

Lab 12: FTP Configuration and Implementation using Packet Tracer

Theory

File Transfer Protocol (FTP) is a standard protocol used to transfer files between a client and server over a TCP/IP network, using port 21 for commands and port 20 for data transfer in active mode. It supports file operations like uploading, downloading, and managing directories through commands such as RETR, STOR, and DELE. FTP operates in active mode, where the server initiates the connection, or passive mode, where the client does. However, FTP lacks built-in encryption, making it less secure for sensitive data. Secure alternatives like FTPS and SFTP are commonly used for encrypted file transfers. FTP can also handle text files in ASCII mode and binary files without modification.

Key Concepts of FTP

1.Client-Server Model:

FTP operates on a client-server architecture where the client initiates file transfer requests, and the server responds by processing those commands. Clients can upload, download, or manage files on the server by communicating directly with it. This structure enables efficient file transfers and management over a network.

2.Ports:

FTP uses port 21 for the control connection to exchange commands and responses, while port 20 handles the data connection in active mode. In passive mode, the server assigns a dynamic port for data transfer. This setup facilitates both command communication and file transfer between client and server.

3.Active and Passive Modes:

In active mode, the client opens a port for data transfer, and the server connects to it. In passive mode, the server opens a port, and the client initiates the connection, making it more firewall-friendly. This helps FTP work smoothly in various network setups.

4.Authentication:

FTP supports anonymous mode, allowing users to access files without a password, and authenticated mode, where a username and password are required. This distinction helps balance ease of access with security controls. Authenticated mode is typically used for sensitive or restricted content.

5.FTP Commands:

Common FTP commands include **LIST** to view files, **RETR** to download, **STOR** to upload, **DELE** to delete, and **MKD** to create directories. These commands allow clients to manage files on the server efficiently. Each command ensures smooth file operations during FTP sessions.

6.Data Types:

FTP supports **ASCII mode** for text files, converting line endings between different systems, and **binary mode** for non-text files, preserving exact byte sequences. Using the correct mode ensures file integrity during transfers. This distinction is crucial for accurate file handling.

7.FTP Security:

FTPS and SFTP secure FTP connections by encrypting data during transfers. FTPS adds SSL/TLS encryption to traditional FTP, while SFTP uses SSH for secure file transfer. Both methods protect data from interception and unauthorized access, enhancing security.

Network Diagram

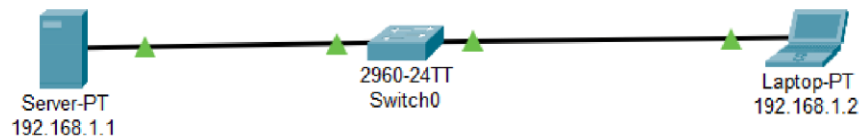


Fig: Network diagram

Configuring FTP Server and FTP Client

FTP Server Configuration

Step1: Click on server and go to IP configuration and set IP address and subnet mask.

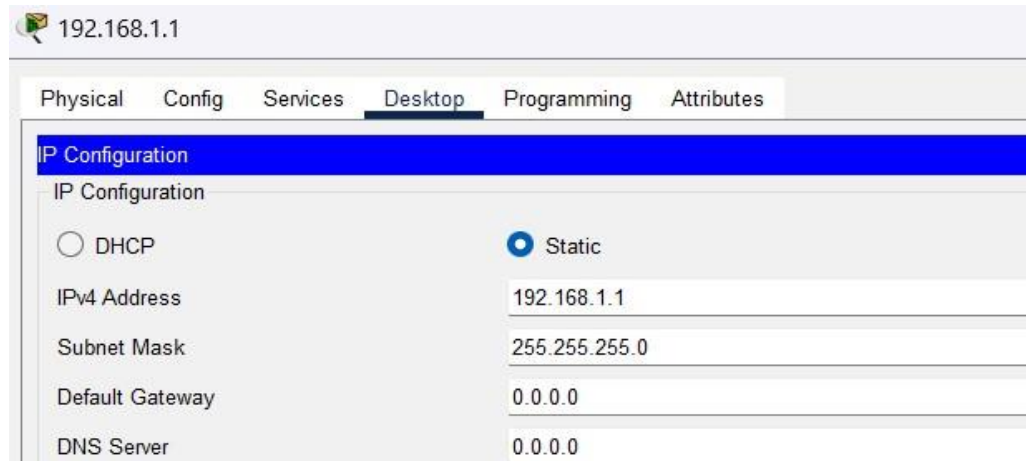


Fig: IP configuration on Sever

Step 2: Click on server, goto service, click ftp and click on ON button.

Step 3: Set username, password and tick write, read, delete, rename and list. Click add,

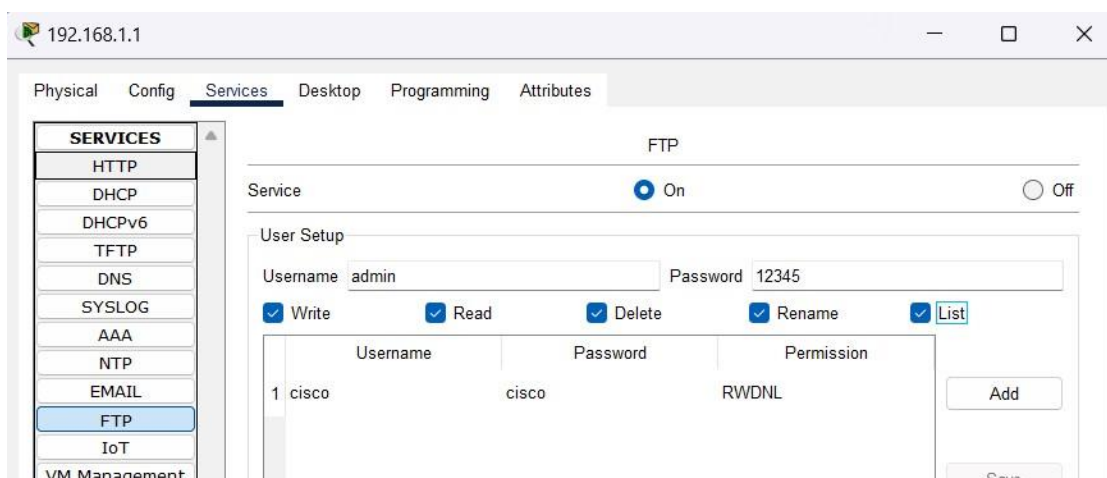


Fig: Server Configuration

Step 4: Go to desktop, click text editor write something and save the file as hello.txt.

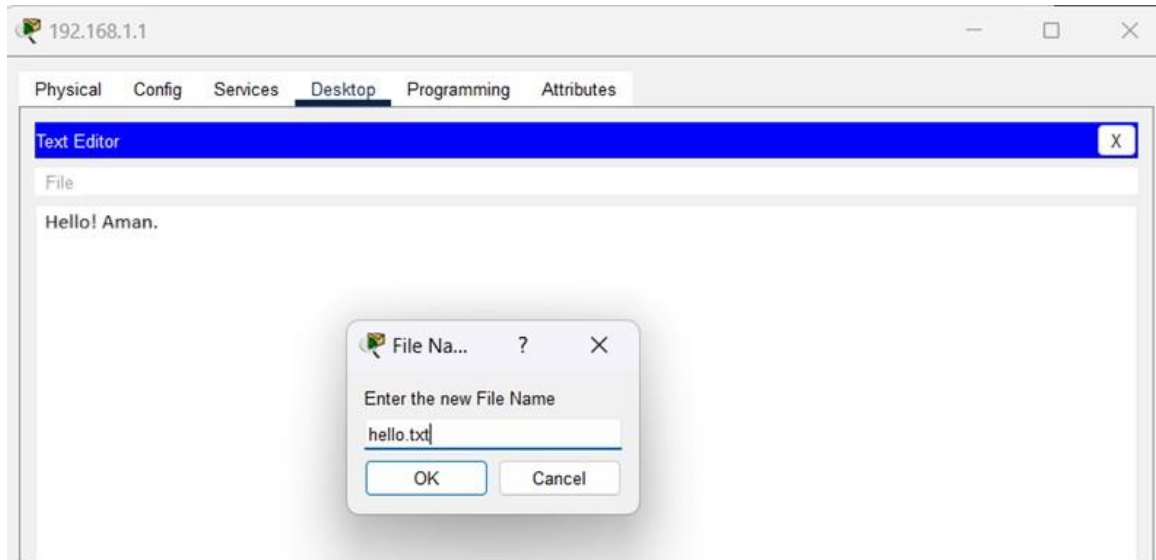


Fig: Creating a file name hello.txt

Step 5: In desktop, open command prompt and type dir command, we can see the file.

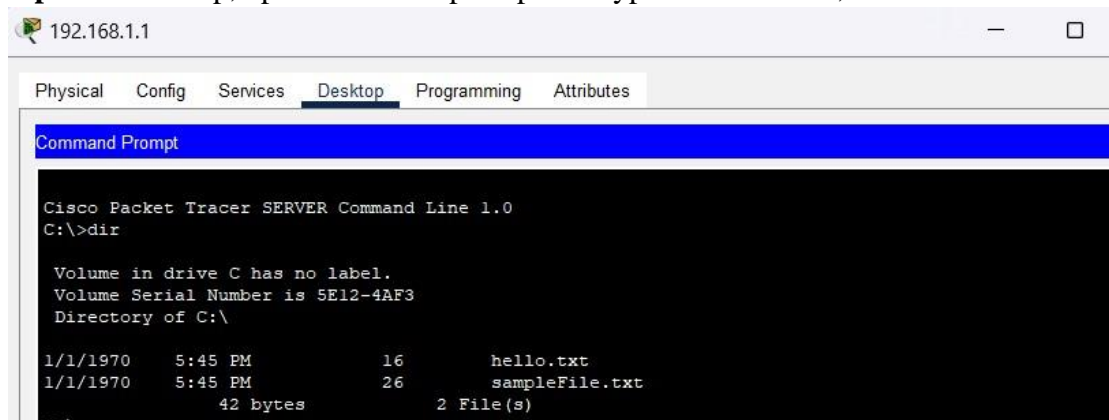


Fig: Using dir command to see file

FTP Client Configuration

Step 1: Click on pc and goto IP configuration and set IP address and subnet mask.

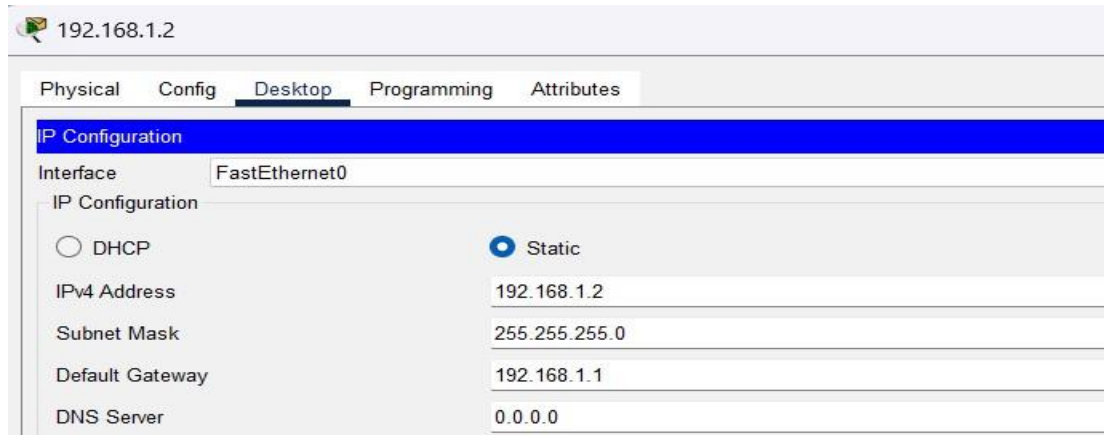


Fig: IP Configuration in Client

FTP Server Connection

In command prompt type command 'ftp 192.168.1.1' then insert username and password, we will be connected to ftp server.

```
C:\>ftp 192.168.1.1
Trying to connect...192.168.1.1
Connected to 192.168.1.1
220- Welcome to PT Ftp server
Username:admin
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

Fig: FTP Server Connection

TRANSFERRING FILE USING PUT COMMAND

Command: Ftp > put hello.txt

Fig: Transferring file using PUT command

RENAME FILE

Command: Ftp > rename hello.txt xyz.txt

```

ftp>
ftp>rename hello.txt xyz.txt

Renaming hello.txt
ftp>
[OK Renamed file successfully from hello.txt to xyz.txt]
ftp>

```

Fig: Renaming file

GET THE FILE AND SAVE THE COPY ON OUR MACHINE

Command: Ftp > get xyz.txt

```

ftp>
ftp>get xyz.txt

Reading file xyz.txt from 192.168.1.1:
File transfer in progress...

[Transfer complete - 16 bytes]

16 bytes copied in 0 secs
ftp>

```

Fig: Saving the copy of the file in pc

GO TO PC

Command: Ftp> quit ftp

```

ftp>
ftp>
ftp>quit ftp

221- Service closing control connection.
C:\>

```

Fig: Quit the FTP

DISPLAYING THE FILES Command: PC > dir

```

C:\>
C:\>dir

Volume in drive C has no label.
Volume Serial Number is 5E12-4AF3
Directory of C:\

1/1/1970    5:45 PM           16      hello.txt
1/1/1970    5:45 PM           26      sampleFile.txt
1/1/1970    5:45 PM           16      xyz.txt
               58 bytes          3 File(s)

C:\>

```

Fig: Displaying the files

Conclusion

In this lab, we effectively configured and implemented an FTP server and client using Cisco Packet Tracer. We established a seamless FTP connection between the client and server, facilitating various file operations such as transfers, renaming, and retrieval using commands like ``put``, ``get``, and ``rename``. Our tests of the upload and download functionalities showcased the proficiency of FTP in managing file transfers within a network, providing a comprehensive understanding of its operations and limitations. This hands-on experience underscored the practical applications and potential challenges associated with FTP in network environments.