

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, 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.



[Ministerio de Educación y Formación Profesional](#) (Dominio público)

Materiales formativos de FP Online propiedad del Ministerio de Educación y Formación Profesional.

[Aviso Legal](#)

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: Linux Ubuntu 20.04 LTS
- ✓ 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

Reflexiona

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 en Internet, 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.

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.

Para saber más

En el siguiente enlace, página web del RFC 959 sobre FTP, puedes encontrar traducido el estándar RFC sobre FTP.

[Página web del RFC 959 sobre FTP.](#)

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

[Página web del RFC 4251 sobre SSH.](#)

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. Perteneciente a la familia de protocolos de red TCP 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.



[rftc](#) (CC BY-NC-SA)

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 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.

Autoevaluación

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?

- ☐ 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.

Mostrar retroalimentación

Solución

1. Correcto
2. Incorrecto
3. Correcto
4. Incorrecto

1.2.- Cliente FTP.

Citas para pensar

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://



[rfc](#) (CC BY-NC-SA)

¿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 "tabulado" 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)? 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/1.1, SOCKS5 y soporte de FTP-Proxy.
- ✓ Fichero de registro.
- ✓ Sincronización de directorios de navegación.
- ✓ Búsqueda de archivos remoto.

Para saber más

En el siguiente enlace puedes acceder a la página web oficial de FileZilla donde puedes descargarlo y encontrar documentación sobre el mismo.

[Página web oficial de FileZilla.](#)

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.



[rfc](#) (CC BY-NC-SA)

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, bases de datos _____ SQL y servidores _____ RADIUS.

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**.

[Howto AuthFiles.](#)

- ✓ 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: Es necesario establecer la directiva **AuthPAMAuthorative** a 'on'.

[Directiva AuthPAMAuthorative.](#)

- ✓ 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](#)

[Howto SQL.](#)

- ✓ LDAP: Para ello emplea el módulo **mod_ldap**.
- ✓ RADIUS: Para ello emplea el módulo **mod_radius**.

Mediante la [directiva UserPassword](#) se puede crear una contraseña para un usuario particular que sobrescribe 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.

Para saber más

En el siguiente archivo encontrarás más información sobre PAM.

[Información sobre PAM](#) (0.17 MB)

1.4.- Modos de conexión del cliente.

Reflexiona

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?

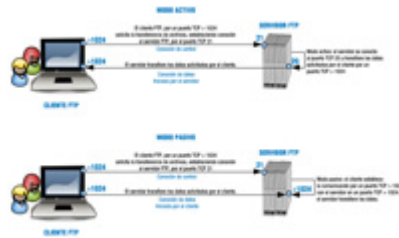
Ya se ha comentado que el servidor FTP a diferencia de otros servidores necesita dos puertos TCP para hacer posible la transferencia de archivos. Ahora bien, ¿son estos puertos siempre los mismos o no? ¿son independientes del tipo de cliente y servidor o no? Pues, básicamente depende de dos factores: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

A priori, si no modificamos la configuración del servidor ftp éste 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.



[rfc](#) (CC BY-NC-SA)

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**.

Reflexiona

Autoevaluación

Ejercicio de relacionar

Extensión de archivo	Relación	Tipo de Transferencia
txt (texto).	<input type="checkbox"/>	1. ascii.
html (página web).	<input type="checkbox"/>	2. binario.
doc (documento).	<input type="checkbox"/>	

Extensión de archivo	Relación	Tipo de Transferencia
zip (comprimido).	<input type="checkbox"/>	
ps (postscript).	<input type="checkbox"/>	
mp3 (audio).	<input type="checkbox"/>	
tar (comprimido).	<input type="checkbox"/>	
tgz (comprimido).	<input type="checkbox"/>	
bz2 (comprimido).	<input type="checkbox"/>	

Todos los archivos que no son texto plano son tipo binario.

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.



[_rfc_](#) (CC BY-NC-SA)

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 un '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**.

Para saber más

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.

Autoevaluación

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, mientras 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, mientras que el grupo clase y los demás no poseen ese permiso.

Mostrar retroalimentación

Solución

1. Correcto
2. Incorrecto
3. Correcto
4. Incorrecto

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 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.rediris.es:

1. Básicamente la sintaxis es la siguiente:

```
ftp [-pínegvd] [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:

```
# 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:

```
# 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

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**.

[illegible]

Para saber más

[Cliente en modo texto lftp](#) (0.01 MB)

1.7.1.- Comandos ftp.



[rfc](#) (CC BY-NC-SA)

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

Comandos FTP

ABRIR/CERRAR CONEXIÓN	
COMANDO/S Y ARGUMENTOS	EXPLICACIÓN
<code>open servidor</code>	Inicia conexión remota con un servidor ftp.
<code>close / disconnect</code>	Finalizan la sesión ftp sin cerrar la consola ftp.
<code>bye / quit / exit</code>	Terminan la sesión ftp y salen de la consola ftp.
<code>!</code>	Sale a línea de comandos del sistema operativo temporalmente sin cortar la conexión. Para volver, teclea <code>exit</code> en la línea de comandos.
AYUDA	
COMANDO/S Y ARGUMENTOS	EXPLICACIÓN
<code>? / help</code>	Muestra una lista de los comandos disponibles.
<code>? comando / help comando</code>	Muestra la información relativa al comando.
TRABAJAR CON DIRECTORIOS	
COMANDO/S Y ARGUMENTOS	EXPLICACIÓN
<code>cd directorio</code>	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.

Para saber más

Puedes descargar en el siguiente documento una tabla con más comandos ftp.

[Tabla con más comandos ftp.](#) (0.05 MB)

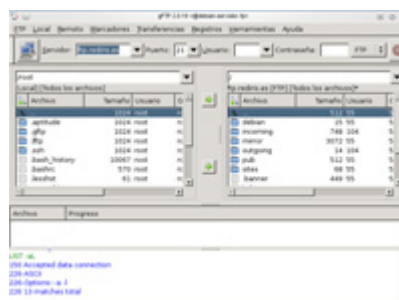
El servicio de transferencia de ficheros está bien, pero obliga a entender el funcionamiento de un servidor ftp mediante el uso de sus comandos. La verdad, es que no es muy interactivo, ¿entonces..., no existe la posibilidad de trabajar de otro modo más interactivo? Pues si, mediante clientes ftp de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

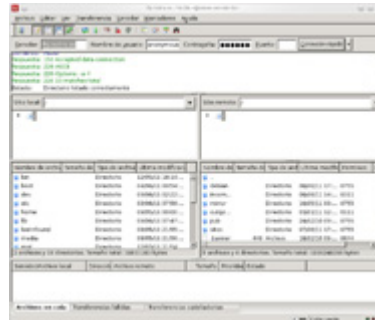


rfc (CC BY-NC-SA)

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.



[rftc](#) (CC BY-NC-SA)

Para saber más

Para instalar el cliente gráfico gFTP, seguiremos los siguientes pasos:

1. Actualizar sistema operativo:

```
# apt-get update
# apt-get upgrade
```

2. Instalar cliente FTP en modo gráfico:

```
# 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.

<https://www.youtube.com/embed/nBm-2rgkf5Y>

[Resumen textual alternativo](#)

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.

ftp://nombre_servidor_ftp:puerto

- ✔ **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.

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



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-.

https://sarreplec.caib.es/pluginfile.php/10978/mod_resource/content/1/DAW04_v1/ArchivosUnidad/Moodle/DAW04_completa_offline/D... 24/53

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:

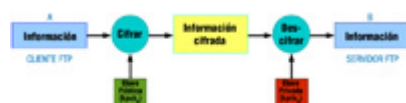
- ✓ SSL Implícito:
 - ➔ Como conexiones HTTPS.
 - ➔ Usa los puertos 990 y 989.
- ✓ SSL Explícito
 - ➔ El cliente usa los mismos puertos estándar FTP: 20 y 21 pero se efectúa el cifrado en ellos.
 - ➔ Usa AUTH SSL.
- ✓ TLS Explícito:
 - ➔ Similar a SSL Explícito pero usa AUTH TLS.

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.

Funcionamiento del cifrado asimétrico



[rfc](#) (CC BY-NC-SA)



[rfc](#) (CC BY-NC-SA)

$$\begin{aligned}
 &A(\text{inf}) \rightarrow \text{inf cifrada} \rightarrow B [\text{descifrar inf}] \rightarrow B(\text{inf}) = A(\text{inf}) \\
 &A(\text{inf}) \rightarrow \text{inf cifrada} = [(\text{inf})]_{k_{\text{pub}}B} \rightarrow B [\text{inf. cifrada}]_{k_{\text{priv}}B} \rightarrow B(\text{inf}) = A(\text{inf})
 \end{aligned}$$

Funcionamiento del cifrado asimétrico	
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.

Para saber más

En el siguiente enlace encontrarás más información sobre asegurar FTP con TLS.

[Asegurar FTP con TLS.](#)

En este otro enlace encontrarás más información sobre el protocolo TLS.

[Protocolo TLS.](#)

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

Reflexiona

¿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.



[rftc](#) (CC BY-NC-SA)

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?:



[The ProFTPD Project \(GNU/GPL\)](#)

- 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 proftpd en un sistema Operativo Debian/Ubuntu 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, ProFTPD debería 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:124:65534::/run/proftpd:/usr/sbin/nologin
ftp:x:125:65534::/srv/ftp:/usr/sbin/nologin
```

Para saber más

Te proponemos el siguiente enlace a un vídeo práctico sobre la instalación y uso de proftpd en una distribución GNU/Linux basada en Debian. Aunque es una versión antigua, el procedimiento es prácticamente igual al de Ubuntu.

<https://www.youtube.com/embed/ijol1TDpci>

[Resumen textual alternativo](#)

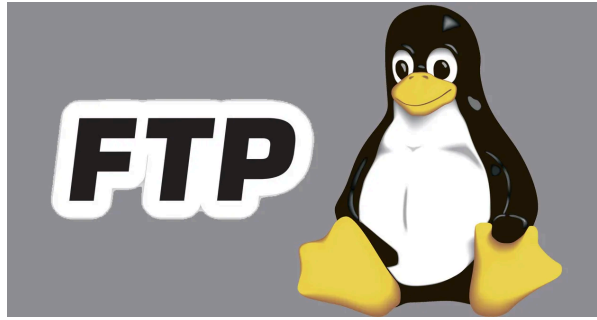
En este otro vídeo podemos ver la instalación del servidor ProFTP en Ubuntu<.

<https://www.youtube.com/embed/FNf-E3gcvWg>

[Resumen textual alternativo](#)

Debes conocer

En el siguiente enlace podemos ver una descripción detallada de cómo instalar ProFTPd en Ubuntu:



RedesZone.net (Todos los derechos reservados)

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/proftpd/proftpd.conf>
- ✓ La posibilidad de configurar hosts virtuales o virtual hosts, 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.



[rfc](#) (CC BY-NC-SA)

Debes conocer

En el siguiente enlace encontrarás la página web oficial de ProFTPD.

[Página web oficial de ProFTPD.](#)

En el siguiente enlace encontrarás información sobre las directivas de ProFTPD.

[Directivas de ProFTPD.](#)

Una vez instalado ProFTPD existirán dos ficheros de especial interés:

- ✓ El fichero [/etc/ftpusers](#) (0.01 MB), ya comentado, 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.
- ✓ El fichero de configuración principal ([/etc/proftpd/proftpd.conf](#)) similar al del siguiente enlace: [proftpd.conf](#) (0.01 MB)

En el fichero proftpd.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 Ubuntu, disponemos de un archivo de configuración [/etc/proftpd/proftpd.conf](#) (0.01 MB)

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.



[_ftc_](#) (CC BY-NC-SA)

Para saber más

En el siguiente enlace encontrarás ejemplos de configuraciones del servidor ProFTPD.

[Ejemplos de configuraciones del servidor ProFTPD](#)

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.



[rftc](#) (CC BY-NC-SA)

Para hacer que proftpd permita conexiones mediante usuarios del sistema y usuarios anónimos debes modificar el fichero [/etc/proftpd/proftpd.conf](#) (0.01 MB)

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 $H>
# En Debian/Ubuntu ~ftp=/home/ftp . Este directorio será la raíz de los direct>
# ftp, esto es, /home/ftp estará enjaulado (chroot) de tal forma que aunque el >
# quisiera acceder a otros directorios situados fuera de /home/ftp no podrá acc>
<Anonymous ~ftp>
# Después de hacer login anónimo mediante ftp el servidor se ejecuta con el u>
# nogroup
User ftp
Group nogroup
# La siguiente línea permite hacer login con el usuario "anonymous" igual que>
UserAlias anonymous ftp
# Cambios de apariencia, todos los ficheros parecerán pertenecer al usuario y>
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 d>
<Directory *>
  <Limit WRITE>
    DenyAll
  </Limit>
</Directory>
</Anonymous>
# Fin de la configuración 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](#) (0.01 MB)

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
    # 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.

Debes conocer

En el siguiente enlace encontrarás información sobre la directiva Order de ProFTPD.

[Directiva Order de ProFTPD.](#)

En el siguiente enlace encontrarás información sobre la directiva Allow de ProFTPD.

[Directiva Allow de ProFTPD.](#)

En el siguiente enlace encontrarás información sobre la directiva Deny de ProFTPD.

[Directiva Deny de ProFTPD.](#)

Ejercicio resuelto

Según lo visto anteriormente, ¿cómo permitirías al usuario invitado establecer conexión desde las redes: 192.168.200 y 10.0.200?

Mostrar retroalimentación

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.**

Ejercicio resuelto

Y si además, ¿quisieras permitir el acceso desde los dominios **tuhostA.tudominio.edu,** **tuhostB.tudominio.edu** y

tuhostC.tudominio.edu?**Mostrar retroalimentación**

Pues, de la misma forma aque 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.



[Jason Whittaker](#) (CC BY-NC-SA)

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,

```
ftppasswd --passwd --name user-empresa1 --file /etc/passwd.usuarios.virtuales --i
```

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:

```
ftppasswd --passwd --name user-empresa2 --file /etc/passwd.usuarios.virtuales --i
```



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 sí, 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/
```



[guerkmachine](#) (CC BY-NC-SA)

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 ProFTPD, 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.

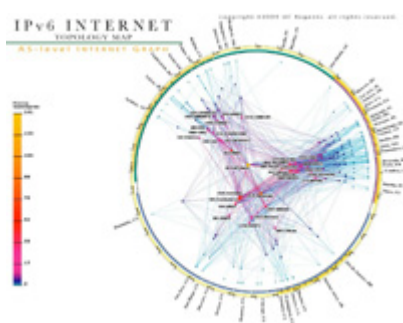
RequireValidShell off → No es necesario tener una shell declarada en el fichero /etc/shells

<<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 puerto TCP 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



[Brajeshwar](#) (CC BY-NC-SA)

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 los virtualhost, 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 ProFTPD, 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.

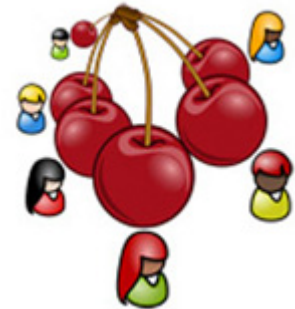
RequireValidShell off → No es necesario tener una shell declarada en el fichero `/etc/shells`.

</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:



[rftc](#) (CC BY-NC-SA)

```
<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 saber más

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

[Documentación oficial de ProFTPD sobre mod_quotatab](#)

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.li
```

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
-----
Name: user-empresal
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
```



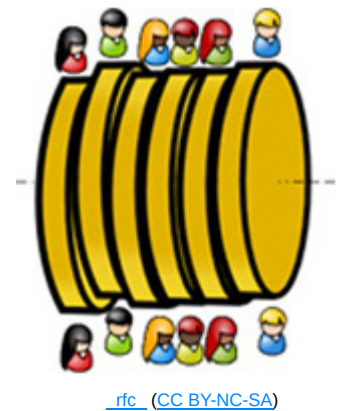
- ✓ Para actualizar la cuota de un usuario, por ejemplo, user-empresa1:

```
# ftpquota --update-record --type=limit --name=user-empresal --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.limit
-----
Name: user-empresal
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:

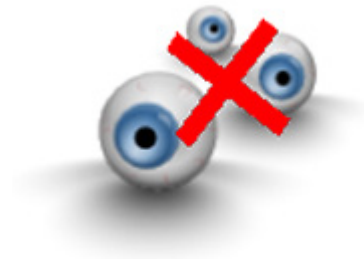
```
# ftpquota --show-records --type=limit --table-path=/etc/proftpd/ftpquota.limit
ftpquota: (empty table)
```

Para saber más

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.



[_rfc](#) (CC BY-NC-SA)

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` y ejecución del comando `openssl`.

```
# apt-get install openssl
# openssl req -x509 -newkey rsa:1024 -keyout /etc/ssl/private/proftpd.key
  /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](#)

(0.01 MB)

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](#) (0.01 MB) .

Debes conocer

Te proponemos el siguiente enlace de un vídeo práctico sobre cómo configurar TLS en el servidor ProFTPD y cómo configurar una plantilla de

FileZilla que soporta conexión TLS. La configuración se realiza sobre una distribución GNU/Linux basada en Debian.

<https://www.youtube.com/embed/jvdR5nZ30gE>

[Resumen textual alternativo](#)

