UD 4. INSTALACIÓN Y ADMINISTRACIÓN DE SERVIDORES FTP

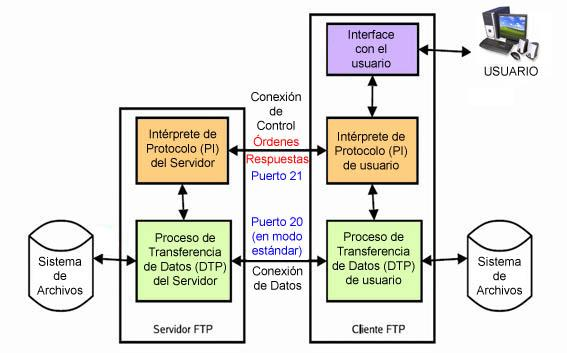
El servicio FTP o **protocolo de transferencia de archivos** (FILE TRANSFER PROTOCOL) es un protocolo de la capa de aplicación que proporciona un mecanismo estándar de transferencia de archivos entre sistemas a través de redes TCP/IP. Pero el protocolo FTP va más allá, permite administrar ficheros permitiendo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

1. **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.

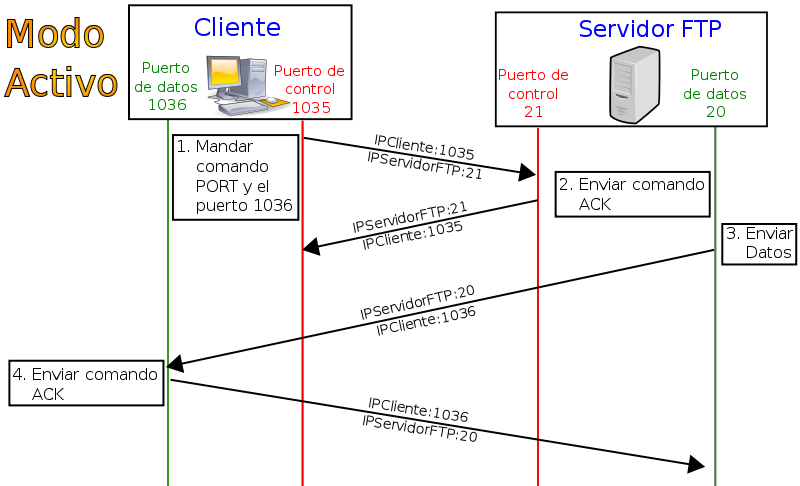
* **FTP anónimo**: Este modo se utiliza generalmente cuando el servidor FTP se usa para distribuir cualquier tipo de archivo o archivos a un número muy elevado de usuarios en una situación en la que la identificación no es muy importante. Si por ejemplo hemos realizado una aplicación de software libre 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 fuera el caso (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 están en el servidor Linux.

1. **- Modos de conexión del cliente.**

FTP soporta dos modos de conexión del cliente

* 1. El modo activo.
  2. El modo pasivo.

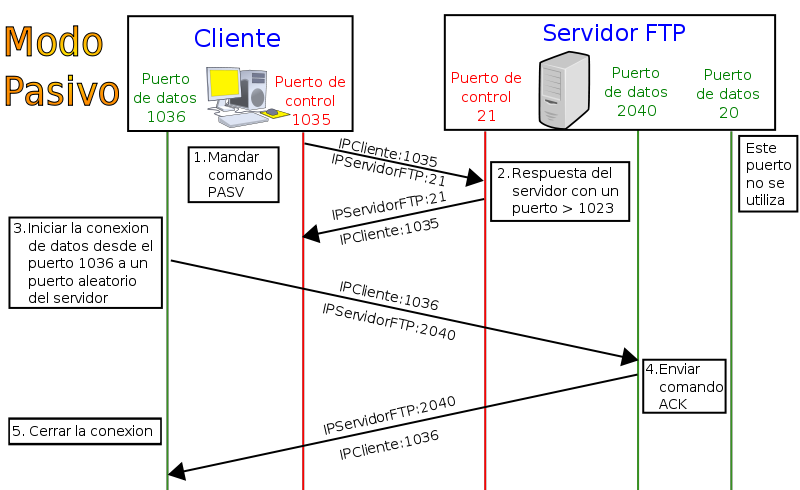
**Modo activo**



Se establecen dos conexiones distintas, la primera conexión con el comando PORT, establece una comunicación para la transmisión de comandos a través de un puerto aleatorio mayor que el 1024 del cliente FTP hacia el puerto 21 del servidor FTP y por esa misma conexión se le notifica al servidor FTP cual es el puerto de nuestro cliente FTP que está a la espera de los datos. Es el servidor FTP el que inicia la transmisión de datos, desde su puerto 20 al puerto que aleatoriamente el cliente FTP le ha indicado. Se le llama modo activo porque la transmisión de datos es iniciada por el propio servidor FTP.

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**



El cliente FTP inicia la conexión con el servidor FTP mediante el envío del comando PASV. En este punto el cliente FTP establece una comunicación mediante un canal de control el cual generalmente utiliza un puerto aleatorio mayor al 1024 para comunicarse con el servidor FTP a través de su puerto 21. Al pasar a modo pasivo el cliente FTP pedirá al servidor FTP un puerto, el cual deberá ser aleatorio y mayor al 1024, recibida la contestación, será el cliente FTP el que establezca la conexión de datos al servidor FTP a

través del puerto especificado anteriormente.

**Modo Activo vs Modo Pasivo**

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.

**Tipos de transferencia: binaria vs ascii**

En el protocolo FTP existen 2 tipos de transferencia

* 1. ASCII
  2. 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 **ASCII**

Adecuado para transferir archivos que solo contengan caracteres imprimibles (archivos ASCII, no archivos resultantes de un procesador de texto), por ejemplo páginas HTML, pero no las imágenes que puedan contener. Se transforman algunos símbolos de control para mantenerlos compatibles entre diferentes sistemas, por ejemplo, si el archivo está alojado sobre un servidor linux, el salto de línea para los archivos de texto es "\n" (byte 10 en decimal). Si el cliente es un sistema Mac, el salto de línea es "\r" (byte 13 en decimal), este modo cambia estos símbolos de control para que el archivo sea legible en ambos lados, al igual que si se envía a un sistema windows, el salto de línea es "\r\n" (dos bytes, 13 y 10). Si se usa este modo en archivos que no son de texto plano, en el caso de intercambiarse entre diferentes sistemas, ese archivo quedará corrupto. Texto puro como por ejemplo .html, .php, .asp, .xml, .txt…

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

Este tipo es usado cuando se trata de archivos comprimidos, ejecutables para PC, imágenes, archivos de audio, entre otros. Archivos comprimidos, ejecutables para PC (.exe), imágenes, video, audio, imágenes de sistema operativo (.iso), archivos .doc, .xls…

Los comandos para transferir en un modo u otro son los siguientes:

|  |  |
| --- | --- |
| binary | Activa el modo de transferencia binario. |
| ascii | Activa el modo de transferencia en modo texto ASCII. |

1. **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).

**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**

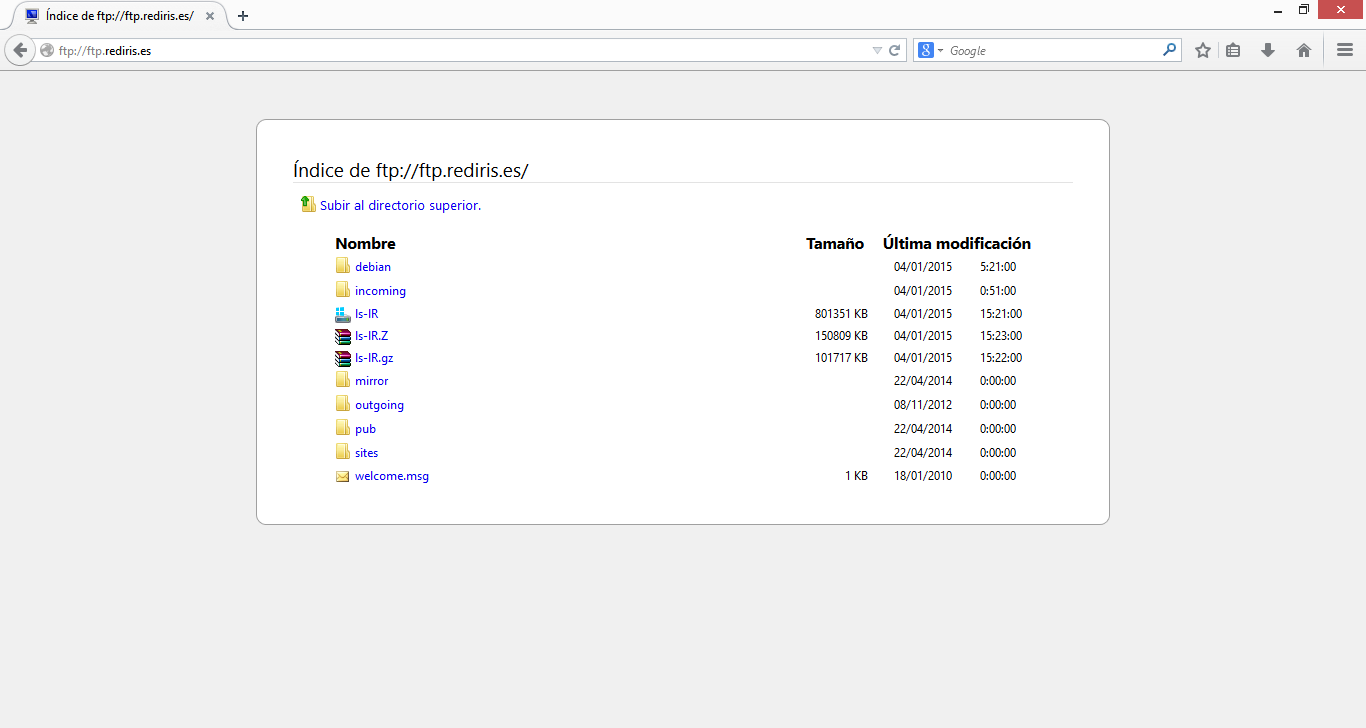
protocolo://[usuario:pass@]servidorFTP[:puerto]/carpeta

**EJEMPLO**

<ftp://jose:783638@ftp.informatik.tu-muenchen.de:1036/kino>

<ftp://ftp.rediris.es:21>

ftp://ftp.rediris.es

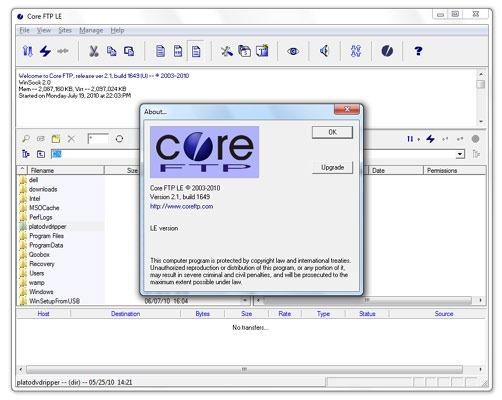


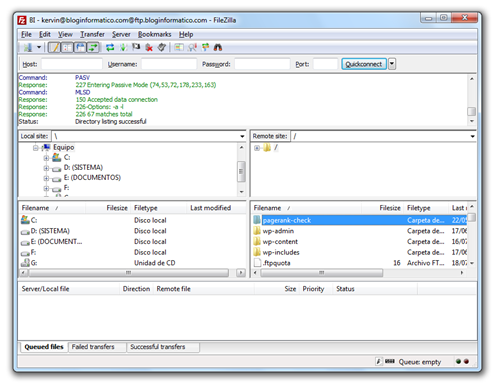
**Utilización de herramientas gráficas.**

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





**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.

prompt off   
mput \*

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.

1. **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. Descargamos para Ubuntu 14.04 desde:

<http://packages.ubuntu.com/trusty/amd64/vsftpd/download>

Instalamos

sudo dpkg -i vsftpd\_3.0.2-1ubuntu2\_amd64.deb

O bien con

apt-get install vsftpd

Una vez instalado podemos ver que se ha añadido un script de autoarranque en /etc/init.d/vsftpd (Ojo si lo hemos instalado con apt-get no se añade este script-usar service)

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 –l /srv

1. **Configuración del servicio de transferencia de archivos I.**

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

* El archivo contiene un conjunto de directivas que determinan el comportamiento del servidor.
* Cada directiva tiene el formato < directiva >=< valor>
* **IMPORTANTE: No debe haber espacios antes y después del signo "=".**
* 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.
* Los comentarios son a nivel de línea y se utiliza el símbolo almohadilla (#).
* Las directivas que no se especifiquen en el fichero de configuración, utilizan su valor por defecto.
* Para consultar las directivas: man vsftpd.conf
* Otras páginas
  + <http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/s1-ftp-vsftpd-conf.html>
  + [**http://manpages.ubuntu.com/manpages/precise/en/man5/vsftpd.conf.5.html**](http://manpages.ubuntu.com/manpages/precise/en/man5/vsftpd.conf.5.html)

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 la ejecución de cualquier comando de escritura sobre el servidor.

write\_enable=YES

**connect\_from\_port\_20**

Si se activa, vsftpd tiene suficientes privilegios para abrir el puerto 20 en el servidor con las transferencias en modo activo.

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

**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.

**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).
* Para 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:

* Haz una copia del fichero de configuración. Borra el contenido del fichero y haz los siguientes puntos
  + Activa el modo standalone y activa el usuario anónimo para que sólo pueda leer.
  + Activa la visualización de mensajes al entrar al directorio home del usuario anónimo. Para ello muestra el siguiente mensaje cuando los usuarios entren dentro del directorio por defecto: “Bienvenido al home del usuario anónimo”.
  + Haz que el número máximo de clientes sea 2. Prueba que al abrir más de 2 conexiones nos avisa del error.

**Solución**

**# a la escucha el servidor**

**listen=YES**

**# allow anonymous FTP?**

**anonymous\_enable=YES**

**# Activate directory messages**

**dirmessage\_enable=YES**

**# Max clientes**

**max\_clients=2**

**Meter .message en /src/ftp**

**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. Pero ojo porque este fichero no existe por defecto hay que generarlo.

OJO!!!Si se utiliza en combinación con la directriz chroot\_local\_user A YES, los usuarios locales listados en el archivo especificado en la directriz chroot\_list\_file, *NO* se colocan en una prisión chroot al conectarse.

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:

1. 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.

1. 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?

[**http://mviera.io/blog/entendiendo-umask/**](http://mviera.io/blog/entendiendo-umask/)

***umask*** *realiza la* ***diferencia a nivel de bits*** *utilizando el operador* ***AND****. Para aclararlo, veamos un ejemplo con los permisos de un fichero:*

*Los permisos base para un fichero son 666 (para un directorio 777) y el valor de umask, por defecto, es 022. Estos son valores en* ***formato octal****, pero el kernel opera a nivel de bits, es decir, en binario. Así que traduciremos estos valores a binario:*

*666 = 110 110 110*

*022 = 000 010 010*

*Para hacer la diferencia con el operador* ***AND*** *tendremos que invertir el valor de 022 con* ***Not****, quedando el siguiente valor:*

*Not 022 = 111 101 101*

*Aplicamos el operador* ***AND*** *entre ambos valores:*

*666 = 110 110 110*

*Not 022 = 111 101 101 AND*

*--------------------------*

*110 100 100*

*El valor resultante es 111 100 100 que si lo volvemos a traducir a octal resulta ser 644.*

Si queremos que los ficheros tengan los mismos permisos base que directorios hay que modificar la directiva siguiente.

local\_umask=011

file\_open\_mode=0777

EJERCICIO

¿Con qué permisos se subirán ficheros/directorios al servidor en los siguientes casos?

* 1. Configuración del fichero vsftpd.conf

local\_umask=011

file\_open\_mode=0777

sol: Ficheros\_\_\_\_\_\_766\_\_\_\_\_\_ y Directorios\_\_\_766\_\_\_\_\_\_\_

* 1. Configuración del fichero vsftpd.conf

local\_umask=077

sol: Ficheros\_\_600 \_\_\_\_\_\_ y Directorios\_\_\_\_\_700\_\_\_\_\_

110.110.110

NOT MASK 111.000.000

110.000.000 = 600

EJERCICIO: Crea un usuario “ejercicio” (adduser ejercicio). Activa el acceso privado en el servidor para únicamente tu usuario de sistema (prueba que el usuario “ejercicio” no puede acceder). Tu usuario local tendrá permisos de escritura, por lo que podrán subir archivos y crear nuevos directorios en su directorio home con permisos 742. Ese usuario por defecto estará enjaulado en su directorio home, no pudiendo salir de él. Limita la tasa de transferencia a 10Kb/s a todos los clientes locales. Pruébalo.

*Ayuda: Si el sistema nos da el sig 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.*

**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 2015 1 127.0.0.1 0 /ftptere/hola.htm b \_ o r teresa 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>

EJERCICIO. Investiga sobre estas 3 directivas.

Registra los accesos primero con la directiva xferlog\_enable=YES. ¿en qué fichero se registran los accesos, con qué formato?

Ahora añade la directiva xferlog\_file=/var/log/vsftpd.log. ¿en qué fichero se registran los accesos, con qué formato?

Ahora añade xferlog\_std\_format=YES y elimina xferlog\_file=/var/log/vsftpd.log. ¿en qué fichero se registran los accesos, con qué formato?

Ahora añade xferlog\_file=/var/log/vsftpd.log. ¿en qué fichero se registran los accesos, con qué formato?

**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

/etc/vsftpd/users/angel

local\_root=/usuarios

write\_enable=NO

EJERCICIO Ahora da acceso al usuario “ejercicio” al ftp. Ningún otro usuario local (excepto teresa) con cuenta en el sistema debe tener permisos de acceso al ftp. Este usuario tendrá un directorio home diferente a teresa...

EJERCICIO: Instala dos clientes gráficos gftp y filezilla y prueba a conectarte.