UD 4. INSTALACIÓN Y ADMINISTRACIÓN DE SERVIDORES FTP

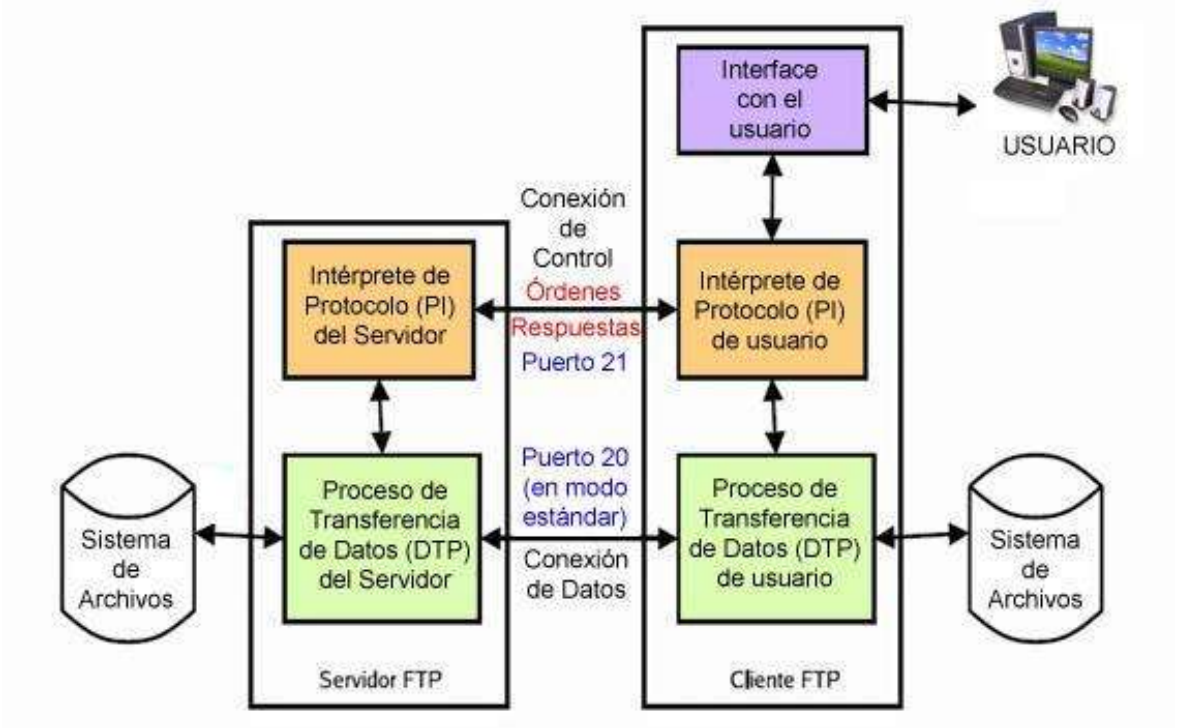
El servicio FTP o **protocolo de transferencia de archivos** (FILE TRANSFER PROTOCOL) es un protocolo de la capa de aplicación que proporciona un mecanismo estándar de transferencia de archivos entre sistemas a través de redes TCP/IP. Pero el protocolo FTP va más allá, permite administrar ficheros permitiendo acciones como borrarlos, renombrarlos, crear carpetas, borrarlas…

Con el estado actual de Internet y las múltiples opciones de trasferencia de archivos en la web puede parecer algo innecesario pero sigue siendo una opción sencilla y específica por lo que en ámbitos profesionales continua gozando de buenísima salud. Por ejemplo sigue siendo el método más habitual para subir archivos, actualizaciones o modificaciones de contenido a un servidor web, especialmente en el modo de hosting.

1. EL SERVICIO FTP

FTP está basado en la arquitectura CLIENTE-SERVIDOR es decir los archivos se almacenan en un SERVIDOR, que ejecuta el servicio FTP. Los equipos remotos se pueden conectar utilizando un CLIENTE FTP y principalmente LEER archivos del servidor o COPIAR archivos al servidor. Esta conexión es independiente del sistema operativo y del sistema de archivos utilizado (Windows: FAT32, NTFSA y Linux: Ext3…etc)

El funcionamiento por defecto utiliza el puerto 20 para datos y el 21 de control.



El intérprete de protocolo (PI) de CLIENTE inicia la conexión de control en el puerto 21.
Como mínimo esta conexión requiere el nombre de usuario, la contraseña, el servidor al que se conecta y el puerto.

Una vez establecida la conexión, el cliente transmite una serie de órdenes por el puerto de control desde su ordenador hasta el servidor. Estas órdenes FTP especifican parámetros como:

- Puerto de datos
- Modo de transferencia
- La naturaleza de la operación sobre el sistema de archivos (almacenar, recuperar, añadir, borrar, etc.).

El proceso de transferencia de datos (DTP), el servidor contesta a estas órdenes permitiendo el envío o la recepción de datos por el puerto indicado (20 en modo activo) en función de los parámetros que se hayan especificado.

2. TIPOS DE USUARIOS Y ACCESOS AL SERVICIO.

En FTP existen dos tipos básicos de usuario, los corrientes y los anónimos. Realmente definen el tipo de acceso porque indican si te estás autenticando con un usuario concreto o estás utilizando una cuenta anónima que generalmente no requiere autenticación. Un servidor FTP puede servir ambos tipos simultáneamente.

- **FTP anónimo**: Este modo se utiliza generalmente cuando el servidor FTP se usa para distribuir cualquier tipo de archivo o archivos a un número muy elevado de usuarios en una situación en la que la identificación no es muy importante. Si por ejemplo hemos realizado una aplicación de software libre y queremos distribuirla es una buena opción. En este tipo de conexión solo se le pide al cliente un nombre de usuario anónimo (generalmente y por defecto es **anonymous**) y si acaso (no siempre) una contraseña. Una vez nos hemos conectado al servidor tendremos acceso al directorio anónimo y sus subdirectorios.

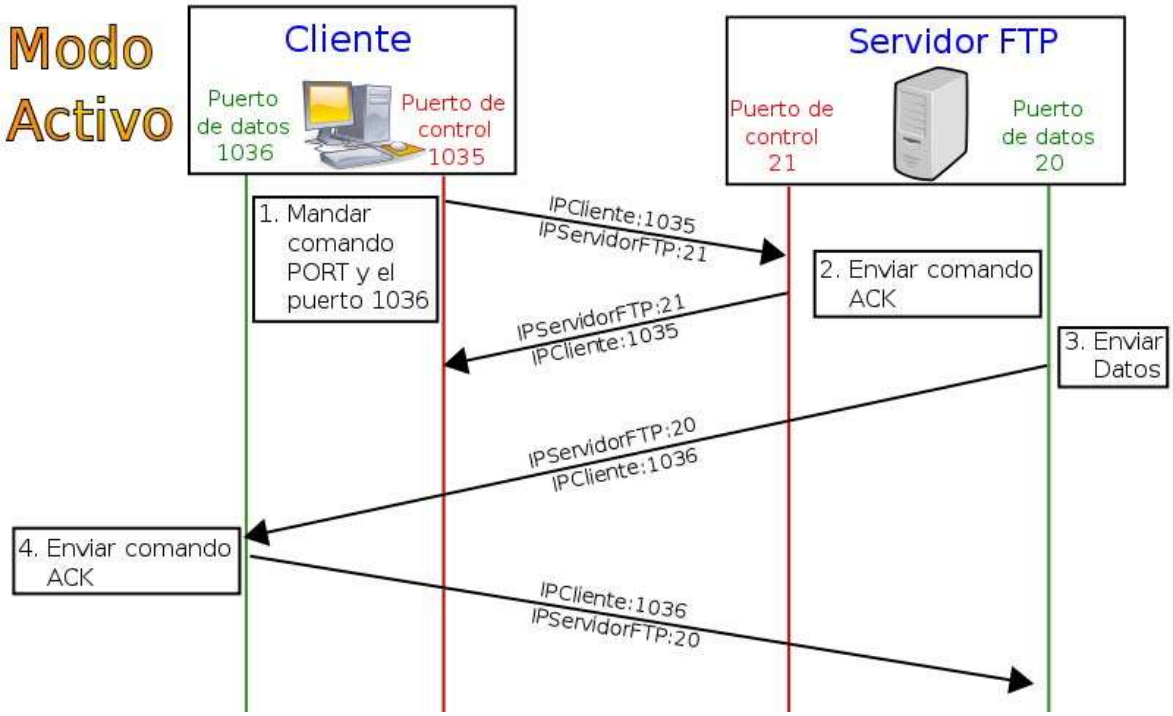
- **FTP corriente**: En este caso los usuarios de FTP **son los que existen en la máquina en la que instalamos el servidor. Estos usuarios podrán leer de y copiar a su directorio personal archivos remotamente**. Las mismas credenciales que tienen en la máquina serán las que necesiten para conectarse mediante FTP. Existen también los denominados **usuarios virtuales** que tendrán credenciales FTP pero no cuenta en el servidor Linux.

3. - MODOS DE CONEXIÓN DEL CLIENTE.

FTP soporta dos modos de conexión del cliente

a. El modo activo.
b. El modo pasivo.

Modo activo



1.- El cliente se conecta desde un puerto de control aleatorio superior al 1024 (en nuestro caso el 1035) al puerto de control del servidor, puerto 21. El cliente empieza a escuchar por el puerto 1036 y envía este puerto de control al servidor.

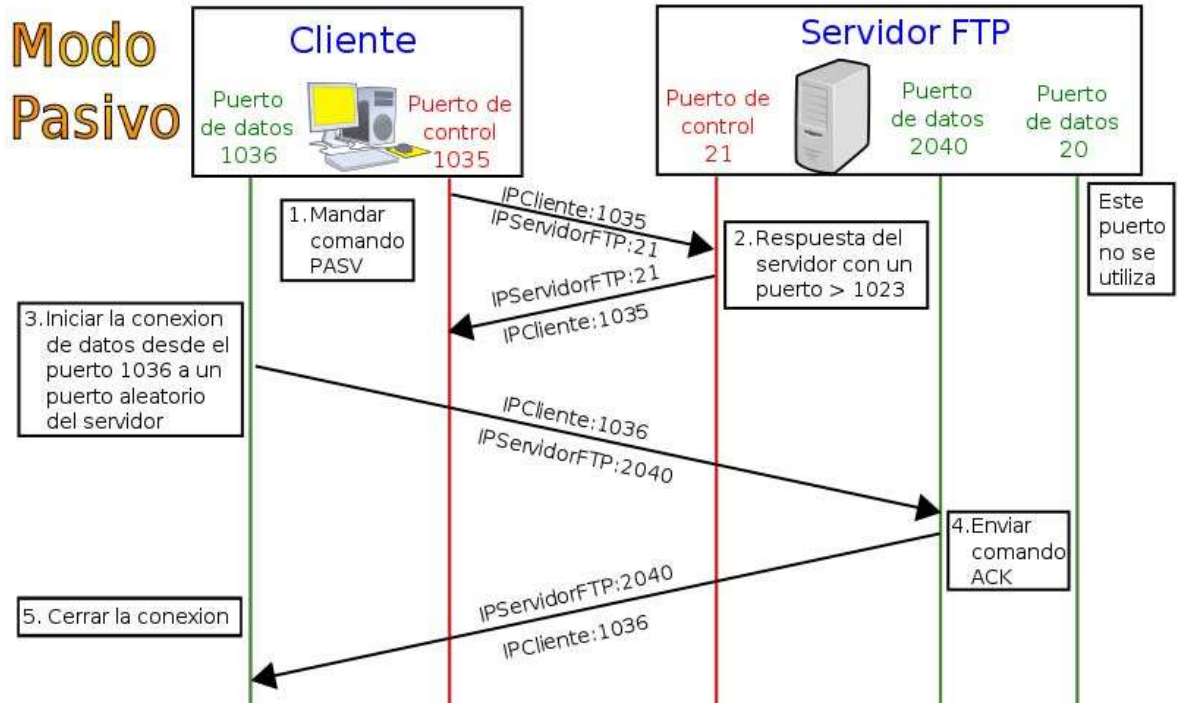
2.- El servidor responde con un ACK al puerto de control del cliente.

3.- El servidor inicia una conexión entre su puerto de datos (puerto 20) y el puerto de datos del cliente (puerto 1036)

4.- El cliente responde con un ACK al servidor.

Lo anterior tiene un grave problema de seguridad, y es que la máquina cliente debe estar dispuesta a aceptar cualquier conexión de entrada en un puerto superior al 1024, con los problemas que ello implica si tenemos el equipo conectado a una red insegura como Internet. De hecho, los cortafuegos que se instalen en el equipo para evitar ataques seguramente rechazarán esas conexiones aleatorias. SOLUCIÓN: modo pasivo.

Modo pasivo



1.- El cliente se conecta desde un puerto de control aleatorio superior al 1024 (en nuestro caso el 1035) al puerto de control del servidor, puerto 21. El cliente envía un comando PASV al servidor.

2.- El servidor responde con un ACK, desde un puerto aleatorio superior al 1024 (en nuestro caso el 2040) al puerto de control del cliente.

3.- El cliente inicia una conexión entre su puerto de datos del cliente (puerto 1036) y el puerto de datos del servidor (puerto 2040)

4.- El servidor responde con un ACK al cliente.

Modo Activo vs Modo Pasivo

En la descarga de datos, en el modo activo se abre una conexión para datos desde el servidor FTP al cliente FTP, esto es, una conexión de fuera hacia adentro, entonces, si el cliente FTP se encuentra detrás de un firewall, este filtrará o bloqueará la conexión entrante.

En el modo pasivo es el cliente FTP el que inicia tanto la conexión de control como la de datos, con lo cual el firewall del cliente no tendrá ninguna conexión entrante que filtrar.

4.- Tipos de transferencia: binaria vs ascii

En el protocolo FTP existen 2 tipos de transferencia

a. ASCII
b. binarios

Es importante conocer cómo debemos transportar un archivo a lo largo de la red para no destruirlo durante la transmisión con distintos tipos de sistemas de archivos. Al ejecutar la aplicación FTP, debemos utilizar uno de estos comandos (o poner la correspondiente opción en un programa con interfaz gráfica).

- Tipo **Binario** (por defecto bit a bit)

Archivos comprimidos, ejecutables para PC (.exe), imágenes, video, audio, imágenes de sistema operativo (.iso), archivos .doc, .xls…

- Tipo **ASCII** (byte a byte)

Texto puro como por ejemplo .html, .php, .asp, .xml, .txt…

5. CLASIFICACIÓN DE LOS CLIENTES FTP

Los clientes FTP pueden ser de distinta naturaleza

- Integrados en el navegador (Internet Explorer, Firefox, Opera…)
- Cliente FTP en modo gráfico (Los hay de pago, freeware…)
- Cliente FTP en modo consola que suelen ir integrados en el sistema operativo
- Otros (Clientes FTP integrados en páginas web programados con PHP, ASP por ejemplo en servidores de hospedaje o hosting).

6. Utilización del servicio de transferencia de archivos desde el navegador.

Cuando el servidor FTP permite el acceso vía web, el navegador puede utilizarse como cliente. Basta con teclear la URL del servidor FTP correspondiente y, de esa forma, acceder a los archivos para los que se tenga los permisos adecuados.

SINTAXIS
protocolo://[usuario:pass@]servidorFTP[:puerto]/carpeta
EJEMPLO
ftp://jose:783638@ftp.informatik.tu-muenchen.de:1036/kino
ftp://ftp.rediris.es:21
ftp://ftp.rediris.es

7. Utilización de herramientas gráficas.

Los clientes FTP gráficos constituyen el mecanismo más sencillo para subir y bajar archivos de un servidor FTP. En los entornos Windows y GNU/Linux existe una gran variedad.

Algunos de los clientes FTP más conocidos son:

- gFTP (Linux)(Aplicaciones-Internet)
- Filezilla client: funciona en Windows, Linux y Mac OS X
- CuteFTP
- Core FTP Lite (Windows)
- AceFTP

8. Cliente FTP en modo consola

Por defecto vienen instalados en el sistema operativo. También al instalar un servidor FTP, el propio paquete suele llevar incluido un cliente que funciona en modo consola. El cliente FTP permite ejecutar de manera directa un conjunto de órdenes FTP. Para establecer una conexión FTP en modo línea de orden sigue la siguiente sintaxis:

$ ftp [nombre_maquina| IP]
ó
$ ftp
ftp> open [nombre_maquina| IP]

Comandos más importantes a nivel de cliente FTP

El comando ftp es un comando que actúa como cliente y está incluido en la mayoría de los sistemas operativos como Windows, Linux o Mac y nos permite hacer la transferencia de ficheros entre un cliente y un servidor.
Su uso es el siguiente:
ftp servidor
Donde servidor es la dirección IP o nombre del servidor al que queremos subir o del que queremos descargar ficheros.
Otra forma de realizar la conexión es la siguiente:
ftp ftp> open servidor
Una vez establecida la conexión podemos hacer utilizar diferentes comandos:
? o help: Nos da un listado de los diferentes comandos que podemos ejecutar desde el cliente.
help : Nos da ayuda de cada comando.
Los siguientes comandos nos permiten navegar por el cliente y servidor, creando su estructura de ficheros y directorios:

cd : Nos permite acceder a una carpeta del servidor. El valor de la carpeta puede ser absoluto o relativo.
lcd : Nos permite acceder a una carpeta del cliente. El valor de la carpeta puede ser absoluto o relativo.
pwd: Nos visualiza el directorio en el que estamos ubicados dentro del servidor.
!pwd: Nos visualiza el directorio en el que estamos ubicados dentro del cliente.
ls o dir: Nos lista los ficheros y directorios del servidor.
!ls o !dir: Nos lista los ficheros y directorios del cliente.
delete: Borra un fichero del servidor.
!delete: Borra un fichero del cliente.
rename: Renombra un fichero del servidor.
!rename: Renombra un fichero del cliente.
rmdir o rm: Borra un directorio del servidor.
!rmdir o !rm: Borra un directorio del cliente.
mkdir: Crea un directorio en el servidor.
!mkdir: Crea un directorio en el cliente.
Para el tipo de transferencia tenemos los siguientes comandos:
asc o ascii: Establece la transmisión de tipo ascii.
bin o binary: Establece la transmisión de tipo binario.
type: Indica el tipo de transferencia.
Ya sólo nos queda subir o descargar los ficheros. Para ello tenemos los siguientes comandos:
put: Transfiere un fichero local al servidor remoto.
mput: Transfiere uno o varios ficheros locales al servidor remoto.
get: Transfiere un fichero del servidor remoto al ordenador local.
mget: Transfiere uno o varios ficheros del servidor remoto al ordenador local.
Los comandos put y get admiten uno o dos parámetros, el segundo parámetro nos permite renombrar el archivo una vez subido o descargado.
Los comandos mput y mget admiten expresiones regulares (? para indicar un carácter y * para indicar varios caracteres). Cada vez que subimos o bajamos un archivo nos va a pedir una confirmación (a la que tendremos que responder y o yes). Si no queremos que nos pida dicha confirmación podremos utilizar el comando prompt.
prompt: Nos permite habilitar o deshabilitar el modo interactivo.
Para desconectarnos y/o cerrar la conexión tenemos los siguientes comandos:
close: Termina la sesión ftp pero no sale del programa.
bye o quit: Termina la sesión ftp y sale del programa.

9. INSTALACIÓN DEL SERVICIO DE TRANSFERENCIA DE ARCHIVOS.

En Linux existen muchos servidores FTP diferentes. Los dos más populares actualmente son ProFTPd y vsFTPd. El primero es más sencillo de utilizar y sus archivos de configuración y estructura similar hacen que se parezca mucho a Apache. Sin embargo el segundo es el servidor FTP por defecto en las principales distribuciones de Linux lo que hace que sea más sencillo de instalar y además se considera más seguro.
Para hacer funcionar el servidor instalaremos el paquete VSFTPD.

Las siguientes indicaciones están referidas a una instalación en DEBIAN.

Podemos hacerlo realizando directamente la instalación de vsFTPd (servidor ftp) con el comando: `apt install vsftpd`

Y del FTP cliente con el comando: `apt install ftp`

Una vez instalado podemos ver que se ha añadido un script de autoarranque en
/etc/init.d/vsftpd

Para arrancar, parar, reiniciar, o ver su estado se utiliza el comando
`service vsftpd {start | stop | restart | status}`

Ya podemos conectarnos al servidor ftp mediante un cliente en modo consola
`ftp localhost`

Lo que nos pedirá unas credenciales para acceder. Por defecto el usuario anonymous no está habilitado por lo que nos conectamos como el usuario de nuestra máquina y su contraseña. Observa que usamos `quit` para desconectarnos.

Características principales del servidor vsftpd

Al instalar el servidor se crean los siguientes archivos y directorios en el sistema:

/etc/init.d/vsftpd	Fichero que inicia el servidor ftp
/usr/bin/vsftpd	Archivo ejecutable
/etc/vsftpd.conf	Archivo de configuración del servidor
/srv/ftp/	Directorio predeterminado de los usuarios anónimos. Su propietario es root y su grupo ftp.
/etc/ftpusers	Fichero que contiene la lista de usuarios que no se podrán conectar al servidor.

Comprobamos que al instalarlo se ha creado el usuario ftp y que su directorio home es /srv/ftp
cat /etc/passwd (muestra los usuarios del sistema)
cat /etc/group (muestra los grupos)
Comprobamos que se ha creado el directorio /srv/ftp y que su propietario es el usuario root y su grupo es ftp.
ls –la /srv

10. CONFIGURACIÓN DEL SERVICIO DE TRANSFERENCIA DE ARCHIVOS.

El fichero de configuración del servidor se llama vsftpd.conf y se encuentra en el directorio /etc.
Aspectos importantes:

a) El archivo contiene un conjunto de directivas que determinan el comportamiento del servidor. Cada directiva tiene el formato < directiva >=< valor>, como es habitual en este tipo de archivos de configuración.
b) **IMPORTANTE: No debe haber espacios antes y después del signo "=".**
c) Tres tipos de directivas en función de lo que vale su campo valor
   - Booleanas, en cuyo caso el campo valor puede contener YES o NO.
   - Numéricas.
   - De cadena.
d) Los comentarios son a nivel de línea y se utiliza el símbolo almohadilla (#).
e) Las directivas que no se especifiquen en el fichero de configuración, utilizan su valor por defecto.

Para consultar las directivas: man vsftpd.conf

Información de Debian:
http://www.zeppelinux.es/instalacion-y-configuracion-del-servidor-ftp-vsftpd-en-linux-debian/

Las directivas más importantes de este fichero son las siguientes:

Directivas generales

listen

El servidor FTP se puede ejecutar en 2 modos:

Modo Standalone: Como un servicio del sistema, controlado desde el arranque. Responde más rápido a las peticiones.

Modo inetd: que se inicie automáticamente en el momento en que se recibe una petición, libera más recursos en caso de no usarse el servicio

Para establecer el servicio en modo Standalone editaremos el archivo de configuración y nos aseguramos de que contiene la línea:
listen=YES

ftpd_banner

Con esta directiva se puede mostrar un mensaje de bienvenida cuando un usuario se conecte al servidor FTP.
ftpd_banner=Welcome to FTP service.

dirmessage_enable

Permite mostrar mensajes a los usuarios al acceder por primera vez a ciertos directorios. Este mensaje se encuentra dentro del directorio al que se entra. El nombre de este archivo se especifica en la directriz **message_file** y por defecto es .message.
dirmessage_enable=YES

max_clients

Indica el número máximo de clientes que podrán conectarse simultáneamente al servidor. En el siguiente ejemplo son 5.

```
max_clients=5
```

max_per_ip

Este parámetro establece el número máximo de conexiones que se pueden realizar desde una misma dirección IP. Tome en cuenta que algunas redes acceden a través de un servidor proxy o puerta de enlace haciendo NAT y debido a esto podrían quedar bloqueados innecesariamente algunos accesos.
En el siguiente ejemplo se limita el número de conexiones por IP simultáneas a 5.

```
max_per_ip=5
```

data_connection_timeout

Para indicar el tiempo máximo que el servidor espera cuando una transferencia no progresa, se utiliza la directiva:

```
data_connection_timeout=300
```

idle_session_timeout

Para indicar el tiempo máximo que el servidor espera a desconectarse entre órdenes (en reposo):

```
idle_session_timeout=300
```

write_enable
Con esta directiva se puede permitir o denegar que los usuarios locales puedan subir archivos al servidor.
```
write_enable=YES
```

connect_from_port_20
Cuando se activa, vsftpd se ejecuta con privilegios suficientes para abrir el puerto 20 (ftp-data) en el servidor durante las transferencias de datos en modo activo. Al desactivar esta opción, se permite que vsftpd se ejecute con menos privilegios, pero puede ser incompatible con algunos clientes FTP.
```
connect_from_port_20=YES
```

listen_port
Especifica el puerto por el que escucha el servidor. Por defecto el 21

pasv_enable
Cuando está habilitada esta opción se permiten conexiones en modo pasivo.

pasv_max_port y pasv_min_port
Especifica el puerto más alto y más bajo en conexiones pasivas. Esta configuración es útil para la creación de reglas en el firewall. Entre 1024-65535

pasv_address=192.168.1.27
La siguiente directriz especifica la dirección IP del lado público del servidor (IP pública de nuestro encaminador) para los servidores que se encuentran detrás de cortafuegos Network Address Translation (NAT). Esto permite que vsftpd entregue la dirección correcta de retorno para las conexiones pasivas.

11. Usuarios anónimos

anonymous_enable

El acceso anónimo permitirá a cualquier persona que conozca nuestra dirección IP conectarse al servicio y navegar por el directorio por defecto del usuario anónimo que es /srv/ftp. Este modo es idóneo para compartir archivos que no necesitan una especial protección.

Si el valor de esta directiva es YES cualquier usuario se puede conectar al servidor dando el nombre de usuario anonymous o ftp.

```
anonymous_enable=YES
```

anon_upload_enable

Esta directiva indica si los usuarios anónimos pueden cargar archivos en el servidor. Cuando se usa con la directriz write_enable, los usuarios anónimos pueden cargar archivos al directorio padre que tiene permisos de escritura.

```
anon_upload_enable=YES
```

anon_mkdir_write_enable

Si esta directiva tiene el valor YES en combinación con la directriz write_enable, los usuarios anónimos pueden crear nuevos directorios dentro de un directorio que tiene permisos de escritura.

```
anon_mkdir_write_enable=YES
```

anon_max_rate

Se utiliza para limitar la tasa de transferencia a usuarios anónimos. En el siguiente ejemplo se limita la tasa de transferencia a los usuarios anónimos a 10Kb/s.

```
anon_max_rate=1024
```

anon_root

Se utiliza para determinar la ruta por defecto del usuario anonymous.

```
anon_root=/home/ftp-pub/
```

RECORDATORIO SOBRE PERMISOS LINUX

- Los permisos están divididos en tres tipos: lectura, escritura y ejecución. Estos permisos pueden ser fijados para tres clases de usuarios: el propietario del archivo o directorio, los integrantes del grupo al que pertenece y todos los demás usuarios.
- El permiso de lectura permite a un usuario leer el contenido del archivo o en el caso de un directorio, listar el contenido del mismo (usando ls).
- El permiso de escritura permite a un usuario escribir y modificar el archivo (inclusive, eliminarlo). Para directorios, el permiso de escritura permite crear nuevos archivos o borrar archivos ya existentes en el mismo.
- Por último, el permiso de ejecución permite a un usuario ejecutar el archivo si es un programa. Para directorios, el permiso de ejecución permite al usuario ingresar al mismo (por ejemplo, con el comando cd).

Así para poder acceder a un archivo, debe de tener permiso de ejecución de todos los directorios a lo largo del camino de acceso al archivo, además de permiso de lectura del archivo en particular.

El comando chmod se usa para establecer los permisos de un archivo. Sólo el propietario puede cambiar los permisos del archivo (además, claro está, del administrador del sistema, el usuario root). La sintaxis de chmod es:

chmod [a,u,g,o][+,-][r,w,x] <archivos>

EJERCICIO RESUELTO: Hacer que el usuario ftp pueda escribir en el directorio /srv/ftp/directorio_subir_archivos

1) Creamos el directorio donde podremos subir archivos como anónimo

```
sudo mkdir /srv/ftp/directorio_subir_archivos
```

2) El propietario será root, tenemos que hacer que el usuario ftp pueda escribir en él, así que cambiamos de propietario el “directorio_subir_archivos” y decimos que sea el usuario “ftp” su dueño.

```
sudo chown –R ftp  /srv/ftp/directorio_subir_archivos
```

3) Modificamos el archivo de configuración de vsftpd

```
anonymous_enable=YES
write_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
```

4) Reiniciamos el servidor y nos volvemos a conectar con el cliente ftp. Comprobamos que al raíz no nos deja subir archivos, ni crear directorios, pero en “directorio_subir_archivos” sí que podemos.

12. Usuarios locales

Los usuarios locales son los que tienen cuenta en el sistema (/etc/passwd). A continuación se detallan los parámetros de configuración relacionados con usuarios locales.

local_enable

El acceso privado permite establecer nombres de usuario y contraseña para acceder al servicio, cada usuario podrá acceder tan solo a sus propios archivos en /home. Este es un sistema idóneo para organizar la información por usuarios y otorgar un cierto grado de seguridad a los archivos.

Esta línea indica que si se permite o no el **acceso de usuarios locales a sus respectivas carpetas privadas**. Si se permite el acceso habría que poner

```
local_enable=YES
```

Si no queremos que se permita el acceso, habrá que darle el valor NO o comentar la línea.

chroot_local_user

Permite enjaular a los usuarios dentro de su propio directorio personal. Si en el fichero de configuración aparece chroot_local_user=NO, entonces el usuario tiene acceso a todo el sistema de archivos, en función de los permisos asignados. Cuando un usuario local se conecta y en el fichero de configuración aparece chroot_local_user=YES, entonces enjaulamos a los usuarios dentro de su propio directorio personal, sin posibilidad de acceder a todo el sistema de ficheros. Se mejora por tanto la seguridad.

chroot_list_enable y chroot_list_file

Cuando está activada la primera, se coloca en una prisión de chroot a los usuarios locales listados en el archivo especificado en la directriz chroot_list_file cuyo valor por defecto es /etc/vsftpd.chroot_list.

Este fichero no existe por defecto hay que generarlo.

Ten en cuenta los siguientes modos de ejecución:

```
# 1. Todos los usuarios locales enjaulados:
chroot_local_user=YES
chroot_list_enable=NO

# 2. Algunos usuarios enjaulados:
chroot_local_user=NO
chroot_list_enable=YES
# Create the file /etc/vsftpd.chroot_list with a list of the jailed users.

# 3. Algunos usuarios libres
chroot_local_user=YES
chroot_list_enable=YES
# Create the file /etc/vsftpd.chroot_list with a list of the "free" users.
```

userlist_deny, userlist_enable y userlist_file
Con estos commandos se puede configurar qué usuarios pueden acceder al ftp. Por defecto el fichero /etc/ftpusers contiene una lista de usuarios que no pueden realizar un ftp autentificado, pero con estos comandos se puede ampliar la funcionalidad.
Se pueden tener dos configuraciones diferentes:

a) Denegar a ciertos usuarios locales poder conectarse

```
userlist_enable=YES
userlist_deny=YES
userlist_file=/etc/vsftpd.denied_users
```

Añade la lista de usuarios denegados en el fichero vsftpd.denied_users un usuario por línea.

Recuerda que además en /etc/ftpusers también hay algunos usuarios que no se les permite conectarse.

b) Permitir conectarse sólo a algunos usuarios

```
userlist_enable=YES
userlist_deny=NO
userlist_file=/etc/vsftpd.allowed_users
```

Añade la lista de usuarios en el fichero vsftpd.allowed_users un usuario por línea

local_max_rate
Se utiliza para limitar la tasa de transferencia en bytes por segundo a los usuarios locales del servidor. En el siguiente ejemplo se limita la tasa de transferencia a 10Kb/s: `local_max_rate=1024`

local_umask

Se pueden establecer los permisos con los que quedará el archivo/directorio al subirlo al servidor FTP. La máscara 022 quiere decir que los **directorios** que serán creados tendrán permisos 755 (propietario -> lectura, escritura, ejecución. grupo -> lectura y ejecución. otros -> lectura y ejecución). Y los **archivos** creados tendrán permisos 644 (propietario -> lecturay escritura. grupo -> lectura. otros -> lectura).

Cómo se calcula la **umask**? Para saber la máscara por defecto, se debería hacer la siguiente operación: (permisos base XOR permisos deseados = máscara).
Es decir si queremos que los usuarios locales suban carpetas con los permisos 640 (lectura y escritura para el usuario, lectura para grupo y ningún permiso para el resto) debemos calcular la máscara inversa. Para ello hacemos un XOR a **777 (con los directorios),** es decir 777 XOR 640 → 137 con los archivos subidos por los usuarios locales serán 640.
Ten en cuenta que todos los valores con los que trabajamos son en formato octal.

111 111 111 → 777 (permisos bases de directorios)
110 100 000 → 640 (permisos deseados

001 011 111 (137 sería el valor de la máscara por defecto que utilizaría en la directiva **local_umask**)

Con los ficheros la operación XOR hay que hacerla con 666 en vez de con 777 (directiva file_open_mode por defecto). Si se quiere hacer con 777 hay que modificar la siguiente directiva.

Realmente estamos haciendo una operación XOR binaria no una resta decimal.

```
local_umask=011
file_open_mode=0777
```

Web donde se explica umask
https://mviera.io/blog/entendiendo-umask/

Ayuda: Si el sistema nos da el siguiente error:
500 OOPS: vsftpd: refusing to run with writable root inside chroot()
Este error se da porque vsftpd no permite que los usuarios puedan escribir en su carpeta raíz (esto se soluciona con la versión vsftpd 3 con la directiva "allow_writeable_chroot=YES"). Si estamos con la versión 2.3, una solución es quitarle los permisos de escritura a esa carpeta. Esto hará que el usuario no pueda hacer nada en su propia carpeta, así que lo mejor es crearle otra dentro con permisos normales para que pueda utilizarla.

13. Personalizar usuarios autenticados

Es muy útil realizar una configuración de acceso diferente para cada usuario. Por ejemplo puede interesar que un usuario pueda subir archivos y otros no, o bien direccionar algún usuario hacia un directorio diferente de su directorio $HOME, como por ejemplo un directorio donde subir el sitio web de un usuario, que posteriormente pueda ser consultado a través de apache.

En el fichero vsftpd.conf agregar la directiva:
user_config_dir=/etc/vsftpd/users

Donde /etc/vsftpd/users tiene que ser un directorio que contiene ficheros con el mismo nombre de los usuarios. Crear tantos ficheros como usuarios autentificados,

Ejemplo a miguel le vamos a permitir ver y descargar lo hay en /usuarios.

```
Creamos el archivo /etc/vsftpd/users/miguel; con el siguiente contenido

dirlist_enable=YES
download_enable=YES
local_root=/usuarios
write_enable=NO
```

Otro ejemplo a pepe le vamos a permitir escribir en /var/www/html/web

```
Creamos un nuevo archivo /etc/vsftpd/users/pepe; con el siguiente contenido

dirlist_enable=YES
download_enable=YES
local_root=/var/www/html/web
write_enable=YES
```

14. Logs

Para registrar las descargas y las subidas

```
xferlog_enable=YES
xferlog_file=/var/log/vsftpd.log
xferlog_std_format=YES
```

FORMATO XFERLOG

```
Sun Jan 11 19:27:43 2018 1 127.0.0.1 0 /ftptere/hola.htm b _ o r miguel ftp 0 *
c
```

current-time transfer-time remote-host file-size filename transfer-type special-action-flag direction access-mode username service-name authentication-method authenticated-user-id completion-statu

http://www.castaglia.org/proftpd/doc/xferlog.html