximo de subidas de archivos diferentes del servidor ftp, de tal forma que para archivos de tamaño no muy grandes se puedan emplear aplicaciones web y no se sufra un corte en la subida de archivos y, para archivos grandes, se emplee el servidor ftp.

Normalmente las empresas de alojamiento web (hosting) permiten la subida de archivos mediante un servidor ftp y poseen documentos sobre cómo operar con éste, esto es, documentación que explica cómo conectarse a sus servidores ftp a través de algún cliente ftp, como por ejemplo: `FileZilla`, `Cute FTP`, `Fetch` o `Transmit`. También suelen permitir usar `SCP` o `SFTP` para transferir ficheros de forma segura mediante un canal cifrado. Por ejemplo en Filezilla se puede establecer la conexión de forma cifrada directamente, sólo con indicar como puerto `TCP` el número del servidor `SSH`, por defecto, `22`.

También debes saber que muchos editores web permiten subir tu aplicación web al servidor con el protocolo FTP, esto te puede resultar más sencillo que el uso de una aplicación de FTP independiente.

Eso si, sea cual sea el método ftp que utilices para subir archivos y actualizar tu web, se desaconseja el uso de aplicaciones no actualizadas que podrían comprometer la seguridad de tu web.

A continuación puedes ver errores típicos, junto con sus soluciones, que puedes encontrar al subir tu aplicación mediante un servidor FTP:

- ✓ Tu cliente FTP muestra el error **`access denied`**, o similar, cuando subes o borras ficheros y carpetas: Comprueba que tu usuario FTP tenga permisos suficientes sobre la carpeta o fichero en la que deseas subir o que deseas borrar.
- ✓ Tus páginas no son reconocidas de forma automática al acceder a tu dominio: Los servidores GNU/Linux son sensibles a mayúsculas y minúsculas por lo que verifica el nombre de tus archivos.
- ✓ El cliente de FTP te muestra el mensaje **`too many connections from your IP address`**: Esto quiere decir que existen más conexiones abiertas con el servidor FTP desde la misma dirección IP de las permitidas. En ese caso, asegúrate que no exista ninguna aplicación, como un cortafuegos, que pueda estar bloqueando las conexiones abiertas, y provocando, de esta forma, que se establezcan más intentos de conexión de los necesarios.

2.- Instalación del servidor proftpd.

Caso práctico

Bien —pensó María— ya es la hora, una vez configurado el servidor web, alojada la página y comprobada su visibilidad a través de Internet, toca permitir la subida de archivos de gran tamaño a la carpeta upload preparada para tal fin. Así que, déjame pensar, María, ¿qué debo hacer?:

1. *Esta empresa maneja archivos de gran tamaño, por lo que habrá que configurar un servidor ftp para que en la subida de archivos la conexión no se corte, que es lo más probable que ocurra a través de la subida de archivos mediante la propia aplicación web.*
2. *Mirar qué servicio han contratado, puesto que el nivel de cuota en disco puede variar según el servicio.*
3. *Crear un usuario virtual para que pueda subir archivos, —¿cómo lo haré? ¿mediante base de datos SQL? —Uhm..., creo que lo mejor será crear un usuario virtual y, como solamente se trata de un usuario, pues lo crearé mediante un archivo de autenticación.*
4. *Si se necesita crear otro usuario, pues otro en el archivo de autenticación y listo. ¿Y si necesito crear grupos? Pues lo mismo, en un archivo de autenticación de grupos.*
5. *Se necesita que la comunicación sea cifrada, con lo cual se debe emplear el protocolo SSL. —¿Uhm...? Mejor el protocolo TLS, que es el sucesor de SSL. Pero, claro, si empleo cifrado voy a tener que utilizar otros puertos TCP distintos del servidor ftp que maneja por defecto: el 21 para la conexión de control y el 20 para la conexión de datos.*
6. *Ya está, mejor empleo TLS Explícito, de tal forma que puedo seguir utilizando los mismos puertos 20 y 21 para el cifrado.*
7. *Y todo esto no debe modificar la configuración ya realizada para otras empresas.*

*Pues clarísimo, lo que tengo que hacer es utilizar el servidor **ProFTPD**.*

Entonces:

- ✓ *Primero, comprobar si en este servidor dedicado está instalado y, si no lo está, instalarlo.*
- ✓ *Segundo, configurar el servidor proftpd de la siguiente forma:*
  - ➔ *Configuración independiente para esta empresa.*
  - ➔ *Usuario virtual en un archivo de autenticación.*
  - ➔ *Cifrado TLS Explícito para asegurar el cifrado.*
  - ➔ *Cuota de disco.*
  - ➔ *Permisos de subida en la carpeta upload correspondiente.*

Pues, manos a la obra María, que se va haciendo tarde.

¿Por qué ProFTPd? Pues porque es un servidor FTP bajo licencia GPL altamente configurable, así permite:

- ✓ Usuarios virtuales con:
  - ➔ LDAP
  - ➔ BBDD: MySQL, PostgreSQL...
  - ➔ Ficheros de autenticación (ficheros de texto).
- ✓ Personalizar opciones según usuario/grupo.
- ✓ Seguridad mediante cifrado SSL/TLS.
- ✓ Configuraciones independientes mediante virtualhosts.

Para instalar el servidor protftpd en un sistema Operativo Debian 6.0 (squeeze) ejecutar el comando `apt-get install proftpd`. En la instalación deberás elegir si ProFTPd va a ejecutarse como un servicio desde `inetd` o como un **servidor independiente**. Ambas opciones tienen sus ventajas. Si sólo recibes unas pocas conexiones FTP diarias, probablemente sea mejor ejecutar ProFTPd desde inetd para ahorrar recursos. Por otro lado, con más tráfico, ProFTPddebería ejecutarse como un servidor independiente para evitar crear un proceso nuevo por cada conexión entrante.

En la instalación se crearán los usuarios `proftpd` y `ftp` con grupo `nogroup` y sin posibilidad de acceso a una consola del sistema. Se puede comprobar en el fichero `/etc/passwd` donde encontrarás nuevas líneas similares a las siguientes:

```
proftpd:x:106:65534::/var/run/proftpd:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
```

Te proponemos el siguiente enlace a un vídeo práctico sobre la instalación y uso de protfpd en una distribución GNU/Linux basada en Debian.

http://www.youtube.com/watch?feature=player_embedded&v=ijoI1lTDpcI

Resumen

Se ve una consola de comandos del usuario **root**, y en la misma el contenido del script `Instalacion_ProFTPD.sh`, a saber:

```
#!/bin/bash
# Actualizar repositorios
apt-get update
# Actualizar sistema operativo
apt-get upgrade
# Buscar paquete proftpd
apt-cache search proftpd
# Instalar paquete proftpd-basic
apt-get install proftpd-basic
# Ruta de configuración de ProFTPD: /etc/proftpd
ls -l /etc/proftpd
# Servicio proftpd: Posibilidades
/etc/init.d/proftpd
# Servicio proftpd: Posibilidades
service proftpd

# Con esta configuración cualquier usuario del sistema puede acceder por ftp

# Modificar /etc/proftpd/proftpd.conf y descomentar las líneas correspondientes a Anonymous
# Puedes recargar la nueva configuración con:
## /etc/init.d/proftpd reload
## ó
## service proftpd reload

# Puedes reiniciar el servicio mediante:
## /etc/init.d/proftpd restart
## ó
## service proftpd restart

# Con esta nueva configuración puede accepter por ftp el usuario anónimo
# anonymous o su alias: ftp

# Para desinstalar proftpd
# apt-get remove proftpd
```

Se ejecuta el script mediante el comando:

```
sh Instalacion_ProFTPD.sh
```

Una vez acabada la ejecución del script se ejecuta el comando `ftp`. A continuación se pulsa la telca '`o`' para abrir una conexión a un servidor ftp, se escribe `localhost` y se conecta al servidor local ftp. Para establecer la conexión se escribe el nombre del usuario del sistema `usuario-privado` y su contraseña. La conexión se ha establecido. Ahora dentro del servidor ftp, se ejecutan los comandos: `ls`, `pwd`, `help` y `quit`.

Una vez desestablecida la conexión se desinstala el servidor proftpd mediante el comando:

```
apt-get remove proftpd.
```

2.1.- Configuración de proftpd.

Su configuración es similar a la configuración del Servidor Apache, con lo cual si posees conocimientos sobre Apache tendrás mucho ganado, así tiene:

- ✓ Un fichero de configuración principal `/etc/protftpd/proftpd.conf`
- ✓ La posibilidad de configurar hosts virtuales (*dominio independiente que se puede alojar en un mismo servidor ftp*) o virtualhosts (*hosts virtuales*), de tal forma que un mismo servidor ftp puede alojar múltiples dominios

con sus configuraciones correspondientes, y todo lo que no esté incluido en la definición de cada virtualhost se heredará de la configuración principal.
- ✓ Configuración a través de directivas.
- ✓ Contextos de configuración: global, directorio, virtualhost, anonymous.
- ✓ Modularización. Al igual que Apache se pueden activar/desactivar funcionalidades a través de módulos.

En el siguiente enlace encontrarás la página web oficial de ProFTPD.
http://www.proftpd.org/
En el siguiente enlace encontrarás información sobre las directivas de ProFTPD.
http://www.proftpd.org/localsite/Userguide/linked/ref-directives.html

Una vez instalado ProFTPD existirán dos ficheros de especial interés:
- ✓ El fichero `/etc/ftpusers`, ya comentado, que contiene un lista de usuarios que no tienen permiso de acceso por FTP. Por razones de seguridad al menos los siguientes usuarios deberían estar listados en este fichero: `root`, `bin`, `uucp`, `news`. Ten en cuenta que las líneas en blanco y las líneas que comiencen por el carácter '`#`' serán ignoradas.
- ✓ El fichero de configuración principal (`/etc/proftpd/proftpd.conf`)

En el fichero `protfpd.conf`:
- ✓ Las líneas en blanco y las líneas que comiencen por el carácter '`#`' serán ignoradas.
- ✓ Las líneas que comienzan por `Include` recogerán la configuración de los ficheros que la acompañan.
- ✓ `User proftpd` y `Group nogroup` identifican al usuario y grupo con el que se ejecuta proftpd.
- ✓ El soporte `LDAP`, `SQL`, `TLS`, `virtualhosts` y `cuotas` están desactivados, ver líneas: `#Include /etc/proftpd/ldap.conf`, `#Include /etc/proftpd/sql.conf`, `#Include /etc/proftpd/tls.conf`, `#Include /etc/proftpd/virtuals.conf` y `QuotaEngine off`.
- ✓ El mensaje de bienvenida se encuentra en el fichero `welcome.msg`
- ✓ Está configurado por defecto el modo de conexión ftp activo en el puerto TCP 21.
- ✓ Los usuarios que puedan conectarse por ftp:
  - ➔ Necesitan una consola de comandos activa, esto es, debe poseer una consola presente dentro del fichero **`/etc/shells`**
  - ➔ Pueden moverse por todo el sistema de ficheros, esto es, no están encerrados (jaula chroot) en sus directorios `/home`, puesto que la directiva `DefaultRoot ~` está comentada. Por seguridad sería conveniente descomentar la línea y recargar la configuración del servidor.
- ✓ Para evitar ataques de denegación de servicio solamente se permiten 30 conexiones simultáneas: `MaxInstances 30`
- ✓ Los permisos para los ficheros y directorios creados en la conexión ftp son: `644` y `755` respectivamente, ya que, **`umask 022 022`**, donde el primer grupo de tres números identifican los permisos de los ficheros y el segundo grupo identifica los permisos de los directorios.
- ✓ Encontrarás al final del mismo un ejemplo de configuración para usuarios anónimos.

Una vez retocada la configuración del servidor proftpd sólo reconocerá estos cambios cuando recargues su configuración, con lo cual debes ejecutar el comando:

```
/etc/init.d/proftpd restart
```

Si la configuración es correcta, y no quieres reiniciar proftpd, puedes recargar la configuración mediante el comando:

```
/etc/init.d/proftpd reload
```

A continuación veremos distintas configuraciones del servidor proftpd.

2.2.- Configurar el servidor como ftp privado.

Una vez instalado el servidor proftpd, en Debian 6 (squeeze), disponemos de un archivo de configuración /etc/proftpd/proftpd.conf

Como has podido comprobar en la sección anterior, posee una configuración tipo por defecto. Ésta ya permite la conexión a tu servidor. ¿Con qué usuarios? Con cualquier usuario del sistema que posea una consola de comandos activa definida en `/etc/shells`.

¿Cómo? Pues simplemente ejecutando cualquier cliente ftp que establezca una conexión con el puerto TCP 21 a tu servidor ftp. Por ejemplo utilizando el cliente de comandos ftp sería:

```
ftp usuario_del_sistema@servidor_ftp
```

donde,
`servidor_ftp`: puede ser el nombre de tu servidor ftp en `/etc/hosts` o resuelto por DNS, o la IP de tu servidor ftp

Si no eres capaz de conectar revisa la configuración de tu cortafuegos y la sección 1.6 donde se exponen soluciones a problema típicos en conexiones ftp.

En el siguiente enlace encontrarás ejemplos de configuraciones del servidor ProFTPD.
http://www.proftpd.org/docs/example-conf.html

2.3.- Configurar el servidor como ftp privado y anónimo.

Igual te interesa contar en la configuración de tu servidor ftp con un usuario anónimo, el cual establecerá la conexión con cualquier contraseña y tendrá permisos diferentes a los usuarios del sistema (privados). Normalmente los permisos del usuario anónimo en un servidor ftp se establecerán para que solamente pueda moverse por los directorios y descargar archivos, nunca subirlos, esto es, normalmente el usuario anónimo no podrá crear ni eliminar ficheros y directorios.

Para hacer que proftpd permita conexiones mediante usuarios del sistema y usuarios anónimos debes modificar el fichero/etc/proftpd/proftpd.conf

La modificación consiste en retocar su configuración activando a mayores la conexión mediante usuarios anónimos, puesto que los usuarios del sistema por defecto ya poseen acceso mediante ftp y se conectan con la misma contraseña del sistema.

Por lo tanto, al final del fichero incorpora las siguientes líneas:

```
# Inicio de la configuración Anonymous
     # Usuario anónimo que entrará en el directorio ~ftp, esto es, en la variable $HOME del
usuario ftp
     # En Debian 6 (squeeze) ~ftp=/home/ftp . Este directorio será la raíz de los directorios
en la conexión
     # ftp, esto es, /home/ftp estará enjaulado (chroot) de tal forma que aunque el usuario
anónimo ftp
     # quisiera acceder a otros directorios situados fuera de /home/ftp no podrá acceder.
     <Anonymous ~ftp>
     # Después de hacer login anónimo mediante ftp el servidor se ejecuta con el usuario ftp y
con el grupo
     # nogroup
     User     ftp
     Group     nogroup
     # La siguiente línea permite hacer login con el usuario "anonymous" igual que si fuera el
usuario "ftp"
     UserAlias     anonymous ftp
     # Cambios de apariencia, todos los ficheros parecerán pertenecer al usuario y grupo ftp
     DirFakeUser     on ftp
     DirFakeGroup     on ftp
     # No es necesario tener una shell en /etc/shells
     RequireValidShell     off
```

```
     # Limitar el máximo número de logins anónimos concurrentes a 10
     MaxClients     10
     # Mensaje de conexión en el fichero welcome.msg
     DisplayLogin     welcome.msg
     # No permitir ESCRITURA en cualquier directorio al usuario anonymous, alias del usuario
ftp
     <Directory *>
      <Limit WRITE>
         DenyAll
      </Limit>
     </Directory>
     # Fin de la configuración Anonymous
     </Anonymous>
```

No olvides recargar la nueva configuración para que los cambios tengan efecto, ejecutando el comando:

```
/etc/init.d/proftpd restart ó /etc/init.d/proftpd reload.
```

Esta configuración permitirá conectar al servidor ftp mediante el usuario **ftp** o por su alias **anonymous** empleando una contraseña cualquiera. Una vez conectado solamente tendrá acceso al contenido de la carpeta /home/ftp y no podrá subir ni eliminar nada de ella.

2.4.- Configurar el servidor como ftp anónimo.

Puedes configurar proftpd para que permita conexiones mediante usuarios anónimos obligando a conectar sin poseer una contraseña del sistema, esto es, conectando con una contraseña cualquiera. Para ello debes modificar de nuevo el fichero /etc/proftpd/proftpd.conf

Por lo tanto, cambia la configuración del usuario anonymous ftp de tal forma que al final del fichero aparezcan las siguientes líneas:

```
<Anonymous ~ftp>
   User     ftp
   Group    nobody
   # No es necesario tener una shell en /etc/shells
   RequireValidShell          off
   # No se requiere contraseña en la conexión
   AnonRequirePassword        off
   # No permitir ESCRITURA en cualquier directorio al usuario ftp
   <Directory *>
      <Limit WRITE>
         DenyAll
      </Limit>
   </Directory>
</Anonymous>
```

Puedes crear un usuario anónimo con carácter privado, esto es, que requiera contraseña para establecer la conexión. Por ejemplo, en la configuración siguiente se convierte el usuario del sistema 'invitado', para el servidor ftp, en un usuario anónimo que requiere contraseña para establecer la conexión. Además, solamente tendrá permisos de escritura desde cualquier equipo que conecte mediante la dirección de red 192.168.200.

```
<Anonymous ~invitado>
   User     invitado
   Group    nobody
   # Se requiere la contraseña de sistema del usuario invitado en la conexión
   AnonRequirePassword        on
   # No permitir ESCRITURA en cualquier directorio al usuario invitado a no ser que establezca
conexión
   # de la red 192.168.200.
   <Directory *>
      <Limit WRITE>
         Order allow, deny
         Allow from 192.168.200.
         Deny from all
      </Limit>
   </Directory>
</Anonymous>
```

Puedes convertir cualquier usuario privado (del sistema) que posea una consola de comandos válida en `/etc/shell` en un usuario anónimo. Por ejemplo en las configuraciones anteriores sólo tendrías que sustituir el usuario '`ftp`' y el usuario '`invitado`' por el nombre de un usuario existente en el sistema operativo.

En el siguiente enlace encontrarás información sobre la directiva Order de ProFTPD.
http://www.proftpd.org/docs/directives/linked/config_ref_Order.html
En el siguiente enlace encontrarás información sobre la directiva Allow de ProFTPD.
http://www.proftpd.org/docs/directives/linked/config_ref_Allow.html
En el siguiente enlace encontrarás información sobre la directiva Deny de ProFTPD.
http://www.proftpd.org/docs/directives/linked/config_ref_Deny.html

Según lo visto anteriormente, ¿cómo permitirías al usuario invitado establecer conexión desde las redes: 192.168.200 y 10.0.200?
Añadiendo en la misma línea de la directiva **Allow from 192.168.200**. la nueva red separando las redes mediante una coma, tal que así: **Allow from 192.168.200.,10.0.200**.

Y si además, ¿quisieras permitir el acceso desde los dominios `tuhostA.tudominio.edu`**,** `tuhostB.tudominio.edu` **y** `tuhostC.tudominio.edu`**?**
Pues, de la misma forma que anteriormente, añadiendo en la misma línea de la directiva **Allow from 192.168.200** las nuevas redes o dominios separándolos mediante signos coma, tal que así:
Allow from 192.168.200., 10.0.200., tuhostA.tudominio.edu, tuhostB.tudominio.edu, tuhostC.tudominio.edu

2.5.- Configurar el servidor ftp con múltiples dominios.

Anteriormente hemos visto cómo poder configurar el servidor ftp con múltiples usuarios, pero todos pertenecientes al mismo sitio/dominio, entonces, ¿no se puede configurar usuarios pertenecientes a distintos dominios en el mismo servidor ftp? La respuesta es que sí, sí se puede, ¿cómo?, mediante la configuración de hosts virtuales o virtualhosts. Éstos básicamente lo que hacen es permitir que un mismo servidor ftp pueda alojar múltiples dominios, así configurando hosts virtuales podemos alojar: **empresa1.com**, **empresa2.com**,…, **empresaN.com** en el mismo servidor ftp. Cada empresa tendrá su virtualhost único e independiente de los demás.

Aunque como se ha comentado anteriormente cada virtualhost es único e independiente de los demás, todo aquello que no esté incluido en la definición de cada virtualhost se heredará de la configuración principal: **`proftpd.conf`** **(`/etc/proftpd/proftpd.conf`)**. Así, si quieres definir una directiva común en todos los virtualhost no debes modificar cada uno de los virtualhost introduciendo esa directiva sino que debes definir esa directiva en la configuración principal del servidor ftp ProFTPD, de tal forma que todos los virtualhost heredarán esa directiva, por ejemplo en `proftpd.conf` puedes encontrar la directiva **`TimeoutIdle 1200`**, que establece la directiva **`TimeoutIdle`** igual a 1200 segundos, esto es, indica el número máximo de segundos que puede estar un usuario sin hacer nada, pasado ese tiempo se cierra la conexión ftp.

En la definición de la directiva VirtualHost podemos poner la IP del servidor FTP ó bien el nombre DNS correspondiente. En nuestro escenario, la IP_Servidor_FTP=192.168.200.250, **`ftp.empresa1.com`** y **`ftp.empresa2.com`** identifican a la misma máquina.

Hay que tener en cuenta que si las IP empleadas son **IP privadas**, sin existencia en Internet, siempre que se haga referencia a las mismas a través de nombre de dominios, deberá existir un **servidor DNS**

que las resuelva en local o bien, en su defecto, deberán existir las entradas correspondientes en el fichero del sistema local `/etc/hosts`.

Independientemente de si configuras virtualhosts basados en IP o en nombre, puedes utilizar usuarios del sistema, pero también puedes crear los usuarios virtuales que quieras en un fichero similar a **`/etc/passwd`** y llamarlo en la configuración mediante la directiva **`AuthUserFile`**, entonces:

✓ Ejecuta el siguiente comando que creará un fichero de autenticación para usuarios virtuales,

```
ftpasswd --passwd --name user-empresa1 --file /etc/passwd.usuarios.virtuales --uid 107 --home
/var/ftp/empresa1 --shell /bin/false
```

donde,

- `ftppasswd`, es el comando que permite crear los usuarios virtuales.
- `--passwd`, es el parámetro que pedirá la contraseña del usuario.
- `--name user-empresa1`, identifica al usuario virtual de nombre `user-empresa1`.
- `--file /etc/passwd.usuarios.virtuales`, creará, en caso de no existir, o modificará, en caso de existir el fichero de autenticación de usuarios virtuales.
- `--uid 107`, es el identificador perteneciente al usuario del sistema **ftp**. Se puede saber ejecutando el comando: `id ftp` .
- `--home /var/ftp/empresa1`, identifica a donde se conecta el usuario.
- `--shell /bin/false`, identifica una consola de comandos que no permite conexión como usuario del sistema.

Ejecuta también el comando:

```
ftpasswd --passwd --name user-empresa2 --file /etc/passwd.usuarios.virtuales --uid 107 --home
/var/ftp/empresa2 --shell /bin/false
```

A continuación prosigues, dependiendo si deseas configurar virtualhosts basados en IP o virtualhosts basados en nombre.

2.6.- Virtualhosts basados en nombre.

En la definición de la directiva `VirtualHost` podemos poner la IP del servidor FTP ó bien el nombre DNS correspondiente. En nuestro escenario, la `IP Servidor FTP=192.168.200.250`, `ftp.empresa1.com` y `ftp.empresa2.com` identifican a la misma máquina y a la misma IP. Ahora si, cada virtualhost, así como el servidor principal, deben servir en un puerto TCP distinto.

¿Cómo lo haces? Sigues el procedimiento:

1. En la configuración de ProFTPD (`/etc/proftpd/proftpd.conf`) debes activar la configuración del fichero **`virtuals.conf`** descomentando la línea:

```
Include /etc/proftpd/virtuals.conf
```

2. Agrega la configuración virtualhost para empresa1 en el fichero **`/etc/proftpd/virtuals.conf`**

```
<VirtualHost 192.168.200.250>
      Port 2121
      ServerName "Servidor FTP empresa1"
      AuthUserFile /etc/passwd.usuarios.virtuales
      DefaultRoot /var/ftp/empresa1/
      RequireValidShell off
</VirtualHost>
```

3. Agrega la configuración virtualhost para empresa2 en el fichero **`/etc/proftpd/virtuals.conf`**

```
<VirtualHost ftp.empresa2.com>
      Port 2122
      AuthUserFile /etc/passwd.usuarios.virtuales
      ServerName "Servidor FTP empresa2"
      DefaultRoot /var/ftp/empresa2/
      RequireValidShell off
</VirtualHost>
```

4. Configura permisos en las carpetas `/var/ftp/empresa1/` y `/var/ftp/empresa2/` para los usuarios virtuales:

```
chown ftp /var/ftp/empresa1/ /var/ftp/empresa2/
```

5. Recarga la configuración del servidor.

```
/etc/init.d/proftpd restart
```

Explicación fichero virtualhost:

`<VirtualHost IP_Servidor_FTP>` → Inicio etiqueta **virtualhost**: define la IP del servidor ftp. También puede ser `<VirtualHost Nombre_DNS_Servidor_FTP>`

`Port numero` → Identifica el puerto TCP por el que espera la conexión el servidor FTP

`ServerName "Servidor FTP empresaX"` → Configura el nombre que se muestra en la conexión de los usuarios.

`DefaultRoot /var/ftp/empresaX/` → Definición de la ruta que sirve ProFTPFD, en este caso: `/var/ftp/empresaX/` mediante la directiva `DefaultRoot`, esto es, indica que los usuarios cuando conecten con el servidor ftp estarán enjaulados en la ruta `/var/ftp/empresaX/`, con lo cual no podrán acceder a otro directorio que no esté contenido dentro de éste.

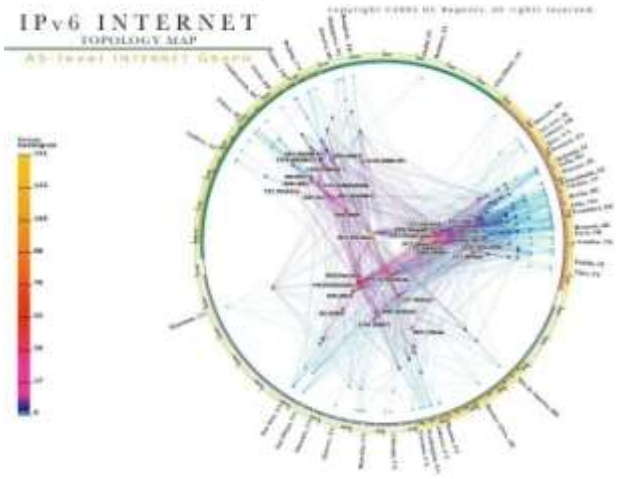
`RequireValidShell off` → No es necesario tener una Shell declarada en el fichero `/etc/shells`

`</VirtualHost>` → Fin de la etiqueta `VirtualHost`: fin de la definición de este virtualhost para la empresa1.

2.7.- Virtualhosts basados en IP.

En la definición de la directiva `VirtualHost` podemos poner la IP del servidor FTP o bien el nombre DNS correspondiente. En nuestro escenario, la `IP_Servidor_FTP=192.168.200.250`, `ftp.empresa1.com` y `ftp.empresa2.com` identifican a la misma máquina y a distintas IP. Ahora es indiferente el puertoTCP en el que sirve cada virtualhost, así como el servidor principal, esto es, puede ser el mismo o no ya que ahora cada puerto está relacionado con una IP distinta.

La IP que debemos poner ahora en la definición de la directiva `Virtualhost` cambia, cada IP corresponde a una interfaz de red del servidor FTP, en nuestro escenario: `IP_Servidor_FTP=192.168.200.250`, `ftp.empresa1.com` se identifica con `192.168.200.251` y `ftp.empresa2.com` con `192.168.200.252`



Este método no aporta ventajas sobre el anterior, es más, aún puede ser más difícil de mantener si las IP del servidor FTP se modifican con cierta frecuencia.

¿Cómo lo haces? Sigues el mismo procedimiento usado para los virtualhost basados en nombre, únicamente se diferencia en la configuración de losvirtualhost, así:

1. Modifica la configuración virtualhost para empresa1 en el fichero `/etc/proftpd/virtuals.conf`

```
<VirtualHost 192.168.200.251>
      ServerName "Servidor FTP empresa1"
      AuthUserFile /etc/passwd.usuarios.virtuales
      DefaultRoot /var/ftp/empresa1/
      RequireValidShell off
</VirtualHost>
```

2. Agrega la configuración virtualhost para empresa2 en el fichero `/etc/proftpd/virtuals.conf`

```
<VirtualHost ftp.empresa2.com>
      AuthUserFile /etc/passwd.usuarios.virtuales
      ServerName "Servidor FTP empresa2"
```

```
                        DefaultRoot /var/ftp/empresa2/
                        RequireValidShell off
                    </VirtualHost>
```

3. Configura permisos en las carpetas `/var/ftp/empresa1/` y `/var/ftp/empresa2/` para los usuarios virtuales:

```
chown ftp /var/ftp/empresa1/ /var/ftp/empresa2/
```

4. Recarga la configuración del servidor.

```
/etc/init.d/proftpd restart
```

Explicación fichero virtualhost:

`<VirtualHost IP_Servidor_FTP>` → Inicio etiqueta virtualhost: define la IP del servidor ftp. También puede ser **`<VirtualHost Nombre_DNS_Servidor_FTP>`**

`ServerName "Servidor FTP empresaX"` → Configura el nombre que se muestra en la conexión de los usuarios.

`DefaultRoot /var/ftp/empresaX/` → Definición de la ruta que sirve ProFTPFD, en este caso: `/var/ftp/empresaX/` mediante la directiva **`DefaultRoot`**, esto es, indica que los usuarios cuando conecten con el servidor ftp, estarán enjaulados en la ruta `/var/ftp/empresaX/`, con lo cual no podrán acceder a otro directorio que no esté contenido dentro de éste.

`RequireValidShell off` → No es necesario tener una shell declarada en el fichero `/etc/shells` .

`</VirtualHost>` → Fin de la etiqueta **`VirtualHost`**: fin de la definición de este virtualhost para la empresa1.

2.8.- Cuotas de disco para los usuarios (I).

La capacidad de almacenamiento no es infinita, por lo tanto será interesante saber cómo crear cuotas de disco para los usuarios y ya puestos para los usuarios en los virtualhosts.

El archivo **`/etc/proftpd/proftpd.conf`** llama mediante la directiva **`Include`** al archivo **`/etc/proftpd/modules.conf`** en el que están activadas las cuotas (**`LoadModule mod_quotatab.c`**, **`LoadModule mod_quotatab_file.c`**), luego para activarlas tienes que sustituir en el archivo **`/etc/proftpd/proftpd.conf`** el código:

```
<IfModule mod_quotatab.c>
   QuotaEngine off
</IfModule>
```

por el código siguiente:

```
<IfModule mod_quotatab.c>
   QuotaEngine on
   QuotaLog /var/log/proftpd/quota.log
   <IfModule mod_quotatab_file.c>
      QuotaLimitTable file:/etc/proftpd/ftpquota.limittab
      QuotaTallyTable file:/etc/proftpd/ftpquota.tallytab
   </IfModule>
</IfModule>
```

donde,

`<IfModule mod_quotatab.c> … </IfModule>` → Indica que si el módulo `mod_quotatab.c` está cargado en el archivo `/etc/proftpd/modules.conf` se realizarán las directivas que contengan.

`QuotaEngine on` → Activa las cuotas.

`QuotaLog /var/log/proftpd/quota.log` → Indica el archivo de registro sobre cuotas.

`<IfModule mod_quotatab_file.c> … </IfModule>` → Indica que si el módulo `mod_quotatab_file.c` está cargado en el archivo

`/etc/proftpd/modules.conf` se realizarán las directivas que contengan.

`QuotaLimitTable file:/etc/proftpd/ftpquota.limittab` → Indica el archivo sobre el límite de cuotas `Limit`.

`QuotaTallyTable file:/etc/proftpd/ftpquota.tallytab` → Indica el archivo sobre el límite de cuotas `Tally`.

Para ProtFTPD existen básicamente dos tipos de cuotas: `limit` y `tally`.

- ✓ `Limit`: Es la cuota que te interesa si estás pensando en restringir el espacio en disco a los usuarios. Éste puede ser `soft`, cuando existe un espacio de gracia(tamaño en bytes) que puede sobrepasar el límite, o `hard` cuando no existe un espacio de gracia.
- ✓ `Tally`: Utilizado cuando quieres limitar el número de ficheros que se utilizan.

Para mayor información sobre las cuotas puedes visitar la documentación oficial de ProFTPD sobre `mod_quotatab`.

http://www.proftpd.org/docs/contrib/mod_quotatab.html

La forma más sencilla de crear las cuotas es hacer el símil `upload=espacio en disco`, con lo cual los archivos subidos no pueden ocupar más del que queramos darle como espacio en disco, esto es, los bytes subidos funcionan como espacio en disco, ya que no existe diferencia entre ellos, pues los bytes cargados a través de FTP se almacenan automáticamente en el disco, por lo que deberías emplear la cuota tipo `limit`.

Puedes crear las cuotas mediante el comando ftpquota:

```
# ftpquota --create-table --type=limit --table-path=/etc/proftpd/ftpquota.limittab
# ftpquota --create-table --type=tally --table-path=/etc/proftpd/ftpquota.tallytab
```

Por ejemplo, si quisieras limitar a un usuario de nombre `user-empresa1` el espacio de subida en 4 GB:

```
# ftpquota --add-record --type=limit --name=user-empresa1 --quota-type=user \
 --bytes-upload=4 --units=Gb --table-path=/etc/proftpd/ftpquota.limittab
```

Y si quisieras limitar la subida y bajada a 4 GB y 2 GB respectivamente al usuario `user-empresa2`:

```
# ftpquota --add-record --type=limit --name=user-empresa2 --quota-type=user \
 --bytes-upload=4 --bytes-download=2 --units=Gb --table-path=/etc/proftpd/ftpquota.limittab
```

2.8.1.- Cuotas de disco para los usuarios (II).

Bien, pero, ¿cómo verificar el funcionamiento de las cuotas?. Y si quisieras comprobar la cuota de un usuario, ¿es posible? ¿Y si quisieras actualizarla?¿Y desactivarlas para algún usuario?¿Y borrarlas? Pues, utilizas el comando `ftpquota` como sigue:

- ✓ Para ver los registros de cuotas, esto es, a quién se le está ejerciendo las cuotas:

```
# ftpquota --show-records --type=limit --table-path=/etc/proftpd/ftpquota.limittab
-----------------------------------------
 Name: user-empresa1
 Quota Type: User
 Per Session: False
 Limit Type: Hard
 Uploaded bytes: 4294967296.00
 Downloaded bytes: unlimited
 Transferred bytes: unlimited
 Uploaded files: unlimited
 Downloaded files: unlimited
 Transferred files: unlimited
```

- ✓ Para actualizar la cuota de un usuario, por ejemplo, `user-empresa1`:

```
# ftpquota --update-record --type=limit --name=user-empresa1 --quota-type=user \
 --bytes-upload=2300 --units=Mb --table-path=/etc/proftpd/ftpquota.limittab
```

con lo cual, si compruebas de nuevo los registros, verás que los cambios surgieron efecto:

```
# ftpquota --show-records --type=limit --table-path=/etc/proftpd/ftpquota.limittab
-----------------------------------------
 Name: user-empresa1
 Quota Type: User
 Per Session: False
 Limit Type: Hard
 Uploaded bytes: 2411724800.00
 Downloaded bytes: unlimited
 Transferred bytes: unlimited
 Uploaded files: unlimited
 Downloaded files: unlimited
 Transferred files: unlimited
```

✓ Para desactivar la cuota de un usuario debes borrar el registro, por ejemplo, user-empresa1:

```
# ftpquota --delete-record --type=limit --name=user-empresa1 --quota-type=user
```

con lo cual, si compruebas de nuevo los registros, verás que los cambios surgieron efecto:

```
# ftpquota --show-records --type=limit --table-path=/etc/proftpd/ftpquota.limittab
ftpquota: (empty table)
```

Puedes ver la ayuda del comando `ftpquota` mediante: `ftpquota --help`.
No olvides recargar la configuración del servidor ProFTPD: `/etc/init.d/proftpd restart`.

2.9.- Acceso seguro mediante TLS.

En Debian 6 (**`squeeze`**) al instalar el paquete proftpd ya se puede establecer la conexión por TLS, siempre y cuando se configure el archivo **`/etc/proftpd/tls.conf`** y procedas como sigue:

1. Edita el archivo **`/etc/proftpd/proftpd.conf`** y descomenta la línea:

```
Include /etc/proftpd/tls.conf
```

2. Crea las claves, pública y privada, para la conexión cifrada:
   - ✓ Método 1: Instalación del paquete openssl (*Paquete de herramientas de administración y bibliotecas relacionadas con la criptografía, que suministran funciones criptográficas, entre otros a navegadores web, para acceso seguro a sitios mediante el protocolo HTTPS*) y ejecución del comando `openssl`.

```
# apt-get install openssl
# openssl req -x509 -newkey rsa:1024 -keyout /etc/ssl/private/proftpd.key -out \
 /etc/ssl/certs/proftpd.crt -nodes -days 3650
Generating a 1024 bit RSA private key
..++++++
.................................++++++
writing new private key to '/etc/ssl/private/proftpd.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Madrid
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:EMPRESA1
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []: ftp.empresa1.com
Email Address []:
```

   - ✓ Método 2: Comando `proftpd-gencert`. Los dos comandos anteriores se resumen en uno, con la salvedad que el certificado será válido solamente durante 1 año y no 10:

```
# proftpd-gencert
```

3. Modifica los permisos:

```
# mv /etc/ssl/private/proftpd.key /etc/ssl/
# chmod 0600 /etc/ssl/proftpd.key
# chmod 0644 /etc/ssl/certs/proftpd.crt
```

4. Modifica el fichero **`/etc/proftpd/tls.conf`** como se indica en el fichero ejemplo tls.conf
5. Recarga la configuración del servidor ProFTPD:

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

6. Comprueba la configuración mediante el usuario del servidor ftp **`invitado`**, creado anteriormente, con un cliente FTPES, es decir, un cliente ftp que permita la conexión por TLS como FileZilla.

Puedes verificar en tiempo real las conexiones con el servidor ftp revisando los archivos de registro mediante los comandos:

```
tail -f /var/log/proftpd/proftpd.log
tail -f /var/log/proftpd/tls.log
```

7. Si deseas, puedes hacer valer la configuración para todos los usuarios, incluso aquellos pertenecientes a virtualhost, modificando el fichero `/etc/proftpd/tls.conf` como se indica en el fichero ejemplo tls2.conf.

Te proponemos el siguiente enlace de un vídeo práctico sobre cómo configurar TLS en el servidor ProFTPD y cómo configurar una plantilla deFileZilla que soporta conexión TLS. La configuración se realiza sobre una distribución GNU/Linux basada en Debian.

http://www.youtube.com/watch?feature=player_embedded&v=jvdR5nZ30gE

Resumen:

Se ve una consola de comandos del usuario **root**, donde se muestra el contenido del script `Instalacion_ProFTPD_TLS_Filezilla_FTPES.sh`, a saber:

```
#!/bin/bash
# Actualizar repositorios
apt-get update
# Actualizar sistema operativo
apt-get upgrade
# Buscar paquete proftpd
apt-cache search proftpd
# Instalar paquete proftpd-basic
apt-get install proftpd-basic
# Ruta de configuración de ProFTPD: /etc/proftpd
ls -l /etc/proftpd
# Servicio proftpd: Posibilidades
/etc/init.d/proftpd
# Servicio proftpd: Posibilidades
service proftpd

# Con esta configuración cualquier usuario del sistema puede acceder por ftp

# Modificar /etc/proftpd/proftpd.conf y descomentar la línea: Include /etc/proftpd/tls.conf
TEMPORAL=`mktemp`
cat    /etc/proftpd/proftpd.conf    |    sed    "s/\#Include    \/etc\/proftpd\/tls.conf/Include
\/etc\/proftpd\/tls.conf/g" > $TEMPORAL
mv $TEMPORAL /etc/proftpd/proftpd.conf

# Modificar /etc/proftpd/tls.conf para que tenga el siguiente contenido:
cat > /etc/proftpd/tls.conf << EOF
##########################################Fichero
/etc/proftpd/tls.conf################################################
#
# Proftpd sample configuration for FTPS connections.
#
# Note that FTPS impose some limitations in NAT traversing.
# See http://www.castaglia.org/proftpd/doc/contrib/ProFTPD-mini-HOWTO-TLS.html
# for more information.
#

<IfModule mod_tls.c>
<global>
TLSEngine on
TLSLog /var/log/proftpd/tls.log
</global>


TLSProtocol SSLv23

<global>
TLSRSACertificateFile /etc/ssl/certs/proftpd.crt
TLSRSACertificateKeyFile /etc/ssl/private/proftpd.key
TLSOptions NoCertRequest
TLSVerifyClient off
TLSRequired on
TLSRenegotiate required off
</global>
</IfModule>
```

```
##########################################Fin
/etc/proftpd/tls.conf##################################################

EOF

# Generar el certificado
proftpd-gencert

# Puedes recargar la nueva configuración con:
## /etc/init.d/proftpd reload
## ó
## service proftpd reload

# Puedes reiniciar el servicio mediante:
/etc/init.d/proftpd restart
## ó
## service proftpd restart

# Con esta nueva configuración puedes acceder mediante FTPES
# Para ello crearemos una Plantilla en FileZilla:
## 1) Ver script FileZilla_Perfiles.sh en el vídeo:
## FileZilla. Crear sitios (perfiles) --> http://www.youtube.com/watch?v=nBm-2rgkf5Y
## 2) Arrancar FileZilla como usuario del sistema alumno.
 su -c filezilla alumno

# Para desinstalar proftpd
# apt-get remove proftpd
Se ejecuta el script mediante el comando:
sh Instalacion_ProFTPD_TLS_FileZilla_FTPES.sh
```

Antes de finalizar la ejecución se escribe, como consecuencia del comando `proftpd-gencert` para la generación del certificado:

```
Country Name: ES
Locality Name: Madrid
Organization Name: Empresa
Common Name: ftp.empresa.ftpes.local
```

Una vez acabada la ejecución del script aparece el cliente ftp gráfico **FileZilla**. En la interface arriba a la izquierda aparece el primer icono **Abrir el Gestor de Sitios**. Se hace clic en el mismo y aparece un panel con dos secciones: la de la izquierda donde aparecen los Sitios configurados y la de la derecha donde aparecen las opciones configuradas de cada Sitio Seleccionado:
En la sección de la izquierda aparecen dos sitios configurados, estando seleccionado el sitio **`FTPES-PLANTILLA`** y monstrándose a la derecha la configuración del mismo:

- ✓ En la caja de texto Servidor: `localhost`.
- ✓ En `Server Type`: `FTPES` – FTP sobre TLS/SSL explícito.
- ✓ En `Logon Type`: Preguntar la contraseña.
- ✓ En la caja de texto Usuario: `alumno`.

y se pulsa el botón **Aceptar**, desapareciendo el panel de configuración.

Se vuelve a pulsar en el icono **Abrir el Gestor de Sitios** y aparece el nuevo sitio (perfil) configurado **`FTPES-PLANTILLA`**. Ahora se pulsa el botón **Conectar** en la sección de la derecha del panel y aparece una caja de texto preguntando la contraseña del usuario alumno, se escribe, acepta y aparece el certificado digital de conexión a la espera de aceptarlo, se acepta y se establece la conexión con el servidor **`localhost`**.

A continuación se accede al Gestor de Sitios y se borra la plantilla **`FTPES-PLANTILLA`** haciendo clic en el botón **Borrar** y aceptando la confirmación de borrado, luego se accede de nuevo al Gestor de Sitios para pulsar en el botón Nuevo Sitio, apareciendo una caja de texto donde se escribe el nombre del sitio a configurar (nombre del perfil a guardar). Se escribe **`FTPES-PLANTILLA`** y se activa la sección de la derecha.

En la sección de la derecha, al tener seleccionado **`FTPES-PLANTILLA`**, se escriben de nuevo los parámetros anteriores:

- ✓ En la caja de texto Servidor: `localhost`
- ✓ En `Server Type`: `FTPES` – FTP sobre TLS/SSL explícito
- ✓ En `Logon Type`: Preguntar la contraseña
- ✓ En la caja de texto Usuario: `alumno`

y se pulsa el botón **Aceptar**, desapareciendo el panel de configuración.

Se hace clic en el icono **Desconectar del servidor actualmente visible** y se vuelve a pulsar en el icono **Abrir el Gestor de Sitios** y aparece el nuevo sitio(perfil) configurado `FTPES-PLANTILLA`. Ahora se pulsa el botón **Conectar** en la sección de la derecha del panel y se establece la conexión con el servidor `localhost`.

Anexo I - PAM

¿Qué es PAM?

La idea original de los Pluggable Authentication Modules, en adelante PAM, fue de Sun y sus especificaciones se encuentran recogidas en RFC 86.0. Sin embargo, muchos otros sistemas adoptaron esta solución y cuentan desde hace tiempo con sus propias implementaciones.

En este sentido, GNU/Linux no es una excepción y, gracias a Red Hat, disfruta ya desde hace años de la funcionalidad que ofrece Linux-PAM (*se hará referencia "PAM" y "Linux-PAM" indistintamente*)

Pero, ¿qué es PAM exactamente? Tal y como puede leerse en la FAQ (*http://www.kernel.org/pub/linux/libs/pam/FAQ*) oficial del proyecto, PAM es, básicamente, un mecanismo flexible para la autenticación de usuarios. Y quizás esta característica, la flexibilidad, sea su aportación más importante.

A lo largo de los años, desde los primeros sistemas UNIX, los mecanismos de autenticación han ido evolucionando y han aparecido nuevas opciones: desde mejoras del clásico `/etc/passwd` —como la shadow— hasta dispositivos hardware orientados a la autenticación. Y, claro está, cada vez que aparecía y se popularizaba un nuevo método, los desarrolladores debían modificar sus programas para darles soporte.

PAM permite el desarrollo de programas independientes del mecanismo de autenticación a utilizar. Así es posible que un programa que aproveche las facilidades ofrecidas por PAM sea capaz de utilizar desde el sencillo `/etc/passwd` hasta dispositivos hardware —como lectores de huella digital—, pasando por servidores LDAP (*Lightweight Directory Access Protocol*) o sistemas de gestión de bases de datos. Y, por supuesto, todo esto sin cambiar ni una sola línea de código.

Pero PAM va más allá todavía, permitiendo al administrador del sistema construir políticas diferentes de autenticación para cada servicio.

En resumen, podrían sintetizarse las ventajas más importantes de PAM en los siguientes puntos:

- ✓ Ofrece un esquema de autenticación común y centralizado.
- ✓ Permite a los desarrolladores abstraerse de las labores de autenticación.
- ✓ Facilita el mantenimiento de las aplicaciones.
- ✓ Ofrece flexibilidad y control tanto para el desarrollador como para el administrador de sistema.

Grupos de gestión

A pesar de lo que se ha explicado anteriormente, la misión de PAM no es, únicamente, comprobar que un usuario es quien dice ser —autenticación—. Su alcance es mucho mayor y pueden dividirse sus tareas en cuatro grupos independientes de gestión, cada uno de los cuáles se encarga de un aspecto diferente de los servicios restringidos.

`account` (cuenta) En este grupo se engloban tareas que no están relacionadas directamente con la autenticación. Algunos ejemplos son permitir/denegar el acceso en función de la hora, los recursos disponibles o, incluso, la localización. Ofrece verificación de cuentas de usuario. Por ejemplo, se encarga de determinar si el usuario tiene o no acceso al servicio, si su contraseña ha caducado, etc.

`authentication` (autenticación) Tareas encaminadas a comprobar que, efectivamente, el usuario es realmente quien dice ser. A menudo, cuando se habla de PAM, sólo se tiene en cuenta esta tarea,

ignorando las demás. Estas tareas ofrecen incluso un sistema de credenciales que permiten al usuario ganar ciertos privilegios —fijados por el administrador—.

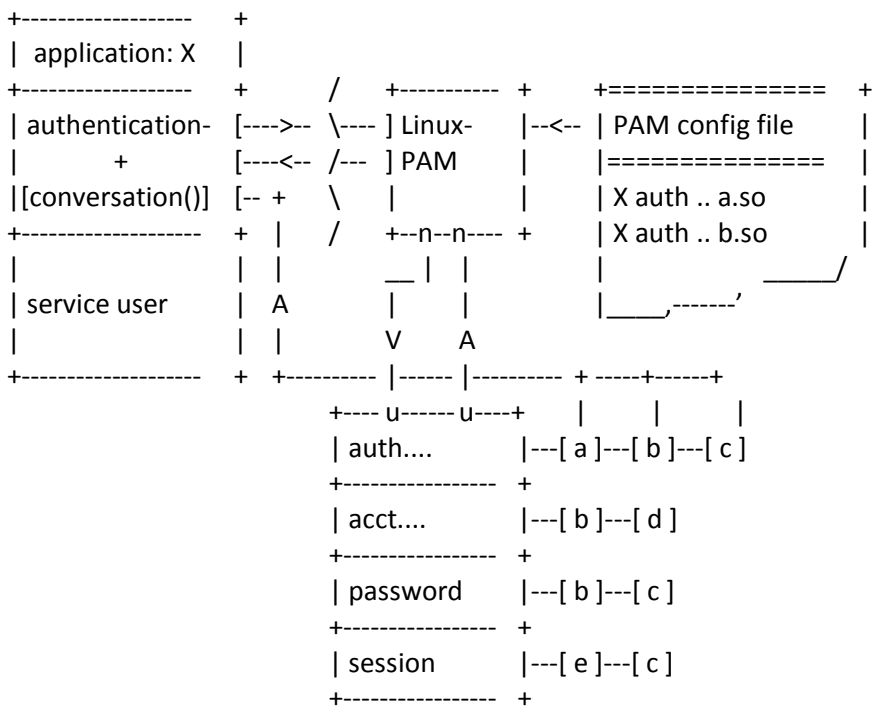
`password` (contraseña) Se encarga de mantener actualizado el elemento de autenticación asociado a cada usuario —por ejemplo, su contraseña—. Acciones como comprobar la fortaleza de una clave son típicas de este grupo.

`session` (sesión) En este grupo se engloban tareas que se deben llevar a cabo antes de iniciarse el servicio y después de que este finalice. Es especialmente útil para mantener registros de acceso o hacer accesible el directorio home del usuario.

Arquitectura

Hasta ahora se ha estudiado cuál es la misión de PAM y qué grupos de tareas lleva a cabo. Así que, la siguiente pregunta es: ¿cómo se organizan todas estas ideas?

La figura ilustra de forma clara su arquitectura.



Supóngase que una aplicación X quiere hacer uso de las facilidades ofrecidas por PAM. Para ello, interactúa con la biblioteca de Linux-PAM, sin tener que conocer ningún detalle acerca de como está configurado el sistema para la aplicación X. Será precisamente esta biblioteca quien se encargue de leer la configuración de PAM para conocer qué política de autenticación ha de aplicarse —combinando de forma conveniente una serie de módulos—.

Los módulos se colocan en una pila según el grupo de gestión y el orden en el que aparecen en la configuración —un módulo puede pertenecer a varios grupos—, para ser utilizados por PAM cuando corresponda. Este aspecto es tremendamente importante, ya que como se verá más adelante, el orden de los módulos en la pila va a determinar, en gran medida, el comportamiento de PAM para un servicio dado. En la figura, para la tarea de autenticación, se invocará primero al módulo a, luego a b y, finalmente, a c (*Pueden existir variaciones en el flujo, pero por el momento no se van a tener en cuenta*).

Finalmente, PAM ofrece a la aplicación una serie de funciones para llevar a cabo las diferentes tareas de cada grupo (autenticar, abrir sesión, etc.), mientras que la aplicación brinda a PAM una función de conversación destinada a intercambiar información textual.

Gracias a esta función, PAM se libera de tener que preocuparse de cómo enviar/recibir información del cliente —cuadros de diálogo, intercambio en un terminal, protocolos de red, etc.—.

Configuración

Enfoques de la organización de la configuración

La configuración de Linux-PAM puede organizarse siguiendo dos esquemas diferentes:

- ✓ Poner toda la configuración en el fichero /etc/pam.conf —la más antigua—.
- ✓ Colocar la configuración de cada servicio en ficheros separados bajo el directorio `/etc/pam.d/`. Este tipo de organizaciones ha ido ganando adeptos, y proyectos como Apache también hacen uso de un esquema de este tipo.
- ✓ Una combinación de las dos anteriores: por ejemplo, leer primero `/etc/pam.d/` y luego `/etc/pam.conf` (*como en el caso de Red Hat Linux*).

Afortunadamente, en todos los casos se utiliza la misma sintaxis, con la salvedad de que en el caso de `/etc/pam.conf` hay que indicar el servicio al que pertenece cada directiva (*con /etc/pam.d/ esta información está implícita en el nombre del fichero de configuración*).

En aras de la simplicidad, en adelante se supondrá que se usa el enfoque combinado, si bien este es un detalle que, conceptualmente, carece de importancia.

```
$ ls /etc/pam.d/
chage     chsh      groupadd     kde-np     other      shadow     su         system-auth     xdm
chfn      cups      kde          login      passwd     sshd       sudo       useradd         xserver
```

Organización de la configuración en ficheros separados

Reglas

Los ficheros de configuración de PAM están compuestos por una serie de reglas a aplicar, una por línea (*aunque pueden dividirse en varias usando el caracter *). Se ignoran todas las líneas que comiencen por el caracter '`#`'.

Todas estas reglas tienen la siguiente forma:

```
servicio* tipo control ruta [argumentos]
```

A continuación, se va a profundizar en el significado y la sintaxis de cada uno de estos campos. Hay que mencionar que, exceptuando los casos de ruta y argumentos (*depende del módulo*), en el resto de los campos no se hacen distinciones entre mayúsculas y minúsculas.

Pero antes, y para que lector se haga una idea de la estructura de este fichero, un pequeño ejemplo. Esta configuración pertenece al servicio login de Red Hat Linux `/etc/pam.d/login`.

```
#%PAM-1.0
auth       required pam_securetty.so
auth       required pam_stack.so service=system-auth
auth       required pam nologin.so
account    required pam_stack.so service=system-auth
password   required pam_stack.so service=system-auth
session    required pam_stack.so service=system-auth
session    optional pam_console.so
```

Servicio

Este primer campo indica el nombre de la aplicación/servicio correspondiente, como por ejemplo `login`, `su` o `smtp`. Tal y como se explicó anteriormente, se utiliza sólo en `/etc/pam.conf`, ya que en el caso de la configuración mediante ficheros individuales el nombre servicio se encuentra implícito en el del fichero.

Existe un nombre de servicio reservado, `other`, que se utiliza para especificar reglas por defecto. Así, en el caso de que, por ejemplo, no hubiese reglas para `smtp`, se le aplicarían las asociadas al "servicio" `other`.

Tipo

Como se vio anteriormente, PAM puede dividir su actividad en cuatro tareas de gestión. Y tipo expresa, precisamente, el área al que se destina esta regla. Puede adoptar uno de estos valores: `auth`, `account`, `session` o `password`. Como ya se había mencionado con anterioridad, los módulos correspondientes a un mismo área forman una pila.

Control

Este campo indica a PAM qué hacer en caso de éxito/fallo del módulo de la regla en cuestión, ejecutándose en serie todos los del mismo tipo. El orden en la pila hay que tenerlo muy en cuenta a la hora de determinar el valor adecuado para este campo.

Actualmente, pueden usarse dos sintaxis: la más simple consiste en una sola palabra clave, mientras que la otra (*más precisa y compleja*) implica el uso de corchetes y parejas valor-acción. Para facilitar la compresión, en primer lugar se va a estudiar la sintaxis más simple para, a continuación, explorar su alternativa compleja.

Usando la sintaxis sencilla, este campo puede tomar únicamente cuatro valores diferentes:

- ✓ `required` Indica que es necesario que el módulo tenga éxito para que la pila también lo tenga. Si se produce un fallo, no se notifica hasta que se procesa el resto de la pila.
- ✓ `requisite` En esencia, es igual que el anterior, con la diferencia de que en caso de fallo, el control se devuelve inmediatamente a la aplicación.
- ✓ `sufficient` El éxito en este módulo, si no se ha producido un fallo en los procesados anteriormente en la pila, es "suficiente". Llegados a este punto, el procesamiento se detiene (*ignorando incluso posibles required posteriores*). Un fallo no siempre resulta definitivo para la pila.
- ✓ `optional` Por lo general, PAM ignora los módulos marcados con este indicador. Su valor será tenido en cuenta sólo en caso de que no se haya llegado a ningún valor concreto de éxito o fracaso (*por ejemplo, PAM IGNORE*).

La sintaxis alternativa de configuración ofrece mayor flexibilidad y funcionalidad, pero introduce cierta complejidad que en muchos casos no es deseable. Se basa en asociar parejas valor-acción. Así, pueden asociarse acciones con los valores devueltos por los módulos.

```
[valor1=acción1 valor2=acción2 valor3=acción3 ...]
```

Los posibles valores de $valor_i$ se encuentran documentados aquí.

En lo que a las partes derechas se refiere, $accion_i$ puede ser un número entero positivo n o cualquiera de estos valores:

`ignore` El valor que devuelve este módulo no se tiene en cuenta.

`bad` Si se trata del primer módulo en fallar, el valor que devuelva la pila será el que devuelva este módulo.

`die` Equivalente al anterior, pero en este caso se devuelve el control a la aplicación de inmediato.

`ok` Si hasta el momento, el estado de la pila conduce a un éxito (`PAM_SUCCESS`), el código que devuelve será sobreescrito por el de este módulo. En caso contrario, deja el estado tal y como se lo encuentra.

`done` Funciona del mismo modo que ok, con la salvedad de que, llegado a este punto, devolverá el control a la aplicación.

`reset` Se olvida el estado de la pila hasta el momento y se sigue con el siguiente módulo de la pila.

El valor entero positivo n que antes se mencionó, indica que deben saltarse los siguientes n módulos. Esto permite al administrador tener, en cierto modo, control sobre el flujo de ejecución del proceso.

Para ilustrar la relación entre una y otra sintaxis de configuración, a continuación se expresan los posibles valores de la sintaxis sencilla en función de expresiones de la sintaxis compleja.

required	equivale a	`[success=ok newauthok_reqd=ok ignore=ignore default=bad]`.
requisite	equivale a	`[success=ok newauthok_reqd=ok ignore=ignore default=die]`.
sufficient	equivale a	`[success=done new authok reqd=done default=ignore]`.
optional	equivale a	`[success=ok new authtok reqd=ok default=ignore]`.

Ruta

Este campo contiene la ruta del módulo que se va a utilizar: si empieza por el carácter '`/`' se indica una ruta absoluta. En otro caso, será relativa a `/lib/security/`.

Argumentos

Se trata de argumentos que pueden ser pasados al módulo para su operación. Generalmente, los argumentos son específicos para cada módulo y deberían estar documentados. Si se pasara un argumento no válido, el módulo lo ignoraría, aunque debería usar `syslog` para informar del error.

En cuanto a la sintaxis, hay que señalar que si se quieren introducir espacios en blanco en un argumento, éste deberá ir encerrado entre corchetes. Si lo que se pretende hacer es pasar un corchete como parte del parámetro, habrá hacer uso del carácter de escape '`\`'.

```
squid auth required pam_mysql.so user=passwd_query passwd=mada \
db=eminence [query=select user name from internet service where \
username_='%u' and password=PASSWORD('%p') and service='web_proxy']
```

En el ejemplo anterior puede observarse como se combinan distintos elementos para formar una regla bastante compleja. Nótese que los valores %u y %p son sustituidos por el nombre de usuario y la contraseña respectivamente.

Algunos módulos disponibles

Hasta el momento se han estudiado los fundamentos, la arquitectura y la configuración de PAM. Sin embargo, poco se ha dicho acerca de los módulos que ofrece. Esta sección está dedicada a describir, sin entrar en demasiado detalle, algunos de los módulos que vienen con la distribución oficial de PAM en su versión 0.77.

Esto no pretende ser una guía de referencia exhaustiva, por lo que para detalles más concretos sobre estos módulos se remite al lector a la documentación del proyecto. Simplemente, se expondrá una breve descripción de cada módulo y se mostrarán cuáles son los principales argumentos que admiten.

pam console.so

Grupo `session y auth`.

Este módulo permite el cambio de los permisos y de los propietarios de los ficheros indicados en `/etc/security/console.perms` cuando un usuario accede al sistema mediante una consola física y no hay ningún otro usuario registrado en el sistema. Cuando el usuario abandona la sesión, los permisos y propietarios originales se restauran.

En el caso de que varios usuarios trabajasen al mismo tiempo en la consola física, los ficheros sólo se le asignarían al primero en registrarse en el sistema, aunque esto no suele ocurrir.

Los parámetros admitidos por este módulo son:

`allow nonroot tty` Bloquea la consola y cambia los permisos incluso si el propietario del terminal no es el superusuario.

`permsfile=fichero` Permite especificar un fichero alternativo al `/etc/security/console.perms` del cual leer la base de datos de permisos.
`fstab=fichero` Permite indicar un fichero alternativo al fstab. El fichero fstab almacena información relativa a los dispositivos que se pueden montar en el sistema, indicando donde deben de montarse y el sistema de ficheros en el que se encuentra.

pam cracklib.so

Grupo `password`

Se trata de un módulo preventivo que se encarga de notificar al usuario de la debilidad de su clave cuando ve que puede "romperse" usando un diccionario. Este módulo entra en funcionamiento cuando se establece o modifica la clave de un usuario.

Básicamente este módulo hace pasar la clave por la biblioteca `cracklib`. En caso de que pase como clave segura, intentará las siguientes comprobaciones, comparando la nueva clave con la antigua:
Palíndromo Comprueba que la nueva contraseña no sea la vieja al revés.
Cambio de mayúsculas Realiza varias combinaciones cambiando mayúsculas por minúsculas y viceversa.
Similar Comprueba que las claves se diferencian en más de `difok` caracteres.
Simple Verifica el tamaño de la clave.
Rotada Comprueba que la clave nueva no es una rotación de la antigua.
Ya usada Se asegura de que la clave no se ha usado con anterioridad. Las contraseñas ya usadas se almacenan en `/etc/security/opasswd`.

Así mismo este módulo permite un gran número de parámetros, ahora se mostraran los más importantes:
`debug` Muestra información mas detallada a través de `syslog`. Esta opción no imprimirá las claves en el fichero de log.
`retry=N` Permite especificar el número de veces que se volverá a pedir la clave en caso de que no sea segura. Por defecto es **1**.
`difok=N` Indica el número de caracteres que deben ser diferentes entre la clave anterior y la nueva.
`minlen=N` Indica el número mínimo de caracteres que debe tener una clave.
`use_authok` Evita que el usuario introduzca la nueva contraseña tomándola del módulo que se encuentra por delante en la pila.

pam deny.so

Grupo `account`,`authentication`, `password` y `session`.

El objetivo de este módulo es producir un fallo siempre. Si este módulo es el único que se encuentra en la pila, se considerará que ésta ha fallado.

pam env.so

Grupo `authentication`

Permite establecer las variables de entorno por defecto o sustituir los valores de las variables ya establecidas cuando un usuario se registra en el sistema. El fichero donde se definen dichas variables se encuentra en `/etc/security/pam_env.conf`.

Las cláusulas de este fichero son de la siguiente manera:

```
VARIABLE [DEFAULT=[valor]] [OVERRIDE=[valor]]
```

donde la cláusula `default` especificará el valor a tomar por defecto y `override` el valor por el que será sustituido el contenido de la variable.

Este módulo admite cuatro parámetros:
`debug` Muestra información mas detallada sobre el módulo en syslog.
`conffile=fichero` Especifica el fichero de configuración alternativo.
`envfile=fichero` Especifica el fichero alternativo a /etc/enviroment que contiene las variables de entorno establecidas para el sistema.
`readenv=0/1` Especifica si se lee o no el fichero con las variables de entorno. Por defecto sí se lee.

pam limits.so

Grupo `session`

Controla los límites impuestos en el fichero `/etc/security/limits.conf` a los recursos disponibles. Las limitaciones se pueden aplicar a un usuario, a un grupo o a todos los usuarios del sistema. Los recursos que se pueden limitar desde este módulo van desde el máximo tiempo de CPU asignable, hasta el número máximo de ficheros que puede tener bloqueados simultáneamente.

Este módulo permite los siguiente parámetros:
`debug` Muestra información añadida a través de syslog.
`conf=fichero` Permite especificar un fichero de configuración alternativo.
`change uid` Permite cambiar el `uid` real para los usuarios que se han establecido límites. Esta opción se usa principalmente en sistemas en los que al poner un límite de 0 procesos el usuario no puede ni abrir la sesión.
`utmp early` Corrige el problema generado por algunas aplicaciones que intentan crear `utmp` para el usuario antes de que este entre en el sistema.

La sintaxis del fichero de configuración es la siguiente:

```
<dominio> <tipo> <elemento> <valor>
```

donde `<dominio>` es el nombre de usuario, nombre de grupo, el comodín '`*`' o el comodín '`%`'. `<tipo>` indicara si se trata de un limite duro (`hard`) débil (`soft`) o los dos tipos simultáneamente (`-`). Finalmente, el parámetro `<elemento>` se puede sustituirse por varios valores que indican el recurso que se esta limitando.

pam nologin.so

Grupo `account` y `authentication`

Este módulo sólo deja entrar a los usuarios del sistema si el fichero `/etc/nologin` no existe. En caso contrario, sólo el superusuario puede ingresar en el sistema. Al resto de los usuarios se les mostrará el contenido de dicho archivo.

`Pam nologin` permite dos parámetros:
`successok` Devuelve PAM SUCCESS en vez de PAM IGNORE en caso de no encontrar el fichero `/etc/nologin`.
`file=fichero` Permite indicar un fichero alternativo a `/etc/nologin`.

pam permit.so

Grupo `account`, `authentication`, `password` y `session`.

`Pam permit` funciona justamente al contrario del módulo `pam deny.so`. Para cada llamada al módulo este devuelve un acierto , `PAM SUCCESS`. Por esta razón este módulo es muy inseguro y debe de ser usado con precaución extrema.

pam rootok.so

Grupo `authentication`

`Pam rootok.so` devuelve un acierto siempre que el identificador real del usuario sea 0 (*el usuario root*). Con ello se consigue que el usuario root no necesite introducir la clave para acceder a los servicios asociados a este módulo. Al usar `pam rootok` hay que tener mucho cuidado ya que plantea un grave problema de seguridad. Si asociaramos `pam rootok` a un proceso que se arranca con el sistema, como por ejemplo `login`, el `uid` real sera **0** y por lo tanto acertará siempre, independientemente de que intentemos registrarnos con un usuario con `uid` distinto de **0**.

pam securetty.so

Grupo `authentication`

Sirve para limitar las consolas en las que se puede autenticar el usuario root. El fichero `/etc/securetty` contiene una lista de consolas seguras.

pam stack.so

Grupo `account`, `authentication`, `password` y `session`.

Este módulo permite la asociación en pila de varios módulos PAM devolviendo un acierto, en caso de que toda la pila devuelva un acierto. En caso de que uno de los módulos de la pila devuelva un error, pam stack devolverá el código de error devuelto por el módulo que ha fallado.

pam wheel.so

Grupo `authentication` Limita la autenticación como `root` a los usuarios del grupo `wheel`.

Los parámetros admitidos por `pam wheel` son los siguientes:

`debug` Muestra información añadida a través del `syslog`.

`use uid` Modifica el comportamiento del módulo usando el el `uid` del proceso y no el nombre de usuario de `getlogin`.

`trust` Con esta opción se consigue que los usuarios que pertenecen al grupo `wheel` cambiarse por el usuario `root` sin usar la clave. Debe de usarse con mucho cuidado.

`deny` Este parámetro hace que el módulo se comporte al revés denegando a los usuarios del grupo `wheel` la posibilidad de convertirse en `root`.

`group=XXXX` Permite especificar los grupos a los que se les permite la autenticación.

pam xauth.so

Grupo `session`

Se encarga de redireccionar las “cookies” de autenticación de un usuario a otro en el sistema X-Window. Esto se usa para que, cuando un usuario se haga pasar por otro mediante el uso del comando `su` o alguno de sus derivados, pueda seguir lanzando aplicaciones como el nuevo usuario sin necesidad de salir y volver a entrar en el entorno X-Window.

Ejemplos de configuración

Ahora que ya se han estudiado tanto los fundamentos de configuración como el cometido de los módulos más destacados de Linux-PAM, parece buena idea fijar la atención en una serie de ejemplos reales, con el fin de ilustrar los conceptos vistos con anterioridad.

Para ello se comentará a continuación la configuración que da Red Hat Linux a algunos de sus servicios. Concretamente, `login`, `passwd`, `su` y `other`.
Sin embargo, en primer lugar se va a tener en cuenta un servicio "artificial", `system-auth`. Este engloba políticas comunes a muchos otros servicios que lo incluyen en su configuración haciendo uso del módulo `pam_stack`. La ventaja de este esquema es que, en caso de querer cambiar el comportamiento común de varios servicios, tan sólo es necesario modificar `system-auth`.

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
### auth ###
# Inicializa las variables de entorno definidas en /etc/security/pam_env.conf.
auth required /lib/security/$ISA/pam_env.so
# Autentica al usuario al estilo 'unix' tradicional (/etc/passwd). Se detiene
# aquí si tiene éxito.
# nullok: permite contraseñas en blanco.
# likeauth: pam_sm_setcred() devuelve lo mismo que pam_sm_authenticate()
auth sufficient /lib/security/$ISA/pam_unix.so likeauth nullok
# Si se llega hasta aquí sin éxito, se falla.
auth required /lib/security/$ISA/pam_deny.so
### account ###
# Basándose en /etc/shadow, determina el estado de la cuenta de usuario (si ha
# expirado, debe cambiar la contraseña, etc.).
account required /lib/security/$ISA/pam_unix.so
### password ###
# Comprueba que la clave no es vulnerable a un ataque simple basado en
# diccionario.
# retry: 3 intentos para cambiar.
# type: modifica el mensaje. Para type=XXX, sería 'New XXX password: '.
password required /lib/security/$ISA/pam_cracklib.so retry=3 type=
# Actualiza la contraseña.
# nullok: permite cambiar una contraseña en blanco. Sin este parámetro,
# una clave en blanco se interpreta como señal de que la cuenta se
# encuentra desactivada.
# use_authtok: toma como nueva contraseña la obtenida por el módulo anterior
# (cracklib en este caso).
# md5: usa la función MD5 para cifrar la contraseña.
password sufficient /lib/security/$ISA/pam_unix.so nullok use_authtok md5 shadow
# Si se llega hasta aquí sin éxito, se falla.
password required /lib/security/$ISA/pam_deny.so
### session ###
# Limita los recursos para la sesión del usuario en base al contenido del
# archivo /etc/security/limits.conf.
session required /lib/security/$ISA/pam_limits.so
# Registra el acceso al servicio en syslog, tanto al principio como al final
# de la sesión.
session required /lib/security/$ISA/pam_unix.so
```

Sólo un apunte más, los comentarios de los ficheros que se presentan en esta sección han sido añadido por los autores de este documento, por lo que es probable que introduzcan algunas imprecisiones.

login

El programa `login` ofrece a los usuarios la posibilidad de abrir una sesión en el sistema.
Este es uno de los puntos de acceso más sensibles y que, por lo general, ofrece un mayor riesgo para la integridad del sistema. Así pues, la configuración de Linux-PAM en este (*como en todos los servicios, a decir verdad*) debe estar especialmente cuidada.

Por lo general, login pide un nombre de usuario y una contraseña. Después de llevar a cabo una serie de comprobaciones e inicializaciones, al usuario se le ofrece un intérprete de comandos (`shell`).

La configuración de este servicio en Red Hat Linux hace uso intensivo del servicio "artificial" `system-auth`, siendo idénticas las pilas `account` y `password`. Las diferencias radican, principalmente, en que al superusuario sólo se le permite acceder desde terminales físicas (`tty`), se deshabilita la entrada si

existe el fichero `/etc/nologin` y se realizan cambios de permisos/propietario según lo indicado en `/etc/security/console.perms`.

```
#%PAM-1.0
### auth ###
# Deshabilita el 'login' para el root exceptuando los tty's.
auth required pam_securetty.so
# Procesa los módulos 'auth' del servicio 'system-auth'.
auth required pam stack.so service=system-auth
# Deshabilita la entrada de cualquier usuario que no sea el 'root' cuando el
# fichero /etc/nologin existe.
auth required pam_nologin.so
### account ###
# Procesa los módulos 'account' del servicio 'system-auth'.
account required pam_stack.so service=system-auth
### password ###
# Procesa los módulos 'password' del servicio 'system-auth'.
password required pam_stack.so service=system-auth
### session ###
# Procesa los módulos 'session' del servicio 'system-auth'.
session required pam_stack.so service=system-auth
# Cambia permisos y propietarios de ciertos ficheros según se indica en
# /etc/security/console.perms si el acceso se realiza por medio de una
# consola 'física'. Los permisos/propietarios originales se restauran al
# salir.
session optional pam_console.so
```

passwd

`passwd` es la utilidad que permite modificar la contraseña de las cuentas asociadas a los usuarios y a los grupos. Actuando como un usuario normal del sistema solamente podrá modificar la propia contraseña, mientras que el superusuario podrá modificar la contraseña de cualquier cuenta. Así mismo, solamente el administrador de un grupo podrá cambiar la contraseña del grupo.

Además de permitir el cambio de la contraseña, `passwd` permite modificar otros parámetros asociados a una cuenta de usuario como son el nombre, el shell de inicio o la fecha de caducidad de la clave.

```
#%PAM-1.0
### auth ###
# Procesa los módulos 'authentication' del servicio 'system-auth'
auth required pam_stack.so service=system-auth
### account ###
# Procesa los módulos 'account' del servicio 'system-auth'
account required pam_stack.so service=system-auth
### password ###
# Procesa los módulos 'password' del servicio 'system-auth'
password required pam_stack.so service=system-auth
```

su

`su` es una aplicación que permite convertirse en otro usuario una vez que tenemos una sesión ya abierta. Cuando se invoca sin parámetros el `su` actúa como si se quisiera convertir al superusuario, `root`. En caso contrario su intentara convertirse en el usuario pasado por parámetro.

A continuación se mostrara el fichero de configuración del pam para la utilidad `su` suministrado en RedHat.

```
#%PAM-1.0
### auth ###
# Permite que el usuario root use la utilidad su sin suministrar
# ninguna contraseña.
auth sufficient /lib/security/$ISA/pam rootok.so
# Descomentar la siguiente linea para indicar que los usuarios
# pertenecientes al grupo wheel sean usuarios en los que se confía y
# puedan realizar un 'su' sin suministrar contraseña.
#auth sufficient /lib/security/$ISA/pam_wheel.so trust use_uid
# Descomentar la siguiente linea para indicar que los usuarios
# pertenecientes al grupo wheel son los únicos usuarios a los que se
```

```
# les permite usar la utilidad 'su'
#auth required /lib/security/$ISA/pam_wheel.so use_uid
# Procesa los módulos 'authentication' del servicio 'system-auth'.
auth required /lib/security/$ISA/pam stack.so service=system-auth
### account ###
# Procesa los módulos 'account' del servicio 'system-auth'.
account required /lib/security/$ISA/pam_stack.so service=system-auth
### password ###
# Procesa los módulos 'password' del servicio 'system-auth'.
password required /lib/security/$ISA/pam stack.so service=system-auth
### session ###
# Procesa los módulos 'session' del servicio 'system-auth'.
session required /lib/security/$ISA/pam_stack.so service=system-auth
# Carga el modulo xauth como opcional, permitiendo que el usuario al
# que se ha cambiado mediante el su pueda seguir lanzando aplicaciones X-Window
# en la sesión abierta por el usuario original.
session optional /lib/security/$ISA/pam_xauth.so
```

other

Tal y como se explicó con anterioridad, cuando PAM no encuentra un fichero de configuración para un servicio en particular, aplica las reglas del servicio especial `other`. Por tanto, puede decirse que se trata de la política por defecto.

Por motivos de seguridad, es aconsejable que este servicio deniegue cualquier acceso. Así, si Linux-PAM no encuentra configuración para un servicio, simplemente denegará el acceso, minimizando posibles amenazas.

Con esta premisa, la configuración resulta tremendamente simple, tal y como puede apreciarse en las siguientes líneas.

```
#%PAM-1.0
# Por defecto, se deniega el acceso a cualquiera
auth required /lib/security/$ISA/pam_deny.so
account required /lib/security/$ISA/pam_deny.so
password required /lib/security/$ISA/pam deny.so
session required /lib/security/$ISA/pam_deny.so
```

Valores devueltos por los módulos de PAM

Cada módulo de PAM devuelve un valor al ser invocado. Este indica si se ha producido algún fallo o si, por el contrario, todo ha ido bien.

PAM permite especificar comportamientos para cada uno de estos posibles valores en el fichero de configuración . Para ello, se utiliza la sintaxis de pares `valor=acción`, colocando en la parte izquierda de la asignación la la etiqueta del valor (*cualquiera de los valores descritos a continuación sin el prefijo PAM y escrito en minúscula; por ejemplo,* `auth err`).

authentication

`PAM AUTH ERR` El usuario no se autenticó.
`PAM AUTHINFO UNAVAIL` El módulo no fue capaz de acceder a la información de autenticación. Esto se debe generalmente a un fallo hardware o de acceso a la red.
`PAM CRED ERR` Este valor se devuelve cuando el módulo no pudo establecer las credenciales del usuario.
`PAM CRED EXPIRED` Indica que las credenciales del usuario han caducado.
`PAM CRED INSUFFICIENT` Por alguna razón, la aplicación no tiene suficientes credenciales para autenticar al usuario.
`PAM CRED UNAVAIL` El módulo no pudo obtener las credenciales de los usuarios.
`PAM MAXTRIES` Los módulos devuelven este valor cuando han alcanzado el número máximo de reintentos de autenticar al usuario.

`PAM USER UNKNOWN` El nombre de usuario es desconocido para el sistema de 'authentication'.
`PAM SUCCESS` Este valor se devuelve cuando el módulo ha tenido éxito.

account

`PAM ACCT EXPIRED` La cuenta del usuario ha caducado y ya no puede volver a acceder con dicha cuenta.
`PAM AUTH ERR` El usuario no se autenticó.
`PAM AUTHTOKEN REQD` El token de autenticación ha caducado. Antes de volver a llamar a esta función se debería pedir un nuevo token.
`PAM SUCCESS` Este valor se devuelve cuando el módulo ha tenido éxito.
`PAM USER UNKNOWN` El nombre de usuario es desconocido para el sistema de '`account`'.

password

`PAM AUTHOK DISABLE AGING` El token de autenticación ha sido desactivado.
`PAM AUTHTOK ERR` El módulo fue incapaz de obtener un nuevo token de autenticación.
`PAM AUTHTOK RECOVERY ERR` No se consiguió leer el anterior token de autenticación.
`PAM AUTHTOK LOCK BUSY` El token de autenticación estaba bloqueado cuando se intentó cambiar.
`PAM PERM DENIED` Permiso denegado.
`PAM SUCCESS` Este valor se devuelve cuando el módulo ha tenido éxito.
`PAM TRY AGAIN` Las comprobaciones anteriores fallaron. Se pide que se vuelva a hacer la misma llamada.
`PAM USER UNKNOWN` El nombre de usuario es desconocido para el sistema de 'password'.

session

`PAM SESSION ERR` Se ha producido un fallo al intentar abrir o cerrar la sesión.
`PAM SUCCESS` Este valor se devuelve cuando el módulo ha tenido éxito.

Anexo II - proftpd.conf

```
#
# /etc/proftpd/proftpd.conf -- This is a basic ProFTPD configuration file.
# To really apply changes reload proftpd after modifications.
#

# Includes DSO modules
Include /etc/proftpd/modules.conf

# Set off to disable IPv6 support which is annoying on IPv4 only boxes.
UseIPv6           on
# If set on you can experience a longer connection delay in many cases.
IdentLookups          off

ServerName        "Debian"
ServerType        standalone
DeferWelcome          off

MultilineRFC2228      on
DefaultServer         on
ShowSymlinks          on

TimeoutNoTransfer     600
TimeoutStalled        600
TimeoutIdle       1200

DisplayLogin                    welcome.msg
DisplayChdir                .message true
ListOptions                 "-l"

DenyFilter        \*.*/

# Use this to jail all users in their homes
# DefaultRoot         ~

# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
# RequireValidShell           off

# Port 21 is the standard FTP port.
Port           21

# In some cases you have to specify passive ports range to by-pass
# firewall limitations. Ephemeral ports can be used for that, but
# feel free to use a more narrow range.
# PassivePorts                  49152 65534

# If your host was NATted, this option is useful in order to
# allow passive tranfers to work. You have to use your public
# address and opening the passive ports used on your firewall as well.
# MasqueradeAddress           1.2.3.4

# This is useful for masquerading address with dynamic IPs:
# refresh any configured MasqueradeAddress directives every 8 hours
<IfModule mod_dynmasq.c>
# DynMasqRefresh 28800
</IfModule>

# To prevent DoS attacks, set the maximum number of child processes
# to 30.  If you need to allow more than 30 concurrent connections
# at once, simply increase this value.  Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd)
MaxInstances          30

# Set the user and group that the server normally runs at.
User           proftpd
Group          nogroup

# Umask 022 is a good standard umask to prevent new files and dirs
# (second parm) from being group and world writable.
Umask          022  022
# Normally, we want files to be overwriteable.
AllowOverwrite        on
```

```
# Uncomment this if you are using NIS or LDAP via NSS to retrieve passwords:
# PersistentPasswd           off

# This is required to use both PAM-based authentication and local passwords
# AuthOrder       mod_auth_pam.c* mod_auth_unix.c

# Be warned: use of this directive impacts CPU average load!
# Uncomment this if you like to see progress and transfer rate with ftpwho
# in downloads. That is not needed for uploads rates.
#
# UseSendFile         off

TransferLog /var/log/proftpd/xferlog
SystemLog   /var/log/proftpd/proftpd.log

<IfModule mod quotatab.c>
QuotaEngine off
</IfModule>

<IfModule mod_ratio.c>
Ratios off
</IfModule>


# Delay engine reduces impact of the so-called Timing Attack described in
# http://security.lss.hr/index.php?page=details&ID=LSS-2004-10-02
# It is on by default.
<IfModule mod_delay.c>
DelayEngine on
</IfModule>

<IfModule mod ctrls.c>
ControlsEngine        off
ControlsMaxClients    2
ControlsLog           /var/log/proftpd/controls.log
ControlsInterval      5
ControlsSocket        /var/run/proftpd/proftpd.sock
</IfModule>

<IfModule mod_ctrls_admin.c>
AdminControlsEngine off
</IfModule>

#
# Alternative authentication frameworks
#
#Include /etc/proftpd/ldap.conf
#Include /etc/proftpd/sql.conf

#
# This is used for FTPS connections
#
#Include /etc/proftpd/tls.conf

#
# Useful to keep VirtualHost/VirtualRoot directives separated
#
#Include /etc/proftpd/virtuals.conf

# A basic anonymous configuration, no upload directories.

# <Anonymous ~ftp>
#   User          ftp
#   Group             nogroup
#   # We want clients to be able to login with "anonymous" as well as "ftp"
#   UserAlias         anonymous ftp
#   # Cosmetic changes, all files belongs to ftp user
#   DirFakeUser   on ftp
#   DirFakeGroup on ftp
#
#   RequireValidShell         off
#
#   # Limit the maximum number of anonymous logins
#   MaxClients        10
#
#   # We want 'welcome.msg' displayed at login, and '.message' displayed
```

```
#   # in each newly chdired directory.
#   DisplayLogin              welcome.msg
#   DisplayChdir      .message
#
#   # Limit WRITE everywhere in the anonymous chroot
#   <Directory *>
#     <Limit WRITE>
#       DenyAll
#     </Limit>
#   </Directory>
#
#   # Uncomment this if you're brave.
#   # <Directory incoming>
#   #   # Umask 022 is a good standard umask to prevent new files and dirs
#   #   # (second parm) from being group and world writable.
#   #   Umask                 022  022
#   #             <Limit READ WRITE>
#   #             DenyAll
#   #             </Limit>
#   #             <Limit STOR>
#   #             AllowAll
#   #             </Limit>
#   # </Directory>
#
# </Anonymous>
```

Anexo III - tls.conf

```
##########################################Fichero
/etc/proftpd/tls.conf###################################################
#
# Proftpd sample configuration for FTPS connections.
#
# Note that FTPS impose some limitations in NAT traversing.
# See http://www.castaglia.org/proftpd/doc/contrib/ProFTPD-mini-HOWTO-TLS.html
# for more information.
#

<IfModule mod_tls.c>
TLSEngine                               on
TLSLog                                  /var/log/proftpd/tls.log
TLSProtocol                             SSLv23
#
# Server SSL certificate. You can generate a self-signed certificate using
# a command like:
#
# openssl req -x509 -newkey rsa:1024 \
#          -keyout /etc/ssl/private/proftpd.key -out /etc/ssl/certs/proftpd.crt \
#          -nodes -days 365
#
# The proftpd.key file must be readable by root only. The other file can be
# readable by anyone.
#
# chmod 0600 /etc/ssl/private/proftpd.key
# chmod 0644 /etc/ssl/certs/proftpd.crt
#
TLSRSACertificateFile                   /etc/ssl/certs/proftpd.crt
TLSRSACertificateKeyFile                /etc/ssl/proftpd.key
#
# CA the server trusts
#TLSCACertificateFile                /etc/ssl/certs/CA.pem
# or avoid CA cert
TLSOptions                              NoCertRequest
#
# Authenticate clients that want to use FTP over TLS?
#
TLSVerifyClient                         off
#
# Are clients required to use FTP over TLS when talking to this server?
#
TLSRequired                             on
#
# Allow SSL/TLS renegotiations when the client requests them, but
# do not force the renegotations.  Some clients do not support
# SSL/TLS renegotiations; when mod_tls forces a renegotiation, these
# clients will close the data connection, or there will be a timeout
# on an idle data connection.
#
TLSRenegotiate                          required off
</IfModule>

##########################################Fin
/etc/proftpd/tls.conf###################################################
```

Anexo IV - tls2.conf

```
#########################################Fichero
/etc/proftpd/tls.conf#################################################
#
# Proftpd sample configuration for FTPS connections.
#
# Note that FTPS impose some limitations in NAT traversing.
# See http://www.castaglia.org/proftpd/doc/contrib/ProFTPD-mini-HOWTO-TLS.html
# for more information.
#

<IfModule mod_tls.c>
<global>
TLSEngine                               on
TLSLog                                  /var/log/proftpd/tls.log
</global>


TLSProtocol                             SSLv23

<global>
#
# Server SSL certificate. You can generate a self-signed certificate using
# a command like:
#
# openssl req -x509 -newkey rsa:1024 \
#          -keyout /etc/ssl/private/proftpd.key -out /etc/ssl/certs/proftpd.crt \
#          -nodes -days 365
#
# The proftpd.key file must be readable by root only. The other file can be
# readable by anyone.
#
# chmod 0600 /etc/ssl/private/proftpd.key
# chmod 0644 /etc/ssl/certs/proftpd.crt
#
TLSRSACertificateFile                   /etc/ssl/certs/proftpd.crt
TLSRSACertificateKeyFile                /etc/ssl/proftpd.key
#
# CA the server trusts
#TLSCACertificateFile                    /etc/ssl/certs/CA.pem
# or avoid CA cert
TLSOptions                              NoCertRequest
#
# Authenticate clients that want to use FTP over TLS?
#
TLSVerifyClient                         off
#
# Are clients required to use FTP over TLS when talking to this server?
#
TLSRequired                             on
#
# Allow SSL/TLS renegotiations when the client requests them, but
# do not force the renegotations.  Some clients do not support
# SSL/TLS renegotiations; when mod_tls forces a renegotiation, these
# clients will close the data connection, or there will be a timeout
# on an idle data connection.
#
TLSRenegotiate                          required off
</global>
</IfModule>

#########################################Fin
/etc/proftpd/tls.conf#################################################
```