Fawn Write-up

Prepared by : g4utam262

Introduction

What is FTP?
According to definition on wikipidia:

The File Transfer Protocol (FTP) is a standard communication protocol used to transfer computer files from a server to a client on a computer network. FTP is built on a client server model architecture using separate control and data connections between the client and the server. FTP users may authenticate themselves with a clear-text sign-in protocol, generally in the form of a username and password. However, they can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password and encrypts the content, FTP is often secured with SSL/TLS (FTPS) or replaced with SSH File Transfer Protocol (SFTP).

Enumeration

Firstly, let us check if our VPN connection is established. Using the ping protocol can help with this since it is a low-overhead method of reaching the target to get a response, thus confirming our connection is established, and the target is reachable.
Low-overhead means that very little data is sent to the target by default, allowing us to quickly check the status of the connection without having to wait for a whole scan to complete beforehand. The ping protocol can be invoked from the terminal using the ping {target_IP} command, where {target_IP} is the IP address of your instance of the Fawn machine, as displayed on the Hack The Box webpage.

Note that this might not always work in a large-scale corporate environment, as firewalls usually have rules to prevent pinging between hosts, even in the same subnet (LAN), to avoid insider threats and discover other hosts and services.

```
┌──(root㉿FRIDAY)-[/home/G1]
└─# ping 10.129.74.245
PING 10.129.74.245 (10.129.74.245) 56(84) bytes of data.
64 bytes from 10.129.74.245: icmp_seq=1 ttl=62 time=271 ms
64 bytes from 10.129.74.245: icmp_seq=2 ttl=62 time=271 ms
64 bytes from 10.129.74.245: icmp_seq=3 ttl=62 time=271 ms
64 bytes from 10.129.74.245: icmp_seq=4 ttl=62 time=272 ms
64 bytes from 10.129.74.245: icmp_seq=5 ttl=62 time=267 ms
64 bytes from 10.129.74.245: icmp_seq=6 ttl=62 time=271 ms
64 bytes from 10.129.74.245: icmp_seq=7 ttl=62 time=273 ms
^C
--- 10.129.74.245 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6007ms
rtt min/avg/max/mdev = 266.770/270.883/272.672/1.778 ms
```

We can cancel the ping command by pressing CTRL+C .
We can see that responses are being received from the target host. This means that the host is reachable through the VPN tunnel we formed. We can now start scanning the open services on the host.

```
┌──(root㉿FRIDAY)-[/home/G1]
└─# nmap -sV 10.129.74.245
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-13 20:11 IST
Nmap scan report for 10.129.74.245
Host is up (0.39s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.45 seconds
```

Scanning using nmap, we can see that FTP service is open and running on port 21. The -sV switch stands for version detection. Using this switch will consequently make our scan take longer but will offer us more insight into the version of the service running on the previously detected port. This means that at a glance, we would be able to tell if the target is vulnerable due to running outdated software or if we need to dig deeper to find our attack vector.

Foothold

It is time we interacted with the target. In order to access the FTP service, we will use the `ftp` command on our own host. It's good practice to have a quick check that your `ftp` is up to date and installed properly. Running the command below will display the same output as pictured if your `ftp` service is installed. Otherwise, it will continue with the installation. The `-y` switch at the end of the command is used to accept the installation without interrupting the process to ask you if you'd like to proceed

```
┌──(root㉿FRIDAY)-[/home/G1]
└─# apt install ftp -y
ftp is already the newest version (20230507-2).
ftp set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
```

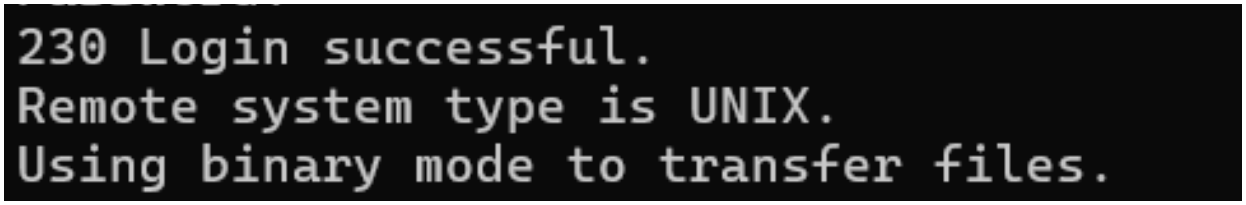
After it has done installing, you can run the `ftp -?` command to see what the service is capable of.

```
┌──(root㉿FRIDAY)-[/home/G1]
└─# ftp -?
usage: ftp [-46AadefginpRtVv] [-N NETRC] [-o OUTPUT] [-P PORT] [-q QUITTIME]
           [-r RETRY] [-s SRCADDR] [-T DIR,MAX[,INC]] [-x XFERSIZE]
           [[USER@]HOST [PORT]]
           [[USER@]HOST:[PATH][/]]
           [file:///PATH]
           [ftp://[USER[:PASSWORD]@]HOST[:PORT]/PATH[/][;type=TYPE]]
           [http://[USER[:PASSWORD]@]HOST[:PORT]/PATH]
           [https://[USER[:PASSWORD]@]HOST[:PORT]/PATH]
           ...
       ftp -u URL FILE ...
```

We can see that we can connect to the target host using the command below. This will initiate a request to authenticate on the FTP service running on the target, which will return a prompt back to our host:

```
┌──(root㉿FRIDAY)-[/home/G1]
└─# ftp 10.129.74.245
Connected to 10.129.74.245.
220 (vsFTPd 3.0.3)
Name (10.129.74.245:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

The prompt will ask us for the username we want to log in with. Here is where the magic happens. A typical misconfiguration for running FTP services allows an `anonymous` account to access the service like any other authenticated user. The `anonymous` username can be input when the prompt appears, followed by any password whatsoever since the service will disregard the password for this specific account.



Hitting `Enter` after filling in the password, we can see that we are logged in successfully. Our terminal changes in order to show us that we can now issue `ftp` commands.

Typing in the `help` command allows us to view which commands are available. You will be able to see this pattern with every script and service that you have access to. Typing either the `-h` , `--help` , or `help` commands will always issue a list of all the commands available to you as a user, with descriptions occasionally included. If you would like to learn about a specific command in more depth, you can use a different command: `man {commandName}` . However, for now, let us get back to our target.

```
ftp> help
Commands may be abbreviated.  Commands are:

!               delete          hash            mlsd            pdir            remopts         struct
$               dir             help            mlst            pls             rename          sunique
account         disconnect      idle            mode            pmlsd           reset           system
append          edit            image           modtime         preserve        restart         tenex
ascii           epsv            lcd             more            progress        rhelp           throttle
bell            epsv4           less            mput            prompt          rmdir           trace
binary          epsv6           lpage           mreget          proxy           rstatus         type
bye             exit            lpwd            msend           put             runique         umask
case            features        ls              newer           pwd             send            unset
cd              fget            macdef          nlist           quit            sendport        usage
cdup            form            mdelete         nmap            quote           set             user
chmod           ftp             mdir            ntrans          rate            site            verbose
close           gate            mget            open            rcvbuf          size            xferbuf
cr              get             mkdir           page            recv            sndbuf          ?
debug           glob            mls             passive         reget           status
```

Some of the commands listed here seem familiar to us. We already know how to use `ls` and `cd` . Let us issue the first command and view the contents of the folder.

```
ftp> ls
229 Entering Extended Passive Mode (|||60079|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0              32 Jun 04  2021 flag.txt
226 Directory send OK.
```

As you can notice from the output, the operation of FTP services also issues the status for the commands you are sending to the remote host. The meaning of status updates are as follows:

200 : PORT command successful. Consider using PASV.
150 : Here comes the directory listing.
226 : Directory send OK.

Now, we can proceed to download the flag.txt to our host. We can use the get command, followed by the name of the file we want to download.

```
ftp> get flag.txt
local: flag.txt remote: flag.txt
229 Entering Extended Passive Mode (|||56884|)
150 Opening BINARY mode data connection for flag.txt (32 bytes).
100% |***********************************************************
226 Transfer complete.
32 bytes received in 00:01 (0.02 KiB/s)
```

This will trigger the download of the file to the same directory you were in when you issued the ftp {machineIP} command. If we exit the FTP service, we will see the same file on our host now.

```
ftp> bye
221 Goodbye.

┌──(root㉿FRIDAY)-[/home/G1]
└─# ls
flag.txt  simple.py

┌──(root㉿FRIDAY)-[/home/G1]
└─# cat flag.txt
035db21c881520061c53e0536e44f815
```

We can now take the flag and submit it on the platform in order to own the box! Nice work!