## 1. FTP

**1. Historia**

El Protocolo de Transferencia de Archivos (FTP) fue introducido en abril de 1971 con la publicación del RFC 114, convirtiéndose en uno de los primeros protocolos para la transferencia de archivos en redes ARPANET, precursor de Internet. En 1973, el RFC 354 revisó y formalizó la estructura general del protocolo, sentando las bases para su implementación más amplia.

A lo largo de los años, FTP ha experimentado múltiples actualizaciones para mejorar su funcionalidad y adaptarse a las necesidades emergentes de la red. Una de las revisiones más significativas fue la publicación del RFC 765 en 1980, que introdujo mejoras técnicas y optimizaciones.

La versión más ampliamente adoptada en la actualidad es la definida en el RFC 959, publicada en octubre de 1985. Este documento consolidó y amplió las especificaciones anteriores, proporcionando un estándar más robusto y flexible. El RFC 959 introdujo mejoras significativas en la gestión de la transferencia de archivos, la autenticación y la interactividad entre el cliente y el servidor, estableciendo una base sólida para la interoperabilidad en redes TCP/IP.

Desde su concepción, FTP ha sido una herramienta esencial para la transferencia de datos entre sistemas. Sin embargo, con el avance de las tecnologías de seguridad, su uso ha disminuido en favor de protocolos más seguros como SFTP (SSH File Transfer Protocol) y FTPS (FTP Secure), que ofrecen cifrado para proteger la integridad y confidencialidad de los datos durante la transferencia.

FTP sigue siendo relevante en ciertos contextos, especialmente en redes controladas y ambientes donde la simplicidad y la compatibilidad son prioritarias. Sin embargo, su falta de cifrado nativo y la vulnerabilidad a ataques han llevado a una adopción creciente de alternativas más seguras en entornos donde la seguridad es crítica.

**2. ¿Qué es?**

El Protocolo de Transferencia de Archivos (FTP) es un estándar de la red para la transferencia de archivos entre sistemas conectados a una red TCP/IP. Funciona bajo una arquitectura cliente-servidor, donde un cliente se conecta a un servidor para descargar o enviar archivos. El servicio FTP es ofrecido por la capa de aplicación del modelo de capas de red TCP/IP, utilizando normalmente los puertos 20 y 21.

**3. Modelo de Funcionamiento**

El modelo de funcionamiento de FTP describe la comunicación entre el cliente y el servidor a través del Intérprete de Protocolo (PI) y el Proceso de Transferencia de Datos (DTP). La comunicación es independiente del sistema de archivos utilizado en cada computadora, lo que permite la interoperabilidad entre sistemas operativos. Los dos canales principales usados para la comunicación entre el cliente y el servidor:

* Canal de Comandos (PI): Utiliza el puerto 21 y se encarga del control y la administración de la conexión.
* Canal de Datos (DTP): Utiliza el puerto 20 (en modo activo) o un puerto definido por el servidor (en modo pasivo) para la transferencia real de archivos.

El proceso de transferencia involucra varios pasos:

1. Establecimiento de Conexión: El cliente inicia la conexión con el servidor a través del canal de comandos.
2. Autenticación: El cliente proporciona credenciales para autenticarse en el servidor.
3. Comandos de Archivo: El cliente envía comandos al servidor para navegar y gestionar archivos.
4. Transferencia de Datos: Utilizando el canal de datos, se realizan las transferencias de archivos según los comandos emitidos.
5. Cierre de Conexión: Una vez completadas las transferencias, el cliente cierra la conexión.

**4. Modos de Conexión**

FTP admite dos modos de conexión del cliente: modo activo (o estándar, o PORT) y modo pasivo (o PASV). Ambos modos implican que el cliente establece inicialmente una conexión con el servidor mediante el puerto 21, que es el canal de control. Sin embargo, difieren en cómo se establece la conexión de datos.

Modo Activo:

1. Establecimiento del Canal de Control: El cliente se conecta al servidor mediante el puerto 21.
2. Comando PORT: El cliente envía un comando PORT al servidor por el canal de control, indicando un puerto aleatorio mayor que el 1024 en el cliente, donde está dispuesto a recibir la conexión de datos.
3. Creación del Canal de Datos: El servidor, usando su puerto 20, inicia una conexión hacia el puerto especificado por el cliente para establecer el canal de datos.

El principal problema del modo activo es de seguridad, ya que la máquina cliente debe aceptar conexiones entrantes en puertos superiores al 1024. Esto puede ser problemático en redes inseguras como Internet, ya que muchos firewalls rechazan conexiones aleatorias entrantes para evitar ataques.

Modo Pasivo: Fue desarrollado con el fin de resolver los problemas de seguridad del modo activo

1. Establecimiento del Canal de Control: El cliente se conecta al servidor mediante el puerto 21.
2. Comando PASV: El cliente envía un comando PASV al servidor por el canal de control.
3. Respuesta del Servidor: El servidor responde con un número de puerto aleatorio mayor que 1024 donde está dispuesto a recibir la conexión de datos.
4. Creación del Canal de Datos: El cliente, utilizando un puerto aleatorio propio mayor que 1024, inicia una conexión hacia el puerto especificado por el servidor para establecer el canal de datos.

El modo pasivo es preferido en entornos de red donde existen restricciones de seguridad, ya que evita que el cliente tenga que aceptar conexiones entrantes, mitigando así los problemas asociados con firewalls y redes inseguras.

Proceso de Transferencia

Independientemente del modo utilizado (activo o pasivo), antes de cada nueva transferencia, el cliente debe enviar el comando correspondiente (PORT para modo activo o PASV para modo pasivo) para iniciar una nueva conexión de datos. En el modo activo, el servidor recibirá la conexión de datos en su puerto 20, mientras que en el modo pasivo, el servidor recibirá la conexión en un puerto aleatorio especificado en su respuesta al comando PASV.

Con estos modos de conexión, FTP permite una flexibilidad en la forma en que se establecen las conexiones de datos, lo que es crucial para adaptarse a diferentes configuraciones de red y requisitos de seguridad.

**5. Tipos de Transferencia de Archivos**

En el protocolo FTP, existen dos tipos principales de transferencia de archivos: ASCII y binario. Es crucial elegir el tipo de transferencia adecuado para evitar la corrupción de los archivos durante el proceso de transmisión.

Tipo ASCII: el tipo ASCII es adecuado para transferir archivos que solo contengan caracteres imprimibles, como archivos de texto plano y páginas HTML (excluyendo las imágenes). Este modo se utiliza para garantizar que los archivos de texto sean legibles y compatibles entre diferentes sistemas operativos, ajustando automáticamente los caracteres de control según sea necesario:

* Linux: Salto de línea es "\n" (byte 10 en decimal).
* Mac: Salto de línea es "\r" (byte 13 en decimal).
* Windows: Salto de línea es "\r\n" (dos bytes, 13 y 10).

Al transformar estos caracteres de control, se asegura que los archivos de texto sean interpretados correctamente en cualquier sistema. Sin embargo, usar el modo ASCII para archivos que no son de texto plano (como binarios o ejecutables) puede resultar en la corrupción de dichos archivos, ya que los cambios en los caracteres de control pueden alterar la estructura del archivo.

Tipo Binario: el tipo binario se utiliza para transferir archivos que contienen datos binarios, como archivos comprimidos, ejecutables, imágenes, archivos de audio, entre otros. En este modo, los archivos se transfieren tal cual, sin ningún tipo de transformación o ajuste, preservando la integridad del archivo original. Este modo es esencial para asegurar que los archivos complejos y no textuales se transfieran sin errores.

### **Ejemplos de Transferencias según la Extensión del Archivo**

| **Extensión de archivo** | **Tipo de transferencia** |
| --- | --- |
| txt (texto) | ASCII |
| html (página WEB) | ASCII |
| doc (documento) | Binario |
| ps (postscript) | ASCII |
| hqx (comprimido) | ASCII |
| Z (comprimido) | Binario |
| ZIP (comprimido) | Binario |
| ZOO (comprimido) | Binario |
| Sit (comprimido) | Binario |
| pit (comprimido) | Binario |
| shar (comprimido) | Binario |
| uu (comprimido) | Binario |
| ARC (comprimido) | Binario |
| tar (empaquetado) | Binario |

Software de Transferencia FTP: Existen diversas soluciones de software que facilitan la transferencia de archivos mediante FTP. Algunos de los más conocidos son:

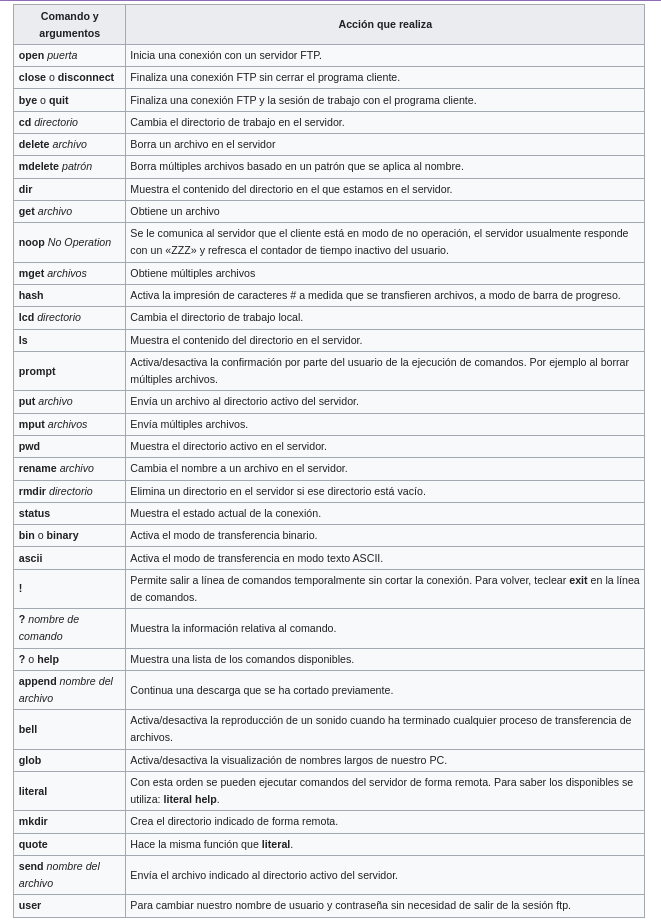
* FileZilla: Software libre y de código abierto que soporta FTP, SFTP y FTPS.
* CuteFTP: Software de pago (shareware) que ofrece una interfaz amigable y soporte para múltiples protocolos de transferencia.

Al utilizar estas herramientas, se puede seleccionar fácilmente el tipo de transferencia adecuado (ASCII o binario) para cada archivo, garantizando así la integridad y compatibilidad de los datos transferidos.

**6. Servidores y Clientes FTP**

Un servidor FTP es un programa que permite el intercambio de datos entre diferentes servidores/ordenadores. Por otro lado, un cliente FTP es un programa instalado en el ordenador del usuario que se conecta al servidor FTP para transferir archivos. Los servidores FTP suelen utilizarse para alojamiento web, copias de seguridad, entre otros fines.

**7. Comandos FTP**



**8. Vulnerabilidades y Problemas de Seguridad**

FTP presenta varias vulnerabilidades significativas debido a su falta de cifrado y mecanismos robustos de autenticación. Algunas de las principales vulnerabilidades incluyen:

* Ataques Man-in-the-Middle: Los datos y credenciales transmitidos pueden ser interceptados y modificados por un atacante debido a la falta de cifrado.
* Fuerza Bruta: Las contraseñas de FTP pueden ser fácilmente atacadas mediante métodos de fuerza bruta debido a la ausencia de protección adicional, como el cifrado.
* Acceso No Autorizado: Sin cifrado, los atacantes pueden obtener acceso a datos sensibles que se transmiten en texto plano, lo que compromete la integridad y la privacidad de la información.

Para mitigar estos problemas de seguridad, es recomendable utilizar alternativas más seguras, como SCP (Secure Copy Protocol) y SFTP (SSH File Transfer Protocol). Estas soluciones cifran todo el tráfico, protegiendo la información sensible durante la transferencia. SCP y SFTP están incluidos en el paquete SSH (Secure Shell), proporcionando un nivel adicional de seguridad y confiabilidad para las transferencias de archivos. Utilizar estos protocolos garantiza que los datos estén protegidos contra interceptaciones y accesos no autorizados, resolviendo las vulnerabilidades inherentes a FTP.

## 2. FTPS

**1. Definición:**

FTPS (FTP sobre SSL) es una extensión de FTP, debido a que este último no es un canal seguro de transferencia de archivos. Además, FTPS surge como alternativa al RFC 2228.

Para solucionar el problema de la confidencialidad (cifrado de los datos) en la autenticación y en la transferencia de datos, se decidió añadir una capa de seguridad SSL/TLS al propio protocolo FTP. FTPS y FTPES también se conocen como FTP over TLS/SSL, y están basados en el propio protocolo FTP. No debe confundirse con el protocolo seguro de transferencia de ficheros SFTP, el cual suele ser usado con SSH.

**2. Funcionamiento:**

Una conexión FTPS se autentica con un identificador de usuario, una contraseña y un certificado de clave pública. Al conectarse a un servidor FTPS, un cliente FTPS verificará primero la fiabilidad del certificado del servidor.

* Cuando una autoridad de certificación (CA) de confianza firma estos certificados, garantiza que el cliente se está conectando a un servidor seguro y de confianza, lo que ayuda a protegerse de los ataques de intermediarios.
* Aquellos certificados no firmados por una CA de confianza, que se conocen como certificados autofirmados, pueden hacer que el cliente FTPS genere una advertencia diciendo que el certificado no es válido. El cliente puede optar por aceptar el certificado o rechazar la conexión.

FTPS (sobre SSL/TLS) utiliza certificados X.509 para la autenticación. Estos certificados digitales incluyen una clave pública de cifrado e información sobre el propietario del certificado. La clave pública tiene dos funciones principales: validación y cifrado de datos. La clave pública tiene asociada una clave privada. Esta clave privada se almacena separada del certificado y se utiliza para descifrar el mensaje cifrado por la clave pública.

**3. Desventajas:**

* Un servidor FTPS es complejo al configurar comparado con uno que utiliza FTP estándar.
* Puede llegar a utilizar más recursos de CPU y ancho de banda por el cifrado SSL/TLS.
* Al solicitar la apertura de puertos adicionales en el firewall para permitir la transferencia de archivos segura, puede aumentar los riesgos de seguridad si es que no se configura correctamente.
* La gestión de certificados SSL/TLS puede ser compleja y requiere renovaciones periódicas, lo que añade una capa adicional de mantenimiento.

**4. Ventajas:**

Al proporcionar una capa adicional mediante el cifrado SSL/TLS garantiza que los datos transferidos entre el cliente y el servidor estén protegidos contra posibles ataques de escucha o interceptación, esto demuestra una clara ventaja sobre el propio FTP. Además, el uso del FTPS, puede ayudar a cumplir con los requisitos de seguridad y privacidad de datos al admitir varios métodos de autenticación incluyendo las certificaciones digitales.

Por otro lado, facilita la integración con firewalls al utilizar puertos estándar (21 y 990) y protocolos estándar (TCP/IP)

**5. Uso de RFC:**

El RFC que utiliza FTPS es el RFC 2228**(FTP Security Extensions)** que proporciona una guía detallada sobre cómo implementar el cifrado TLS en el protocolo FTP. Describe cómo se pueden realizar las negociaciones de seguridad entre el cliente y el servidor FTP para establecer una conexión segura. Este RFC define un comando llamado “AUTH”, permitiendo que el cliente solicite un intercambio seguro de comandos y datos utilizando un mecanismo de autenticación específico, como TLS. También describe el proceso de negociación entre el cliente y el servidor para establecer una conexión segura. Por último, proporciona directrices para permitir la interoperabilidad entre clientes y servidores que no admiten las extensiones de seguridad

**6. Vulnerabilidades:**

* Se podría interceptar y modificar las comunicaciones entre el servidor FTPS y el cliente si es que la autenticación del servidor no se realizó correctamente o si los certificados no se validan correctamente por parte del cliente.
* En el caso de que se utilicen algoritmos de cifrado débiles o protocolos obsoletos en la implementación de FTPS, se puede explotar esta vulnerabilidad para comprometer la confidencialidad de los datos transferidos.
* El servidor puede ser expuesto a vulnerabilidades si es que no se actualiza regularmente o no se asegura adecuadamente.
* En el caso de que las credenciales del servidor se transmitan o almacenen de manera insegura, podrían ser interceptadas por el atacante y utilizadas para acceder al servidor de manera no autorizada.

## Biografía

<https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_archivos>

<https://www.goanywhere.com/es/blog/que-es-ftps>

<https://www.serv-u.com/what-is-file-transfer-protocol-secure>