**Writeup LycoReco**

**National Cybersecurity Week 2022 - Penyisihan**



**Anggota tim:**

**Muhammad Garebaldhie Er Rahman (average kobo enjoyer)**

**Frederik Imanuel Louis (azuketto)**

**Rachel Gabriela Chen (chaerla)**

# **Daftar Isi**

[**Daftar Isi**](#_jtobumv3ewy0) **2**

[**Welcoming Party**](#_dx0lkflrke9q) **3**

[Free Flag](#_q39crf75vs1g) 3

[Flag: NCW22{hology5{}\_ada\_yg\_lg\_h0lo\_gy\_juga\_ga\_hr\_1n1?}](#_2bgcgjq5b6dq) 3

[**Crypto**](#_64cqylr2dk0) **4**

[sabeb64](#_m36wkrp4fafi) 4

[Flag: NCW22{jusT\_a\_littl3\_d1ff3r3nt\_C0ncept\_0f\_base\_6nam\_emp4t\_x1x1x1\_1be1ec3c3ea96b983a75e17dd3d44e8b}](#_cxhxyvi5mid5) 9

[Cakemath](#_mkc9tct3snp3) 9

[Flag: NCW22{j4c0bi\_symb0ls\_101\_w!th\_c0mposit3\_numbers}](#_w5fkz12d4cdl) 11

[Turmoil](#_kswv8q9hi3st) 12

[Flag: NCW22{ReCoVer\_PriMes\_tO\_dIsCoVEr\_PriMes}](#_ceccsy3wv8bb) 18

[**Rev**](#_b9or4c5td3i8) **19**

[Count the Flag but Easier](#_di7ojb3oyuq9) 19

[Flag: NCW22{18612224}](#_fr2vz5x59zaf) 21

[**Foren**](#_421l36uhtqh) **22**

[Downloader](#_m6mj3ftoe9vq) 22

[Flag : NCW22{2filmes\_svchost.exe\_104.37.35.127\_Denmark}](#_mf5cuc1ioaxy) 25

[BEC Chitchat](#_isikc1u0chkh) 26

[Flag : NCW22{roger alex\_+120932132\_HK0PR06MB2867.apcprd06.prod.outlook.coma\_2002:a05:6a10:8a43:b0:2f4:89f4:8483}](#_1ijm6tx17bll) 28

[**Web**](#_2dq0cmr5d6no) **29**

[file&reading .INC](#_fmjlqtph1fv0) 29

[Flag : NCW22{f1L7eR\_15\_n0T\_3n0u9h\_1372846}](#_1yd27chfah80) 32

[**Misc**](#_ey8zwsr4q0gy) **33**

[Mr. Decryptor](#_5jdukguw1cie) 33

[NCW22{fuiyoohhh\_master\_of\_crypto\_right\_here!!!}](#_ie4rj5z1zqcw) 35

# Welcoming Party

| Free Flag 100  Welcome to NCW2022!  Yesterday was fun so we hope today you had more fun! Here's your free flag -> NCW22{hology5{}\_ada\_yg\_lg\_h0lo\_gy\_juga\_ga\_hr\_1n1?} |
| --- |

### 

Tinggal copas dapet flag horeeeee

### Flag: NCW22{hology5{}\_ada\_yg\_lg\_h0lo\_gy\_juga\_ga\_hr\_1n1?}

# 

# Crypto

| sabeb64 331  Jamet and Mamang are 2 scientists who want to study everything about cryptography. They intended to make a cryptographic tool, but there is something strange about the encoder they have made. The problem is, the encoder has already been used before the decoder was even designed. plus, Jamet accidentally spilled coffee on the source code, so now we can't see the encoding function. Can you solve it for them?  tips: analyze the output and compare with the og ;)  Chall: nc 103.167.136.75 9988 Backup: nc 103.167.136.123 9988  Author: kisanak#5303 |
| --- |

Berikut merupakan chall soal:

| import binascii  import random  charset = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'  charset2 = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'  def generateSecret():  secret = "".join(random.choices(charset2, k=30))  return secret.encode()  def sabebencode(text):  pass  *####################################*  *####################################*  *####################################*  *##ceritanya ini rusak kesiram kopi##*  *####################################*  *####################################*  *####################################*  if \_\_name\_\_ == '\_\_main\_\_':  print('Hi! I am Decoda, your personal base64 encoder 0\_0')  print('People said my calculations a little bit something... dunno. But, i\'ll do my best 0\_0')  print("My boss said bruteforce isn't needed, so you're only given 10 chances to use the encoder")  print('====================================================')  print('Please make a choice from the options below:')  count = 0  secret = generateSecret()  while True:  if count == 10:  print("You've reached your limit")  print("Thank you and have a good time! ^\_^")  break  print(f"Encode remaining: {10 - count}")  print('1. Print encoded secret')  print('2. Encode plaintext')  print('3. Submit secret and get flag')  inputan = input('>> ')  if inputan == '1':  print('Here\'s your encoded secret sir:')  res = sabebencode(secret)  print(res)  print('')  print('Anything else? 0\_0')  elif inputan =='2':  print('Please input your plaintext in hex 0\_0.')  text = input('>>')  try:  text = binascii.unhexlify(text)  res = sabebencode(text)  print('Beep boop... here\'s the result sir:')  print(res)  print('')  print('Anything else? 0\_0')  count+=1  except:  print('Sorry sir... weird input detected 0\_0\n')  elif inputan == '3':  print("Please submit the administrator's secret")  secretsubmit = input('>>').encode()  try:  if secretsubmit == secret:  flag = 'REDACTED'  print(f"Welcome back, admin. Here's your flag: {flag}\n")  else:  print("Wrong secret! You are not admin >:[\n");  except:  print('Sorry sir... weird input detected 0\_0\n')  else:  print('Sorry sir... weird input detected 0\_0\n') |
| --- |

Pada dasarnya, terdapat encoding oracle yang menerima plaintext dalam hex sebanyak maximum sepuluh kali, dan kita harus dapat menebak secret dari encoded secret yang diberikan (opsi 1). Setelah mencoba untuk melakukan encode beberapa kali, kami menyadari bahwa tiap dua karakter plaintext selalu menghasilkan tiga karakter ciphertext yang sama.

### 

Misalnya, pada contoh diatas, karakter ‘AA’ selalu diencode menjadi ‘IJB’. Dari fakta tersebut kami menyusun solver yang menanyakan hasil encode dari semua pasangan karakter yang mungkin (charset2 = ‘A-Z’) dalam satu string (‘AAABACAD…’).

from pwn import \* *# pip install pwntools*

import json

import codecs

import binascii

from Crypto.Util.number import long\_to\_bytes

from base64 import \*

from base58 import \*

import binascii

from pwn import \*

r = remote('103.167.136.75', 9988, level = 'debug')

charset = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'

charset2 = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

ls = [c for c in charset]

ls2 = [c for c in charset2]

ask = ""

cnt = 0

mpIdx={}

for a in charset2:

for b in charset2:

ask += str(a)

ask += str(b)

p = ""

p += str(a)

p += str(b)

mpIdx[cnt] = p

cnt+=1

r.recvuntil(b'>>')

r.sendline(b'1')

r.recvuntil(b'Here\'s your encoded secret sir:\n')

flag = r.recvline()[:-1]

flag = flag.decode()

r.recvuntil(b'>>')

r.sendline(b'2')

r.recvuntil(b'>>')

r.sendline(binascii.hexlify(ask.encode()))

r.recvuntil(b':\n')

ct = r.recvline()[:-1].decode()

mp = {}

for i in range(0, len(ct), 3):

cur = ct[i:i+3]

mp[cur] = mpIdx[i//3]

secret = ""

for i in range(0 ,len(flag), 3):

cur = flag[i:i+3]

secret += mp[cur]

print(f"secret:{secret}")

r.recvuntil(b'>>')

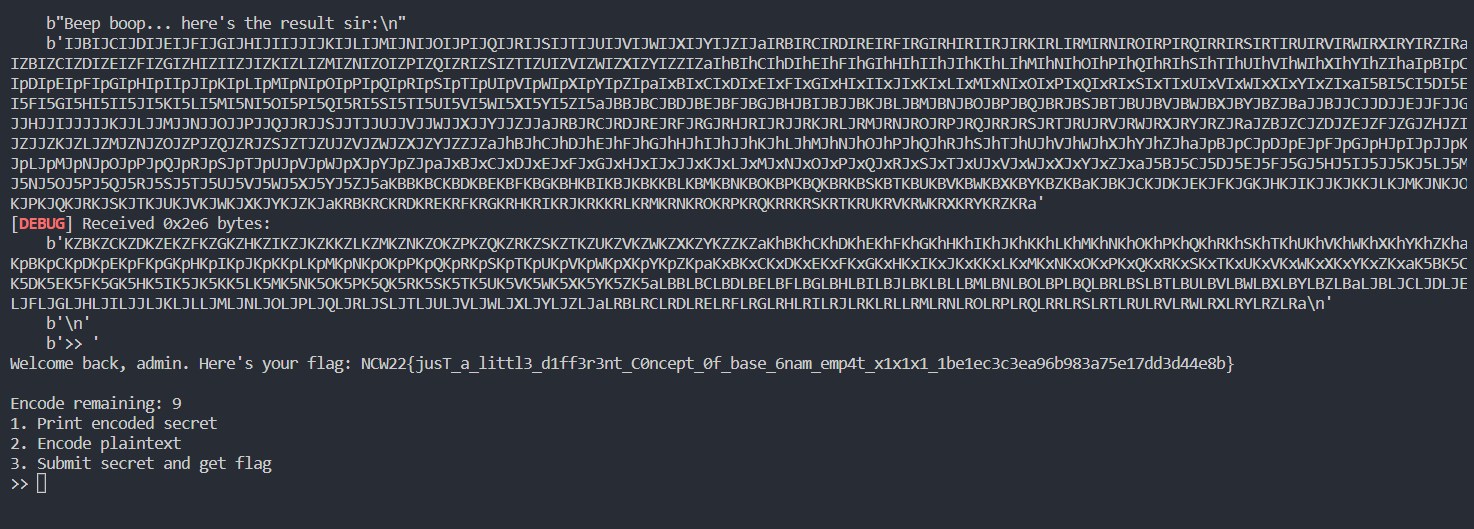
r.sendline(b'3')

r.recvuntil(b'>>')

r.sendline((secret.encode()))

r.interactive()

Setelah mendapatkan hasil encode, kami memetakan setiap 3 karakter hasil encode ke 2 karakter plaintext. Kemudian, menggunakan dictionary tersebut, kami mendecrypt encoding secret, dimana untuk tiap 3 karakter hasil encode, diperoleh 2 karakter plaintext secret. Perhatikan bahwa pada chall, panjang secret adalah 30 karakter, sehingga solusi ini cukup untuk memperoleh secret.



### Flag: NCW22{jusT\_a\_littl3\_d1ff3r3nt\_C0ncept\_0f\_base\_6nam\_emp4t\_x1x1x1\_1be1ec3c3ea96b983a75e17dd3d44e8b}

### 

# 

| Cakemath 451  Rehan, Jamet's acquintance, has just started learning cryptography and he chose this encryption from soneone's theory that he thought the implementation is pretty secure. What are the probability that someone had to break this encryption tho?  Rehan said this is a warmups ~  If you could break it, wrap the secret with NCW22{.\*}  Author: aseng#2055 |
| --- |

Berikut merupakan chall yang digunakan pada soal:

import random,math,os

from Crypto.Util.number import \*

flag = b'REDACTED'

c\_n1 = random.getrandbits(512)

c\_n2 = random.getrandbits(512)

assert c\_n1 != c\_n2

encrypted = []

bittersweet = [int(r) for r in "{:b}".format(bytes\_to\_long(flag))]

print(f"bittersweet: {bittersweet}")

for sugar in bittersweet:

while True:

gc\_c = random.randint(65536, c\_n1)

if math.gcd(gc\_c, c\_n1) == 1:

encrypted.append((pow(c\_n2,sugar)\*pow(gc\_c,2)) % c\_n1)

break

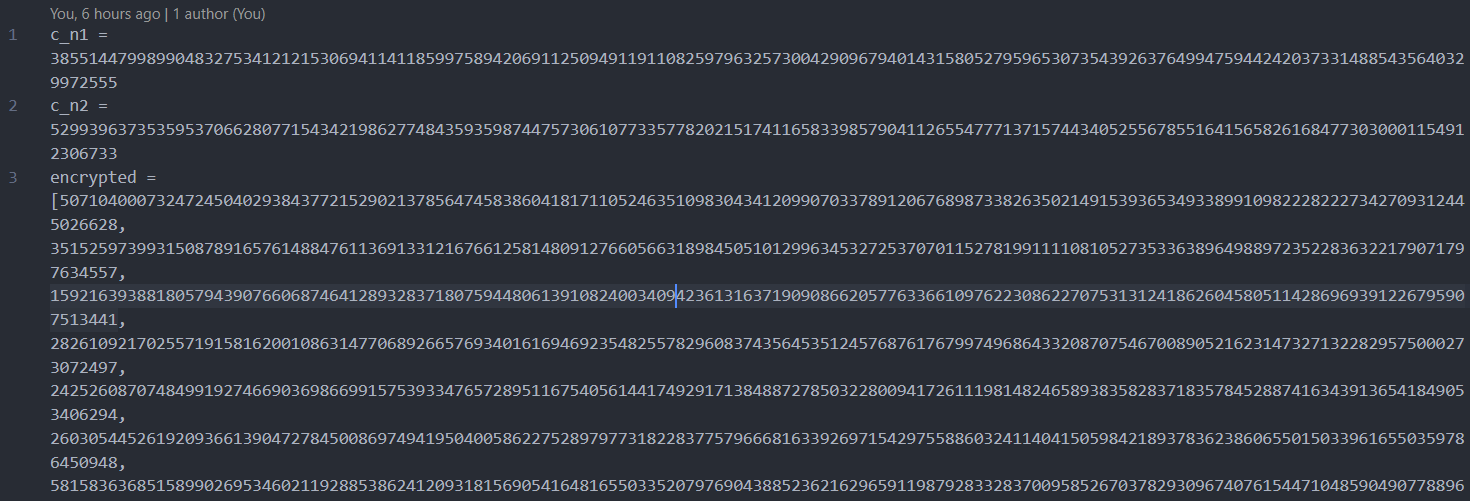
print("c\_n1 =",str(c\_n1))

print("c\_n2 =",str(c\_n2))

print("encrypted = ", end='')

print(encrypted)

Pada chall, plaintext dipecah menjadi tiap bitnya dan dimasukkan kedalam bittersweet. Kemudian, mulai dari MSBnya, tiap bit akan menentukan apakah c\_n2 dikalikan dengan pow(gc\_c,2) pada append encrypted.



Dari nilai c\_n1 dan c\_n2 yang diberikan, kita mendapatkan bahwa gcdnya adalah 29, dan perhatikan bahwa pada chall, gcd(gc\_c, c\_n1) haruslah satu. Maka, kita dapat mendapatkan tiap bit flag dengan mudah. Jika pada entri encrypted, nilai dapat dibagi oleh 29, maka bit flag nyala pada posisi tersebut. Hal tersebut dikarenakan nilai encrpypted tidak mungkin bisa habis dibagi 29 jika encrypted[i]=pow(gc\_c, 2, c\_n1) karena 29 membagi c\_n1 tetapi 29 tidak membagi gc\_c. Maka, kita dapat membuat solver berdasarkan fakta tersebut. (nilai encrypted redacted karena terlalu banyak)

import random,math,os

from Crypto.Util.number import \*

for c in encrypted:

if (c%29 != 0):

flag = flag\*2

else:

flag = flag\*2+1

print(long\_to\_bytes(flag))

### Flag: NCW22{j4c0bi\_symb0ls\_101\_w!th\_c0mposit3\_numbers}

\*note: sepertinya intended solution penulis lebih sulit dari ini (jacobi symbols apaan gan wkwk)

| Turmoil 491  Jamet and Mamang were assigned to build an RSA encryption machine to encrypt their boss's messages. But Jamet forgot to delete the code from the debugging process so some of it values are leaked. Is it really that dangerous?  Author: kisanak#5303 Chall: nc 103.167.136.75 9966 Backup: nc 103.167.136.123 9966 |
| --- |

### 

Berikut merupakan chall yang digunakan pada soal:

from Crypto.Util.number import \*

import random

import math

m = bytes\_to\_long(b"REDACTED")

def primevalue(value):

ind = 0

while not isPrime(value):

value += 1

ind += 1

print(f"added {ind}")

return value

def generatePrimes():

print("generating random numbers...")

p = random.getrandbits(512)

q = random.getrandbits(512)

print("calculating next prime...")

p = primevalue(p)

q = primevalue(q)

print("comparing values...")

print("=======| > |=======") if p > q else print("=======| < |=======")

return p, q

while True:

try:

print("==================================")

print("J&M's EncRSAyption MacRSAhine 3000")

print("==================================")

print("1. Spill Administrator's message")

print("2. Encrypt your own message")

pilihan = input("[?] ")

if pilihan == "1":

p, q = generatePrimes()

e = primevalue(random.randint(3, 100000))

n = p\*q

c = long\_to\_bytes(pow(m, e, n))

print("\nToo bad this part is highly confidential, only the ciphertext are allowed to be displayed.")

print(f"[>] {c.hex()}\n")

elif pilihan == "2":

p, q = generatePrimes()

n = p \* q

phi = (p-1)\*(q-1)

print("")

print("Input plaintext")

message = bytes\_to\_long(input("[?] ").encode())

while True:

print("Input your exponent")

e = int(input("[?] "))

if math.gcd(e, phi) != 1:

print('Error! exponent and totient have same factor(s)')

else:

d = inverse(e, phi)

print("Imma just show this to you, you already know your own message anyway")

print(f"d = {d}")

print("Use this exponent? (Y/N)")

choice = input("[?] ")

if choice == "Y" or choice == "y":

break

elif choice == "N" or choice == "n":

pass

c = pow(message, e, n)

print("Here's the result:")

print(f"[>] n = {n}\n[>] e = {e}\n[>] c = {c}\n")

except:

print('Error\n')

Chall pada dasarnya memberikan encryption oracle dan dapat mengenkripsi flag. Pada proses enkripsi di encryption oracle, pertama dilakukan random dua nilai awal p dan q (masing-masing 512 bytes).

Kemudian, p dan q akan digeser sampai keduanya prima, dan offset diberikan. Kemudian diberikan juga informasi apakah p > q atau p < q. Setelah itu, menggunakan nilai e yang kita berikan, nilai d diberikan oleh chall, dan dilakukan enkripsi plaintext yang kita masukkan.

Semua faktor tersebut mengarah pada attack random module python (random menggunakan modul, bukan os.urandom()). Untuk melakukan serangan, kita memerlukan nilai p dan q awal. Karena offset sudah diberikan, dan informasi mengenai p < q atau p > q juga diberikan, kita cukup memfaktorkan nilai n menggunakan nilai d dan e yang kita miliki, dan hal tersebut dapat dilakukan menggunakan algoritma yang dideskripsikan pada <https://www.di-mgt.com.au/rsa_factorize_n.html>.

Kemudian, kita akan menggunakan nilai p dan q awal yang kita peroleh untuk mereverse pRNG python, yang merupakan mersenne twister. Hal ini dapat kita lakukan menggunakan modul randcrack yang disediakan python (<https://github.com/tna0y/Python-random-module-cracker>). Pada modul tersebut, kita perlu memasukkan tepat 624 \* 32 bit yang diproduksi oleh modul random. Karena pada chall diproduksi 2 buah bilangan 512 bit, maka kita harus mendapatkan 624 \* 32 / 512 = 39 buah bilangan tersebut, atau 20 kali loop.

from pwn import \* *# pip install pwntools*

import json

import codecs

from Crypto.Util.number import \*

from math import gcd *#for gcd function (or easily implementable to avoid import)*

import random *#for random elements drawing in RecoverPrimeFactors*

from randcrack import RandCrack

def failFunction():

print("Prime factors not found")

def outputPrimes(a, n):

p = gcd(a, n)

q = int(n // p)

if p > q:

p, q = q, p

print("Found factors p and q")

print("p = {0}".format(str(p)))

print("q = {0}".format(str(q)))

return p,q

def RecoverPrimeFactors(n, e, d):

"""The following algorithm recovers the prime factor

s of a modulus, given the public and private

exponents.

Function call: RecoverPrimeFactors(n, e, d)

Input: n: modulus

e: public exponent

d: private exponent

Output: (p, q): prime factors of modulus"""

k = d \* e - 1

if k % 2 == 1:

failFunction()

return 0, 0

else:

t = 0

r = k

while(r % 2 == 0):

r = int(r // 2)

t += 1

for i in range(1, 101):

g = random.randint(0, n) *# random g in [0, n-1]*

y = pow(g, r, n)

if y == 1 or y == n - 1:

continue

else:

for j in range(1, t): *# j \in [1, t-1]*

x = pow(y, 2, n)

if x == 1:

p, q = outputPrimes(y - 1, n)

return p, q

elif x == n - 1:

continue

y = x

x = pow(y, 2, n)

if x == 1:

p, q = outputPrimes(y - 1, n)

return p, q

rc = RandCrack()

def feed(n):

mask = 0xffffffff

while n>0:

cur = mask & n

rc.submit(cur)

n = n >> 32

def primevalue(value):

ind = 0

while not isPrime(value):

value += 1

ind += 1

print(f"added {ind}")

return value

def generatePrimes():

print("generating random numbers...")

p = rc.predict\_getrandbits(512)

q = rc.predict\_getrandbits(512)

print("calculating next prime...")

p = primevalue(p)

q = primevalue(q)

print("comparing values...")

print("=======| > |=======") if p > q else print("=======| < |=======")

return p, q

ip = "103.167.136.75"

*#sock = int*

sock = 9966

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

feedCount = 0

while feedCount < 39:

r.recvuntil(b'[?]')

r.sendline(b'2')

r.recvuntil(b'added ')

a = int(r.recvline()[:-1].decode())

r.recvuntil(b'added ')

b = int(r.recvline()[:-1].decode())

r.recvuntil(b'comparing values...\n')

compare = r.recvline()[:-1]

*#print(f"a:{a}, b:{b}")*

r.recvuntil(b'[?]')

r.sendline(b'hello')

e = 65537

r.recvuntil(b'[?]')

r.sendline(b'65537')

r.recvuntil(b'd = ')

d = int(r.recvline()[:-1].decode())

r.recvuntil(b'[?]')

r.sendline(b'Y')

r.recvuntil(b'[>] n = ')

n = int(r.recvline()[:-1].decode())

p, q = RecoverPrimeFactors(n, e, d)

assert(p\*q == n)

if (p>q and compare == b'=======| < |======='):

p, q = q, p

elif (p<q and compare == b'=======| > |======='):

p, q= q, p

feed(p-a)

feedCount+=1

if feedCount==39:

break

feed(q-b)

feedCount+=1

rc.predict\_getrandbits(512)

p, q = generatePrimes()

e = primevalue(rc.predict\_randint(3, 100000))

tot = (p-1)\*(q-1)

d = pow(e, -1, tot)

r.recvuntil(b'[?]')

r.sendline(b'1')

r.recvuntil(b'[>] ')

ct = int(r.recvline()[:-1].decode(),16)

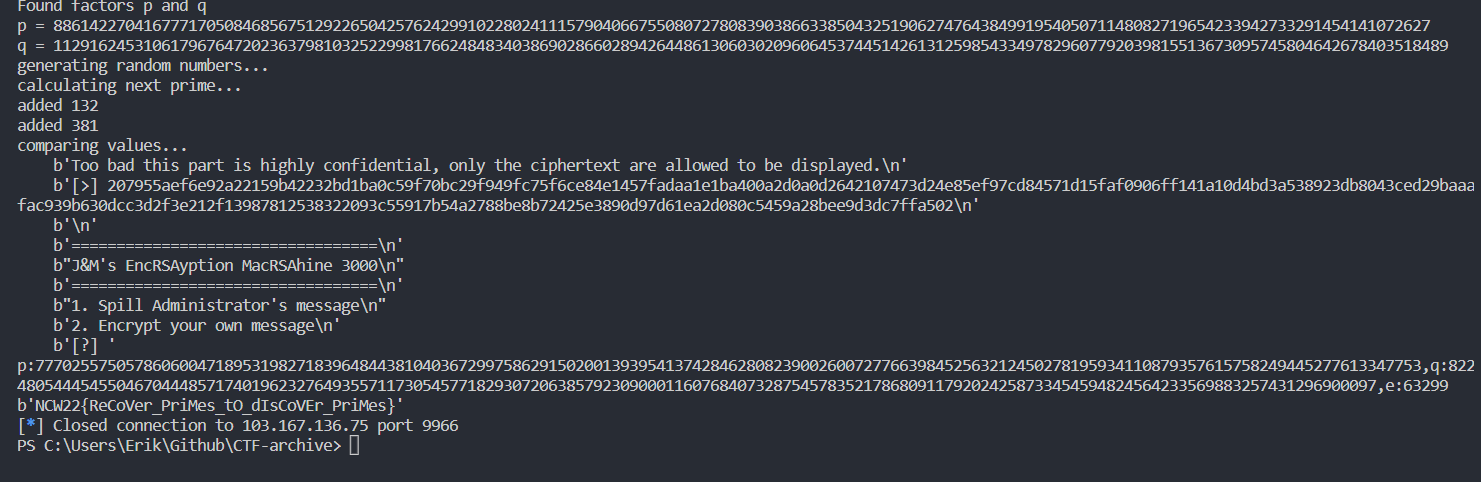
print(f"p:{p},q:{q},e:{e}")

flag = pow(ct, d, p\*q)

print(long\_to\_bytes(flag))

Kemudian, setelah berhasil mereverse random module, kita harus melakukan predict 512 bits satu kali untuk menyinkronkan state random pada server dengan state local karena server sebenarnya sudah melakukan getRandBits(512) 40 kali, bukan 39 kali.

Terakhir, karena kita sudah dapat memprediksi keluaran nilai random dari server, maka kita dapat mengetahui nilai p, q, dan e yang akan digunakan jika kita meminta encrypted flag dari server. Kita dapat dengan mudah menghitung nilai d dan memperoleh flag.



### Flag: NCW22{ReCoVer\_PriMes\_tO\_dIsCoVEr\_PriMes}

# **Rev**

| Count the Flag but Easier 331  Mikael is still struggling to find his way so the author makes an easier way. Help him to get the value! Example if the value is 100 then write in the answer NCW22{100}.  Author: Lawson Schwantz #3021 |
| --- |

### 

Diberikan sebuah txt file yang berisi assembly

| fungsi():  push rbp  mov rbp, rsp  mov DWORD PTR [rbp-4], 20  mov DWORD PTR [rbp-8], 10  mov DWORD PTR [rbp-12], 20  mov eax, DWORD PTR [rbp-4]  imul eax, DWORD PTR [rbp-8]  lea ecx, [rax+2]  mov eax, DWORD PTR [rbp-12]  mov edx, eax  sal eax, 2  sub edx, eax  lea eax, [rcx+rdx]  mov DWORD PTR [rbp-16], eax  sal DWORD PTR [rbp-16], 20  cmp DWORD PTR [rbp-16], 100000000  jg .L2  mov eax, DWORD PTR [rbp-16]  lea edx, [rax+3]  test eax, eax  cmovs eax, edx  sar eax, 2  mov DWORD PTR [rbp-16], eax  jmp .L3  .L2:  cmp DWORD PTR [rbp-16], 100000000  jle .L4  cmp DWORD PTR [rbp-16], 500000000  jg .L4  mov eax, DWORD PTR [rbp-16]  lea edx, [rax+7]  test eax, eax  cmovs eax, edx  sar eax, 3  mov DWORD PTR [rbp-16], eax  jmp .L3  .L4:  mov eax, DWORD PTR [rbp-16]  mov edx, eax  shr edx, 31  add eax, edx  sar eax  mov DWORD PTR [rbp-16], eax  .L3:  nop  pop rbp  ret |
| --- |

### 

Lalu kami menelusuri satu persatu dari assembly tersebut dimulai dari fungsi() hingga berakhir di .L3

| fungsi():  push rbp  mov rbp, rsp  mov DWORD PTR [rbp-4], 20 # rbp - 4 = 20  mov DWORD PTR [rbp-8], 10 # rbp - 8 = 10  mov DWORD PTR [rbp-12], 20 # rbp - 12 = 20  mov eax, DWORD PTR [rbp-4] # eax = 20  imul eax, DWORD PTR [rbp-8] # eax = 20 \* 10 = 200  lea ecx, [rax+2] -> # eax = 200 + 2 = 202  mov eax, DWORD PTR [rbp-12] # eax = 20  mov edx, eax -> # edx = 20  sal eax, 2 -> # eax = 80  sub edx, eax -> # edx = -60 & 0xffffffff (32bit) = 0xffffffc4  lea eax, [rcx+rdx] # eax = 202 - 60 = 142  mov DWORD PTR [rbp-16], eax # rbp-16 = 142  sal DWORD PTR [rbp-16], 20 # rbp - 16 = 148897792  cmp DWORD PTR [rbp-16], 100000000 # its greater so go to L2  jg .L2  mov eax, DWORD PTR [rbp-16]  lea edx, [rax+3]  test eax, eax  cmovs eax, edx  sar eax, 2  mov DWORD PTR [rbp-16], eax  jmp .L3  .L2:  cmp DWORD PTR [rbp-16], 100000000 # Still greater  jle .L4  cmp DWORD PTR [rbp-16], 500000000 # < 500000000 so continue  jg .L4  mov eax, DWORD PTR [rbp-16] # eax = 148897792  lea edx, [rax+7], # edx = 148897792 + 7  test eax, eax # eax = 0  cmovs eax, edx # eax = 0 so skip cmovs  sar eax, 3 # eax = 18612224  mov DWORD PTR [rbp-16], eax # rbp-16 = 18612224  jmp .L3 # GOTO .L3  .L4:  mov eax, DWORD PTR [rbp-16]  mov edx, eax  shr edx, 31  add eax, edx  sar eax  mov DWORD PTR [rbp-16], eax  .L3:  nop  pop rbp  Ret # 18612224 |
| --- |

Setelah didapat hasil yang di return maka tinggal di wrap dengan NCW22 lalu didapat flag

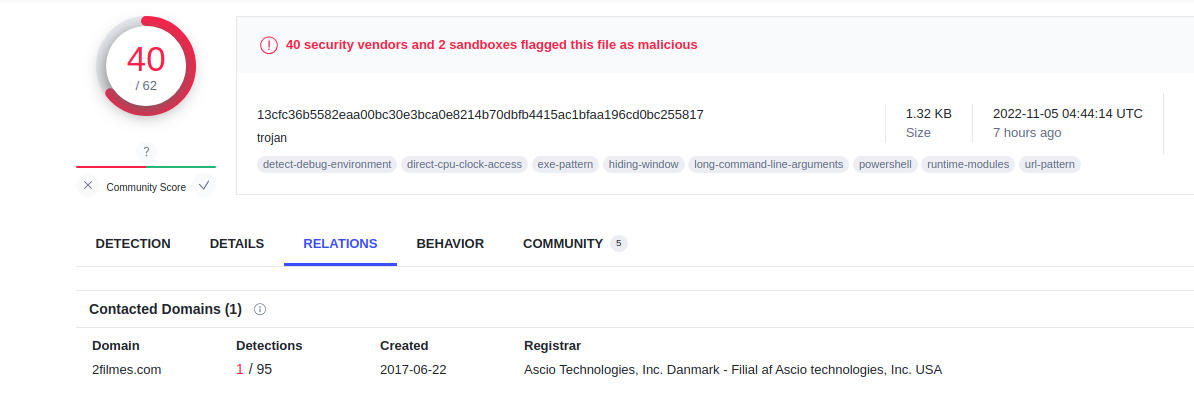
### Flag: NCW22{18612224}

# **Foren**

| Downloader 244  Hey folks, before you dive into the challenge, you have to know that this file contains real malware that is collected from malware's global database where all the malwares are quarantined there for further investigation by Forensic Investigator and for study purposes by people who want to sharpen their Forensic skill. (Reallife-like CTF :P)  So, in order to solve this challenge you can use many free tools and with your unique analysis skill to get the answers according to the given questions. This one is easy to solve and doesn't require any advanced analysis technique. Also DO NOT FORGET to delete this file after the competition is done, just for safety reason.  Again, i'm not responsible of any risk if you reject this warning.  You can access the questions here : https://tinyurl.com/wb9w957c  Flag is ALL 4 Questions concatenated with "\_" .  For example, NCW22{answer1\_answer2\_answer3\_answer4}  Here is the netcat service to validate your answer :  nc 103.167.136.75 1112  nc 103.167.136.123 1112 |
| --- |

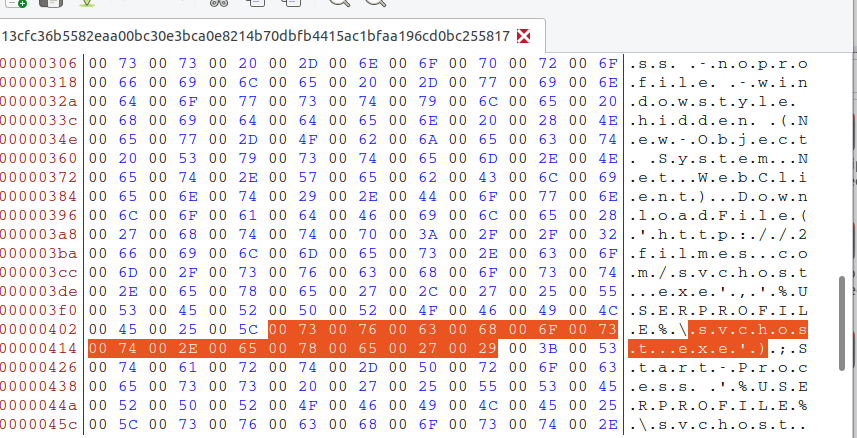
Diberikan sebuah file .zip. Dari deskripsi, diketahui bahwa file ini merupakan file trojan yang bisa dianalisis dengan “free tools”. Kami menemukan sebuah tool online untuk menganalisis file trojan yang decompress dalam zip tersebut, yaitu virustotal.com.

Setelah dianalisis, kami mendapatkan:



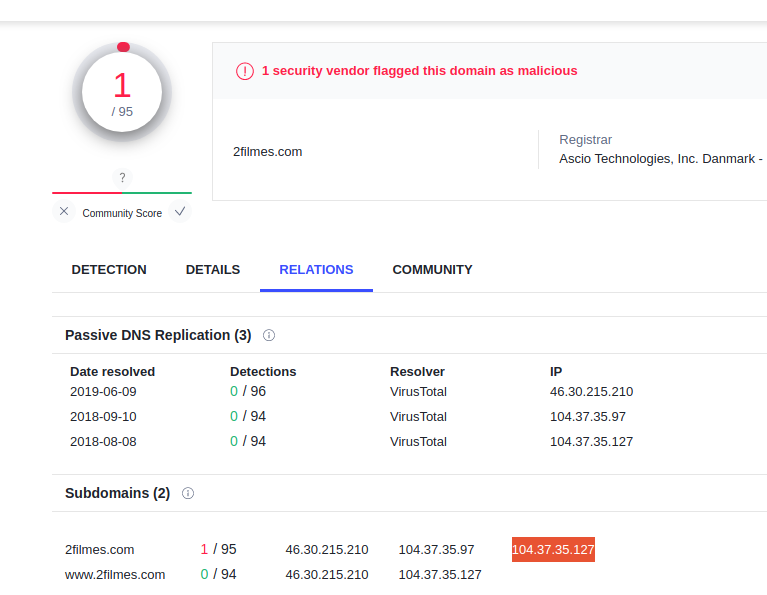
Nama domain (Answer1): 2filmes

Dengan menganalisa hex dump dari file trojan, kami mendapatkan:



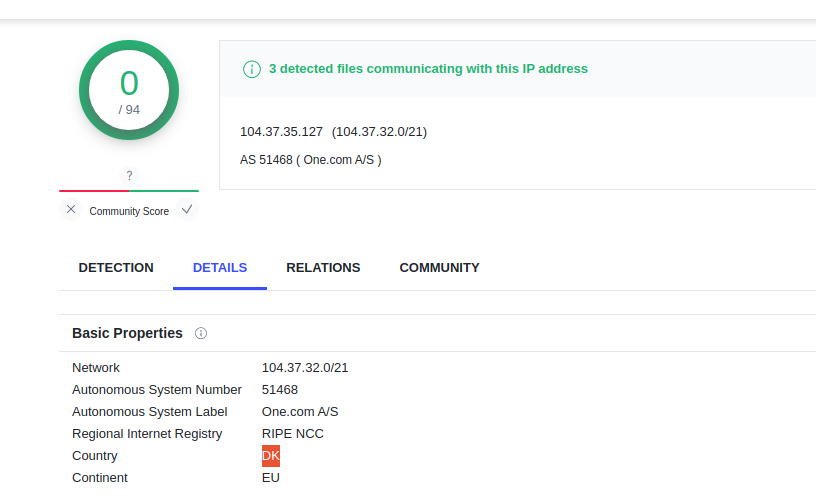
Nama file trojan (Answer2): svchost.exe

Melanjutkan analisa di virustotal.com, kami mengikuti link report domain 2filmes.com dan mendapatkan IP-nya:



IP Address (Answer3): 104.37.35.127

Lalu kami mengecek ip 104.37.35.127 dengan tool yang sama dan mendapatkan:



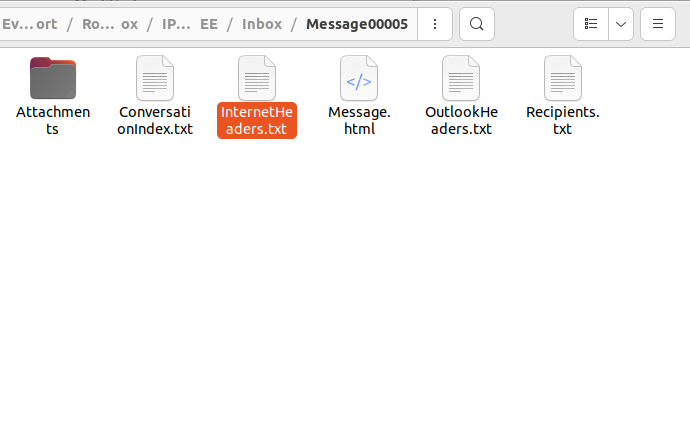
Dengan googling didapatkan, DK adalah kode untuk Denmark.

Negara (Answer4): Denmark

### Flag : NCW22{2filmes\_svchost.exe\_104.37.35.127\_Denmark}

| BEC Chitchat 500  "A few days ago, I went to a store wanting to buy groceries, but the store was closed. Then, in front of the store I saw a banner with an email referring to the owner of the store. I contacted the email several times and I received a reply message from the email along with a brochure. Long story short, I go back home and opened the brochure from my computer and my computer got hacked, and I only realized after a few days later."  As a Forensic Expert, you are given a document to analyze these evidences : |
| --- |

Diberikan sebuah file .zip yang berisi file .ost. Kami pun menggunakan pff-tools untuk menganalisis file tersebut. Setelah di-extract, kami menemukan folder yang berisi email-email yang diterima. Lalu, kami mengecek folder Message00005:

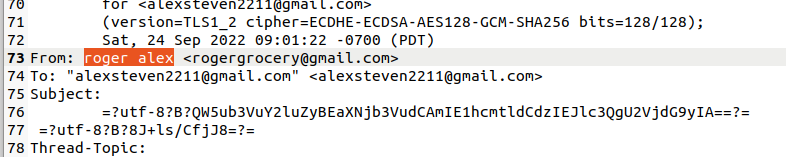


Dengan membuka dan menganalisa InternetHeaders.txt, kami menemukan jawaban-jawaban untuk pertanyaan.

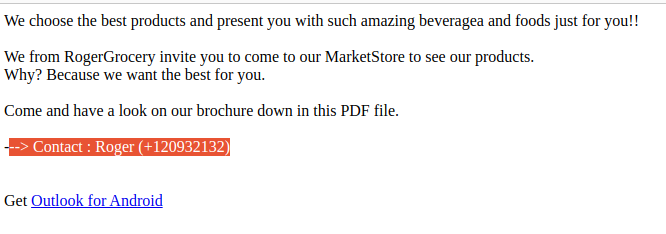
1. What's the name of suspected person(attacker) that send the malicious brochure?

(FULLNAME all lower case + if the name consists of two words like "Ismail Marzuki" then seperate those with whitespace character)

roger alex



2. What is the attacker's phone number? (dari file Message.html pada dir yang sama)



+120932132

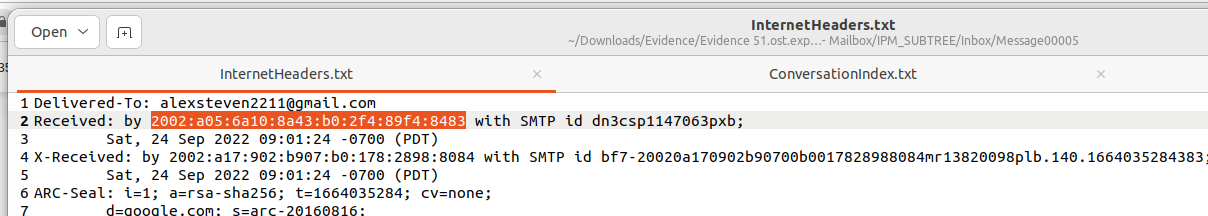
3. What is the Address(FQDN) that is close to the source email(sender)?

(E.g : VG7SCF8EV1D.prod.ncwctf.donat.gula.id)



HK0PR06MB2867.apcprd06.prod.outlook.coma

4.



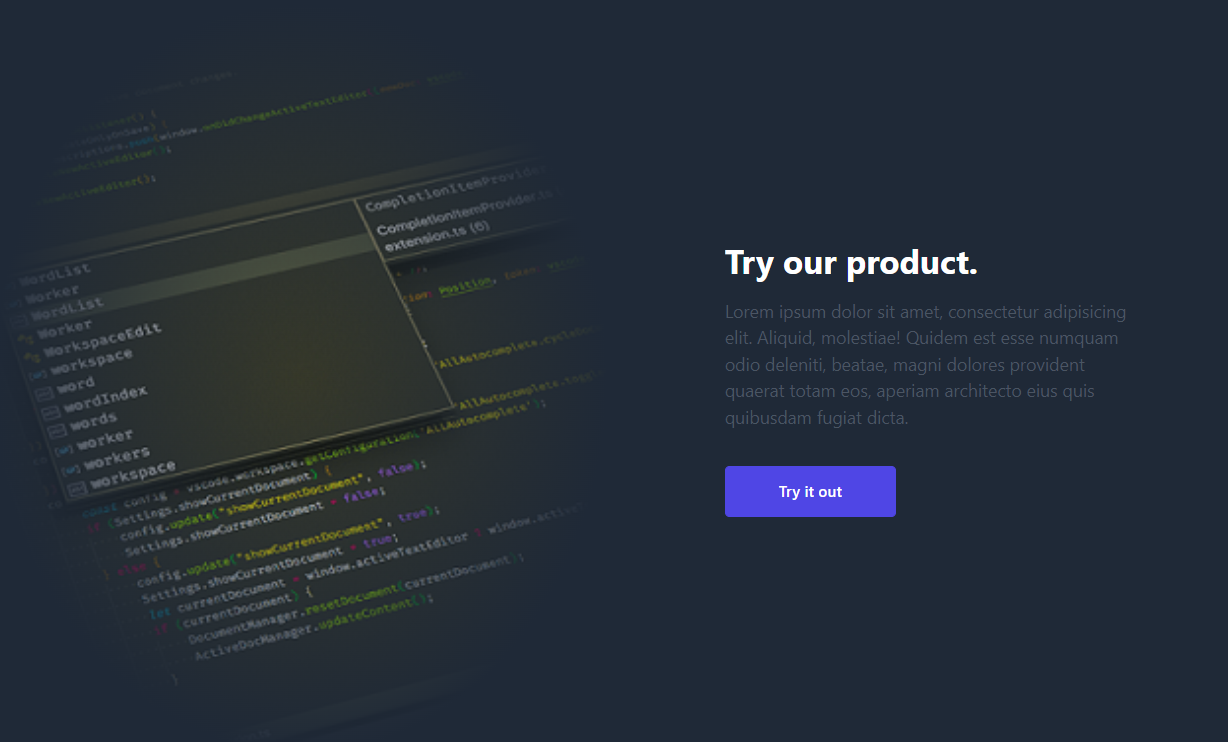
2002:a05:6a10:8a43:b0:2f4:89f4:8483

### Flag : NCW22{roger alex\_+120932132\_HK0PR06MB2867.apcprd06.prod.outlook.coma\_2002:a05:6a10:8a43:b0:2f4:89f4:8483}

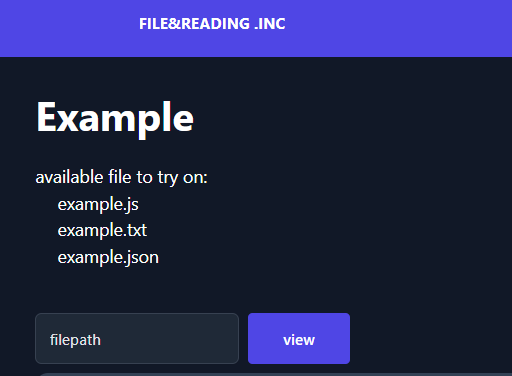
# Web

| file&reading .INC 100  New challenger has entered the arena, a start up company named file & reading incorporated has just made an announcement that they are making some sort of web based file reading tool for server maintainer, the possibility seems endless.  The flag is at /flag.txt  Author: mitm#0012  backup: http://103.167.136.123:54170/  http://103.167.136.75:54170/ |
| --- |

Di web yang diberikan, terdapat sebuah tombol Try It Out:

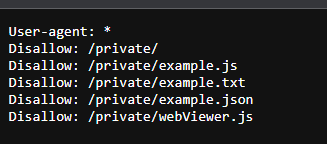


Setelah meng-klik tombol tersebut, kita menemukan page yang dapat digunakan untuk directory traversal:



Setelah mencoba payload /flag.txt, ../flag.txt, dsb., kita menyimpulkan bahwa ../ dan / difilter oleh website.

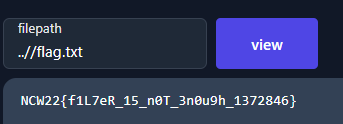
Kita kemudian melakukan GET request pada /robots.txt dan mendapatkan:



Kita kembali ke page /viewer, dan memasukkan webViewer.js pada kolom yang tersedia dan mendapatkan source code:

| const fs = require('fs');  const process = require('process')  function changeDir(){  try {  process.chdir("./private/")  } catch (err){}  }  function changeBack(){  try {  process.chdir("../")  } catch (err){}  }  function safetyCheck(filepath){  if(!filepath){return "example.txt"}  let safePath = filepath;  let hasSlash = false;  let hasDotDot = false;  if(safePath.startsWith("/")){  safePath = safePath.replace("/", "");  hasSlash = true  }    for(let x=0; x < 10; x++){  if(!safePath.includes("../")){break};  if(x==9){return "nicetry.txt"};  safePath=safePath.replaceAll("../","");  hasDotDot = true  }    if(!fs.existsSync(safePath) && (hasDotDot || hasSlash)){return "nicetry.txt"};  if(!fs.existsSync(safePath)){return "invalidFile.txt"};  return safePath;  }  module.exports = {  readfile:(filepath)=>{  try {  changeDir();  const data = fs.readFileSync(`${safetyCheck(filepath)}`, 'utf8');  changeBack();  return data;  } catch (err) {  console.error(err);  }  }  } |
| --- |

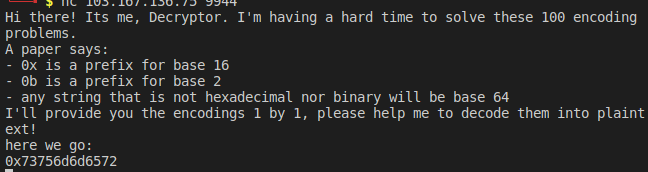
Setelah menganalisis code tersebut, kami mencari flag dengan memasukkan **..//flag.txt** pada kolom yang tersedia dan mendapatkan flag.



### Flag : NCW22{f1L7eR\_15\_n0T\_3n0u9h\_1372846}

# Misc

| Mr. Decryptor **100**  A friend of Mr. Bin, Mr. Decryptor, followed his friend's path and started to learn programming. He is headed to a series of cryptographic problems that needs to be decrypted. Please help Mr. Decryptor!  Chall: nc 103.167.136.75 9944  Author: darmads#5575 |
| --- |



Ketika melakukan nc diberikan instruksi untuk melakukan decode hasil yang degenerate oleh program ke dalam plaintext. Karena hanya ada dua jenis input yaitu hex dan binary maka tinggal lakukan pengecekan apakah kedua karakter yang diterima berakhiran b atau x sisanya tinggal rubah ke bytes dengan long\_to\_bytes. Berikut full solvernya

from pwn import \* *# pip install pwntools*

import json

import codecs

from Crypto.Util.number import long\_to\_bytes

from base64 import \*

ip = "103.167.136.75"

*#sock = int*

sock = 9944

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

r.recvuntil(b':\n')

r.recvuntil(b':\n')

while True:

f = r.recvline()[:-1]

if f[:2] == b'0x':

f = f[2:]

ans = int(f.decode(), 16)

ans = long\_to\_bytes(ans)

elif f[:2] == b'0b':

f = f[2:]

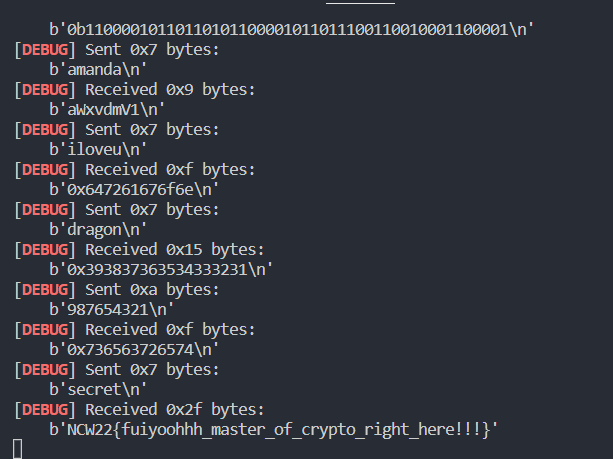
ans = int(f.decode(), 2)

ans = long\_to\_bytes(ans)

else:

ans = b64decode(f)

r.sendline(ans)



### NCW22{fuiyoohhh\_master\_of\_crypto\_right\_here!!!}