

Unidad 4



Servicio de transferencia de archivos

Servicios de
red e internet



Índice



4.1. Tipos de transferencia de archivos

- 4.1.1. Protocolo FTP
- 4.1.2. Protocolo TFTP
- 4.1.3. Protocolo FTP TLS/SSL y SFTP

4.2. Instalación y configuración del servicio de transferencia de archivos

4.3. Tipos de usuarios y accesos al servicio: permisos y cuotas

- 4.3.1. Modos de acceso al servicio
- 4.3.2. Usuario anónimo
- 4.3.3. Usuarios locales de equipo

4.4. Modos de conexión del cliente

- 4.4.1. Modo activo
- 4.4.2. Modo pasivo



Introducción

El servicio de transferencia de archivos entre varios equipos se crea cuando en las redes surge la necesidad de poder pasar cierta cantidad de archivos o datos mediante la conexión de red, además, esto debe de hacerse con cierta rapidez y de manera fiable.

Con este servicio conseguimos que los clientes ligeros que se encuentran conectados a la red, se puedan descargar cierta cantidad de recursos y poder ir accediendo a ellos.

A lo largo de esta unidad veremos como conectarnos a los servicios FTP, la manera más común de configurarlos y distintos medios de conexión que nos encontramos.

Al finalizar esta unidad

- + Seremos capaces de establecer la utilidad y modo de operación el servicio de transferencia de archivos.
- + Sabremos como instalar y configurar servidores de transferencia de archivos.
- + Conoceremos como crear usuarios para acceso remoto al servidor.
- + Podremos configurar el acceso anónimo.
- + Distinguiremos los límites en los distintos modos de acceso.
- + Sabremos comprobar el acceso al servidor en modo activo y en modo pasivo.
- + Conoceremos como realizar pruebas con clientes en línea de comandos y con clientes en modo gráfico.
- + Podremos utilizar el navegador como cliente del servicio de transferencia de archivos.



4.1.

Tipos de transferencia de archivos

4.1.1. Protocolo FTP

El protocolo FTP, o *File Transfer Protocol* se trata de un protocolo orientado a la conexión del tipo TCP y que está diseñado para que se transmitan de manera rápida y bidireccional.

Los datos aparecen transparentes para los clientes por el modo en el que se almacenan, ya que se consigue que sean entendibles por ambas partes. Este protocolo viene determinado por la RFC 959.

Los formatos de datos que más se usan en la transferencia de archivos son:

- > **ASCII:** el más común y principalmente usado para archivos de texto.,
- > **Imagen (binario):** si mandamos cadenas de bits contiguos, estamos mandando este tipo de formato.
- > **EBCDIC.** Se indica su uso cuando el cliente usa este formato en concreto.

Como se usan formatos para los datos, realmente es totalmente indiferente que sistemas de archivos usen los dos conectores de la información, tanto cliente como servidor.

Para esto, el protocolo realiza dos tipos de conexiones:

- > **Conexión de control.** Se usan comandos que interactúan con el servidor. El servidor recibe las peticiones del control en el puerto 21.
- > **Conexión de datos.** Se usa para transmitir propiamente los datos. El servidor se prepara por defecto para recibir los datos a través del puerto 20. Pero, esto último puede cambiar dependiendo del modo de conexión.

El protocolo puede usarse tanto en terminal como por interfaz gráfica.

El protocolo FTP funciona principalmente en base a los siguientes elementos:

- > **Servidor-PI (Server Protocol Interpreter):** es el proceso que se encarga de escuchar las peticiones que realizan los clientes.
- > **Servidor-DTP (Server Data Transfer Process):** es un proceso interno en escucha para establecer conexiones de datos con clientes y transferir información. Se encarga también de la preparación de los datos antes de enviarlos.
- > **Servidor-FTP:** es el conjunto del *Servidor-PI* y el *Servidor-DTP*.
- > **Cliente-PI (Client Protocol Interpreter):** proceso ocurrido en el cliente que se encarga del inicio de la conexión con el *Servidor-PI* para ejecutar comandos de control.



- > **Cliente-DTP (Client Data Transfer Process):** es el proceso del cliente que se encuentra a la espera de recibir conexión de datos del *Servidor-DTP* y así poder iniciar la transferencia.
- > **Cliente-FTP:** conjunto que forman el *Cliente-PI* y el *Cliente-DTP*.

La siguiente imagen podría ser ejemplo de cómo funciona el protocolo FTP:

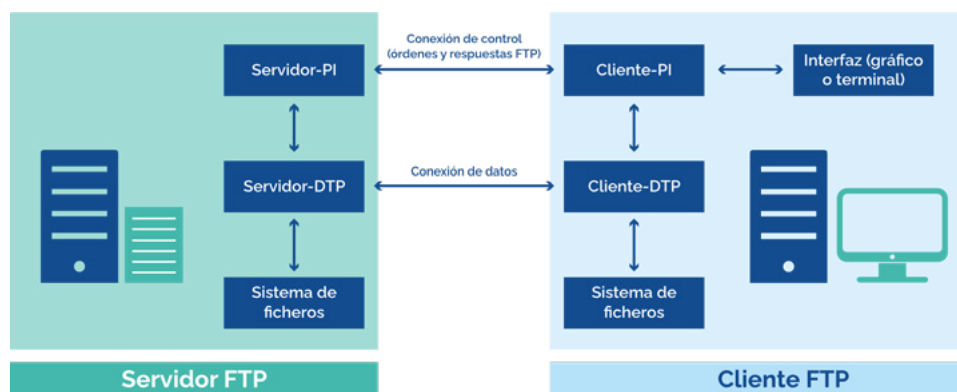


Imagen 1. Estructura del funcionamiento de FTP

Cada vez que realizamos alguna operación desde el Cliente-PI, identificamos esta operación con un comando ejecutado desde la interfaz. A veces, los nombres de los comandos coinciden con las operaciones, pero esto no siempre es así.

Podemos englobar las operaciones más comunes del protocolo FTP como vemos en el siguiente cuadro.

Operaciones más comunes de FTP agrupadas por categorías		
Operaciones de control de acceso	USER	Comando usado con un argumento que identifica que usuario se va a conectar.
	PASS	Comando usado con un argumento que identifica la contraseña que va a tener el usuario anteriormente especificado.
	CWD	Comando con el que cambiamos el directorio de trabajo actual desde un cliente.
	QUIT	Comando para finalizar la sesión FTP.
Operaciones de parámetros de transferencia	PORT	Se especifica cual va a ser el puerto usado en la transferencia de datos.
	PASV	La orden solicitada al Servidor-DTP para que escuche en modo pasivo por un puerto que no es el asignado por defecto para transmisiones.
	TYPE	Especifica que tipo de fichero se va a transmitir.
Operaciones de servicio	LIST	Lista el contenido del directorio actual
	MKD	Crea un directorio en el servidor
	DELE	Borra un fichero del servidor
	PUT	Loja un fichero
	GET	Descarga mediante la conexión de datos, un fichero.



Para ver cómo funciona esto, podemos conectarnos a un servidor FRP anónimo, es decir, no necesitamos credenciales para su acceso.

Los pasos para seguir serían los siguientes:

1. Lanzamos desde una terminal, el comando:

`ftp ftp.rediris.es`

2. Lanzamos por ejemplo el comando `ls`.

```
root@servidor:~# ftp ftp.rediris.es
Connected to ftp.rediris.es.
220- Bienvenido al servicio de replicas de RedIRIS.
220- Welcome to the RedIRIS mirror service.
220 Only anonymous FTP is allowed here
Name (ftp.rediris.es:profesor): anonymous
230- RedIRIS - Red Académica y de Investigación Española
230- RedIRIS - Spanish National Research Network
230-
230- ftp://ftp.rediris.es -- http://ftp.rediris.es
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful
```

Imagen 1. Conexión FTP

Podemos ver que, en principio, todo funciona de manera correcta con nuestra primera interacción con un FTP.

4.1.2. Protocolo TFTP

El protocolo TFTP (*Trivial File Transfer Protocol*) es una modificación del protocolo FTP que ya hemos estudiado pero que procesa la información enviada o recibida por el servidor de manera más rápida porque no usa ni autenticación ni operaciones complejas. Este tipo de comunicaciones se basan en UDP, porque queremos tener antes rapidez que seguridad. Se rige por la RFC 1350.

Las principales características de TFTP son:

- > El servidor escucha las peticiones en el puerto 69 UDP.
- > No se pueden crear, borrar o nombrar directorios.
- > Admite tres tipos de ficheros para transmitir:
 - » **Netascii**: es similar al ASCII.
 - » **Octet**: similar al binario.
 - » **Mail**: es el utilizado en la mensajería de correo electrónico.
- > No tiene implementadas ni identificación ni encriptación.

En este protocolo, tenemos cinco tipos de paquetes distintos dependiendo de que operación queramos realizar:

- > **READ REQUEST**. Se solicita realizar una operación de lectura en el servidor.
- > **WRITE REQUEST**. Se solicita realizar una operación de escritura en el servidor.
- > **DATA**. Se indica que son datos por transferir.
- > **ACK**. Se indica que la transmisión ha sido correcta.
- > **ERROR**. Se indica que ha habido algún error durante la transmisión de información.

Los principales usos que se le dan a un servidor TFTP son:

- > Transferencia de imágenes de sistemas operativos.
- > Transferencia de ficheros de configuración para redes.
- > Transferencia de sistemas de arranque para ciertos clientes.



4.1.3. Protocolo FTP TLS/SSL y SFTP

Cuando enviamos algo de información mediante el protocolo FTP, esto se hace en texto plano, ya que se no tiene ningún mecanismo de seguridad implementado, por lo que necesitamos algo de encriptación para que las comunicaciones en protocolo FTP puedan viajar encriptadas.

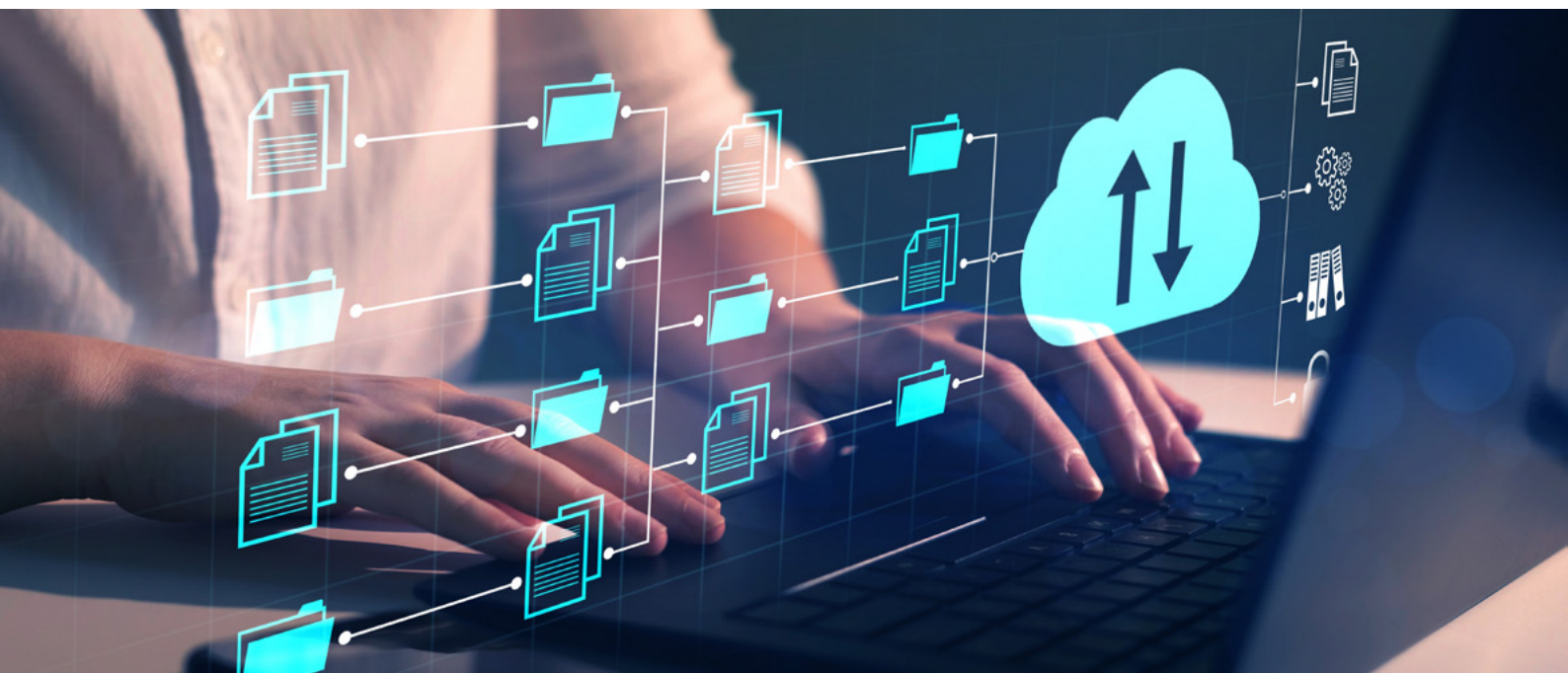
El protocolo *FTPS* o *FTP sobre TLS o SSL* se trata de una variante del protocolo FTP con una capa de encriptación agregada justo debajo de la capa de aplicación de los paquetes. Con esto, se usan algoritmos de encriptación para que el canal entre cliente y servidor quede encriptado. Se rige por la RFC 4217.

Por otro lado, existe el protocolo *SFTP* o *SSH File Transfer Protocol* que está totalmente diseñado para agregar a las conexiones SSH las características FTP. Esto se entiende mejor del siguiente modo:

1. Lo primero que establecemos es una conexión SSH entre un cliente y un servidor.
2. En segunda instancia, a través de la sesión SSH, se intercambia la información por FTP.

Esta información se encuentra encriptada y viaja por el puerto 22 TCP, como SSH.

Se puede abrir una conexión segura con FTPS sin necesidad de tener que abrir una conexión SSH.



4.2. Instalación y configuración del servicio de transferencia de archivos

Vamos a explicar ahora como sería la instalación del servicio y su configuración básica en un servidor Ubuntu. Los procesos que se encargan de cumplir con las peticiones FTP son los *daemons* o demonios., siendo el más usado *vsFTPd*.

Los pasos para realizar la instalación del servicio son:

1. En una terminal, como usuario administrador, actualizamos repositorios.
2. Una vez tenemos actualizados los paquetes, instalamos el servicio:

`apt install vsftpd`

```
root@servidor:~# apt install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  vsftpd
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 122 kB de archivos.
Se utilizarán 322 kB de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu impish/main amd64 vsftpd amd64 3.0.3-13build1 [122 kB]
Descargados 122 kB en 0s (412 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete vsftpd previamente no seleccionado.
(Leyendo la base de datos ... 107830 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../vsftpd_3.0.3-13build1_amd64.deb ...
Desempaquetando vsftpd (3.0.3-13build1) ...
Configurando vsftpd (3.0.3-13build1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /lib/systemd/system/vsftpd.service.
Procesando disparadores para man-db (2.9.4-2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.
```

Imagen 2. Instalación de *vsftpd*

3. Comprobamos que se encuentra activo por defecto:

`systemctl status vsftpd`

```
root@servidor:~# systemctl status vsftpd
• vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-07-05 12:25:50 UTC; 49s ago
     Process: 4457 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
    Main PID: 4458 (vsftpd)
       Tasks: 1 (limit: 2209)
      Memory: 676.0K
         CPU: 2ms
    CGroup: /system.slice/vsftpd.service
           └─4458 /usr/sbin/vsftpd /etc/vsftpd.conf

jul 05 12:25:50 servidor systemd[1]: Starting vsftpd FTP server...
jul 05 12:25:50 servidor systemd[1]: Started vsftpd FTP server.
root@servidor:~# _
```

Imagen 3. Estado del servicio: activo

Como en los demás servicios, se activan y se paran con el comando `systemctl` y sus respectivos argumentos.



4. Vamos ahora a comprobar que se ha creado el archivo de configuración del servicio, que es `/etc/vsftpd.conf` y tiene un aspecto como el siguiente:

```
GNU nano 5.6.1 /etc/vsftpd.conf
# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# Read 155 lines
```

Imagen 4. Fichero `/etc/vsftpd.conf`

En este archivo, los parámetros de configuración del servicio aparecen en forma de pares del siguiente modo: *directiva* = *valor*.

Las principales directivas de este servicio son:

- > **Anonymous_enable:** habilita el acceso a usuarios anónimos, sin necesidad de identificación.
- > **Local_enable:** permite que accedan al servicio FTP usuarios locales del sistema operativo.
- > **Write_enable:** permite que cuando un usuario se conecte, pueda crear directorios, ficheros, renombramiento, etc.
- > **Local_umask:** indica la máscara por defecto que van a tener los archivos creados en el servidor por usuarios que hayan accedido.
- > **Anon_upload_enable:** permite al usuario anónimo subir ficheros al servidor.
- > **Download_enable:** permite que se puedan descargar ficheros alojados en el servidor.
- > **Xferlog_enable:** permite que exista el registro de actividad, que se alojará en un fichero especificado en la directiva `vsftpd_log_file` y que por defecto es `/var/log/vsftpd.log`.
- > **Local_max_rate:** indica la ratio máxima de transferencia por cada usuario local. Siempre se indica en bytes/segundos.
- > **Max_per_ip:** indica cual es el número máximo de clientes que pueden estar conectados por IP.



4.3.

Tipos de usuarios y accesos al servicio: permisos y cuotas

Para poder acceder a la información que tenemos alojada en nuestro servidor de FTP, necesitamos tener usuario.

Todas las características de usuarios, permisos y distribución de ficheros y directorios serán similares a las de Linux, por lo que es necesario entender estas para trabajar con FTP.

4.3.1. Modos de acceso al servicio

Cuando accedamos a la información, tenemos dos métodos de hacerlo, o bien desde una línea de comandos donde los usuarios indican instrucciones o bien desde un entorno gráfico que sirve para interactuar con los elementos visuales, pero al final, realizan las mismas acciones.

Modo de acceso mediante terminal

Cuando usamos este tipo de acceso, su principal característica es que consumen pocos recursos y que se puede usar desde cualquier sistema operativo con el servicio FTP activo, independientemente de que tenga o no entorno gráfico.

Lo normal y más común es usarlo desde un terminal y siempre se va a abrir usando el comando `ftp`. Con cada comando que usemos, se van ejecutando las ordenes en el servidor y nos irá devolviendo respuestas a nuestras solicitudes.

A continuación, vamos a enumerar los principales comandos que se usan en la conexión con el servidor FTP, destacando que el primero de ellos se usa para la conexión con el servidor y los siguientes se usan una vez tenemos la sesión iniciada.

Comandos FTP para el acceso mediante terminal	
Comando	Descripción
<code>ftp servidor_ftp</code>	Se establece la conexión con alguno de los servidores remotos específicos.
<code>type</code>	Se muestra el modo de transferencia en el que se está trabajando, ya sea binario o texto.
<code>ascii</code>	Se cambia el modo de transferencia a ASCII.
<code>binary</code>	Se cambia el modo de transferencia a binario.
<code>status</code>	Se visualiza el estado de la configuración del servidor.
<code>user</code>	Conecta con el servidor FTP con un usuario concreto.
<code>pwd</code>	Se muestra el nombre completo del directorio de trabajo actual.

Comandos FTP para el acceso mediante terminal	
Comando	Descripción
ls	Se muestra el contenido de un directorio. Podemos especificar la opción -l para sacar información detallada del contenido.
mkdir → Linux md → Windows	Se crea un directorio en el servidor FTP que estará alojado en el directorio actual.
rmdir → Linux rd → Windows	Se borra un directorio del servidor.
get fichero_origen fichero_destino	Se descarga el <i>fichero_origen</i> en el directorio actual y se guarda con el nombre <i>fichero_destino</i> . Si el segundo de los nombres no se especifica, se descarga con el mismo nombre.
put fichero_origen fichero_destino	Se carga el <i>fichero_origen</i> en el directorio actual del servidor guardándolo con el nombre especificado en <i>fichero_destino</i> . Si no se especifica el nombre de destino, se conserva el original.
mget expresión	Se descargan varios ficheros al mismo tiempo siempre que sean coincidentes con una expresión usando comodines.
bye	Se desconecta la sesión que tenemos abierta en el servidor.
lcd	Se muestra el directorio actual de trabajo en el cliente, se puede usar para moverse por la estructura de directorios local si se usa con nombres de directorios.

Modo de acceso mediante cliente gráfico

La principal de las ventajas que el uso de aplicaciones gráficas supone, es la comodidad que nos ofrece con respecto a un entorno de terminal. En muchas ocasiones, cuando los usuarios no tienen muchos conocimientos informáticos, es una buena alternativa y la más recomendable. Es por esto, que es el modo de acceso más usado en las grandes empresas.

Tenemos tres métodos principales de conexión gráfica:

- **Mediante el explorador de archivos:** se especifica en la barra de directorios la cadena ftp://servidor_ftp. Esta manera es la que más se parece al acceso común a carpetas de los directorios locales.

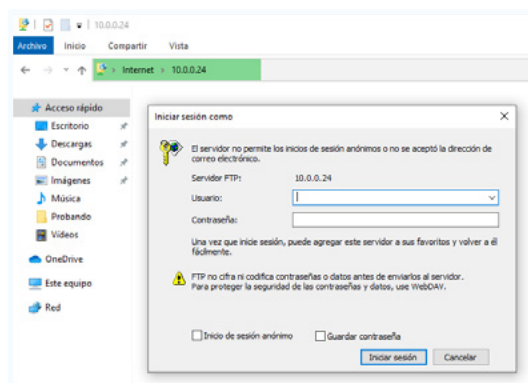


Imagen 5. Conexión a servidor FTP mediante el explorador de archivos de Windows

- **Mediante navegadores web:** en este caso, en un navegador, más concretamente en su barra de búsqueda, especificamos lo siguiente: ftp://usuario:contraseña@servidor_ftp. Teniendo en cuenta que:

- » **Usuario.** Es el nombre de usuario para acceso al servicio.
- » **Contraseña.** La contraseña asignada al usuario anterior.

Hay ocasiones en las que directamente desde el explorador de archivos se nos puede abrir el navegador, y otras en las que la conexión solo permite la descarga de ficheros.

- **Mediante software cliente de FTP:** este es el caso en el que usamos aplicaciones o nativas de los sistemas o de terceros, que están diseñadas para la compartición de recursos sobre FTP. En estas aplicaciones es común que se añadan funcionalidades o propiedades extra que nos ayuden a trabajar con el protocolo FTP. Dos ejemplos pueden ser *Filezilla* y *WinSCP*.

Cualquiera de las formas anteriores es totalmente válida, y todo dependerá de si se usa un sistema operativo u otro, del nivel de conocimiento de los usuarios, etc.



4.3.2. Usuario anónimo

El usuario anónimo o *anonymous* del servicio FTP es el que por defecto sirve para conectarte a un servidor FTP sin necesidad de especificar credenciales.

Normalmente, el usuario anónimo tendrá el acceso restringido y además, puede que haya ocasiones en las que no se pueda habilitar. En *vsFTPD* se puede habilitar, pero por defecto viene deshabilitado.

Para habilitarlo debemos de abrir el fichero */etc/vsftpd.conf* y añadir la siguiente directiva:

anonymous_enable=yes

Para que los usuarios se puedan conectar correctamente, necesitamos que exista un directorio creado para la gestión de los archivos.

Nosotros vamos a crear con el comando **mkdir** el directorio */var/ftp* y añadimos en el fichero la directiva siguiente:

anon_root=/var/ftp

```
#  
# Allow anonymous FTP? (Disabled by default).  
anonymous_enable=yes  
anon_root=/var/ftp  
#
```

Imagen 6. Habilitando las directivas correspondientes al usuario anónimo

Una vez tenemos los ficheros editados, reiniciamos el servicio con **systemctl**.

Ahora, probamos la conexión con el servidor y nos saldrá algo como la siguiente imagen:

```
root@servidor:/var# ftp 10.0.0.24  
Connected to 10.0.0.24.  
220 (vsFTPd 3.0.3)  
Name (10.0.0.24:profesor): anonymous  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> _
```

Imagen 7. Acceso a FTP con el usuario anónimo

Cabe destacar que por defecto el usuario anónimo solo tiene acceso de lectura en el directorio especificado y sus subdirectorios.



4.3.3. Usuarios locales de equipo

Para conectarnos al servidor FTP, *vsFTPD* nos da la opción de poder usar usuarios locales del servidor para iniciar sesión en el servicio. Esto se puede configurar de modo que cada usuario pueda trabajar únicamente en un directorio asignado y no tenga acceso a otros. Este proceso es conocido como *enjaular usuarios*.

Para usar usuarios del sistema, tendremos que activar la siguiente directiva en */etc/vsftpd.conf*:

chroot_local_user=yes

Al realizar dicha acción, los usuarios quedan enjaulados en el directorio que tengan predeterminado.

Si queremos dar libertad a los usuarios, es decir, que puedan navegar por todos los directorios, debemos de añadir dos directivas al fichero:

- > **Chroot_list_enable=yes** → Esta directiva indica que existen usuarios que no estarán enjaulados.
- > **Chroot_list_enable=/etc/vsftpd.chroot_list** → Esta directiva indica en que fichero se vana a reflejar que usuarios no están enjaulados. Por defecto el fichero es */etc/vsftpd.chroot_list*.

```
#  
chroot_local_user=YES  
#  
chroot_list_enable=YES  
# (default follows)  
#  
chroot_list_file=/etc/vsftpd.chroot_list  
#
```

Imagen 8. Directivas para activar usuarios locales no enjaulados

Ejemplo

Vamos a ver ahora como crearíamos un usuario local y le daríamos permiso para trabajar únicamente en su directorio. Los pasos que debemos seguir son los siguientes:

1. Lo primero es crear el usuario:

```
useradd -d directorio_usuario -m -s directorio_  
shell usuario
```

2. Ahora le asignamos una contraseña a este usuario:

```
passwd usuario
```

```
root@servidor:~# useradd -d /home/ftp_user -m -s /bin/bash ftp_user  
root@servidor:~# passwd ftp_user  
New password:  
Retype new password:  
passwd: password updated successfully  
root@servidor:~# _
```

Imagen 9. Creamos el usuario local



- Una vez que tenemos el usuario, debemos comentar en el fichero `/etc/vsftpd.conf` las líneas:

```
chroot_list_enable=yes
chroot_list_enable=/etc/vsftpd.chroot_list
```

Esto lo hacemos para que el usuario sí que esté enjaulado.

- Reiniciamos el servicio.
- Comprobamos que el servicio funciona conectándonos al servicio con el comando:

```
ftp dirección_servidor
```

```
root@servidor:~# ftp 10.0.0.24
Connected to 10.0.0.24.
220 (vsFTPd 3.0.3)
Name (10.0.0.24:profesor): ftp_user
331 Please specify the password.
Password:
500 OOPS: vsftpd: refusing to run with writable root inside chroot()
Login failed.
ftp> quit
421 Service not available, remote server has closed connection
root@servidor:~# _
```

Imagen 10. Error de conexión

Vemos que nos ha dado un error, vamos a ver cómo solucionarlo, porque tenemos dos formas:

- » La primera de estas formas es añadiendo al fichero de configuración, la directiva:

```
allow_writeable_chroot=yes
```

Con esta directiva damos acceso al usuario para que pueda escribir en su carpeta.

```
#
chroot_local_user=YES
allow_writeable_chroot=yes_
#
```

Imagen 11. Habilitamos la directiva de escritura

- » La segunda de las formas es cambiar el permiso del directorio `root` del usuario para que solo tenga permiso de escritura, que sería con el comando:

```
chmod a-w directorio_usuario
```

Ahora, hayamos hecho cualquiera de las dos formas, debemos de reiniciar de nuevo el servicio.

- Volvemos a comprobar el servicio y vemos que ahora funciona correctamente:

```
root@servidor:~# ftp 10.0.0.24
Connected to 10.0.0.24.
220 (vsFTPd 3.0.3)
Name (10.0.0.24:profesor): ftp_user
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> _
```

Imagen 12. Comprobamos que podemos acceder correctamente con un usuario local

4.4.

Modos de conexión del cliente

A la hora de solicitar acceso al servicio FTP, usamos un puerto para el envío de los comandos de control, y otro distinto para el envío de los datos. La principal diferencia entre los modos de conexión se basa en los puertos que usamos para las conexiones de datos. Mientras que en uno se usa el puerto 20, en otro se usa otro distinto.

4.4.1. Modo activo

El modo activo era por defecto el modo que estaba presente en el servicio en las primeras versiones de este servicio. Este modo se basa en que cuando un cliente se conecta a un servidor, este es el que indica el puerto que se va a usar para la conexión de datos, así, el cliente recibirá las peticiones siempre en el puerto 20, pero el cliente en el puerto que previamente se haya especificado.

En la siguiente imagen podemos ver la secuencia que sucede cuando se usa una conexión activa entre el cliente y el servidor:

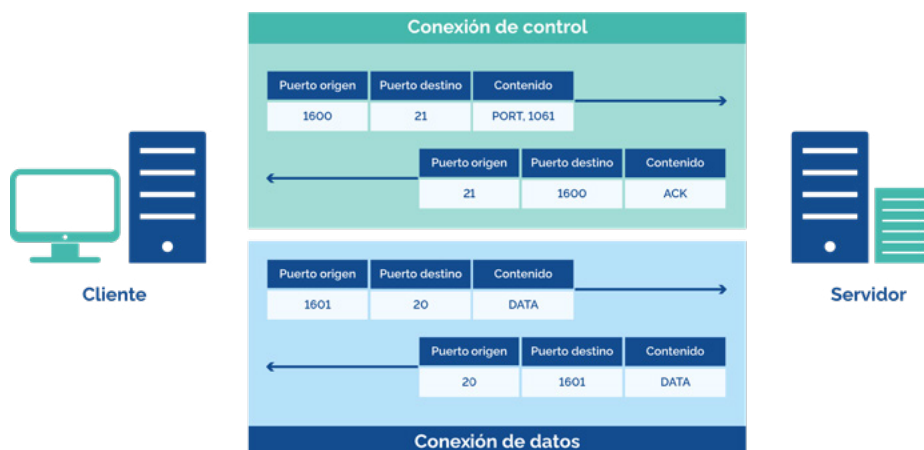


Imagen 13. Modo de conexión activo

Cuando se da por iniciada una conexión activa, el cliente envía al servidor los paquetes con destino al puerto 21 y origen de puerto aleatorio, pero siempre mayor que 1024. Dentro de dicho paquete se envía el comando **PORT** que va acompañado del número de puerto usado por el cliente para la transferencia de los datos. Suele ser un número más que el puerto de origen. Si el servidor, al recibirlo está de acuerdo. Se envía un *acknowledgment* (ACK) al cliente. Una vez conocemos los puertos implicados en la comunicación, da comienzo la transmisión.

Nuestro servidor se usa por defecto también modo activo, por lo que, si queremos deshabilitar este modo, deberíamos de activar el modo pasivo con el comando **passive**.

Para volver a activar el modo pasivo, debemos de lanzar de nuevo el mismo comando.

```
ftp> passive
Passive mode on.
ftp> passive
Passive mode off.
ftp> _
```

Imagen 14. Activación y desactivación del modo pasivo

4.4.2. Modo pasivo

En este modo, el pasivo, el servidor es el que indica que puerto debe ser en el que iniciamos la conexión de datos, por lo que no siempre se usa el puerto por defecto.

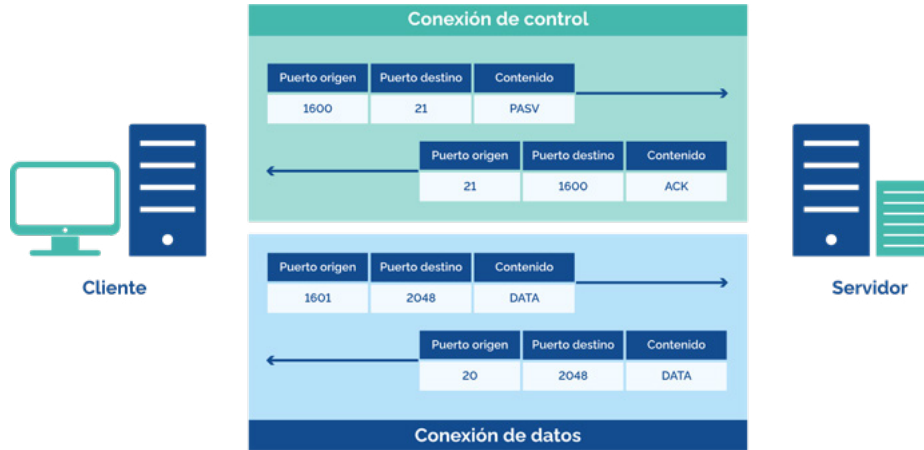


Imagen 15. Modo de conexión pasivo

Como vemos en la anterior imagen, cuando la comunicación entre cliente y servidor se inicia, se envía el comando **PASV** hasta el puerto de destino 21. EL servidor indica a modo de respuesta al cliente, cual es el puerto que se va a usar en la transmisión de los datos, siempre mayor que 1024. Una vez el cliente recibe las indicaciones, se da inicio a la transmisión desde un puerto distinto al de la conexión de control.

Para habilitar el modo pasivo, ya hemos visto que debemos de usar el comando **passive** una vez dentro del servidor.





 www.universae.com

