TEMA 4: File Transfer Protocol (FTP)

Índice

1. Introducción

Una de las principales ventajas que ofrecen las redes basadas en TCP/IP es la posibilidad de transferir información entre los equipos que forman parte de ella. Existen muchos servicios de red que permiten a los usuarios enviar ficheros de unos sistemas a otros, p.e., servicios de correo (adjuntando los mensajes en los mensajes), servicios HTTP (a través de enlaces que abren ficheros de un servidor web), servicios para compartir recursos de red (Samba/SMB/CIFS), etc. Algunos de estos sistemas no están diseñados exclusivamente para transferir ficheros por lo que no están optimizados para ello.
En este tema vamos a explicar un servicio cuyo principal objetivo es la transferencia de ficheros entre distintos sistemas: FTP (File Transfer Protocol).

2. Servicio FTP

FTP es un protocolo de la capa de aplicación diseñado para ofrecer un servicio estándar de transferencia de ficheros entre equipos de una red TCP/IP.

2.1. Características

FTP es uno de los servicios de transferencia de ficheros más antiguo y aún así se sigue usando en Internet y redes corporativas.

Permite a los usuarios:

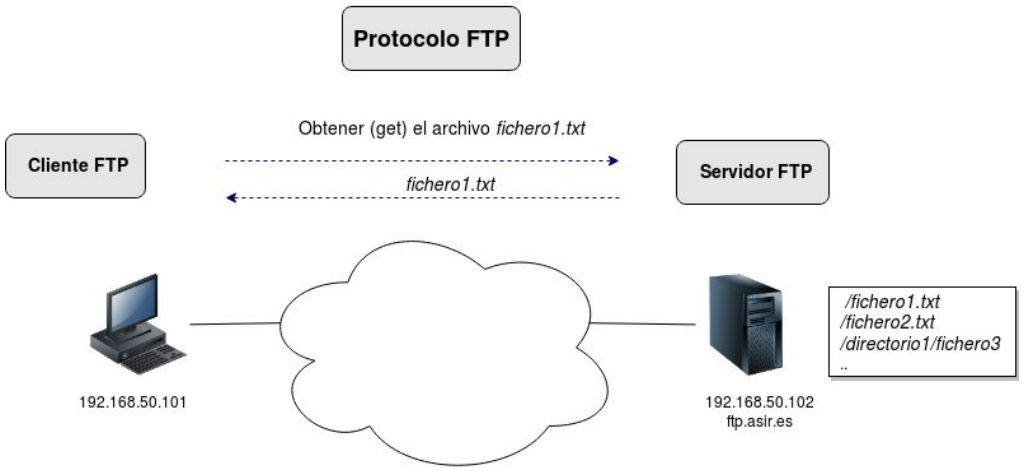
- Acceder a sistemas remotos y listar directorios y ficheros
- Transferir ficheros desde o hacia el sistema remoto, es decir, subir (upload) o bajar (download) ficheros.
- Realizar acciones adicionales en el sistema remoto como renombrar, borrar, crear arvchivos y carpetas, cambiar permisos, descomprimir…

Es un servicio fácil de usar y configurar para los administradores, rápido en la transferencia y oculta a los usuarios los detalles de los sistemas operativos usados. Existen múltiples implementaciones tanto libres como propietarias.

2.2. Componentes y funcionamiento

Su funcionamiento se basa en el modelos **cliente/servidor** y consta de los siguientes componentes:

- **Clientes FTP**: Acceden al sistema de ficheros del equipo donde están instalados y establecen conexiones con los servidores FTP para subir o descargar ficheros.
- **Servidores FTP:** Acceden al sistema de ficheros del equipo donde están instalados, manejan las conexiones de los clientes y en función de los privilegios definidos permiten la descarga y/o la subida de ficheros.
- **Protocolo FTP**: Conjunto de normas y reglas en base a las cuales "dialogan" los clientes y servidores FTP. Usa TCP como protocolo de transporte.

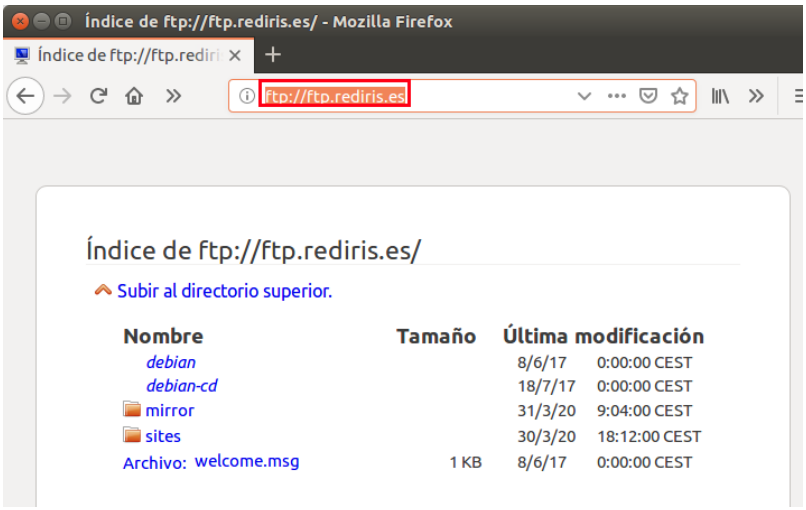


Actividad:

- Busca en Internet servidores FTP públicos y elige alguno (por ejemplo, ftp.rediris.es)
- Desde el terminal de tu equipo, usa el cliente FTP en línea de comandos para conectarte al servidor anterior usando un usuario anónimo (ftp ftp.rediris.es, usuario **anonymous**, contraseña en blanco). Una vez conectado, ejecuta el comando **ls** para listar el contenido del directorio donde has accedido.

```
administrador@marta-casa:~$ 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:administrador): 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
150 Connecting to port 47268
drwxr-xr-x    4 14         50              3864 Sep 20  2017 .
drwxr-xr-x    4 14         50              3864 Sep 20  2017 ..
lrwxrwxrwx    1 14         50                23 Jun  8  2017 debian -> sites/de
bian.org/debian
```

- Usa el navegador como cliente FTP para conectarse al sitio anterior. Observa que se accede al mismo directorio que al usar el cliente en línea de comandos



2.3. Servidores FTP

Un servidor FTP es un programa que atiende y procesa las conexiones de los clientes FTP y que puede acceder al sistema de ficheros del equipo en el que está instalado permitiendo la subida y bajada de ficheros. Ofrece múltiples opciones de configuración para establecer privilegios de los usuarios, limitaciones de subida y descarga, tiempos de conexión y espera, etc.

Existen muchos servidores FTP tanto para sistemas libres como propietarios. Algunos de los más usados son:

- **Sistemas Linux/Unix**
  - vsftpd (http://http://vsftpd.beasts.org/)
  - proftpd (http://www.proftpd.org/)
  - pure-ftpd (https://www.pureftpd.org/)
- **Sistemas Windows**
  - Servidor FTP incluido en IIS (Internet Information Server) (http://www.iis.net)
  - Filezilla Server (http://filezilla-project.org)
  - Serv-U (http://www.serv-u.com)

2.4. Clientes FTP

Programas que acceden al servidor FTP para subir o descargar ficheros.

Existen múltiples clientes FTP, tanto para sistemas libres como propietarios. Se pueden clasificar según la interfaz de usuarios que ofrecen:

2.4.1. Clientes en línea de comandos

La mayoría de los S.O. integran un cliente FTP que se puede invocar desde línea de comandos con la orden ftp. Para iniciar una conexión se usa la sintaxis ***ftp servidor*** (donde "servidor" puede ser una IP o un nombre de dominio, como vimos en la actividad anterior).

Una vez establecida la conexión, el cliente puede usar una serie de comandos (por ejemplo, *ls, get, put, mget, mput, cd, lcd…*) para listar el contenido de los directorios del servidor, bajar o subir ficheros, etc Dependiendo del sistema operativo los comandos que se usan pueden variar. Con el comando **help** o **?** se pueden consultar los comandos disponibles.

Si se quiere ejecutar un comando del equipo local se pone delante el símbolo **!**. Por ejemplo, ***ls*** muestra un listado del directorio actual del servidor y ***!ls*** o ***!dir*** muestran un listado del directorio actual del cliente. Hay una excepción, el comando **cd** cambia el directorio del servidor. Para cambiar el directorio del cliente se debe usar **lcd**, no !cd.

2.4.2. Clientes gráficos

Ofrecen al usuario una interfaz grácica que facilita la conexión al servidor y la transferencia de ficheros. Suele integrar funciones adicionales.

Algunos de los más usados son:

- Filezilla Client (http://filezilla-project.org)
- WinSCP (http://winscp.net)
- Gftp (http://www.gftp.org)
- SmartFCT (http://www.smartftp.com)
- CuteFTP (https://www.globalscape.com/cute-ftp/version/930)

2.4.3. Navegadores/exploradores

Los navegadores (Firefox, Internet Explorer, Google Chrome, Safari…) y los exploradores de archivos (Explorer, Nautilus) actuales pueden actuar como clientes ftp. Para usarlos hay que indicar en la dirección que se realizará la conexión a un servidor ftp:

- Formato general: ftp://[usuario][:password]@servidor
- Ejemplo: ftp://ftp.rediris.es
- Para indicar conexiones con el usuario anonynous no hay que indicar ningún usuario ni contraseña.

Ofrecen un cliente limitado pero sencillo de usar.

2.5. Tipos de acceso

Los servidores FTP permiten dos tipos de acceso desde los clientes:

- **Acceso anónimo**
  - El cliente FTP se conecta al servidor usando un usuarios especial anónimo. Como nombres para este usuario se suelen emplear **anonymous** y/o **ftp**.
  - Generalmente el usuario anónimo solo puede descargar archivos y solo puede acceder a un directorio del servidor, aunque el administrador lo puede configurar de otra forma.

- **Acceso autorizado**
  - El cliente FTP se conecta con un usuario que debe existir en el servidor. Los usuarios pueden ser:
    - Usuarios locales del S.O. donde está instalado el servidor FTP.
    - Usuarios "virtuales", creados para el acceso FTP. Sus credenciales se pueden almacenar en bases de datos, servicios de directorios, ficheros de texto, etc.
  - Una vez que se ha autenticado, el usuario accede a un directorio del servidor (directorio "home" del servidor para ese usuario). Según cómo se haya configurado, podrá ver más directorios o no.
  - En el servidor se configuran los privilegios de cada usuario (descargar, subir, borrar, limitación de espacio, acceso a otros directorios…).

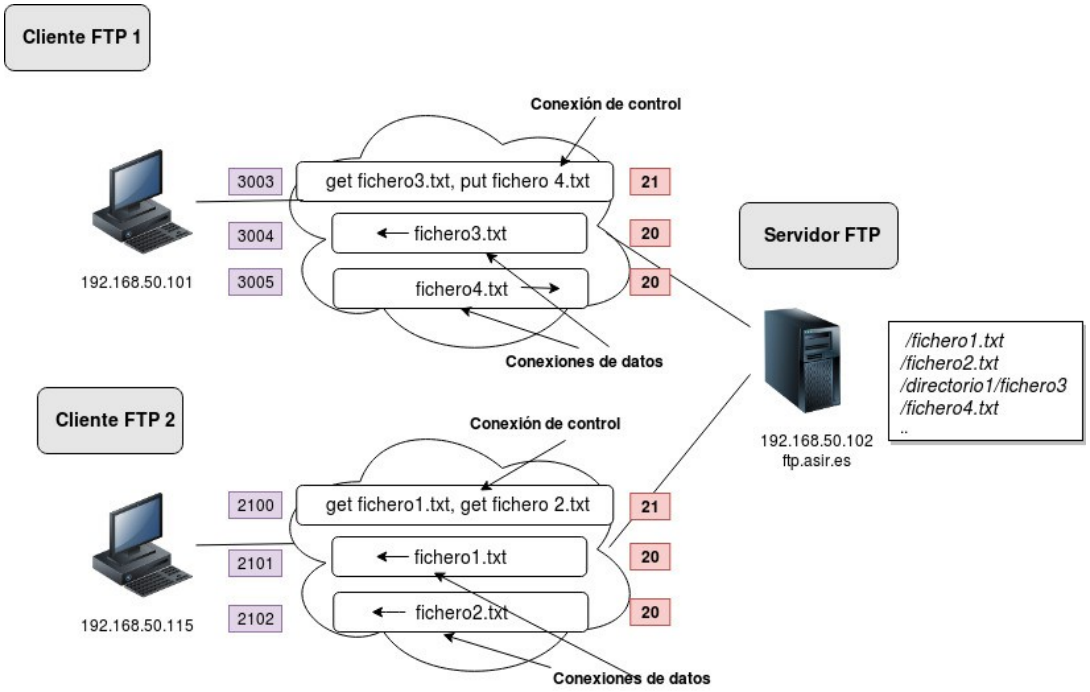
2.6. Conexiones y modos

FTP es un servicio basado exclusivamente en TCP y usa varias conexiones y puertos.

2.6.1. Conexión y control y conexiones de datos

Los servidores y clientes FTP mantienen conexiones TCP independientes para control y para transferencia de datos.

- **Conexión de control**: Cuando el cliente solicita una conexión FTP con el servidor, se establece una conexión de control. El cliente le envía los comandos de descarga (get), subida (put), listado (ls), etc.. y recibe respuestas del servidor que le informan de cómo atiende las peticiones. Esta conexión permanece activa hasta que el usuario cierra la sesión o hasta que el servidor la finaliza porque caduca o por inactividad (timeout). Los servidores pueden atender varias conexiones de control simultáneamente, tantas como se configure en el servidor para evitar sobrecarga.
- **Conexiones de datos**: Cuando el cliente solicita la transferencia de información, se crea una nueva conexión (conexión de datos) que se cierra a finalizar esa transmisión. Por tanto, para una conexiónd de control pueden existir múltiples conexiones de datos, tantas como trasferencias, y hasta un máximo que se configura en el servidor para evitar sobrecarga.



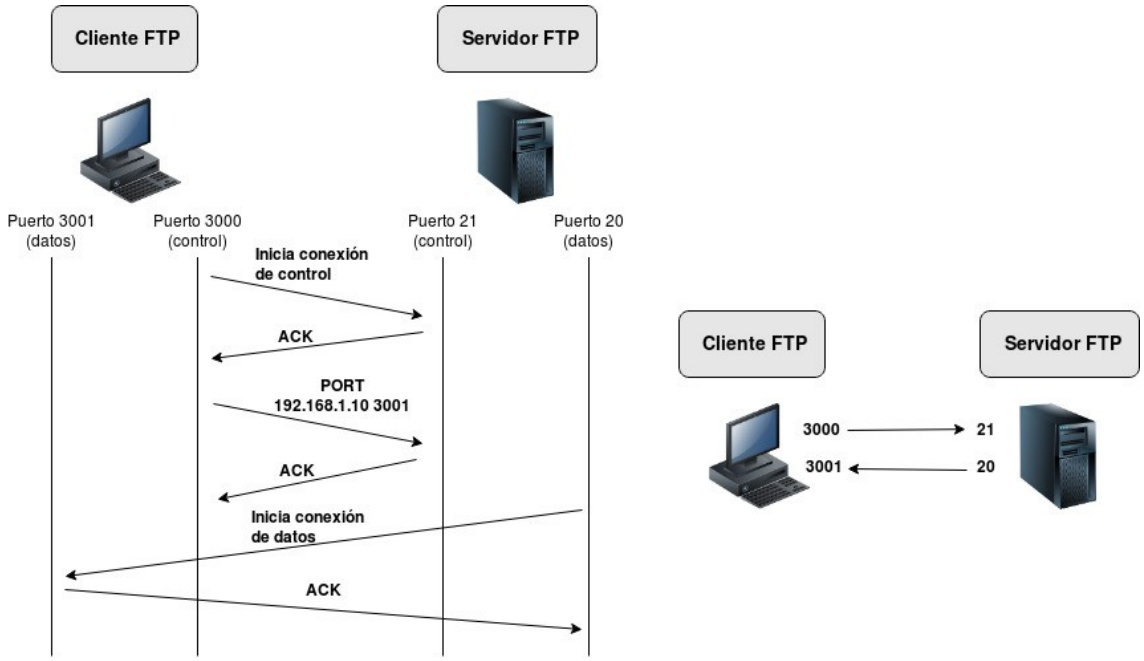
Por la conexión de control nunca se envían datos, y por las de datos nunca se envían comandos de control.

Inicialmente los servidores FTP usaban el puerto 21/TCP para atender las conexiones de control (esto continúa siendo así) y el puerto 20/TCP para iniciar las conexiones de datos. Actualmente no usan siempre el puerto 20 para las conexiones de datos. Los clientes usan puertos mayores a 1023 para iniciar o atender conexiones.

Un cliente FTP puede iniciar ua conexión a un servidor de dos frormas distintas, que se conocen como **modo activo** y **modo pasivo.**

2.6.2. Modo activo

Es el modo nativo de FTP y su configuración y administración es más sencilla:.



1. El cliente solicita el servicio FTP del servidor iniciando una conexión de control:
   - Abre un puerto local superior a 1023 (en el ejemplo 3000)
   - Establece una conexión TCP con el puerto 21 del servidor (FTP)
2. Cuando se solicita la transferencia de ficheros (p.e., el cliente hace un get):
   - El cliente envía el comando PORT al servidor en que especifica su dirección IP y el puerto que "abrirá" para la conexión de datos (en el ejemplo 3001).
   - El servidor inicia una conexión TCP desde su puerto 20 hacia el puerto que le ha indicado el cliente (en el ejemplo 3001). A través de esa conexión se realiza la transferencia de la información.

Por tanto, es el servidor el que inicia las conexiones de datos y el cliente tiene que abrir puertos para atender dichas conexiones. La máquina cliente tiene que aceptar conexiones a puertos superiores a 1023 para la transferencia de datos. Esto puede comprometer la seguridad del equipo.

- El cortafuegos del cliente pueden evitar estas conexiones para prevenir ataques.
- De hecho, ***si el equipo cliente está detrás de un enrutador NAT, éste último descartará las conexiones iniciadas por un servidor FTP externo a los puertos que abre el cliente.*** Es muy habitual que los equipos de una red privada que se

conectan a Internet lo hagan a través de un encaminador NAT, por lo que en este caso tendríamos problemas para cconectarnos a un servidor FTP externo.

Los cortafuegos y/o versiones actuales de enrutadores NAT implementan FTP ALG (Aplication Level Gateway), que es un mecanismo que:

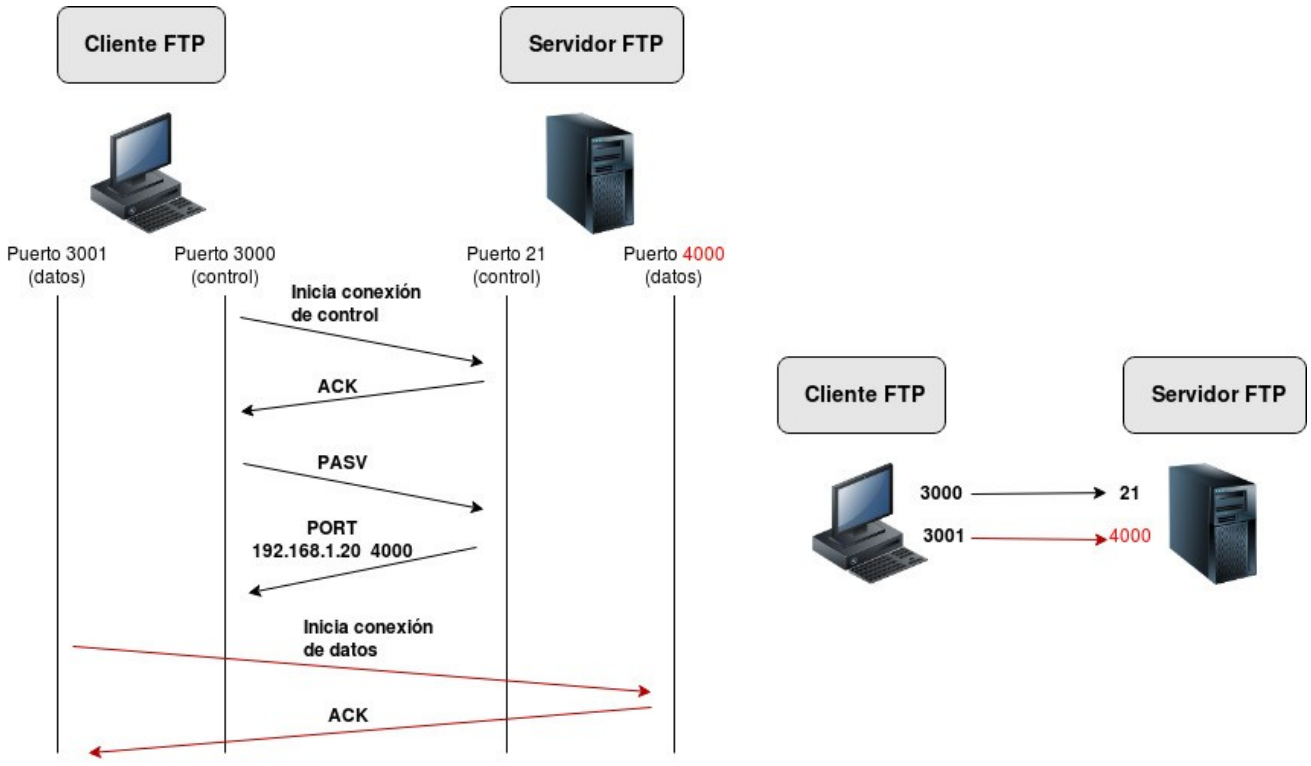
- Analiza las conexiones FTP y se fija en los comandos PORT para registrar la IP privada y el puerto que abre el cliente FTP para aceptar conexiones de datos.
- Los modifica para que el comando PORT que llegue al servidor FTP lo haga con la IP externa y el puerto del equipo que hace de NAT (router).
- Cuando el servidor FTP inicia la conexión de datos, lo hace a la IP externa y puerto del equipo que hace NAT. Este redirige la conexión a la IP privada y puerto del cliente.

Por tanto, sólo si el cortafuegos y/o enrutador NAT implementa ALG, los clientes de la red interna podrán iniciar conexiones FTP con servidores externos usando el modo activo.

Para evitar estos problemas (*que son consecuencia de que es el servidor el que inicia las conexiones de datos*), se desarrolló el modo pasivo.

2.6.3. Modo pasivo

En el modo pasivo es siempre el cliente el que inicia la conexiones con el servidor. El puerto 20 del servidor no se utiliza. Es el modo que se usa para conectarse a un servidor FTP interno desde un cliente externo.



1. El cliente solicita el servicio FTP del servidor iniciando una conexión de control:
   - Abre un puerto local superior a 1023 (en el ejemplo 3000)
   - Establece una conexión TCP con el puerto 21 del servidor (FTP)

- Hasta aquí coincide con el modo activo.

2. Cuando se solicita la transferencia de ficheros (p.e., el cliente hace un get):

- El cliente envía el comando PASV al servidor para activar el modo pasivo. Como respuesta a ese comando el servidor devuelve un número de puerto que tenga disponible (en el ejemplo 4000).
- El cliente inicia una conexión TCP, desde un puerto local superior a 1023 (en el ejemplo 3001) hacia el puerto que el envió el servidor (en el ejemplo 4000).
- Se utiliza esta conexión de datos para realizar la tranferencia de información.

El modo pasivo resuelve el problema de que el cliente tenga que aceptar conexiones en puertos mayores a 1023, pero lo traslada al servidor.

- Ahora el servidor FTP tiene que aceptar conexiones en múltiples puertos, y esto también es una amenaza para la seguridad. Los cortafuegos actuales 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 informó sobre ese puerto.
- Si el servidor FTP está detrás de un enrutador NAT hay que:
  - Configurar en el servidor la IP externa que usa el enrutador NAT y un rango de puertos para aceptar las conexiones de datos.
  - Redirigir el rango de puertos del enrutador NAT al equipo donde está el servidor FTP.

2.6.4. Cortafuegos y encaminadores NAT

El uso de servidores FTP y los modos activo y pasivo implican una configuración adecuada de los cortafuegos y de los encaminadores NAT de la red. Resumimos algunas situaciones comunes:

Cliente FTP

- **Conexión en modo activo**
  - El cortafuegos instalado en el cliente tiene que permitir conexiones TCP salientes hacia el puerto 21 y conexiones TCP entrantes a puertos mayores de 1023 desde el puerto 20.
  - **Si el cliente está detrás del cortafuegos de red y/o de un encaminador NAT**:
    - Si el NAT **no implementa FTP ALG**, existirán problemas porque las conexiones TCP iniciadas desde por servidores FTP externos hacia puertos mayores de 1023 de equipos internos serán filtradas.
    - Si el NAT **implementa FTP ALG**, los clientes podrán usar el modo activo porque sí se permitirán conexiones entrantes a puertos mayoras que 1023.
- **Conexión en modo pasivo**: Los cortafuegos tienen que permitir conexiones TCP salientes hacia el puerto 21 y hacia puertos mayores de 1023.

Servidor FTP

- **Acepta conexiones en modo activo**
  - El cortafuegos tiene que permitir conexiones TCP entrantes al puerto 21 del servidor.
  - El cortafuegos también tiene que permitir conexiones TCP salientes desde el puerto 20 del servidor a puertos superiores a 1023.

- **Acepta conexiones en modo pasivo**
  - El cortafuegos tiene que permitir conexiones TCP entrantes al puerto 21 del servidor, y conexiones TCP entrantes a ciertos puertos mayores de 1023. Es recomendable que el cortafuegos haga 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.
  - **Si el servidor está detrás de un cortafuegos de red y/o de un encaminador NAT:**
    - El cortafuegos tiene que permitir conexiones TCP entrantes al puerto 21
    - Hay que redirigir el puerto 21 del encaminador NAT al puerto 21 del servidor
    - Se debe configurar en el servidor FTP la IP externa del NAT y un rango de puertos para aceptar conexiones de datos
    - Los cortafuegos deben permitir las conexiones TCP entrantes hacia los puertos definidos en el rango.
    - Se deben redirigir el rango de puertos del encaminador NAT al servidor FTP.

2.7. Conexiones y modos

En FTP existen dos modos de transferencia de archivos, ASCII y binario:

- **Formato ASCII (type ascii)**. Se transmite byte a byte. Para archivos de texto (*txt, html, java…)*
- **Formato binario (type bin)**. Se transmite bit a bit. Para archivos que no son de texto *(ejecutables, imágenes, vídeos…).*

Los clientes FCTP permiten definir el formato de transmisión (ASCCI o bin) a utilizar en la transmisión según el tipo de archivos a transferir. Algunos clientes, actualmente casi todos, establecen automáticamente el modo adecuado según el tipo de archivo.

2.8. Seguridad

FTP no es un protocolo seguro. Fue diseñado para ofrecer velocidad pero no seguridad. Se usan mecanismos de autenticación de usuarios para determinar los privilegios de acceso y transferencia en el servidor, pero:

- No se usan mecanismos para garantizar que los equipos involucrados en la transferencia son quienes dicen ser. Es vulnerable a ataques de denegacion de servicio (spoofing).
- Todo el intercambio de información (incluyendo el usuario y la password, y la transferencia de cualquier archivo), se realiza en "texto plano", sin cifrar. Es vulnerable a ataques de análisis de tráfico de red (sniffing).

2.9. FTPS (o FTP/SSL)

FTPS es un conjunto de especificaciones que determinan cómo encapsular FTP en SSL (Secure Sockets Layer) o en TLS (Transport Layer Security), para ofrecer comunicaciones FTP cifradas y, por tanto, seguras. Gracias a los algoritmos de cifrado y a los certificados digitales se puede garantizar la confidencialidad y la integridad de la información transmitida.

Existen dos métodos para implementar FTPS: FTPS explícito (FTPES) y FTPS Implícito:

- **FTPS Implícito**
  - El cliente establece una conexión de control y se establece la conexión SSL/TLS.
  - Si el servidor no soporta FTPS se cierra la conexión.

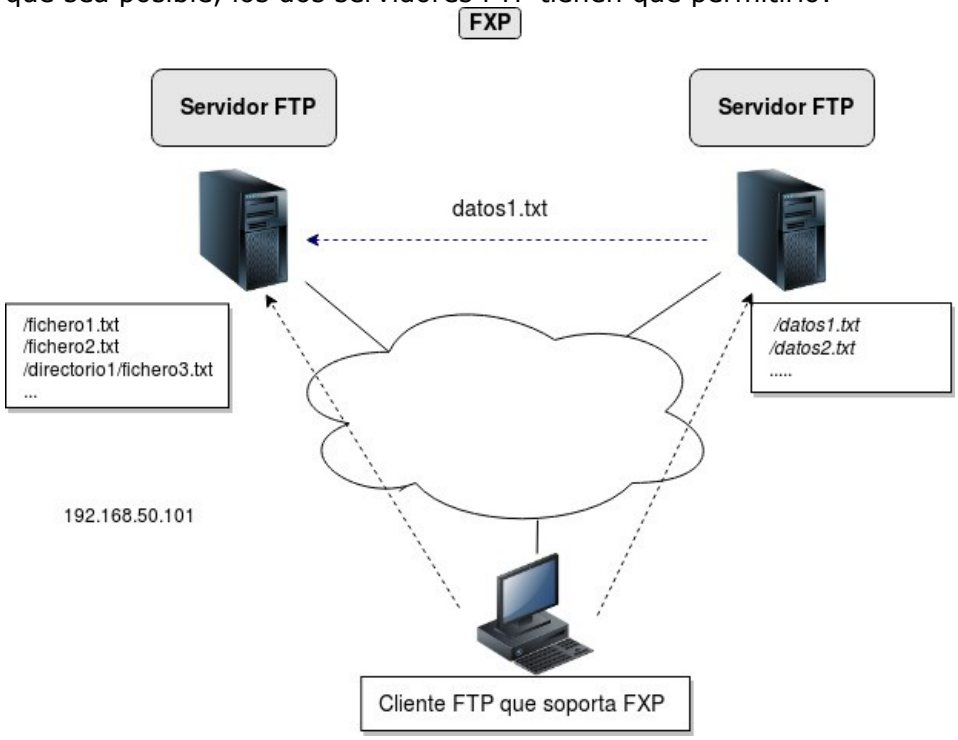
- Todas la comunicaciones, tanto de control como de datos, son cifradas. El cliente y el servidor no negocian.
- Para mantener la compatibilidad con los clientes FTP que no soporten SSL/TLS se usan otros puertos distintos del 20 y 21 para atender las peticiones FTPS (se usan como puertos estándar el 990/TCP para control y el 989/TCP para datos).

- **FTPS Explícito (FTPES)**
  - El cliente establece una conexión de control al puerto 21. Solicita explícitamente que la comunicación sea segura enviando el comando AUTH SSL o AUTH TTL y si el servidor lo soporta se establece una conexión SSL/TLS basándose en algoritmos criptográficos y certificados digitales.
  - Si el servidor no soporta FTPS le ofrece al cliente la posibilidad de usar FTP "normal" no seguro.
  - El cliente y el servidor pueden negociar qué parte de las conexiones de control y de datos se cifra.
  - Es el método recomendado porque permite mayor control sobre la comunicación.

No se debe confundir **FTPS** con **SFTP** (***SSH File Transfer Protocol***), el protocolo de transferencia de ficheros basado en SSH que veremos en el tema de SSH. Ni con enviar el protocolo FTP a través de una conexión SSH (túnel FTP sobre SSH) conocido como *Secure FTP*. Por tanto, ***FTPS, SFTP y Secure FT***P son diferentes.

2.10. Protocolo FXP

FXP (*File eXchange Protocol*) es un protocolo de transferencia de datos directa entre servidores FTP, usando un cliente solo para conectarlos inicialmente. El cliente sólo se usa para la conexión inicial y no para la transferencia de ficheros, que se hace directamente de un servidor a otro. Para que sea posible, los dos servidores FTP tienen que permitirlo:



FXP se puede usar, por ejemplo, para migrar ficheros de un servidor FTP a otro, ahorrando la descarga desde un servidor al cliente y posteriormente la subida desde el cliente al otro servidor (se consigue más rapidez y menos sobrecarga de red).

3. Servicios SFTP/SCP

SSH (Secure Shell Protocol) es un protocolo de capa de aplicación diseñado para ofrecer un servicio de acceso a terminales de equipos remotos. Está basado en el modelo cliente/servidor. El cliente SSH permite establecer conexiones a terminales del equipo donde se ejecuta el servidor SSH. Los servidores SSH usan el puerto 22/TCP como puerto estándar.

A diferencia de otros protocolos de acceso remoto, como Telnet, ofrece autenticación, confidencialidad e integridad.

- Se autentifica a los dos extremos de la conexión:
  - El servidor se autentica ante el cliente con una clave.
  - El cliente se autentica ante el servidor
- Se cifran los datos intercambiados:
  - Nombres de usuarios y passwords viajan cifrados
  - La información transmitida también viaja cifrada

SSH, además de otras funcionalidades, integra mecanismos de transferencia de ficheros garantizando también autenticación, confidencialidad e integridad. Se basa en los protocolos SFTP (*SSH File Transfer Protocol)* y SCP (*Secure Copy Protocol).*

- **SFTP**
  - Permite la transferencia de ficheros entre sistemas remotos
  - Permite listar ficheros y directorios del servidor
  - Permite realizar funciones adicionales en el servidor, como renombrar, borrar, crear archivos y carpetas, cambiar permisos, descomprimir.
- **SCP**
  - Permite la "copia" de ficheros entre sistemas remotos.
  - Hay clientes SCP gráficos que integran funcionalidades adicionales como listar, borrar, et. No son clientes scp "puros".

Los **servidores SSH** atienden peticiones de transferencia de ficheros desde clientes SFTP y/o SCP. Ejemplos de servidores SSH son OpenSSH y WinSSHD.

Existen múltiples **clientes SFTP/SCP** y se pueden clasificar según la interfaz de usuario que ofrecen:

- **Clientes en línea de comandos**
  - Clientes que se pueden invocar desde línea de comandos
  - Clientes scp y sftp (este último con comandos similares a los clientes ftp, como get, put, mget, mput…)
- **Clientes gráficos**
  - La mayoría de los clientes gráficos FTP también pueden actuar como clientes SFTP/SCP