root@localhost:~$ echo "b477l3 0f l337"

BATTLE_OF_1337

CTF 2022

BATTLE OF 1337 OFFICIAL WRITEUP

Writeup By : asylumdx

TABLE OF CONTENT

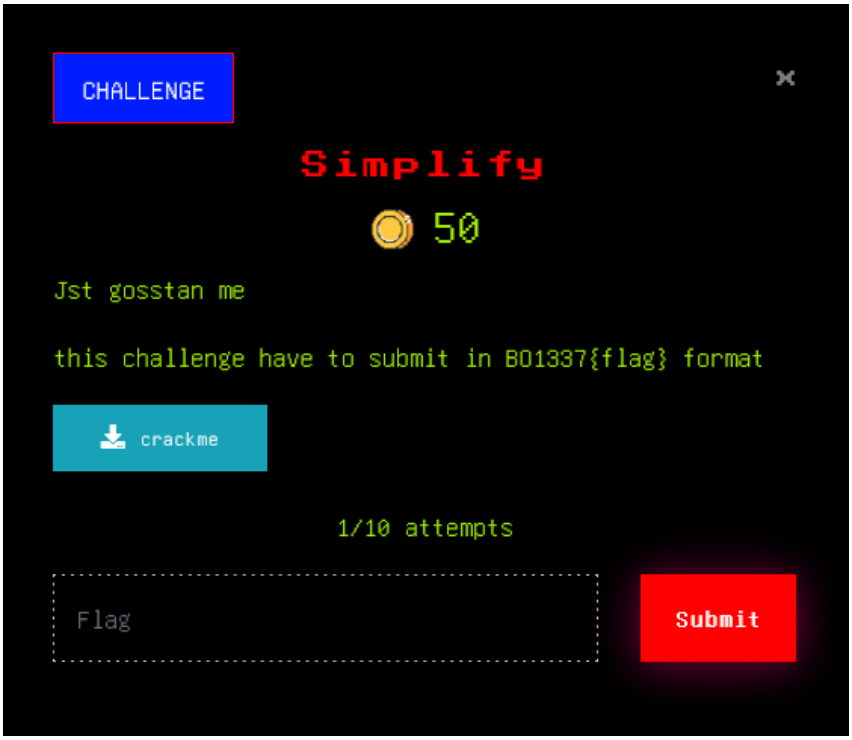
SOLVES

Solves

Challenge	Category	Value	Time
Back To The Future	Osint	50	July 16th, 10:12:10 PM
1Gram	Osint	100	July 16th, 10:18:12 PM
Break The Storage	Web	50	July 16th, 10:33:45 PM
Cat-Dalmantion	Web	50	July 16th, 11:01:21 PM
Rayquaza	Misc	50	July 17th, 12:49:40 AM
GunTher	Misc	100	July 17th, 1:52:58 AM
Simplify	Reverse Engineering	50	July 17th, 4:27:57 AM
Semerah Padi	Net	100	July 17th, 6:39:42 AM
Streamline	Net	176	July 17th, 7:24:09 AM
Snap	Osint	100	July 17th, 7:49:27 AM
Sheng Xiao	Misc	100	July 17th, 5:27:41 PM
Darchrow	Misc	100	July 17th, 8:06:09 PM
Heihawru	Misc	100	July 17th, 8:51:33 PM
RedPoint	Misc	443	July 17th, 9:10:21 PM

REVERSE ENGINEERING

SIMPLIFY

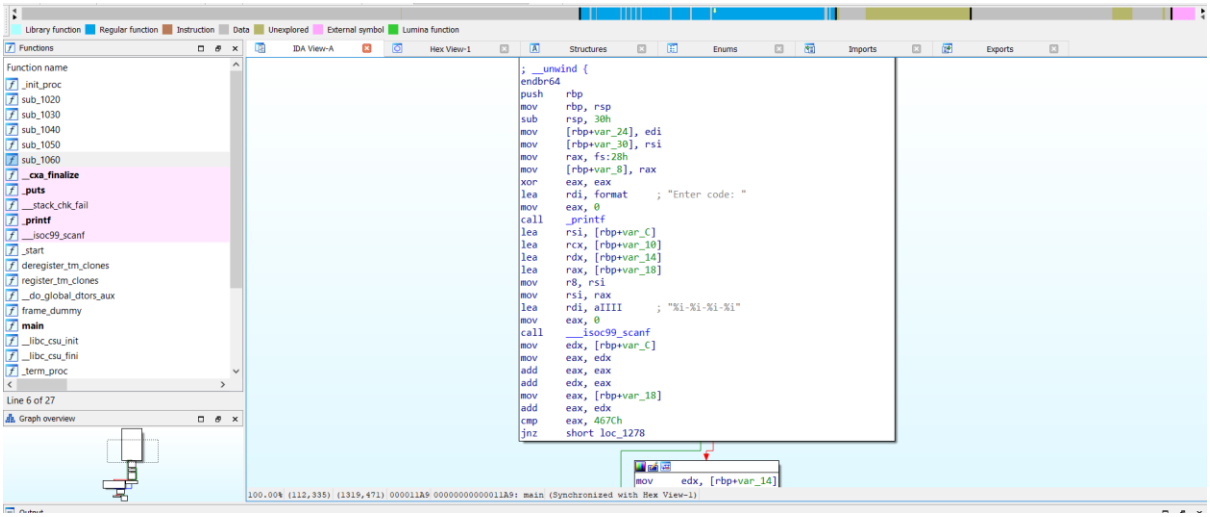


First, I checked the filetype and it turn out to be elf64 binary. Tried running and it ask for code.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# file crackme
crackme: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x
86-64.so.2, BuildID[sha1]=f75eeac944f72065eacfdcc28ebfdcba4d02d639, for GNU/Linux 3.2.0, not stripped
```

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ./crackme
Enter code: 313131
Wrong code ..
```

Then I open it in IDA to see the main function.



Decompile the function using hex ray by pressing f5.

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  unsigned int v4; // [rsp+18h] [rbp-18h] BYREF
  unsigned int v5; // [rsp+1Ch] [rbp-14h] BYREF
  unsigned int v6; // [rsp+20h] [rbp-10h] BYREF
  unsigned int v7; // [rsp+24h] [rbp-Ch] BYREF
  unsigned __int64 v8; // [rsp+28h] [rbp-8h]

  v8 = __readfsqword(0x28u);
  printf("Enter code: ");
  __isoc99_scanf("%i-%i-%i-%i", &v4, &v5, &v6, &v7);
  if ( 3 * v7 + v4 == 18044 && 3 * v6 * v5 == 5174190 && v4 == 1010 && v7 + 49363 * v6 == 63683948 )
    printf("Correct code! The flag is %i-%i-%i-%i\n", v4, v5, v6, v7);
  else
    puts("Wrong code..");
  return 0;
}
```

From here we can see how four variable that is being used for checking the code.

3 * v7 + v4 == 18044 && 3 * v6 * v5 == 5174190 && v4 == 1010 && v7 + 49363 * v6 == 63683948

Since we have v4 value, we can just calculate other value using math.

V4=1010

V5=1337

V6=1290

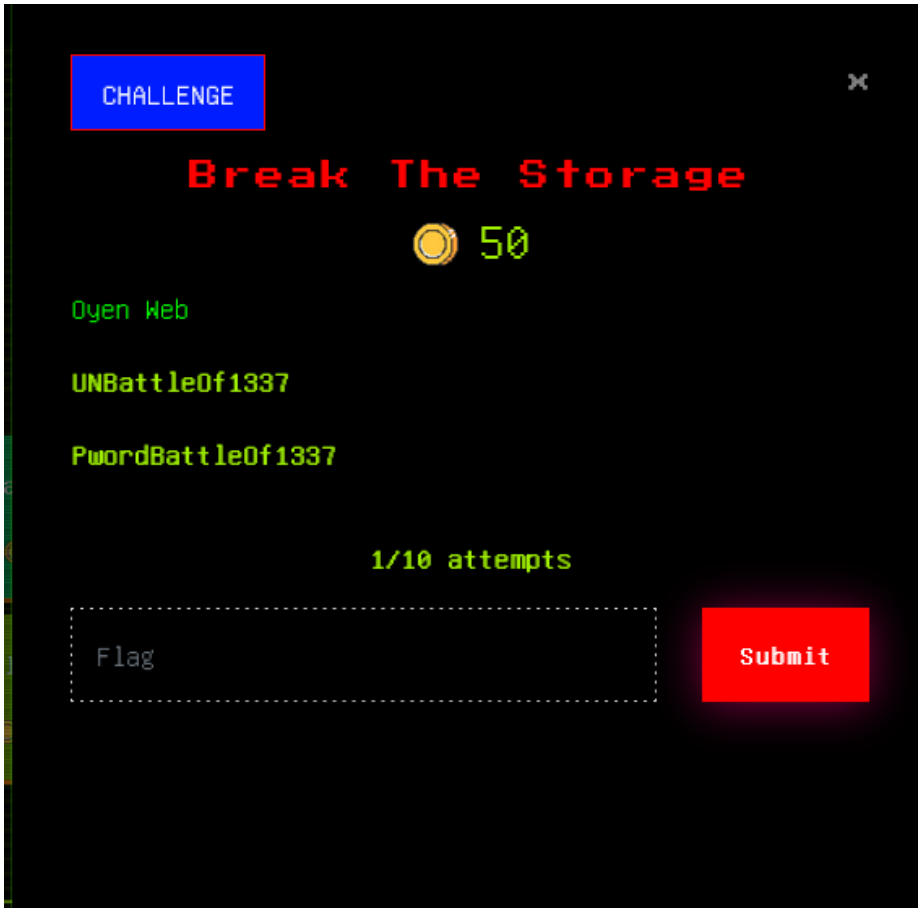
V7=5678

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# ./crackme
Enter code: 1010-1337-1290-5678
Correct code! The flag is 1010-1337-1290-5678
```

BO1337{1010-1337-1290-5678}

WEB

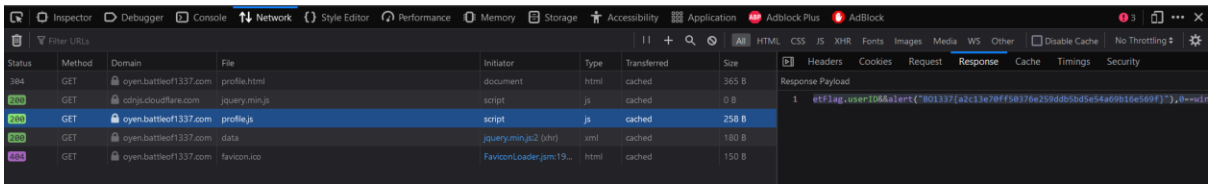
BREAK THE STORAGE



First, login using the given credentials.

Username: **BattleOf1337**

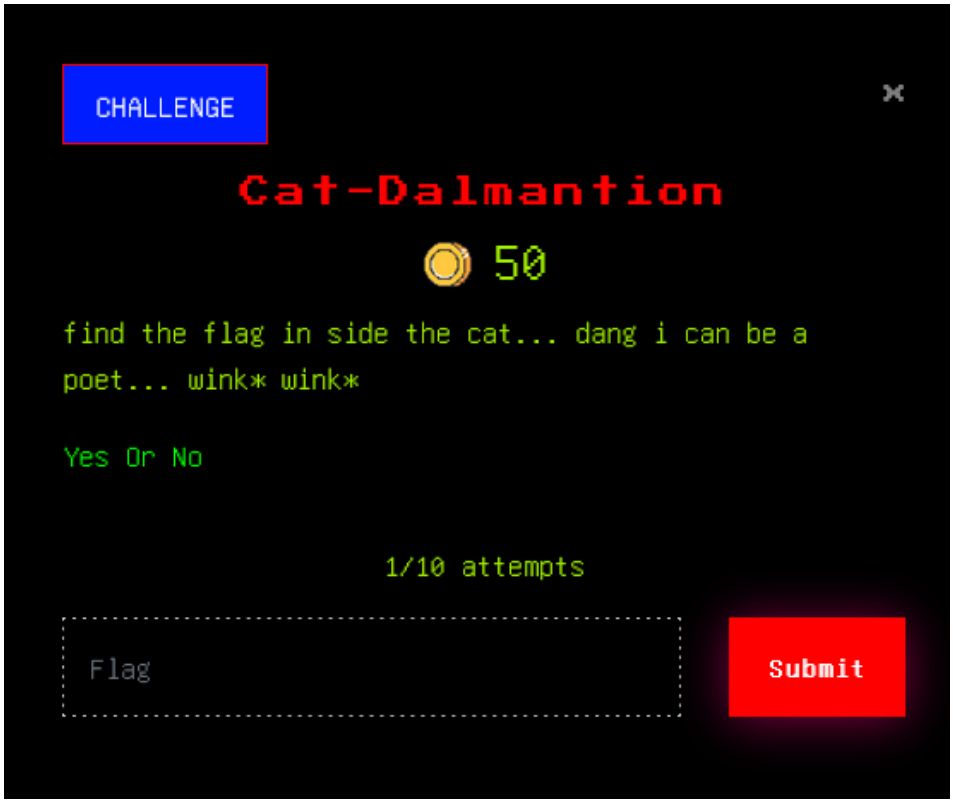
Password: **BattleOf1337**



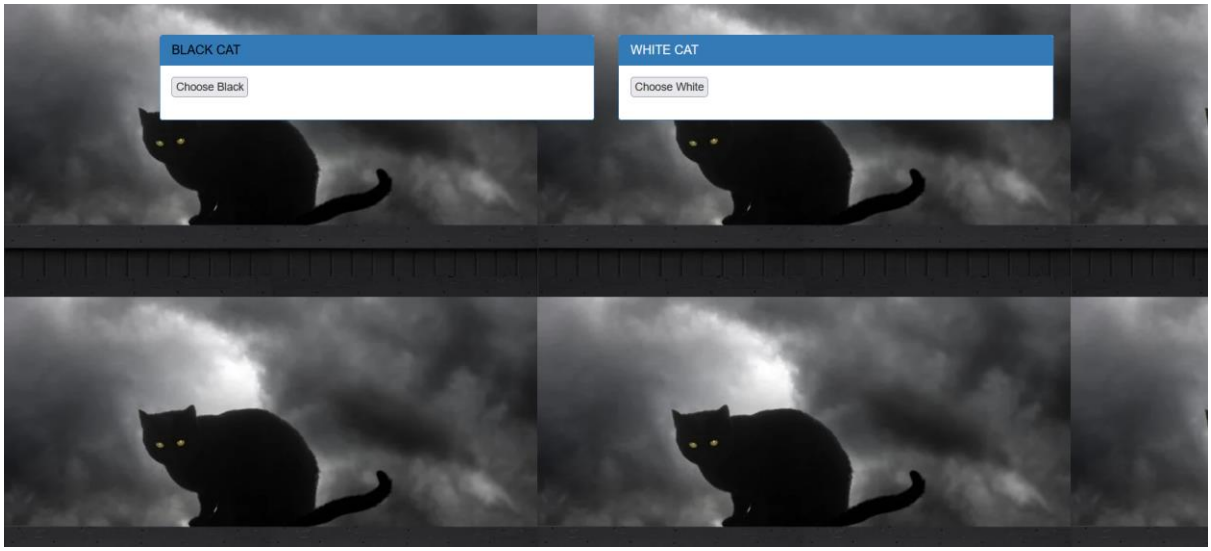
Then, I checked the response request for one of the file loaded in the page and among them is profile.js, inside the file, we can see the flag.

BO1337{a2c13e70ff50376e259ddb5bd5e54a69b16e569f}

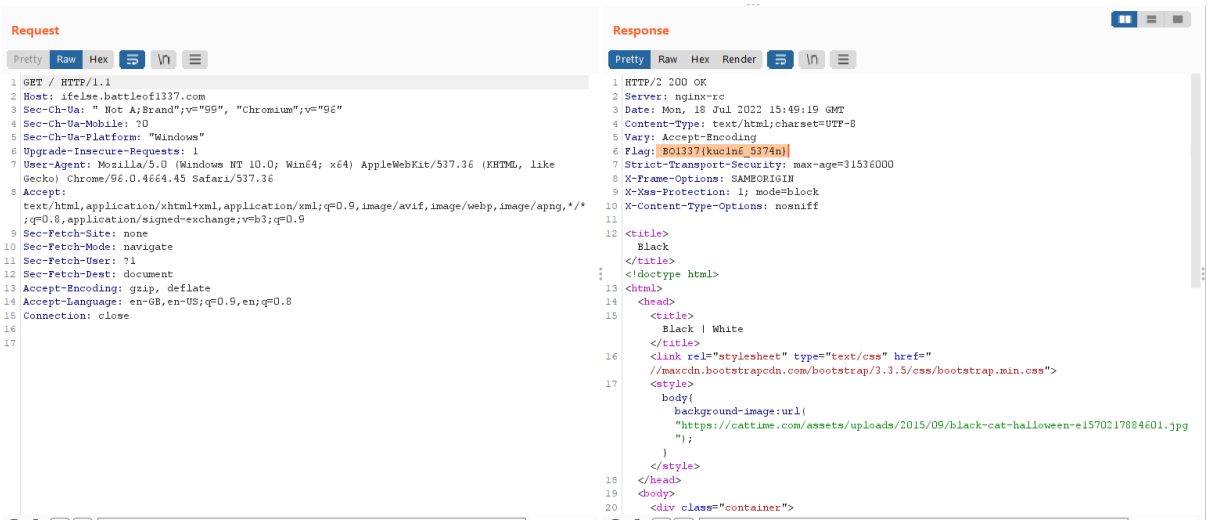
CAT-DALMANTION



https://ifelse.battleof1337.com/



Opening the site, I didn't notice anything at first. Pressing the buttons only changes the background of the site. I proceed to see the request in burp to inspect it further.



Request

```
GET / HTTP/1.1
Host: ifelse.battleof1337.com
Sec-Ch-Ua: " Not A;Brand";v="99", "Chromium";v="96"
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/96.0.4664.45 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=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
```

Response

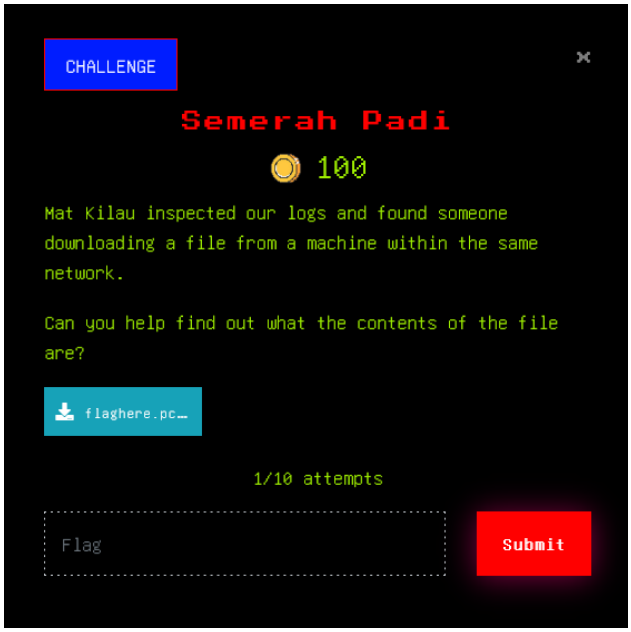
```
HTTP/2 200 OK
Server: nginx-rc
Date: Mon, 18 Jul 2022 15:49:19 GMT
Content-Type: text/html;charset=UTF-8
Vary: Accept-Encoding
Flag: BO1337{kuc1n6_5374n}
Strict-Transport-Security: max-age=31536000
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 1; mode=block
X-Content-Type-Options: nosniff

<title>
  Black
</title>
<!doctype html>
<html>
  <head>
    <title>
      Black | White
    </title>
    <link rel="stylesheet" type="text/css" href="
    //maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
    <style>
      body{
        background-image:url(
        "https://cattime.com/assets/uploads/2015/09/black-cat-halloween-e1570217884601.jpg
        ");
      }
    </style>
  </head>
  <body>
    <div class="container">
```

We can see a Flag parameter in the response.

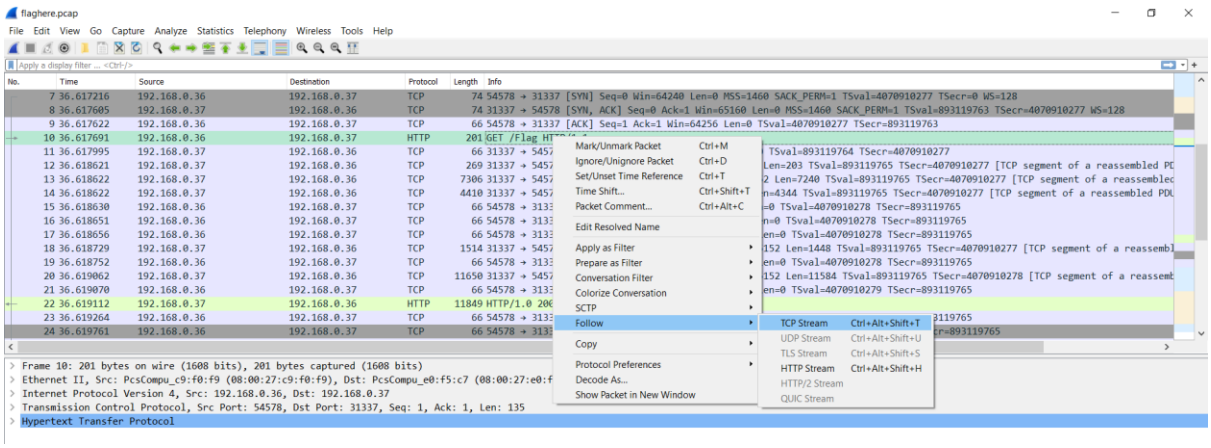
BO1337{kuc1n6_5374n}

NET

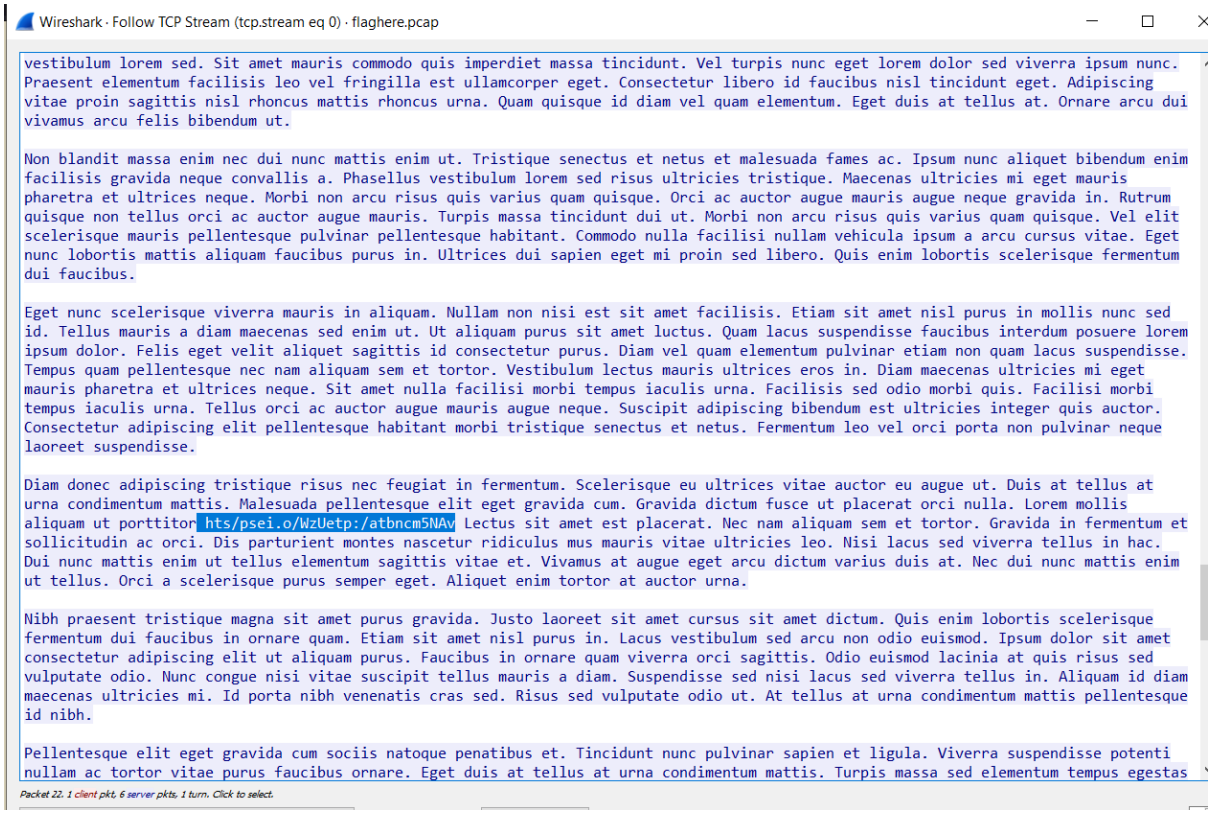
SEMERAH PADI



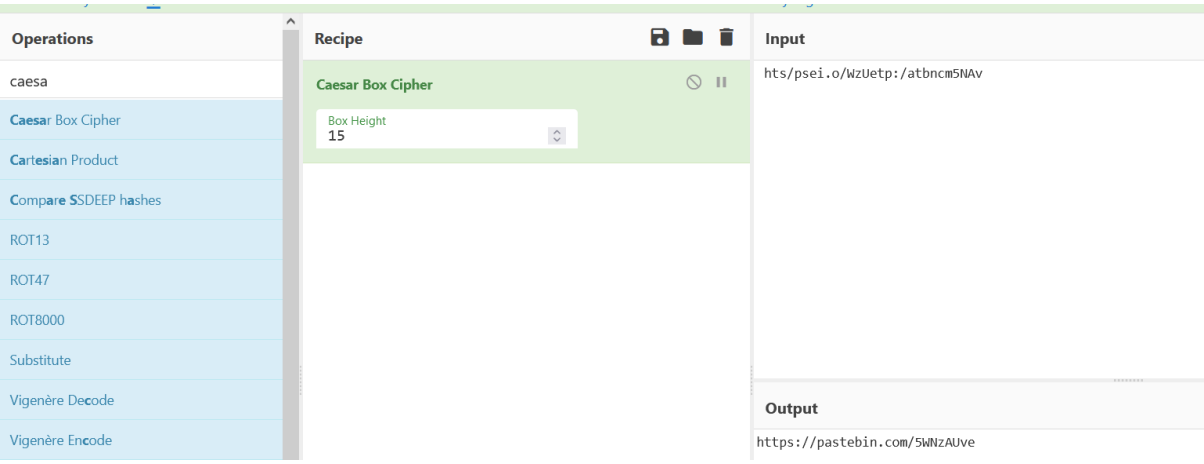
We are given a pcap file “flaghere.pcap”, proceed to open in Wireshark.



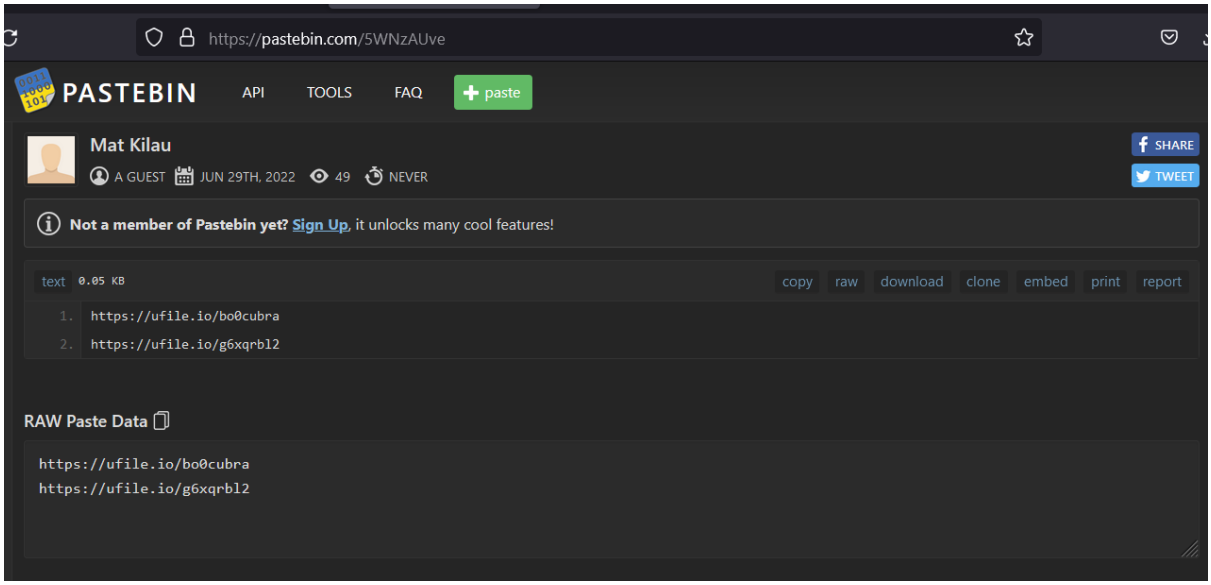
Follow the HTTP/TCP stream of the packets to see the file downloaded by the user.



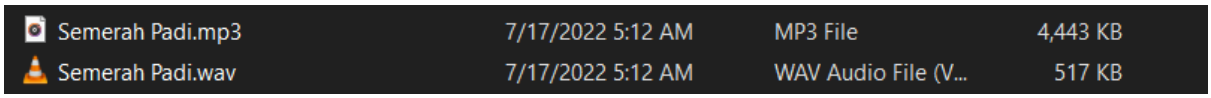
We can see the file contains a lot of dummy text in latin, however among the text we can see one line of text that isn't latin with "/" which usually comes from url.



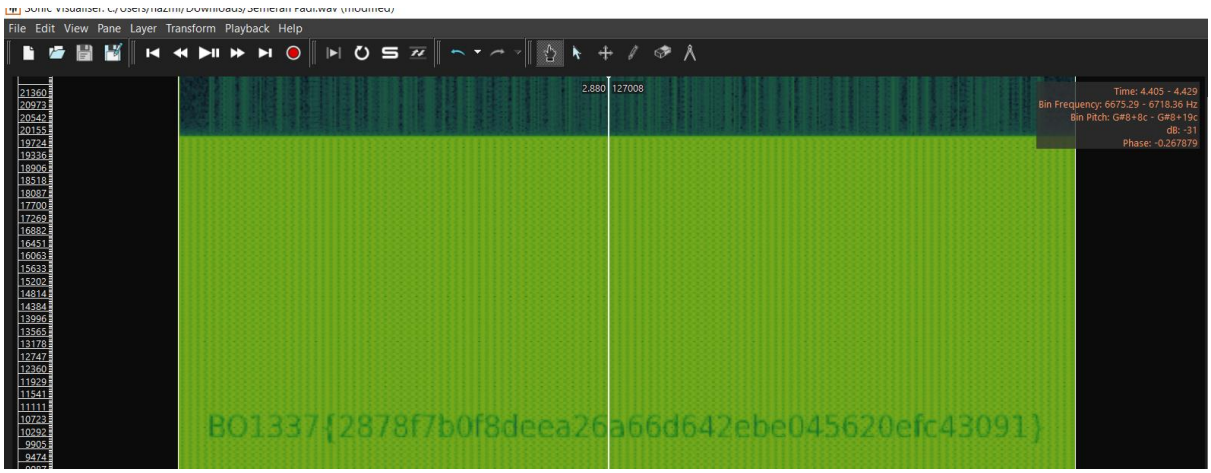
Trying some Caesar decoding tool, I finally get an output using CyberChef Caesar Box Cipher at the 15 height. The decoded text is a pastebin url: https://pastebin.com/5WNzAUve.



Inside the pastebin is two ufile.io file which kept two audio file.



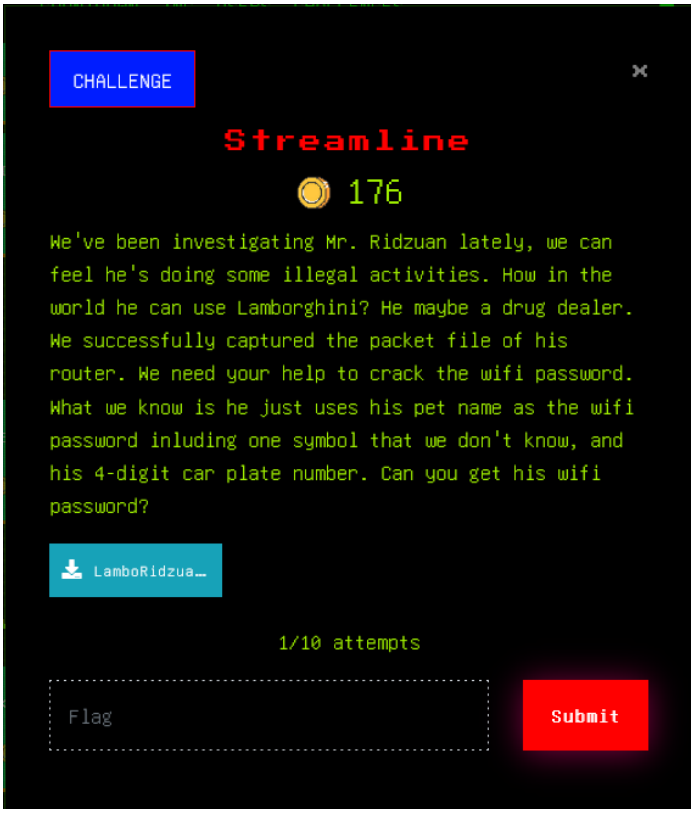
Semerah Padi.mp3	7/17/2022 5:12 AM	MP3 File	4,443 KB
Semerah Padi.wav	7/17/2022 5:12 AM	WAV Audio File (V...	517 KB



Using sonic visualizer, we can see the spectrogram of the audio. The .wav file spectrogram contains the flag.

BO1337{2878f7b0f8deea26a66d642ebe045620efc43091}

STREAMLINE



Downloading the image, we can see a picture containing a lambo and an orange cat

Proceed to use some steganography tools including steghide.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# steghide --extract -sf LamboRidzuan.jpg
Enter passphrase:
the file "handshake.cap" does already exist. overwrite ? (y/n) y
wrote extracted data to "handshake.cap".
```

Managed to extract a .cap file from the image.

handshake.cap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
162	2.428139	ae:0b:fb:d7:e5:a2	7e:f3:d4:13:71:64	802.11	114	Probe Response, SN=1166, FN=0, Flags=....R..., BI=100, SSID=Ridzuan Wifi
163	2.428289	XiaomiCo_1c:53:05	ae:0b:fb:d7:e5:a2	802.11	26	Deauthentication, SN=23, FN=0, Flags=.....
164	2.429993	ae:0b:fb:d7:e5:a2	7e:f3:d4:13:71:64	802.11	114	Probe Response, SN=1166, FN=0, Flags=....R..., BI=100, SSID=Ridzuan Wifi
165	2.430006	ae:0b:fb:d7:e5:a2	XiaomiCo_1c:53:05	802.11	26	Deauthentication, SN=18, FN=0, Flags=.....
166	2.432044	ae:0b:fb:d7:e5:a2	7e:f3:d4:13:71:64	802.11	114	Probe Response, SN=1166, FN=0, Flags=....R..., BI=100, SSID=Ridzuan Wifi
167	2.434036	ae:0b:fb:d7:e5:a2	XiaomiCo_1c:53:05	802.11	26	Deauthentication, SN=24, FN=0, Flags=.....
168	2.434044	ae:0b:fb:d7:e5:a2	7e:f3:d4:13:71:64	802.11	114	Probe Response, SN=1166, FN=0, Flags=....R..., BI=100, SSID=Ridzuan Wifi
169	2.435989	ae:0b:fb:d7:e5:a2	7e:f3:d4:13:71:64	802.11	114	Probe Response, SN=1166, FN=0, Flags=....R..., BI=100, SSID=Ridzuan Wifi
170	2.436101	XiaomiCo_1c:53:05	ae:0b:fb:d7:e5:a2	802.11	26	Deauthentication, SN=25, FN=0, Flags=.....
171	2.437937	ae:0b:fb:d7:e5:a2	7e:f3:d4:13:71:64	802.11	114	Probe Response, SN=1166, FN=0, Flags=....R..., BI=100, SSID=Ridzuan Wifi
172	2.439804	ae:0b:fb:d7:e5:a2	XiaomiCo_1c:53:05	802.11	26	Deauthentication, SN=26, FN=0, Flags=.....
173	2.439812	ae:0b:fb:d7:e5:a2	7e:f3:d4:13:71:64	802.11	114	Probe Response, SN=1166, FN=0, Flags=....R..., BI=100, SSID=Ridzuan Wifi
174	2.441775		XiaomiCo_1c:53:05 (…	802.11	10	Acknowledgement, Flags=.....
175	2.441901	XiaomiCo_1c:53:05	ae:0b:fb:d7:e5:a2	802.11	26	Deauthentication, SN=27, FN=0, Flags=.....
176	2.443730		XiaomiCo_1c:53:05 (…	802.11	10	Acknowledgement, Flags=.....
177	2.443742	XiaomiCo_1c:53:05	ae:0b:fb:d7:e5:a2	802.11	26	Deauthentication, SN=19, FN=0, Flags=.....
178	2.445533		XiaomiCo_1c:53:05 (…	802.11	10	Acknowledgement, Flags=.....
179	2.447781		XiaomiCo_1c:53:05 (…	802.11	10	Acknowledgement, Flags=.....

> Frame 1: 120 bytes on wire (960 bits), 120 bytes captured (960 bits)
> IEEE 802.11 Beacon frame, Flags: .....
> IEEE 802.11 Wireless Management

Opening the file in Wireshark, we can see it contains packet from a wifi network. From the challenge description, we have to crack the wifi password from the packets using the hint give. We can use aircrack-ng to crack the wpa handshake, however we need to have a wordlist. The description given of the wifi password is:

- pet name

- one symbol

- 4 digit car plate number

From this description, we can conclude and guess the password. Since the pet is an orange cat, we Malaysian often call them oyen. I proceed to generate a wordlist using crunch with the details:

-8 characters

-start with oyen

-^= all symbols

-%=number from 0-9

Crunch will generate the wordlist for us.

```
Crunch ending at cat?1585

┌──(root㉿kali)-[/home/kali/Downloads]
└─# crunch 8 8 -t cat^%%%% -o wl1.txt
Crunch will now generate the following amount of data: 2970000 bytes
2 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 330000

crunch: 100% completed generating output

┌──(root㉿kali)-[/home/kali/Downloads]
└─# crunch 11 11 -t oyen^%%%% -o wl2.txt
The maximum and minimum length should be the same size as the pattern you specified.
min = 11  max = 11  strlen(oyen^%%%%)=9

┌──(root㉿kali)-[/home/kali/Downloads]
└─# crunch 9 9 -t oyen^%%%% -o wl2.txt
Crunch will now generate the following amount of data: 3300000 bytes
3 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 330000

crunch: 100% completed generating output
```

```
└─# aircrack-ng handshake.cap -w wl2.txt
Reading packets, please wait...
Opening handshake.cap
Read 2497 packets.

   #  BSSID              ESSID                     Encryption

   1  AE:0B:FB:D7:E5:A2  Ridzuan Wifi              WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening handshake.cap
Read 2497 packets.

1 potential targets


                              Aircrack-ng 1.6

      [00:00:06] 20092/330000 keys tested (3500.80 k/s)

      Time left: 1 minute, 28 seconds                          6.09%

                          KEY FOUND! [ oyen@9367 ]


      Master Key     : E7 F0 93 EC 46 85 39 40 10 90 31 B3 62 CE E0 13
                       77 9E A4 F8 A5 07 00 F8 33 4B 1F 7A 22 BA D5 57

      Transient Key  : 0B 56 BD 4D AF AD 2B 8D 4E A8 CF EC 26 3E C3 5D
                       AC CC 49 8D D3 AD CC AB 73 B9 15 02 3B 90 1F 10
                       6A 1D BB 51 41 84 B3 5D EA D0 94 C4 B3 77 4B 62
                       71 E1 D6 5F 88 B8 D6 F8 57 D4 4F DF D4 96 CF 00
```

Using the wordlist, we can start cracking the password, and finally get the password which is oyen@9367.

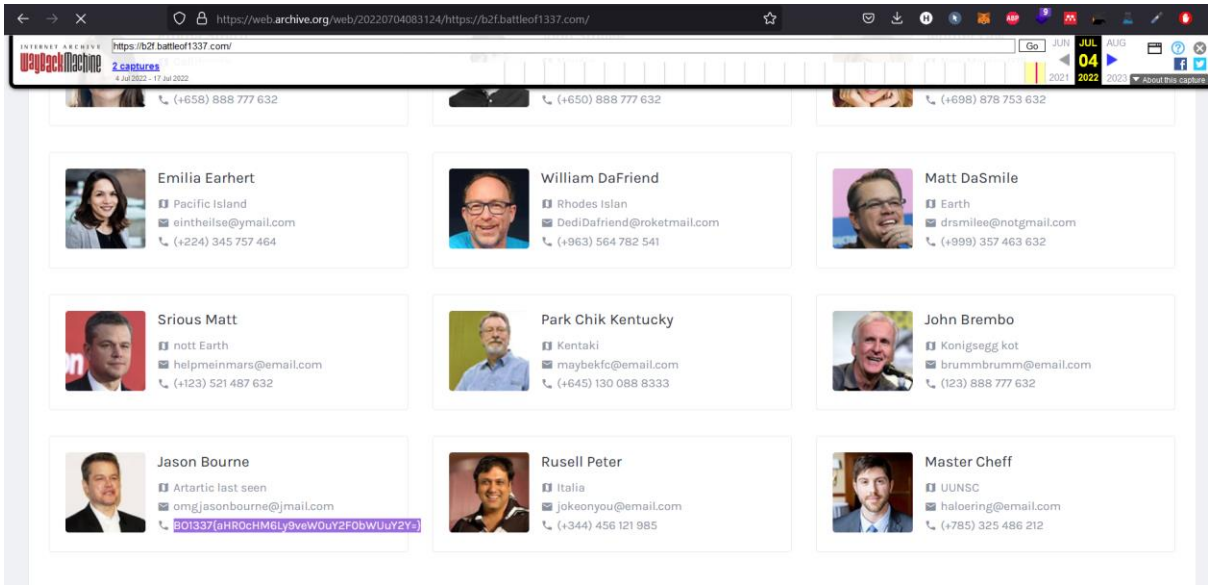
BO1337{oyen@9367}

OSINT

BACK TO THE FUTURE

From challenge name we can guess that we have to see the website from previous date. Using wayback machine we can see the the site has been indexed on 4 July. Opening the site present is with flag.

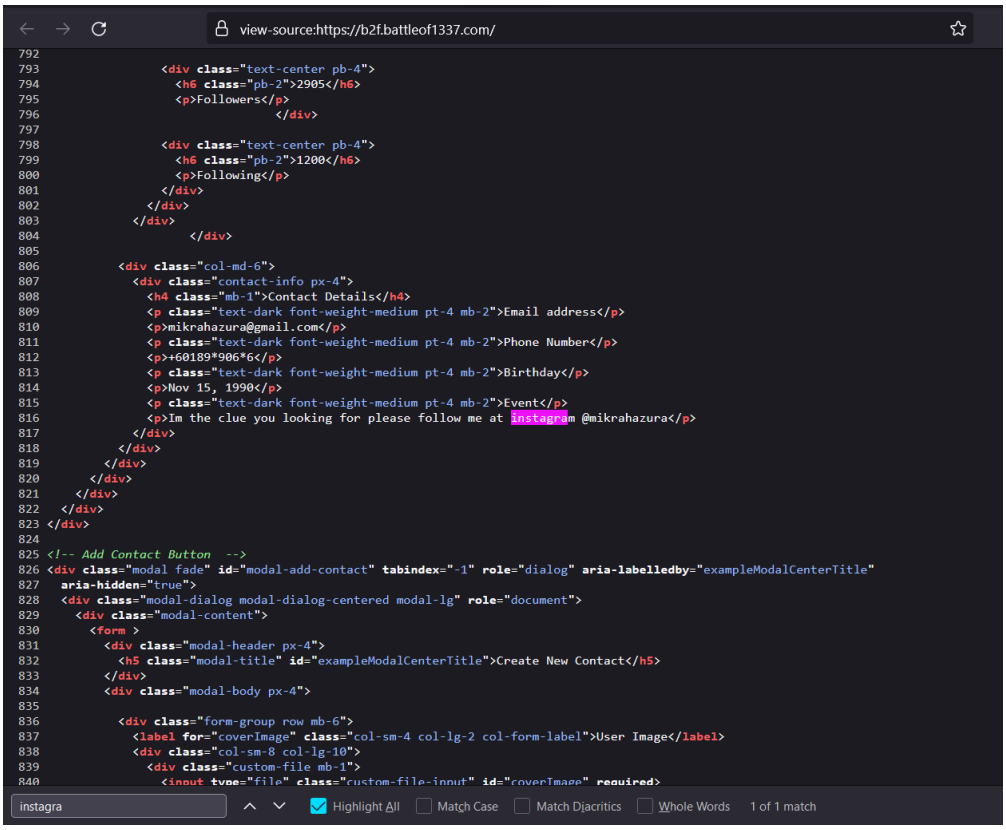
HTTPS://ARCHIVE.ORG/WEB/



BO1337{aHR0cHM6Ly9veW0uY2F0bWUuY2Y=}

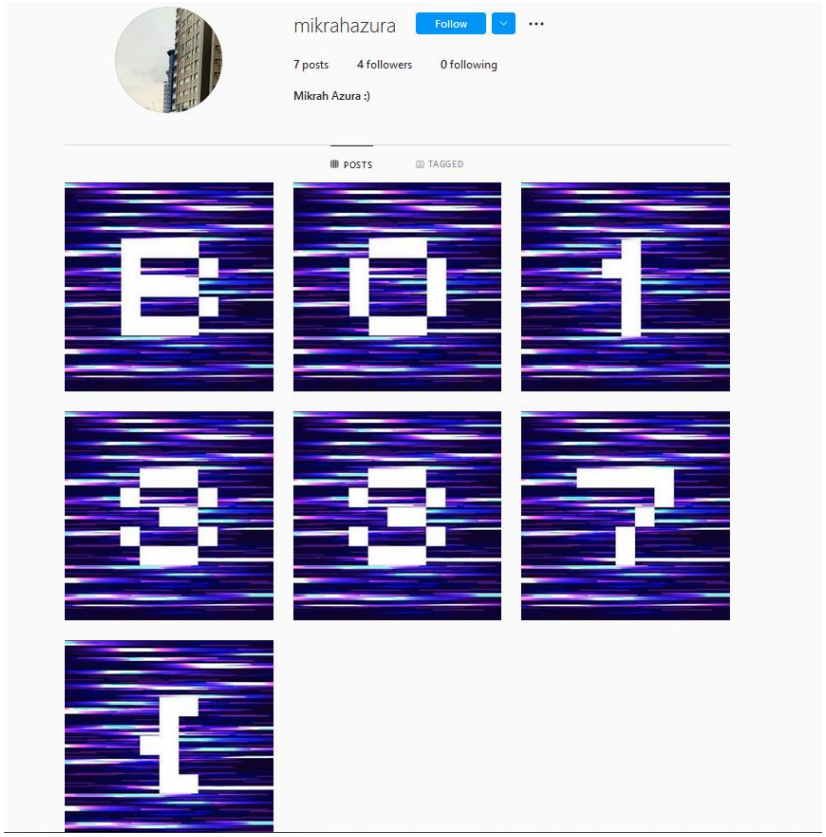
1GRAM

We can search for any Instagram @ from the website and found one @mikrahazura.

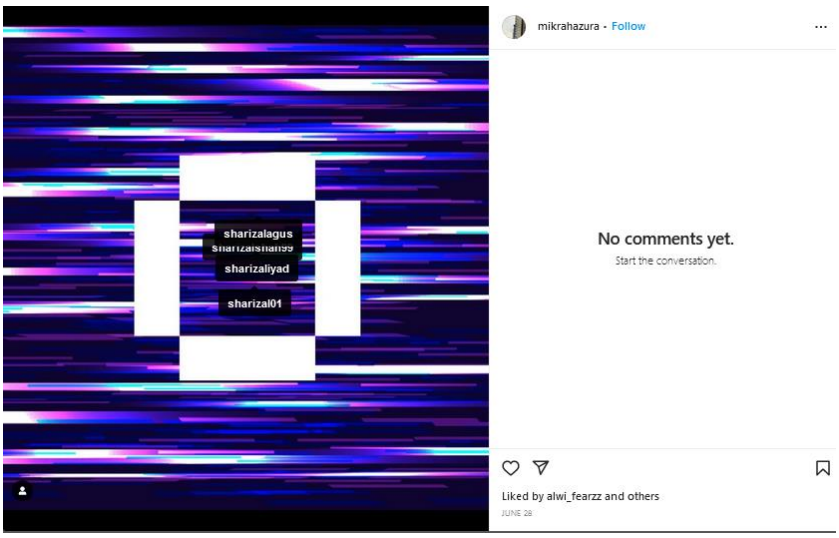



```
792
793                 <div class="text-center pb-4">
794                   <h6 class="pb-2">2905</h6>
795                   <p>Followers</p>
796                               </div>
797
798                 <div class="text-center pb-4">
799                   <h6 class="pb-2">1200</h6>
800                   <p>Following</p>
801                 </div>
802               </div>
803             </div>
804                     </div>
805
806           <div class="col-md-6">
807             <div class="contact-info px-4">
808               <h4 class="mb-1">Contact Details</h4>
809               <p class="text-dark font-weight-medium pt-4 mb-2">Email address</p>
810               <p>mikrahazura@gmail.com</p>
811               <p class="text-dark font-weight-medium pt-4 mb-2">Phone Number</p>
812               <p>+60189*906*6</p>
813               <p class="text-dark font-weight-medium pt-4 mb-2">Birthday</p>
814               <p>Nov 15, 1990</p>
815               <p class="text-dark font-weight-medium pt-4 mb-2">Event</p>
816               <p>Im the clue you looking for please follow me at instagram @mikrahazura</p>
817             </div>
818           </div>
819         </div>
820       </div>
821     </div>
822   </div>
823 </div>
824
825 <!-- Add Contact Button  -->
826 <div class="modal fade" id="modal-add-contact" tabindex="-1" role="dialog" aria-labelledby="exampleModalCenterTitle"
827   aria-hidden="true">
828   <div class="modal-dialog modal-dialog-centered modal-lg" role="document">
829     <div class="modal-content">
830       <form >
831         <div class="modal-header px-4">
832           <h5 class="modal-title" id="exampleModalCenterTitle">Create New Contact</h5>
833         </div>
834         <div class="modal-body px-4">
835
836           <div class="form-group row mb-6">
837             <label for="coverImage" class="col-sm-4 col-lg-2 col-form-label">User Image</label>
838             <div class="col-sm-8 col-lg-10">
839               <div class="custom-file mb-1">
840                 <input type="file" class="custom-file-input" id="coverImage" required>
```

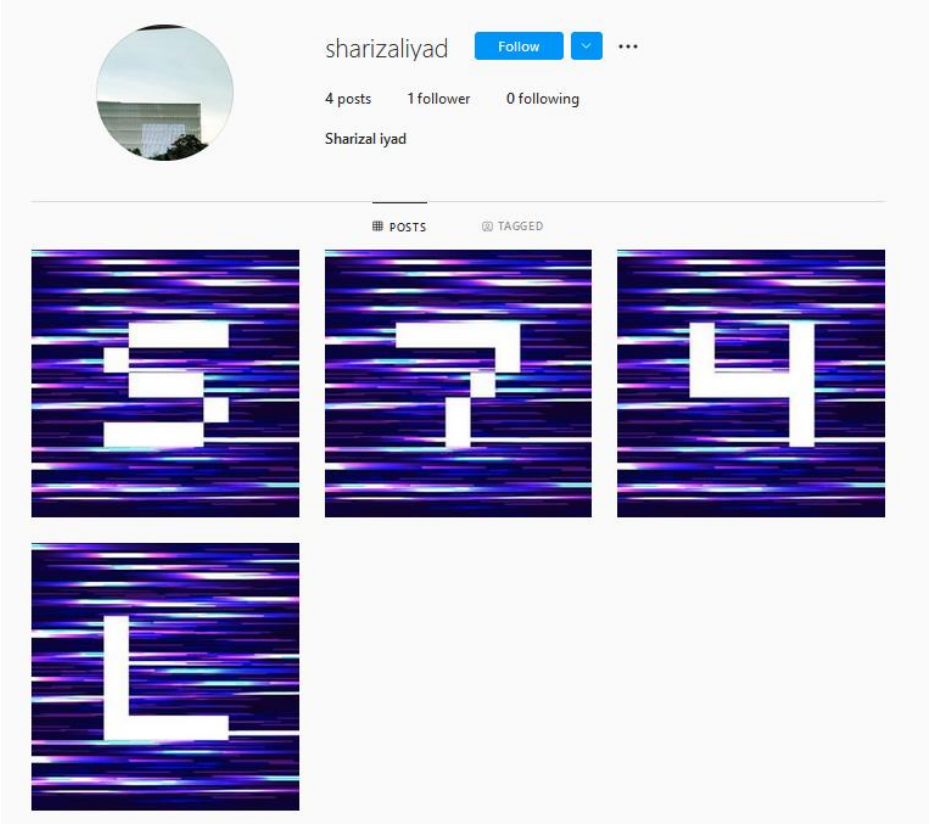
Going to instagram profile, we can see part of the flag.



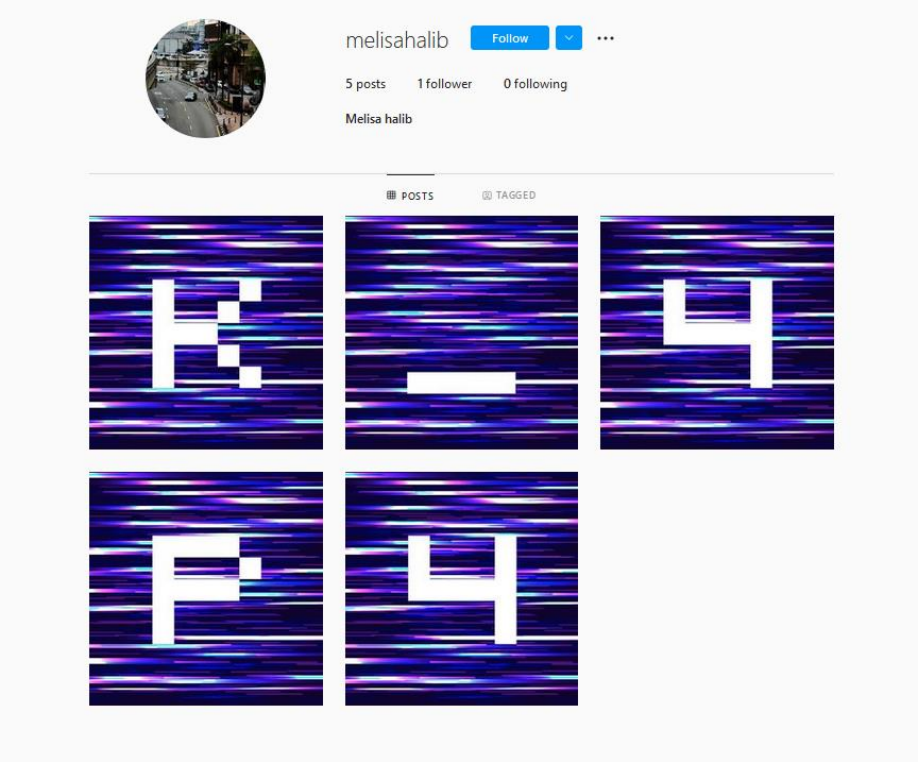
One of the image tagged several profile.

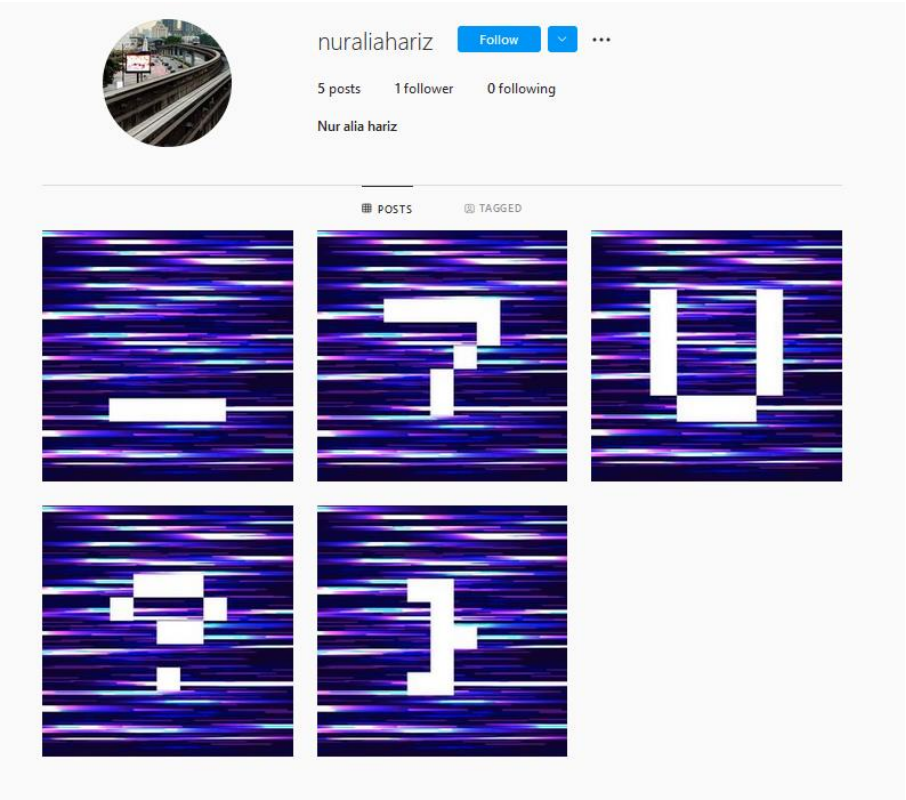


One of the profile contains the next part of the flag.



Proceed to repeat the process, and combine the flag.

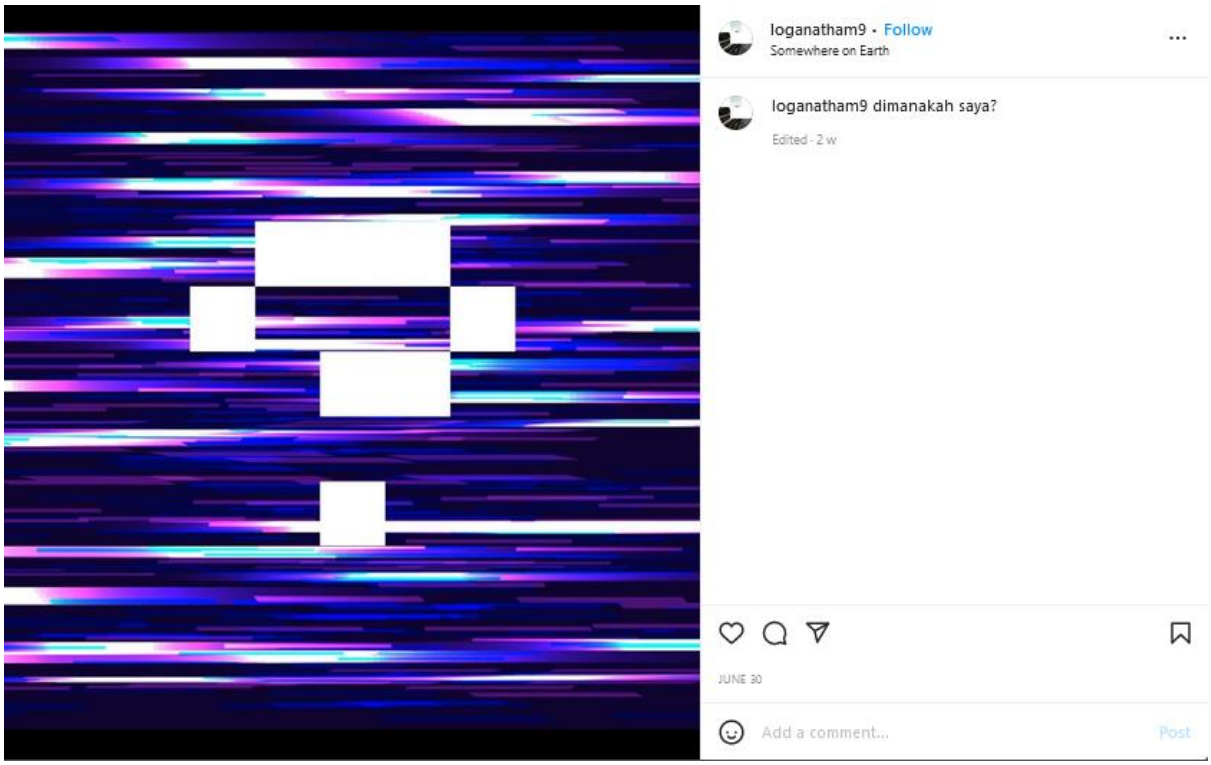




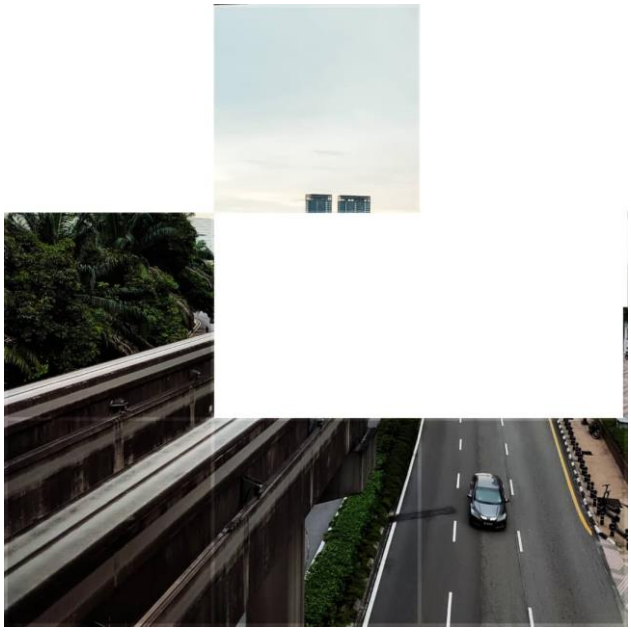
BO1337{S74LK_4P4_7U?}.

SNAP

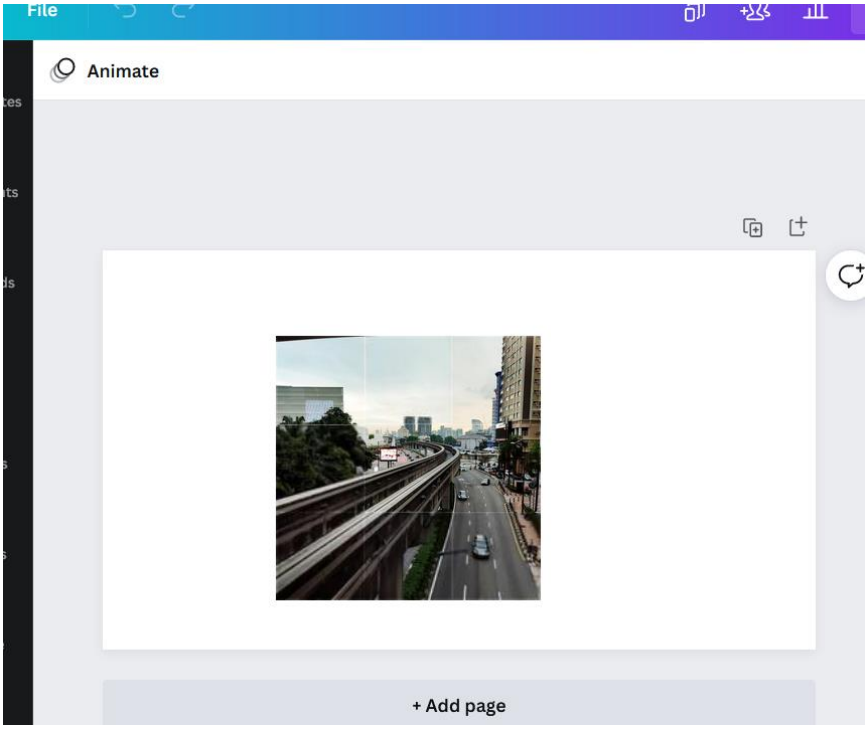
From the previous challenge, we can see another profile



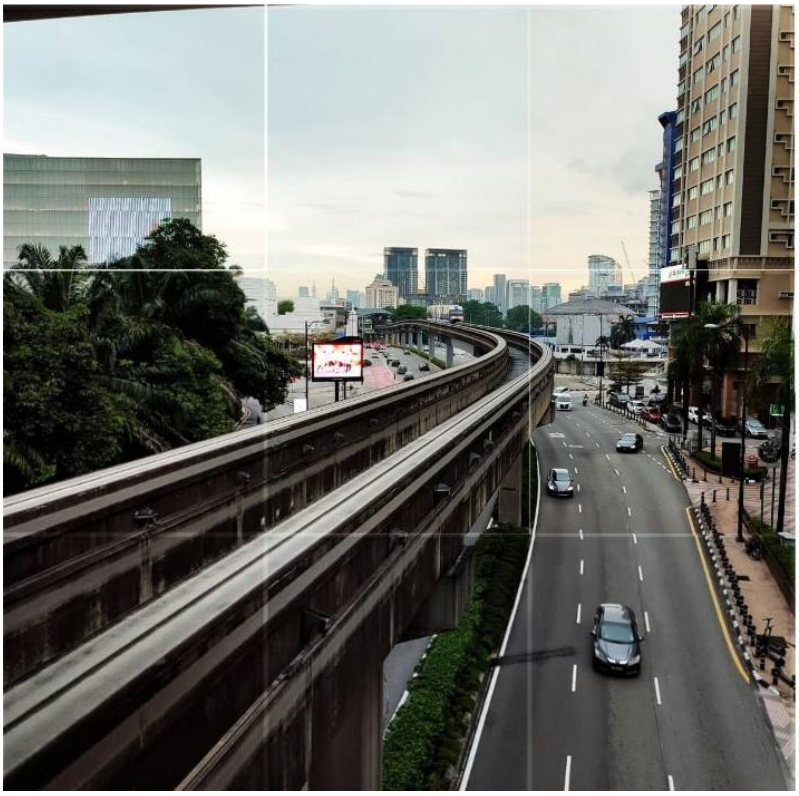
Caption says “dimanakah saya”. I noticed that the profile image on this profile wasn’t completed.



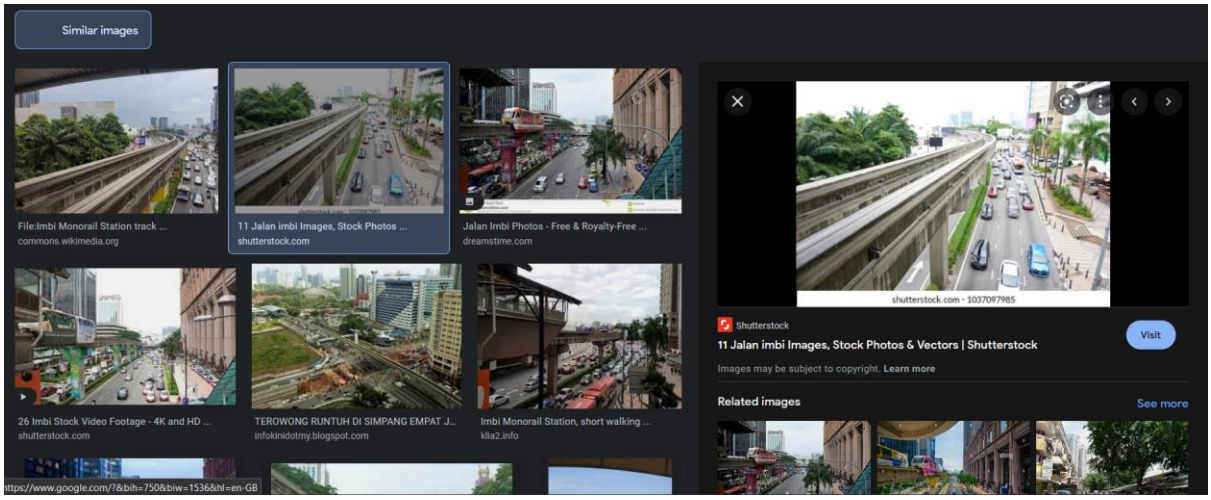
I proceed to download all the profile pictures and used canva to combine the image.



From the grid on the picture, we can guess that we need to use profile picture of other profiles to get the full image.



After getting the full image, I started using reverse image search on Google to find any similar image.

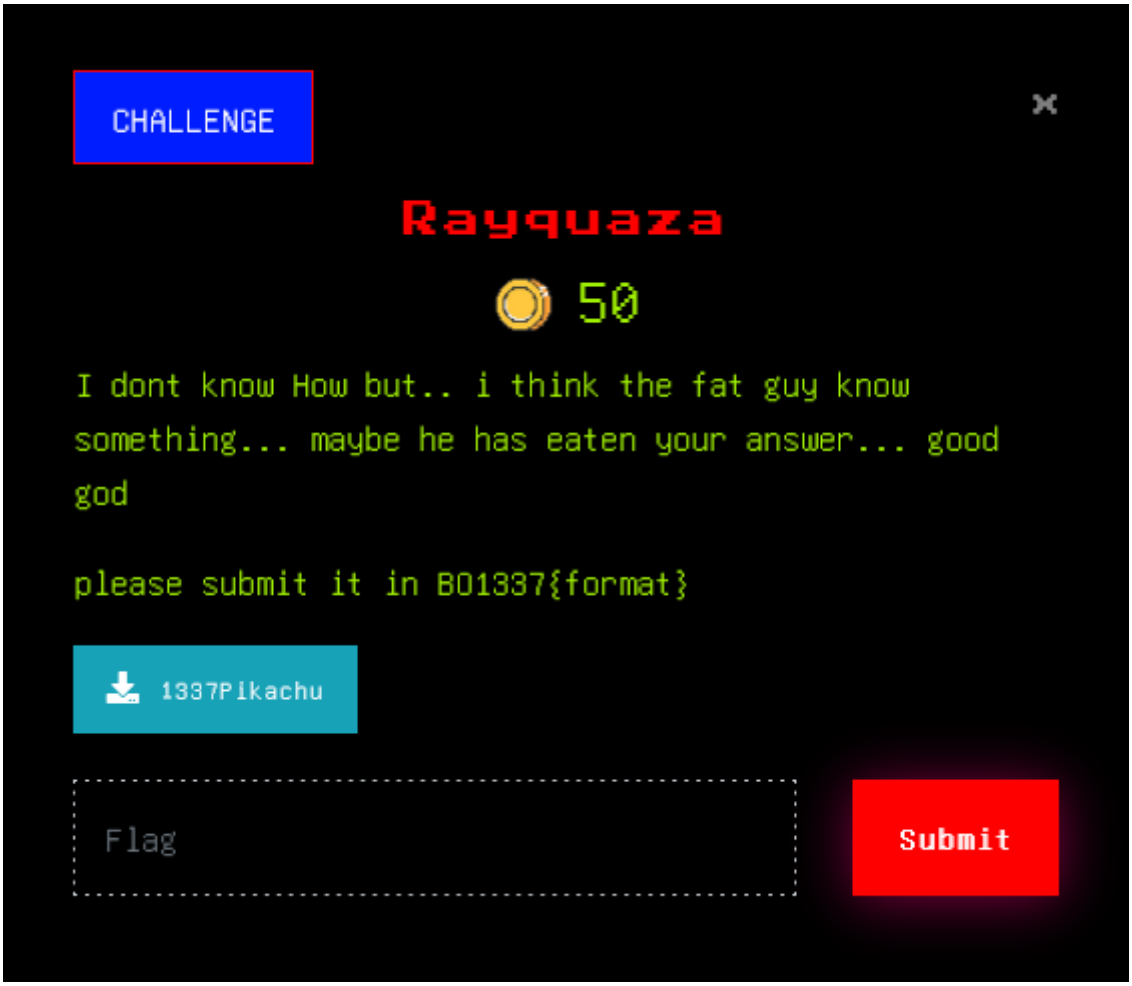


From here we know that the image came from Imbi MRT. I was confused on where the flag was and opened a ticket on whether we need to fill the BO1337{flag} format ourself in order to submit and the admin answered yes. It’s pretty confusing to be honest since no declaration in description.

BO1337{Imbi}

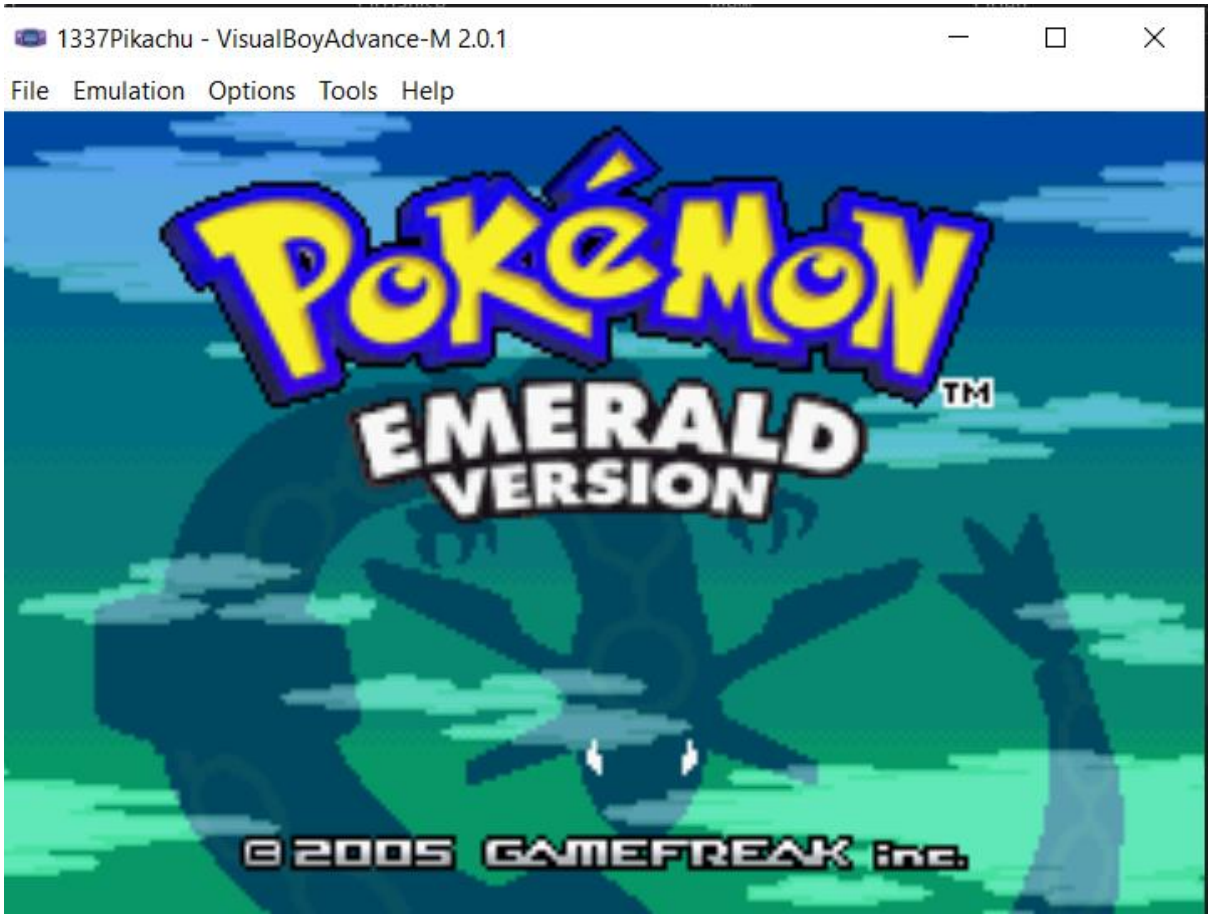
MISC

RAYQUAZA

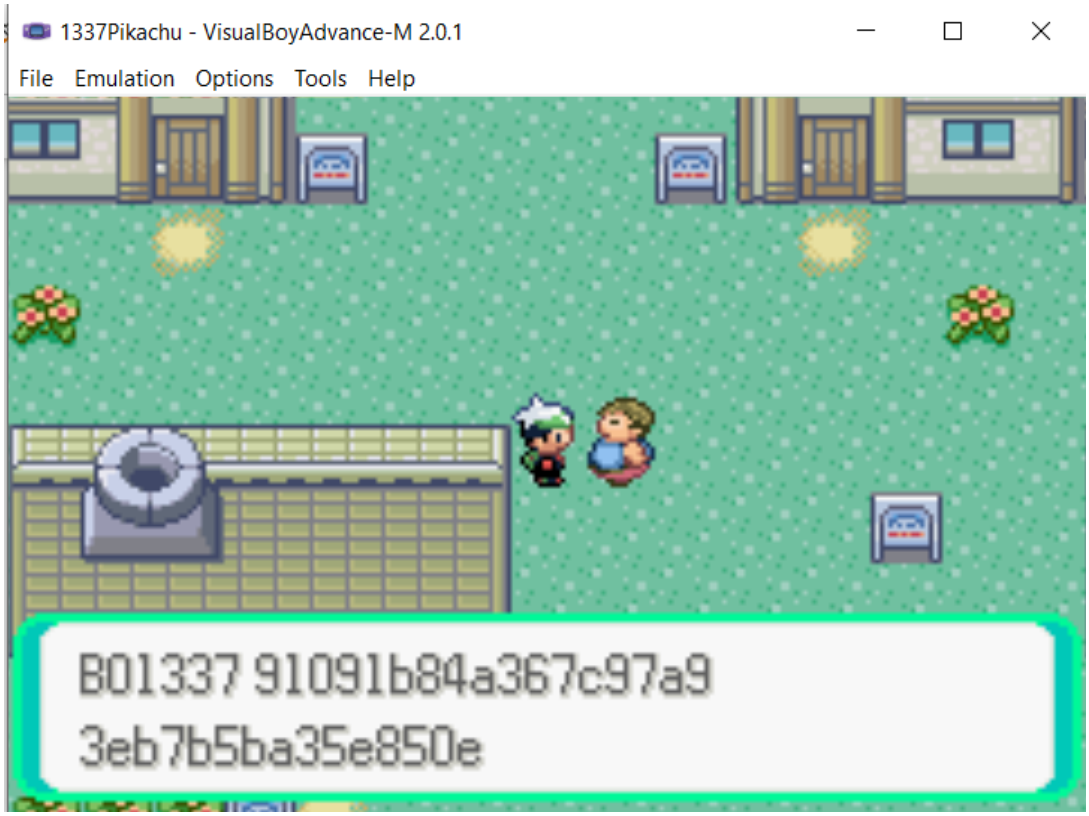


First we can check the file type, it turns out to be a GBA rom. To run the rom, we can use any GBA emulator such as VBA, but first we must rename the file to 1337Pikachu.gba.

```
┌──(root㉿kali)-[/home/kali/Downloads]
└─# file 1337Pikachu
1337Pikachu: Game Boy Advance ROM image: "POKEMON EMER" (BPEE01, Rev.00)
```



Opening the game, it turns to be a pokemon emerald rom.



After talking with the mom and stuff we proceed to go out and find the fat man from the description and he gave us the flag.

BO1337{91091b84a367c97a93eb7b5ba35e850e}

HEIHAWRU

Opening the .txt file, we can see encoded text and numbers with ?:?:? format, I proceed to decode it using DCODE.FR Caesar cipher tool.

sajak_ali-1.txt - Notepad

File Edit Format View Help

```
Lqxh hxkxk qbhxk qryf mbkx gxaf abiqlfa
Ixdr jfkrj molqbfk oxm prkqfh pqbolfa
Arx hxqf zobxqfkb xhr xkaolfa
Axqxkd pbpbhxif fjmxh xpqbolfa (yllj)
Xhr hbgxj hrefixkd hxr ofkar
Ibmxp qxjmxo jxr zfrj pxqr jfkddr
Gxkdxk ybdfqr gxkdxk jbkdxax
Jxzxj jxkx ylibe ofkar Jxtf hxk xax hxk?
(Xppxixjrxixfhrj)
Obxifqfkvx fkf 083
Yfkqxkd obxifqf hrgxafhxk pxoxmxk
Jxhxk ifjx jxkdhrh qxkvx, jxkx pxoxmxkhr?
Jxhxk ifjx jxkdhrh ixdf, jxkx pxoxmxkhr?
Xhr ixmxo, vx jxkx jrpre qbohfkf?
Xhr yxhxo qboyxifhhxk af mxpxo jxzxj jrpplifkf
Jxkx hrkaxifkf? qxh hrkgrkd qfyx
Pxhqf gxaf efyx qxmf clhrp
Zexhox ybomrqxo-mrqxo hbgxo qfqfh ilhrp
Qxcxhro, xpjxox, pbjrx mlpfpf ilqrp
Vrm, ybpq lc ylqe tloiap F'j qeb almbpq
Illh fkql qeb jfoolo, vrm vrm vlr qeb zilpbpq
F cifm ixkdrxdb ifhb pxkatfze
Bfqebo pfab pxjb mexq pefq ifhb dlaaxjk fq
Xka fc vlr dlq x molyibj tfqe jb ybfkd jxixv arab
Ibqp qxhb fq yxzh ql 1511
Vx, vx 1511
Glj yboqfhxj ifaxe abkdxk exjyx axixj xhr
Xhr mxkqxkd hxixe, yxkdrk yfix gxqre
Hxixr mxqxe pxvxm yboqlkdhxqhxk mxore (peee)
Hr mrkvx qbjxk vxkd mrkvx qbjxk
Vxkd ylibe yrxq bkdhxr efixkd qbjxk
Gxaf axof yrxq ixtxk, yxfh yrxq hxtxk
Axof yxdf gxof, yxfh xkdhxq qxkdxk
Pbmrire gxof hb xqxp jxzxj hbkx qxkdhxm
Hxixr fkdhxo hrmlqlkd qrgre gxaf mbkdxhxm
Fhxk arof, dbixjx, pbklelkd, pfxhxm
Oxmmbo mbkfmr mbjylelkd pbjrx hr mxm mxm
Yrhxk qboexkaxi, yrhxk 7bohrxq
Zrjx qboexkaxi axixj xmx vxkd hr yrxq
Gxaf yfix dbkboxi yborzxm, pxjmxf axoxe drpf
Qlilkd zboxfhxk yrkqrq axof hborpf
Yxkdrk

5:6:2 6:1:1 31:3:1 15:3:3 15:3:3 43:4:1 27:2:1 32:3:1 33:1:1 41:3:1 38:3:4 24:2:2 10:5:4 41:5:3 45:6:3 35:1:1 15:3:3 1:3:3 36:2:2 34:1:1 45:2:3 21:2:2 17:1:2 11:4:2
```

Assalamualaikum, dari jeneral
RAP, rythm and poetry
Terjemahannya ritma atas puisi
Jurucakap institusi puisi
Berjasa pada semua macam pensil 2B
Otak kanan tekan tubi pena jadi deltoid
Lagu minum protein rap suntik steroid
Dua kati creatine aku android
Datang sesekali impak asteroid (boom)
Aku kejam kuhilang kau rindu
Lepas tampar mau cium satu minggu
Jangan begitu jangan mengada
Macam mana boleh rindu Mawi kan ada kan?
(Assalamualaikum)
Realitinya ini 083
Bintang realiti kujadikan sarapan
Makan lima mangkuk tanya, mana sarapanku?
Makan lima mangkuk lagi, mana sarapanku?
Aku lapar, ya mana musuh terkini?
Aku bakar terbalikkan di pasar macam mussolini
Mana kundalini? tak kunjung tiba
Sakti jadi hiba tapi fokus
Chakra berputar-putar kejar titik lokus
Tafakur, asmara, semua posisi lotus
Yup, best of both worlds I'm the dopest
Look into the mirror, yup yup you the closest
I flip language like sandwich
Either side same phat shit like goddamn it
And if you got a problem with me being malay dude
Lets take it back to 1511
Ya, ya 1511
Jom bertikam lidah dengan hamba dalam aku
Aku pantang kalah, bangun bila jatuh
Kalau patah sayap bertongkatkan paruh (shhh)
Ku punya teman yang punya teman
Yang boleh buat engkau hilang teman
Jadi dari buat lawan, baik buat kawan
Dari bagi jari, baik angkat tangan

5:6:2 6:1:1 31:3:1 15:3:3 15:3:3 43:4:1 27:2:1 f 32:3:1 33:1:1 41:3:1 38:3:4 24:2:2s 10:5:4d 41:5:3
45:6:3 35:1:1 15:3:33 1:3:3 36:2:2 34:1:1 45:2:3 21:2:2 17:1:2 11:4:2

Turns out to be a song lyric. From here, I start to guess the relation between the lyrics and the number and realize how to decode it. Taking 5:6:2 as an example:

5=Number of lines.

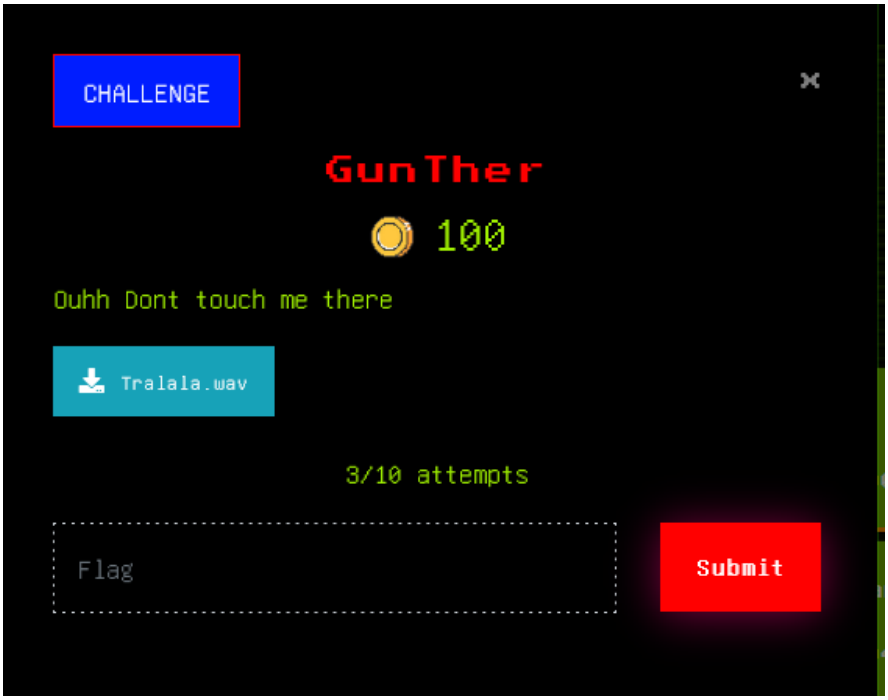
6=Number of words.

2= Number of characters.

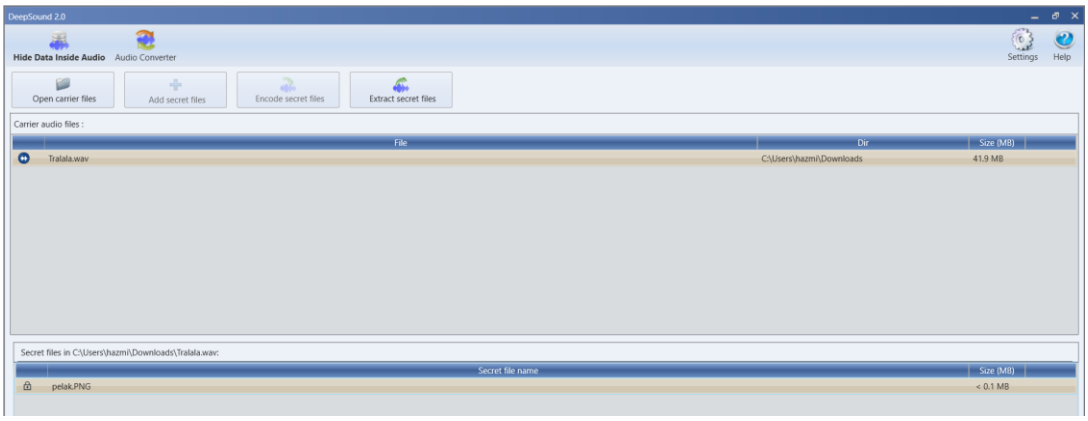
Proceed to do the same using other number and we will get flAgisdarK3noKluai.

BO1337{darK3noKluai}

GunTher



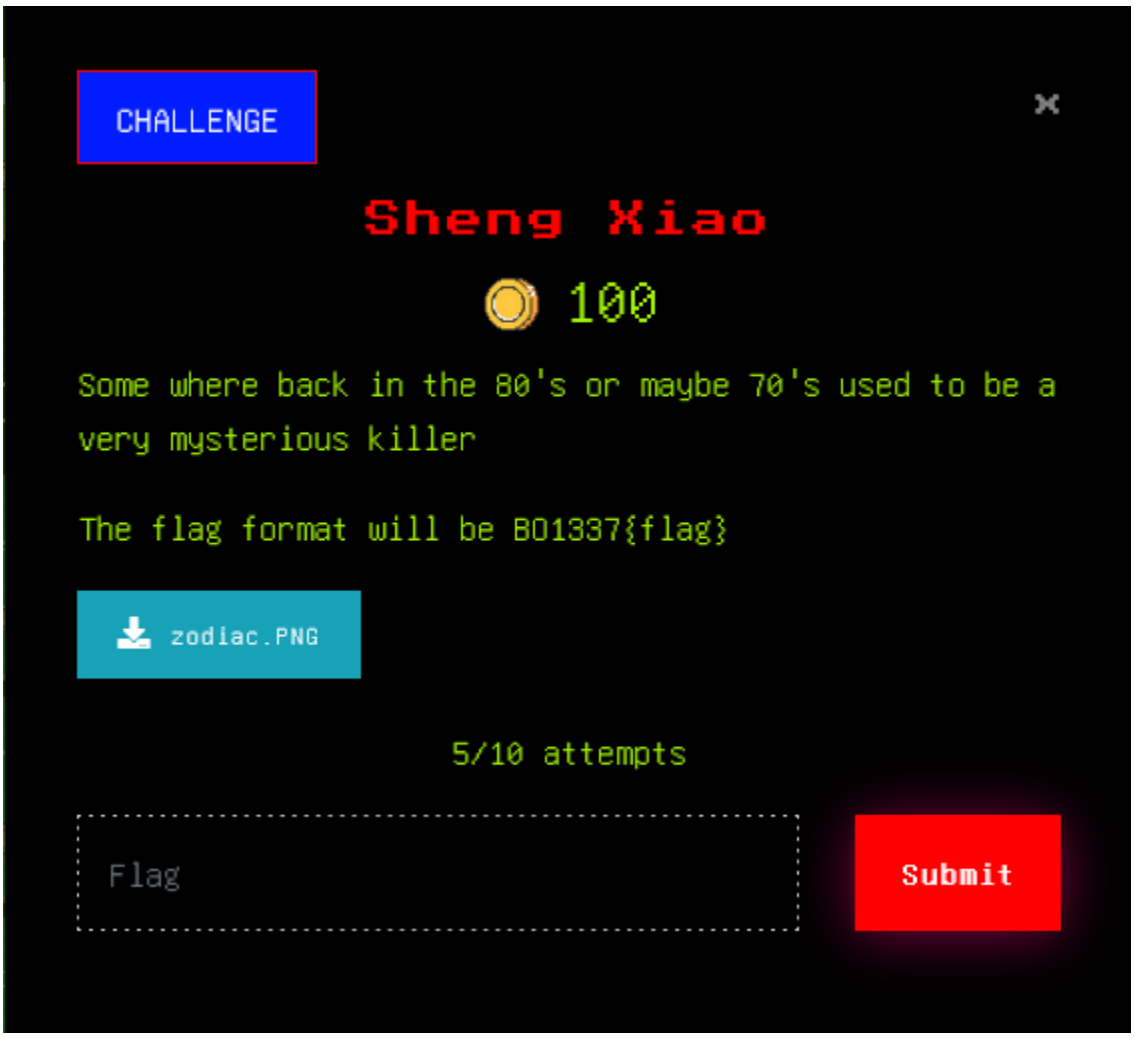
Treating the challenge as steganography, I proceed to use all the audio stega tools such as spectrogram,steghide,binwalk etc.



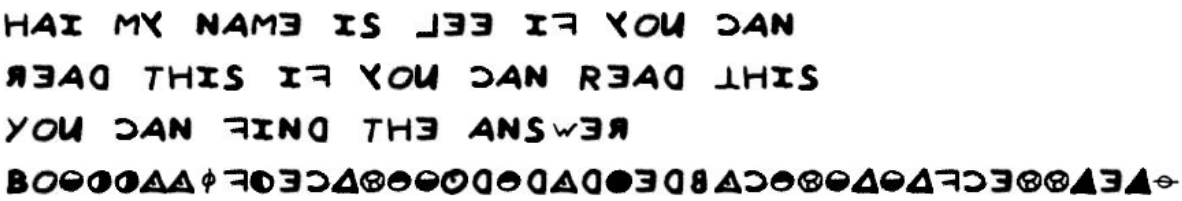
Finally using Deepsound, I manage to extract an image containing the flag.

BO1337{9d6382bf597a3014a8472762fedce888}

SHENG XIAO



Downloading the image, we can see encoded text, from description we can guess it uses the same cipher from zodiac killer.



Searching for some online tools, finally found one that was able to decode the text.

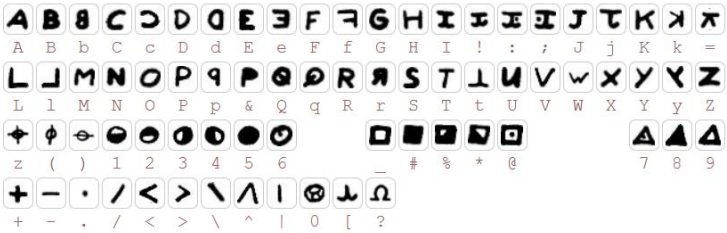
HTTP://ZODIACKILLERCIPHERS.COM/TYPEWRITER/

Make your own cipher text using Zodiac's symbols:

```
BO13377(f4ec90216d2d7d5edb7c201919fce008e8)
```

Your cipher:

Available symbols: hide



Click one of the symbols above to place it in your cipher.

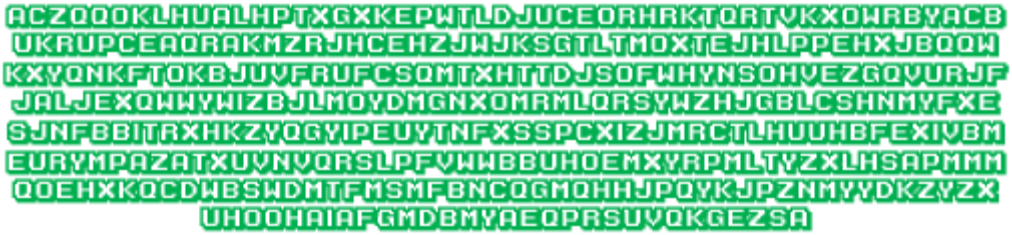
lighter | show html | show bbcode | show links | help

(go back to zodiackillerciphers.com) | (go back to webtoy)

BO1337{f4ec90216d2d7d5edb7c201919fce008e8}

DARCHROW

The image contains text.



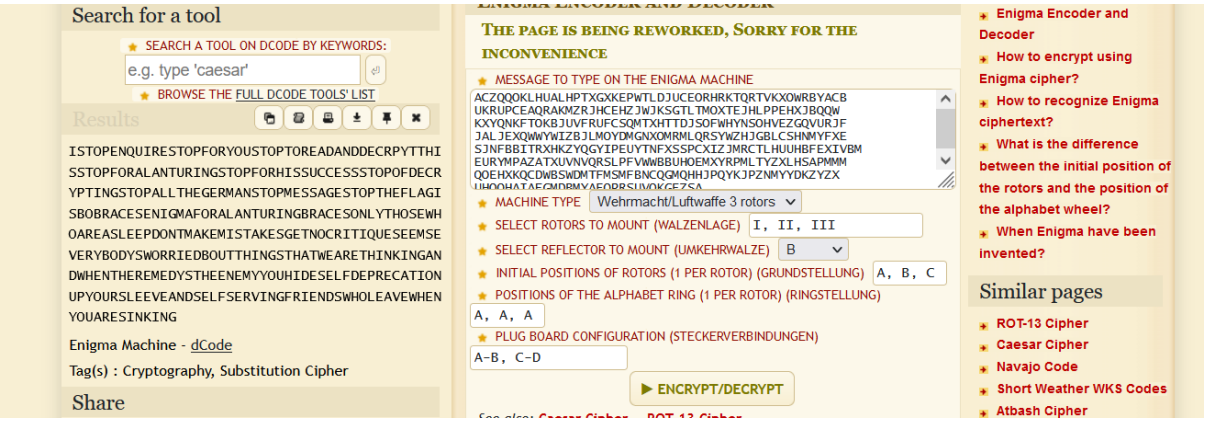
To extract the text, we can use OCR tools such as : HTTPS://WWW.IMAGETOTEXT.INFO/ however we still need to correct the extracted type. From the challenge name darchrow, we can google and know that it is the hero used in Dota which is now known as enigma in Dota 2.

ACZQQOKLHUALHPTXGXKEPWTLDJUCEORHRKTQRTVKXOWRBYACB
UKRUPCEAQRAKMZRJHCEHZJWJKSGTLTMOXTEJHLPPEHXJBQQW
KXYQNKFTOKBJUVFRUFCSQMTXHTTDJSOFWHYNSOHVEZGQVURJF
JALJEXQWWYWIZBJLMOYDMGNXOMRMLQRSYWZHJGBLCSHNMYFXE
SJNFBBITRXHKZYQGYIPEUYTNFXSSPCXIZJMRCTLHUUHBFEXIVBM
EURYMPAZATXUVNVQRSLPFVWWBBUHOEMXYRPMLTYZXLHSAPMMM

QOEHXKQCDWBSWDMTFMSMFBNCQGMQHHJPQYKJPZNMYYDKZYZX
UHOOHAIAFGMDBMYAEQPRSUVQKGEZSA

I knew about the enigma cipher that was used during the word war and proceed to use the tools from dcode.fr to decode it.

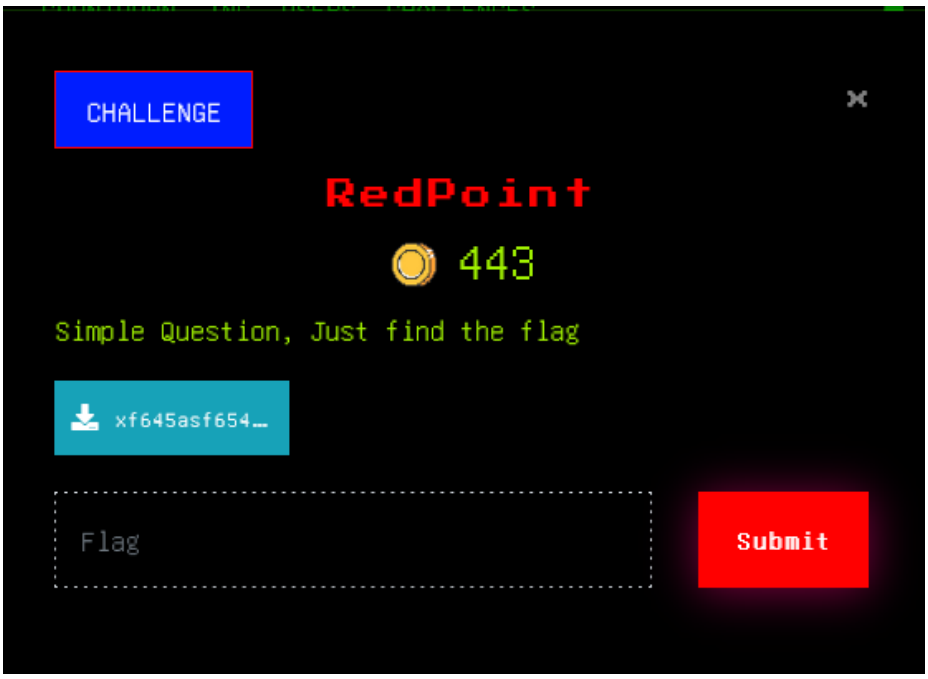
https://www.dcode.fr/cipher-identifier



ISTOPENQUIRESTOPFORYOUSTOPTOREADANDDECRPYTTHISSTOPFORALANTURINGSTOPFORHIS
SUCCESSSTOPOFDECRYPTINGSTOPALLTHEGERMANSTOPMESSAGESTOPTHEFLAGISBOBRACESE
NIGMAFORALANTURINGBRACESONLYTHOSEWHOAREASLEEPDONTMAKEMISTAKESGETNOCRITI
QUESEEMSEVERYBODYSWORRIEDBOUTTHINGSTHATWEARETHINKINGANDWHENTHEREMEDYS
THEENEMYYOUHIDESELFDEPRECATIONUPYOURSLEEVEANDSELFSERVINGFRIENDSWHOLEAVEW
HENYOUARESINKING

BO1337{ENIGMAFORALANTURING}

REDPOINT



Probably one of the most ridiculous challenges. I spend hours trying to extract flag from the image and the file name. When I saw some people started getting the answer, I realize I was overthinking it. Basically, the flag is just what the arrow is pointing at which is the screwdriver.

BO1337{screwdriver}