Section 1.1.4 Computer Networks Fundamentals

FTP

{File Transfer Protocol}

Cracking OSCP: Your Roadmap to Ethical Hacking Success

YouTube: HackProKP – Kailash Parshad

Socials: HackProKP

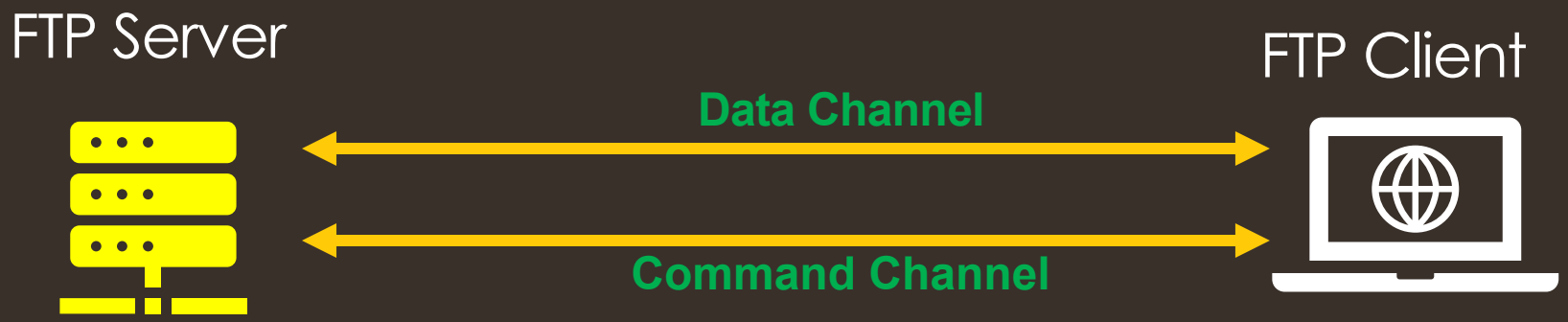
Github: https://github.com/at0m-b0mb/Cracking-OSCP-Your-Roadmap-to-Ethical-Hacking-Success

Complete Youtube Playlist: https://www.youtube.com/watch?v=MvkNbn8i2so&list=PLyrv3TPh3ejYNZipa0OIUvkdjHeUTRJ3J&index=1&t=0s

FTP

File Transfer Protocol is a standard network protocol used to transfer files from one host to another over a **TCP**-based network, such as the Internet. It is a client-server protocol, meaning there's a **client** (the user or software initiating the transfer) and a **server** (the remote system where files are stored).

FTP



FTP

Client-Server Architecture:

- **Client**: The client is the user's device or software that initiates the file transfer. It could be an FTP client program or a web browser with FTP capabilities.
- **Server**: The server is a remote system that stores the files and allows clients to connect to it for file transfer. It runs an FTP server software.

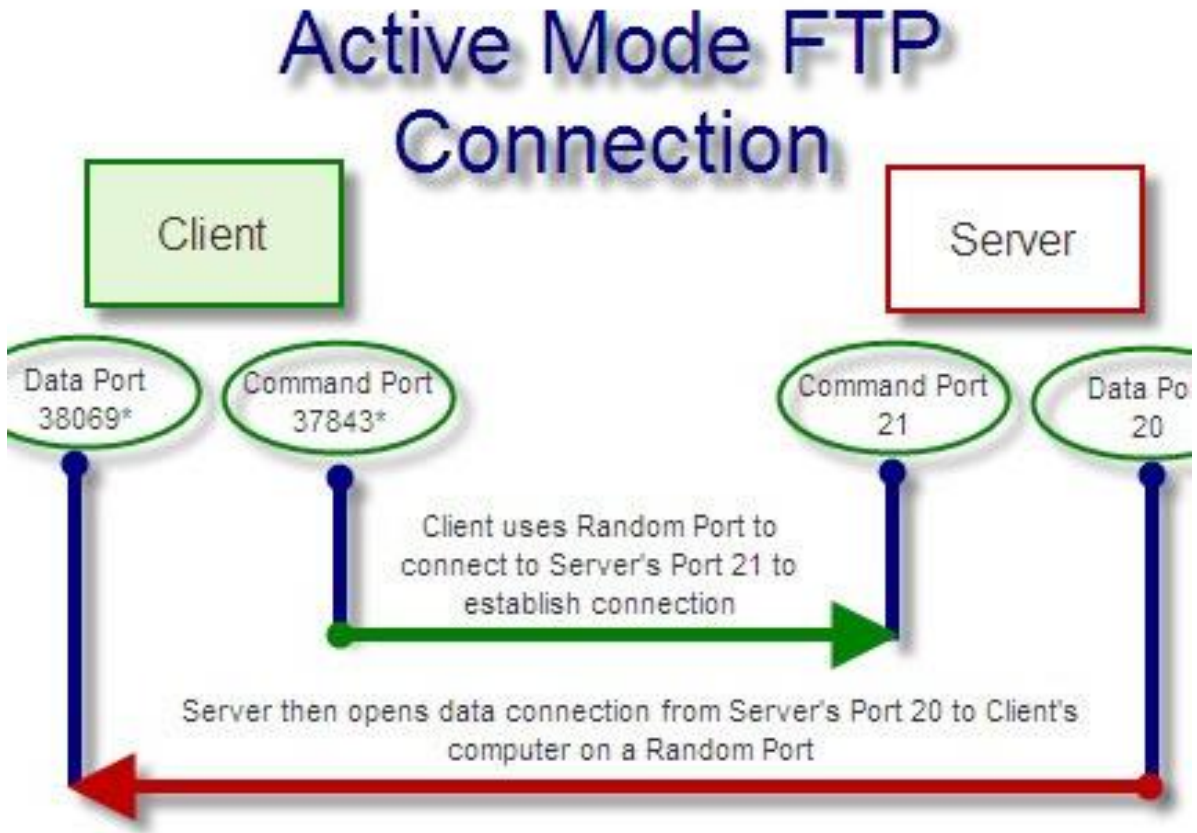
Modes of FTP:

- **Active Mode**: The client opens a random port for data transfer, and the server connects to this port. This mode can be problematic when the client is behind a *firewall* or *NAT (Network Address Translation)*.
- **Passive Mode**: The server takes the initiative to open a data channel for transferring files. more firewall-friendly as the client connects to the server for data transfer, avoiding issues with firewalls blocking incoming connections.

Active Mode:

- In Active mode, the client initiates a connection to the server for both the command channel (used for sending commands) and the data channel (used for transferring files).
- The client sends a PORT command to the server, indicating an IP address and port number to which the server should connect for the data transfer.
- The server then initiates a connection to the specified IP address and port number on the client for the data transfer.

(PORT mode)

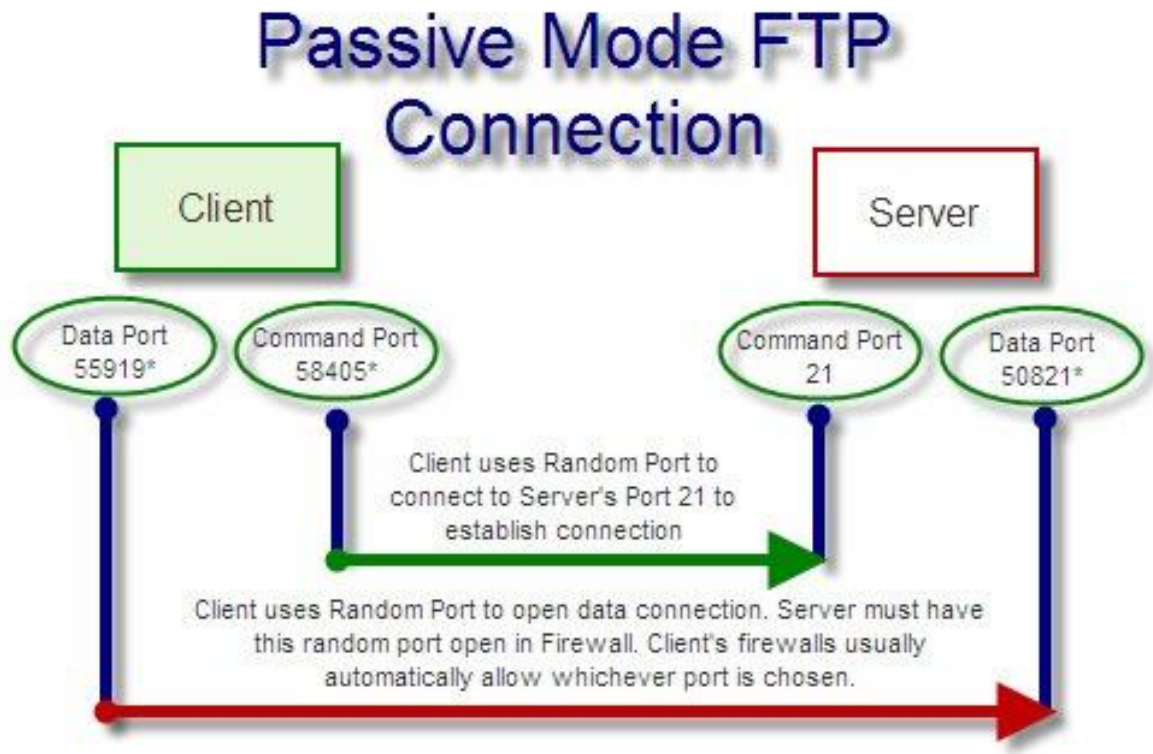


https://cloudzy.com/blog/ftp-active-vs-passive-which-one-is-right-to-your-needs/

Passive Mode:

- In Passive mode, the server takes the initiative to open a data channel for transferring files.
- The client sends a PASV command to the server, and the server responds with an IP address and port number to which the client should connect for the data transfer.
- The client then initiates a connection to the specified IP address and port number on the server for the data transfer.

(PASV mode)



https://cloudzy.com/blog/ftp-active-vs-passive-which-one-is-right-to-your-needs/

FTP Commands:

- **USER**: Specifies the username to log in.
- **PASS**: Specifies the password for the given username.
- **PWD**: Prints the current working directory.
- **CWD**: Changes the current working directory.
- **LIST** or **NLST**: Lists the files in the current directory.
- **RETR**: Retrieves a file from the server.
- **STOR**: Stores a file on the server.
- **QUIT**: Ends the FTP session.

Authentication:

- Users typically need to provide a valid username and password to log in to an FTP server.
- **Anonymous** FTP allows users to log in using a default username (usually "**anonymous**" or "**ftp**") generally without a password.
- Commonly used for **public file repositories**.

Data Transmission:

- ASCII and Binary Modes:
  1. FTP can operate in **ASCII** or **binary mode**.
  2. ASCII mode is used for **text files**, while binary mode is used for **non-text files** to ensure proper data **integrity**.

Security Concerns:

- FTP transmits data in **plain text** (This means that both the commands and the actual file data being transferred are not encrypted), making it susceptible to **eavesdropping**.
- **FTPS (FTP Secure)** and **SFTP (SSH File Transfer Protocol)** are secure alternatives that encrypt the data during transmission.

Security:

- **FTPS (FTP Secure):**
  - ✓ FTPS adds a layer of security by using SSL/TLS encryption for the control and data connections.
  - ✓ It can operate in implicit (990) or explicit (21) mode.
- **SFTP (SSH File Transfer Protocol):**
  - ✓ SFTP uses SSH for a secure connection.
  - ✓ It provides secure file transfer as well as additional features like remote file management and execution of remote commands.

FTPS (FTP Secure):

- **Implicit Mode (Port 990):**
  - ✓ In implicit mode, the secure connection is assumed right from the beginning.
  - ✓ The FTPS server on the server side is configured to listen for connections on port 990 for implicit SSL/TLS connections.
  - ✓ When a client connects to this port, it is expected to start the secure communication immediately.
  - ✓ Implicit mode is less common than explicit mode but is more straightforward for secure communication.
  - ✓ Example FTPS URL for implicit mode:

    ftps://ftp.example.com:990

FTPS (FTP Secure):

- **Explicit Mode (Port 990):**
  - ✓ In explicit mode, the FTPS server initially operates as a standard FTP server on the well-known port 21.
  - ✓ The client connects to the server on this port without any encryption.
  - ✓ After the initial connection is established, the client issues a command (usually AUTH TLS or AUTH SSL) to initiate the secure connection.
  - ✓ The server then responds, and the rest of the communication occurs over the encrypted channel.
  - ✓ Example FTPS URL for explicit mode:

    ftps://ftp.example.com:21

SFTP (SSH File Transfer Protocol):

- SFTP is a completely different protocol from FTP and runs over the Secure Shell (SSH) protocol.
- It encrypts both the control and data channels, providing a secure file transfer environment.
- SFTP is not to be confused with FTPS; they are distinct protocols.

Thank You!
😊 ❤️

Please Like 👍 and Subscribe 📖 to never miss a video 🎥

And to help me with the Algorithm 🤖

And It's completely Free!!! 💵

Love you guys ❤️