

WRITE UP JUNIOR WRECK-IT 5.0

ZIP.SH



Mamo

Zero

Bunz.id

DAFTAR ISI

WRITE UP PENYISIHAN JUNIOR WRECKIT^{5.0}/

|—— MISC/

| |—— Free_Flag/

| |—— README.md

|—— CRYPTO/

| |—— hum45/

| |—— README.md

| |—— MatPem/

| |—— chall.py

| |—— README.md

|—— FORENSIC/

| |—— Broken/

| |—— 100.zip

| |—— README.md

|—— REVERSE ENGINEERING/

| |—— Babysnake/

| |—— chall.pyc

| |—— README.md

| |—— Lets_Go/


| |—— dist.rar

| |—— README.md

|—— README.MD

MISC

FREE FLAG

CHALLENGE **56 SOLVES** 

Free Flag

1

WRECKIT50{just_ch3cking_f0r_y0ur_sani7y}


Ini merupakan format flag daripada challenge WRECK-IT^{5.0}.

Jadi ini merupakan flag gratis untuk memulai, bisa dikatakan ini merupakan Sanity Checking.

This your Flag: WRECKIT50{just_ch3cking_f0r_y0ur_sani7y}

CRYPTO

hum45

CHALLENGE 27 SOLVES 

hum45

100

can you solve this with no clue???

RH95N9U34E+91C9Y34CY80095IAYX9T6ASNAK*9QY9 S9BT90B9SNABT9

format: WRECKIT50{????????}

2/10 attempts

Flag

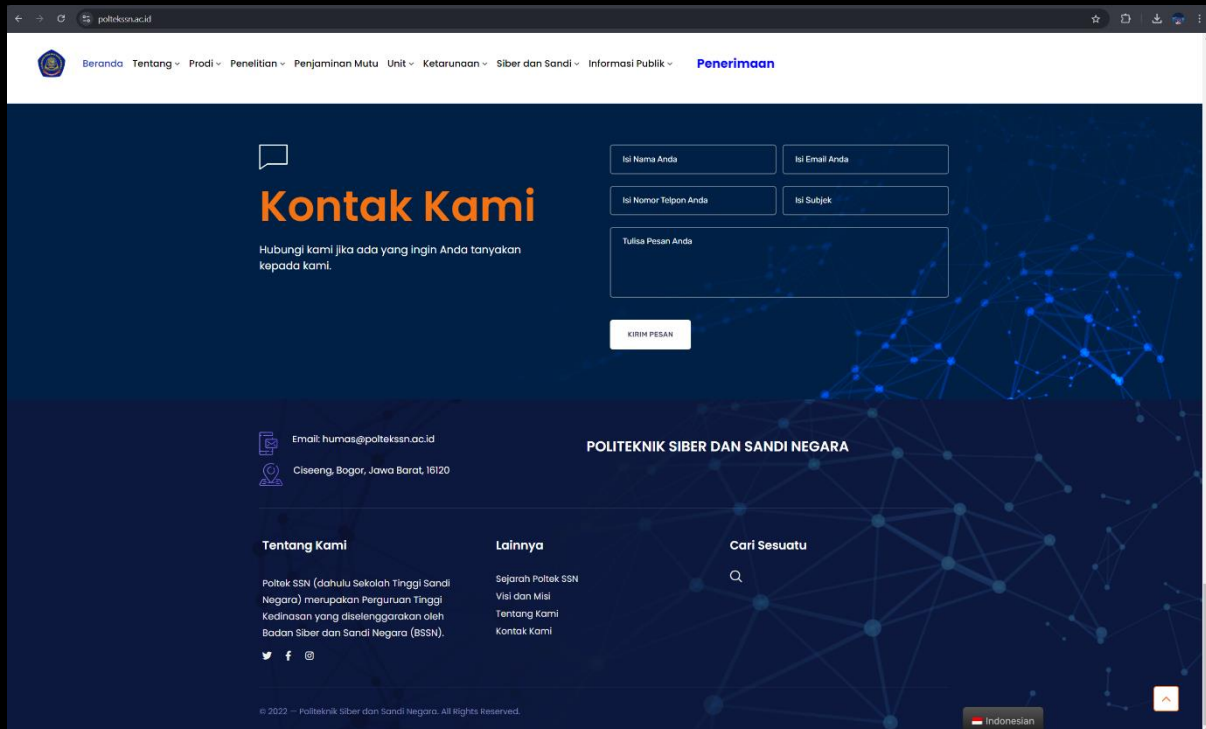
Submit

```
RH95N9U34E+91C9Y34CY80095IAYX9T6ASNAK*9QY9
S9BT90B9SNABT9.HAK*9YCBE+9*34HY8UY9W34EY8DB8MA85N9$Y9IZAW34MB8
CB9W34BY8NB88B8 :7W0
```

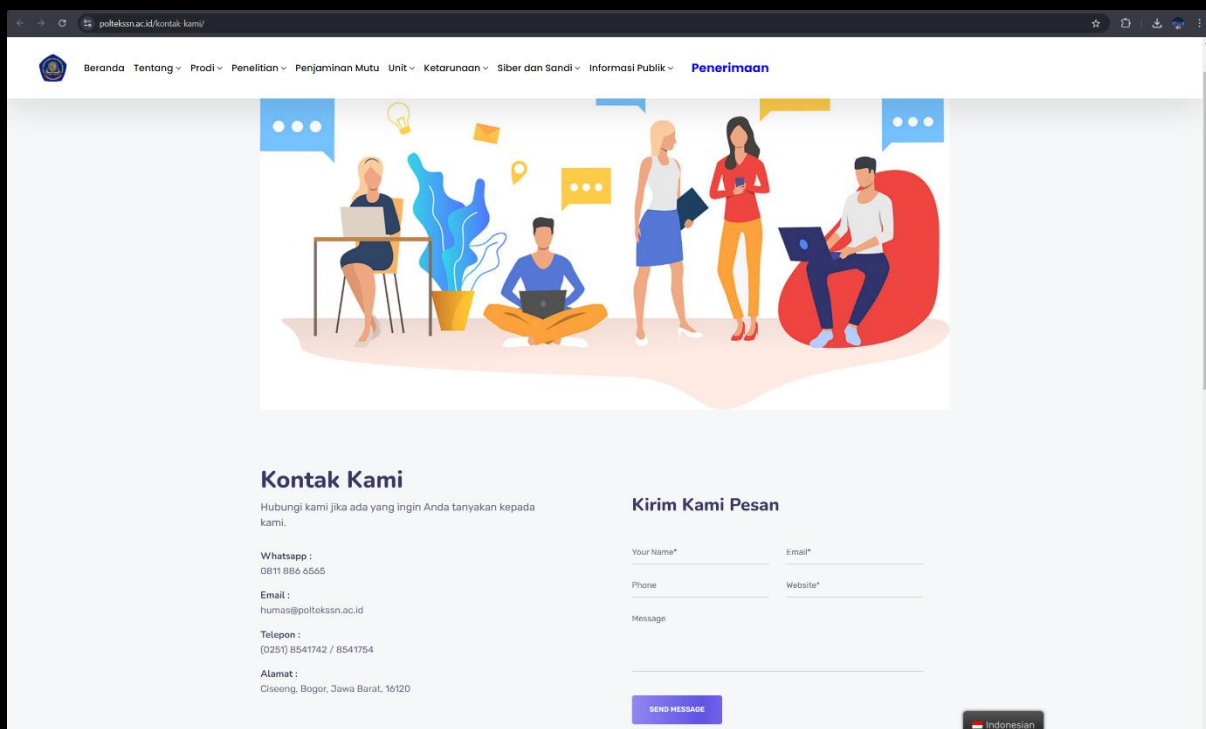
Jika kita melihat judul yaitu 'hum-45' dan disertai dengan simbol-simbol dapat disimpulkan ini merupakan enkripsi basis-45. Mari kita kunjungi [<https://gchq.github.io/CyberChef/>] atau dapat menggunakan alat dekripsi lainnya. Disini kita memberikan recipe "From Base45", teks dekripsi dari teks chipper tersebut muncul dan berisikan pesan berikut:

```
JIKA INGIN MENGIRIM PESAN MELALUI SALURAN YANG RESMI KEPADA
KAMPUS KAMI, KEMANA???
```

Dengan kata kunci tersebut kita dapat melihat kembali kejudul dari tantangan ini "hum45" yang dapat dibaca "humas". Mari kita mengunjungi halaman situs web dari Poltek SSN [<https://poltekssn.ac.id/>]. Pada halaman beranda, jika kalian menggulir layar kebawah hingga mencapai Footer dari halaman situs web tersebut maka kalian akan menemukan kontak dari humas yang berupa alamat email



Kalian juga dapat menemukan hal yang serupa pada halaman [<https://poltekssn.ac.id/kontak-kami/>]



Untuk lebih mudahnya kalian dapat mengunjungi Instagram dari Poltek SSN [<https://www.instagram.com/poltekssn/>] dan melakukan akses ketautan [<https://linktr.ee/poltekssn>]

This your Flag: WRECKIT50{humas@poltekssn.ac.id}

巾ムナアモ巾

CHALLENGE 26 SOLVES 

MatPem

100

SPLXV

 chall.py

Submit

Pada attachment diberikan chall.py seperti dibawah ini

Chall.py

```
import random

FLAG = b"WRECKIT50{????????}"
fint = int(FLAG.hex(),16)
key = [random.getrandbits(4) for _ in range(3)]

pk = [
    [11,14,17,20], [12,15,18,21], [13,16,19,22]
]

result = [sum([key[j]*pk[i][j] for j in range(3)])
for i in range(3)]
```

```

var = ['a','b','c','d']
for i in range(len(pk)):
    equation = ''
    for j in range(len(pk[i])):
        equation += str(pk[i][j])+"*"+var[j]+" + "
    equation = equation[:-3] + " = " + str(result[i])
    print(equation)

key = sum(key)

enc = key*fint
print("Encryted flag:", enc)

```

''' '''

Output:

11*a + 14*b + 17*c + 20*d = 263

12*a + 15*b + 18*c + 21*d = 282

13*a + 16*b + 19*c + 22*d = 301

Encryted flag:

23224837378070255855973270563432031032463911182435722
4567527709756665492238132012558072443413580231257415

''' '''

Setelah melihat chall.py diatas dan melihat hint yang tersedia pada soal, kita dapat menyimpulkan bahwa ini adalah soal untuk menyelesaikan Persamaan Linear.

Kita memiliki tiga persamaan linear yang perlu diselesaikan untuk menemukan nilai a, b, dan c.

Persamaan tersebut adalah:

$$11a+14b+17c=263$$

$$12a+15b+18c=282$$

$$13a+16b+19c=301$$

Sistem persamaan ini memiliki rank kurang dari jumlah variabel, jadi kita tidak bisa menyelesaikannya langsung.

Jadi karena nilai kunci adalah nilai 4-bit (0 sampai 15), kita mencoba semua kombinasi nilai untuk menemukan yang cocok dengan persamaan yang diberikan.

Ini adalah solver yang digunakan untuk soal ini:


[<https://pastebin.com/JEE7A56V>]

```
mamo .../wreckit/mathpem ♡ 20:41 python test.py
Possible key: (3, 14, 2)
Decrypted flag: WRECKIT50{5ist3m_PrSm44n_l1n13r_4_vaRiabEL}
```

This your Flag: WRECKIT50{5ist3m_PrSm44n_l1n13r_4_vaRiabEL}

FORENSIC


乃灰口ケモ口

CHALLENGE 41 SOLVES 

broken
100

rusak rusak rusakkkkk

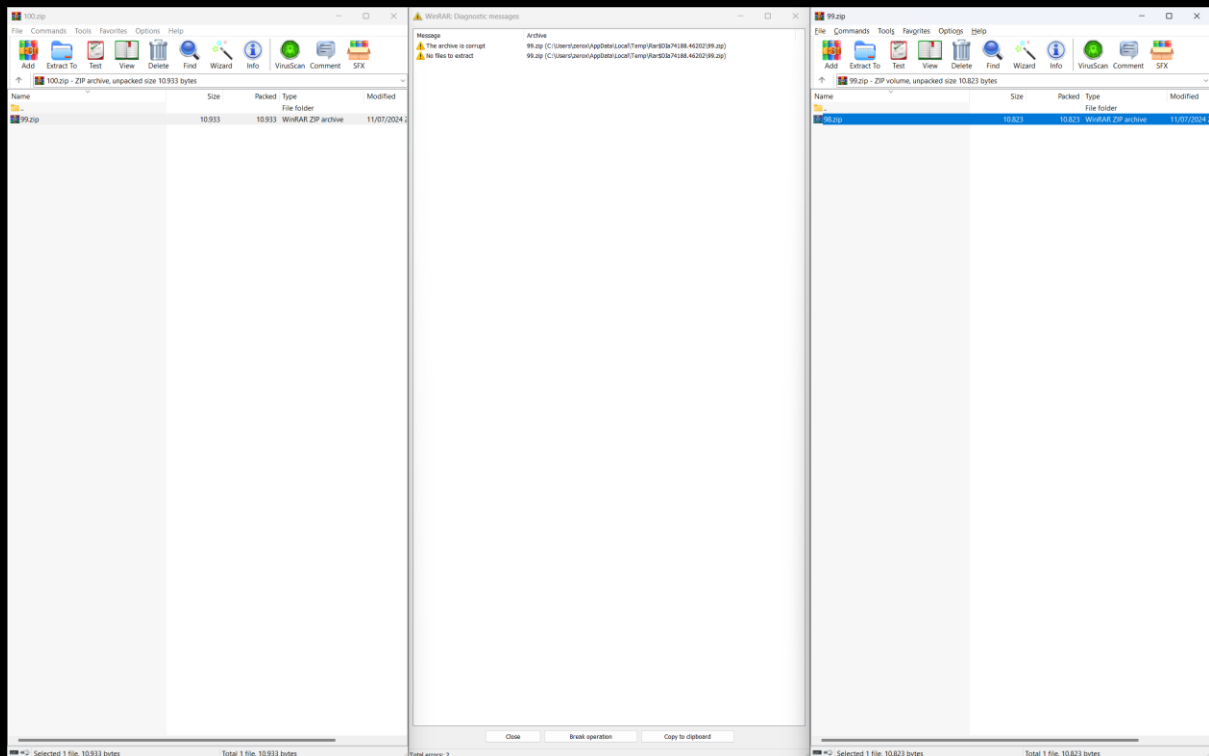
author: k.eii

 100.zip

Flag

Submit

Disini kita diberikan sebuah lampiran berupa file dengan format .zip dengan nama "100.zip". Untuk petunjuk yang kita miliki bahwa file ini merupakan file rusak atau corrupt. Mari kita tes, disini saya mencoba unzip menggunakan WinRAR.



Nah dari aksi kita ini dapat disimpulkan file tersebut rusak. Jika kalian berpikir untuk melakukan repair archive kalian salah besar hal ini dapat dilakukan dengan perintah `$strings [nama file]` dalam kasus ini dapat langsung kita pergunakan dari awal file tersebut diunduh `$strings 100.zip | grep WRECK`.

This your Flag:

`WRECKIT50{huhuhaha_hohohihe_gua_capek_ngezipnya_heheh_huhu_ez_lah}`

REVERSE ENGINEERING

乃ム乃ソヲ乃ムケモ

CHALLENGE

21 SOLVES


✕

Babysnake

100

silahkan puh, babyrevnya puh

author: k.eii

 chall.pyc

Flag

Submit

Pada challenge ini diberikan sebuah file dengan nama main dengan ekstensi .pyc, .pyc ini sepertinya asing di telinga kita semua, mari ulik dulu apa itu .pyc File dengan ekstensi .pyc adalah file Python yang telah dikompilasi. Berbeda dari file.py yang berisi kode sumber Python dalam format teks, file .pyc berisi byte code yang telah dikompilasi, yang merupakan bentuk setengah jadi dari kode sumber dan siap untuk dijalankan oleh Python interpreter. Byte code ini lebih cepat untuk dijalankan dibandingkan dengan menginterpretasi kode sumber setiap kali program dijalankan.

Dalam konteks CTF (Capture The Flag) khususnya pada kategori reverse engineering, file .pyc sering muncul sebagai tantangan. Peserta CTF diharapkan untuk melakukan reverse engineering pada file .pyc untuk memahami bagaimana program tersebut bekerja,

menemukan kerentanan, atau mendapatkan flag yang merupakan tujuan dari tantangan tersebut. Proses ini biasanya melibatkan dekompilasi file .pyc kembali menjadi kode sumber Python yang lebih mudah dibaca dan dipahami, menggunakan alat seperti Decompyle atau Uncompyle, dan mungkin melibatkan analisis lebih lanjut

Jadi Ketika saya coba untuk mendecompilenya menggunakan online tools:

Decompile results

```
8 from base64 import b64encode as b64e, b64decode as b64d
9
10 def xor(data):
11     hasil = []
12     for i, val in enumerate(data):
13         shifted = (val ^ i) << i % 8 | (val ^ i) >> 8 - i % 8
14         hasil.append(shifted & 255)
15     else:
16         return hasil
17
18
19 usr_input = input(">>> ")
20 usr_input = usr_input.encode()
21 enc = b64e(usr_input).decode()
22 mis_pad = len(enc) % 4
23 if mis_pad:
24     enc += "=" * (4 - mis_pad)
25 else:
26     dec = b64d(enc)
27     apani = xor(dec)
28     apatuh = [87, 166, 29, 2, 244, 137, 148, 25, 56, 228, 161, 249, 230, 142,
29              84, 191, 105, 202, 233, 25, 167, 73, 93, 147, 117, 210, 172,
30              187, 151, 47, 80, 62, 16, 138, 68, 242]
31     if apani == apatuh:
32         print("Nais!")
33     else:
34         print("Coba lagi!")
```

Dapat dilihat bahwa program tersebut melakukan operasi XOR pada flag yang sudah di encode menggunakan Base64, jadi saya coba untuk mereverse nya menggunakan solver berikut:

[\[https://pastebin.com/M37PNA8P\]](https://pastebin.com/M37PNA8P)

```
mamo .../wreckit/babypythob 20:30 python test.py
Error decoding base64: Incorrect padding
Original bytes: b'WRECKIT50{b4by_pyth0n_c0mp1led_c0d3}'
```

Ketika saya menjalankannya, walaupun paddingnya salah tapii, yaudah lah ya, yang penting flag.

This your Flag: WRECKIT50{b4by_pyth0n_c0mp1led_c0d3}


lets go

CHALLENGE **22 SOLVES**

Lets Go
100

lets goooo

author: k.eii

 dist.rar

Flag

Submit

Diberikan dua file yaitu flagenc dan file elf wreck1t, Ketika saya mencoba untuk menjalankan file nya, ia memberitahu bagaimana cara program tersebut bekerja

```
ChatGPT return hasil
mamo .../wreckit/lest-go 20:13 ./wreck1t
Usage: <input-file> <output-file>

mamo .../wreckit/lest-go 20:15
Corrupt ZIP File Repair

usr_input = input(">>> ")
usr_input = usr_input.encode()
enc = b64e(usr_input).decode()
mis_pad = len(enc) % 4
```

Dapat dilihat bahwa program tersebut membutuhkan 2 parameter yaitu input-file dan output-file, sebelum saya menganalisisnya lebih dalam, saya mencoba untuk langsung memasukkan flagenc sebagai input-file dan lihat apa yang terjadi

```
mamo ~/wreckit/lest-go ♡ 20:15 ./wreckit flagenc wkwk
Encryption successful!

mamo ~/wreckit/lest-go ♡ 20:17 cat wkwk
WRECKIT50{g0_wrek_1t_ye}
```

Dan Voila! Kita dapat flagnya

This your Flag: WRECKIT50{g0_wrek_1t_ye}

README.MD

<https://fontsgreek.com/xenara-font>

<https://fontzone.net/font-details/courier-new>

<https://www.dafont.com/electroharmonix.font>

<https://www.dafont.com/made-evolve-sans.font>

<https://www.fontspace.com/hacked-font-f28425>

THANKS TO ALL

CYBERSTORM, TCP1P, HUAWEI, SNI, PSSN

BIG THANKS

WRECK-IT^{5.0}