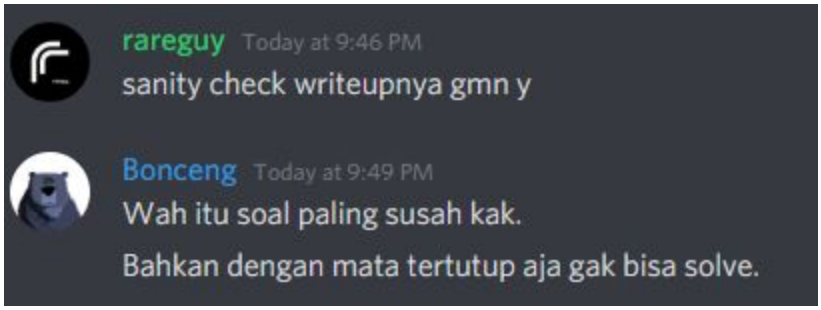
Writeup Penyisihan COMPFEST12

New Decade

Daftar Isi

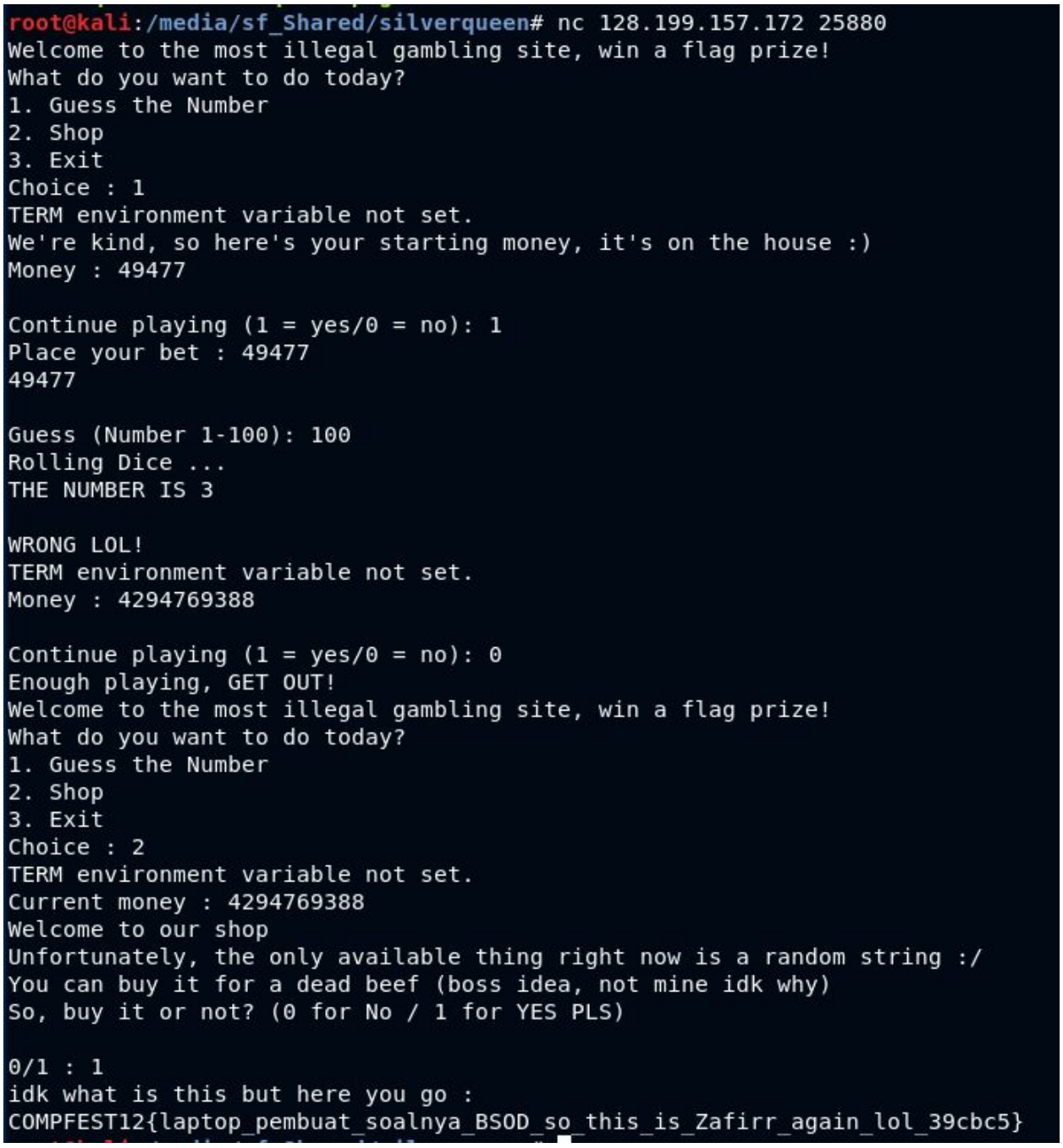
Sanity Check



Flag: COMPFEST12{im_not_insane}

Gambling Problem 2

Soal aneh.



Flag: COMPFEST12{laptop_pembuat_soalnya_BSOD_so_this_is_Zafirr_again_lol_39cbc5}

Mutual Friend

Diberikan 4000 buah bilangan prima, dan sebuah service yang mengambil 2 buah bilangan prima random dan mengenkripsi flag dengan RSA. Ambil 4000 tuple (n,e,c), lalu coba setiap pasang n. Jika ada yang memiliki gcd > 1, maka itu adalah p. Pasangan ini pasti ada, dapat dibuktikan dengan Pigeon Hole Principle. Setelah mendapat p, mencari q menjadi trivial. Masukkan ke RsaCtfTool dan dapatkan flagnya.

```
from pwn import *

nyari n e c
r = remote('128.199.157.172', 27268)

with open('dumpnec.txt', 'a') as out:
    for i in range(4000):
        r.sendline()

r.recvuntil('================================================================================')
        r.recvline()
        n = r.recvline(keepends=False).decode('utf-8').split(' ')[2]
        e = r.recvline(keepends=False).decode('utf-8').split(' ')[2]
        c = r.recvline(keepends=False).decode('utf-8').split(' ')[2]

r.recvuntil('================================================================================')
        r.recvline()
        print(n, e, c, file=out)

r.interactive()
```

Mengumpulkan tuple n,e,c

```
N = []
E = []
C = []
```

```python
import math

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    g, y, x = egcd(b%a,a)
    return (g, x - (b//a) * y, y)

def modinv(a, m):
    g, x, y = egcd(a, m)
    if g != 1:
        raise Exception('No modular inverse')
    return x%m

with open('dumpnec.txt', 'r') as inp:
    for i in range(4000):
        s = inp.readline().split(' ')
        N.append(int(s[0]))
        E.append(int(s[1]))
        C.append(int(s[2]))

    print('Finished Reading.')

for i in range(4000):
    for j in range(i+1, 4000):
        if (math.gcd(N[i], N[j]) != 1):
            p = math.gcd(N[i], N[j])
            print(p)
            print(N[i]//p)  # q
            print(E[i])
            print(C[i])
            exit()
```

Mencari p dan q yang tepat

Flag: COMPFEST12{Euclid_W0uID_b_Pr0Ud_Ov_4l1_7h3sE_MetH_eXpeRt5_a39e7a}

Lost My Source

Fungsi enkripsinya adalah:

```
for ( i = 31; i >= 0; --i )
  v4[31 - i] = v5[i] ^ i ^ v5[63 - i];
```

dengan v4 adalah encrypted dan v5 adalah flag.

Kita tahu bahwa bagian depan flag pasti COMPFEST12{. Coba dixorkan saja, lalu ternyata pola key nya adalah fedcbazyxwv…, teruskan polanya sampai membentuk key asli: fedcbazyxwvutsrqponmlkjihgfedcba. Dengan key tersebut, kita bisa menentukan flag asli.

```
# with open('encrypted.txt', 'rb') as inp:
#     a = inp.read()
#     print(a.hex())

a = '031a1b1c1d1e1f2f4b0e0b4807234b51210f4f102a074c412a2f2120302b2b25'
a = [int(a[i:i+2],16) for i in range(0,len(a),2)]  # convert ke int array
# [3, 26, 27, 28, 29, 30, 31, 47, 75, 14, 11, 72, 7, 35, 75, 81, 33, 15, 79, 16,
42, 7, 76, 65, 42, 47, 33, 32, 48, 43, 43, 37]

# for i in range(31, -1, -1):
#     print((a[31-i]^i), end=", ")

b = [28, 4, 6, 0, 6, 4, 6, 55, 92, 24, 30, 92, 20, 49, 90, 65, 46, 1, 66, 28, 33,
13, 69, 73, 45, 41, 36, 36, 51, 41, 42, 37] # m[i] ^ m[63-i]
key = 'fedcbazyxwvutsrqponmlkjihgfedcba' # pake text COMPFEST12{, dapet polanya

for i in range(len(b)):
    print(chr(b[31-i]^ord(key[i])), end='')
```

Script akhir

Flag: COMPFEST12{Th1s_15_y0ur5_abcdef}

I Hope It is Easy

Fungsi f(n) adalah sebuah fungsi yang memiliki banyak redundant ifs. Intinya, f(n) hanya mengecek apakah n merupakan bilangan kuadrat atau tidak. Kita diberikan sebuah file yang berisi angka-angka. Angka tersebut awalnya adalah sebuah bilangan kuadrat, lalu dixor dengan suatu bilangan yang merupakan ascii dari suatu karakter. Jadi, solusinya adalah membruteforce karakter apa yang digunakan sebagai flag. Untuk pengecekan apakah suatu angka adalah sebuah bilangan kuadrat, kita dapat menggunakan fungsi python yang ada di https://stackoverflow.com/questions/2489435/check-if-a-number-is-a-perfect-square

```
a = [array panjang (dari encrypted.txt)]

import math

#
https://stackoverflow.com/questions/2489435/check-if-a-number-is-a-perfect-square
def is_square(apositiveint):
  x = apositiveint // 2
  seen = set([x])
  while x * x != apositiveint:
    x = (x + (apositiveint // x)) // 2
    if x in seen: return False
    seen.add(x)
  return True

for x in a:
    for j in range(256):
        if is_square(j^x):
            print(chr(j), end="")
```

Script akhir

Flag: COMPFEST12{ez_pz_lemonade_squeez_a42447}

Checkmate

BFS sederhana. Ada 2 pilihan, multisource bfs dari semua knight, atau jadikan target sebuah source. Kita menggunakan pilihan kedua, karena sepertinya lebih simpel. Langsung koding BFS dan tembak ke server. Masalah: inputnya ga ada new line, jadi gabisa pake readline :")

```
from pwn import *

r = remote('128.199.157.172', 27136)

dx = [2,1,-1,-2,-2,-1,1,2]
dy = [1,2,2,1,-1,-2,-2,-1]

for run in range(7):
    print(run)
    a = [[0 for x in range(1000)] for x in range(1000)]
    b = []
    # while 1:
    #     r.recvline()
    #     s = r.recv(1)
    #     if (s == b'Y'):
    #         break
    #     s = r.recvline(keepends=False, timeout=1)
    #     # print(s)
    #     b.append(s.decode('utf-8').split('|')[:-1])

    data = r.recvuntil(': ')
    data = data.decode('utf-8').split('\n')
    for i in range(1, len(data), 2):
        if (data[i][0] == 'Y'):
            break
        b.append(data[i].split('|')[:-1])
    # print(b)

    queue = []
    for i in range(len(b)):
```

```
        for j in range(len(b[0])):
            if (b[i][j] == 'X'):
                queue.append((i,j))
                break
    ans = 0
    print(len(b), len(b[0]))
    while (len(queue) > 0):
        cx, cy = queue.pop(0)
        for i in range(8):
            nx = cx + dx[i]
            ny = cy + dy[i]
            if (nx >= 0 and nx < len(b) and ny >= 0 and ny < len(b[0]) and
a[nx][ny] == 0):
                a[nx][ny] = a[cx][cy] + 1
                queue.append((nx, ny))
                if b[nx][ny] == 'K':
                    ans = a[nx][ny]
                    break
        if ans != 0:
            break
    r.sendline(str(ans))

r.interactive()
```

Script akhir

Akan tetapi, server sering menutup koneksi saat mengirim puzzle ke 5 atau 6. Kami menghabiskan sekitar setengah jam di soal ini hanya karena masalah tersebut. :(
Terima kasih kepada panitia @faishol27 yang cepat menanggapi masalah ini.

Flag: COMPFEST12{y0u_GoT_th3_L_R19ht}

Lost My Source 2

Soal ini “tidak sengaja” kami solve. Pertama, ekstrak fungsi main dari binary dengan menggunakan pyi-archive_viewer. Lalu, buka main.pyc dengan vscode. Eh ternyata ada flagnya disitu wkwk.

```
D:\Shared\lost-my-source-2
λ pyi-archive_viewer.exe main
 pos, length, uncompressed, iscompressed, type, name
[(0, 245, 312, 1, 'm', 'struct'),
 (245, 1108, 1818, 1, 'm', 'pyimod01_os_path'),
 (1353, 4344, 9340, 1, 'm', 'pyimod02_archive'),
 (5697, 7365, 18639, 1, 'm', 'pyimod03_importers'),
 (13062, 1849, 4157, 1, 's', 'pyiboot01_bootstrap'),
 (14911, 405, 570, 1, 's', 'main'),
 (15316, 8245, 22040, 1, 'b', '_bz2.cpython-36m-x86_64-linux-gnu.so'),
 (23561, 102465, 149808, 1, 'b', '_codecs_cn.cpython-36m-x86_64-linux-gnu.so'),
 (126026, 35789, 158032, 1, 'b', '_codecs_hk.cpython-36m-x86_64-linux-gnu.so'),
 (161815,
  9816,
  26928,
  1,
  'b',
  '_codecs_iso2022.cpython-36m-x86_64-linux-gnu.so'),
 (171631, 97279, 272688, 1, 'b', '_codecs_jp.cpython-36m-x86_64-linux-gnu.so'),
 (268910, 78895, 137520, 1, 'b', '_codecs_kr.cpython-36m-x86_64-linux-gnu.so'),
 (347805, 63580, 112944, 1, 'b', '_codecs_tw.cpython-36m-x86_64-linux-gnu.so'),
 (411385, 10713, 29752, 1, 'b', '_hashlib.cpython-36m-x86_64-linux-gnu.so'),
 (422098, 13752, 33592, 1, 'b', '_lzma.cpython-36m-x86_64-linux-gnu.so'),
 (435850,
  24136,
  56600,
  1,
  'b',
  '_multibytecodec.cpython-36m-x86_64-linux-gnu.so'),
 (459986, 2042, 6280, 1, 'b', '_opcode.cpython-36m-x86_64-linux-gnu.so'),
 (462028, 45394, 120088, 1, 'b', '_ssl.cpython-36m-x86_64-linux-gnu.so'),
 (507422, 30120, 66728, 1, 'b', 'libbz2.so.1.0'),
 (537542, 1297187, 2917216, 1, 'b', 'libcrypto.so.1.1'),
 (1834729, 70921, 202880, 1, 'b', 'libexpat.so.1'),
 (1905650, 79560, 153984, 1, 'b', 'liblzma.so.5'),
 (1985210, 1823471, 4683728, 1, 'b', 'libpython3.6m.so.1.0'),
 (3808681, 126584, 294632, 1, 'b', 'libreadline.so.7'),
 (3935265, 230870, 577312, 1, 'b', 'libssl.so.1.1'),
 (4166135, 64264, 170784, 1, 'b', 'libtinfo.so.5'),
 (4230399, 60099, 116960, 1, 'b', 'libz.so.1'),
 (4290498, 11321, 31752, 1, 'b', 'readline.cpython-36m-x86_64-linux-gnu.so'),
 (4301819, 4690, 15368, 1, 'b', 'resource.cpython-36m-x86_64-linux-gnu.so'),
 (4306509, 8303, 24968, 1, 'b', 'termios.cpython-36m-x86_64-linux-gnu.so'),
 (4314812, 207043, 771132, 1, 'x', 'base_library.zip'),
 (4521855, 1140763, 1140763, 0, 'z', 'PYZ-00.pyz')]
? X
extract name? main
to filename? main.pyc
? Q
Traceback (most recent call last):
```

Flag: COMPFEST12{my_fri3nd_s4ys_s0rry_888144}

Kyu Are

Diberikan 9 video yang urutannya tidak penting. Ekstrak semua frame dari video tersebut, lalu gunakan pyzbar untuk mendecode semua gambar tersebut. Tinggal grep deh dan dapatkan flagnya.

```
import cv2
vidcap = cv2.VideoCapture('kyu.avi')
success,image = vidcap.read()
count = 8888
while success:
  cv2.imwrite("frame%d.jpg" % count, image)     # save frame as JPEG file
  success,image = vidcap.read()
#   print('Read a new frame: ', success)
  count += 1
```

Ekstrak frame

```
# https://ctftime.org/writeup/20324

from PIL import Image
from pyzbar.pyzbar import decode

with open('output.txt', 'w') as out:
    for i in range(10000):
        if (i % 100 == 0):
            print(i)
        qr_data = decode(Image.open('frame' + str(i) +
'.jpg'))[0].data.decode("utf-8")
        print(qr_data, file=out)
```

Decode frame

Peppery Spittoon

Soal yang paling satisfying pas dapet flagnya :)

Investigasi dulu prob.py nya, ternyata kita diberikan 6 buah pecahan sebagai “clue” dan 6 buah angka (P3ppeR) sebagai “hint”. Setelah mengotak-atik fungsi generate, diketahui bahwa fungsi tersebut melakukan eleminasi gauss-jordan, dimana clue adalah hasil dari sistem persamaan linear tersebut dan hint adalah angka paling kanan sebelum dilakukan eleminasi. Kita diminta untuk mengrestore matrix awal.

Untuk setiap baris, $ax_1 + bx_2 + cx_3 + dx_4 + ex_5 + fx_6 = p$, dimana x_1 hingga x_6 adalah clue, dan p adalah hint baris yang bersangkutan. Ide awalnya adalah membruteforce bilangan a sampai f. Tetapi, kompleksitasnya adalah $O(n^6)$ dan untuk n = 100, membutuhkan waktu yang cukup lama.

Kita dapat melakukan optimisasi menjadi $n^3 \log n$ dengan memisahkan 3 pecahan positif dan 3 pecahan negatif. Jika jumlahnya bukan 3, request pecahan baru. Without loss of generality, misalkan x1, x2 dan x3 adalah positif dan sisanya adalah negatif. Kita sekarang bisa membruteforce setiap pasangan a,b dan c dan menyimpan hasilnya di sebuah array. Lalu, sort array tersebut berdasarkan hasil $ax_1 + bx_2 + cx_3$.

Lalu, untuk setiap baris, brute force d,e,f untuk mencari tuple yang cocok dengan salah satu elemen array yang ada. Cocok disini artinya $ax_1 + bx_2 + cx_3 - p = dx_4 + ex_5 + fx_6$. Untuk mencari angka yang cocok, dapat digunakan binary search pada array positif.

Satu lagi, karena 112 muncul 2 kali, maka tuple yang didapatkan harus berbeda. Jika sama, mereka akan saling menghilangkan dan tidak ada solusi.

```
from fractions import Fraction

a =
[(-108357082667,52753543263),(-8415384706,5861504807),(76063199230,52753543263),(
80744220590,52753543263),(10486384108,17584514421),(-9668983667,52753543263)]
c = [80,51,112,101,82]

lcm = a[0][1]
```

```
import math
for i in range(6):
    lcm = lcm * a[i][1] // math.gcd(lcm, a[i][1])

# print(lcm)

for i in range(6):
    a[i] = (lcm // a[i][1] * a[i][0], lcm)
    # print(a[i][0])

neg = []
pos = []
for i in range(100):
    for j in range(100):
        for k in range(100):
            neg.append((-1 * (i * a[0][0] + j * a[1][0] + k * a[5][0]), i, j, k))

for i in range(100):
    for j in range(100):
        for k in range(100):
            pos.append((1 * (i * a[2][0] + j * a[3][0] + k * a[4][0]), i, j, k))

neg.sort()

for target in c:
    cnt = 0
    for x in pos:
        l = 0
        r = 999999
        found = -1
        # print(x, x[0]-112*lcm)
        while l <= r:
            m = (l+r)//2
            # print(m, neg[m])
            if x[0] - target * lcm == neg[m][0]:
                found = m
```

```
                break
            elif x[0] - target * lcm > neg[m][0]:
                l = m+1
            else:
                r = m-1
        if (found != -1):
            print(neg[found][1], neg[found][2], x[1], x[2], x[3], neg[found][3],
target, sep = '|', end = '|')
            cnt += 1
            if (cnt == 1 and target == 112):
                continue
            if (cnt == 2 and target == 112):
                break
            if (cnt == 1):
                break
```

Script akhir. Array a didapatkan dari pemanggilan netcat.

EZ Fibonacci

Kalau soal sebelumnya paling *satisfying*, soal ini paling *frustrating*. Frustratingnya bukan gara-gara soalnya susah, tapi gara-gara habis 2 jam disini. 1 jam pertama karena masalah socket, 1 jam kedua karena masalah output server yang korup (?). Padahal, langkah pengerjaannya cukup simpel.

Function1 adalah fungsi fibonacci biasa, tetapi dia melakukan rekursi, sehingga kompleksitasnya O(2^n). Ubah ke dp dan beres.
Function2 adalah fungsi mirip fibonacci. Relasi rekurensnya adalah 3*a(n-1)+a(n-2) untuk a = 1, b = 2, dimana a[0] adalah 2 dan a[1] adalah 5. Dp biasa dan beres.
Function3 mereverse sebuah string, lagi-lagi dengan rekursi. Ubah ke a[::-1] dan beres.
Function4 hanya mengambil string dari indeks ke 145 sampai selesai.

```
from pwn import *

r = remote('128.199.157.172', 21351)

msg = ''

with open('euy.txt', 'r') as euy, open('cluenext.txt', 'w+') as nextf:
    for i in range(4848):
        r.recvuntil('here:')
        text = euy.readline().strip()
        r.sendline(text)
        msg += r.recvline(keepends=False).decode('utf-8').strip()
        if (i % 100 == 0):
            print(i)
    print(msg, file=nextf)

r.interactive()
exit()
```

Script untuk menjawab pertanyaan

Setelah membuat script, coba tembak ke server untuk mendapatkan next clue. Tetapi, selalu gagal di tengah -_-. Habis 1 jam disini :(. Untungnya, panitia @RedGar membantu menyelesaikan masalah ini.

Setelah mendapatkan textnya, coba didecrypt dengan base64. Lho, kok cacat? Cuma depannya saja yang benar. Sekitar 60 karakter di belakang merupakan unprintable character. Tapi di depan ada tulisan mirip Fib0 di posisi 1,2,3,5. Asumsinya ambil karakter di posisi fibonacci. Lho, kok kayaknya kata terakhir salah? Tanya admin lagi dong, ini bermasalah atau tidak. Admin bilang tidak. Yasudah, habis 1 jam ngutak-ngatik itu flag “Fib0n4cc1_l5_T43_K3”. Disubmit jelas salah.

1 jam kemudian, kami mencoba untuk meminta text lagi dengan mensubmit script yang sama. Lho ternyata kali ini benar. Mungkin ada data yang korup saat berkomunikasi dengan server (atau mungkin di servernya yang korup). Ketemu deh flagnya “Fib0n4cc1_l5_T43_K3y”.

Iya, cuma beda 1 huruf tapi ga kepikiran karena udah frustrasi :”)

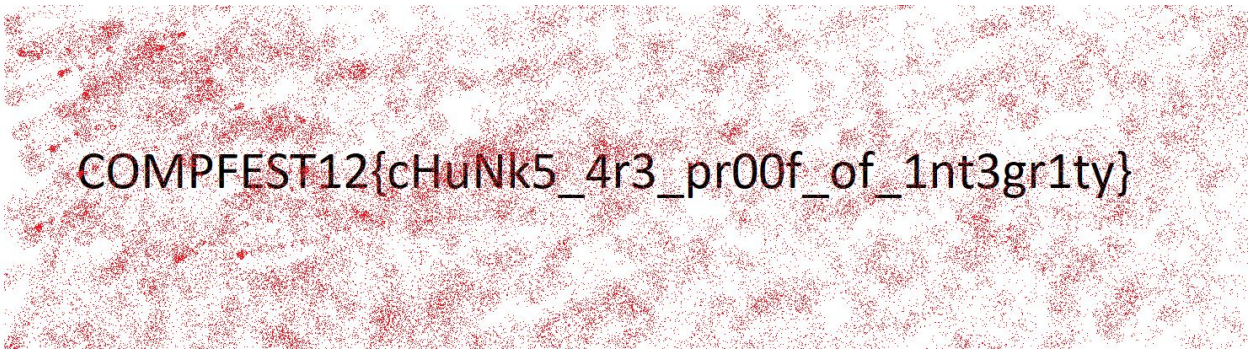
Flag: COMPFEST12{Fib0n4cc1_l5_T43_K3y}

Silverqueen

Simple png header fixing.

- PNG magic number salah, 894558450d0a1a0a menjadi 89504e470d0a1a0a.
- IHDR salah, 00484452 menjadi 49474452
- IDAT salah, ffffffff00446154 menjadi 0000ffa549444154

Ketemu deh gambarnya:



Flag: COMPFEST12{cHuNk5_4r3_pr00f_of_1nt3gr1ty}