Servidores de ficheros mediante FTP y TFTP.

Autor: Enrique V. Bonet Esteban

Introducción.

Los usuarios de una red necesitan, en múltiples ocasiones, hacer públicos archivos para que el resto de usuarios de la red puedan acceder a los mismos. Esto puede hacerse con el servicio de FTP, el cual permite que dos ordenadores intercambien archivos entre sí de forma transparente para el usuario.

El intercambio de archivos entre dos ordenadores es una operación teóricamente sencilla, pero que en la práctica presenta innumerables problemas, debidos, principalmente, a la existencia de multitud de sistemas de archivos distintos, los cuales difieren en un gran número de aspectos. Entre estos aspectos se encuentra:

- Convenciones diferentes para nombrar los archivos.
- Reglas diferentes para recorrer los sistemas de directorios.
- Restricciones de acceso a archivos.
- Formas diferentes de representar texto y datos dentro de los archivos.

Para resolver estos problemas, se desarrolló el protocolo de transferencia de archivos (File Transfer Protocol). FTP es un programa muy básico para la transferencia de archivos entre ordenadores, pero a la vez es muy sencillo y fácil de usar, permitiendo ser utilizado de forma interactiva por un usuario, o bien, desde cualquier otra aplicación.

En FTP, un cliente de FTP establece una conexión con un servidor de FTP a través de lo que se conoce como conexión de control, siendo esta conexión de control una sencilla sesión de NVT[1]. El cliente envía los comandos al servidor a través de la conexión de control y el servidor envía respuestas al cliente a través de la misma.

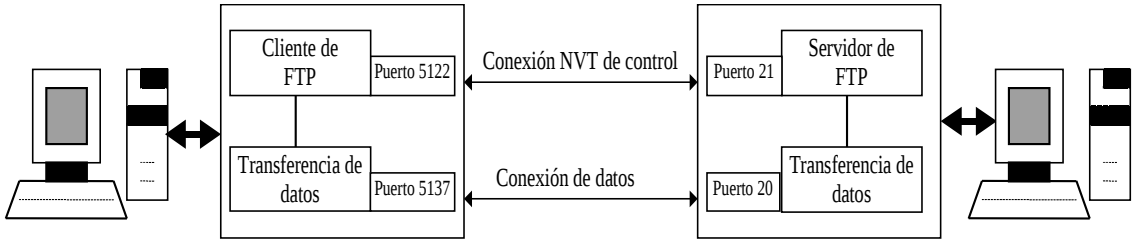
El conjunto de comandos ha ido creciendo con el tiempo y es bastante amplio, aunque no todos los clientes y/o servidores de FTP tienen porque implementar todos los comandos[2]. Si un cliente no contiene un comando que el usuario quiere enviar al servidor, las implementaciones de FTP suelen disponer del comando QUOTE (que puede entenderse como citar literalmente), que permite escribir el comando formal tal como se quiere enviar. La expresión escrita se transmitirá a través de la conexión de control exactamente como fue introducida.

La respuesta general a un comando de control enviado por un cliente es un código de respuesta numérico formado por tres dígitos. Los códigos 1xx indican, de forma general, que se ha comenzado a realizar una acción, los códigos 2xx que el

[1] Network Virtual Terminal es un modelo de terminal de red (virtual) que se verá con más detalle en temas posteriores.
[2] Un listado completo de los comandos existentes puede encontrarse en el apéndice A del tema.

comando se realizó con éxito, los códigos 3xx que se ha alcanzado con éxito un punto intermedio, los códigos 4xx indican un error temporal que puede recuperarse con posterioridad, y los códigos 5xx indican un error permanente y no recuperable.

Sin embargo, si el cliente de FTP ejecuta un comando que solicita la transferencia de archivos, además de la respuesta por la conexión de control, se establece una nueva conexión, conocida como conexión de datos, que es independiente de la conexión de control y por la que se transfiere el archivo solicitado[3]. En la figura siguiente se muestra este modelo, donde suponemos que el servidor está en modo activo y utiliza, por tanto, el puerto 20 para su extremo de la conexión de datos[4].



Transferencia de datos en FTP.

Además de los problemas del nombre de los archivos, etc., el principal problema existente en la transferencia de archivos entre ordenadores es el diferente formato que tienen los mismos datos en función del hardware y sistema operativo que se ejecute. Por tanto, en una transferencia de archivos entre dos ordenadores, ambos necesitan conocer el formato de los datos que van a transmitir y recibir. Para definir el formato de transferencia, FTP utiliza tres atributos: Tipo de los datos, estructura de los datos y modo de transmisión.

El tipo de los datos que se van a transmitir se indica mediante el comando de control TYPE. Los tres tipos de datos más utilizados son texto ASCII, texto EBCDIC y datos binarios, siendo el tipo ASCII el establecido generalmente por defecto.

El envío de datos de tipo ASCII entre computadoras con distinto sistema operativo Linux/UNIX, Windows y MacOS, principalmente, posee el problema de la distinta codificación del final de línea, etc. Para evitar este problema, los archivos de texto son convertidos desde ASCII al formato de NVT en el emisor y de forma inversa en el receptor, con lo que el problema es subsanado.

La transmisión de datos de texto EBCDIC se suele producir solo entre ordenadores IBM, de forma que en ellos no suceden los problemas que ocurren en los datos de tipo ASCII, por lo que los datos son enviados sin ser convertidos a NVT.

Por último, los datos binarios son enviados sin ningún tipo de conversión y sin tener en cuenta si los ordenadores son del mismo tipo (little-endian o big-endian), con lo que la transferencia, aún siendo correcta, puede ser ilegible en el ordenador destino.

[3] La conexión de datos también se utiliza para transferir los listados de los directorios y en general, cualquier respuesta que exceda de una sola línea.
[4] El modo activo y el modo pasivo de funcionamiento del servidor se explican con posterioridad.

La estructura de los datos a enviar, se indica mediante el comando de control STRU. Existen dos tipos de estructuras, estructura de archivo y estructura de registro. La estructura de archivo es la predeterminada y corresponde a que los datos a enviar son una secuencia de bytes. Por el contra, la estructura de registro indica que el archivo esta compuesto por una secuencia de registros de datos.

Por último, el modo de transmisión, indicado por el comando de control MODE, junto con la estructura de los datos vista con anterioridad, determina cuál será el formato de los datos durante la transferencia. Los tres modos de transmisión existentes son flujo, bloque y comprimido, siendo el modo de flujo el predeterminado.

En modo flujo con estructura de datos de archivo, el archivo se transmite como un flujo de bytes. FTP confía que TCP garantizará la integridad de los datos y no inserta ni cabeceras ni delimitadores entre los datos. La única forma de señalar que se ha llegado al final del archivo es terminando la conexión de datos. Sin embargo, si la estructura de datos es de registro, cada registro se delimita con un código de control de fin de registro (End Of Record) de dos bytes (0xFF 0x01). El fin de archivo (End Of File) se representa por otro código de 2 bytes (0xFF 0x02). Para el último registro del archivo, EOR y EOF se pueden representar como 0xFF 0x03. Si el archivo contiene un byte de valor 0xFF, este debe duplicarse antes del envío.

En modo bloque, se transmite un archivo como una serie de bloques de datos. Cada bloque comienza con una cabecera de 3 bytes. La cabecera esta formada por 1 byte que contiene las banderas del descriptor (fin de bloque, fin de fichero, reinicio de marcador) y 2 bytes con la cuenta de bytes, esto es, el número de bytes siguientes. La ventaja principal del uso de este modo es cuando la estructura de datos es de registro, pues el final del archivo aparece claramente identificado, por lo que se puede mantener activada la conexión utilizándola para varias transferencias.

Por último, el modo comprimido, raramente disponible, proporciona un método burdo de comprimir cadenas de bytes repetidos, por lo que para un usuario es mucho mejor utilizar la compresión que proporciona cualquiera de los programas de compresión disponibles antes que utilizar este modo.

El cliente de FTP.

En este punto nos limitaremos a exponer el uso interactivo del cliente de FTP por parte de un usuario. Una explicación de cómo usar el cliente FTP desde un programa de aplicación puede obtenerse mediante una consulta a las páginas de manual[5].

Las funciones esenciales de transferencia de archivos permiten a los usuarios copiar archivos de un sistema a otro, ver listados de directorios y realizar tareas normales, como cambiar de directorio y cambiar el nombre o borrar un archivo.

El cliente de FTP se encuentra en */usr/bin/ftp*, es ejecutado como:

[5] Además del cliente FTP aquí expuesto, que funciona en modo texto, existen multitud de clientes que funcionan en modo gráfico y que pueden ser usados para acceder a cualquier servidor de FTP, pues internamente ejecutan lo comandos, etc., aquí expuestos.

```
> ftp [nombre del ordenador]
```

Un ejemplo de conexión desde nuestro ordenador a glup.irobot.uv.es es el siguiente:

```
> ftp glup.irobot.uv.es
Connected to glup.irobot.uv.es (147.156.222.65).
220 Bienvenido al servicio de FTP del Instituto de Robotica
Name (glup.irobot.uv.es: quique): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /dist/ssh
250 Directory successfully changed.
ftp> get putty.zip
local: putty.zip remote: putty.zip
227 Entering Passive Mode (147,156,222,65,19,244)
150 Opening BINARY mode data connection for putty.zip (190784 bytes).
226 File send OK.
190784 bytes received in 0.0122 secs (1.5e+04 Kbytes/sec)
ftp> quit
221 Goodbye.
```

Como se ha visto, el nombre del ordenador es opcional, pudiendo ejecutarse el cliente de ftp sin ningún parámetro, en cuyo caso el cliente permanece a la espera de nuestros comandos. Si en este modo ejecutamos el comando “help”, podemos obtener un listado completo de los comandos soportados por nuestro cliente de ftp.

```
> ftp
ftp> help
Commands may be abbreviated.  Commands are:

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

Una descripción de la acción de cada uno de los comandos puede obtenerse introduciendo dentro del cliente de FTP “help <comando>”. Por ejemplo:

```
> ftp
ftp> help quote
```

```
quote           send arbitrary ftp command
ftp>
```

El servidor de FTP.

El servidor de FTP es el programa que se encarga de recibir las peticiones de los clientes y procesarlas. En la actualidad, existen, solo en Linux, varios servidores diferentes de FTP. Nosotros describiremos el Very Secure FTPD, que es un servidor con características añadidas que mejoran su seguridad.

El Very Secure FTP se encuentra en */usr/sbin/vsftpd*, siendo, de forma clásica, ejecutado en el arranque del sistema y permaneciendo a la escucha del puerto 21 TCP.

El servidor se encuentra configurado con unas opciones de funcionamiento por defecto, que pueden modificarse mediante los ficheros de configuración que se encuentren en el directorio */etc/vsftpd/*. Generalmente, en dicho directorio solo existe un fichero, de nombre *vsftpd.conf*, que permite configurar el funcionamiento básico del servidor así como sus opciones de seguridad[6].

El formato del fichero *vsftpd.conf* es muy simple. Cada línea que comienza por el símbolo # es un comentario, mientras que el resto de líneas son directivas que configuran el funcionamiento del servidor. Las líneas de directivas tienen el formato:

```
opcion=valor
```

Donde es muy importante destacar que no debe ponerse ningún espacio entre la opción, el signo "=" y el valor.

Las opciones que posee el servidor para su configuración pueden dividirse, para su estudio, en tres grandes grupos, opciones booleanas, opciones numéricas y opciones de cadena.

Las opciones booleanas solo pueden tomar los valores YES o NO. Para su estudio podemos dividirlas en cinco grupos, opciones de configuración del servidor, opciones generales, opciones de los usuarios del sistema, opciones de los usuarios anónimos y otras opciones.

Las opciones de configuración del servidor permiten especificar como se ejecuta el servidor, los modos de transferencia de archivos que permite, etc. Las opciones existentes son:

Opción	Descripción	Defecto	Dependencia
listen	Habilita la ejecución del servidor como un demonio independiente en lugar de ser ejecutado por xinetd. Es excluyente con *listen_ipv6*.	NO	listen_ipv6
listen_ipv6	Igual que *listen*, excepto que escucha conexiones en cualquier dirección IPv6 (::), por lo que también escucha conexiones IPv4. Es excluyente con *listen*.	NO	listen
pasv_enable	Habilita el modo pasivo del servidor.	YES	Ninguna.

[6] Con posterioridad veremos como configurar varios servidores (servidores virtuales) dentro de un mismo servidor físico.

Opción	Descripción	Defecto	Dependencia
port_enable	Habilita el modo activo del servidor.	YES	Ninguna.
connect_from_port_20	Permite al servidor iniciar conexiones activas utilizando el puerto 20, en caso contrario las conexiones activas se inician en un puerto no privilegiado.	NO	Ninguna.
tcp_wrappers	Indica que el servidor compruebe las reglas de los envolventes de acceso. Requiere que el servidor haya sido compilado para soportar los envolventes de acceso.	NO	Ninguna.

Las opciones generales indican si se permite subir y/o bajar ficheros, el modo de almacenamiento de los logs, etc. Sus principales opciones son:

Opción	Descripción	Defecto	Dependencia
download_enable	Permite la descarga de ficheros desde el servidor.	YES	Ninguna.
write_enable	Habilita el uso de los comandos que modifican el sistema de ficheros, permitiendo crear, borrar, etc., directorios y ficheros.	NO	Ninguna.
ascii_download_enable	Permite descargar ficheros en formato ASCII.	NO	Ninguna.
ascii_upload_enable	Permite subir ficheros en formato ASCII.	NO	Ninguna.
dirlist_enable	Permite el listado de los directorios.	YES	Ninguna.
use_localtime	Devuelve la hora de los ficheros y directorios en la hora local.	NO	Ninguna.
hide_ids	Oculta la información sobre usuarios y grupos mostrando estos datos como ftp.	NO	Ninguna.
dirmessage_enable	Permite que se muestre a los usuarios un mensaje cuando acceden a los directorios. El mensaje a mostrar se encuentra, por defecto, en el fichero *.message* de cada directorio, pero puede modificarse con la opción *message_file*.	NO	message_file
xferlog_enable	Habilita la escritura en el fichero de "log" de los ficheros que son descargados o subidos.	NO	vsftpd_log_file xferlog_file
xferlog_std_format	Especifica que el fichero de "log" se escriba en formato de vsftpd, en el fichero indicado por *vsftpd_log_file* si su valor es NO, o bien se escriba en el formato de xferlog, en el fichero indicado por *xferlog_file* si su valor es YES.	NO	Ninguna.

Las opciones que permiten configurar los permisos, etc., de los usuarios locales del sistema son:

Opción	Descripción	Defecto	Dependencia
local_enable	Permite el acceso a los usuarios locales, los cuales se especifican en */etc/passwd*.	NO	Ninguna.
check_shell	Para los usuarios locales, verifica que la shell que tienen asignada se encuentra en una lista de shells autorizadas que se indica en */etc/shells*.	YES	Ninguna.
chmod_enable	Permite a los usuarios locales ejecutar el comando chmod para cambiar los permisos de un fichero.	YES	Ninguna.
chroot_list_enable	Habilita la lista de usuarios que serán "encerrados" en su directorio raíz al acceder por FTP. La lista por defecto se encuentra en */etc/vsftpd.chroot_list*, pero puede modificarse con la opción *chroot_list_file*.	NO	chroot_local_user chroot_list_file

Opción	Descripción	Defecto	Dependencia
chroot_local_user	Invierte el funcionamiento de la opción *chroot_list_enable* y hace que los usuarios sean "encerrados" en su directorio raíz excepto los especificados en la lista.	NO	Ninguna.
passwd_chroot_enable	Si se encuentra habilitada la opción *chroot_local_user*, permite redirigir el directorio donde se encuentra "encerrado" el usuario. La aparición de /./ en el directorio que especifica el directorio raíz indica la localización donde se "encerrara" al usuario.	NO	chroot_local_user
text_userdb_names	Habilita que se muestren los nombres de los usuarios y grupos en lugar de sus números identificadores.	NO	Ninguna.
tilde_user_enable	Permite que se resuelvan las localizaciones indicadas como ~usuario como el directorio raíz del usuario indicado.	NO	Ninguna.
userlist_deny	Indica si la lista contiene los usuarios locales cuyo acceso esta permitido (valor NO) o si la lista contiene los usuarios locales cuyo acceso esta denegado (valor YES). En cualquier caso, la denegación se produce antes de que el usuario pueda introducir su contraseña.	YES	userlist_enable userlist_file
userlist_enable	Indica si esta habilitado el uso de la lista de usuarios permitidos o denegados que se encuentra en el fichero especificado por la opción *userlist_file*.	NO	userlist_file

Las opciones de los usuarios anónimos afectan a todos aquellos usuarios que no se identifican ante el sistema. Sus principales opciones son:

Opción	Descripción	Defecto	Dependencia
anonymous_enable	Permite el acceso de usuarios anónimos, que son reconocidos por "anonymous" y "ftp".	YES	Ninguna.
anon_mkdir_write_enable	Permite a los usuarios anónimos crear directorios si esta habilitada la opción de escribir.	NO	write_enable
anon_other_write_enable	Permite a los usuarios anónimos realizar otras operaciones diferentes de crear directorios y ficheros, tales como borrar y renombrar ficheros.	NO	write_enable.
anon_upload_enable	Permite a los usuarios anónimos subir ficheros si esta habilitada la opción de escribir.	NO	write_enable
anon_world_readable_only	Permite a los usuarios anónimos descargar ficheros solo si estos tienen permisos para ser leídos por todo el mundo.	YES	Ninguna.
chown_uploads	Habilita el cambio del propietario de los ficheros subidos por los usuarios anónimos al indicado en la opción *chown_username*.	NO	chown_username
deny_email_enable	Habilita una lista de correos a los que se les deniega el acceso al servidor con el usuario anónimo. La lista por defecto se encuentra en */etc/vsftpd.banned_emails*, pero puede modificarse con la opción *banned_email_file*.	NO	banned_email_file

Opción	Descripción	Defecto	Dependencia
secure_email_list_enable	Habilita que solo los usuarios anónimos cuya dirección de correo se encuentre en el fichero indicado por *email_password_file* puedan acceder como anónimos.	NO	email_password_file

Otras opciones booleanas que posee el servidor son:

Opción	Descripción	Defecto	Dependencia
allow_anon_ssl	Permite que los usuarios anónimos realicen conexiones utilizando SSL.	NO	ssl_enable
async_abor_enable	Habilita el comando conocido como "async ABOR", el cual permite a los clientes terminar una transferencia de archivos sin necesidad de cortar la conexión de FTP.	NO	Ninguna.
background	Devuelve el control inmediatamente a la shell o proceso que ejecuta el servidor.	NO	Ninguna.
dual_log_enable	Genera dos ficheros de log en lugar de uno solo.	NO	Ninguna.
force_dot_files	Muestra los ficheros y directorios que empiezan por . aunque no se use la opción -a. Los directorios . y .. son excluidos.	NO	Ninguna.
force_anon_data_ssl	Obliga a que las transferencias de datos de las conexiones anónimas se efectúen usando SSL. Debe estar habilitada la opción de uso de SSL.	NO	ssl_enable
force_anon_logins_ssl	Obliga a que el acceso de los usuarios anónimos se efectúe usando SSL. Debe estar habilitada la opción de uso de SSL.	NO	ssl_enable
force_local_data_ssl	Obliga a que las transferencias de datos de los usuarios locales se efectúen usando SSL. Debe estar habilitada la opción de uso de SSL.	YES	ssl_enable
force_local_login_ssl	Obliga a que el acceso de los usuarios locales se efectúe usando SSL. Debe estar habilitada la opción de uso de SSL.	YES	ssl_enable
guest_enable	Configura todos los de usuarios locales como virtuales, convirtiendo al usuario en el indicado en la opción *guest_username*.	NO	guest_username
log_ftp_protocol	Habilita que todos los comandos recibidos y las respuestas sean almacenadas. Es utilizado para "depurar" el servidor.	NO	Ninguna.
ls_recurse_enable	Permite ejecutar "ls" recursivo, esto es "ls -R".	NO	Ninguna.
no_anon_password	No solicita contraseña a los usuarios anónimos.	NO	Ninguna.
no_log_lock	Indica al servidor que no bloquee el acceso a los logs cuando va a escribir en ellos.	NO	Ninguna.
one_process_model	Habilita usar un solo proceso por conexión.	NO	Ninguna.
pasv_promiscuous	Deshabilita la comprobación de que la IP de la conexión de datos en modo pasivo se corresponde con la IP de la conexión de control.	NO	Ninguna.
port_promiscuous	Deshabilita la comprobación de que la IP a la que se establece la conexión se corresponde con la IP de la conexión de control.	NO	Ninguna.
run_as_launching_user	Ejecuta el servidor como el usuario que lo lanza.	NO	Ninguna.
session_support	Indica si el servidor debe identificar la sesión como un "login" y escribir los datos de acceso en los ficheros *wtmp* y *utmp*.	NO	Ninguna.
setproctitle_enable	Habilita la escritura en el fichero de "log" del sistema de la sesión de FTP establecida.	NO	Ninguna.

Opción	Descripción	Defecto	Dependencia
ssl_enable	Habilita la utilización de SSL. Requiere que el servidor haya sido compilado con soporte para SSL.	NO	Ninguna.
ssl_sslv2	Habilita el uso de SSL versión 2. Debe estar habilitada la opción de uso de SSL.	NO	ssl_enable
ssl_sslv3	Habilita el uso de SSL versión 3. Debe estar habilitada la opción de uso de SSL.	NO	ssl_enable
ssl_tlsv1	Habilita el uso de TLS versión 1. Debe estar habilitada la opción de uso de SSL.	YES	ssl_enable
syslog_enable	Indica que la salida de log se escriba en el log del sistema en vez de en el fichero */var/log/vsftpd.log*.	NO	Ninguna.
use_sendfile	Comprueba los beneficios de usar la llamada al sistema *sendfile()* en el servidor. No efectúa ninguna acción cara al usuario.	YES	Ninguna.
virtual_use_local_privs	Permite que los usuarios virtuales tengan los mismos privilegios que los usuarios locales. Por defecto, lo usuarios virtuales tienen los privilegios de los usuarios anónimos.	NO	Ninguna.

Las opciones numéricas son números enteros no negativos, permitiéndose la especificación en formato octal, debiendo utilizarse en tal caso el 0 como el primer dígito del número para indicar que su formato es octal. Las opciones numéricas son[7]:

Opción	Descripción	Defecto
accept_timeout	Número de segundos que se espera el establecimiento de una conexión en modo pasivo.	60
anon_max_rate	Máximo número de bytes por segundo que se transmiten a un usuario anónimo.	0
anon_umask	Valor de la máscara de usuario para la creación de ficheros por los usuarios anónimos.	077
connect_timeout	Número máximo de segundos que se espera la aceptación de una conexión en modo activo.	60
data_connection_timeout	Número máximo de segundos que permanece abierta una conexión de datos sin que se transmita ningún dato.	300
file_open_mode	Modo por defecto en el que son creados los ficheros subidos al servidor. A este modo por defecto se le aplica siempre la máscara del usuario.	0666
ftp_data_port	Puerto desde el que se establece la conexión de datos en modo activo.	20
idle_session_timeout	Número máximo de segundos que permanece abierta una conexión de control en espera de nuevos comandos.	300
listen_port	Puerto en el que se encuentra a la escucha el servidor si no es lanzado por el servidor de xinetd.	21
local_max_rate	Máximo número de bytes por segundo que se transmiten a un usuario local.	0
local_umask	Valor de la máscara de usuario para la creación de ficheros por usuarios locales.	077
max_clients	Número máximo de conexiones que acepta el servidor, siempre que no sea lanzado por el servidor de xinetd. Las conexiones que exceden dicho número reciben un mensaje de error.	0
max_per_ip	Número máximo de conexiones por dirección IP que acepta el servidor de forma simultanea. Solo es valido si no es lanzado por el servidor de xinetd.	0
pasv_max_port	Indica el valor máximo del puerto que puede usarse para la transmisión de datos en modo pasivo.	0

[7] Si una opción numérica tiene valor 0 indica sin limite.

Opción	Descripción	Defecto
pasv_min_port	Indica el valor mínimo del puerto que puede usarse para la transmisión de datos en modo pasivo.	0
trans_chunk_size	Limite de ancho de banda que puede ocupar el servidor.	0

Las opciones de cadena permiten especificar la localización de ficheros, etc., donde se encuentran listas de acceso, ficheros de "log", etc. Estas opciones son:

Opción	Descripción	Defecto
anon_root	Directorio donde acceden los usuarios anónimos.	Ninguno.
banned_email_file	Fichero con la lista de correos a los que se les deniega el acceso al servidor con el usuario anónimo	/etc/vsftpd/banned_email
banner_file	Fichero con el texto a mostrar cuando accede un usuario al servidor. Esta opción prevalece sobre la opción *ftpd_banner*.	Ninguno.
chown_username	Nombre del usuario al que se asignan los ficheros subidos por los usuarios anónimos.	root
chroot_list_file	Fichero que contiene la lista de usuarios a los que afectarán las opciones de ser "encerrados".	/etc/vsftpd/chroot_list
cmds_allowed	Lista de comandos, separados por comas, que aceptará el servidor de FTP.	Ninguno.
deny_file	Patrón que especifica los ficheros que no deben ser accesibles por los usuarios virtuales[8].	Ninguno.
dsa_cert_file	Localización del certificado DSA para conexiones SSL.	Ninguno.
dsa_private_key_file	Localización de la clave privada del certificado DSA para conexiones SSL.	La misma localización que el certificado DSA.
email_password_file	Fichero con las direcciones de correo a las que se autoriza el acceso.	/etc/vsftpd/email_passwords
ftp_username	Nombre del usuario al que se asignan los accesos anónimos. Su directorio de acceso es el directorio raíz de los usuarios anónimos.	ftp
ftpd_banner	Texto a mostrar cuando accede un usuario al servidor.	Ninguno.
guest_username	Nombre del usuario al que se asignan los accesos virtuales si la opción *guest_enable* los ha habilitado.	ftp
hide_file	Patrón que especifica los ficheros que deben ser ocultados a los usuarios virtuales.	Ninguno.
listen_address	Dirección IP en la que escucha el servidor si se ha lanzado directamente, esto es, sin el servidor xinetd.	Ninguno.
listen_address6	Dirección IPv6 en la que escucha el servidor si se ha lanzado directamente, esto es, sin el servidor xinetd.	Ninguno
local_root	Directorio donde el servidor coloca al usuario en caso de acceso por parte de un usuario local.	Ninguno
message_file	Nombre del fichero que contiene el mensaje a mostrar cuando se cambia de directorio.	.message
nopriv_user	Nombre del usuario con el que se ejecuta el servidor para no tener los privilegios de administrador.	nobody
pam_service_name	Nombre del servicio PAM que debe utilizar el servidor	ftp

[8] Un ejemplo de patrón es el siguiente: deny_file={*.mp3,*.mov,.private}

Opción	Descripción	Defecto
pasv_address	Cambia la dirección IP que envía el servidor ante una transferencia de datos en modo pasivo.	Ninguno.
rsa_cert_file	Localización del certificado RSA para conexiones SSL.	/usr/share/ssl/certs/vsftpd.pem
rsa_private_key_file	Localización de la clave privada del certificado RSA para conexiones SSL.	La misma localización que el certificado RSA.
secure_chroot_dir	Nombre del directorio vacío y sin permisos de escritura donde se pueden "encerrar" las conexiones que no requieran acceso al sistema de ficheros.	/usr/share/empty
ssl_ciphers	Selecciona el tipo de cifrado que se utilizará para las conexiones cifradas.	DES-CBC3-SHA
user_config_dir	Especifica el directorio donde se escribirán las opciones de acceso particulares para cada usuario, y que pueden sobrescribir las opciones del fichero de configuración por defecto.	Ninguno
user_sub_token	Genera un directorio para los usuarios virtuales en el cual son "encerrados"[9].	Ninguno
userlist_file	Nombre del fichero que contiene los usuarios a los que se les deniega el acceso si la opción *userlist_enable* está activada.	/etc/vsftpd/user_list
vsftpd_log_file	Nombre del fichero donde se escribe el "log" del servidor	/var/log/vsftpd.log
xferlog_file	Nombre del fichero donde se escribe el "log" del servidor.	/var/log/xferlog

Un ejemplo de fichero */etc/vsftpd/vsftpd.conf* es el siguiente:

```
# Ejecutamos el servidor como demonio
# Podriamos poner listen_ipv6=YES para escuchar en IPv4 e IPv6.
listen=YES
# Permitimos las conexiones activas desde el puerto 20
connect_from_port_20=YES
# Habilitamos el uso de los envolventes de acceso
tcp_wrappers=YES
# Indicamos el servicio PAM que utiliza el servidor
pam_service_name=vsftpd
# Habilitamos la escritura del fichero de log
xferlog_enable=YES
# Indicamos el modo de log deseado
xferlog_std_format=NO
# Habilitamos la muestra de mensajes al cambiar de directorio
dirmessage_enable=YES
# Permitimos la escritura de ficheros
write_enable=YES
# Permitimos el acceso a usuarios locales
local_enable=YES
# Denegamos el acceso a los usuarios no deseados (root, etc.)
userlist_enable=YES
# Especificamos la mascara de escritura de los ficheros
local_umask=022
# Permitimos el acceso de usuarios anónimos
anonymous_enable=YES
```

[9] Un ejemplo es el siguiente:user_sub_token=/home/virtual/$USER, que si accede un usuario de nombre quique, crearía un directorio /home/virtual/quique y lo "encerraría" en su interior.

Otro ejemplo de fichero */etc/vsftpd/vsftpd.conf*, con una configuración utilizada para permitir que tan solo usuarios anónimos puedan acceder a ficheros del servidor es el siguiente:

```
# Ejecutamos el servidor como demonio
# Podriamos poner listen_ipv6=YES para escuchar en IPv4 e IPv6.
listen=YES
# Permitimos las conexiones activas desde el puerto 20
connect_from_port_20=YES
# Habilitamos el uso de los envolventes de acceso
tcp_wrappers=YES
# Indicamos el servido PAM que utiliza el servidor
pam_service_name=vsftpd
# Habilitamos la escritura del fichero de log
xferlog_enable=YES
# Indicamos el modo de log deseado
xferlog_std_format=YES
# Indicamos el fichero donde el log se almacenará
xferlog_file=/var/log/vsftpd.log
# Habilitamos la muestra de mensajes al cambiar de directorio
dirmessage_enable=YES
# Mostramos un mensaje en el acceso al servidor
ftpd_banner=Bienvenido al servicio de FTP
# Ocultamos los usuarios y grupos de los ficheros
hide_ids=YES
# Permitimos el acceso de usuarios anónimos
anonymous_enable=YES
# Limitamos el numero máximo de clientes a 1000
max_clients=1000
# Limitamos el numero máximo de conexiones por cliente a 1
max_per_ip=1
```

Servidores virtuales.

Si un ordenador posee varios interfaces de red, es posible ejecutar varias instancias del servidor vsftpd de forma que cada una de ellas atienda las peticiones de un interfaz de red, especificando en el fichero de configuración de cada uno de los servidores las direcciones IP que escucha, los directorios que utiliza, etc.

Por ejemplo, supongamos que tenemos un ordenador con dos interfaces de red, un interfaz con una dirección IP pública, por ejemplo 147.156.222.65, y el otro interfaz con una dirección privada, por ejemplo 192.168.1.1, de forma que queremos que el interfaz público atienda las peticiones de la red Internet, mientras que el interfaz privado atienda las peticiones de nuestra Intranet.

La configuración de estos dos servidores virtuales es tan sencilla como crear, dentro del directorio */etc/vsftpd* dos ficheros, que llamaremos *publico.conf* y *privado.conf*[10], donde especificaremos las opciones de cada uno de ellos, especialmente la dirección IP en la que permanecen a la escucha mediante la opción *listen_address* que vimos con anterioridad.

[10] Los nombres de los ficheros de configuración de los servidores virtuales deben acabar siempre con la extensión .conf.

De esta forma, dos posibles ficheros de configuración de los servidores virtuales serían:

```
# Servidor publico
# Aqui no es posible usar listen_IPv6 al indicar una IPv4
# en listen_address
listen=YES
listen_address=147.156.222.65
ftpd_banner=Bienvenido al servidor FTP publico
...

# Servidor privado
# Aqui no es posible usar listen_IPv6 al indicar una IPv4
# en listen_address
listen=YES
listen_address=192.168.1.1
ftpd_banner=Bienvenido al servidor FTP privado
...
```

Seguridad del servidor de FTP.

Como hemos visto en las opciones de configuración, el servidor posee dos modos de funcionamiento, modo activo y modo pasivo.

En la especificación original de FTP, el funcionamiento del servidor era en modo activo. Para ello, se definía el comando PORT como el método por defecto para establecer una conexión de datos, de forma que el servidor, desde su puerto TCP 20, intentaba establecer una conexión con la dirección IP y el puerto TCP del cliente que le había sido indicado por la conexión de control.

Este funcionamiento, que es seguro para el servidor, tiene un inconveniente, y es la presencia de un cortafuegos en el cliente, pues de forma general, muchas redes configuran sus cortafuegos de forma que no permiten el establecimiento de una conexión TCP a sus ordenadores, impidiendo el funcionamiento en modo activo del servidor.

Para permitir que estos clientes puedan acceder a los datos del servidor, debe habilitarse el modo pasivo, en el cual el servidor es el que acepta una nueva conexión, que es la conexión de datos, desde el cliente.

Esta conexión se realiza mediante el comando PASV, el cual envía el cliente para indicarle al servidor que consiga un puerto adicional para la conexión de datos y envíe el puerto TCP elegido al cliente por la conexión de control, encargándose el cliente de establecer la conexión de datos con el puerto especificado en el servidor.

El modo de funcionamiento en modo pasivo del servidor presenta problemas de seguridad debido a que se permite establecer nuevas conexiones desde clientes a puertos del servidor sin un cierto control de seguridad.

Sin embargo, los cortafuegos actuales de Linux permiten realizar un seguimiento de las conexiones pasivas de datos, comprobando que el cliente que solicita la conexión al puerto especificado por el servidor se corresponde con el cliente al que se le indicó ese puerto, etc., y que por tanto la conexión se establece para el envío o recepción de datos[11].

Protocolo Trivial de Transferencia de Archivos (TFTP).

En determinadas circunstancias, como por ejemplo la lectura de los programas y archivos de configuración de un router, un conmutador o un ordenador sin disco, es necesario transferir ficheros utilizando un protocolo muy sencillo. Para estos casos, se desarrollo el protocolo trivial de transferencia de archivos (Trivial File Transfer Protocol). TFTP transfiere datos utilizando datagramas de UDP en lugar de conexiones de TCP, lo que simplifica en gran medida el software de comunicaciones necesario[12].

TFTP posee como características fundamentales:

- Envío de bloques de datos de 512 bytes (excepto el último).
- Añadir una sencilla cabecera de 4 bytes a cada bloque de datos.
- Numerar los bloques empezando por 1.
- Admitir transferencia de archivos ASCII o binarios.
- Posibilidad de leer o escribir un archivo remoto.
- No contempla la autenticación del usuario.

En TFTP, el cliente TFTP comienza obteniendo un puerto y enviando, a continuación, el mensaje de petición de lectura o de petición de escritura al puerto UDP 69 del servidor. Entonces el servidor identifica un puerto diferente para el resto de la transferencia de archivos y dirige desde ese puerto todos sus mensajes al puerto del cliente.

La transferencia de datos se realiza intercambiando bloques de datos numerados de forma secuencia, empezando en 1 y ACKs de reconocimiento de los bloques con el número de secuencia del bloque que esta confirmando como recibido, de forma que el emisor tiene que esperar el ACK de un bloque antes de enviar el siguiente. Si no recibe el ACK dentro de un plazo determinado de tiempo, se vuelve a enviar el bloque afectado. De forma similar, si el receptor no recibe datos durante un plazo determinado de tiempo, retransmite un ACK[13].

Existen cinco tipos de mensajes intercambiados en el protocolo, que comienzan con un código de operación que indica el tipo de unidad de datos de protocolo (Protocol Data Unit). El formato de las PDU se muestra a continuación:

[11] En temas posteriores veremos los cortafuegos de Linux y su configuración, incluyendo el análisis de las conexiones FTP de datos en modo pasivo.
[12] Téngase en cuenta la sencillez del protocolo de transporte UDP frente al protocolo de transporte TCP.
[13] Funciona básicamente como cualquier protocolo de comunicación de parada y espera.

	2 bytes	Cadena	1 byte	Cadena	1 byte
Petición de lectura (RRQ):	Código de op.=1	Nombre de archivo	0	Modo	0

	2 bytes	Cadena	1 byte	Cadena	1 byte
Petición de escritura (WRQ):	Código de op.=2	Nombre de archivo	0	Modo	0

	2 bytes	2 bytes	
Datos (DATA):	Código de op.=3	Nº de bloque	Datos

	2 bytes	2 bytes
Confirmación (ACK):	Código de op.=4	Nº de bloque

	2 bytes	2 bytes	Cadena	1 byte
Error (ERROR):	Código de op.=5	Código de error	Mensaje de error	0

Los mensajes de error indican condiciones como “archivo no encontrado” o “falta de espacio para escribir el archivo en el disco”.

Puede observarse que el tamaño de las peticiones de lectura y de escritura varía dependiendo del tamaño de los campos de nombre de archivo y modo, que contienen cada uno una cadena de texto ASCII seguida de un byte con valor 0. El campo modo contiene los valores “netascii”, para indicar código ASCII u “octet” para indicar bytes, que generalmente corresponden a código binario.

El cliente de TFTP.

El cliente de TFTP se encuentra en */usr/bin/tftp* y se ejecuta como:

```
>tftp [nombre del servidor]
```

Un ejemplo de ejecución de un cliente de TFTP, donde se solicita un fichero a un servidor de TFTP es:

```
>tftp glup.irobot.uv.es
tftp> verbose
tftp> get X86PC/UNDI/linux-install/linux.0
Received 12498 bytes in 0.1 seconds
tftp> quit
```

Si no especificamos ningún nombre de ordenador, el cliente solicita el nombre de un ordenador. Si no especificamos ningún nombre (pulsando Enter) el cliente permanece a la espera de nuestros comandos. Si en este modo ejecutamos el comando “help”, podemos obtener un listado completo de los comandos soportados por nuestro cliente de tftp.

```
> tftp
(to)
usage: connect host-name [port]
tftp> help
tftp-hpa 5.2
Commands may be abbreviated.  Commands are:

connect  connect to remote tftp
mode     set file transfer mode
put      send file
get      receive file
quit     exit tftp
verbose  toggle verbose mode
trace    toggle packet tracing
literal  toggle literal mode, ignore ':' in file name
status   show current status
binary   set mode to octet
ascii    set mode to netascii
rexmt    set per-packet transmission timeout
timeout  set total retransmission timeout
?        print help information
help     print help information
```

El servidor de TFTP.

El servidor de TFTP se encuentra en */usr/sbin/in.tftpd*, y es ejecutado por defecto desde el servidor xinetd[14]. El servidor posee una configuración por defecto que puede ser modificada desde la línea de comandos, siendo las opciones de configuración disponibles las siguientes:

Opción	Descripción
-l	Ejecuta el servidor directamente, sin ser ejecutado por el servidor xinetd.
-a [dirección][:puerto]	Si el servidor escucha directamente el puerto sin ser lanzado por el servidor xinetd, especifica la dirección IP y el puerto en que permanece a la escucha.
-c	Permite la creación de nuevos ficheros. Por defecto tftp solo permite escribir en ficheros ya existentes.
-s	Especifica el directorio raíz al que accede el servidor, limitando el acceso a ficheros fuera de ese directorio raíz.
-u usuario	Especifica el usuario como el que se ejecuta el servidor.
-U mascara de usuario	Especifica la máscara con que se crearán los ficheros. La máscara por defecto es cero (nadie puede leerlos o escribir en ellos).
-p	Indica que no se ejecute ninguna comprobación de seguridad adicional excepto las normales para el usuario especificado por la opción –u.
-t timeout	Cuando se ejecuta desde el servidor xinetd, especifica el tiempo que permanece en espera de posteriores conexiones antes de terminar su ejecución.
-m fichero	Especifica un fichero que contiene una serie de operaciones, que pueden ser condicionales, que deben ser ejecutadas.
-v	Incrementa la información facilitada por el servidor.
-r opción de tftp	Indica que la opción de tftp especificada no debe ser admitida. Las opciones que es posible especificar se encuentran en el RFC 2347.
-V	Muestra la versión y configuración por defecto del servidor.

[14] En temas posteriores veremos el servidor xinetd y su funcionamiento.

El servidor xinetd arranca por defecto el servicio de TFTP con la opción –s, indicando como directorio raíz el directorio */tftpboot*. Además, y si no se especifica ninguna opción, el servidor de TFTP solo considerá accesibles los ficheros que pueden ser leídos por cualquier usuario, a menos que las opciones -u y -p sean especificadas, y considera que pueden escribirse solamente los ficheros que pueden ser escritos por cualquier usuario.

Téngase en cuenta que, al no requerir ningún usuario o contraseña en el ordenador, cualquier usuario y ordenador de la red puede acceder a la información contenida en el servidor de TFTP, por lo que deben extremarse las medidas de seguridad, entre ellas el uso de la opción -u para ejecutar el servidor con un usuario que tenga unos privilegios muy limitados en el ordenador.

Ejercicios.

1- Configurar el servidor FTP de un ordenador de forma que solo se permita el acceso a los usuarios anónimos, los cuales deben poder escribir ficheros en un directorio del servidor, pero no deben poder descargarse los archivos que ellos mismos suban.

2- Configurar un servidor de FTP de forma que permita el acceso únicamente a un usuario de nombre “quique”, el cual debe poder leer y escribir ficheros en el servidor.

3- Un ordenador dispone de dos interfaces de red, uno con la IP pública 147.156.222.65 y otro con la IP privada 192.168.1.1. Configurar el servidor de FTP de forma que permita el acceso solo de usuarios anónimos, y solo para lectura, por la IP pública, mientras que permita el acceso solo de usuarios y con permisos de lectura y escritura, por la IP privada.

4- Configurar un servidor, que ejecuta los servicios de FTP y de TFTP, de forma que los usuarios del ordenador puedan, mediante FTP, subir y descargar ficheros, mientras que tanto los usuarios anónimos de FTP como el servidor de TFTP puedan descargar los ficheros existentes, pero no crear nuevos ficheros.

Apéndice A: Comandos de control de FTP.

Comandos de autorización de acceso a archivos.

Comando	Definición	Parámetro(s)
USER	Identifica al usuario	Identificador
PASS	Suministra una contraseña	Contraseña
ACCT	Suministra una cuenta	ID de la cuenta
REIN	Reinicializa al estado de comienzo	Ninguno
QUIT	Desconexión	Ninguno
ABOR	Aborta el comando anterior y la transferencia de datos asociada	Ninguno

Comandos de gestión de archivos y directorios.

Comando	Definición	Parámetro(s)
CWD	Cambia a otro directorio del servidor	Nombre de directorio.
CDUP	Cambia al directorio padre	Ninguno.
DELE	Borra un archivo	Nombre de archivo.
LIST	Lista información de archivos	Nombre de directorio o listado de archivos.
MKD	Crea un directorio	Nombre de directorio.
NLST	Lista de archivos de un directorio	Nombre de directorio.
PWD	Imprime el nombre del directorio de trabajo	Ninguno.

RMD	Elimina un directorio	Nombre de directorio.
RNFR	Identifica un archivo para cambiarlo de nombre	Nombre de archivo.
RNTO	Cambia de nombre un archivo	Nombre de archivo.
SMNT	Monta un sistema de archivos diferente	Identificador.

Comandos que definen el tipo, la estructura y el modo.

Comando	Definición	Parámetro(s)
TYPE	Identifica el tipo de datos y opcionalmente, el formato de impresión, si existe, para la transferencia.	A (ASCII), E (EBCDIC), B (Binario), I (Imagen binaria), N (No-impresión), T (*telnet*), C (ASA)
STRU	Organización del archivo.	F (archivo), R (registro)
MODE	Formato de transmisión.	S (flujo), B (bloque), C (comprimido)

Comandos que realizan la transferencia de archivos.

Comando	Definición	Parámetro(s)
ALLO	Reserva espacio suficiente para los datos que siguen	Número entero de bytes
APPE	Añade un archivo local a uno remoto	Nombres de los archivos
PASV	Pide al servidor que identifique una dirección de IP y un puerto para que el cliente inicie una sesión de datos	Ninguno. El servidor devolverá una dirección de IP y un número de puerto.
PORT	Identifica una dirección de red y un puerto para que el servidor inicie una conexión de datos	Dirección de IP y número de puerto
REST	Identifica un marcador de reinicio, seguido del comando de transferencia que hay que reiniciar	Valor del marcador
RETR	Recupera un archivo	Nombre(s) de archivo(s)
STOR	Guarda un archivo	Nombre(s) de archivo(s)
STOU	Crea un archivo con un nombre único	Nombre de archivo

Otros comandos de información de usuario.

Comando	Definición	Parámetro(s)
HELP	Devuelve información sobre la implementación del servidor.	Ninguno
NOOP	Pide al servidor que responda OK.	Ninguno
SITE	Usado para subcomandos específicos del servidor, que no forman parte del estándar, pero se pueden necesitar para ese servidor.	Ninguno
SYST	Pide al servidor que identifique su sistema operativo.	Ninguno
STAT	Solicita información sobre los parámetros y estado de la conexión.	Ninguno