FTP, File, and Print Server

Hiranya Prasad Bastakoti

Contents

- General Samba Configuration
- CUPS configuration basics
- FTP Principles
- Anonymous FTP Server
- Troubleshooting

FTP

- FTP (File Transfer Protocol) is a protocol for transferring a file from one host to another host.
- Transferring files from a client computer to a server computer is called **"uploading"** and transferring from a server to a client is **"downloading".**
- In a typical FTP session, the user is sitting in front of one host (the local host) and wants to transfer files to or from a remote host.
- In order for the user to access the remote account, the user must provide a user identification and a password.
- After providing this authorization information, the user can transfer files from the local file system to the remote file system and vice versa.
- As shown in Figure below the user interacts with FTP through an FTP user agent. The user first provides the hostname of the remote host, which causes the FTP client process in the local host to establish a TCP connection with the FTP server process in the remote host. The user then provides the user identification and password, which get sent over the TCP connection as part of FTP commands.
- Once the server has authorized the user, the user copies one or more files stored in the local file system into the remote file system (or vice versa).

FTP

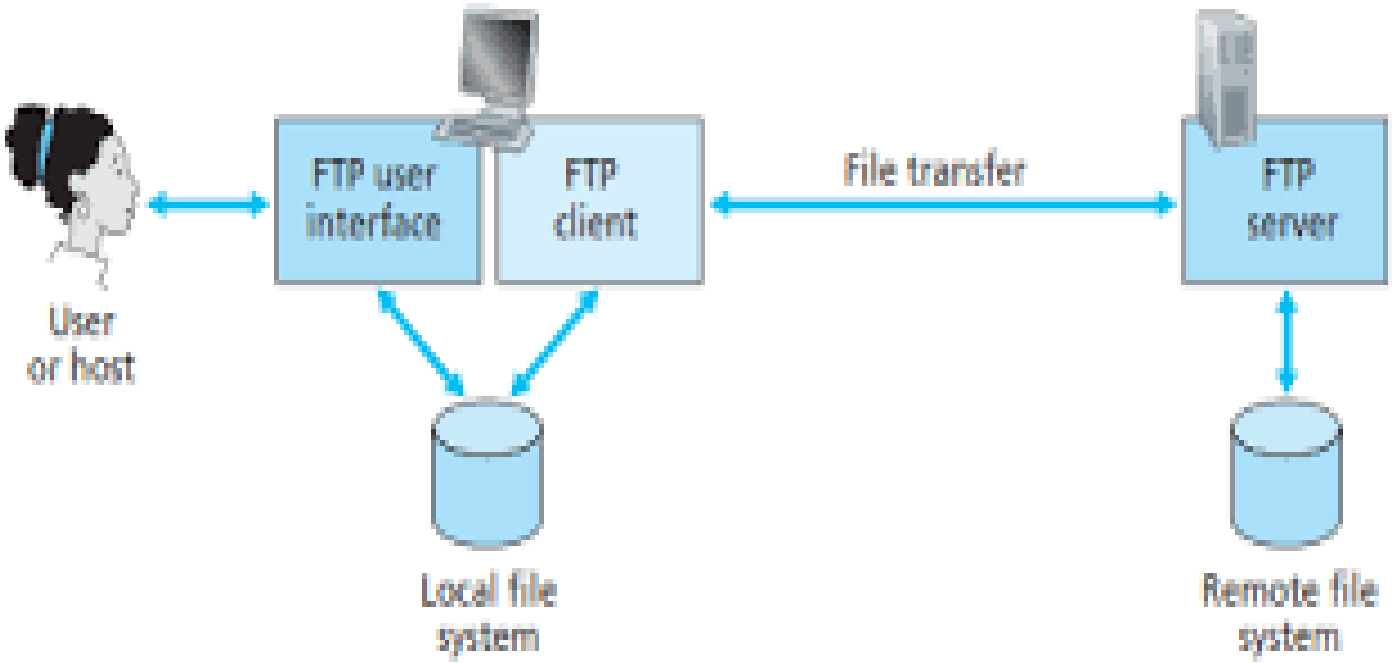


Figure 2.14 ♦ FTP moves files between local and remote file systems

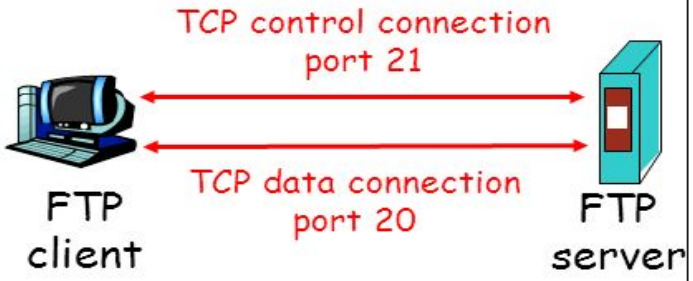
FTP Session

- When an FTP session is started between a client and a server, the client initiates a control TCP connection with the server side.
- The client sends control information over this. When the server receives this, it initiates a data connection to the client side.
- Only one file can be sent over one data connection. But the control connection remains active throughout the user session.
- As we know HTTP is stateless i.e. it does not have to keep track of any user state. But FTP needs to maintain a state about its user throughout the session.

FTP

FTP: Control and Data connections

- FTP client contacts FTP server at port 21
- Client obtains **authorization** over control connection
- Client browses remote directory by sending commands over control connection.
- When server receives a command for a file transfer, the server opens a TCP data connection to client
- After transferring one file, server closes connection.



- Server opens a second TCP data connection to transfer another file.
- FTP server maintains "state": current directory, earlier authentication

FTP Connection

1. **Control Connection:** For sending control information like user identification, password, commands to change the remote directory, commands to retrieve and store files, etc.,
- FTP makes use of a control connection.
- The control connection is initiated on port number 21.
2. **Data connection:** For sending the actual file, FTP makes use of a data connection.
- A data connection is initiated on port number 20.

FTP Principles

- Establishing Connection: The client initiates a connection to the server using FTP client software, specifying the server's IP address and port 21 (the default control port).
- Authentication: The client provides credentials (username and password) to log in to the server. Anonymous access may be allowed, where a generic username is used, and an email address serves as the password.
- Control Connection: Upon successful authentication, a control connection is established between the client and server. This connection facilitates the exchange of FTP commands and responses, remaining open throughout the session.
- Command and Response : The client sends commands over the control connection to request actions from the server. These commands involve uploading, downloading, creating directories, deleting files, and navigating the directory structure. The server responds with success or failure indications for each command.

- Data Transfer: For file transfers, a separate data connection is established between the client and server. FTP supports active and passive modes.
  - Active Mode: The server initiates a connection to the client, which listens on a specific port for data transfer.
  - Passive Mode: The client initiates a connection to the server, which listens on a specific port for data transfer.
- File Transfer: With the data connection established, the actual file transfer occurs. Files are transmitted in blocks or packets. FTP offers ASCII and binary modes.
  - ASCII Mode: Suitable for text-based files, such as HTML or source code, as it transfers files as plain text and adjusts line endings to match the target system's format.
  - Binary Mode: Intended for non-text files like images or executables, ensuring the exact structure of the files is maintained during transfer.
- Closing the Connection: After completing the file transfer, the client or server can issue commands to close the data connection and terminate the FTP session. The control connection remains open until explicitly closed.

FTP commands, responses

Sample commands:

- sent as ASCII text over control channel
- **`USER`** ***`username`***
- **`PASS`** ***`password`***
- **`LIST`** return list of file in current directory
- **`RETR filename`** retrieves (gets) file
- **`STOR filename`** stores (puts) file onto remote host

Sample return codes

- status code and phrase (as in HTTP)
- **`331 Username OK, password required`**
- **`125 data connection already open; transfer starting`**
- **`425 Can't open data connection`**
- **`452 Error writing file`**

Advantages of FTP

- Speed is one of the advantages of FTP(File Transfer Protocol).
- File sharing also comes in the category of advantages of FTP in this between two machines files can be shared on the network.
- Efficiency is more in FTP.
- Unlimited Data Trasfer

Disadvantages of FTP

- Multiple receivers are not supported by the FTP.
- FTP does not encrypt the data this is one of the biggest drawbacks of FTP.
- FTP is unsecured we use login IDs and passwords making it secure but they can be attacked by hackers.

Anonymous File Transfer Protocol

- AFTP (Anonymous File Transfer Protocol) is a network protocol used for transmitting files using TCP-based networks.
- Anonymous file transfer protocol lets a user move files anonymously from one computer to another.
- Anonymous FTP operates at layer 7; anonymous FTP permits anonymous external computer users without any designated password or user ID to access the FTP server i.e., When a user accesses a file, they don't need to identify themselves.
- Hence, all the data contained within a website that allows Anonymous FTP should be considered publicly accessible.

Steps

- The user needs to Log in to the localhost and invoke the FTP program.
- Then open a connection to the host.
- After the connection is established to the remote host, the user needs to log in with the username “anonymous”.
- After supplying the username, the user needs to provide the password.
- After supplying the password, supply whatever FTP commands are required by the user.
- And finally, after it’s finished, exit the FTP program, which will close the connection

Advantages of Anonymous FTP:

- It doesn’t need any authentication.
- It allows fast access to public archives without any web server processes.
- Not only AFTP can transfer more than one file but, it also allows the transfer of multiple directories at the same time.

Disadvantages of Anonymous FTP:

- It is not completely anonymous.
- Less control over who is accessing your FTP server as it is completely public.
- It can compromise the whole system if not used properly. i.e., extra security steps are needed to prevent any exploits.

Samba

- Samba is an open-source software suite designed to facilitate seamless communication between Unix/Linux-based platforms and Windows clients.
- By utilizing the Common Internet File System (CIFS) protocol, Samba enables Unix/Linux systems to offers services that replicate the functionality of native Windows applications.
- At its core, Samba leverages the Server Message Block (SMB) protocol to accomplish four key functions:
- File & print services
- Authentication and Authorization
- Name resolution
- Service announcement (browsing)

- File and print services: Samba allows Unix/Linux servers to offer file and printer sharing capabilities, enabling Windows clients to access and utilize these resources as if they were interacting with a Windows-based server.
- Authentication and Authorization: Samba provides mechanisms for authenticating and authorizing users, ensuring secure access to shared files and printers. It supports various authentication methods, including integration with Windows domains.

- Name resolution: Samba assists in resolving network names between Unix/Linux and Windows systems. This enables seamless identification and communication between machines on the network, despite differences in their underlying operating systems.
- Service announcement (browsing): Samba facilitates the discovery and announcement of available services on the network, allowing Windows clients to browse and connect to shared resources offered by Unix/Linux servers.

CUPS

- CUPS, short for "Common UNIX Printing System," is a modular printing system designed for Unix-like operating systems. It acts as a printer server in networked computer environments. It is freely available under the Apache License.
- CUPS operates on a host computer and handles printing tasks received from client computers. It acts as an intermediary between the clients and the connected printers, managing the printing process.

Components of CUPS

- Print Spooler: CUPS includes a print spooler, which receives and stores print jobs in a queue. This allows multiple print jobs to be managed and processed in an orderly manner.
- Scheduler: The scheduler component in CUPS determines the order in which print jobs are processed and sent to the printers. It ensures fair access to the printers and manages the printer resources efficiently.
- Filter System: CUPS uses a filter system to convert the print data received from client computers into a format that the printers can understand. These filters handle tasks such as format conversion, image rendering, and color management.
- Backend System: The backend system in CUPS provides the interface between the printing system and the physical printers. It handles communication with the printers, manages printer settings, and ensures proper printer functionality.

Configuration steps

1. Install CUPS: Begin by using the package manager to install CUPS on your CentOS system. Open the terminal and execute the following command as a superuser:

   sudo yum install cups

2. Start the CUPS service: Once the installation is complete, initiate the CUPS service using the following command:

   sudo systemctl start cups

3. Enable the CUPS service: To ensure that CUPS starts automatically upon system boot, enable the service with this command:

```
sudo systemctl enable cups
```

4. Access the CUPS web interface: CUPS provides a web-based interface for configuration. Open a web browser and enter the following URL:

   http://localhost:631

5. Configure printers: In the CUPS web interface, navigate to the "Administration" tab and select "Add Printer". Follow the prompts to add and configure your printers. Provide the necessary details such as printer name, description, and connection type. If the printer driver is not automatically detected, you might need to provide it manually.
6. Set access control: By default, CUPS allows any authenticated user to access and configure printers. If you want to restrict access, you can set access control rules. Within the web interface, access the "Administration" tab, click on "Server" in the left sidebar, and proceed to the "Access Control" section. From there, you can configure user and group permissions.

7. Configure printer options: Once the printers are added, fine-tune their settings. In the CUPS web interface, select the printer name and navigate to the "Administration" tab. Modify printer-specific options such as print quality, paper size, and default settings.
8. Restart the CUPS service: After making any changes to the CUPS configuration, it is advisable to restart the CUPS service to apply the modifications. Execute the following command:

```
sudo systemctl restart cups
```

Details: https://linuxconfig.org/linux-cups-tutorial-for-beginners