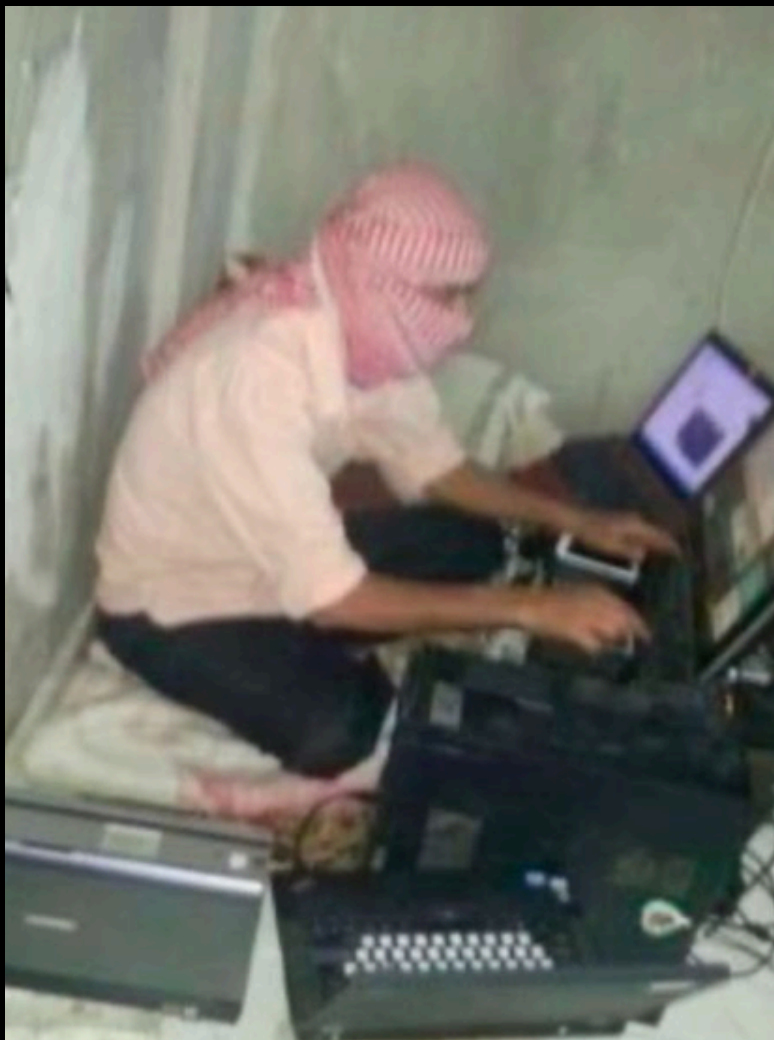


WRITEUP CTF FINDIT 2025  
Team *robot.txt*



Anggota :

Harri Supriadi  
Muhammad 'Azmi Salam  
Anas Miftakhul Falah

# Table of Contents

Findit 2025

—	Cryptography
	└─ caesar chipper
—	MISC
	└─ Absen
	└─ distorted
	└─ your-journey-2
	└─ cek-cek
—	Reverse
	└─ xor-madness
—	Web Exploitation
	└─ Simple Heist
	└─ PixelPlaza
—	OSINT
	└─ Destroyer

# Cryptography

## 1. caesar chiper

Challenge

114 Solves

✕


### caesar cipher

100

author: mojitodev

Pada suatu malam, Tung Tung Tung Tung Sahur ingin mendatangi seorang pemuda yang tidak bangun sahur setelah dipanggil sahur sebanyak 3 kali, tetapi tidak nyaut. Masalahnya adalah pintu kamar pemuda tersebut terkunci dengan password tertentu, tetapi terdapat file **cipher.txt** yang tersimpan dalam flashdisk di dekatnya yang bisa digunakan untuk menemukan passwordnya. Bantulah Tung Tung Tung Tung sahur untuk menemukan passwordnya!

author: **mojitodev**

 ciphertext...

Flag

Submit

Deskripsi :

Diberikan sebuah file berisi ciphertext, dimana saya perlu mendeskripsikan ciphertext tersebut untuk mendapatkan flagnya.

```
D: > Muhammad 'Azmi Salam > Hacking > Competition > Find IT > Find IT 2025 > Final > crypto > caesar > ≡ cipher.txt
1  Ymnx nx f xjhwjy ymj vzny htzw fyyjw. Qnkj ymj bnqq gj f xjhtsi bj bnqq gjfyyj,
2  jshwduynts ymj knwxy ts ymj xtrj tk ymj ufxxfrnsl gjktwj. Tzlm rjxxflj, ymj
3  htsyfsy tk ymj xtrj qnkj f hfjxw ns yjcy. Qjilmynts ymj jshwduy rjxxflj kwtr
4  f wjfi ymj rjxxflj yt ymj fxyjw. Rjxxflj xynsl ymnx KnsiNYHYK{Mrrrr_1_W89qqd_i5sy_pstb_Ym8_U5xxbtwi}
5  |
```

Analisis :

Setelah dianalisis, untuk mendapatkan flag nya saya cukup mengcopy bagian akhirnya saja karena sudah terlihat jelas itu sesuai dengan format flagnya.

Solusi :

Setelah mengcopy bagian akhir dari chipertext tersebut, kemudian saya men decrypt text tersebut menggunakan tools pada web <https://www.dcode.fr/caesar-cipher> dan dari sana saya bisa mendapatkan flagnya.

The screenshot shows the 'CAESAR CIPHER DECODER' interface. On the left, the 'Results' section displays the brute-force mode output, listing 25 possible shifts. The 5th result, a shift of +5 (↖21), is highlighted, showing the decrypted text: 'FindITCTF{Hmmmm\_1\_R89lly\_d5nt\_know\_Th8\_P5ssword}'. On the right, the 'CAESAR SHIFTED CIPHERTEXT' input field contains the ciphertext 'KnsiNYHYK{Mrrrr\_1\_w89qqd\_i5sy\_pstb\_Ym8\_U5xxbtwi}', and the 'Test all possible shifts (26-letter alphabet A-Z)' button is visible below it.

↕	↕
↗5 (↖21)	FindITCTF{Hmmmm_1_R89lly_d5nt_know_Th8_P5ssword}

Flag :

FindITCTF{Hmmmm\_1\_R89lly\_d5nt\_know\_Th8\_P5ssword}

# Misc

## 1. Absen

Challenge

111 Solves

×

Absen

100

ayok absen sebelum marathon ctf

Flag

Submit

Untuk mendapatkan flagnya kita cukup menyalin flagnya yang sudah diumumkan di discord karena ini adalah free flag.



Flag :

FindITCTF{absen\_adick\_adick}

## 2. distorted

Challenge

73 Solves

✕

### distorted

100

GAMBARNYA MLEYOTT. Setiap row bergeser 5 pixels lebih dari row sebelumnya. Gimana nih biar gambarnya kelihatan dan lokasinya bisa dicari?

- Format Flag:  
FindITCTF{Lintang\_Bujur\_Nama\_Tempat}
- case insensitive

author: [Azmi](#)

► Format Lokasi dan Koordinat (Cost: 0 points)

📄 location.p...

Submit

Deskripsi :

Diberikan sebuah file .png yang sudah distorted, setiap rownya bergeser sebanyak 5 pixels lebih dari row sebelumnya.



Analisis :

Untuk melakukan analisis lebih lanjut tentunya kita perlu memperbaiki foto tersebut agar kembali menjadi normal. Setelah itu, kita bisa lanjut melakukan pencarian dimana lokasi itu berada dan kita perlu mendapatkan titik koordinat dan nama tempatnya, kemudian menuliskannya sesuai format flag.

Solusi :

Untuk memperbaiki foto yang sudah didistorsi, kita bisa menggunakan script python seperti ini.

```
from PIL import Image
import numpy as np

img = Image.open("location.png")
pixels = np.array(img)

height, width, _ = pixels.shape

for i in range(height):
    shift = (i * 5) % width
    pixels[i] = np.roll(pixels[i], -shift, axis=0)

fixed_img = Image.fromarray(pixels)
fixed_img.save("fixed_image.png")
fixed_img.show()
```

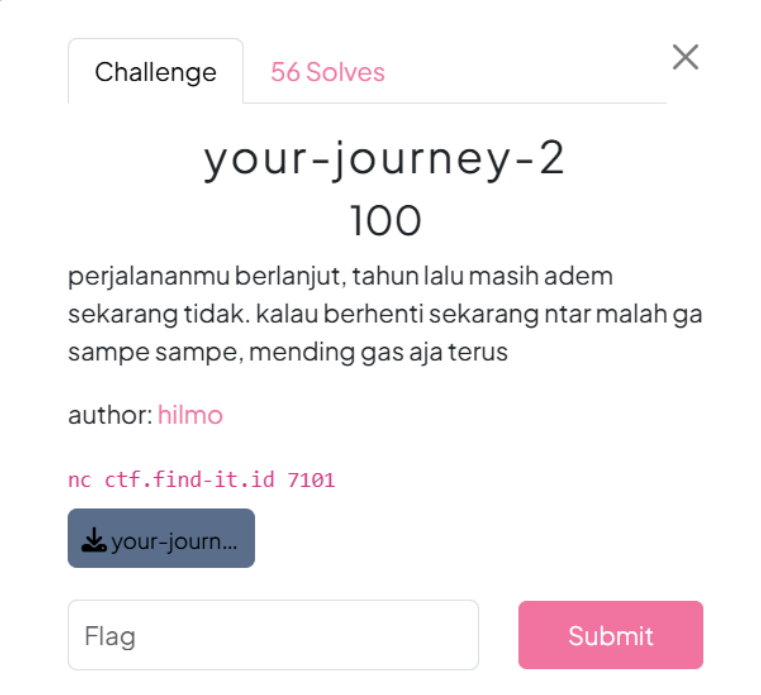
setelah itu, foto yang ditampilkan akan seperti ini.



Kita bisa langsung mendapatkan lokasi tempat tersebut menggunakan google lens, kemudian diketahui bahwa tempat tersebut berlokasi di Gereja Bethany Nginden. Setelah itu, saya menggunakan google maps untuk mengetahui titik koordinatnya dan didapatkanlah -7.306955671262183, 112.77254941881688 tapi sesuai formatnya kita hanya perlu mengambil 4 angka dibelakang koma dan nama lokasinya sesuai yang tertera di google (case sensitive). Maka dari sana kita sudah bisa mendapatkan flagnya.

Flag :  
FindITCTF{-7.3069\_112.7725\_Gereja\_Bethany\_Nginden}

### 3. your-journey-2



Challenge 56 Solves X

## your-journey-2

100

perjalananmu berlanjut, tahun lalu masih adem  
sekarang tidak. kalau berhenti sekarang ntar malah ga  
sampe sampe, mending gas aja terus

author: **hilmo**

nc ctf.find-it.id 7101

📄 your-journ...

Flag Submit

Deskripsi :

Diberikan attachment yang berisi main.py dan word.py, yang berisi alur program dan query yang akan diblokir.

Analisis :

Kita bisa mendapatkan flagnya dengan membuka file yang berada di direktori yang sama. Namun, untuk melakukan itu kita perlu menggunakan query import, tapi import adalah salah satu kata yang diblokir. Jadi, kita perlu memikirkan cara agar pemblokiran kata itu bisa di bypass.



Solusi :

Sebelum mencari cara untuk melakukan bypass, sebenarnya kita dapat melakukan educated guess, dalam beberapa soal CTF kebanyakan flag itu disimpan dalam file bernama flag.txt yang kebetulan dari kata flag.txt itu tidak ada kata yang bisa diblokir. Maka pada saat itu langkah pertama yang saya lakukan adalah menggunakan query ini

```

└─(root@Asoes36)─[/mnt/d/Muhammad 'Azmi Salam/Hacking/Competition/Find IT/Find IT 2025/Final/misc/your-journey-2]
└─# nc ctf.find-it.id 7101

```

Ayo kawan kita bersama  
menanam jagung di kebun kita  
ambil cangkulmu, ambil pangkurmu  
kita bekerja tak jemu-jemu  
cangkul, cangkul, cangkul yang dalam  
tanah yang longgar jagung kutanam

```

Hmm keknya ada yang salah sama lagunya, bukannya "Ayo Ayo Ganyang si b.e.b.a
.n 🌸"
$print(open('flag.txt').read())
FindITCTf{K0n0h4_h4nc4r_0l3h_0knum_p4in}
'NoneType' object is not callable

```

Oh tidak, kamu diserang "Kawan-kawan oknum"

di sana kita dapat melihat flag, akan tetapi pada saat saya submit flagnya, jawaban saya ter verdict incorrect, dan setelah saya fokus baca baca lagi flagnya, ternyata memang benar, pada FindITCTf itu f nya non-kapital, yang mana itu tidak sesuai format. Maka dari sana saya berasumsi kalau itu adalah fake flag. Setelah itu, saya melakukan cara lain tadi yaitu dengan melakukan bypass agar tidak diblokir, saya menggunakan query **exec(input())** untuk melakukan bypass dan saya melakukan import os untuk melihat di direktori ini ada file apa saja.

```
└─# nc ctf.find-it.id 7101
```

```
Ayo kawan kita bersama  
menanam jagung di kebun kita  
ambil cangkulmu, ambil pangkurmu  
kita bekerja tak jemu-jemu  
cangkul, cangkul, cangkul yang dalam  
tanah yang longgar jagung kutanam
```

```
Hmm keknya ada yang salah sama lagunya, bukannya "Ayo Ayo Ganyang si b.e.b.a  
.n 🌸"  
$exec(input())  
print(__import__('os').listdir('.'))  
['endingsatu', 'flag.txt', 'main.py', 'hidden.py', 'word.py', 'endingtiga',  
'endingdua']  
'NoneType' object is not callable
```



```
Oh tidak, kamu diserang "Kawan-kawan oknum"
```

di sana ada file hidden.py, awalnya saya kira flagnya akan disimpan disana dan ketika saya coba untuk print, begini hasilnya.

```
Find IT 2025/Final/misc/your-journey-2]
```

```
└─# nc ctf.find-it.id 7101
```

```
Ayo kawan kita bersama  
menanam jagung di kebun kita  
ambil cangkulmu, ambil pangkurmu  
kita bekerja tak jemu-jemu  
cangkul, cangkul, cangkul yang dalam  
tanah yang longgar jagung kutanam
```

```
Hmm keknya ada yang salah sama lagunya, bukannya "Ayo Ayo Ganyang si b.e.b.a  
.n 🌸"  
$exec(input())  
print(open('hidden.py').read())  
import os
```

```
FLAG = "FIndITCTF{y0u_f0und_1t!_or_d1d_y0u?}"
```

```
def viewfolder(path: str):  
    files = os.listdir(path)  
    for file in files:  
        print(file)
```

```
'NoneType' object is not callable
```



```
Oh tidak, kamu diserang "Kawan-kawan oknum"
```


dan setelah saya baca itu penulisan flagnya tidak sesuai format, maka saya anggap itu fake flag lagi. Di sini saya berasumsi bahwa akan ada banyak fake flag, maka saya harus mengandalkan ilmu dukun untuk mempercepat proses pencarian flag.

Dapat kita lihat di direktori tadi ada 3 folder yang bernama 'endingsatu', 'endingdua', dan 'endingtiga'. awalnya saya sempat berasumsi bahwa flagnya akan terpotong-potong, tapi saya pun mencoba membuka file yang ada di 'endingdua' terlebih dahulu (karena feeling saja).

```
(root@Asoes36)-[mnt/d/Muhammad 'Azmi Salam/Hacking/Competition/Find IT/Find IT 2025/Final/misc]
# nc ctf.find-it.id 7101

Ayo kawan kita bersama
menanam jagung di kebun kita
ambil cangkulmu, ambil pangkurmu
kita bekerja tak jemu-jemu
cangkul, cangkul, cangkul yang dalam
tanah yang longgar jagung kutanam

Hmm keknya ada yang salah sama lagunya, bukannya "Ayo Ayo Ganyang si b.e.b.a
.n 🌸"
$print(open('endingdua/flag.txt').read())
FindITCTF{k0n0h4_m4ju_m4sy4r4k4t_m4kmur}
'NoneType' object is not callable
```



```
Oh tidak, kamu diserang "Kawan-kawan oknum"
```

```
(root@Asoes36)-[mnt/d/Muhammad 'Azmi Salam/Hacking/Competition/Find IT/Find IT 2025/Final/misc]
# |
```

dan ternyata sudah ada flag utuh dengan format yang benar pula. Maka saya langsung submit flag tersebut dan mendapatkan verdict correct.

```
Flag :
FindITCTF{k0n0h4 m4ju m4sy4r4k4t m4kmur}
```

#### 4. cek-cek



Deskripsi :

Diberikan attachment file main.py

Analisis :

Untuk mendapatkan flagnya kita bisa membuka file flag.txt nya saja di direktori yang sama, ini mirip mirip seperti soal your-journey-2, namun bedanya disini kata **flag** dan karakter `.` itu diblokir jadi kita tidak bisa menggunakan cara yang tadi. Selain itu, jika kita memilih opsi 2, kita akan diberikan flag nya tapi yang sudah dienkripsi dalam bentuk hash.

Solusi :

Sebelum melakukan analisis pada hashnya, saya mencoba cara yang lebih sederhana terlebih dahulu, yaitu membuka file-file yang ada di direktori tersebut dengan query `/proc/self/fd/` siapa tau flagnya akan bisa langsung didapatkan dengan cara itu, dan setelah 5 kali percobaan ternyata saya bisa langsung mendapatkan flagnya (hoki menn pake ilmu dukun).

```
(root@Asoes36)-[/mnt/d/Muhammad 'Azmi Salam/Hacking/Competition/Find IT/Find IT 2025/Final/misc/cekcek]
# nc ctf.find-it.id 7001
Do you want check my file?
1. yes
2. no
>>> 1
file name: /proc/self/fd/3
error bang
Do you want check my file?
1. yes
2. no
>>> 1
file name: /proc/self/fd/4
error bang
Do you want check my file?
1. yes
2. no
>>> 1
file name: /proc/self/fd/5
FindITCTF{cl0s3_y0ur_f1l3s_1mmed14t3ly_0r_w0w0_w1ll_f1nd_y0u}
Do you want check my file?
1. yes
2. no
>>> |
```

Flag :

FindITCTF{cl0s3\_y0ur\_f1l3s\_1mmed14t3ly\_0r\_w0w0\_w1ll\_f1nd\_y0u}

# Reverse

## 1. xor Madness

Challenge

107 Solves

✕

### xor Madness

100

Bombombini Gusini adalah seorang mahasiswa tahun pertama jurusan Teknologi Informasi yang tengah mendalami cryptography dan malware analysis di mata kuliah Peretasan Beretika. Suatu hari, dosen memberikan tugas berupa sebuah binary file bernama xor\_madness.bin. Katanya jika ia berhasil mendapatkan "sesuatu" dari binary file tersebut, maka ia akan langsung mendapatkan nilai A. Bantulah ia untuk bisa mendapatkan "sesuatu" tersebut.

author: [mojitodev](#)

 xor\_madn...

Submit

Deskripsi :

Diberikan attachment berupa file binary yang tidak bisa dibaca semua langsung, kalau dibuka di notepad++ akan jadi seperti ini.

```
1  Uz}wZGPGUhzj'Lq } aL"}"LuDEL'tL}j'Lq'}tn
```

Analisis :

Karakter DEL itu dapat kita ubah menjadi hexadesimal x7f karena itu memiliki nilai ASCII 127. Setelah itu, kita dapat me reverse text tersebut menggunakan mesin bruteforce yang sudah saya buat.

Solusi :

Masukan text hexadecimal tersebut ke dalam script, kemudian jalankan scriptnya, darisana kita bisa mendapatkan flagnya.

```
cipher = b"Uz}wZGPGUhZj'Lq } aL\""}\"Lu\\x7f'tL}j'Lq'}tn"

for key in range(256):
    plain = ''.join(chr(c ^ key) for c in cipher)
    if all(32 <= ord(c) <= 126 for c in plain):
        print(f"Key {key:02x}: {plain}")
```

```
(root@Asoes36)-[/mnt/d/Muhammad 'Azmi Salam/Hacking/
# python slv.py
Key 01: T{lv[FQFTi{k&Mp!|!`M#|#Mt~&uM|k&Mp&|uo
Key 03: Vy~tYDSQVkyi$Or#~#b0!~!Ov|$wO~i$Or$~wm
Key 04: Q~ys^CTCQl~n#Hu$y$eH&y&Hq{#pHyn#Hu#ypj
Key 06: S|{q\AVASn|l!Jw&{&gJ${$Jsy!rJ{l!Jw!{rh
Key 07: R}zp]@W@Ro}m Kv'z'fK%z%Krx sKzm Kv zsi
Key 09: \st~SNYN\asc.Ex)t)hE+t+E|v.}Etc.Ex.t}g
Key 0c: Yvq{VK\KYdvf+@},q,m@.q.@ys+x@qf+@}+qxb
Key 0d: XwpzWJ]JXewg*A|-p~LA/p/Axr*yApg*A|*pyc
Key 0f: ZurxUH_HZgue(C~/r/nC-r-Czp({Cre(C~(r{a
Key 10: EjmgJW@WExjz7\ao0mq\2m2\eo7d\mz7\ao7md~
Key 12: GhoeHUBUGzhx5^c2o2s^0o0^gm5f^ox5^c5of|
Key 13: FindITCTF{iy4_b3n3r_1n1_fl4g_ny4_b4ng}
Key 14: AnicNSDSA|n~3Xe4i4uX6i6Xak3`Xi~3Xe3i`z
Key 16: ClkaLQFQC~l|1Zg6k6wZ4k4Zci1bZk|1Zg1kbx
Key 18: MbeoB_H_Mpbr?Ti8e8yT:e:Tmg?lTer?Ti?elv
Key 19: LcdnC^I^Lqcs>Uh9d9xU;d;Ulf>mUds>Uh>dmw
Key 1a: O`gm@]J]Or`p=Vk:g:{V8g8Voe=nVgp=Vk=gnt
Key 1b: NafLA\K\Nsaq<Wj;f;zW9f9Wnd<oWfq<Wj<fou
Key 1c: IfakF[L[Itfv;Pm<a<}P>a>Pic;hPav;Pm;ahr
Key 1d: Hg`jGZMZHugw:QL=`=|Q?`?Qhb:iQ`w:QL:`is
Key 1f: JebhEXOXJweu8Sn?b?~S=b=Sj`8kSbu8Sn8bkq
```

```
(root@Asoes36)-[/mnt/d/Muhammad 'Azmi Salam/Hacking/
```

Flag :

FindITCTF{iy4\_b3n3r\_1n1\_fl4g\_ny4\_b4ng}