OPEN RECRUITMENT
CYBERSECURITY OmahTI

BAYU PUTRA IBANA

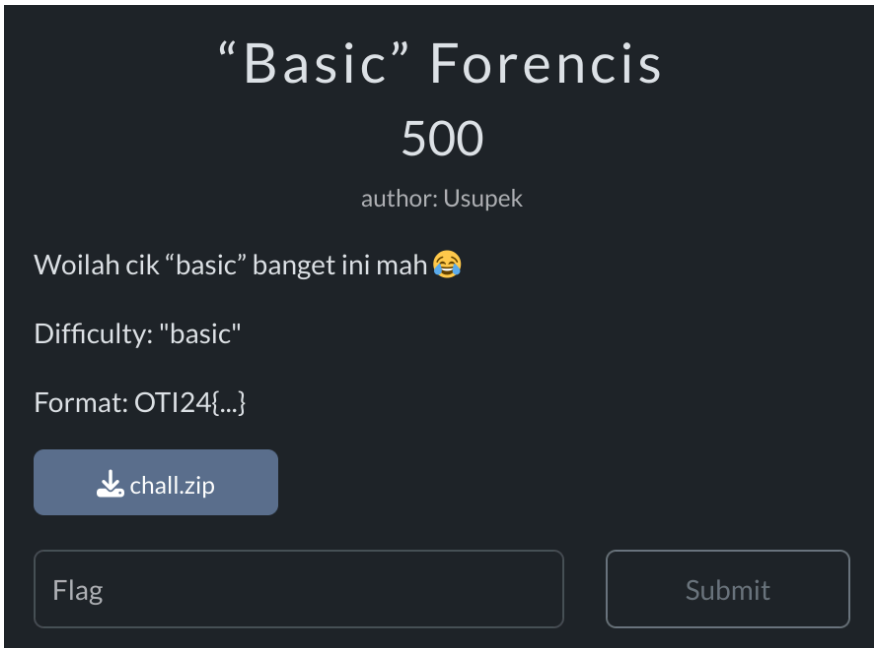
24/536830/PA/22776

Bonus Challenges - OmahTI CTF

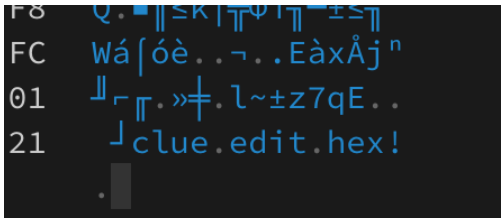
Free Flag

OTI24{W3lc0m3_t0_th3_g4m3}

“Basic” Forensics



I downloaded the files, then unzipped. There are 2 files, part1.7z and part2.png.
i opened the part1.7z in a hex editor and found:



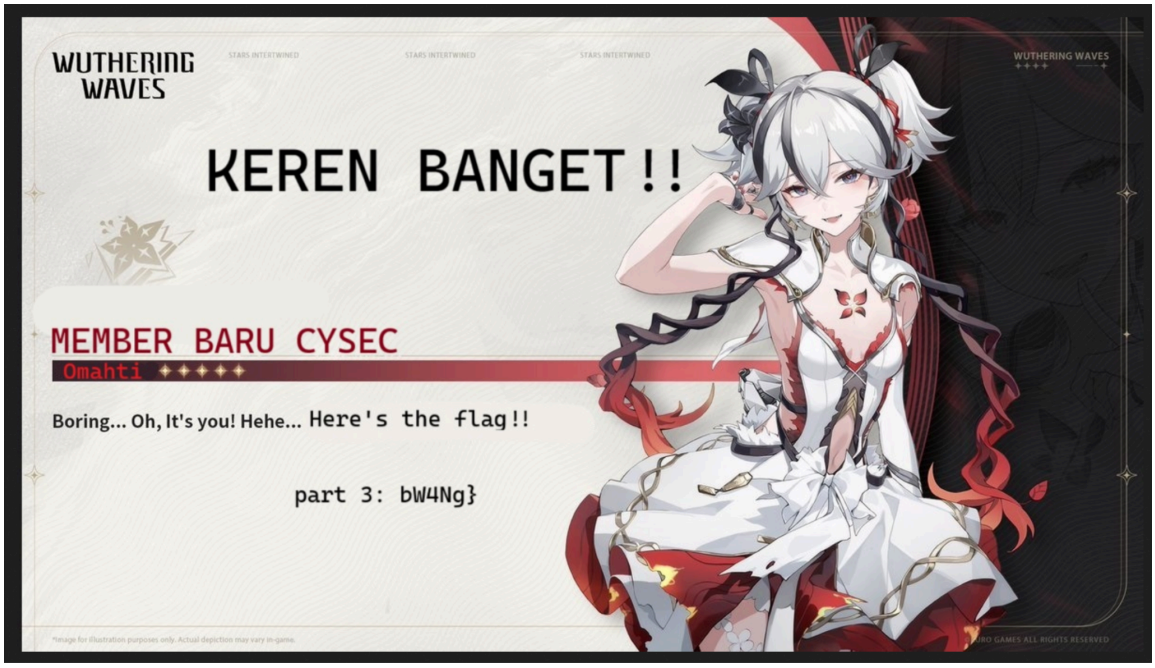
Using binwalk, i was able to find :

```
binwalk part1.7z

                              /home/hush/Downloads/chall/part1.7z
-----------------------------------------------------------------------------------------------------------------
                HEXADECIMAL                             DESCRIPTION
-----------------------------------------------------------------------------------------------------------------
                0x989A9A                                JPEG image, total size: 242408 bytes
-----------------------------------------------------------------------------------------------------------------
```

I'm going to try and export it.

```
[hush@arch chall]$ binwalk -e part1.7z

                                        /home/hush/Downloads/chall/extractions/part1.7z
----------------------------------------------------------------------------------------------------
DECIMAL                         HEXADECIMAL                     DESCRIPTION
----------------------------------------------------------------------------------------------------
10001050                        0x989A9A                        JPEG image, total size: 242408 bytes
----------------------------------------------------------------------------------------------------
[+] Extraction of jpeg data at offset 0x989A9A completed successfully
----------------------------------------------------------------------------------------------------
```

In the extraction folder, i found the flag in an image:

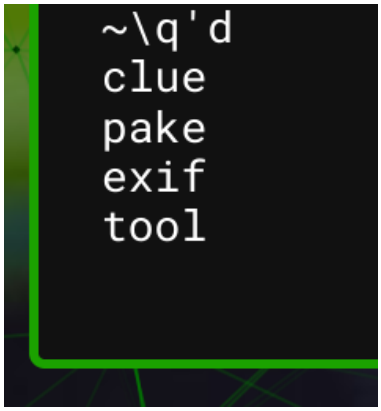


part 3 of the flag : **bW4Ng}**

Let's find the other flags, since the flag consists of multiple pieces.

I tried putting the part2.jpg to scan in aperisolve, and

In the string section, i found this:



```
[hush@arch chall]$ exiftool part2.jpg
ExifTool Version Number         : 13.03
File Name                       : part2.jpg
Directory                       : .
File Size                       : 1942 kB
File Modification Date/Time     : 1980:01:01 00:00:00+07:00
File Access Date/Time           : 2024:11:16 04:35:22+07:00
File Inode Change Date/Time     : 2024:11:16 04:35:21+07:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 120
Y Resolution                    : 120
Comment                         : VGhlIGZsYWcgaXMgSElEZEVuIHlvdSBzaG91bGQgU0VFSyBmb3IgaXQhCg==
Image Width                     : 2133
Image Height                    : 1600
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 2133x1600
Megapixels                      : 3.4
```

After installing and using exiftool to analyze the part2.jpg image, this is what I found.
The comment looks like it can be decoded.

Output

```
The flag is HIDdEn you should SEEK for it!
```

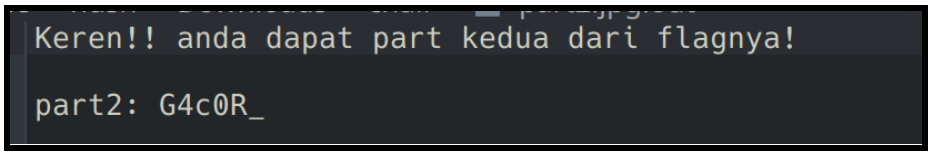
In cyberchef, I decoded this from Base64.
The capital letters spell out HIDE , SEEK, those might be a reference for tools to use,
I used a tool called StegSeek, according to the clue. It is a password cracking tool.
rockyou.txt is used together with it, which is a txt file containing passwords that the tool will brute force to find the passphrase.

```
[hush@arch chall]$ stegseek part2.jpg rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "iloveyou"
[i] Original filename: "part2.txt".
[i] Extracting to "part2.jpg.out".
```

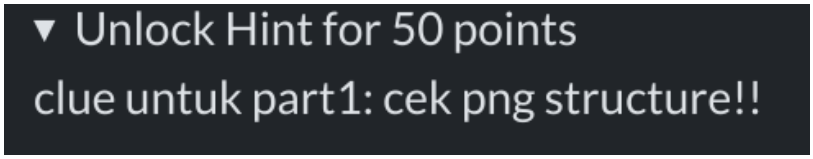
Now i can use the passphrase to use the StegHide tool, another tool to extract hidden files which requires a passphrase, so i entered the passphrase i found before:

```
[hush@arch chall]$ steghide extract -sf part2.jpg
Enter passphrase:
wrote extracted data to "part2.txt".
```

now let’s open part2.txt

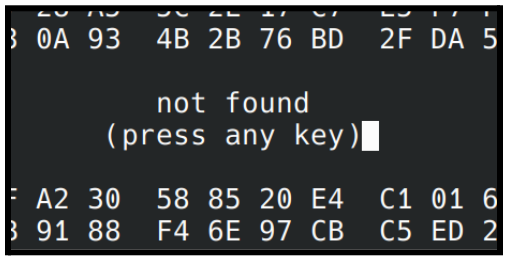


G4c0R_ , that's part2 of the flag. Now what's left is part1.

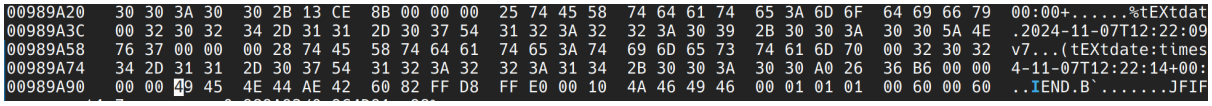


I didn't know what to do to find part1, so I unlocked the hint and it said to check png structure.

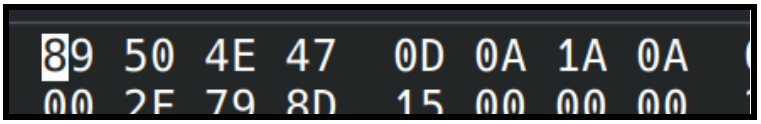
i tried searching for a png file header in hexadecimal, in hexedit,
89 50 4E 47 0D 0A 1A 0A, but it did not work



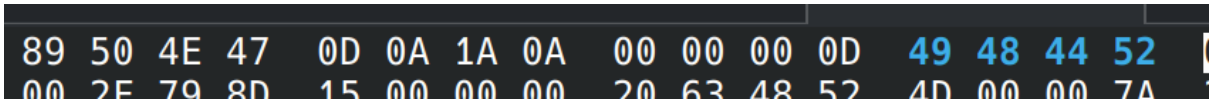
I realized that there are also chunks in PNG, such as IHDR, IDAT, IEND
so i tried searching the structure for each and found the IDAT and also IEND.



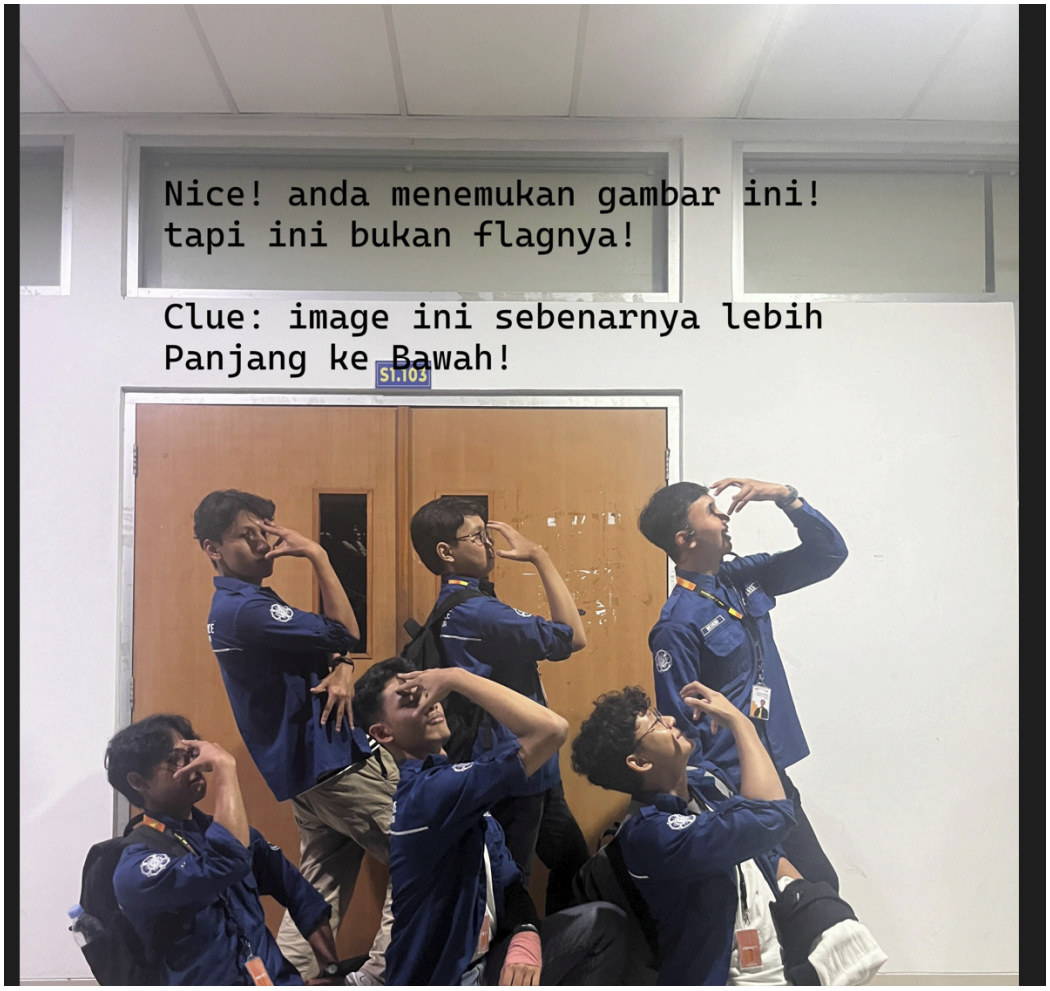
This means that the png file exists, just that something is missing, probably the header which is why I couldn't find it earlier. So I have to add it at the beginning of the file.



so i changed the first 8 bits, and saved changes.
Then I realized I forgot that I also needed to add the IHDR, which was also not found.

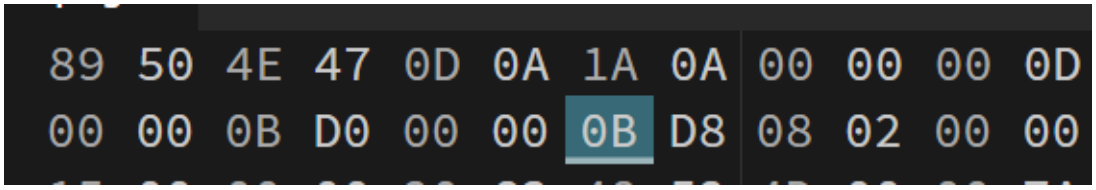


After saving, somehow my file was corrupted but managed to fix it.
here is the image:



it is not the flag yet, but the clue is that the image stretches down, so i have to probably edit the height.

In the hexdump, you need to edit the 21st to the 24th bytes of the png image for the height.



i googled several combinations for bigger sizes, and i tried to change it to 10000 pixels in height:

Exporting the new resized image, i finally got the 1st part of the flag:
OTI24{d4Mn_

Now we combine all three flags:

OTI24{d4Mn_G4c0R_bW4Ng}

😭

WH4T TH3 S1GM4 CH. 1

WH4T TH3 S1GM4 CH. 1

500

author: Mr. Vanum

Try rizzing your laptop

Difficulty: EASY

Format: OTI24{...}

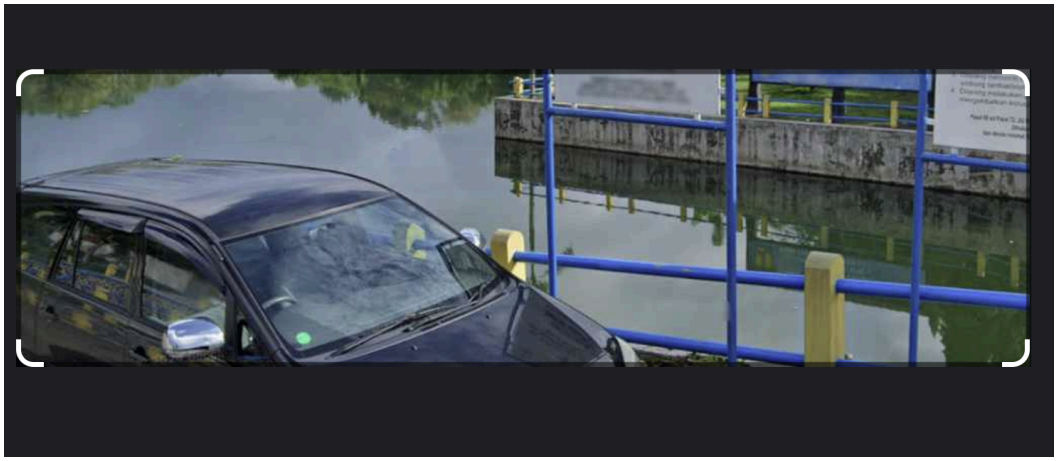
▼ Unlock Hint for 0 points
Di Indonesia

▼ Unlock Hint for 0 points
Deket air :v

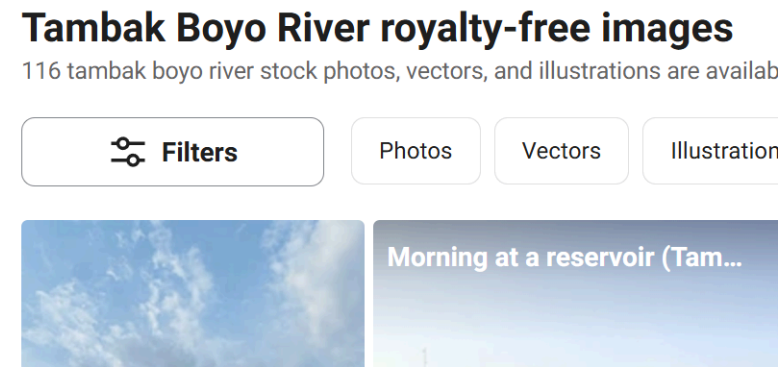
▼ Unlock Hint for 0 points
Look at the sign Look how they shine for you And everything you do
Yeah, they were all white and blue

From the downloaded files, we get 3 images.
there are 3 clues, Di Indonesia, deket air and look at the signs.
I need to find a location in Indonesia and probably enter it as a flag.

So I resorted to google lens to find the potential location.



when lens-ing this picture, i came across a link:



when i scrolled down,

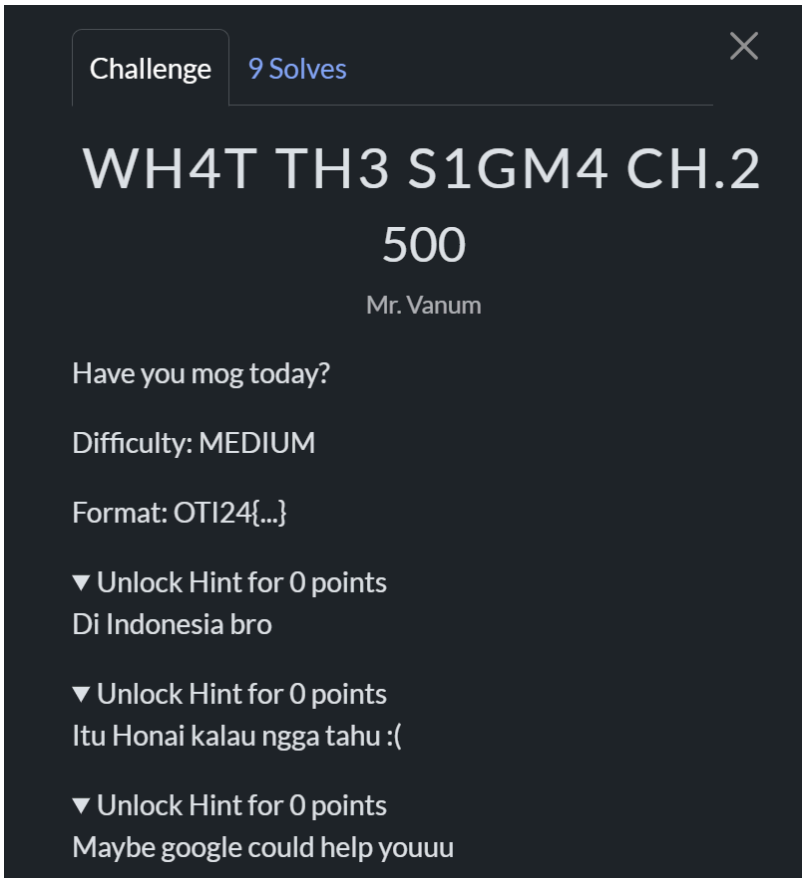
i found this image, which is really similar to the sign in one of the other pictures.

So i entered is as a flag:

OTI24{EMBUNG_TAMBAKBOYO}

which is the correct flag.

WH4T TH3 S1GM4 CH.2



In this challenge, we are given images again, this time i think i have to find a specific building:

-1000 4UR4

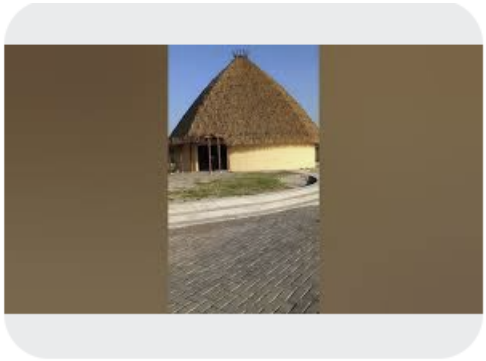
The Skibidi Psycho strikes again, bro, and this time it's Udin—our last hope against the oppressive Vanum Tax system—who's been obliterated. The cursed selfie Udin left behind is more blurry than a TikTok dance meme, but it's clear: this building, this tradition, is key to stopping Umar's rampage. But what building is it? A shrine to the true grindset? A

I'm going to try and use google lens again. The clues are that it's in Indonesia and the picture is a Honai, which is a traditional building and culture in Papua.

after searching, this does not look like it is in papua, because it is usually in a mountain area.

TikTok

Nyore disini makin syahdu karena sudah...



YouTube

Honai di Waduk.Cengklik#Boyo...

In google lens, I came across this and watched the tiktok and youtube video, because it looks similar to the image.

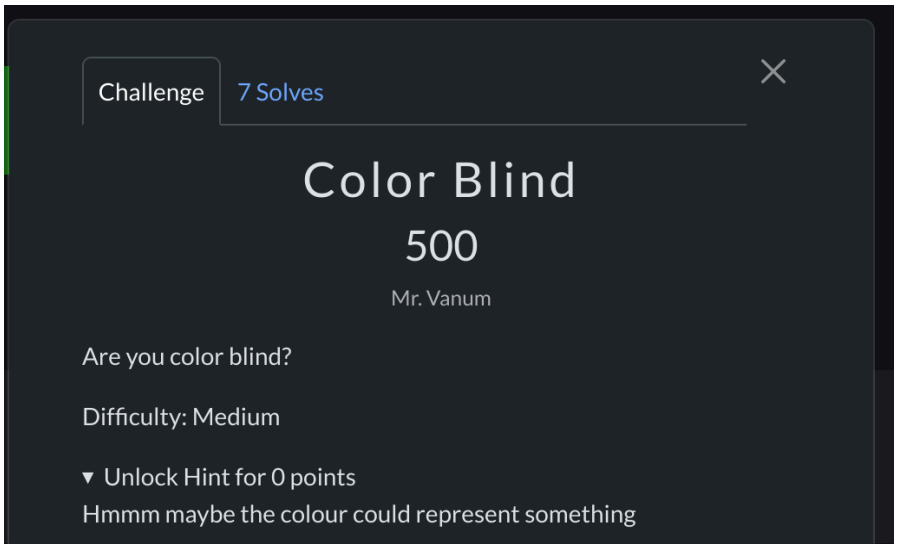
📍 RUMAH HONAI
Waduk Cengklik Sisi Barat Dukuh Ngalang ngalangan Desa Senting Kecamatan Sambi, Boyolali.

From the description of the tiktok video,
turns out it is a replica of Honai house in Boyolali, waduk cengklik.
then i entered the flag:

OTI24{HONAI_WADUK_CENGKLIK}

Which is the correct flag.

Color Blind



By downloading the file, I get an image that is only colors.
The hint suggests that the file could represent something.
First, i used exiftool to extract any information:

```
ExifTool Version Number         : 13.03
File Name                       : colors.png
Directory                       : .
File Size                       : 244 bytes
File Modification Date/Time     : 2024:11:16 18:01:01+07:00
File Access Date/Time           : 2024:11:16 18:01:02+07:00
File Inode Change Date/Time     : 2024:11:16 18:01:01+07:00
File Permissions                : -rw-r--r--
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Image Width                     : 65
Image Height                    : 1
Bit Depth                       : 8
Color Type                      : RGB
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Image Size                      : 65x1
Megapixels                      : 0.000065
```

The file is RGB color type. and it has a width of 65 and height of 1,

Using magick,

```
$ magick colors.png txt:- > pixels.txt
```

I converted the image to pixels and stored it in a text file.

```
# ImageMagick pixel enumeration: 65,1,0,255,srgb
0,0: (98,108,97)  #626C61  srgb(98,108,97)
1,0: (104,98,108)  #68626C  srgb(104,98,108)
2,0: (97,104,98)  #616862  srgb(97,104,98)
3,0: (108,97,104)  #6C6168  srgb(108,97,104)
4,0: (95,104,101)  #5F6865  srgb(95,104,101)
5,0: (108,108,111)  #6C6C6F  srgb(108,108,111)
6,0: (95,104,111)  #5F686F  srgb(95,104,111)
7,0: (119,95,97)  #775F61  srgb(119,95,97)
8,0: (114,101,95)  #72655F  srgb(114,101,95)
9,0: (121,111,117)  #796F75  srgb(121,111,117)
10,0: (95,116,111)  #5F746F  srgb(95,116,111)
11,0: (100,97,121)  #646179  srgb(100,97,121)
12,0: (95,105,95)  #5F695F  srgb(95,105,95)
13,0: (104,111,112)  #686F70  srgb(104,111,112)
14,0: (101,95,121)  #655F79  srgb(101,95,121)
15,0: (111,117,95)  #6F755F  srgb(111,117,95)
16,0: (97,114,101)  #617265  srgb(97,114,101)
17,0: (95,110,111)  #5F6E6F  srgb(95,110,111)
18,0: (116,95,100)  #745F64  srgb(116,95,100)
19,0: (111,105,110)  #6F696E  srgb(111,105,110)
```

The rgb values, which are in brackets can potentially be translated to readable text.
I then extracted only the rgb values :

```
$ grep -oP '\(\d+,\d+,\d+\)' pixels.txt > rgb.txt
```

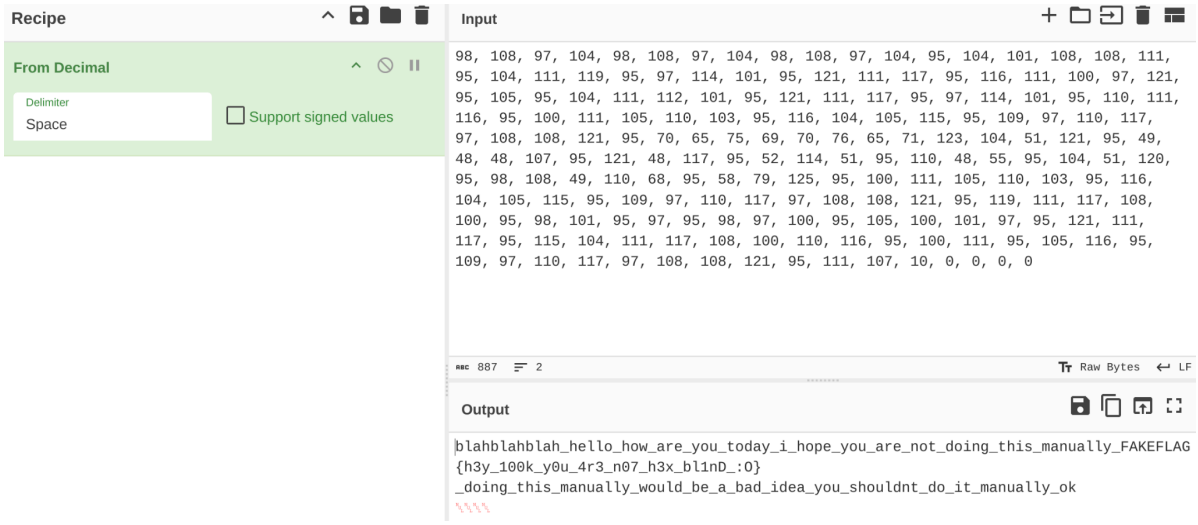
```
(98,108,97)
(98,108,97)
(104,98,108)
(104,98,108)
(97,104,98)
(97,104,98)
(108,97,104)
(108,97,104)
(95,104,101)
(95,104,101)
(108,108,111)
```

Next, I want to decode these values from decimal.
I will need to remove the parentheses and apply spacing for the numbers. I will also modify it so that there are no duplicates in the rgb values.

I used chatgpt to edit the text file..

```
98, 108, 97, 104, 98, 108, 97, 104, 98, 108, 97, 104, 95, 104, 101, 108, 108, 111, 95,
104, 111, 119, 95, 97, 114, 101, 95, 121, 111, 117, 95, 116, 111, 100, 97, 121, 95, 105,
95, 104, 111, 112, 101, 95, 121, 111, 117, 95, 97, 114, 101, 95, 110, 111, 116, 95, 100,
111, 105, 110, 103, 95, 116, 104, 105, 115, 95, 109, 97, 110, 117, 97, 108, 108, 121,
95, 70, 65, 75, 69, 70, 76, 65, 71, 123, 104, 51, 121, 95, 49, 48, 48, 107, 95, 121, 48,
117, 95, 52, 114, 51, 95, 110, 48, 55, 95, 104, 51, 120, 95, 98, 108, 49, 110, 68, 95,
58, 79, 125, 95, 100, 111, 105, 110, 103, 95, 116, 104, 105, 115, 95, 109, 97, 110, 117,
97, 108, 108, 121, 95, 119, 111, 117, 108, 100, 95, 98, 101, 95, 97, 95, 98, 97, 100,
95, 105, 100, 101, 97, 95, 121, 111, 117, 95, 115, 104, 111, 117, 108, 100, 110, 116,
95, 100, 111, 95, 105, 116, 95, 109, 97, 110, 117, 97, 108, 108, 121, 95, 111, 107, 10,
0, 0, 0, 0
```

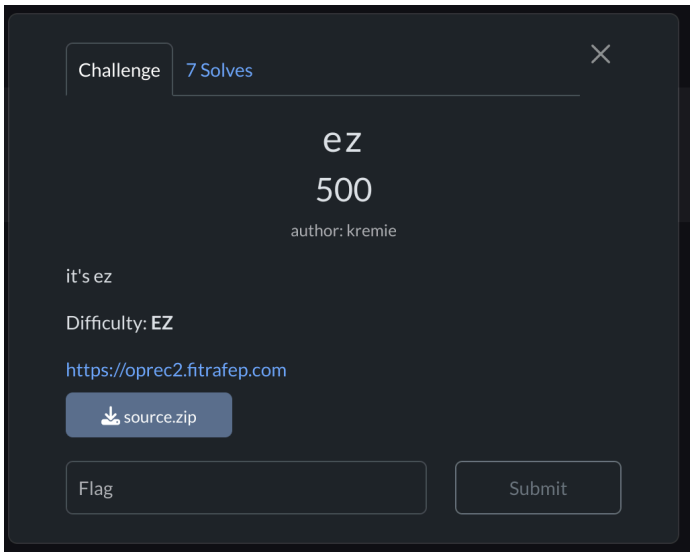
Then I copied and pasted it to cyberchef, and decoded it from decimal.



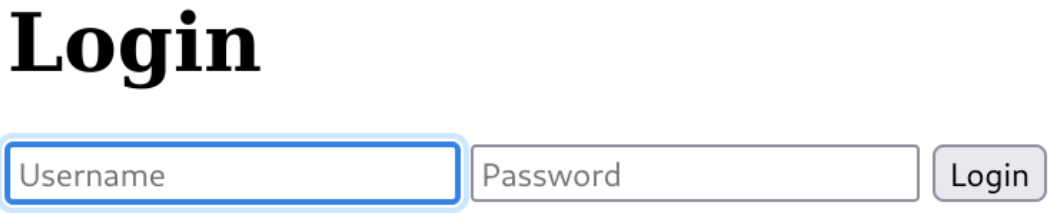
FAKEFLAG{h3y_100k_y0u_4r3_n07_h3x_bl1nD_:O}
the corrected flag:

OTI24{h3y_100k_y0u_4r3_n07_h3x_bl1nD_:O}

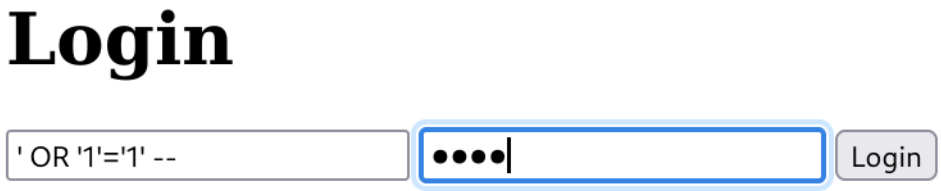
ez



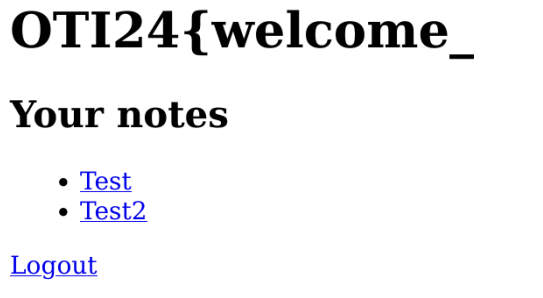
In this challenge, there is a website and source code.



There is a simple login page, let's try a sql injection method to bypass the login page.



the password can be anything



It worked, we got a part of the flag: **OTI24{welcome_**

there is more stuff here, let’s look at it.

when the test and test2 is clicked, there is not much to be found

Notes

- Test

home Logout

Notes

- Test2

home Logout

However if you look at the link:

this is note 1

this is note 2
lets try changing the url to ‘2’

Notes

- kidz_it's_ez_right}

home Logout

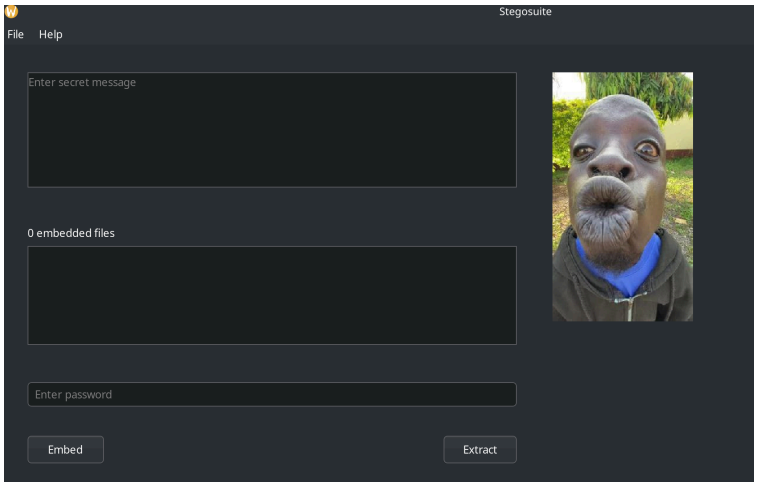
we got another part of the flag.

so we got the complete flag:

OTI24{welcome_kidz_it’s_ez_right}

A LONG JOURNEY

In this challenge, the flag is hidden to 2 halves. the 1st in the picture, 2nd in the author profiles. we can also download a suprise.png
Just like the 1st hint said, lets try using stegosuite.



let's enter the password , which is the slogan for OmahTI, we make it for everyone but without spaces like the level description said.

There is an embedded file **kiwkiw.txt,** it is extracted. let's open it

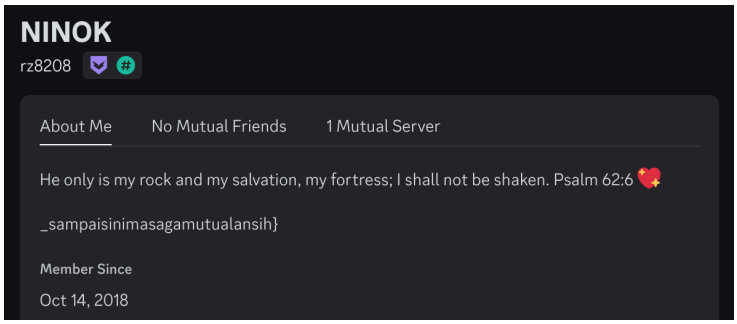
```
1   OTI24{D1C1UM_OT1_CH417 (1ST FLAG)
```

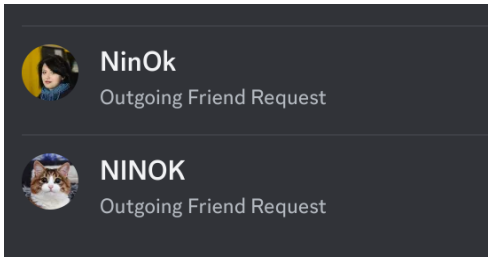
OTI24{D1C1UM_OT1_CH417
That's the 1st flag, let's find the second one.

To get the second half of the flag, You have to be a pro stalker. All I know is that the Author is very Active on 2 Social media's 🧐🤫

the author is NINOK, or Nino, so let's look at his social media,
in instagram:

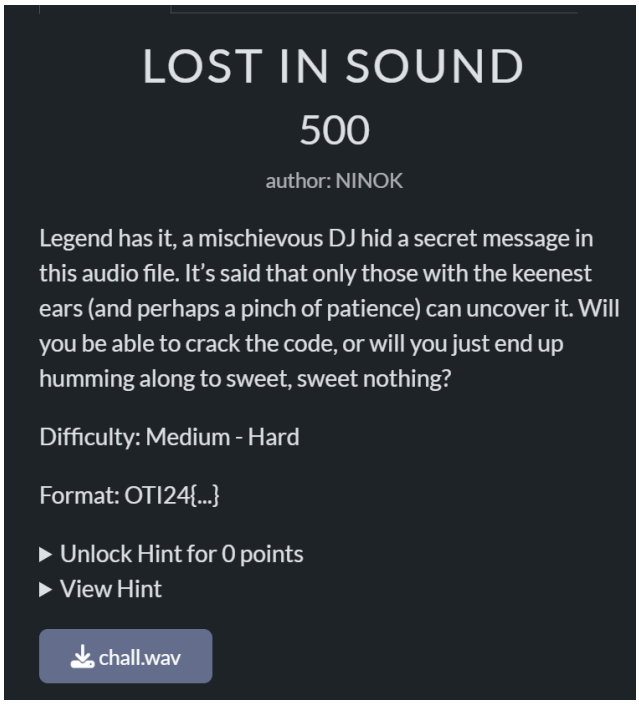
I clicked on the accounts on the bottom and found this, DC could mean discord so I will go there and add as a friend for the usernames mentioned.



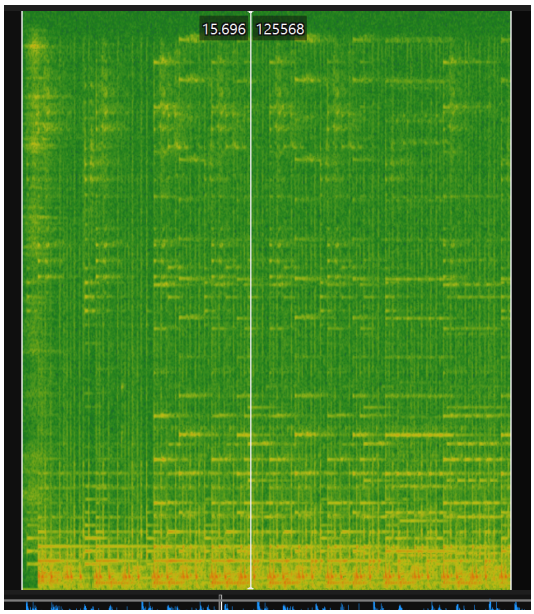


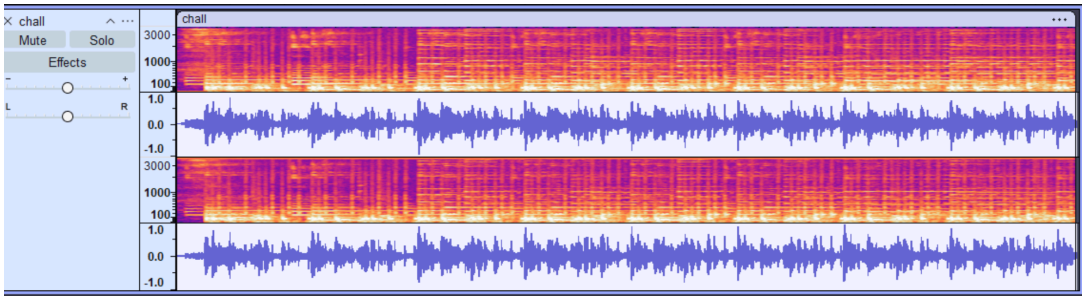
In the bio for NINOK, I found the 2nd part of the flag !
_sampaisinimasagamutualansih
Combine the final flag:
OTI24{D1C1UM_OT1_CH417_sampaisinimasagamutualansih}

LOST IN SOUND



In this challenge, we are given an audio file.
I tried to use tools like Sonic Visualizer and Audacity, to analyze their spectrograms, but I did not find anything.





I unlocked the second hint,

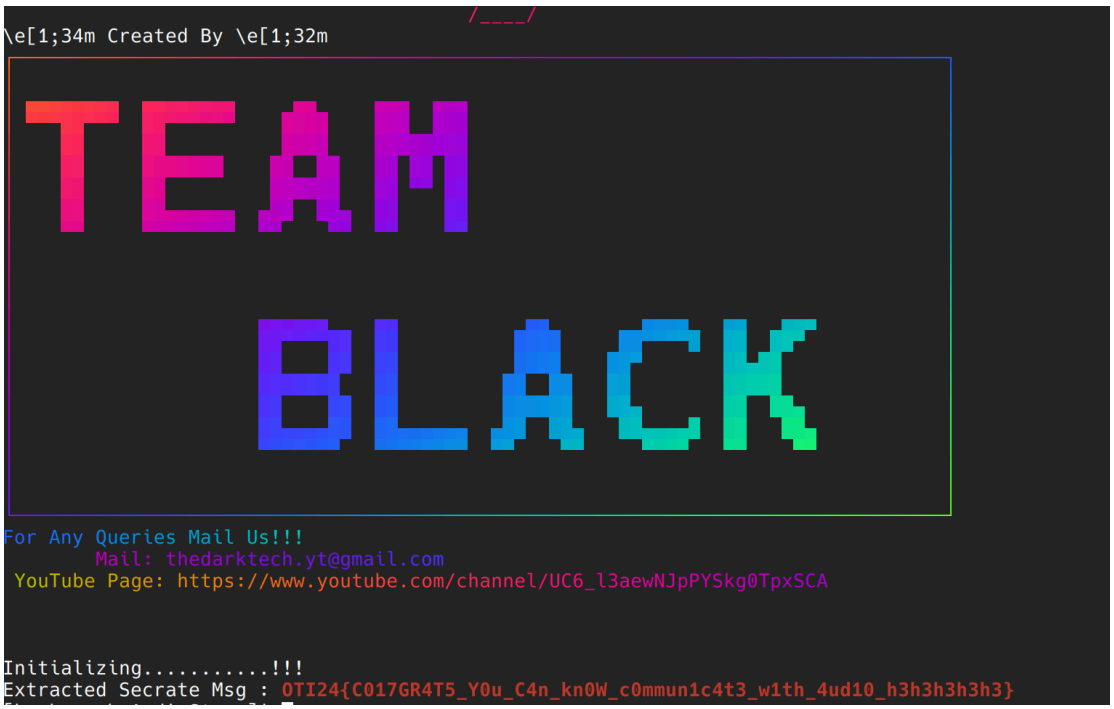
▼ Unlock Hint for 50 points
try to use this tool: https://github.com/thedarktech/AudioStego.git

```
]$ git clone https://github.com/thedarktech/AudioStego.git
```

After installing the tool,

```
]$ cd AudioStego
```

cd into it, then use this command: python3 ExtractMsg.py -f chall.wav
Also make sure the chall.wav is also in the AudioStego directory.
ExtractMsg is a built-in function to extract the msg from the audio file.



Then we get the flag:

OTI24{C017GR4T5_Y0u_C4n_kn0W_c0mmun1c4t3_w1th_4ud10_h3h3h3h3h3}