File Hunter Write-up

What is FTP?

FTP (File Transfer Protocol) is a network protocol for transferring files over the internet. Developed in 1971, it allows users to upload files to or download files from servers. FTP is particularly useful for managing websites and sharing data.

FTP usually consists of two basic components: an FTP server and an FTP client. The FTP server is a centralized computer where files are stored and made available to the outside world. The FTP client is the software that allows users to connect to the FTP server and transfer files. Users usually connect to the FTP server using a username and password, but some servers may also allow **`anonymous access`**.

FTP continues to be popular due to its ease of use and wide compatibility. However, the fact that FTP is an unencrypted protocol carries some risks for data security. For this reason, SFTP (SSH File Transfer Protocol) or FTPS (FTP Secure) is often preferred as a secure alternative for transferring sensitive data.

The following command string is used to connect to an FTP server.

```
ftp <SERVER-IP-ADDRESS> -P <PORT-NUMBER>
```

Note: We don't need to specify a port number when connecting if the FTP service uses port 21 by default.

```
root💀hackerbox:~# ftp 10.0.0.88
Connected to 10.0.0.88.
220 (vsFTPd 3.0.3)
Name (10.0.0.88:hacker): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

In the example above we can see how we can connect **anonymously** to an FTP server.

After connecting to an FTP server, we use the **help** command to see the commands we can run. Below you can see some important commands.

```
help : Shows the commands that can be run and gives information about the commands.

get : Used to get a file.

dir : Lists the contents of a remote directory.

bye : End the FTP session and exit.
```

Information Gathering

Let's run a port scan for our target machine.

Task 1

To learn the version information of the services, we add the **-sV** parameter to our nmap command.

```
root💀hackerbox:~# nmap -sV 172.20.24.150
Starting Nmap 7.80 ( https://nmap.org ) at 2023-11-17 11:50 CST
Nmap scan report for 172.20.24.150
Host is up (0.0014s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 2.0.8 or later
MAC Address: 52:54:00:E7:28:8B (QEMU virtual NIC)
Service Info: Host: Welcome

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.99 seconds
```

Task 2

FTP stands for **File Transfer Protocol**, as explained earlier in this article.

System Access

Let's try to connect to our target machine with FTP.

```
root💀hackerbox:~# ftp 172.20.24.150
Connected to 172.20.24.150.
220 Welcome to anonymous Hackviser FTP service.
Name (172.20.24.150:root):
```

Task 3

When trying to connect to FTP, we see a welcome message without entering a username and password yet. As we can see from this message, let's try to connect to the FTP service **`anonymously`**.

```
root💀hackerbox:~# ftp 172.20.24.150
Connected to 172.20.24.150.
220 Welcome to anonymous Hackviser FTP service.
Name (172.20.24.150:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Yes, we have successfully connected to the server anonymously.

Anonymous FTP access allows anyone to access specific files on the server without requiring a username and password. This is often used for the distribution of open source software, public data or big files.

To connect anonymously to an FTP server, one connects to the server address via the FTP client, usually by typing "anonymous" in the username field and keeping the password field blank. Once the connection is established, users can view and download files from the server.

Anonymous FTP provides easy accessibility and makes it easier to share information. However, for security reasons, server owners also need to carefully manage the accessible content.

Task 4
We type the `help` command to display the commands we can run.

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

!         dir           mdelete   qc          site
$         disconnect    mdir      sendport    size
account   exit          mget      put         status
append    form          mkdir     pwd         struct
ascii     get           mls       quit        system
bell      glob          mode      quote       sunique
binary    hash          modtime   recv        tenex
bye       help          mput      reget       tick
case      idle          newer     rstatus     trace
cd        image         nmap      rhelp       type
cdup      ipany         nlist     rename      user
chmod     ipv4          ntrans    reset       umask
close     ipv6          open      restart     verbose
cr        lcd           prompt    rmdir       ?
delete    ls            passive   runique
debug     macdef        proxy     send
```

Task 5
Let's run the `ls` command to view the files on the FTP server we are connected to.

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp            25 Sep 08 08:07 userlist
226 Directory send OK.
```

Task 6
We can use the **`get`** command that we see in the command list to download files.

```
ftp> help get
get             receive file
```

Task 7
We can download the **`userlist`** file on the FTP server with the **`get`** command and close the FTP connection. Then we just need to read the contents of the file with the **`cat`** command.

```
ftp> get userlist
local: userlist remote: userlist
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for userlist (25 bytes).
226 Transfer complete.
25 bytes received in 0.00 secs (121.4630 kB/s)
ftp> bye
221 Goodbye.
root💀hackerbox:~# cat userlist
jack:hackviser
root:root
```

-

Congratulations 🙌

✨ You have successfully completed all tasks in this warmup.