WU FINAL TechnoFairCTF 2023

Tim Moai 🗿

Beluga
Kiinzu
Wrth

Cryptography

- Sekte Pemuja Osha Oshi
  - **TecnoFairCTF{Sekte_Pemuda_Pemuja_0sh4_0sh1}**
- Jurus Rahasia Teleport
  - **TechnoFairCTF{H1ngg@_d1_@ngk@5@_5@n@_D1_kej@uh@n_pertemu@n_@nt@r@_k1t@_berdu@_Temp@t_y@ng_t@k_@d@_51@p@pun_H155@t5u_telep0-0rt0}**

Web Exploitation

- WordPress Lover
  - **TechnoFairCTF{sourc3_code_review_on_wordpress_plugin_is_fun_r1ght_you_c4n_g3t_CVE_from_th1s_activity}**

Reverse Engineering

- Kokekokko
  - **TechnoFairCTF{1nu_w4nw4n_k43ru_k3r0k3r0_k1tsun3_R1ng_d1ng_d1ng_d1ng_d1ng3r1ng3d1ng}**

Misc

- Finish Him
  - **TechnoFairCTF{Bentrok_CJ_jirr}**
- Kang Spam Sticker
  - **TechnoFairCTF{halo_dek}**

Cryptography

Sekte Pemuja Osha Oshi

Diberikan script python berikut beserta outputnya:

```
from Crypto.Util.number import *
from sage.all import *
import random
def encrypt(p):
    p=bin(p)[2:]
    p='0'*(len(p)%64)+p
    p=[int(p[i:i+16],2) for i in range(0,len(p),16)]
```

```
    key=random.getrandbits(16)
    p=[i^key for i in p]
    return p

flag=b'REDACTED'
p=getPrime(1024)
q=getPrime(1024)
e=0x10001
n=p*q

temp=p&((1<<37)-1)
p>>=81
p<<=81
p+=temp
p=encrypt(p)

flag=bytes_to_long(flag)
enc=pow(flag,e,n)
print(f'enc : {enc}')
print(f'n : {n}')
print(f'e : {e}')
print(f'leaked_p :{p}')
```

Terdapat leaked p 1024 bit dengan 1024-81 bits msb diketahui dan 37 bits lsb diketahui, sehingga ada 44 bits yang hilang, tapi leaked p nya ini di encrypt dulu dengan cara dibagi per 16 bit lalu di xor dengan suatu key.

Keynya ini ngga perlu di bruteforce, ada 44 bits yang isinya 0, sehingga teorinya ada 2 block 16 bits yang isinya 0 semua, sehingga hasil encryptnya adalah 0 ^ key = key,
Kalau diperhatikan outputnya ini ada 2 block yang identik diujung, jadi ini bisa kita asumsikan block yang isinya 0 semua yang kita cari.

```
14671, 16685, 16685, 16698, 34390, 41472]
```

```
XXXXXXXXXXXXXXXX ^ key = something
0000000000000000 ^ key = key (16685)
```

```
0000000000000000 ^ key = key (16685)
00000XXXXXXXXXX ^ key = something else
```

```
#sage
leaked_p =[45076, 36169, 27950, 3563, 58188, 12614, 34400, 51608, 49317, 7186...
key = 16685
for i in range(len(leaked_p)):
    leaked_p[i] ^^= key

p = 0
for i in leaked_p:
    p <<= 16
    p += i
```

Nah setelah dapat key dan leaked p aslinya tinggal kita faktorisasi coppersmith factorization saja, saya menggunakan referensi disini yang menggunakan implementasi dari sini.

```
#sage
enc = 2900475475260289...
n = 29353585191166156442189968012744774406...
e = 65537
leaked_p =[45076, 36169, 27950, 3563, 58188, 12614, 34400...
key = 16685
for i in range(len(leaked_p)):
    leaked_p[i] ^^= key

p = 0
for i in leaked_p:
    p <<= 16
    p += i

N = Integer(n)
def solve(lo, hi):
    knownbits = lo + hi
    sizep = 1024
    real_p = Integer(p)
    R = 2 ** lo
```

```
    invR = inverse_mod(R, N)

    P.<x> = PolynomialRing(Zmod(N))
    f = x + (real_p) * invR
    x0 = f.small_roots(X=2^(sizep-(knownbits))-1, beta=0.44, epsilon=1/16)[0]
    ans = Integer(x0 * R) + real_p
    return ans

p = solve(37, 1024-81)
q = n // p
phi = (p-1)*(q-1)
d = inverse_mod(e, phi)
flag = pow(enc, d, n)
print(bytes.fromhex(hex(flag)[2:]))
```

```
└─$ sage solveosha.sage
b'flagnya adalah TecnoFairCTF{Sekte_Pemuda_Pemuja_0sh4_0sh1}'
```

Flag: **TecnoFairCTF{Sekte_Pemuda_Pemuja_0sh4_0sh1}**
Emang typo kayaknya flagnya

Jurus Rahasia Teleport

Diberikan script berikut:

```
from pipit_generator import generator
from secret import flag
gen=generator()
chance=4
while chance>0:
    chance-=1
    print(f"tebak angka pipit selanjutnya !")
    inp_user=input(f"$ ")
    x_pred,y_pred=inp_user.split(',')
    x_pred,y_pred=int(x_pred),int(y_pred)
    x,y=gen.next(),gen.next()
    if(x_pred==x and y_pred==y):
        print(f"Pipit berhasil ditangkap! ini untukmu! {flag}")
    else:
        print(f"lokasi yang kamu berikan salah!")
        print(f"Pipit ditemukan pada posisi koordinat {x},{y}")
```

pipit_generator.py:

```
from random import getrandbits
from Crypto.Util.number import *
from sage.all import *
class generator():
    def __init__(self):
        self.a=getrandbits(64)
        self.c=getrandbits(64)
        self.m=getPrime(64)
        self.seed=getrandbits(64)%self.m
        U=[]
        for i in range(64):
            U.append([])
            for j in range(64):
                if j==i+1:
                    U[i].append(1)
```

```
                else:
                    U[i].append(0)
        U=Matrix(GF(2),U)
        L=[]
        for i in range(64):
            L.append([])
            for j in range(64):
                if j==i-1:
                    L[i].append(1)
                else:
                    L[i].append(0)
        L=Matrix(GF(2),L)
        self.U=U
        self.L=L

    def next(self):
        res=self.seed
        res='{:064b}'.format(res)
        res=vector(GF(2),list(res))
        res=res+((self.U**13)*res)
        res=res+((self.L**21)*res)
        res=''.join([str(i) for i in res])
        res=int(res,2)
        self.seed=(self.a*self.seed+self.c)%self.m
        return res
```

Pada dasarnya ini adalah chall predict lcg, dilihat dari self.seed=(self.a*self.seed+self.c)%self.m dibagian akhir, nah kita diberikan 4 percobaan, karena 1 nya harus benar buat dapat flag berarti hanya 3 kali leak saja, per leak ada x dan y, total 6 state sehingga feasible buat kita recover

Nah yang jadi masalah adalah seednya ini di encrypt macam macam sehingga kita harus mengembalikannya menjadi seed asli dulu. Kalau diperhatikan terdapat 2 matrix tambahan, U dan L, U itu adalah matrix 64x64 yang ada diagonal atasnya isinya 1, dan L itu yang diagonal dibawahnya itu isinya 1, kira-kira seperti ini

```
U           L
0 1 0 0 0   0 0 0 0 0
0 0 1 0 0   1 0 0 0 0
0 0 0 1 0   0 1 0 0 0
0 0 0 0 1   0 0 1 0 0
```

```
0 0 0 0 0   0 0 0 1 0
```

Nah apa yang terjadi kalau matrix ini dipangkat lalu dikalikan dengan seed kita? Mari kita lihat dengan menambahkan beberapa print

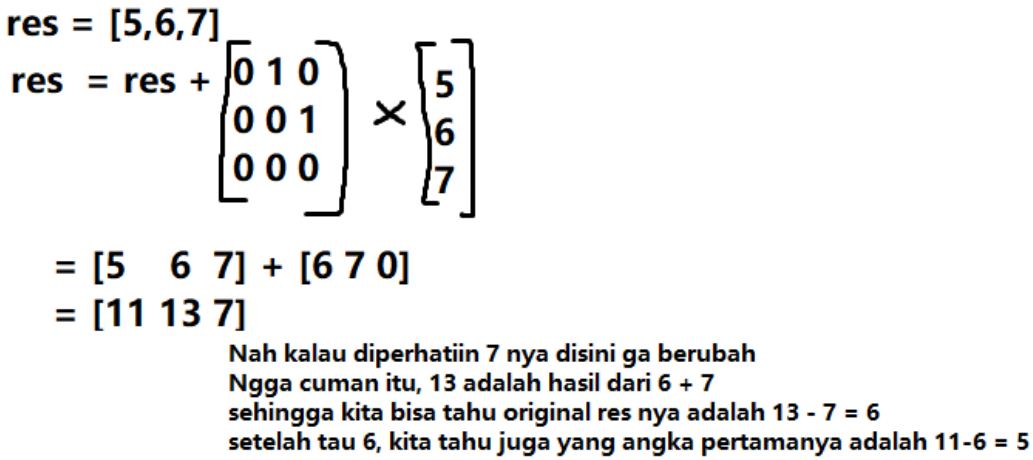
```
    def next(self):
        res=self.seed
        res='{:064b}'.format(res)
        res=vector(GF(2),list(res))
        print(''.join([str(i) for i in res]))
        res=res+((self.U**13)*res)
        print(''.join([str(i) for i in res]))
        res=res+((self.L**21)*res)
        print(''.join([str(i) for i in res]))
        res=''.join([str(i) for i in res])
        res=int(res,2)
        self.seed=(self.a*self.seed+self.c)%self.m
        return res
```

```
└─$ python3 jurusrahasia.py
tebak angka pipit selanjutnya !
$ 1,1
0001011010000011010100011001110000010001110010001111100011000111
0111110010110000110100111010010100001110110100000001100011000111
0111110010110000110100000100000010001000010011010011000010110001
0100000011111001000000001011100111101111111010000001100000110101
0110000011101110001111010100010011101100111011101011100000110101
0110000011101110001111100100001110011101000001001001111101010010
lokasi yang kamu berikan salah!
Pipit ditemukan pada posisi koordinat 8984910232187383985,6984588532216274770
tebak angka pipit selanjutnya !
$
```

Perhatikan baik baik

```
X
0001011010000011010100011001110000010001110010001111100011000111
0111110010110000110100111010010100001110110100000001100011000111
0111110010110000110100000100000010001000010011010011000010110001

Y
0100000011111001000000001011100111101111111010000001100000110101
0110000011101110001111010100010011101100111011101011100000110101
0110000011101110001111100100001110011101000001001001111101010010
```

Saat res += U**13 * res, kita bisa lihat 13 bit lsb nya tidak berubah, lebih dari itu, hasil dari res[n] dengan n < 64-13 adalah res sebelumnya + res[n+13], mohon maap kalau penjelasannya agak gajelas, saya coba visualisasiin aja



Nah sekarang bayangin aja dengan U**13, berarti nanti 13 bit lsb nya tidak akan berubah, lalu bit ke x originalnya adalah res[x] - res[x+13], hal yang sama juga terjadi di L**21 tapi sekarang msb nya yang ga berubah

```
state = []
for i in range(3):
    r.sendlineafter(b'$ ', b'0,0')
    r.recvuntil(b'Pipit ditemukan pada posisi koordinat')
    x, y = eval(r.recvline())
    x = list(map(int, bin(x)[2:].zfill(64)))
    y = list(map(int, bin(y)[2:].zfill(64)))
    for i in range(21, len(x)):
        x[i] ^= x[i-21]
        y[i] ^= y[i-21]
    for i in range(len(x)-13-1, -1, -1):
        x[i] ^= x[i+13]
        y[i] ^= y[i+13]
    x = int(''.join(map(str, x)), 2)
    y = int(''.join(map(str, y)), 2)
    print("pred", x, y)
    state.append(x)
    state.append(y)
```

Disini pake xor karena sama aja sebenarnya karena matrixnya di GF(2)

Setelah dapet statenya, tinggal crack lcg nya, saya pakai implementasi disini, dan tinggal simulasiin buat predict x dan y nya

```
from sage.all import *
from Crypto.Util.number import inverse
from pwn import *
# r = process(['python3', 'chall (1).py'])
# context.log_level = 'debug'
r = remote("103.152.242.197", 54259)
def crack_unknown_increment(states, modulus, multiplier):
    increment = (states[1] - states[0]*multiplier) % modulus
    return modulus, multiplier, increment

def crack_unknown_multiplier(states, modulus):
    multiplier = (states[2] - states[1]) * inverse_mod(states[1] - states[0],
modulus) % modulus
    return crack_unknown_increment(states, modulus, multiplier)

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)

state = []
for i in range(3):
    r.sendlineafter(b'$ ', b'0,0')
    r.recvuntil(b'Pipit ditemukan pada posisi koordinat')
    x, y = eval(r.recvline())
    x = list(map(int, bin(x)[2:].zfill(64)))
    y = list(map(int, bin(y)[2:].zfill(64)))
    for i in range(21, len(x)):
        x[i] ^= x[i-21]
        y[i] ^= y[i-21]
    for i in range(len(x)-13-1, -1, -1):
        x[i] ^= x[i+13]
        y[i] ^= y[i+13]
    x = int(''.join(map(str, x)), 2)
    y = int(''.join(map(str, y)), 2)
```

```
    print("pred", x, y)
    state.append(x)
    state.append(y)

m, a, c = crack_unknown_modulus(state)
print(m, a, c)
class generator():
    def __init__(self, a, c, m, seed):
        self.a=a
        self.c=c
        self.m=m
        self.seed=seed
        U=[]
        for i in range(64):
            U.append([])
            for j in range(64):
                if j==i+1:
                    U[i].append(1)
                else:
                    U[i].append(0)
        U=Matrix(GF(2),U)
        L=[]
        for i in range(64):
            L.append([])
            for j in range(64):
                if j==i-1:
                    L[i].append(1)
                else:
                    L[i].append(0)
        L=Matrix(GF(2),L)
        self.U=U
        self.L=L

    def next(self):
        res=self.seed
        res='{:064b}'.format(res)
        res=vector(GF(2),list(res))
        res=res+((self.U**13)*res)
        res=res+((self.L**21)*res)
```

```
        res=''.join([str(i) for i in res])
        res=int(res,2)
        self.seed=(self.a*self.seed+self.c)%self.m
        return res

g = generator(a, c, m, state[-1])
g.next()
x = g.next()
y = g.next()
print(x, y)
r.sendlineafter(b'$ ', f'{x},{y}')
r.interactive()
```

```
└─$ python3 solvepipit.py
[+] Opening connection to 103.152.242.197 on port 54259: Done
pred 9808956625654904053 6111560424002585659
pred 1474850778200538305 7842536623446590726
pred 1294489394728243698 5433061978601118997
11850790146134293469 7644862029112664697 3306315078464563298
6709712298055377447 11382445057174622360
/mnt/d/technical/ctf/technofair/final/solvepipit.py:85: BytesWarning: Text is not bytes; assuming ASCII, n
o guarantees. See https://docs.pwntools.com/#bytes
  r.sendlineafter(b'$ ', f'{x},{y}')
[*] Switching to interactive mode
/home/wrth/.local/lib/python3.11/site-packages/pwnlib/tubes/tube.py:860: DeprecationWarning: isSet() is de
precated, use is_set() instead
  while not go.isSet():
/home/wrth/.local/lib/python3.11/site-packages/pwnlib/tubes/tube.py:879: DeprecationWarning: isSet() is de
precated, use is_set() instead
  while not go.isSet():
Pipit berhasil ditangkap! ini untukmu! TechnoFairCTF{H1ngg@_d1_@ngk@5@_5@n@_D1_kej@uh@n_pertemu@n_@nt@r@_k
1t@_berdu@_Temp@t_y@ng_t@k_@d@_51@p@pun_H155@t5u_telep0-0rt0}
[*] Got EOF while reading in interactive
$
[*] Interrupted
[*] Closed connection to 103.152.242.197 port 54259
```

Flag:
TechnoFairCTF{H1ngg@_d1_@ngk@5@_5@n@_D1_kej@uh@n_pertemu@n_@nt@r@_k1t@_berdu@_Temp@t_y@ng_t@k_@d@_51@p@pun_H155@t5u_telep0-0rt0}

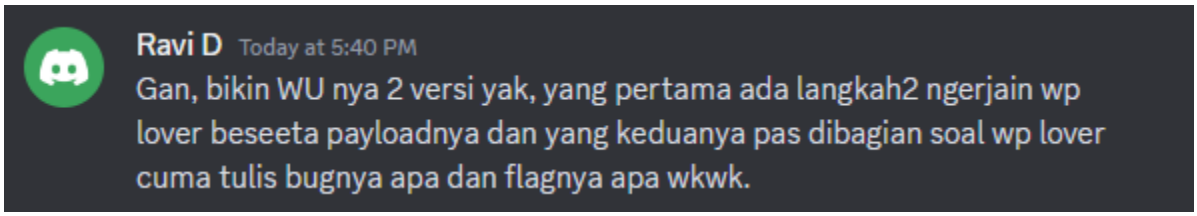
Note: gatau kenapa kadang eror gitu jadi di run aja sampe bisa

```
└─$ python3 solvepipit.py
[+] Opening connection to 103.152.242.197 on port 54259: Done
pred 7470974272545058971 7273568005858776444
pred 4546039623401057980 2197791616743071750
pred 5362139725457130038 9174483135006407939
25926479452794716522 18157177743943033942 21053831869147749906
Traceback (most recent call last):
  File "/mnt/d/technical/ctf/technofair/final/solvepipit.py", line 82, in <module>
    x = g.next()
        ^^^^^^^^
  File "/mnt/d/technical/ctf/technofair/final/solvepipit.py", line 73, in next
    res=res+((self.U**13)*res)
             ~~~~~~~~~~~^^~~~
  File "sage/structure/element.pyx", line 3826, in sage.structure.element.Matrix.__mul__ (build/cythonized
/sage/structure/element.c:24147)
    return coercion_model.bin_op(left, right, mul)
  File "sage/structure/coerce.pyx", line 1248, in sage.structure.coerce.CoercionModel.bin_op (build/cython
ized/sage/structure/coerce.c:11885)
    raise bin_op_exception(op, x, y)
TypeError: unsupported operand parent(s) for *: 'Full MatrixSpace of 64 by 64 dense matrices over Finite F
ield of size 2' and 'Vector space of dimension 65 over Finite Field of size 2'
[*] Closed connection to 103.152.242.197 port 54259
```

Web Exploitation

WordPress Lover

Sesuai arahan panitia karena emang ini exploit 0-day vulnerability. PoC bakalan di share kalau bug sudah di fix



Bug Chain: Unauthorized sql injection > php insecure deserialization > rce

Flag : TechnoFairCTF{sourc3_code_review_on_wordpress_plugin_is_fun_r1ght_you_c4n_g3t_CVE_from_th1s_activity}

Reverse Engineering

Kokekokko

Diberikan Dua buah file yaitu **main.lua** dan **FuraguEnc.lua**, kedua file ini sudah tercompile sehingga bisa langsung kita gunakan dengan menggunakan **lua** di Kali Linux. Ketika di Run, **main.lua** meminta sebuah input dan kemudian akan me-ouput karakter jepang. Sekarang kita akan coba untuk Decompile **.lua** ini dengan menggunakan tools **luadec** (viruscamp/luadec: Lua Decompiler for lua 5.1 , 5.2 and 5.3 (github.com)).

Awalnya by default saya menggunakan version 5.1, namun setelah saya menggunakan command **./luadec -dis ../../main.lua** saya mendapatkan error, yang bertuliskan luadec version not compatible to decompile 5.3, jadi saya install lagi yang luadec dengan version 5.3, dan didapatkan hasil awalnya.

```
┌──(kiinzu㉿Kiinzu)-[~/finaltekno/luadec/luadec]
└─$ ./luadec -dis ../../main.lua
cannot find blockend > 30 , pc = 29, f->sizecode = 30
; Disassembled using luadec 2.2 rev: 895d923 for Lua 5.3 from https://github.com/viruscamp/luadec
; Command line: -dis ../../main.lua

; Function:        0
; Defined at line: 0
; #Upvalues:       1
; #Parameters:     0
; Is_vararg:       1
; Max Stack Size:  5

    0 [-]: GETTABUP  R0 U0 K0      ; R0 := U0["require"]
    1 [-]: LOADK     R1 K1         ; R1 := "furaguEnc"
    2 [-]: CALL      R0 2 2        ; R0 := R0(R1)
    3 [-]: GETTABUP  R1 U0 K2      ; R1 := U0["print"]
    4 [-]: LOADK     R2 K3         ; R2 := "Do you know what giseigo is?"
    5 [-]: CALL      R1 2 1        ;  := R1(R2)
    6 [-]: GETTABUP  R1 U0 K4      ; R1 := U0["io"]
    7 [-]: GETTABLE  R1 R1 K5      ; R1 := R1["write"]
    8 [-]: LOADK     R2 K6         ; R2 := "Furagu : "
    9 [-]: CALL      R1 2 1        ;  := R1(R2)
   10 [-]: GETTABUP  R1 U0 K4      ; R1 := U0["io"]
   11 [-]: GETTABLE  R1 R1 K8      ; R1 := R1["read"]
   12 [-]: LOADK     R2 K9         ; R2 := "*l"
   13 [-]: CALL      R1 2 2        ; R1 := R1(R2)
   14 [-]: SETTABUP  U0 K7 R1      ; U0["flag"] := R1
   15 [-]: GETTABLE  R1 R0 K10     ; R1 := R0["chekkuFuragu"]
   16 [-]: GETTABUP  R2 U0 K7      ; R2 := U0["flag"]
   17 [-]: LOADK     R3 K11        ; R3 := "Onomatopoeia"
   18 [-]: LOADK     R4 K12        ; R4 := "1412"
   19 [-]: CALL      R1 4 2        ; R1 := R1(R2 to R4)
   20 [-]: TEST      R1 0          ; if R1 then goto 22 else goto 26
   21 [-]: JMP       R0 4          ; PC += 4 (goto 26)
   22 [-]: GETTABUP  R1 U0 K2      ; R1 := U0["print"]
   23 [-]: LOADK     R2 K13        ; R2 := "ピンポン！ 正解!"
   24 [-]: CALL      R1 2 1        ;  := R1(R2)
   25 [-]: JMP       R0 3          ; PC += 3 (goto 29)
   26 [-]: GETTABUP  R1 U0 K2      ; R1 := U0["print"]
   27 [-]: LOADK     R2 K14        ; R2 := "ブッブー... 違います\136... wwwwwwww"
   28 [-]: CALL      R1 2 1        ;  := R1(R2)
   29 [-]: RETURN    R0 1          ; return
```

Kalau di translate kurang lebih akan sepeerti ini bentuk psuodocodenya

```
require "furaguEnc" -- Load the "furaguEnc" module

print("Do you know what giseigo is?") -- Print the message

io.write("Furagu : ") -- Display "Furagu : "

flag = io.read("*l") -- Read user input and store it in the variable "flag"

chekkuFuraguResult = chekkuFuragu(flag, "Onomatopoeia", "1412") -- Call the "chekkuFuragu"
function with arguments "flag", "Onomatopoeia", and "1412"

if chekkuFuraguResult then -- If the result is true
    print("ピンポン! 正解!") -- Print "ピンポン! 正解!"
else
    print("ブッブー... 違います... WWWWWWWW") -- Print "ブッブー... 違います... WWWWWWWW"
end

return -- End the program
```

Karena kita sekarang tahu bahwa **main.lua** ini memanggil **furaguEnc** dengan menggunakan 3 parameter,sekarang kita akan coba mendissasemble **furaguEnc.lua** untuk lebih memahami workflow dari Enc nya.

```
  0 [-]: NEWTABLE  R3 35 0      ; R3 := {} (size = 35,0)
  1 [-]: LOADK     R4 K0        ; R4 := 86
  2 [-]: LOADK     R5 K1        ; R5 := 1264
  3 [-]: LOADK     R6 K2        ; R6 := 100
  4 [-]: LOADK     R7 K3        ; R7 := 48
  5 [-]: LOADK     R8 K4        ; R8 := 112
  6 [-]: LOADK     R9 K5        ; R9 := 1296
  7 [-]: LOADK     R10 K6       ; R10 := 146
  8 [-]: LOADK     R11 K7       ; R11 := 228
  9 [-]: LOADK     R12 K8       ; R12 := 150
 10 [-]: LOADK     R13 K9       ; R13 := 1968
 11 [-]: LOADK     R14 K10      ; R14 := 130
 12 [-]: LOADK     R15 K11      ; R15 := 348
 13 [-]: LOADK     R16 K12      ; R16 := 230
 14 [-]: LOADK     R17 K13      ; R17 := 128
 15 [-]: LOADK     R18 K14      ; R18 := 34
 16 [-]: LOADK     R19 K15      ; R19 := 136
 17 [-]: LOADK     R20 K16      ; R20 := 20
 18 [-]: LOADK     R21 K17      ; R21 := 560
 19 [-]: LOADK     R22 K18      ; R22 := 158
 20 [-]: LOADK     R23 K19      ; R23 := 196
 21 [-]: LOADK     R24 K20      ; R24 := 216
 22 [-]: LOADK     R25 K21      ; R25 := 720
 23 [-]: LOADK     R26 K22      ; R26 := 240
 24 [-]: LOADK     R27 K23      ; R27 := 132
 25 [-]: LOADK     R28 K24      ; R28 := 124
 26 [-]: LOADK     R29 K25      ; R29 := 400
 27 [-]: LOADK     R30 K26      ; R30 := 200
 28 [-]: LOADK     R31 K27      ; R31 := 332
 29 [-]: LOADK     R32 K16      ; R32 := 20
 30 [-]: LOADK     R33 K28      ; R33 := 1104
 31 [-]: LOADK     R34 K29      ; R34 := 152
 32 [-]: LOADK     R35 K30      ; R35 := 176
 33 [-]: LOADK     R36 K31      ; R36 := 92
 34 [-]: LOADK     R37 K32      ; R37 := 736
 35 [-]: LOADK     R38 K33      ; R38 := 98
 36 [-]: LOADK     R39 K34      ; R39 := 0
 37 [-]: LOADK     R40 K22      ; R40 := 240
 38 [-]: LOADK     R41 K35      ; R41 := 1472
 39 [-]: LOADK     R42 K36      ; R42 := 166
 40 [-]: LOADK     R43 K37      ; R43 := 408
 41 [-]: LOADK     R44 K38      ; R44 := 142
 42 [-]: LOADK     R45 K32      ; R45 := 736
```

```
   43 [-]: LOADK     R46 K39      ; R46 := 56
   44 [-]: LOADK     R47 K40      ; R47 := 60
   45 [-]: LOADK     R48 K41      ; R48 := 254
   46 [-]: LOADK     R49 K42      ; R49 := 368
   47 [-]: LOADK     R50 K43      ; R50 := 118
   48 [-]: LOADK     R51 K44      ; R51 := 188
   49 [-]: LOADK     R52 K45      ; R52 := 194
   50 [-]: LOADK     R53 K4       ; R53 := 112
   51 [-]: SETLIST   R3 50 1      ; R3[0] to R3[49] := R4 to R53 ;
R(a)[(c-1)*FPF+i] := R(a+i), 1 <= i <= b, a=3, b=50, c=1, FPF=50
   52 [-]: LOADK     R4 K16       ; R4 := 20
   53 [-]: LOADK     R5 K46       ; R5 := 352
   54 [-]: LOADK     R6 K47       ; R6 := 250
   55 [-]: LOADK     R7 K48       ; R7 := 1984
   56 [-]: LOADK     R8 K49       ; R8 := 14
   57 [-]: LOADK     R9 K50       ; R9 := 476
   58 [-]: LOADK     R10 K51      ; R10 := 114
   59 [-]: LOADK     R11 K52      ; R11 := 1648
   60 [-]: LOADK     R12 K53      ; R12 := 220
   61 [-]: LOADK     R13 K54      ; R13 := 384
   62 [-]: LOADK     R14 K55      ; R14 := 238
   63 [-]: LOADK     R15 K56      ; R15 := 832
   64 [-]: LOADK     R16 K57      ; R16 := 4
   65 [-]: LOADK     R17 K58      ; R17 := 204
   66 [-]: LOADK     R18 K59      ; R18 := 138
   67 [-]: LOADK     R19 K60      ; R19 := 1408
   68 [-]: LOADK     R20 K15      ; R20 := 136
   69 [-]: LOADK     R21 K61      ; R21 := 412
   70 [-]: LOADK     R22 K44      ; R22 := 188
   71 [-]: LOADK     R23 K62      ; R23 := 672
   72 [-]: LOADK     R24 K63      ; R24 := 18
   73 [-]: LOADK     R25 K54      ; R25 := 384
   74 [-]: LOADK     R26 K64      ; R26 := 154
   75 [-]: LOADK     R27 K65      ; R27 := 1376
   76 [-]: LOADK     R28 K66      ; R28 := 52
   77 [-]: LOADK     R29 K67      ; R29 := 496
   78 [-]: LOADK     R30 K68      ; R30 := 90
   79 [-]: LOADK     R31 K69      ; R31 := 800
   80 [-]: LOADK     R32 K70      ; R32 := 246
   81 [-]: LOADK     R33 K71      ; R33 := 316
   82 [-]: LOADK     R34 K72      ; R34 := 96
   83 [-]: LOADK     R35 K73      ; R35 := 1888
   84 [-]: LOADK     R36 K10      ; R36 := 130
```

```
   85 [-]: SETLIST   R3 33 2      ; R3[50] to R3[82] := R4 to R36 ;
R(a)[(c-1)*FPF+i] := R(a+i), 1 <= i <= b, a=3, b=33, c=2, FPF=50
   86 [-]: LEN       R4 R0        ; R4 := #R0
   87 [-]: LEN       R5 R3        ; R5 := #R3
   88 [-]: EQ        1 R4 R5      ; if R4 ~= R5 then goto 90 else goto 92
   89 [-]: JMP       R0 2         ; PC += 2 (goto 92)
   90 [-]: LOADBOOL  R4 0 0       ; R4 := false
   91 [-]: RETURN    R4 2         ; return R4
   92 [-]: NEWTABLE  R4 0 0       ; R4 := {} (size = 0,0)
   93 [-]: NEWTABLE  R5 0 0       ; R5 := {} (size = 0,0)
   94 [-]: NEWTABLE  R6 0 0       ; R6 := {} (size = 0,0)
   95 [-]: LOADK     R7 K74       ; R7 := 1
   96 [-]: LEN       R8 R0        ; R8 := #R0
   97 [-]: LOADK     R9 K74       ; R9 := 1
   98 [-]: FORPREP   R7 8         ; R7 -= R9; pc += 8 (goto 107)
   99 [-]: GETTABUP  R11 U0 K75   ; R11 := U0["string"]
  100 [-]: GETTABLE  R11 R11 K76  ; R11 := R11["byte"]
  101 [-]: SELF      R12 R0 K77   ; R13 := R0; R12 := R0["sub"]
  102 [-]: MOVE      R14 R10      ; R14 := R10
  103 [-]: ADD       R15 R10 K74  ; R15 := R10 + 1
  104 [-]: CALL      R12 4 0      ; R12 to top := R12(R13 to R15)
  105 [-]: CALL      R11 0 2      ; R11 := R11(R12 to top)
  106 [-]: SETTABLE  R4 R10 R11   ; R4[R10] := R11
  107 [-]: FORLOOP   R7 -9        ; R7 += R9; if R7 <= R8 then R10 := R7;
PC += -9 , goto 99 end
  108 [-]: LOADK     R7 K74       ; R7 := 1
  109 [-]: LEN       R8 R1        ; R8 := #R1
  110 [-]: LOADK     R9 K74       ; R9 := 1
  111 [-]: FORPREP   R7 8         ; R7 -= R9; pc += 8 (goto 120)
  112 [-]: GETTABUP  R11 U0 K75   ; R11 := U0["string"]
  113 [-]: GETTABLE  R11 R11 K76  ; R11 := R11["byte"]
  114 [-]: SELF      R12 R1 K77   ; R13 := R1; R12 := R1["sub"]
  115 [-]: MOVE      R14 R10      ; R14 := R10
  116 [-]: ADD       R15 R10 K74  ; R15 := R10 + 1
  117 [-]: CALL      R12 4 0      ; R12 to top := R12(R13 to R15)
  118 [-]: CALL      R11 0 2      ; R11 := R11(R12 to top)
  119 [-]: SETTABLE  R5 R10 R11   ; R5[R10] := R11
  120 [-]: FORLOOP   R7 -9        ; R7 += R9; if R7 <= R8 then R10 := R7;
PC += -9 , goto 112 end
  121 [-]: LOADK     R7 K74       ; R7 := 1
  122 [-]: LEN       R8 R2        ; R8 := #R2
  123 [-]: LOADK     R9 K74       ; R9 := 1
  124 [-]: FORPREP   R7 11        ; R7 -= R9; pc += 11 (goto 136)
```

```
  125 [-]: GETTABUP  R11 U0 K78   ; R11 := U0["tonumber"]
  126 [-]: SELF      R12 R2 K77   ; R13 := R2; R12 := R2["sub"]
  127 [-]: MOVE      R14 R10      ; R14 := R10
  128 [-]: MOVE      R15 R10      ; R15 := R10
  129 [-]: CALL      R12 4 0      ; R12 to top := R12(R13 to R15)
  130 [-]: CALL      R11 0 2      ; R11 := R11(R12 to top)
  131 [-]: GETTABUP  R12 U0 K79   ; R12 := U0["table"]
  132 [-]: GETTABLE  R12 R12 K80  ; R12 := R12["insert"]
  133 [-]: MOVE      R13 R6       ; R13 := R6
  134 [-]: MOVE      R14 R11      ; R14 := R11
  135 [-]: CALL      R12 3 1      ;  := R12(R13 to R14)
  136 [-]: FORLOOP   R7 -12       ; R7 += R9; if R7 <= R8 then R10 := R7;
PC += -12 , goto 125 end
  137 [-]: NEWTABLE  R7 0 0       ; R7 := {} (size = 0,0)
  138 [-]: LOADK     R8 K74       ; R8 := 1
  139 [-]: LEN       R9 R4        ; R9 := #R4
  140 [-]: LOADK     R10 K57      ; R10 := 4
  141 [-]: FORPREP   R8 19        ; R8 -= R10; pc += 19 (goto 161)
  142 [-]: NEWTABLE  R12 4 0      ; R12 := {} (size = 4,0)
  143 [-]: GETTABLE  R13 R4 R11   ; R13 := R4[R11]
  144 [-]: ADD       R14 R11 K74  ; R14 := R11 + 1
  145 [-]: GETTABLE  R14 R4 R14   ; R14 := R4[R14]
  146 [-]: ADD       R15 R11 K81  ; R15 := R11 + 2
  147 [-]: GETTABLE  R15 R4 R15   ; R15 := R4[R15]
  148 [-]: ADD       R16 R11 K82  ; R16 := R11 + 3
  149 [-]: GETTABLE  R16 R4 R16   ; R16 := R4[R16]
  150 [-]: SETLIST   R12 4 1      ; R12[0] to R12[3] := R13 to R16 ;
R(a)[(c-1)*FPF+i] := R(a+i), 1 <= i <= b, a=12, b=4, c=1, FPF=50
  151 [-]: LOADK     R13 K57      ; R13 := 4
  152 [-]: LOADK     R14 K74      ; R14 := 1
  153 [-]: LOADK     R15 K83      ; R15 := -1
  154 [-]: FORPREP   R13 5        ; R13 -= R15; pc += 5 (goto 160)
  155 [-]: GETTABUP  R17 U0 K79   ; R17 := U0["table"]
  156 [-]: GETTABLE  R17 R17 K80  ; R17 := R17["insert"]
  157 [-]: MOVE      R18 R7       ; R18 := R7
  158 [-]: GETTABLE  R19 R12 R16  ; R19 := R12[R16]
  159 [-]: CALL      R17 3 1      ;  := R17(R18 to R19)
  160 [-]: FORLOOP   R13 -6       ; R13 += R15; if R13 <= R14 then R16 :=
R13; PC += -6 , goto 155 end
  161 [-]: FORLOOP   R8 -20       ; R8 += R10; if R8 <= R9 then R11 := R8;
PC += -20 , goto 142 end
  162 [-]: LOADK     R8 K74       ; R8 := 1
  163 [-]: LEN       R9 R7        ; R9 := #R7
```

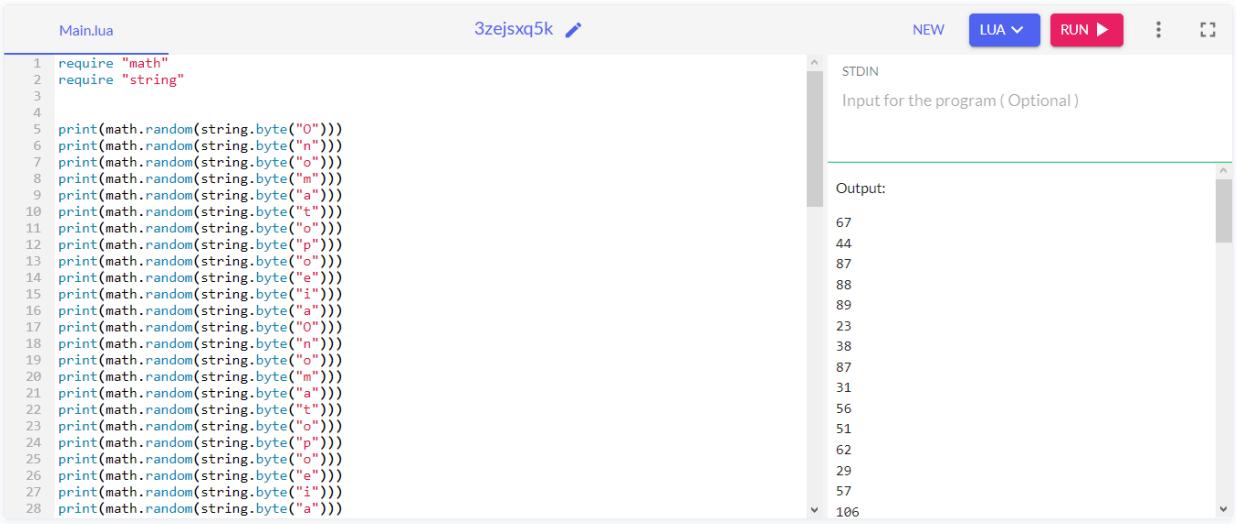
```
  164 [-]: LOADK     R10 K74      ; R10 := 1
  165 [-]: FORPREP   R8 25        ; R8 -= R10; pc += 25 (goto 191)
  166 [-]: GETTABLE  R12 R7 R11   ; R12 := R7[R11]
  167 [-]: GETTABUP  R13 U0 K84   ; R13 := U0["math"]
  168 [-]: GETTABLE  R13 R13 K85  ; R13 := R13["random"]
  169 [-]: SUB       R14 R11 K74  ; R14 := R11 - 1
  170 [-]: LEN       R15 R5       ; R15 := #R5
  171 [-]: MOD       R14 R14 R15  ; R14 := R14 % R15
  172 [-]: ADD       R14 K74 R14  ; R14 := 1 + R14
  173 [-]: GETTABLE  R14 R5 R14   ; R14 := R5[R14]
  174 [-]: CALL      R13 2 2      ; R13 := R13(R14)
  175 [-]: BXOR      R12 R12 R13  ; R12 := R12 ~ R13
  176 [-]: SETTABLE  R7 R11 R12   ; R7[R11] := R12
  177 [-]: GETTABLE  R12 R7 R11   ; R12 := R7[R11]
  178 [-]: SUB       R13 R11 K74  ; R13 := R11 - 1
  179 [-]: LEN       R14 R6       ; R14 := #R6
  180 [-]: MOD       R13 R13 R14  ; R13 := R13 % R14
  181 [-]: ADD       R13 K74 R13  ; R13 := 1 + R13
  182 [-]: GETTABLE  R13 R6 R13   ; R13 := R6[R13]
  183 [-]: SHL       R12 R12 R13  ; R12 := R12 << R13
  184 [-]: SETTABLE  R7 R11 R12   ; R7[R11] := R12
  185 [-]: GETTABLE  R12 R7 R11   ; R12 := R7[R11]
  186 [-]: GETTABLE  R13 R3 R11   ; R13 := R3[R11]
  187 [-]: EQ        1 R12 R13    ; if R12 ~= R13 then goto 189 else goto
191
  188 [-]: JMP       R0 2         ; PC += 2 (goto 191)
  189 [-]: LOADBOOL  R12 0 0      ; R12 := false
  190 [-]: RETURN    R12 2        ; return R12
  191 [-]: FORLOOP   R8 -26       ; R8 += R10; if R8 <= R9 then R11 := R8;
PC += -26 , goto 166 end
  192 [-]: LOADBOOL  R8 1 0       ; R8 := true
  193 [-]: RETURN    R8 2         ; return R8
  194 [-]: RETURN    R0 1         ; return
```

Dari Sini kita bisa langsung coba untuk membuat python yang mirip dengan hasil disassemble kita (ini dikuli 🥲)

```
import random

R3 = [86, 1264, 100, 48, 112, 1296, 146, 228, 150, 1968, 130, 348, 230, 128, 34,
136, 20, 560, 158, 196, 216, 720, 240, 132, 124, 400, 200, 332, 20, 1104, 152,
176, 92, 736, 98, 0, 240, 1472, 166, 408, 142, 736, 56, 60, 254, 368, 118, 188,
194, 112, 20, 352, 250, 1984, 14, 476, 114, 1648, 220, 384, 238, 832, 4, 204,
138, 1408, 136, 412, 188, 672, 18, 384, 154, 1376, 52, 496, 90, 800, 246, 316,
96, 1888, 130]

def convert_to_ascii(text):
    return [ord(c) for c in text]

# Default
R1 = "Onomatopoeia"
R2 = "1412"

#BRUTE
R0 = "Technofair"+"X"*73

if len(R0) != len(R3):
    print("len not equal")
    exit()

R4 = convert_to_ascii(R0)
R5 = convert_to_ascii(R1)
R6 = list(map(int, R2))

R7 = []
for R11 in range(0, len(R4), 4):
    R12 = R4[R11:R11+4]
    for R16 in R12[::-1]:
        R7.append(R16)

import random
for i in range(len(R7)):
    R12 = R7[i]
    R12 = R12 ^ random.randint(0, R5[i%len(R5)])
    R12 = R12 << R6[i%len(R6)]
```

```
    if R12 != R3[i]:
        print("Failed", i)
        exit()

print("Success")
```

Nah jadi input kita hanya di xor sama random dan di shift terus dibandingin sama angka di R3, jadi kita bisa balikin aja dari R3, tapi ada randomnya, jadi harus kita simulasiin dulu, karena ngga punya lua compiler local jadinya saya run online aja disini wkwkwk



Lua online compiler

Lalu hasil randomnya ini tinggal kita hardcode aja buat decrypt R3 nya

```
R3 = [86, 1264, 100, 48, 112, 1296, 146, 228, 150, 1968, 130, 348, 230, 128, 34,
136, 20, 560, 158, 196, 216, 720, 240, 132, 124, 400, 200, 332, 20, 1104, 152,
176, 92, 736, 98, 0, 240, 1472, 166, 408, 142, 736, 56, 60, 254, 368, 118, 188,
194, 112, 20, 352, 250, 1984, 14, 476, 114, 1648, 220, 384, 238, 832, 4, 204,
138, 1408, 136, 412, 188, 672, 18, 384, 154, 1376, 52, 496, 90, 800, 246, 316,
96, 1888, 130]

def convert_to_ascii(text):
    return [ord(c) for c in text]

# Default
R1 = "Onomatopoeia"
R2 = "1412"

R0 = "X"*83
```

```python
if len(R0) != len(R3):
    print("len not equal")
    exit()

R4 = convert_to_ascii(R0)
# R5 = convert_to_ascii(R1)
R5 = [67, 44, 87, 88, 89, 23, 38, 87, 31, 56, 51, 62, 29, 57, 106, 100, 62, 84,
16, 68, 2, 25, 15, 79, 13, 45, 15, 12, 97, 26, 57, 94, 69, 30, 67, 51, 39, 108,
33, 85, 52, 90, 45, 100, 32, 36, 85, 90, 6, 105, 59, 10, 19, 77, 99, 40, 8, 3,
49, 7, 19, 107, 101, 93, 26, 63, 42, 86, 57, 68, 56, 4, 35, 103, 104, 79, 28, 86,
72, 40, 77, 17, 47, 86]
R6 = list(map(int, R2))

R7 = []
for R11 in range(0, len(R4), 4):
    R12 = R4[R11:R11+4]
    for R16 in R12[::-1]:
        R7.append(R16)

import random
flag = []
for i in range(len(R7)):
    R12 = R7[i]
    R12 = R12 ^ random.randint(0, R5[i%len(R5)])
    R12 = R12 << R6[i%len(R6)]

    flag.append(chr(R3[i] >> R6[i%len(R6)] ^ R5[i%len(R5)]))
    # if R12 != R3[i]:
    #     print("Failed", i)
    #     exit()

flag = ["".join(flag[i:i+4][::-1]) for i in range(0, len(flag), 4)]
print("".join(flag))
print("Success")
```

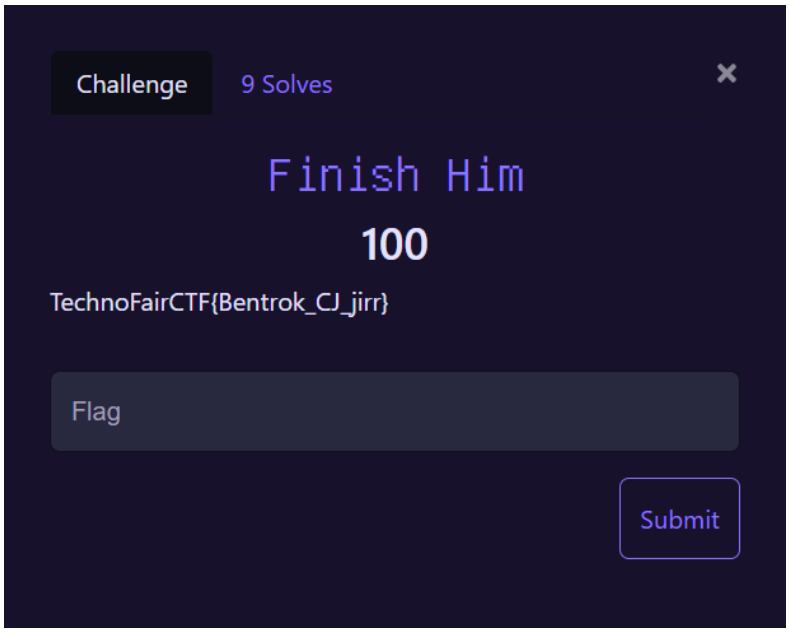
```
┌──(wrth㉿Wrth)-[/mnt/d/technical/ctf/technofair/final/crypto-attacks]
└─$ python3 solvekako.py
TechnoFairCTF{1nu_w4nw4n_k43ru_k3r0k3r0_k1tsun3_R1ng_d1ng_d1ng_d1ng_d1ng3r1ng3d1ng}
Success
┌──(wrth㉿Wrth)-[/mnt/d/technical/ctf/technofair/final/crypto-attacks]
```

Flag:
TechnoFairCTF{1nu_w4nw4n_k43ru_k3r0k3r0_k1tsun3_R1ng_d1ng_d1ng_d1ng_d1ng3r1ng3d1ng}

Misc

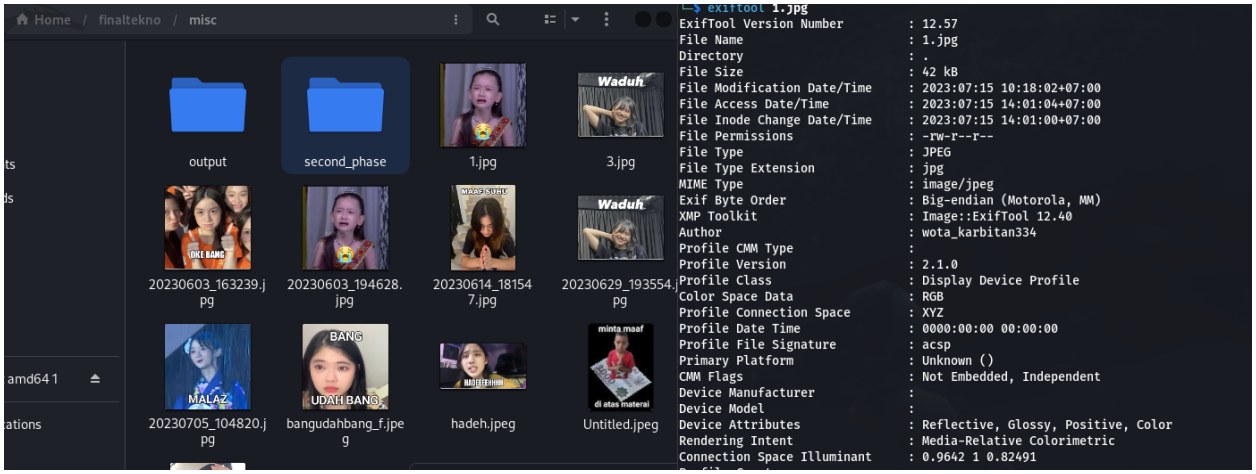
Finish Him

Free flag

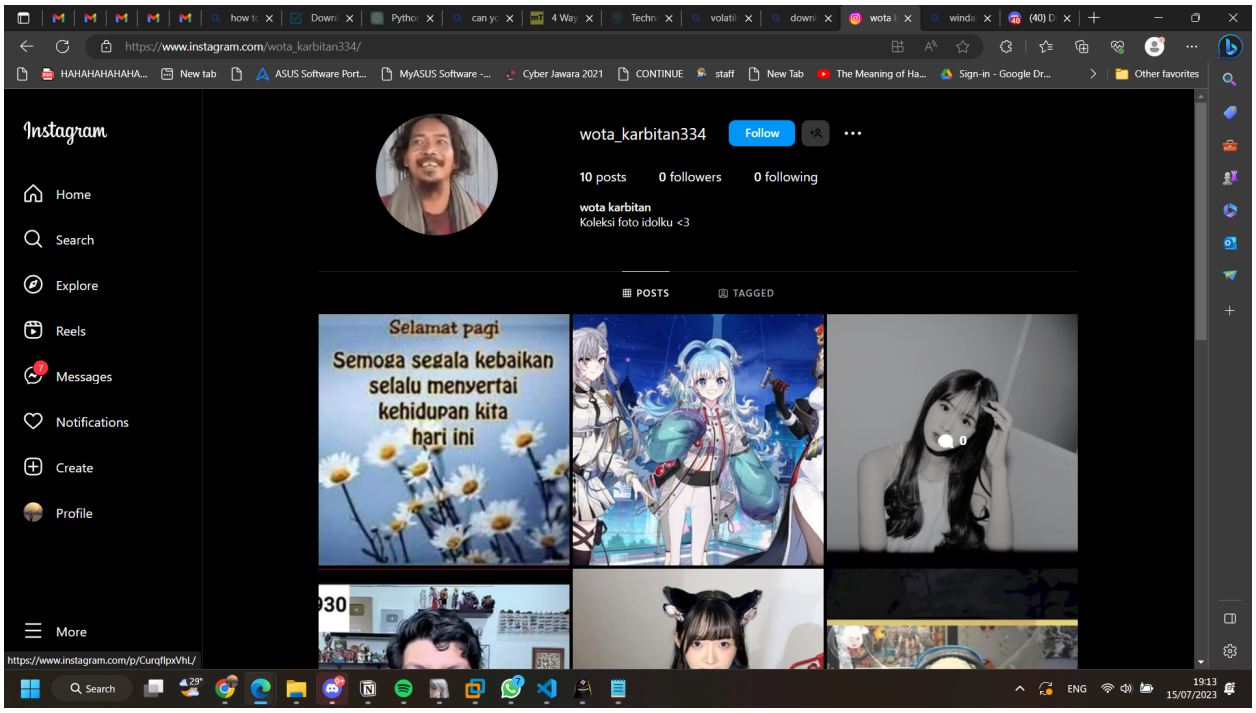


Kang Spam Stiker

Dari Deskripsi Soal, kita sudah tahu siapa si “Kang Spam Stiker” ini, tidak lain tidak bukan adalah si **AnYujin**. Nah masalahnya ya nie, stiker yang dia kirim itu ada 17 dan kita musti check 1 per satu sticker yg mana, setelah di **exiftool** 1 satu saya coba cari yang mana yang ada datanya beda, ada 1 gambar dengan link download https://cdn.discordapp.com/attachments/1129339733059846226/1129612817323003955/1.jpg yang memiliki “Author” di metadatanya (Sesuai dengan Hints 2), nah dari sini kita dapat username bernama **wota_karbitan334**



Ya intinya gambarnya yg anak nangis itulah yang si spammer kirim di Digital-Forensic. Sekarang kita tinggal cari ini username adanya dimana aja, entah itu di Instagram, Twitter, Telegram, atau sosmed apapun yang pakai username. Ternyata ketemunya di **Instagram**.

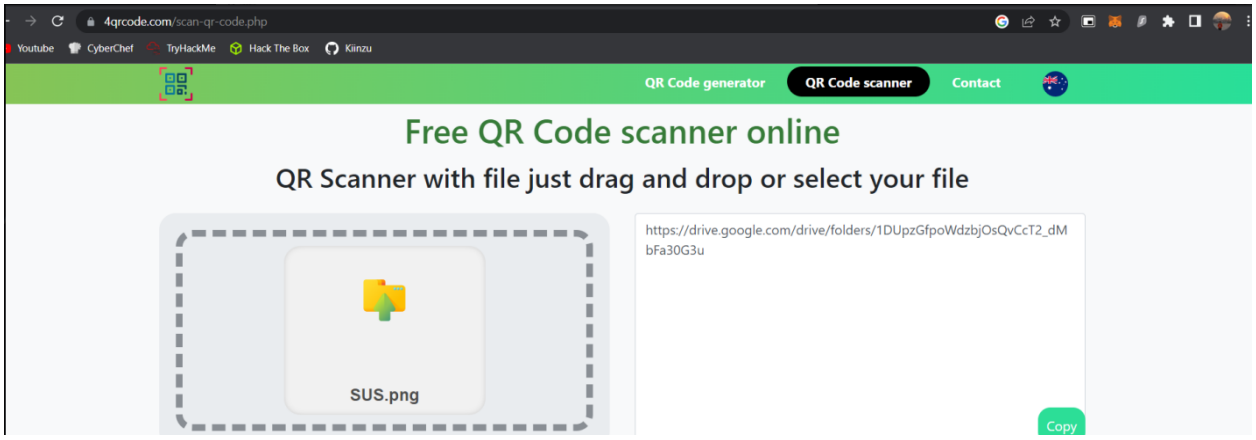


Nah tadi tuh awalnya saya bingung dan sempet tanya untuk next hintnya kayak gimana ke Probsetnya, dan keluarlah hints baru (karena mungkin saya tersesat & memang tersesat faktanya)

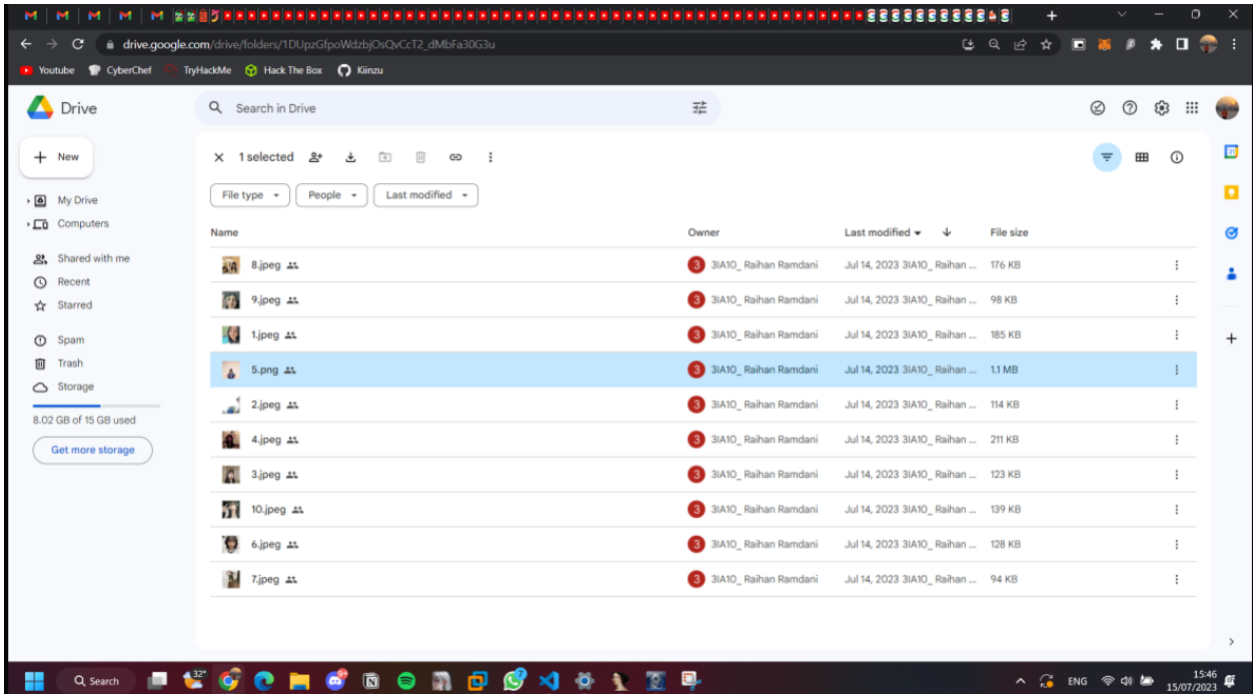
“waduh kamu sudah **melihat lihat** koleksiku 😁!?!?!”

Tambah Bingung jelas, lalu saya coba lihat-lihat lagi di Koleksi si wota ini, dan ada gambar yang menurut saya “kurang pas”, yaitu si foto bang Windah

Kayak, siapa sih yang pajang QR Code di dinding pake bingkai kayu, lalu sudah deh saya coba snippet itu si QR lalu saya masukan ke website ini untuk dibaca



Bener aja rupanya ada gdrive, yasudah kita langsung buka aja gdrivenya dan flagnya ada di salah satu foto yang ada disana.



Flag = TechnoFairCTF{halo_dek}