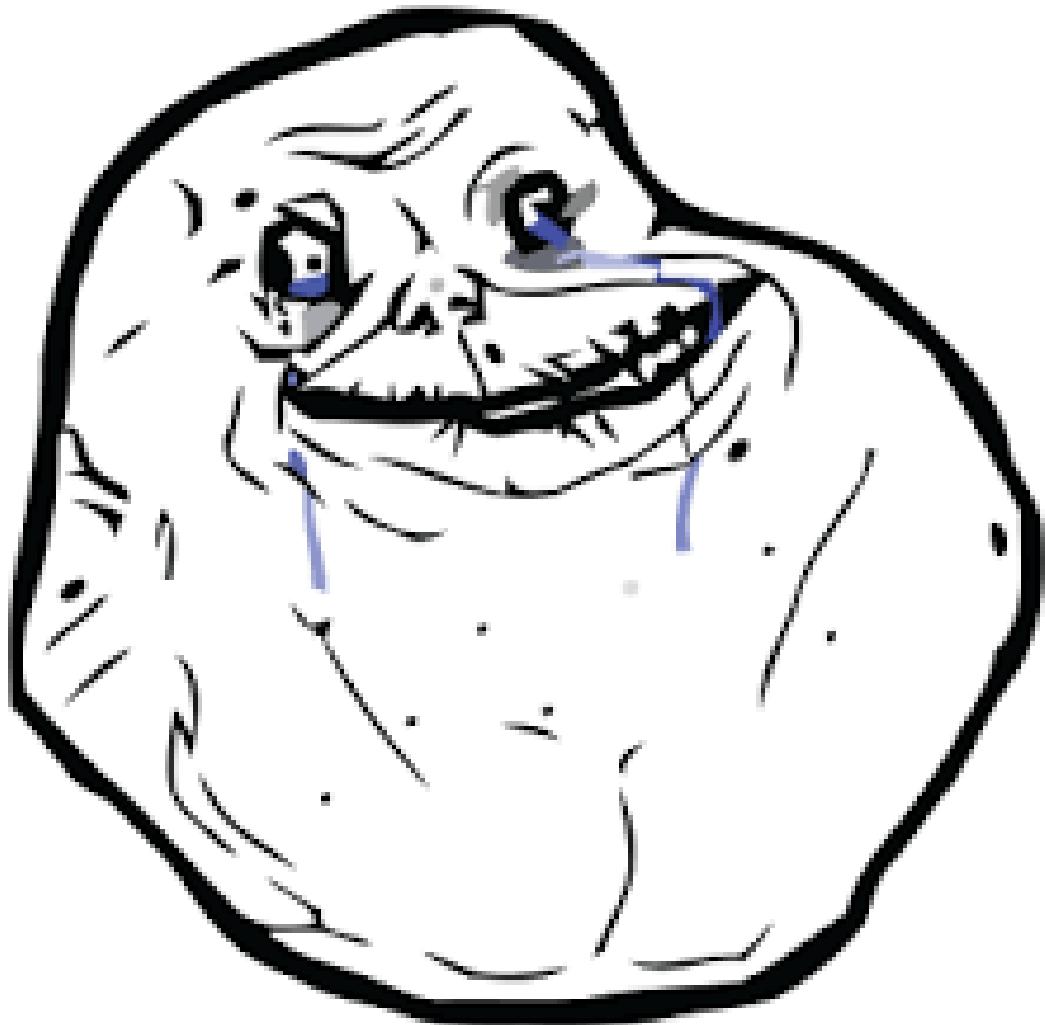


Nice Try



Bagus_Zoxce
Cacinappgbesaralaska
Lurifos

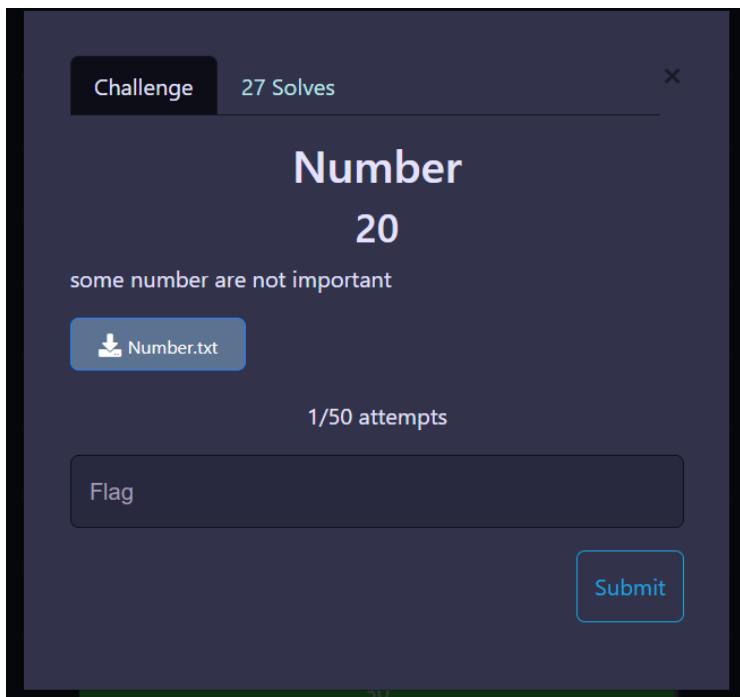
Daftar isi

Daftar isi	2
CRYpto	4
Number	4
EzRSA	5
Vigenere Cipher	6
ASCII-PI	7
RevTrans	8
RandomSHA	10
BanyakPrima	13
Liyue Scholar	16
Happy Forever	19
REV	21
SimpleActivationKey	21
Misc	23
Wordle Moment	23
OSINT	26
‘JOB’	26
Dimana	27
Commander	31
Forensic	35
Dingin	35
Citra	37
Morse	39
HierarchicalUGM	44
Forensic Digital Qiqi	45

CRYpto

Number

Some number are not important



src/cara penyelesaian

Diberikan file Number.txt yang berisi string

7020210520211020210020273202842026720284202702021232025220283202
6720233202732029520289202482028520295202712024820284202952024920
284202125.

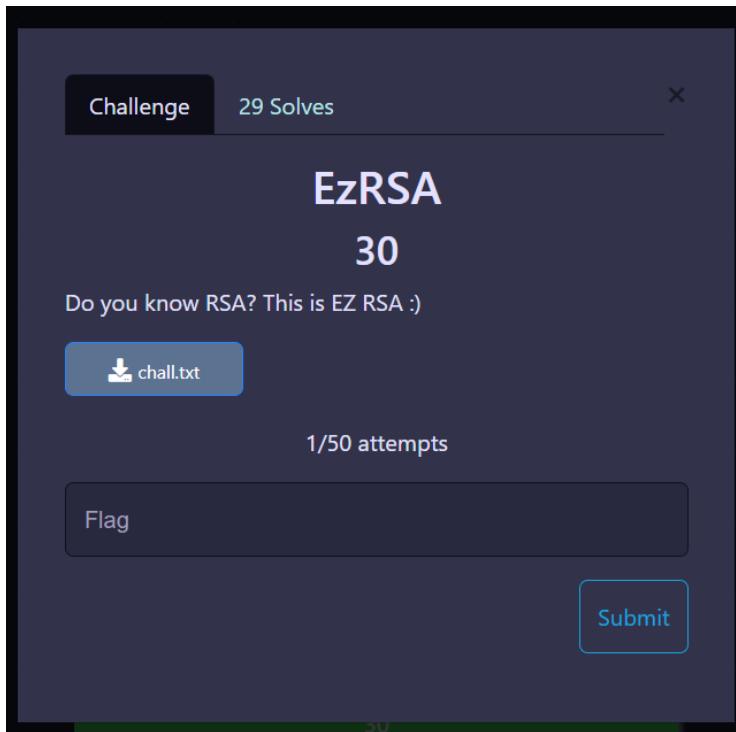
Dari deskripsi soal terdapat hint “some number are not important”, dengan melakukan pengamatan dalam string kami menemukan bahwa string `202` selalu berulang, curiga kalo ini separator akhirnya kami coba dan ubah ke ascii.

```
xs =  
"7020210520211020210020273202842026720284202702021232025220283202672023320  
2732029520289202482028520295202712024820284202952024920284202125"  
print(''.join(chr(int(a)) for a in xs.split('202')))
```

flag: `FindITCTF{4SC!_Y0U_G0T_1T}`

EzRSA

Do you know RSA? This is EZ RSA :)



src/cara penyelesaian

Diberi file chall.txt yg berisi public key rsa dan ciphertext.

```
c = 48194219261855563203215132311565813649624212082168963448568445899090
e = 65537
n = 634700503487766158115038509619422295969417670380913058190145906857689
```

Sekilas terlihat kalau **modulusnya** terlalu kecil, sehingga mudah untuk difaktorkan.

Yauds mari kita cobaaa, dengan bantuan mbah dukun factordb didapat **p** dan **q**.

Karena dah dapat **p** dan **q** maka kita bisa tau private exponent-nya.

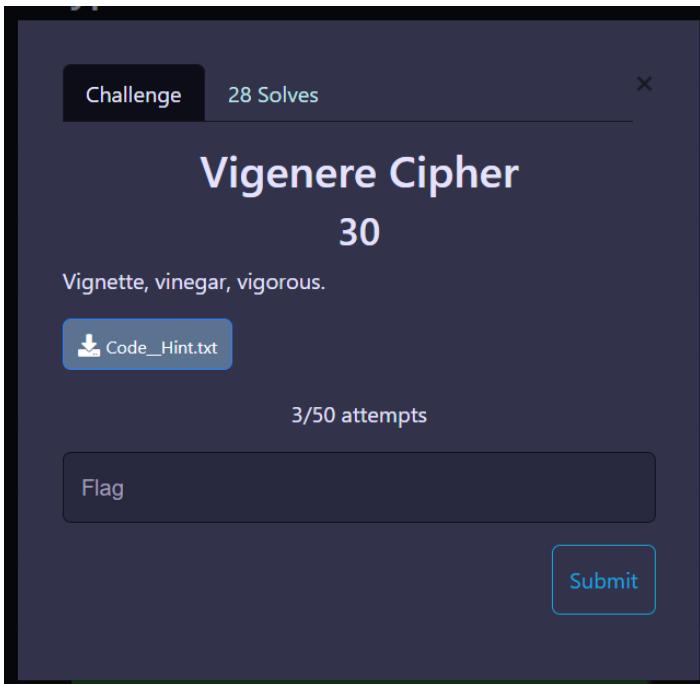
solve.py

```
p = 357239540285693222206601921599
q = 1776680439629332739835832171799514119911
from Crypto.Util.number import *
d = inverse(e, (p-1)*(q-1))
m = pow(c, d, p*q)
print(long_to_bytes(m))
```

flag: FindITCTF{3a5y_12iGht?}

Vigenere Cipher

Vignette, vinegar, vigorous.



src/cara penyelesaian

Diberi file yang berisi

Hint: vigorous and sincere. 45 5A 43 52 59 50 54 4F.

Code: XGQLYGM0ZHV1VLTKSID

Note: Masukkan jawaban yang ditemukan ke dalam format flag CTF Find IT untuk mendapatkan flag seutuhnya.

Dari judul problem sudah diberi tahu tipe enkripsinya, jadi tinggal cari tau kuncinya dan di hint terdapat 8 bilangan hex yg sus. Jadi kami coba pake bilangan itu sebagai kunci.

solve.py

```
def dekrip(ct, key):
    return "".join([chr((ord(ct[i])-ord(key[i%len(key)])) + 26) % 26 +
        ord('A')) for i in range(len(ct))])
key = "".join(chr(int(x,16)) for x in "45 5A 43 52 59 50 54 4F".split(' '))
cipher = "XGQLYGM0ZHV1VLTKSID"
print('FindITCTF{' + dekrip(cipher, key) + '}')
```

flag: FindITCTF{THOUARTAVIGENEREEE}

ASCII-PI

Hint sudah sangat menunjukkan apa yang harus dilakukan.



src/cara penyelesaian

Diberi file berisi 38 bilangan dan sebuah hint

'Alpha sierra charlie india squared' loves eating the first 10 pies. I wish everyone can divide them.

Katanya suka makan 10 pie pertama, kemudian kami cek length dari "FindITCTF{" juga merupakan 10. Jadi, kami coba cari hubungan ascii dari string tersebut dengan 10 bilangan pertama pada soal. Pas dibagi ternyata membentuk barisan bilangan pi (oh ternyata ini toh maksudnya 10 pie :D).

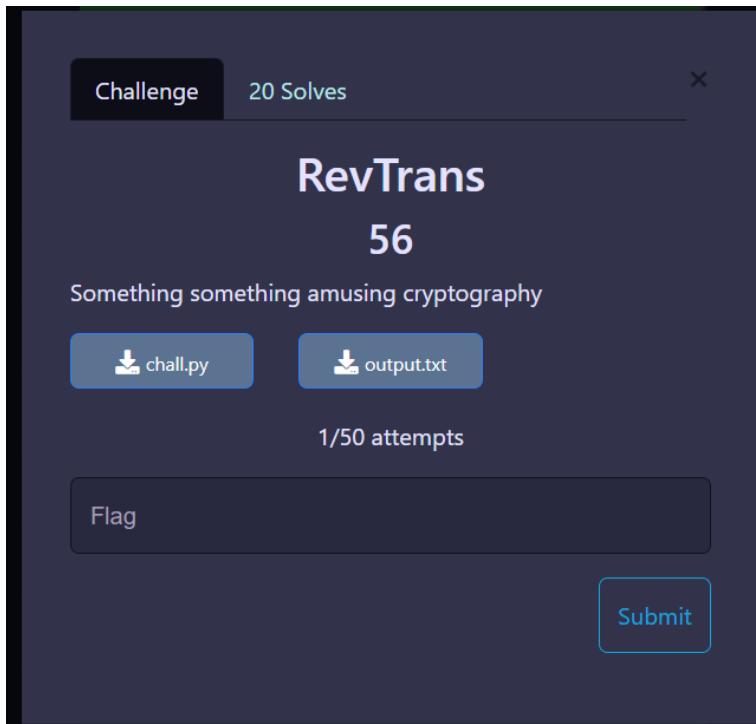
solve.py

```
with open('./crypto/asciipi/Code.txt', 'r') as f:  
    num = [int(i) for i in f.readlines()]  
  
pi =  
'3141592653589793238462643383279502884197169399375105820974944592307816406  
2862089986280348253421170679'  
  
pi = [int(i) for i in pi[:10]]  
for i,j in enumerate(num):  
    divisor = pi[i%10]  
    print(chr(j//divisor), end="")
```

flag: FindITCTF{i_10ve_c0nsumIng_3.14-pI+<3}

RevTrans

Something something amusing cryptography



src/cara penyelesaian

Diberikan file chall.py dan outputnya.

chal.py

```
flag = 'FindITCTF{REDACTED}'\n\ndef split_len(seq, length):\n    return [seq[i:i + length] for i in range(0, len(seq), length)]\n\ndef encode(key, plaintext):\n    order = {\n        int(val): num for num, val in enumerate(key)\n    }\n\n    reversetext = ''\n    ciphertext = ''\n    i=len(plaintext)-1\n    while i >= 0:\n        reversetext = reversetext + plaintext[i]\n        i=i-1\n    for index in sorted(order.keys()):\n        ciphertext = ciphertext + reversetext[order[index]]\n\n    return ciphertext
```

```

for part in split_len(reversetext, len(key)):
    try:ciphertext += part[order[index]]
    except IndexError:
        continue
return ciphertext
print(encode('4321', flag))

```

Fungsi `split_len` digunakan untuk men-split string menjadi bagian-bagian dengan panjang length.

Fungsi `encode` untuk mengenkripsi. Pertama-tama `plaintext` direverse, kemudian untuk setiap huruf pada part dari `split_len` akan diganti posisinya sesuai dict `order`.

Awalnya kami bingung gimana cara bikin decryptor-nya. Tapi, karena panjang flagnya diketahui (dari `ciphertext`) dan juga keynya tertera di source code. Maka kami coba untuk membuat flag custom sendiri, yaitu `flag='1234'*7`. Setelah dienkripsi ternyata jadi `'1111111222222333333444444'` wah menarik. Jadi, untuk deskripsinya ya tinggal petakan aja dari `ciphertext` ke flag custom kita tadi. Seperti judul soalnya, reverse dan transpose :D

solve.py

```

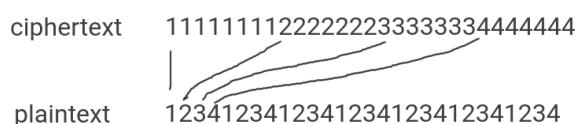
def decrypt(ct):
    x = ''
    for j in range(7):
        for i in range(j, len(ct), 7):
            x += ct[i]

    parts = split_len(x[::-1], 4)
    for i in parts:
        print(i[::-1], end="")

decrypt(enc)

```

Fungsi `decrypt` dia mengambil setiap karakter dan memindah posisinya

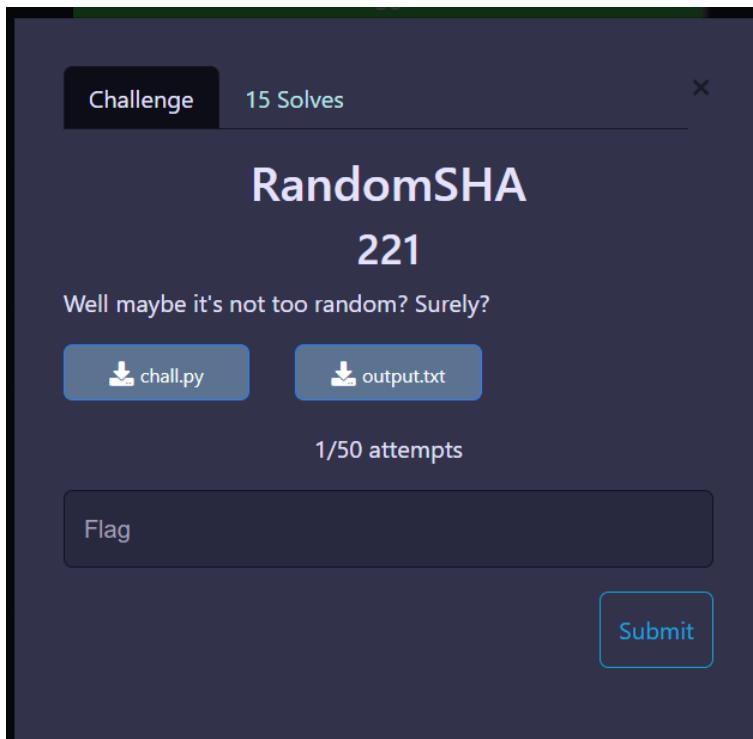


Setelah jadi kemudian di-reverse dan displit, dan tiap part sudah merupakan flag, hanya saja urutan terbalik, sehingga harus direverse lagi.

flag: `FindITCTF{you_solve_it_yeay}`

RandomSHA

Well maybe it's not too random? surely?



src/cara penyelesaian

Diberikan file `chall.py` dan outputnya.

chall.py

```
import random, string
import hashlib
flag = "FindITCTF{REDACTED}"
enc_flag = ""
random.seed("FINDIT")
now = ""
ct = []
for c in flag:
```

```

if c.islower():
    enc_flag += chr((ord(c)-ord('a')+random.randrange(0,26))%26 + ord('a'))
elif c.isupper():
    enc_flag += chr((ord(c)-ord('A')+random.randrange(0,26))%26 + ord('A'))
elif c.isdigit():
    enc_flag += chr((ord(c)-ord('0')+random.randrange(0,10))%10 + ord('0'))
else:
    enc_flag += c
for c in enc_flag:
    now += c
    ct.append(
        int(hashlib.sha512(now.encode()).hexdigest(), 16)>>256
    )
print(f"ct = {ct}")

```

File ini melakukan random shift terhadap plaintext dan kemudian menyimpannya dalam bentuk **hash**. Dapat dilihat bahwa library random menggunakan seed statis, sehingga nilai randomnya bisa diregenerasi ulang.

Karena **ciphertext** berupa hash yang merupakan oneway function, sehingga satu-satunya cara untuk mendapatkan **plaintext**/flag adalah dengan bruteforce.

**note: variable ct dipersingkat agar penulisan wu tidak memakan banyak halaman*

solve.py

```

ct = # output.txt
import random, hashlib, string
lib = string.printable

def getEnc(f):
    random.seed("FINDIT")
    enc = ""
    for c in f:
        if c.islower():
            enc += chr((ord(c)-ord('a')+random.randrange(0,26))%26 + ord('a'))
        elif c.isupper():
            enc += chr((ord(c)-ord('A')+random.randrange(0,26))%26 + ord('A'))
        elif c.isdigit():
            enc += chr((ord(c)-ord('0')+random.randrange(0,10))%10 + ord('0'))

```

```
        else:
            enc += c
        return enc

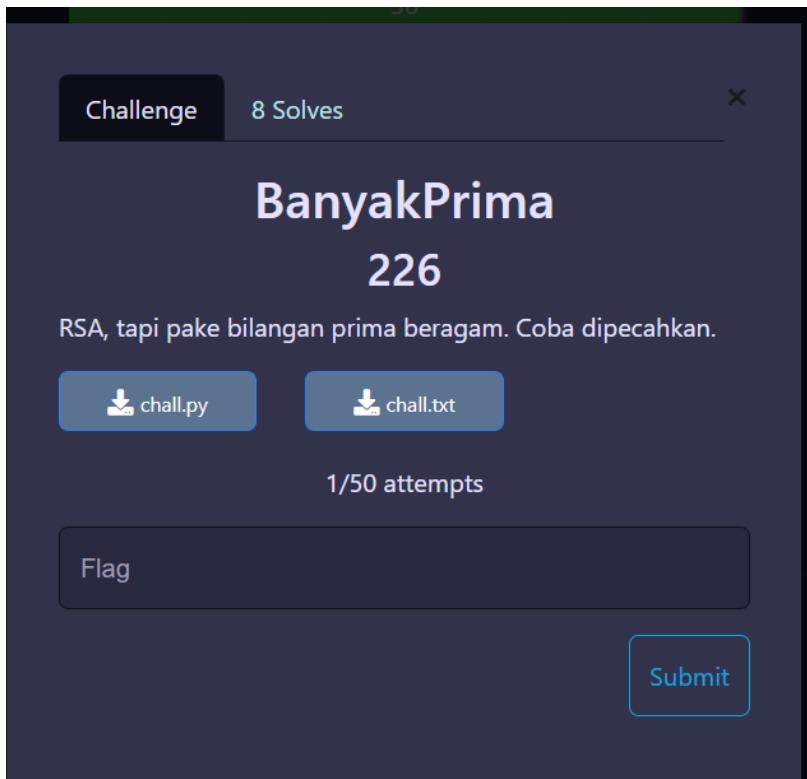
known_flag = "FindITCTF"
while 1:
    for guess in lib:
        enc_flag = getEnc(known_flag+guess)
        now = ""
        for i,c in enumerate(enc_flag):
            now += c
            if int(hashlib.sha512(now.encode()).hexdigest(), 16)>>256 != ct[i]:
                break
            else:
                known_flag += guess
                break
    if known_flag[-1] == '}':
        print(known_flag)
        break
```

Fungsi `getEnc` untuk mengenkripsi flag tebakan (`seed` di reset tiap dipanggil), kemudian hasilnya di-hash. Apabila hasil hash sama dengan `ciphertext` maka tebakan benar, jika berbeda maka lanjut ke tebakan berikutnya.

flag: `FindITCTF{W3ll_R4nd0m_4nd_SHA_r1ghtt?}`

BanyakPrima

RSA, tapi pake bilangan prima beragam. Coba dipecahkan.



src/cara penyelesaian

Diberi file **chall.py** dan outputnya.

chall.py

```
from Crypto.Util.number import *
from secret import flag

def nextPrime(prime):
    if isPrime(prime):
        return prime
    else:
        return nextPrime(prime+1)

p = getPrime(128)
q = nextPrime(7*p)
r = nextPrime(p*q)
s = nextPrime(q*r)
n = p*q*r*s
```

```

phin = (p-1)*(q-1)*(r-1)*(s-1)
e = 65537
assert GCD(e,phin) == 1
m = bytes_to_long(flag.encode("ascii"))
c = pow(m,e,n)
c = long_to_bytes(c).hex()
print(c)
print(n)

```

Pertama-tama kami coba faktorisasi modulus `n` menggunakan mbah dukun factordb, namun hasilnya nihil. `N` terlalu besar untuk difaktorisasi.

Kemudian kami curiga apa jangan-jangan private keynya cukup kecil, sehingga kami mencoba menggunakan `wiener attack`. Namun hasilnya tetap nihil.

Oke, kita coba `boneh-durfee` dengan harapan private keynya cukup kecil. Namun, tetap tidak bisa T_T.

Lalu saya memutuskan untuk membaca ulang scnya. Ternyata prime factor yang random hanyalah `p`, sedangkan `q, r, s` dibuat dengan fungsi `nextprime`. Sehingga `q, r, s` dapat dinyatakan dalam `p`. Apa bisa pakai `fermat factorization`? Ternyata tidak, karena fermat cuma bisa untuk 2 faktor prima, sedangkan ini punya 4.

Kemudian kami coba membuat persamaan untuk modulusnya.

$$\begin{aligned} q &= 7p + i \\ r &= pq + j \\ s &= qr + k \end{aligned}$$

dengan i, j, k adalah konstanta yang kecil. Karena cukup kecil maka kita bisa abaikan.

$$\begin{aligned} n &= pqrs \approx p(7p)(7p^2)(49p^3) \\ n &\approx 7^4 p^7 \end{aligned}$$

Jadi, nilai p sekitar:

$$p \approx \sqrt[7]{\frac{n}{7^4}}$$

Sampai sini sudah terlihat bayangan solusi. Karena kita tahu aproksimasi nilai `p` maka kita bisa mencarinya.

solve.py

```
from Crypto.Util.number import *
import gmpy2
from output import n,e,c

def nextPrime(prime):
    if isPrime(prime):
        return prime
    else:
        return nextPrime(prime+1)

def getN(p):
    q = nextPrime(7*p)
    r = nextPrime(p*q)
    s = nextPrime(q*r)
    return p*q*r*s

def find_p(p_guess,n):
    while True:
        n_guess = getN(p_guess)
        if n < n_guess:
            p_guess = p_guess - 2
        elif n > n_guess:
            p_guess = p_guess + 2
        else:
            assert (n == n_guess)
            return p_guess

guess = gmpy2.iroot(n//(7**4),7)[0]
guess += 1
# memastikan kalo guess adalah bilangan ganjil
assert(guess%2 == 1)
p = find_p(guess,n); q = nextPrime(7*p)
r = nextPrime(p*q); s = nextPrime(q*r)
phi = (p-1)*(q-1)*(r-1)*(s-1)
d = inverse(e,phi)
print(long_to_bytes(pow(c,d,n)))
```

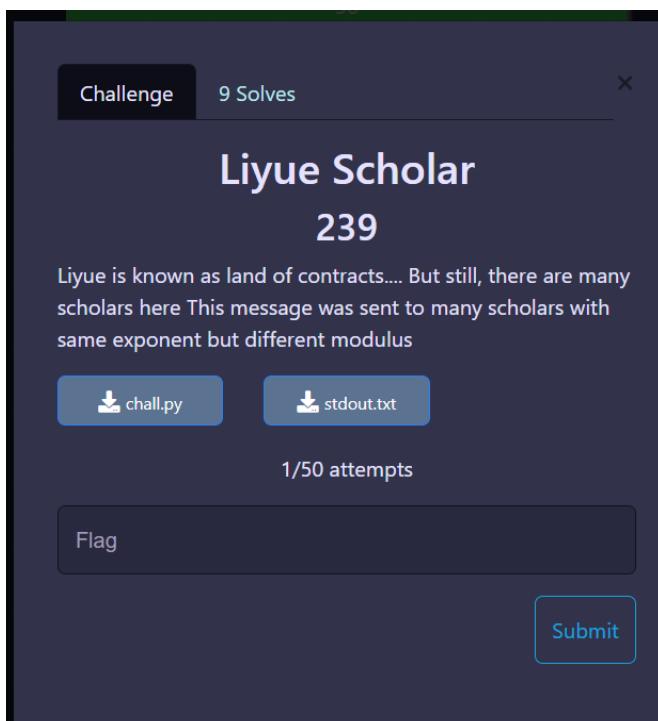
Pertama-tama kita tebak bahwa p adalah $\sqrt[7]{\frac{n}{7^4}}$, setelah itu kita generate q, r, s . Kemudian kalikan semua, jika $pqr < n$, maka increment nilai p dan sebaliknya.

Terakhir check apakah tebakan sudah benar dengan assert. Setelah dapat `p`, maka kita bisa membuat private keynya.

flag: `FindITCTF{Mult1Pr1m3_RSA_HuH??_abcdefghiJ}`

Liyue Scholar

Liyue is known as land of contracts.....But still, there are many scholars here. This message was sent to many scholars with same exponent but different modulus.



src/cara penyelesaian

Diberi file chall.py dan outputnya.

chall.py

```
from Crypto.Util.number import getPrime, bytes_to_long
from secret import FLAG, e

m = bytes_to_long(FLAG)
modulos = []
ciphertexts = []
```

```

for i in range(1,e+1):
    p = getPrime(256)
    q = getPrime(256)

    N = p*q

    print(f"n_{i} = {N}, c_{i} = {pow(m, e, N)}")
    modulus.append(N)
    ciphertexts.append(pow(m, e, N))

```

Sekilas terlihat kalo pesan dienkripsi sebanyak e kali dengan modulus yang berbeda beda. Kemudian kami cek outputnya dan ternyata terdapat 7 output sehingga dapat diketahui kalau nilai e adalah 7.

Dari source code dapat dibuat persamaan bahwa c_1, c_2, \dots, c_7 adalah sebagai berikut

$$\begin{aligned}
c_1 &\equiv m^7 \pmod{n_1} \\
c_2 &\equiv m^7 \pmod{n_2} \\
&\dots \\
c_7 &\equiv m^7 \pmod{n_7}
\end{aligned}$$

Karena semua nilai n diketahui, kita bisa mencari nilai m^7 menggunakan Chinese Remainder Theorem. (Tentu Saja dengan bantuan geeksforgeeks)

solve.py

```

import gmpy2
gmpy2.get_context().precision = 4096
from Crypto.Util.number import *
from gmpy2 import root
from output import *

def chinese_remainder_theorem(items):
    N = 1
    for a, n in items:
        N *= n
    result = 0
    for a, n in items:
        m = N // n
        r, s, d = extended_gcd(n, m)
        if d != 1:

```

```

        raise "error"
    result += a * s * m
    return result % N

def extended_gcd(a, b):
    x, y = 0, 1
    lastx, lasty = 1, 0
    while b:
        a, (q, b) = b, divmod(a, b)
        x, lastx = lastx - q * x, x
        y, lasty = lasty - q * y, y
    return (lastx, lasty, a)

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 get_value(filename):
    with open(filename) as f:
        value = f.readline()
    return int(value, 16)

ciphertexts = [c_1, c_2, c_3, c_4, c_5, c_6, c_7]
modulus = [n_1, n_2, n_3, n_4, n_5, n_6, n_7]
m7 = chinese_remainder_theorem([(c_1, n_1), (c_2, n_2), (c_3, n_3), (c_4, n_4), (c_5, n_5), (c_6, n_6), (c_7, n_7)])
print(long_to_bytes(int(root(m7, 7)))))

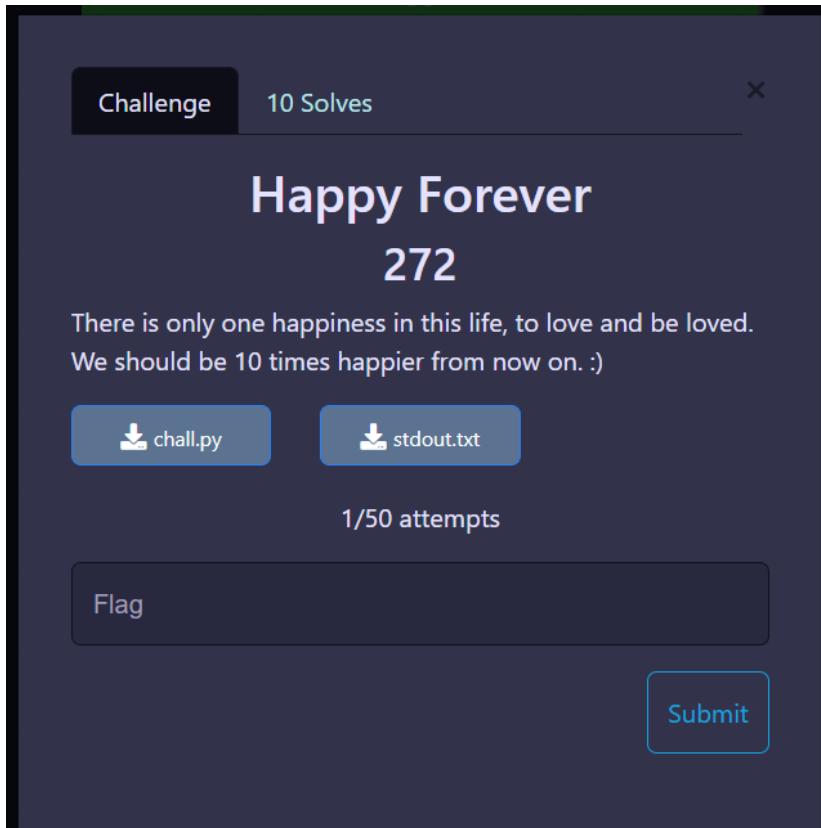

```

Dari Chinese Remainder Theorem kita dapat m^7 sehingga tinggal di akarkan dan dapat plaintextnya.

flag: FindITCTF{Ju5T_51Mpl3_ch1N353_r3M41ND3r_tH30r3M_1234567}

Happy Forever

There is only one happiness in this life, to love and be loved. We should be 10 times happier from now on :)



src/cara penyelesaian

Diberikan file **chall.py** dan outputnya.

```
from secret import HAPPY_NUM, FLAG
import base64

FLAGenc = FLAG[::-1].encode().hex()
ciphertext = [(ord(c)^HAPPY_NUM[i]+i)) for i,c in enumerate(FLAGenc)]

print(f"cipher = {ciphertext}")
```

Plaintext dienkripsi dengan alur `reverse -> hex -> xor`.

Awalnya kami bingung gimana solve problem ini, saat mencoba mendapatkan `HAPPY_NUM` menggunakan xor dengan "[FindITCTF](#)" kami mendapat bahwa `HAPPY_NUM` tidak berpola. Jadi sepertinya susah, lalu kemudian kami coba search

“Happy Number” di google dan sampailah di [wikipedia](#). Dari sini kami baru tau kalo happy number is a thing dan saat kami cocokkan dengan `HAPPY_NUM` ternyata sama.

Okey dah dapat kuncinya, jadi tinggal decrypt dengan xor karena xor fungsi simetris.

Step dekripsi `xor -> unhex/toascii -> reverse`

solve.py

```
ct = [54, 108, 63, 41, 36, 37, 17, 31, 27, 12, 8, 118, 97, 101, 85, 3, 88, 9, 64, 64, 79, 186, 164, 160, 174, 149, 246, 249, 237, 189, 232, 131, 223, 146, 206, 305, 317, 291, 294, 286, 284, 374, 379, 358, 362, 367, 337, 270, 344, 328, 333, 332, 328, 439, 425, 405, 403, 400, 415, 469, 388, 399, 500, 500, 509, 424, 480, 488, 458, 608, 547, 553, 533, 512, 512, 515, 599, 588, 692, 738, 693, 640, 652, 675, 759, 763, 740, 736, 738, 750, 726, 732, 735, 724, 705, 714, 823, 876, 827, 800, 788, 796]

HAPPY_NUM = [1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49, 68, 70, 79, 82, 86, 91, 94, 97, 100, 103, 109, 129, 130, 133, 139, 167, 176, 188, 190, 192, 193, 203, 208, 219, 226, 230, 236, 239, 262, 263, 280, 291, 293, 301, 302, 310, 313, 319, 320, 326, 329, 331, 338, 356, 362, 365, 367, 368, 376, 379, 383, 386, 391, 392, 397, 404, 409, 440, 446, 464, 469, 478, 487, 490, 496, 536, 556, 563, 565, 566, 608, 617, 622, 623, 632, 635, 637, 638, 644, 649, 653, 655, 656, 665, 671, 673, 680, 683, 694, 700, 709, 716, 736, 739, 748, 761, 763, 784, 790, 793, 802, 806, 818, 820, 833, 836, 847, 860, 863, 874, 881, 888, 899, 901, 904, 907, 910, 912, 913, 921, 923, 931, 932, 937, 940, 946, 964, 970, 973, 989, 998, 1000]

assert(len(ct) %2 == 0)

pt_hex = ''

for i,c in enumerate(ct):
    pt_hex += chr(c^(HAPPY_NUM[i]+i))

print(bytes.fromhex(pt_hex).decode()[:-1])
```

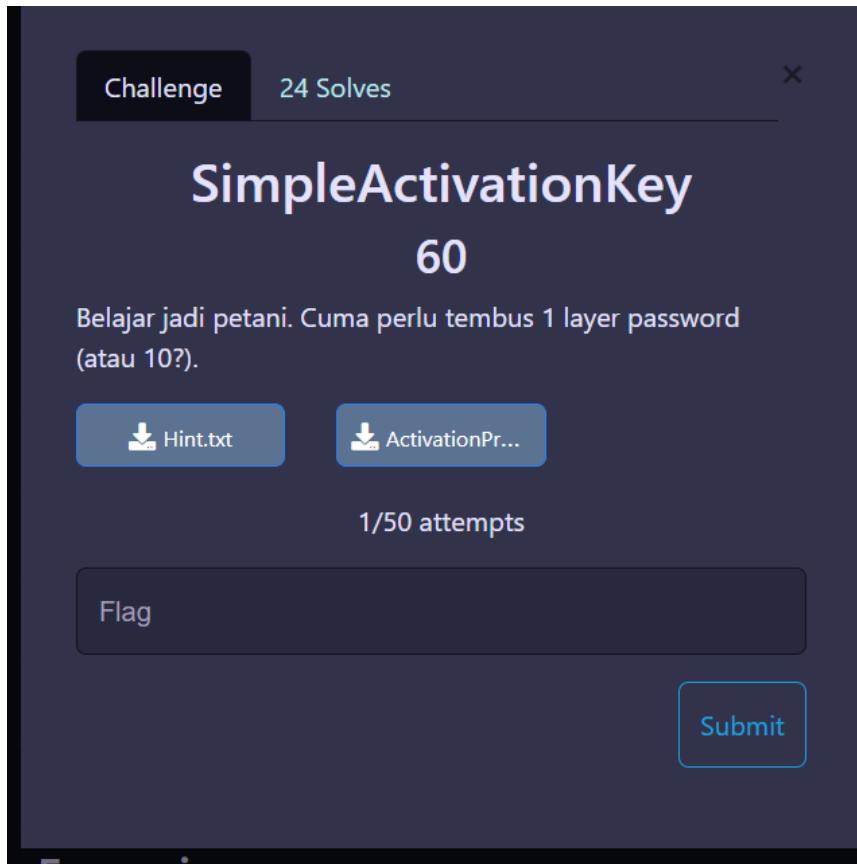
Pertama-tama kita xor dulu untuk mendapatkan plaintext dalam bentuk hex, kemudian ubah ke ascii dan terakhir reverse.

flag: `FindITCTF{1_5H0Uld_B3_h4pPy_4f73r4Ll_r19H7?_999999}`

REV

SimpleActivationKey

Belajar jadi petani. Cuma perlu tembus 1 layer password (atau 10?).



src/cara penyelesaian

Diberikan file executable dan sebuah hint.

Pertama-tama cek tipe filenya,

```
[lurifos@Bondowoso] 🐾 [~/ctf/2022/findit/qual/rev/simple_activator]
(master)$ file ActivationProgramR1.exe
ActivationProgramR1.exe: PE32 executable (console) Intel 80386, for MS Windows
```

Hmm ternyata file executable windows. Karena kami malas ganti os, jadi langsung aja load ghidra hehe :D.

Dan langsung ketemu flagnya ada di .rdata

```
00405051 00          ;           0001  
          s_Correct!_The_flag_is:_FindITCTF{_004050a0      XREF[1]:    _main:0040155d(*)  
004050a0 43 6f 72      ds      "Correct!\nThe flag is: FindITCTF{j4D1_Sk1Dr0w...  
72 65 63  
74 21 0a ...
```

Alternate solution:

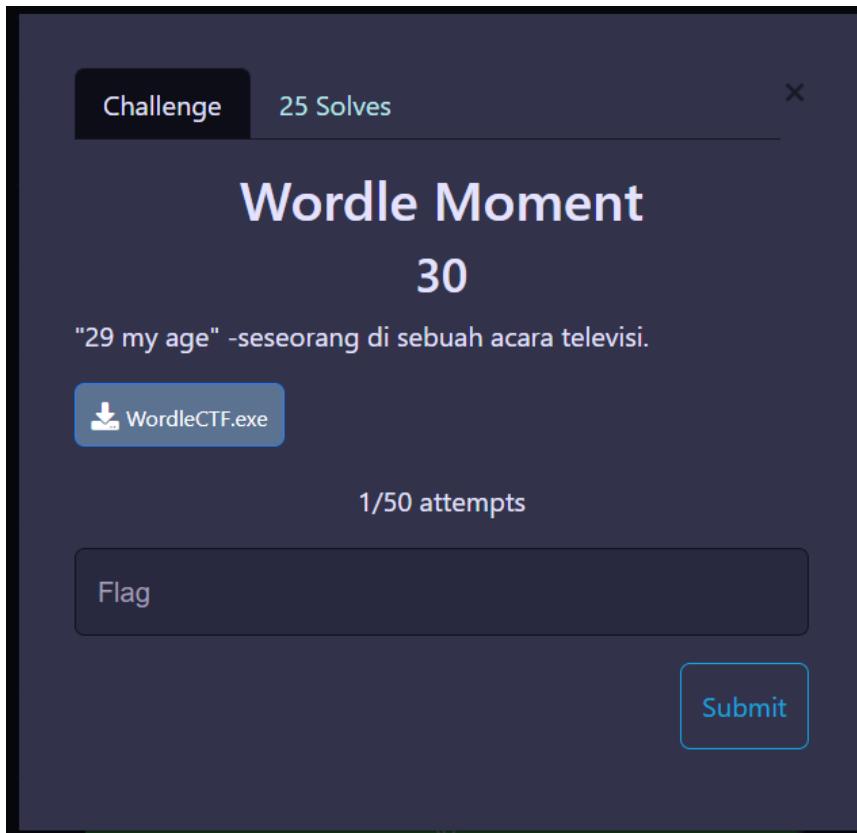
```
$ strings ActivationProgramR1.exe | grep CTF  
The flag is: FindITCTF{j4D1_Sk1Dr0w_aD4LAH_Imp!4NkU}
```

flag: `FindITCTF{j4D1_Sk1Dr0w_aD4LAH_Imp!4NkU}`

Misc

Wordle Moment

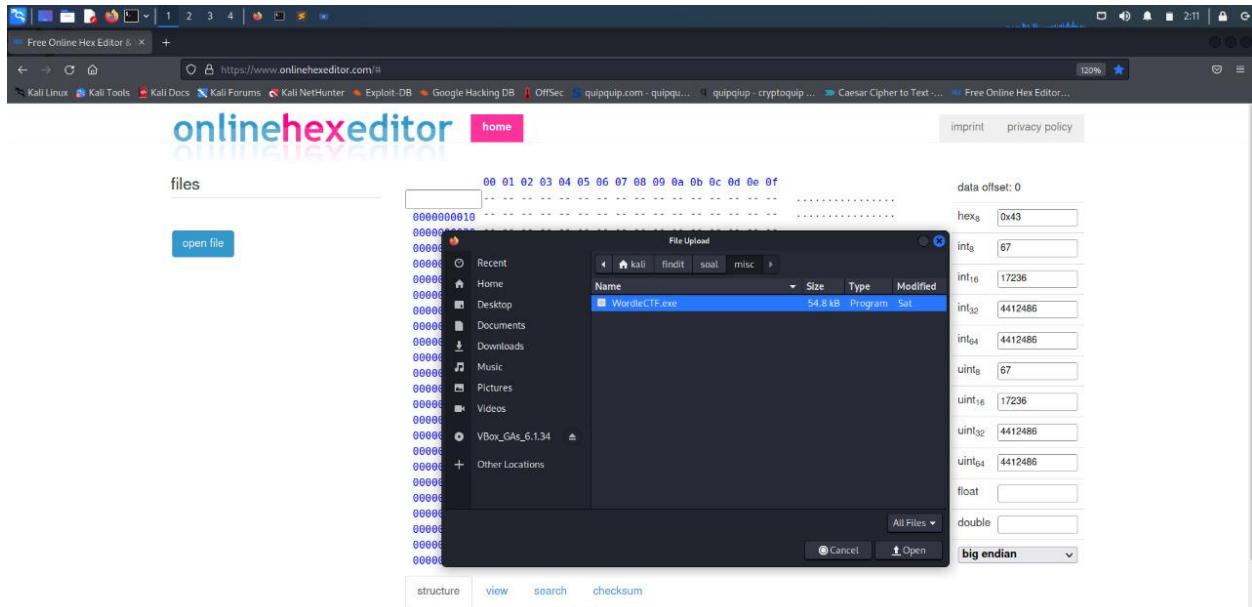
"29 my age" -seseorang di sebuah acara televisi.

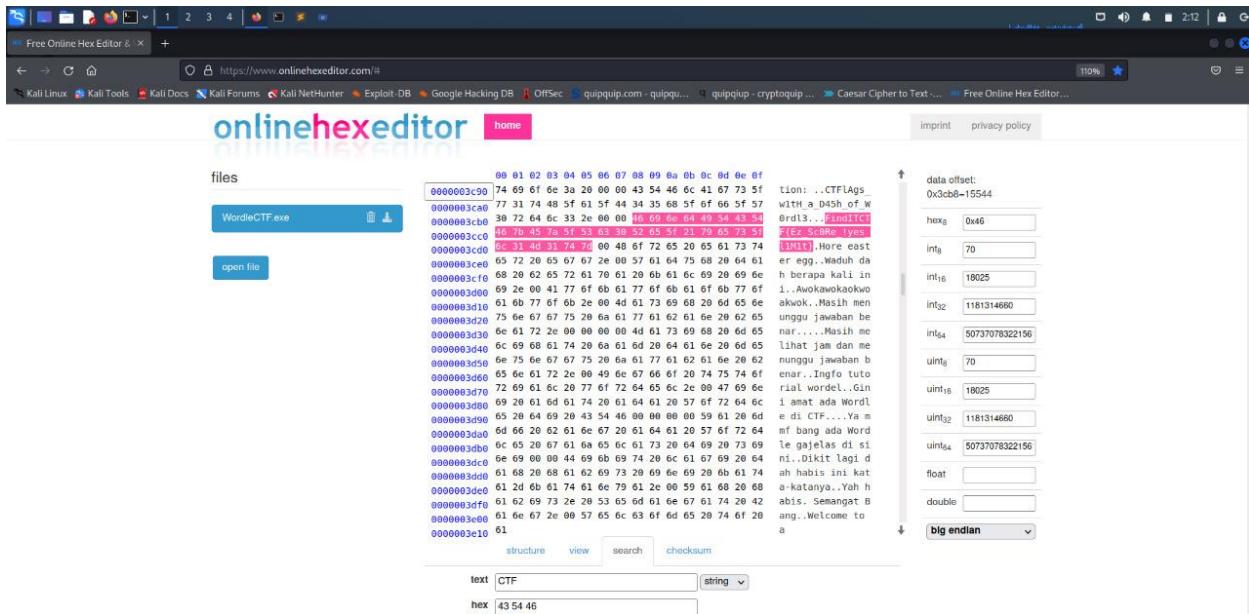


src/cara penyelesaian

Diberikan sebuah file WordleCTF.exe, hal yang kami lakukan memeriksa file tersebut dengan command 'File & cat'. Selain itu kami mencoba membukanya pada sublime text untuk mendapatkan informasi lebih lanjut.

Ketika kami melakukan hal tersebut kami berasumsi bahwa flag tersebut berada pada Hexadesimal karena pada command ‘cat’ terdapat banyak sekali string, maka dari itu kami mencoba untuk membukanya pada laman web Hex editor Online





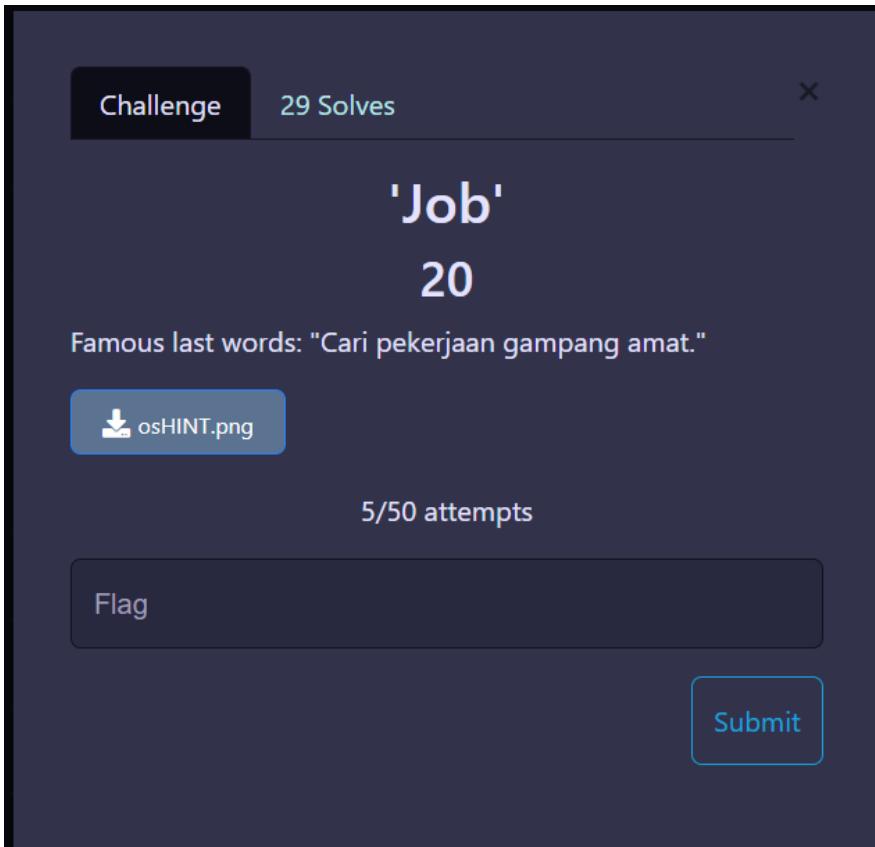
Setelah itu kamu mencoba untuk mencari kata kunci untuk mendapatkan flag tersebut yaitu 'CTF', dan benar sesuai dugaan bahwa flag tersebut ada di bilangan Heksadesimal.

flag: **FindITCTF{Ez_Sc0Re_!yes_l1Mit}**

OSINT

'JOB'

Famous last words: "Cari pekerjaan gampang amat".



src/cara penyelesaian

Diberikan gambar screenshot berupa experience LinkedIn seseorang.



Kami menduga flagnya berada di akun LinkedIn tersebut sehingga kami mencari akun yang berkaitan dengan GMRT dan apply filter Hardware Programmer. Kami menemukan akun milik Giga Hidjrika Aura dan melihat flag pada kolom Activity.

Activity

171 followers

+ Follow

Giga Hidjrika Aura Adkhy commented on a post • 4d

You found the flag! It's FindITCTF{Op3n_s0urc3_InGf0}

8

4 comments

Flag: `FindITCTF{Op3n_s0urc3_InGf0}`

Dimana

This photo was taken somewhere near Yogyakarta City. Can you find the name of the street taken from this photo?

(Bracket it with FindITCTF{StreetName}).

Challenge 30 Solves X

Dimana

30

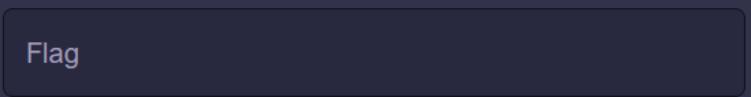
This photo was taken somewhere near Yogyakarta City. Can you find the name of the street taken from this photo? (Bracket it with FindITCTF{StreetName})

 [chall.png](#)

1/50 attempts

Flag

Submit



src/cara penyelesaian

Diberikan gambar Google Maps chall.png



Terlihat tulisan TK Khalifah di bagian kiri, bimbingan belajar pada papan iklan, dan percetakan di bagian kanan.

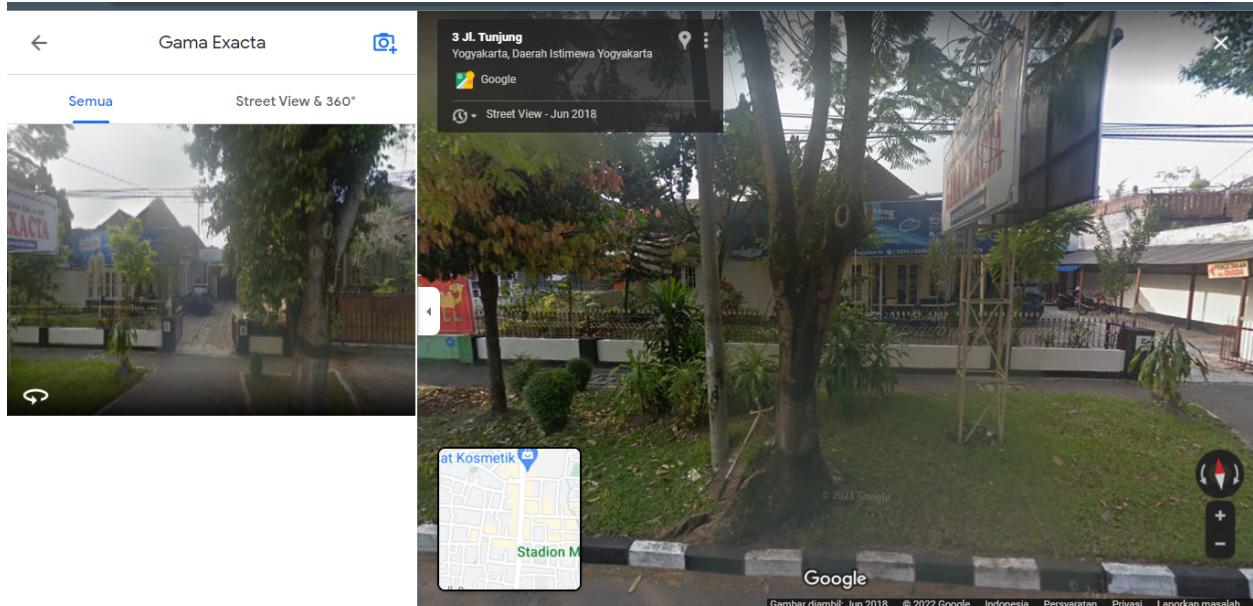


Nama LBB yang ada kurang jelas sehingga kami mencari TK Khalifah terlebih dahulu di Google. Namun scope TK Khalifah terlalu luas sehingga sulit mendeteksi yang ada pada gambar.

The screenshot shows a Google search results page for "tk khalifah jogja". The first result is "TK Khalifah Sewon" with a rating of 3.0 stars. The second result is "TK Khalifah Bantul (Jl Lingkar Timur Ringroad Bantul)" with a rating of 5.0 stars. The third result is "TK Khalifah Wirobrajan" with a rating of 5.0 stars. The fourth result is "TK Khalifah Baciro" with a rating of 5.0 stars. Below the results, it says "Menampilkan hasil 1 - 15" and there is a checkbox for "Perbarui hasil bila peta digeser". To the right of the search results is a map of TK Khalifah Bantul located at Jl. Lkr. Timur, Area Sawah, Trirenggo, Kec. Bantul, Kabupaten Bantul, Daerah Istimewa Yogyakarta. The map shows the building's location relative to other landmarks like "KUNIN DASTER BANTUL - Toko..." and "Soto Bumbong Manding".

Setelah pemikiran panjang kami menebak kata pertama di papan iklan LBB adalah GAMA sehingga kami mencari "Lembaga Bimbingan Belajar Gama Jogja" di Google.

The screenshot shows a Google search results page for "lembaga bimbingan belajar gama jogja". The top result is "Bimbining Belajar Gama Jogja - Pusat Belajar" with a link to <https://gama jogja-kukusan.business.site>. The snippet says: "Sehubung telah mendekati penilaian akhir semester genap, kami dari LBB Gama Jogja ingin menawarkan kepada siswa/ yang berminat untuk melanjutkan bimbingan ...". To the right is the logo for "Bimbining Belajar Gama Jogja". The second result is "LBB GAMA EXACTA YOGYAKARTA – Just another ..." with a link to <https://gamaexacta.com>. The snippet says: "Selamat Datang di Website Resmi Gama Exacta Yogyakarta. Salam Hormat, Meskipun nilai UN bukan sebagai syarat mutlak kelulusan siswa, namun NEM tersebut akan ...". Below the snippet is a small image of an office interior. At the bottom of the search results, it says "Tidak ada: lembaga | Harus menyertakan: lembaga" and "Anda mengunjungi halaman ini pada 08/05/22".



Tulisan di papan tersebut adalah Gama Exacta yang terletak di Jl. Tunjung.

flag: **FindITCTF{Tunjung}**

Commander

This photo is in Indonesia. Find the full name of the commander and his title without spaces and full capslock who are still serving in that place this year.(Bracket it with FindITCTF{COMMANDERNAME,TITLEABBREVIATION}).

Challenge 28 Solves X

Commander

45

This photo is in Indonesia. find the full name of the commander and his title without spaces and full capslock who are still serving in that place this year.(Bracket it with FindITCTF{COMMANDERNAME,TITLEABBREVIATION})

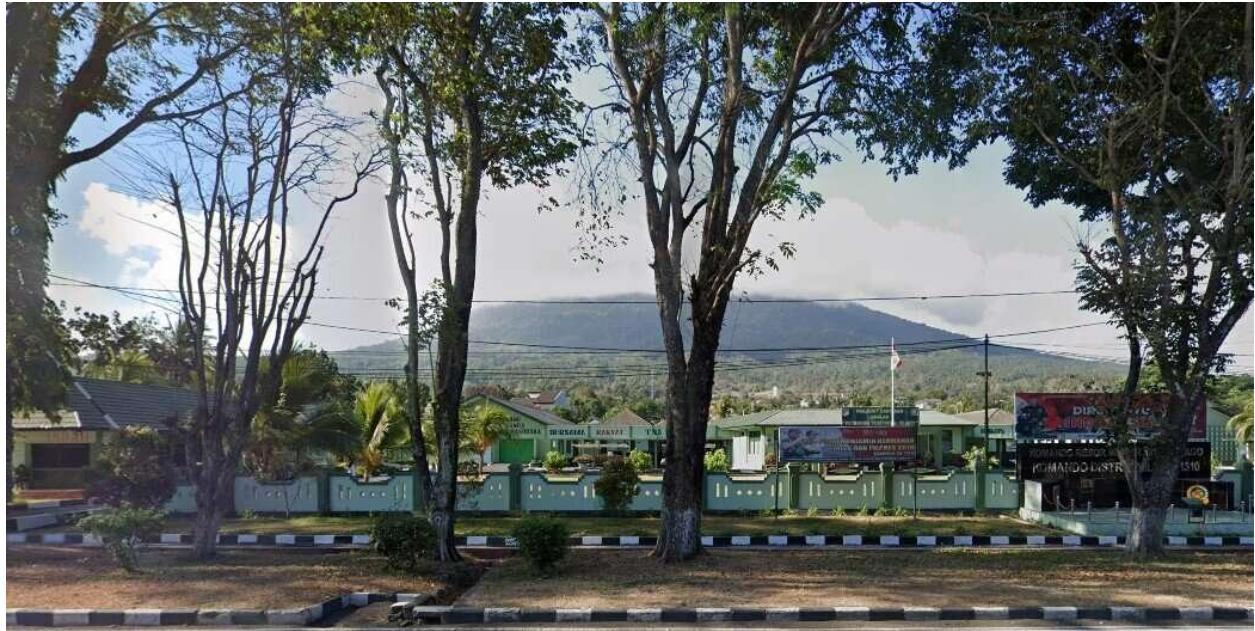
[pict_1.jpg](#)

31/50 attempts

Flag

Submit

```
src/cara penyelesaian
```



Diberikan gambar pict_1.jpg dan kita disuruh untuk mencari nama komandan yang menjabat di tempat tersebut tahun ini. Flag yang dibutuhkan adalah FindITCTF{NAMAKOMANDAN,SINGKATANGELAR}.



Hal pertama yang terlihat jelas adalah tulisan Komando Distrik pada pagar tembok yang menandakan bahwa ini adalah bangunan militer dan kita sedang mencari komandan KODIM. Di sebelah kanan pohon terlihat nomor yang tidak begitu jelas. Tebakan awal kami adalah 1510, lalu kami mencari Komando Distrik Militer 1510 di Google, namun tidak mendapatkan hasil yang memuaskan. Pada tembok tersebut terlihat juga KOMANDO RESOR dan AGO. Lalu kami mencari daftar Komandan Resor Militer di Wikipedia.

28.	Komando Resor Militer 092/Maharajalila	Tanjung Selor	Kodam VI/Mulawarman	A	korem101antasari.mil.id ^{ago}	1/2			
29.	Komando Resor Militer 101/Antasari	Banjarbaru		A	korem101antasari.mil.id ^{ago}				
30.	Komando Resor Militer 102/Panju Panjung	Palangka Raya	Kodam XII/Tanjungpura	A	korem102panjupanjung.com ^{ago} Diarsipkan ²⁰¹⁹⁻¹¹⁻²⁴ di Wayback Machine.				
31.	Komando Resor Militer 121/Alambhana Wanawai	Sintang		A	korem121abw.mil.id ^{ago}				
32.	Komando Resor Militer 131/Santiago	Manado	Kodam XIII/Merdeka	A	korem131santi ^{ago} .blogspot.com ^{ago}				
33.	Komando Resor Militer 132/Tadulako	Palu		A	korem132-tniad.mil.id ^{ago}				
34.	Komando Resor Militer 133/Nani Wartabone	Gorontalo		A	korem133.blogspot.com ^{ago}				
35.	Komando Resor Militer 141/Toddopuli	Watampone	Kodam XIV/Hasanuddin	A	korem141.kodam14hasanuddin-tniad.mil.id ^{ago} Diarsipkan ²⁰¹⁹⁻¹¹⁻¹⁷ di Wayback Machine.				
36.	Komando Resor Militer 142/Taroada Tarogau	Mamuju		A	korem142.kodam14hasanuddin-tniad.mil.id ^{ago} Diarsipkan ²⁰¹⁹⁻¹¹⁻¹⁷ di Wayback Machine.				
37.	Komando Resor Militer 143/Halu Oleo	Kendari		A	korem143.kodam14hasanuddin-tniad.mil.id ^{ago} Diarsipkan ²⁰¹⁹⁻¹²⁻⁰² di Wayback Machine.				
38.	Komando Resor Militer 151/Binaliya	Ambon	Kodam XVI/Pattimura	A	korem151binaliya.mil.id ^{ago} Diarsipkan ²⁰¹⁹⁻¹⁰⁻²⁵ di Wayback Machine.				
39.	Komando Resor Militer 152/Baabullah	Ternate		A	korem152-tniad.mil.id ^{ago}				

Terdapat satu Korem yang berakhiran AGO yakni Santiago. Kami kemudian mencari daftar Kodim yang dibawahi Korem Santiago dan menemukan satu yang memiliki nomor mirip dengan gambar yakni 1310.

1.1 Satuan Teritorial 1.2 Satuan Samping 2 Komandan 3 Referensi	Staf Setiawan
--	--------------------------------

Satuan [sunting | sunting sumber]

Satuan Teritorial [sunting | sunting sumber]

- Kodim 1301/Sangihe
- Kodim 1302/Minahasa
- Kodim 1303/Bolaang Mongondow
- Kodim 1309/Manado
- [Kodim 1310/Bitung](#)
- Kodim 1312/Tak

Satuan Samping

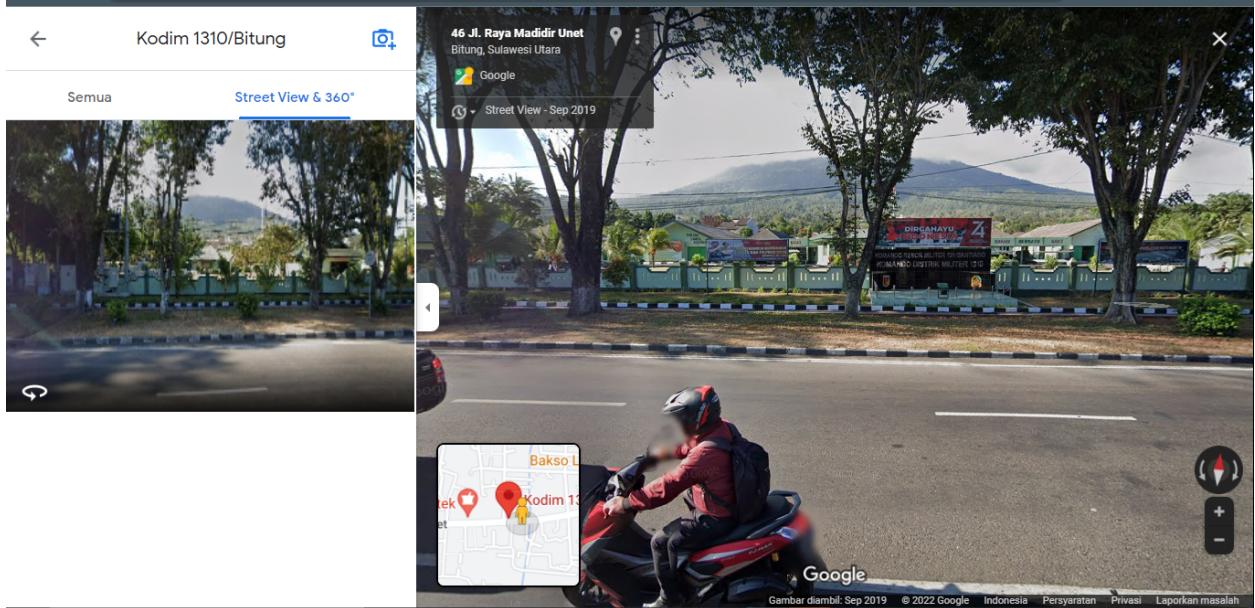
- Denpom XIII/1
- Den Zibang 1/XI
- Den Bekang XII
- Den Pal XIII/1 K
- Den Kesyah Re
- Hub Rem 131/S
- Ajenrem 131/Santiago
- Yonif 717 (Dalam Tahap Pembangunan)

Kodam Distrik Militer (Kodim)

1310/Bitung merupakan satuan kewilayahan yang berada dibawah Korem 131/Santiago. Kodim 1310/Bitung memiliki wilayah teritorial yang meliputi Kabupaten Minahasa Utara dan Kota Bitung. Markas Kodim 1310/Bitung berada di Jalan Raya Madidir Unet, Kota



Kami menemukan bahwa Kodim 1310/Bitung adalah nama tempat yang ada pada gambar pict_1.jpg



Kemudian kami mencari nama komandan Kodim Bitung tahun 2022 di Google dan menemukan nama Letkol Arm Yoki Efriandi, M.Han. pada laman beritamanado.com. Namun flag masih belum tepat hingga berkali-kali percobaan. Akhirnya kami menemukan bahwa ada nama yang lebih lengkap di laman Wikipedia tadi tinggal scroll ke bawah.

[ia.org/wiki/Komando_Distrik_Militer_1310](https://en.wikipedia.org/wiki/Komando_Distrik_Militer_1310)

Daftar Koramil:

1. Koramil 1310-01/Bitung
2. Koramil 1310-02/Lembbeh
3. Koramil 1310-03/Likupang
4. Koramil 1310-04/Dimembe
5. Koramil 1310-05/Kauditan
6. Koramil 1310-06/Airmadidi

Daftar Komandan:

1. Letkol Inf Denny Masengi (2011)★
2. Letkol Inf Hardo Toga P. Sitohang (2011—2013)
3. Letkol Inf Yarnedi Mulyadi S.I.P. (2013—2015)
4. Letkol Inf Rofiq Yusuf, S.Sos. (2015—2016)
5. Letkol Inf Deden Hendayana, S.E. (2016—2018)
6. Letkol Inf Kusnandar Hidayat, S.Sos.^[1] (2018—2021)
7. Letkol Inf Benny Lesmana, S.E., M.Han. (2021—2022)
8. [Letkol Arm Yoki Efriandi Gumay, M.Han. \(2022—Sekarang\)](#)

1. ^ "Kusnandar Pindah Ke Kodam, Kodim 1310 Bitung Beralih Keper". *INAnews*. 2020-08-04. Diakses tanggal 2022-02-04.

Kategori: Komando Distrik Militer

flag: FindITCTF{ LETKOLARMYOKIEFRIANDIGUMAY, M.HAN. }

Forensic

Dingin

Seorang pengembara sedang kedinginan. Tolong bantu dia!

The screenshot shows a challenge interface with the following details:

- Title:** Dingin
- Score:** 20
- Description:** Seorang pengembara sedang kedinginan. Tolong bantu dia!
- Download Button:** chall.txt
- Status:** 1/50 attempts
- Actions:** Flag, Submit

src/cara penyelesaian

Diberikan file chall.txt

Di Bagian bawah file terdapat banyak whitespace atau blank space , dan juga di title sudah diberi hint “dingin”. Jadi kemungkinan besar ini adalah whitespace stegano yang bisa di-solve menggunakan stegsnow.

```

kali㉿kali:~/fndit/seal/forensic/dingin
File Actions Edit View Help
(kali㉿kali:~/fndit/seal/forensic/dingin)
└─$ ls
chall.txt
(kali㉿kali:~/fndit/seal/forensic/dingin)
└─$ subl chall.txt
(kali㉿kali:~/fndit/seal/forensic/dingin)
└─$ 501URVRJQ1RGezRxDuhfS2VEaU5naW0tL9kNE5fTWVuQzRSVM9THzNMauVfSDROZZFUXzc3Nzc3Nzc3fQ==
(kali㉿kali:~/fndit/seal/forensic/dingin)
└─$ 

```

File Edit Selection Find View Goto Tools Project Preferences Help

chall.txt x

-/fndit/seal/forensic/dingin/chall.txt - Sublime Text (UNREGISTERED)

and in the descriptions of the weapon Snow-Tombed Starsilver, as well as the catalyst Frostbearer; and of the Princess' Box, Priest's Box, and Scribe's Box; as well as Frescos within the Mural Room where the Snow-Tombed Starsilver is found. Together they tell of a kingdom, a princess gifted with the power to foretell through visions and painting, and of a tragedy or fall of that Kingdom. The Fall is brought about by the Skyfrost Nail which falls and destroys the silver ley line tree on the mountain- this causes the change in climate from verdant to frostbitten. Artifacts found in Dragongspine tell the story of the Princess and people of that time, and the story of an outlander entrusted with the Starsilver weapon - when the outlander returns from his quest he finds the city dead and abandoned, and leaves the weapon in the secret chamber.

The decoded messages ("Record of Serial No.") found on broken Ruin Guards in the region may also be related, but also may relate to the fall of Khaenri'ah, from where Ruin Guards originate.

Line 1, Column 1 Tab Size: 4 Plain Text

Setelah menggunakan stegsnow kami mendapatkan sebuah string yang kami duga string tersebut berasal dari base 64, kemudian kami coba pada laman web cyberchef

CTF FindIt! 2022 x stegsnow | Kali Linux Tools x CTFTime.org / Hacktivity x Magic - CyberChef x +

Kali Linux | Kali Tools | Kali Docs | Kali Forums | Kali NetHunter | Exploit-DB | Google Hacking DB | OffSec | quipquip.com - quipquip... | quipquip - cryptoquip... | Caesar Cipher to Text... | Free Online Hex Editor...

Last build: 25 days ago

Operations

Search...

Favourites

- To Base64
- From Base64
- To Hex
- From Hex
- To Hexdump
- From Hexdump
- URL Decode
- Regular expression
- Entropy
- Fork
- Magic
- Data format
- Encryption / Encoding
- Public Key
- Arithmetic / Logic

Recipe

Magic

Depth 3

Crib (known plaintext string or regex)

Input

501URVRJQ1RGezRxDuhfS2VEaU5naW0tL9kNE5fTWVuQzRSVM9THzNMauVfSDROZZFUXzc3Nzc3Nzc3fQ==

Output

Recipe (click to load)	Result snippet	Properties
<code>From_Base64('A-Za-z0-9+=',true)</code>	<code>KMTETICTF{4QuH_KeDiNgin4N_d4N_MeNc4R1_S33LiE_H4NgaT_777777777777}</code>	Valid UTF8 Entropy: 4.38
<code>From_Base64('A-Za-z0-9+=\r\n',true)</code>	<code>KMTETICTF{4QuH_KeDiNgin4N_d4N_MeNc4R1_S33LiE_H4NgaT_777777777777}</code>	Valid UTF8 Entropy: 4.38
	<code>S01URVRJQ1RGezRxDuhfS2VEaU5naW0tL9kNE5fTWVuQzRSVM9THzNMauVfSDROZZFUXzc3Nzc3Nzc3fQ==</code>	Matching ops: From Base64 Valid UTF8 Entropy: 4.36

STEP BAKE! Auto Bake

Di sini terlihat jelas dengan resep 'magic' langsung didapatkan flagnya

Flag: **KMTETICTF{4QuH_KeDiNgin4N_d4N_MeNc4R1_S33LiE_H4NgaT_777777777777}**

Citra

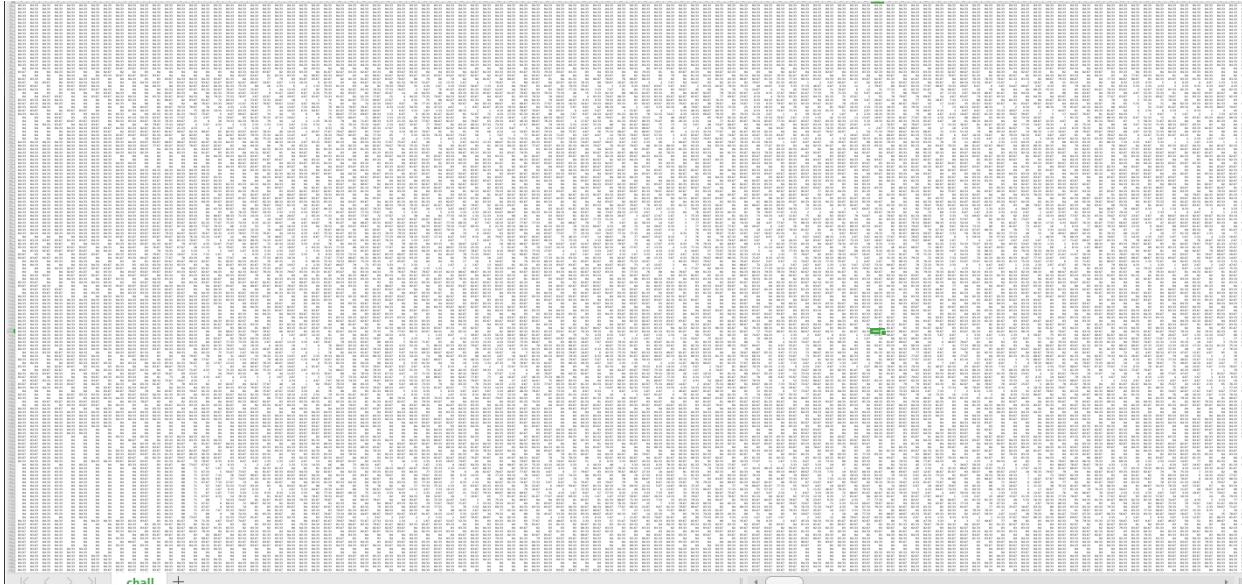
Eh, ada Kamisato Ayaka? Mana? Bawa 16 angka? Maksudnya?



src/cara penyelesaian

Diberikan file chall.txt

Filenya berisi bilangan dalam format comma separated yang nilainya sekitar 0-100. Kemudian kami mencoba membukanya menggunakan wps, setelah di zoom out hasilnya seperti sebuah gambar.



Kami curiga kalau ini merupakan gambar yang diubah ke teks. Kemudian kami cek banyak kolomnya ternyata 1920, wow seperti familiar ya, tentu saja ini adalah width dari gambar hd. Sehingga kemungkinan besar tiap nilai merepresentasikan sebuah pixel dari gambar tersebut. Tapi bukannya pixel itu terdiri 3 value yaitu RGB? Oh tentu tidak ferguso, gambar hitam putih hanya perlu 1 value yaitu intensitas.

solve.py

```
import enum
from PIL import Image

img = Image.new('L', (1920, 1112))
with open('./forensic/citra/chall.csv', 'r') as f:
    lines = f.readlines()
    for i,line in enumerate(lines):
        for j,value in enumerate(line.split(',')):
            img.putpixel((j,i), int(float(value)))

img.show()
```

Program tersebut membaca setiap baris dari file, kemudian tiap value yang ada di baris digunakan sebagai intensitas pixel. Setelah itu ditampilkan, dan muncullah waifu sang problem setter.



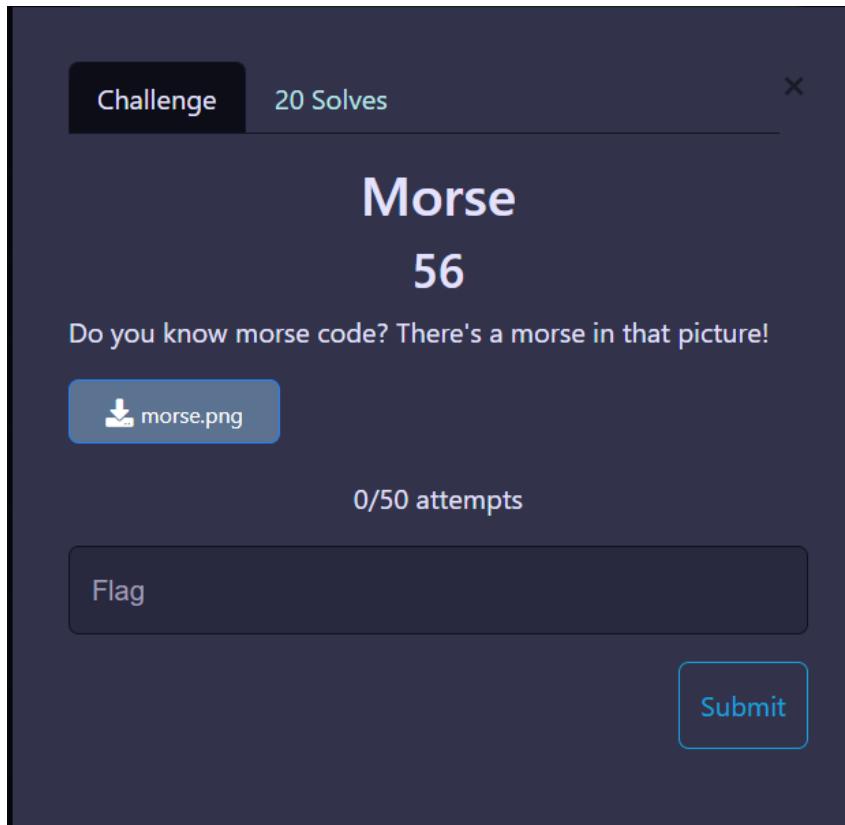
Dapat dilihat terdapat string di pojok gambar, dari hint soal bisa diketahui kalau ini adalah string hex. Cus kita ubah ke ascii.

```
# didapat dari gambar
string =
'''46696E6449544354467B59345F4E64346B5F5461755F4B306B5F4E64344B5F54346E794
15F4D34737433725F393939393939393939397D
'''
print(bytes.fromhex(string))
```

flag: FindITCTF{Y4_Nd4k_Tau_K0k_Nd4K_T4nyA_M4st3r_99999999999}

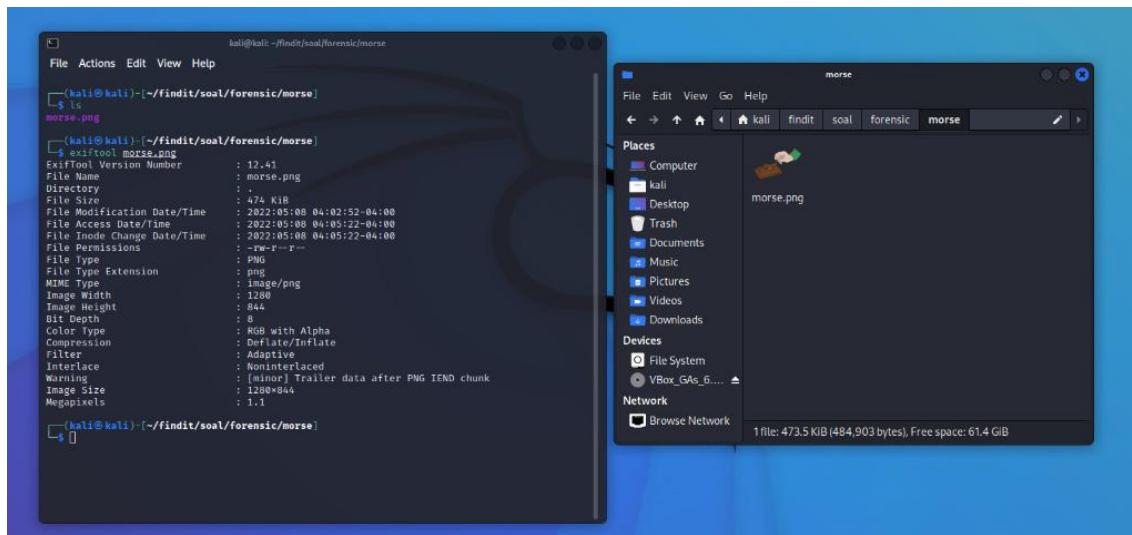
Morse

Do you know morse code? There's a morse in that picture!



Src/cara penyelesaian

Kami di berikan sebuah gambar berformat .png, kami mencoba untuk mencari informasi melalui 'exiftool' namun tidak ada yang bisa di ambil.



Kemudian kami mencoba menggunakan laman web untuk mencari informasi lebih detail pada foto tersebut, kemudian didapatkan sebuah informasi bahwa adanya file berformat 'Zip archive' yang tertera pada 'binwalk'.

Aperi'Solve

File Upload

Recent

- Home
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos
- VBox_GAs_6.1.34
- Other Locations

Name	Size	Type	Modified
morse.png	484.9 kB	Image	Yesterday

online platform which performs layer analysis on
orm also uses zsteg, steghide, outguess, exiftool,
st and strings for deeper steganography analysis.
ports the following images format: .png, .jpg, .gif,
.jpe, .tiff...

Select a file or drag here

SELECT A FILE

SUBMIT

Image Files

Cancel Open

files (-extract) ?

Test all options of zsteg (-all) ?

I've got a password!

ENABLE

ENABLE

DISABLE

FileModifyDate 2022-05-08 02:19:11+00:00

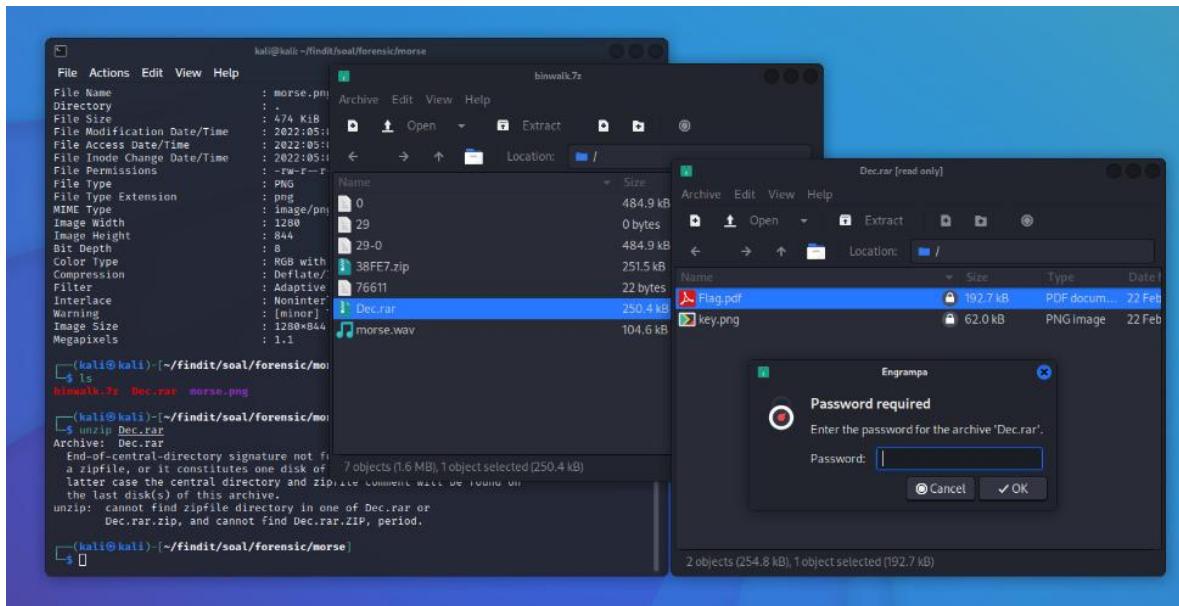
Binwalk

	DECIMAL	HEXADECIMAL	DESCRIPTION
	0	0x0	PNG image, 1280 x 844, 8-bit/color RGBA, non-interlaced
	41	0x29	Zlib compressed data, best compression
	233447	0x38FE7	Zip archive data, at least v2.0 to extract, compressed size: 250396, uncompressed size: 250356, name: Dec.rar
	48380	0x76228	Zip archive data, at least v2.0 to extract, compressed size: 782, uncompressed size: 104604, name: morse.wav
	484881	0x76611	End of Zip archive, footer length: 22
WARNING:	Extractor.execute failed to run external extractor 'jar xvf "%e": [Errno 2] No such file or directory: 'jar', 'jar xv... "%e" might not be installed correctly'		

DOWNLOAD FILES

Extractor

Kami mencoba untuk download file tersebut hingga didapatkan sebuah informasi terkait flag yang ada pada zip tersebut.



Saat check file zipnya ternyata berpassword. Kemudian ada juga file wav, namun tidak bisa dibuka, dan kami mencoba memperbaiki header file tersebut akhirnya bisa dibuka. Dengan menggunakan tools morse code online,

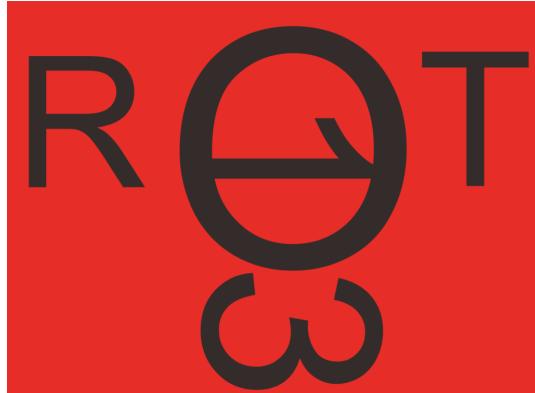
The screenshot shows a web-based application for decoding Morse code. At the top, there is a text input field containing the message 'UR UR UR S4ND1KU K3R3N'. Below the text input is a green button labeled 'Clear message'. To the right of the text input is a small portrait of a woman. Below the text input are several configuration controls: 'WPM' set to 20, 'Farnsworth WPM' set to 19, 'Frequency (Hz)' set to 563, 'Minimum volume' set to -60, 'Maximum volume' set to -30, and 'Volume threshold' set to 200. There are two checkboxes labeled 'Manual' next to these controls. Below these controls is a large area for visualizing the Morse code. It features a red vertical line representing the audio signal amplitude over time. Below this is a waveform visualization showing the individual dots and dashes of the Morse code. At the bottom, there is a binary representation of the Morse code as a series of black and white squares.

Didapat string “UR UR UR S4ND1KU K3R3N”, kemudian kami menggunakan string tersebut untuk mengextract file zip tadi. Dan mendapat 2 file yaitu flag.pdf berpassword dan key.png.

Setelah kami check dengan pngcheck ternyata file key.png adalah gambar yg korup sehingga kami coba memperbaikinya dengan PCRT.

```
[master]$ python2 ~/ctf/tools/PCRT/PCRT.py -i key.png -o out.png
```

Setelah di restore menjadi gambar berikut:

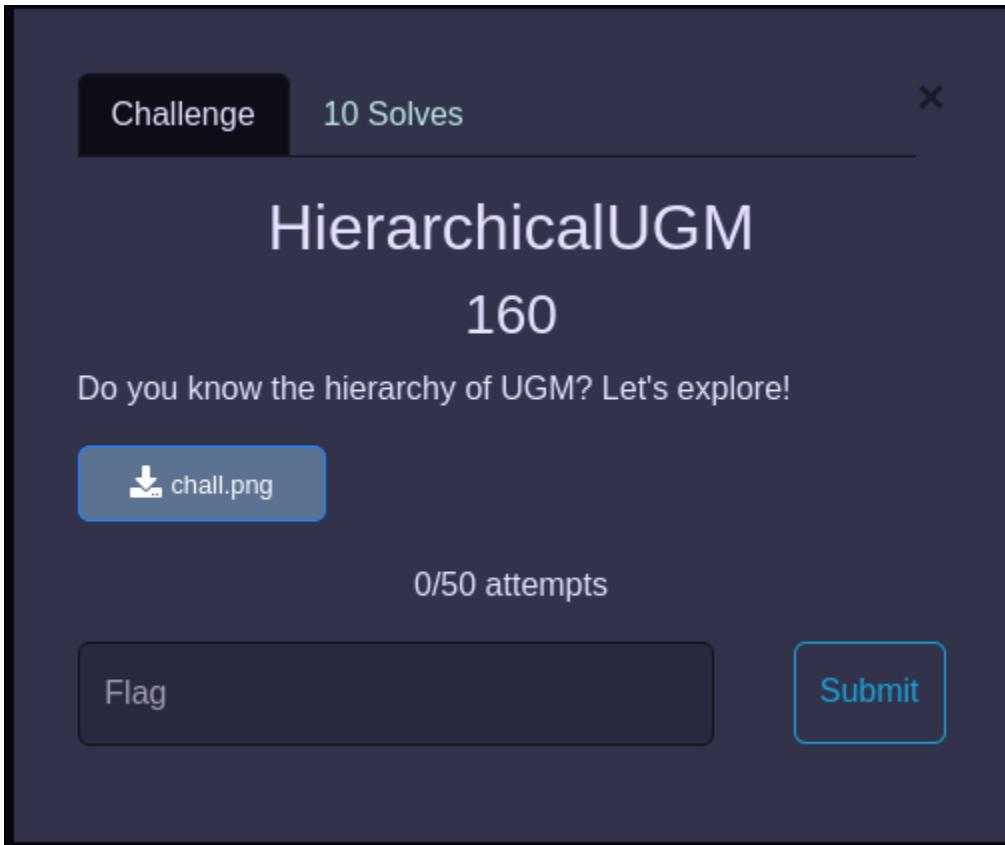


Ternyata setelah flag IEND juga masih ada string “oHx4_c1aGh”. Sangat jelas ya, kami melakukan dekripsi menggunakan algoritma rot13 ke string “oHx4_c1aGh” menggunakan tool online dan didapat string “bUk4_p1nTu”. Selanjutnya memakai string tersebut sebagai password file flag.pdf

[UNSUBMITTED]: FindITCTF{ D3cod3r_J0os5_T3n4n }

HierarchicalUGM

Do you know the hierarchy of UGM? Let's explore!



Src/cara penyelesaian

Diberikan file chall.png,

Kami ekstrak menggunakan binwalk dan mendapatkan file pict1.png.

Kami ekstrak file pict1.png menggunakan binwalk dan mendapat file pict2.png

Kami ekstrak file pict2.png menggunakan binwalk dan mendapat file **38E6** yang merupakan file png yang korup.

Kemudian kami perbaiki menggunakan hex editor dan mendapat gambar berikut:



flag:

Forensic Digital QiQi

Sudah cukup jelas pada notes. Something is "packed" by HuTao, tho....

Challenge 20 Solves [X](#)

Forensic Digital QiQi

42

Sudah cukup jelas pada notes. Something is "packed" by Hu Tao, tho.....

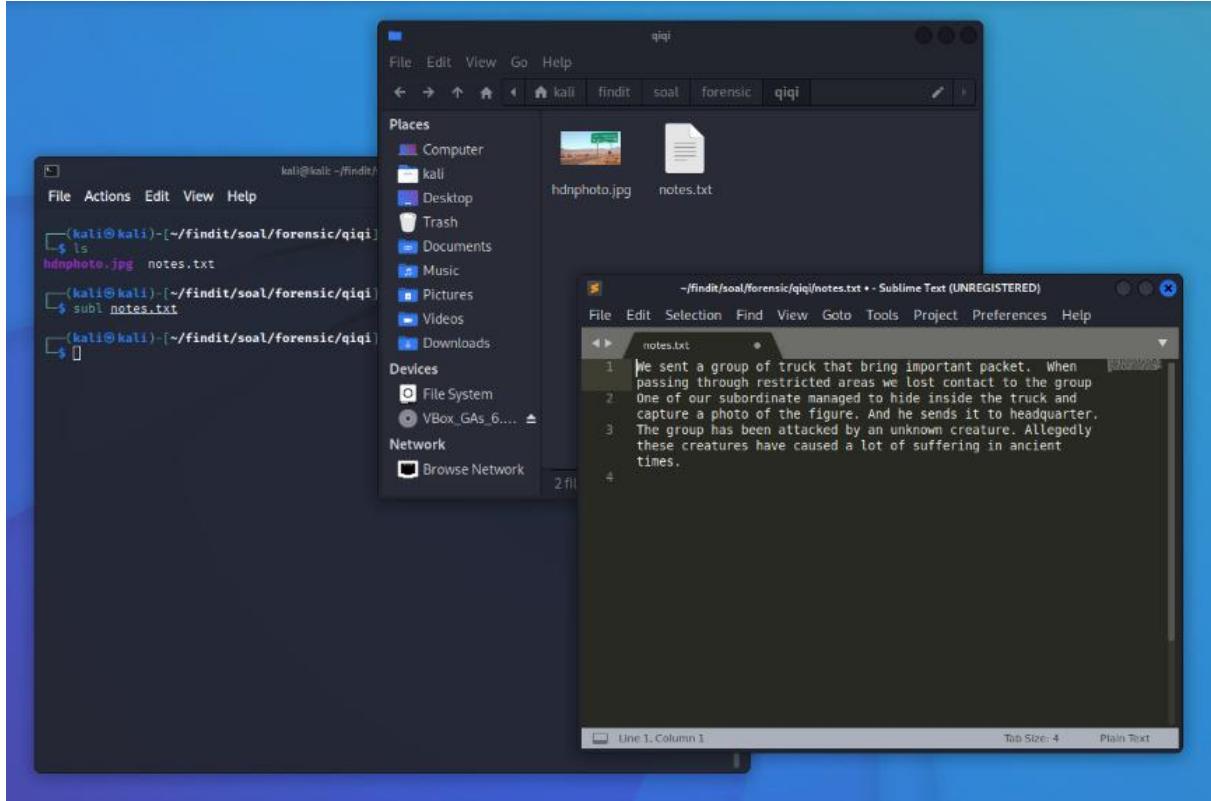
[hdnphoto.jpg](#) [notes.txt](#)

0/50 attempts

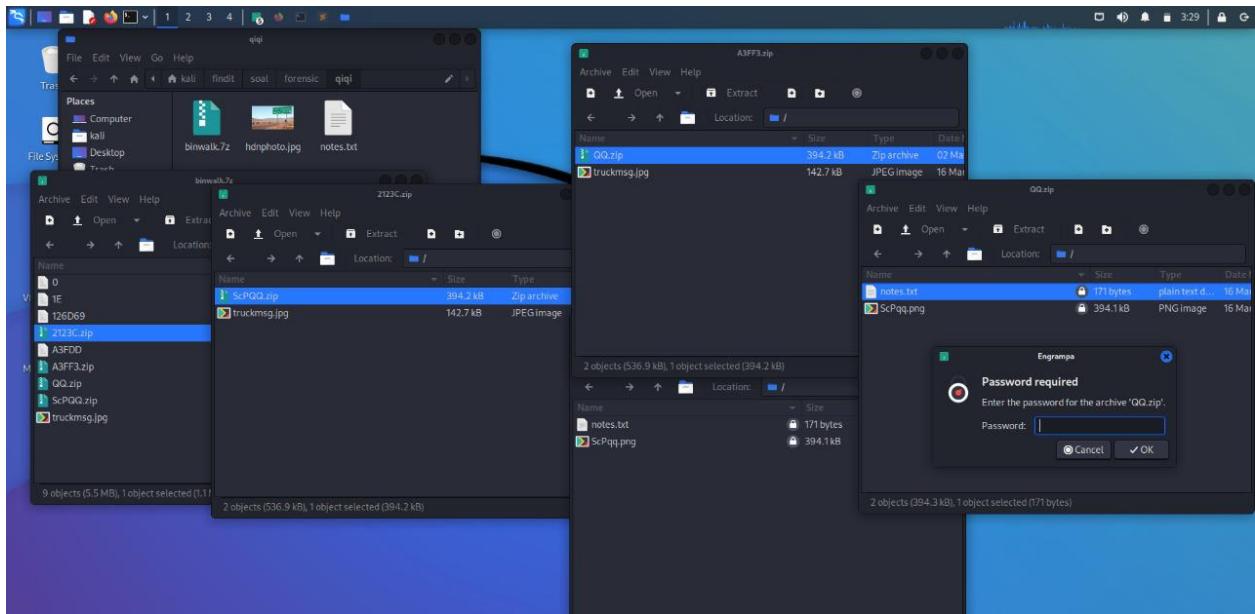
Flag Submit

Src/cara penyelesaian

Diberikan sebuah file berformat ‘.jpg & txt’. Yang kami lakukan membuka notes.txt yang merupakan hint dari soal yang diberikan, namun kami tidak menemukan informasi lebih lanjut



Kami melakukan analisis terhadap foto tersebut menggunakan laman web “aperisolve”, dan pada bagian ‘binwalk’ terdapat banyak hal yang tersembunyi sehingga kami mendownload file tersebut, dan kami mendapatkan sebuah file zip.



Pada file zip tersebut terdapat zip di dalamnya yang menuju ke zip qiqi dan terdapat 'note.txt' yang kami asumsikan flag berada di dalam sana, namun membutuhkan password untuk membukanya dan kami menemukan sebuah password common dari sebuah gambar yang ada pada zip

Namun ketika password tersebut kami masukan tidak terjadi apa - apa atau gagal membuka file tersebut :D

Setelah itu kami juga mencoba bruteforce file QQ.zip menggunakan wordlist rockyou, tetapi hasilnya nihil, tidak ada password yang cocok.

Kami putus asa, eh maksudnya tidak putus asa dan mencoba untuk melakukan hal yang sama ke file ScPQQ.zip. Yaitu brute force menggunakan john, dan boom terjadi ledakan.

```
[lurifos@Kulen Progo] 🐻 [/tmp/qiqi/_hdnphoto.jpg.extracted]
~/ctf/tools/JohnTheRipper/run/john hashscp --wordlist=/home/lurifos/ctf/rockyou.txt
```

```
0g 0:00:09:37 68.16% (ETA: 21:45:52) 0g/s 16911p/s 33822c/s 33822C/s baustralia25..barkerana
0g 0:00:09:38 68.28% (ETA: 21:45:52) 0g/s 16911p/s 33822c/s 33822C/s barkera2..bamler
0g 0:00:09:39 68.40% (ETA: 21:45:52) 0g/s 16911p/s 33823c/s 33823C/s bamlegsi..bahobahomo
Explosion      (ScPQQ.zip/ScPqq.png)
Explosion      (ScPQQ.zip/notes.txt)
```

Setelah diekstrak, terdapat file notes dan ScPqq.PNG, kemudian kami melakukan analisis di file png tersebut dan menemukan flag di meta foto.

Title	:	FindITCTF{QiQi_Rule5_7he_Wor1d}
Author	:	muhammadrifatba
Creator Tool	:	Canva
Image Size	:	500x500
Megapixels	:	0.250

[UNSUBMITTED]: FindITCTF{QiQi_Rule5_7he_Wor1d}

<481817566689890542614046840158500529277107859321>