

Write Up Gemastik 13

Persatuan Intel Negara Gaijin



Rio Darmawan

Ahmad Fauzzan Maghribi

Widi Afandi

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

REVERSE ENGINEERING

Mr. Simple

Challenge

9 Solves

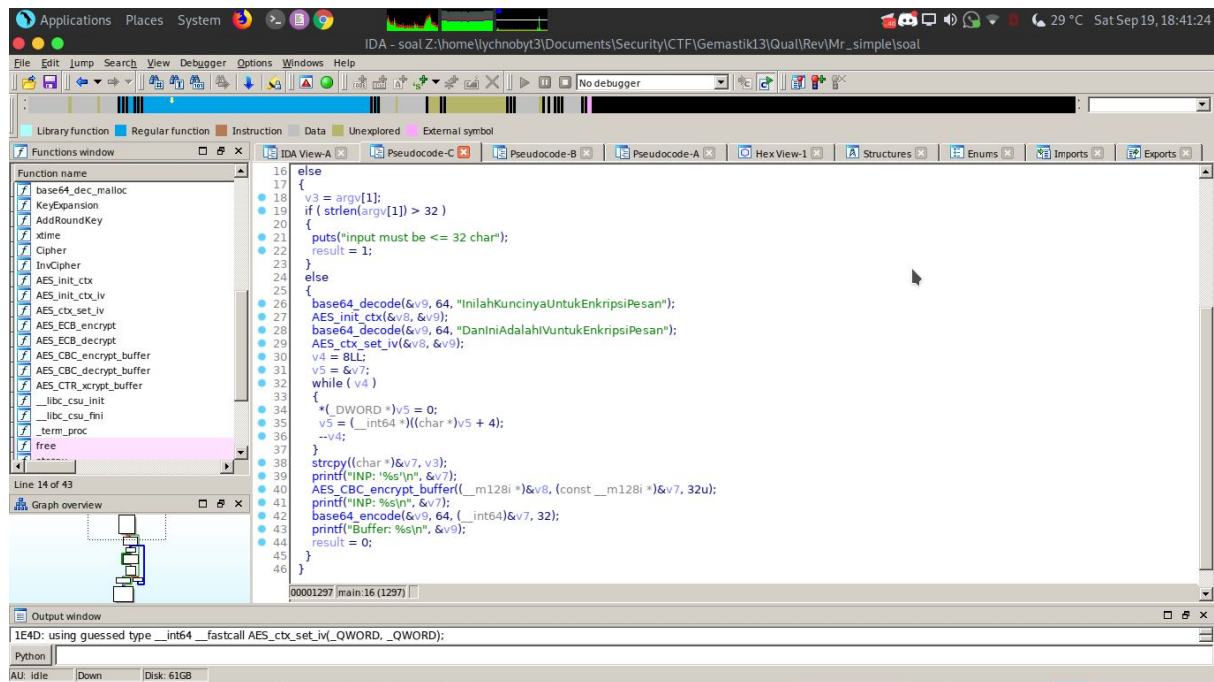


Mr. Simple 200

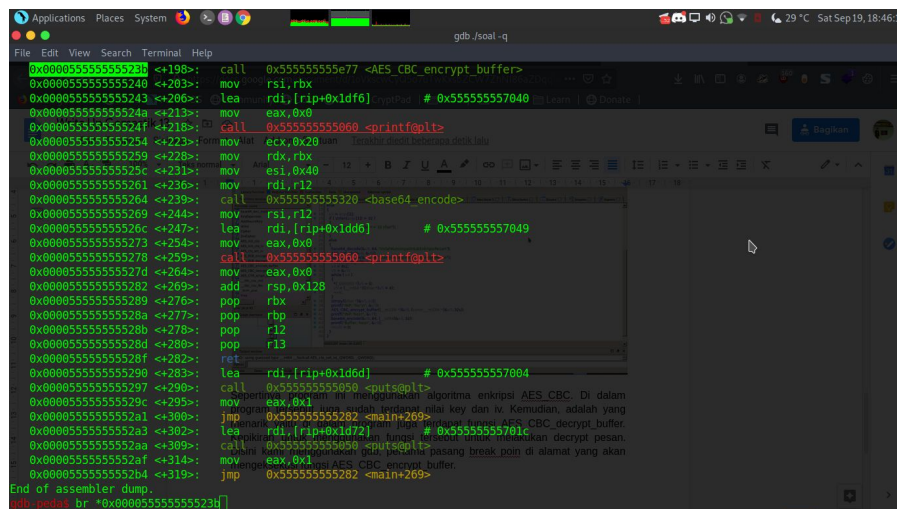
Di bawah sebuah piramida di Mesir, Oingo dan Boingo menemukan sebuah program beserta keluarannya. Ketika dijalankan program tersebut menghasilkan keluaran terenkripsi. Oingo dan Boingo tidak mengetahui apa algoritma yang digunakan program tersebut, sampai tiba-tiba kutukan pharaoh menyerang. Oingo dan Boingo harus memasukkan plaintext yang sesuai dengan keluaran yang dihasilkan oleh program tersebut. Yang diketahui Oingo dan Boingo hanyalah keluaran program yaitu
zOHmai4ZLj2j50vYcWZhGdftB9lCmGINOiKtjKID+Cc=
Bantu Oingo dan Boingo menyelamatkan diri dengan mencari plaintext dari keluaran tersebut.

Unlock Hint for 45 points

Diberikan sebuah file ELF-64bit not-stripped, hasil decompilennya.



Sepertinya program ini menggunakan algoritma enkripsi AES_CBC. Di dalam program tersebut juga sudah terdapat nilai key dan iv. Kemudian, adalah yang menarik yaitu di dalam program juga terdapat fungsi AES_CBC_decrypt_buffer. Kepikiran untuk menggunakan fungsi tersebut untuk melakukan decrypt pesan. Disini kami menggunakan gdb, pertama pasang break poin di alamat yang akan mengeksekusi fungsi AES_CBC_encrypt_buffer.

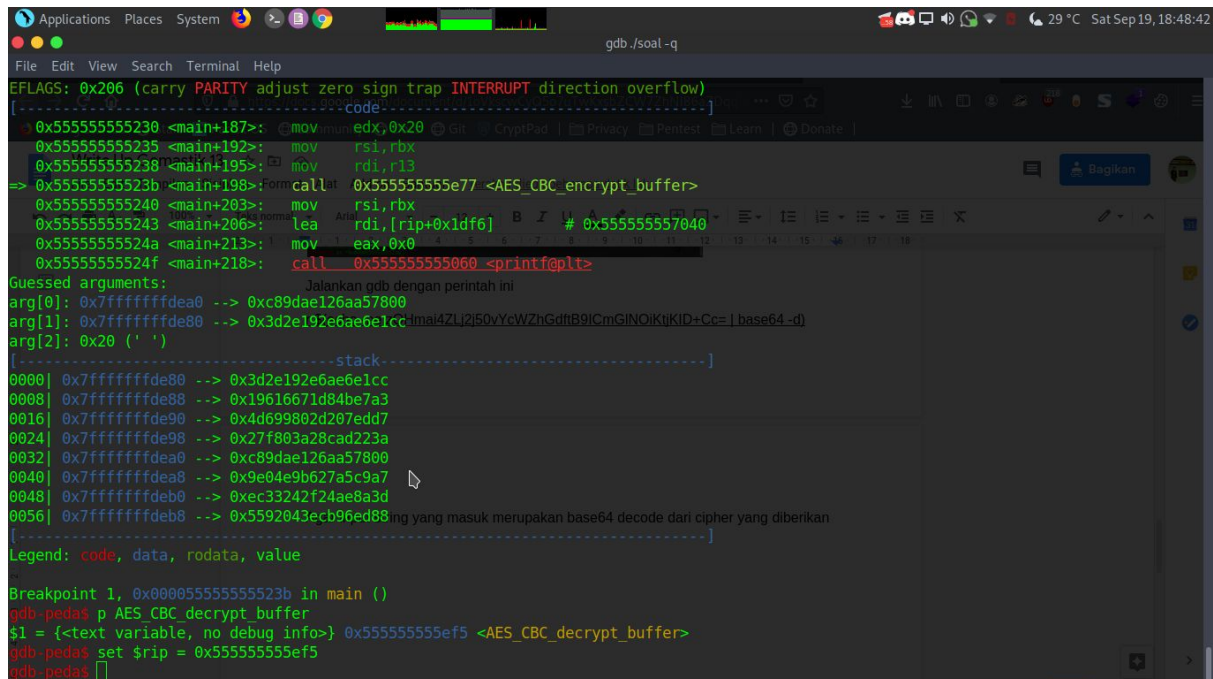


Jalankan gdb dengan perintah ini

`r $(echo -ne zOHmai4ZLj2j50vYcWZhGdftB9ICmGINOiKtjKID+Cc= | base64 -d)`

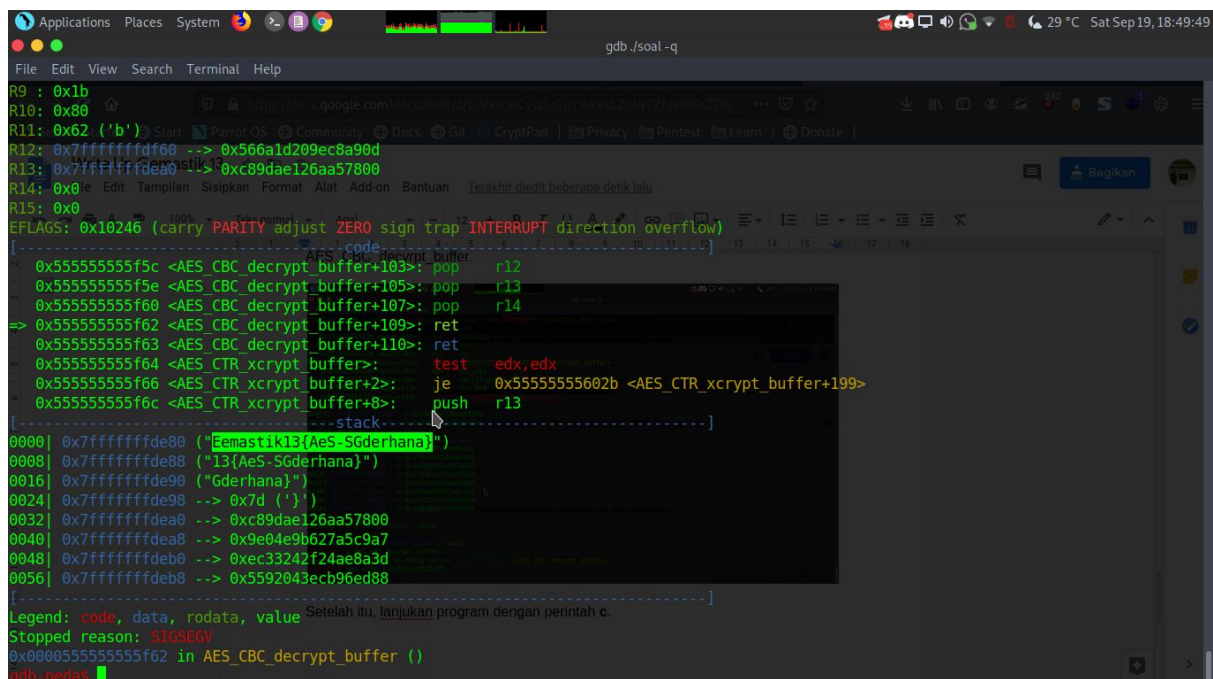
Agar input string yang masuk merupakan base64 decode dari cipher yang diberikan.

Kemudian saat breakpoint, set register RIP ke alamat fungsi AES_CBC_decrypt_buffer.



```
gdb ./soal -q
EFLAGS: 0x206 (carry PARITY adjust zero sign trap INTERRUPT direction overflow)
[Code window showing assembly instructions]
0x5555555230 <main+187>: mov     edx,0x20
0x5555555235 <main+192>: mov     rsi,rbx
0x5555555238 <main+195>: mov     rdi,r13
=> 0x555555523b <main+198>: call    0x55555555e77 <AES_CBC_encrypt_buffer>
0x5555555240 <main+203>: mov     rsi,rbx
0x5555555243 <main+206>: lea     rdi,[rip+0x1df6] # 0x555555557040
0x555555524a <main+213>: mov     eax,0x0
0x555555524f <main+218>: call    0x555555555060 <printf@plt>
Guessed arguments:
arg[0]: 0x7fffffffdea0 --> 0xc89dae126aa57800
arg[1]: 0x7fffffffde80 --> 0x3d2e192e6ae6e1cc
arg[2]: 0x20 (' ')
[Stack window showing memory addresses and values]
0000| 0x7fffffffdea0 --> 0x3d2e192e6ae6e1cc
0008| 0x7fffffffde80 --> 0x19616671d84be7a3
0016| 0x7fffffffde90 --> 0x4d699802d207edd7
0024| 0x7fffffffde98 --> 0x27f803a28cad223a
0032| 0x7fffffffdea0 --> 0xc89dae126aa57800
0040| 0x7fffffffdea8 --> 0x9e04e9b627a5c9a7
0048| 0x7fffffffdeb0 --> 0xec33242f24ae8a3d
0056| 0x7fffffffdeb8 --> 0x5592043ecb96ed88
Legend: code, data, rodata, value
Breakpoint 1, 0x0000555555523b in main ()
gdb-peda$ p AES_CBC_decrypt_buffer
$1 = {<text variable, no debug info>} 0x55555555ef5 <AES_CBC_decrypt_buffer>
gdb-peda$ set $rip = 0x55555555ef5
gdb-peda$
```

Setelah itu, lanjutkan program dengan perintah c. Akan muncul flag yang sepertinya belum benar, namun bisa diperbaiki manual



```
gdb ./soal -q
R9 : 0x1b
R10: 0x80
R11: 0x62 ('b')
R12: 0x7fffffffdf60 --> 0x566a1d209ec8a90d
R13: 0x7fffffffde80 --> 0xc89dae126aa57800
R14: 0x0
R15: 0x0
EFLAGS: 0x10246 (carry PARITY adjust ZERO sign trap INTERRUPT direction overflow)
[Code window showing assembly instructions]
0x55555555f5c <AES_CBC_decrypt_buffer+103>: pop     r12
0x55555555f5e <AES_CBC_decrypt_buffer+105>: pop     r13
0x55555555f60 <AES_CBC_decrypt_buffer+107>: pop     r14
=> 0x55555555f62 <AES_CBC_decrypt_buffer+109>: ret
0x55555555f63 <AES_CBC_decrypt_buffer+110>: ret
0x55555555f64 <AES_CTR_xcrypt_buffer>: test    edx,edx
0x55555555f66 <AES_CTR_xcrypt_buffer+2>: je      0x55555555602b <AES_CTR_xcrypt_buffer+199>
0x55555555f6c <AES_CTR_xcrypt_buffer+8>: push    r13
[Stack window showing memory addresses and values]
0000| 0x7fffffffde80 ("gemastik13{AeS-Sgderhana}")
0008| 0x7fffffffde88 ("13{AeS-Sgderhana}")
0016| 0x7fffffffde90 ("Gderhana")
0024| 0x7fffffffde98 --> 0x7d ('.')
0032| 0x7fffffffdea0 --> 0xc89dae126aa57800
0040| 0x7fffffffdea8 --> 0x9e04e9b627a5c9a7
0048| 0x7fffffffdeb0 --> 0xec33242f24ae8a3d
0056| 0x7fffffffdeb8 --> 0x5592043ecb96ed88
Legend: code, data, rodata, value
Stopped reason: SIGSEGV
0x000055555555f62 in AES_CBC_decrypt_buffer ()
gdb-peda$
```

Flag : gemastik13{AeS-Sederhana}

BINARY EXPLOIT

Repeat After Me

Challenge

33 Solves

×

Repeat After Me

200

He likes you..... he will repeat what you said

180.250.135.6:9090

Flag

Submit

Diberikan sebuah service remote, yang hanya menampilkan apa yang kita inputkan. Namun ketika mencoba memasukkan input yang sangat panjang, flag langsung muncul.

```
lychnoby3@parrot ~/Documents/Security/CTF/Gemastik13/Qual/Rev/Mr_simple
$ nc 180.250.135.6 9090
saya akan mengulang perkataan ada. masukkan karakter!!!!!! AAAAAA
anda memasukkan : AAAAAA
lychnoby3@parrot ~/Documents/Security/CTF/Gemastik13/Qual/Rev/Mr_simple
$ nc 180.250.135.6 9090
saya akan mengulang perkataan ada. masukkan karakter!!!!!! AAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA
anda memasukkan : AAAAAAAAAAAAgemastik13{st4ck_c4n4ry_m4k3s_m3_dyzyy}
lychnoby3@parrot ~/Documents/Security/CTF/Gemastik13/Qual/Rev/Mr_simple
```

Flag : gemastik13{st4ck_c4n4ry_m4k3s_m3_dyzyy}

STEGANOGRAPHY

AH↓HA↑HA↑HA↑HA↑

Challenge


5 Solves

×

AH↓HA↑HA↑HA↑HA↑

200

Unlock Hint for 100 points

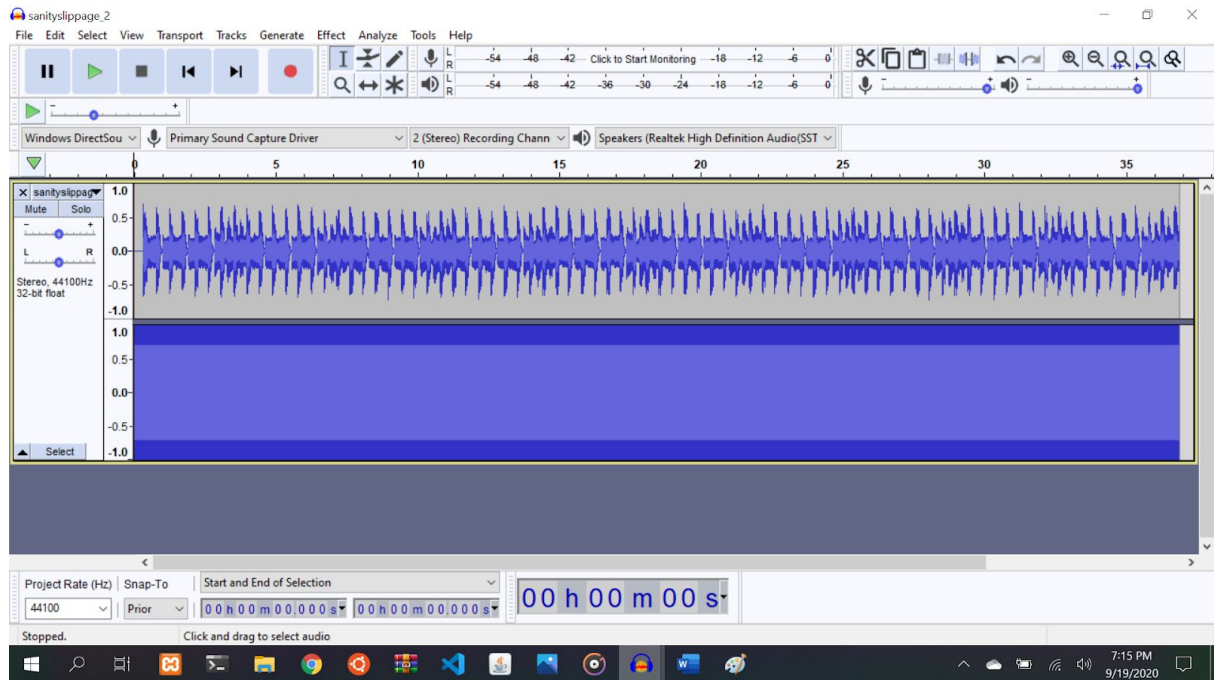
 sanityslippag...

Flag

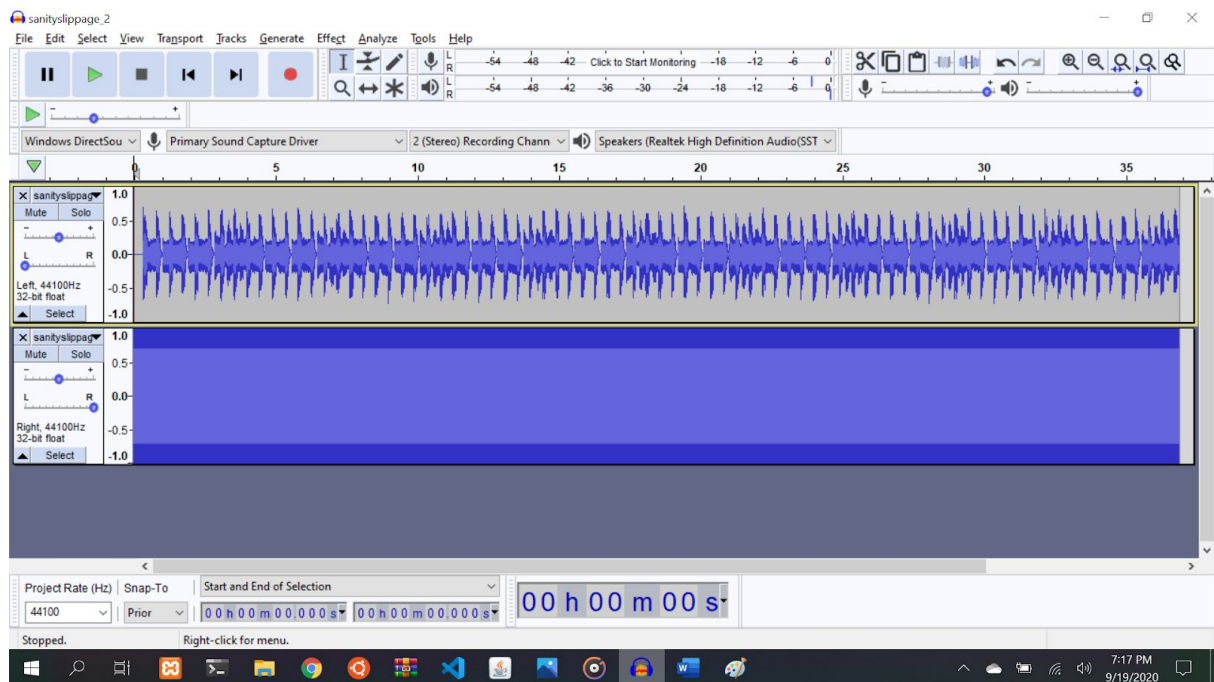
Submit

diberikan sebuah file berekstensi .wav , setelah saya play saya sudah mengasumsikan bahwa ini adalah sinyal radio transceiver sstv sendiri adalah singkatan dari Slow Scan Television.
namun ketika saya analisa , terdapat audio musik yang mengganggu sinyal tersebut. jadi sinyal sstv tidak bisa terscan dengan jelas.

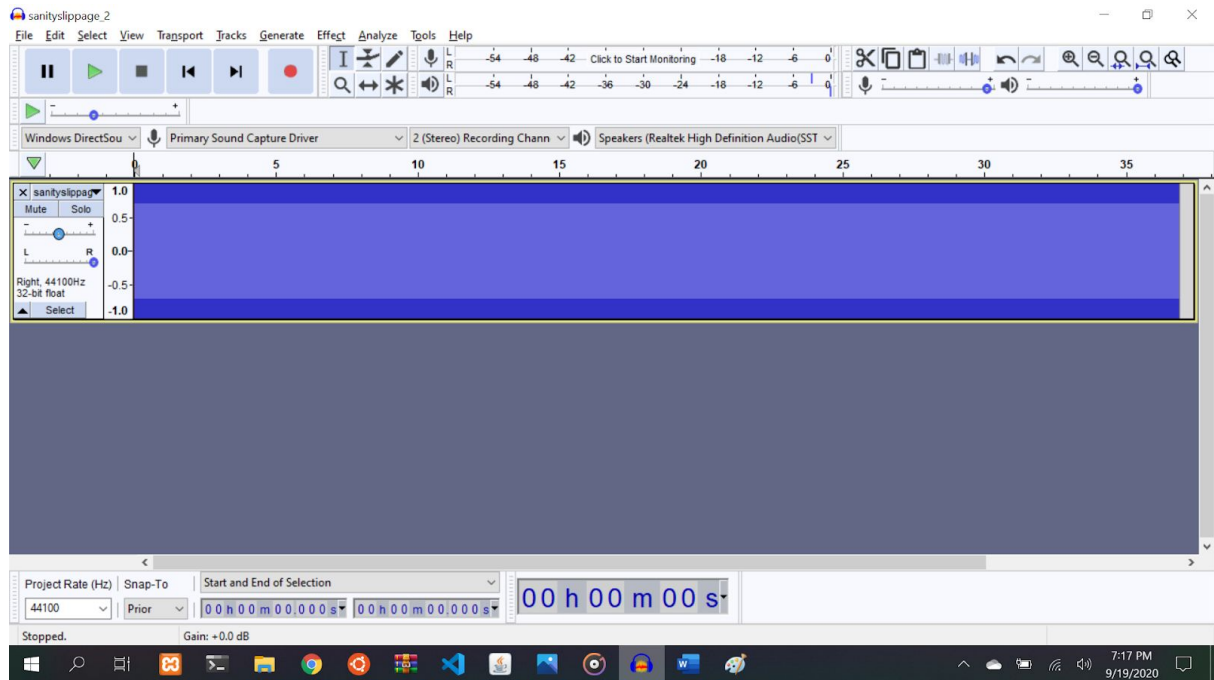
langsung saya analisa menggunakan “audacity” untuk menghapus/menghilangkan audio musiknya



saya split audio tersebut



lalu saya hapus audio musiknya



dan saya langsung play sinyal SSTV nya dan saya scan menggunakan tool “Robot36 SSTV Image Decoder” yang ada di android saya.



dan muncul flagnya

flag : gemastik13{yougotme_peko}

STEGANOGRAPHY

Missing Something ?

Challenge

28 Solves

×

Missing Something ?

100

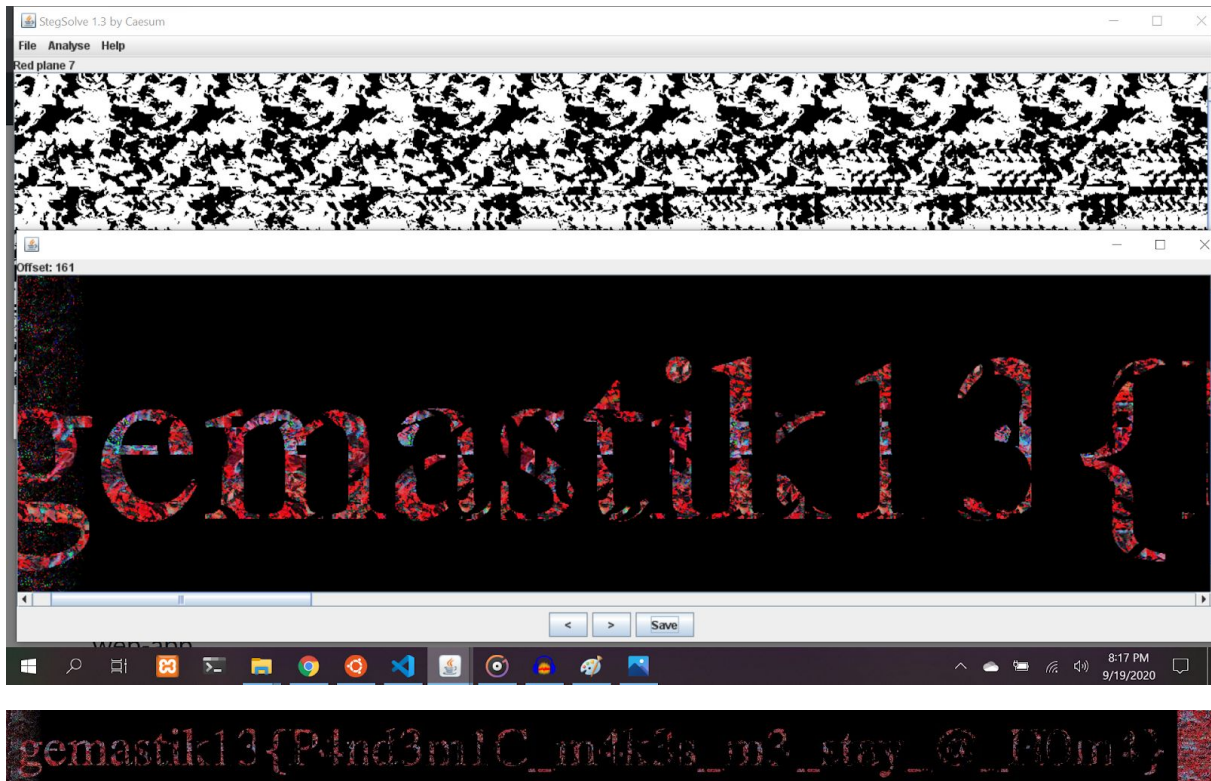
anya mendapatkan sebuah gambar, namun anya tidak tahu maksud dari gambar ini. dapatkah anda membantu anya ?

 anya-missing...

Flag

Submit

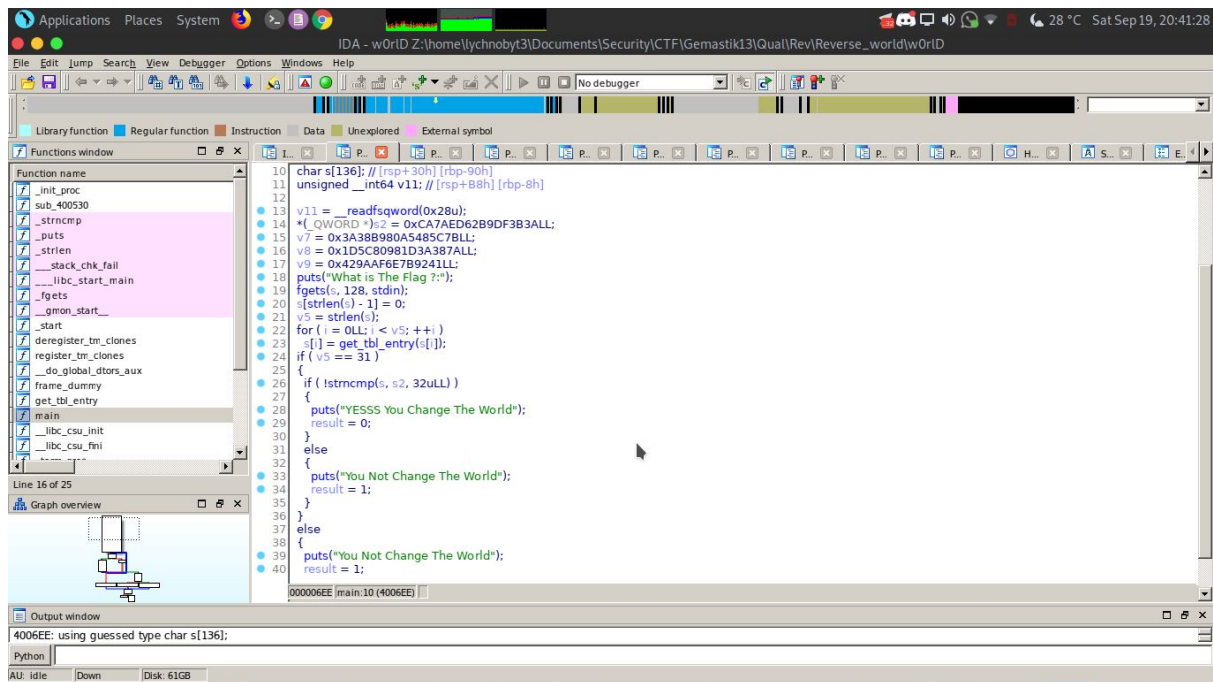
diberikan sebuah gambar , setelah saya analisa menggunakan binwalk, command “strings” dll. saya tidak menemukan apa2. selanjutnya saya analisa menggunakan aplikasi steg solver dan menggunakan fitur stereogram solver dan inilah hasil dari analisa menggunakan tools tersebut



flag:gemastik13{P4nd3m1C_m4k3s_m3_stay_@_H0m4}

REVERSE WORLD

Diberikan sebuah file ELF 64bit, dimana program meminta inputan flag. Tapi inputan tersebut ada di-replace dengan sebuah fungsi `get_tbl_entry`, baru kemudian di cek apakah sama dengan nilai di variabel `s2`.



Isi dari fungsi `get_tbl_entry` sebagai berikut, ada dua array yang digunakan disini yaitu `trans_tbl` dan `byte_601081`. Tapi setelah dicek dua array tersebut hanya berbeda 1 byte di awal.

```

1 int64 fastcall get_tbl_entry(char a1)
2 {
3     unsigned __int64 i; // [rsp+Ch] [rbp-8h]
4
5     for ( i = 0LL; i <= 254; ++i )
6     {
7         if ( trans_tbl[2 * i] == a1 )
8             return (unsigned __int8)byte_601081[2 * i];
9     }
10    return 0LL;
11 }

```

Selanjutnya, saya buat script untuk membalikan nilai dari `s2` terhadap array `byte`, untuk mendapatkan flag. Berikut script yang kami gunakan.

Solver.py

```

magic =
[0x3a,0x3b,0xdf,0xb9,0x62,0xed,0x7a,0xca,0x7b,0x5c,0x48,0xa5,
0x80,0xb9,0x38,0x3a,0x7a,0x38,0x3a,0x1d,0x98,0x80,0x5c,0x1d,

```

0x41,0x92,0x7b,0x6e,0xaf,0x9a,0x42,0x00]

```
trans_tbl = [0x1,
0xd1,0x02,0xc4,0x03,0xc9,0x04,0x13,0x05,0x05,0x06,0x81,0x07,
0x84,0x08,0x34,0x09,0x17,0x0a,0xe2,0x0b,0xc5,0x0c,0x5e,0x0d,
0xe6,0x0e,0x4c,0x0f,0x51,0x10,0x26,0x11,0x12,0x12,0x68,0x13,
0x07,0x14,0x7d,0x15,0x8e,0x16,0x69,0x17,0xce,0x18,0xd3,0x19,
0x37,0x1a,0xb1,0x1b,0x0c,0x1c,0x11,0x1d,0x53,0x1e,0x9f,0x1f,0
x91,0x20,0x09,0x21,0x99,0x22,0xaa,0x23,0xb7,0x24,0x06,0x25,
0xf4,0x26,0xcc,0x27,0xa9,0x28,0xfc,0x29,0x6a,0x2a,0x08,0x2b,0
x50,0x2c,0x1c,0x2d,0x45,0x2e,0xe8,0x2f,0xa8,0x30,0xcf,0x31,0x
7b,0x32,0x6e,0x33,0x5c,0x34,0xb0,0x35,0xc1,0x36,0xde,0x37,0x
7c,0x38,0xa0,0x39,0x66,0x3a,0x40,0x3b,0x19,0x3c,0xad,0x3d,0x
43,0x3e,0xe9,0x3f,0x77,0x40,0xb8,0x41,0x2b,0x42,0x23,0x43,0x
a5,0x44,0x4a,0x45,0x7f,0x46,0x1b,0x47,0xd5,0x48,0x1e,0x49,0x
ba,0x4a,0x5d,0x4b,0x33,0x4c,0xa1,0x4d,0xec,0x4e,0x60,0x4f,0x
8d,0x50,0x04,0x51,0xbf,0x52,0xea,0x53,0x1a,0x54,0x98,0x55,0x
2d,0x56,0xf2,0x57,0x41,0x58,0xef,0x59,0x15,0x5a,0x59,0x5b,0xf
6,0x5c,0x6f,0x5d,0x2c,0x5e,0x95,0x5f,0x1d,0x60,0x5b,0x61,0xb9
,0x62,0xf8,0x63,0x39,0x64,0x9a,0x65,0x3b,0x66,0x3c,0x67,0x3a,
0x68,0x80,0x69,0x7a,0x6a,0xd6,0x6b,0xca,0x6c,0xaf,0x6d,0xdf,0
x6e,0x38,0x6f,0x92,0x70,0x96,0x71,0x4f,0x72,0x0a,0x73,0x62,0x
74,0xed,0x75,0xfe,0x76,0x3d,0x77,0x71,0x78,0x9e,0x79,0x3f,0x
7a,0xc8,0x7b,0x48,0x7c,0x2a,0x7d,0x42,0x7e,0xe1,0x7f,0x97,0x
80,0xdc,0x81,0xbb,0x82,0xbd,0x83,0xd9,0x84,0xc2,0x85,0x54,0x
86,0x44,0x87,0x87,0x88,0x94,0x89,0x8c,0x8a,0x18,0x8b,0x82,0x
8c,0x35,0x8d,0xa2,0x8e,0xf5,0x8f,0x27,0x90,0x16,0x91,0xfa,0x9
2,0x0b,0x93,0xa7,0x94,0x56,0x95,0x1f,0x96,0xf3,0x97,0xd7,0x9
8,0x64,0x99,0x21,0x9a,0x22,0x9b,0x0f,0x9c,0xc3,0x9d,0x79,0x9
e,0xbc,0x9f,0x29,0xa0,0x61,0xa1,0x55,0xa2,0x47,0xa3,0x0e,0xa
4,0xcb,0xa5,0x6c,0xa6,0x89,0xa7,0x36,0xa8,0xae,0xa9,0x7e,0xa
a,0xdb,0xab,0x72,0xac,0x3e,0xad,0xfd,0xae,0xff,0xaf,0xb4,0xb0,
0x5f,0xb1,0x9c,0xb2,0xdd,0xb3,0x76,0xb4,0x4b,0xb5,0xa3,0xb6,
0xfb,0xb7,0xc6,0xb8,0x03,0xb9,0x65,0xba,0x8b,0xbb,0x2e,0xbc,
0x9d,0xbd,0x5a,0xbe,0xf7,0xbf,0xf0,0xc0,0x46,0xc1,0xe4,0xc2,0
xb6,0xc3,0x83,0xc4,0xb2,0xc5,0x9b,0xc6,0x8a,0xc7,0xd0,0xc8,0
x01,0xc9,0x4d,0xca,0x25,0xcb,0x67,0xcc,0xab,0xcd,0x8f,0xce,0x
90,0xcf,0xd4,0xd0,0x10,0xd1,0x57,0xd2,0x86,0xd3,0xbe,0xd4,0x
ac,0xd5,0x73,0xd6,0x0d,0xd7,0x74,0xd8,0xd2,0xd9,0x24,0xda,0x
```


02,0xdb,0xcd,0xdc,0xa4,0xdd,0xf9,0xde,0x93,0xdf,0x52,0xe0,0x6d,0xe1,0xeb,0xe2,0xe0,0xe3,0x6b,0xe4,0x85,0xe5,0x14,0xe6,0xc7,0xe7,0x70,0xe8,0x58,0xe9,0xd8,0xea,0x32,0xeb,0xe3,0xec,0xc0,0xed,0xf1,0xee,0x78,0xef,0xe5,0xf0,0x30,0xf1,0x31,0xf2,0x88,0xf3,0xb3,0xf4,0x4e,0xf5,0xa6,0xf6,0xda,0xf7,0x28,0xf8,0xee,0xf9,0x75,0xfa,0x49,0xfb,0x2f,0xfc,0xe7,0xfd,0x63,0xfe,0x20,0xff,0xb5]

byte =

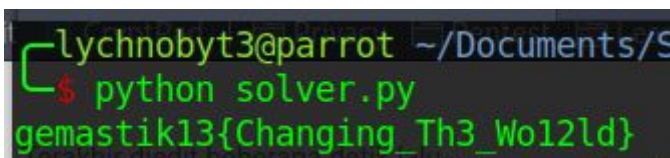
[0xd1,0x02,0xc4,0x03,0xc9,0x04,0x13,0x05,0x05,0x06,0x81,0x07,0x84,0x08,0x34,0x09,0x17,0x0a,0xe2,0x0b,0xc5,0x0c,0x5e,0x0d,0xe6,0x0e,0x4c,0x0f,0x51,0x10,0x26,0x11,0x12,0x12,0x68,0x13,0x07,0x14,0x7d,0x15,0x8e,0x16,0x69,0x17,0xce,0x18,0xd3,0x19,0x37,0x1a,0xb1,0x1b,0x0c,0x1c,0x11,0x1d,0x53,0x1e,0x9f,0x1f,0x91,0x20,0x09,0x21,0x99,0x22,0xaa,0x23,0xb7,0x24,0x06,0x25,0xf4,0x26,0xcc,0x27,0xa9,0x28,0xfc,0x29,0x6a,0x2a,0x08,0x2b,0x50,0x2c,0x1c,0x2d,0x45,0x2e,0xe8,0x2f,0xa8,0x30,0xcf,0x31,0x7b,0x32,0x6e,0x33,0x5c,0x34,0xb0,0x35,0xc1,0x36,0xde,0x37,0x7c,0x38,0xa0,0x39,0x66,0x3a,0x40,0x3b,0x19,0x3c,0xad,0x3d,0x43,0x3e,0xe9,0x3f,0x77,0x40,0xb8,0x41,0x2b,0x42,0x23,0x43,0xa5,0x44,0x4a,0x45,0x7f,0x46,0x1b,0x47,0xd5,0x48,0x1e,0x49,0xba,0x4a,0x5d,0x4b,0x33,0x4c,0xa1,0x4d,0xec,0x4e,0x60,0x4f,0x8d,0x50,0x04,0x51,0xbf,0x52,0xea,0x53,0x1a,0x54,0x98,0x55,0x2d,0x56,0xf2,0x57,0x41,0x58,0xef,0x59,0x15,0x5a,0x59,0x5b,0xf6,0x5c,0x6f,0x5d,0x2c,0x5e,0x95,0x5f,0x1d,0x60,0x5b,0x61,0xb9,0x62,0xf8,0x63,0x39,0x64,0x9a,0x65,0x3b,0x66,0x3c,0x67,0x3a,0x68,0x80,0x69,0x7a,0x6a,0xd6,0x6b,0xca,0x6c,0xaf,0x6d,0xdf,0x6e,0x38,0x6f,0x92,0x70,0x96,0x71,0x4f,0x72,0x0a,0x73,0x62,0x74,0xed,0x75,0xfe,0x76,0x3d,0x77,0x71,0x78,0x9e,0x79,0x3f,0x7a,0xc8,0x7b,0x48,0x7c,0x2a,0x7d,0x42,0x7e,0xe1,0x7f,0x97,0x80,0xdc,0x81,0xbb,0x82,0xbd,0x83,0xd9,0x84,0xc2,0x85,0x54,0x86,0x44,0x87,0x87,0x88,0x94,0x89,0x8c,0x8a,0x18,0x8b,0x82,0x8c,0x35,0x8d,0xa2,0x8e,0xf5,0x8f,0x27,0x90,0x16,0x91,0xfa,0x92,0x0b,0x93,0xa7,0x94,0x56,0x95,0x1f,0x96,0xf3,0x97,0xd7,0x98,0x64,0x99,0x21,0x9a,0x22,0x9b,0x0f,0x9c,0xc3,0x9d,0x79,0x9e,0xbc,0x9f,0x29,0xa0,0x61,0xa1,0x55,0xa2,0x47,0xa3,0x0e,0xa4,0xcb,0xa5,0x6c,0xa6,0x89,0xa7,0x36,0xa8,0xae,0xa9,0x7e,0xaa,0xdb,0xab,0x72,0xac,0x3e,0xad,0xfd,0xae,0xff,0xaf,0xb4,0xb0,0x5f,0xb1,0x9c,0xb2,0xdd,0xb3,0x76,0xb4,0x4b,0xb5,0xa3,0xb6,0xfb,0xb7,0xc6,0xb8,0x03,0xb9,0x65,0xba,0x8b,0xbb,0x2e,0xbc,

```
0x9d,0xbd,0x5a,0xbe,0xf7,0xbf,0xf0,0xc0,0x46,0xc1,0xe4,0xc2,0xb6,0xc3,0x83,0xc4,0xb2,0xc5,0x9b,0xc6,0x8a,0xc7,0xd0,0xc8,0x01,0xc9,0x4d,0xca,0x25,0xcb,0x67,0xcc,0xab,0xcd,0x8f,0xce,0x90,0xcf,0xd4,0xd0,0x10,0xd1,0x57,0xd2,0x86,0xd3,0xbe,0xd4,0xac,0xd5,0x73,0xd6,0x0d,0xd7,0x74,0xd8,0xd2,0xd9,0x24,0xda,0x02,0xdb,0xcd,0xdc,0xa4,0xdd,0xf9,0xde,0x93,0xdf,0x52,0xe0,0x6d,0xe1,0xeb,0xe2,0xe0,0xe3,0x6b,0xe4,0x85,0xe5,0x14,0xe6,0xc7,0xe7,0x70,0xe8,0x58,0xe9,0xd8,0xea,0x32,0xeb,0xe3,0xec,0xc0,0xed,0xf1,0xee,0x78,0xef,0xe5,0xf0,0x30,0xf1,0x31,0xf2,0x88,0xf3,0xb3,0xf4,0x4e,0xf5,0xa6,0xf6,0xda,0xf7,0x28,0xf8,0xee,0xf9,0x75,0xfa,0x49,0xfb,0x2f,0xfc,0xe7,0xfd,0x63,0xfe,0x20,0xff,0xb5]
```

```
flag = ""
for j in range(len(magic)):
    for i in range(255):
        if byte[i*2] == magic[j]:
            flag += chr(trans_tbl[i*2])
            break

print flag
```

Setelah dijalankan, hasilnya



```
lychnoby3@parrot ~/Documents/S
$ python solver.py
gemastik13{Changing_Th3_Wo12ld}
```

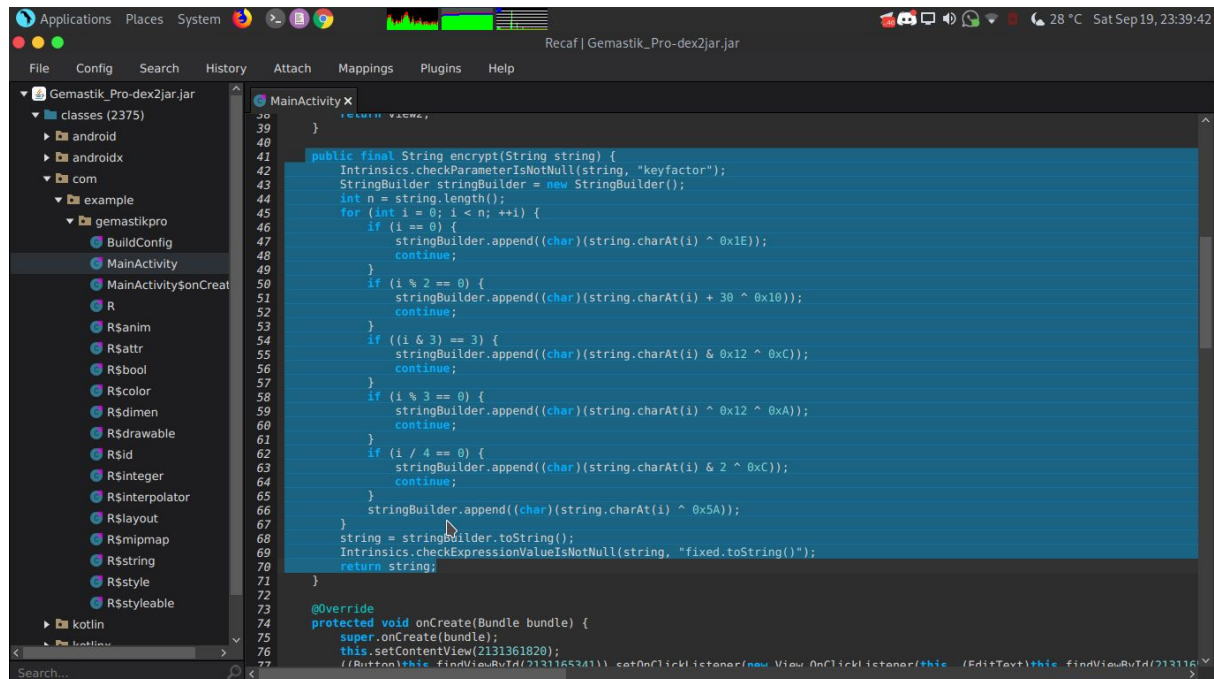
Flag : gemastik13{Changing_Th3_Wo12ld}

Gemastik Premium

Diberikan file .apk, langsung decompile. Intinya ada fungsi pengecekan dan enkripsi. Alur programnya nerima inputan, lalu dienkrpsi baru kemudian dicek. Untuk menyelesaikannya

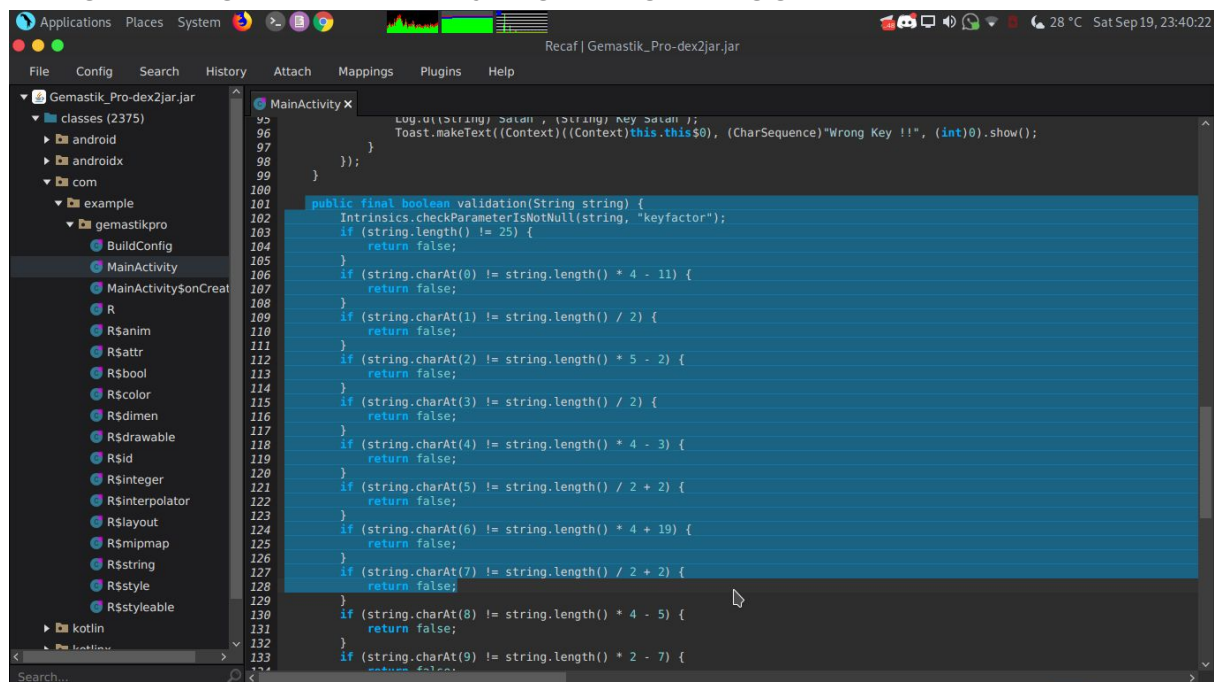
pertama cari string pengecekan, baru gunakan z3 untuk mencari inputan yang tepat untuk fungsi enkripsi.

Fungsi enkripsi



```
30
31
32
33
34
35
36
37
38
39
40
41 public final String encrypt(String string) {
42     Intrinsics.checkNotNull(string, "keyfactor");
43     StringBuilder stringBuilder = new StringBuilder();
44     int n = string.length();
45     for (int i = 0; i < n; ++i) {
46         if (i % 2 == 0) {
47             stringBuilder.append((char)(string.charAt(i) ^ 0x1E));
48             continue;
49         }
50         if (i % 2 == 0) {
51             stringBuilder.append((char)(string.charAt(i) + 30 ^ 0x10));
52             continue;
53         }
54         if ((i & 3) == 3) {
55             stringBuilder.append((char)(string.charAt(i) & 0x12 ^ 0xC));
56             continue;
57         }
58         if (i % 3 == 0) {
59             stringBuilder.append((char)(string.charAt(i) ^ 0x12 ^ 0xA));
60             continue;
61         }
62         if (i / 4 == 0) {
63             stringBuilder.append((char)(string.charAt(i) & 2 ^ 0xC));
64             continue;
65         }
66         stringBuilder.append((char)(string.charAt(i) ^ 0x5A));
67     }
68     string = stringBuilder.toString();
69     Intrinsics.checkNotNull(string, "fixed.toString()");
70     return string;
71 }
72
73 @Override
74 protected void onCreate(Bundle bundle) {
75     super.onCreate(bundle);
76     this.findViewById(2131361820);
77     findViewById(2131361821).setOnClickListener(new View.OnClickListener() {
78         public void onClick() {
79             // TODO: Implement the click listener logic
80         }
81     });
82 }
```

Fungsi pengecekan (panjang banget, ngga keliatan semua)



```
92
93
94
95
96
97
98
99
100
101 public final boolean validation(String string) {
102     Intrinsics.checkNotNull(string, "keyfactor");
103     if (string.length() != 25) {
104         return false;
105     }
106     if (string.charAt(0) != string.length() * 4 - 11) {
107         return false;
108     }
109     if (string.charAt(1) != string.length() / 2) {
110         return false;
111     }
112     if (string.charAt(2) != string.length() * 5 - 2) {
113         return false;
114     }
115     if (string.charAt(3) != string.length() / 2) {
116         return false;
117     }
118     if (string.charAt(4) != string.length() * 4 - 3) {
119         return false;
120     }
121     if (string.charAt(5) != string.length() / 2 + 2) {
122         return false;
123     }
124     if (string.charAt(6) != string.length() * 4 + 19) {
125         return false;
126     }
127     if (string.charAt(7) != string.length() / 2 + 2) {
128         return false;
129     }
130     if (string.charAt(8) != string.length() * 4 - 5) {
131         return false;
132     }
133     if (string.charAt(9) != string.length() * 2 - 7) {
134         return false;
135     }
136     return true;
137 }
```

Kemudian kami membuat script untuk mendapatkan flag, yang menurut kami berhasil. Tapi ngga, masih ada karakter yang hilang.

solver.py

```
password = [None] * 25
password[0] = 25 * 4 - 11
password[1] = 25 / 2
password[2] = 25 * 5 - 2
password[3] = 25 / 2
password[4] = 25 * 4 - 3
password[5] = 25 / 2 + 2
password[6] = 25 * 4 + 19
password[7] = 25 / 2 + 2
password[8] = 25 * 4 - 5
password[9] = 25 * 2 - 7
password[10] = 25 * 5 + 12
password[11] = 25 / 2
password[12] = 25 * 2 + 15
password[13] = 25 * 2 + 11
password[14] = 25 * 6 - 1
password[15] = 25 + 3
password[16] = 25 * 5 + 5
password[17] = 25 / 5
password[18] = 25 * 6 - 6
password[19] = 25 + 3
password[20] = 25 * 6 + 3
password[21] = 25 * 2 - 6
password[22] = 25 * 6 + 6
password[23] = 25 + 5
password[24] == 25 * 5 + 14;
```

```
from z3 import *
```

```
flag = [BitVec('x{}'.format(x), 32) for x in range(25)]
```

```
s = Solver()
```

```
for i in range(len(flag)): #printable range 0x20 - 0x7f atau
```

```

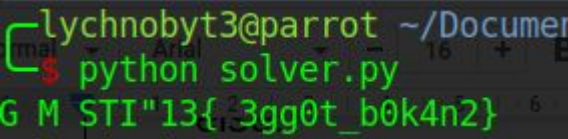
(32-127)
s.add(flag[i] >= 32)
s.add(flag[i] < 127)

for x in range(25):
    if (x == 0):
        s.add((flag[x] ^ 30) == password[x]);
    elif ((x % 2) == 0):
        s.add(((flag[x] + 30) ^ 16) == password[x]);
    elif ((x & 3) == 3):
        s.add(((flag[x] & 18) ^ 12) == password[x]);
    elif ((x % 3) == 0):
        s.add(((flag[x] ^ 18) ^ 10) == password[x]);
    elif ((x / 4) == 0):
        s.add(((flag[x] & 2) ^ 12) == password[x]);
    else:
        s.add((flag[x] ^ 90) == password[x]);

if s.check() == z3.sat:
    model = s.model()
    fixed = ".join([chr(int(str(model[flag[i]]))) for i in
range(25)])
    print fixed

```

Ketika dijalankan



```

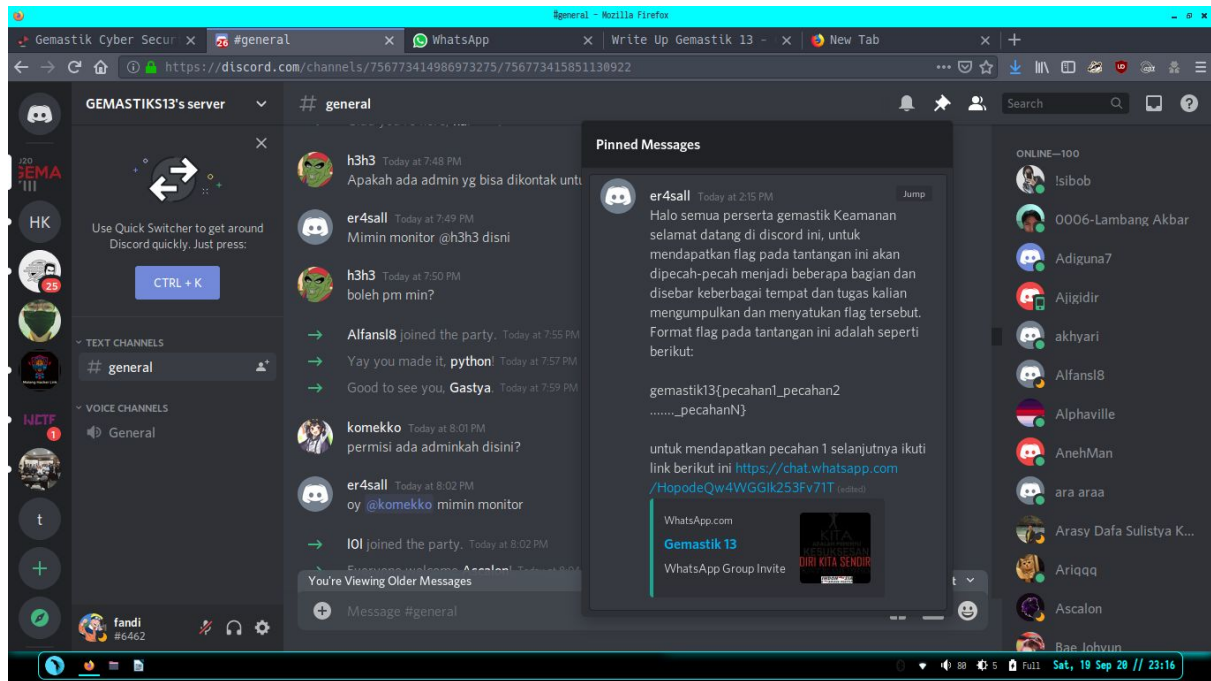
lychnoby3@parrot ~/Documen
$ python solver.py
G M STI"13{_3gg0t_b0k4n2}

```

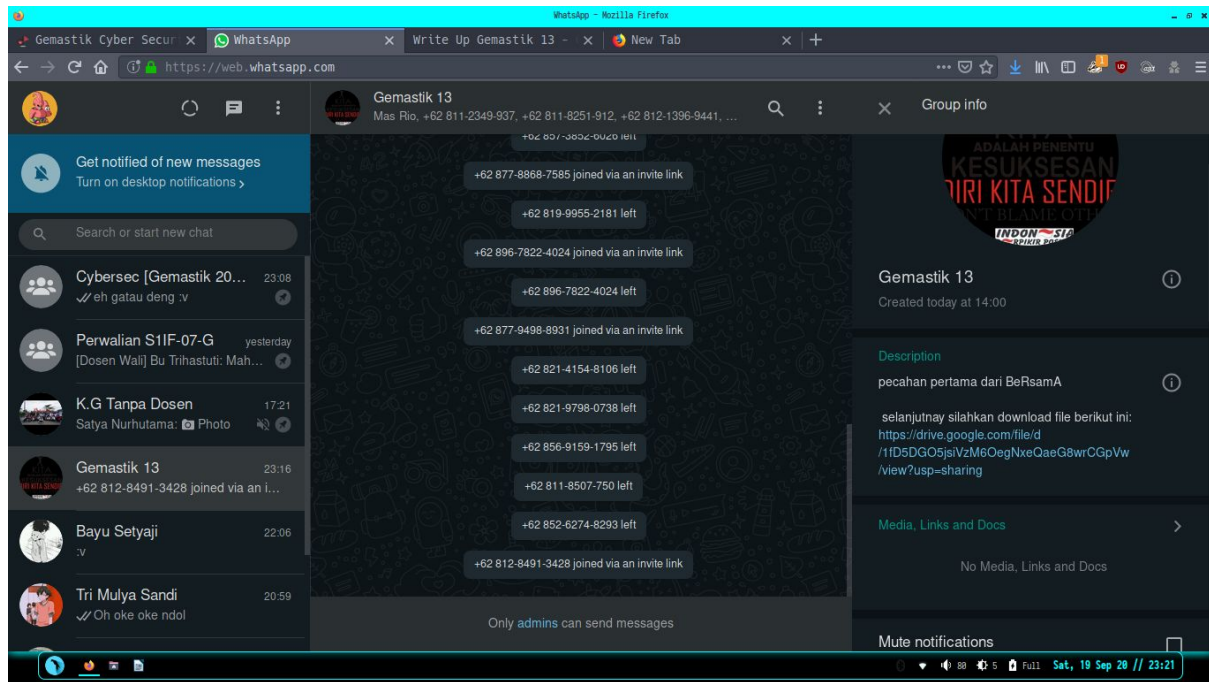
Dah, gitu aja. Ngga pinter nebak-nebak flag

MIX AROUND THE WORLD

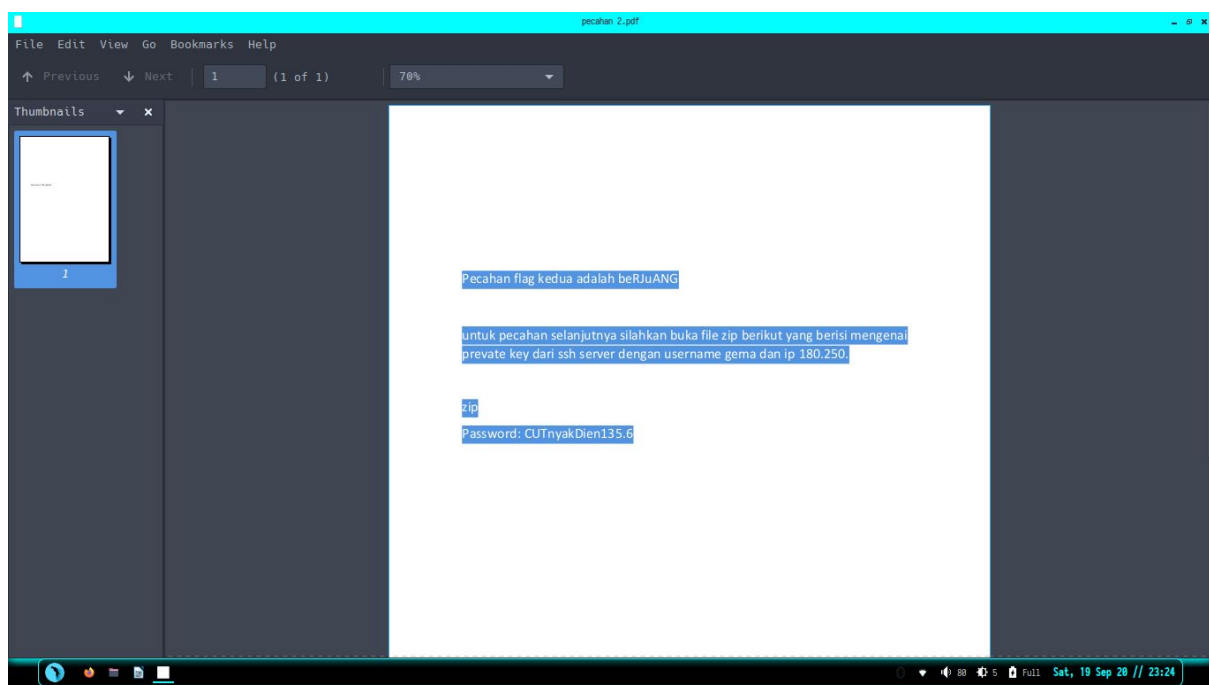
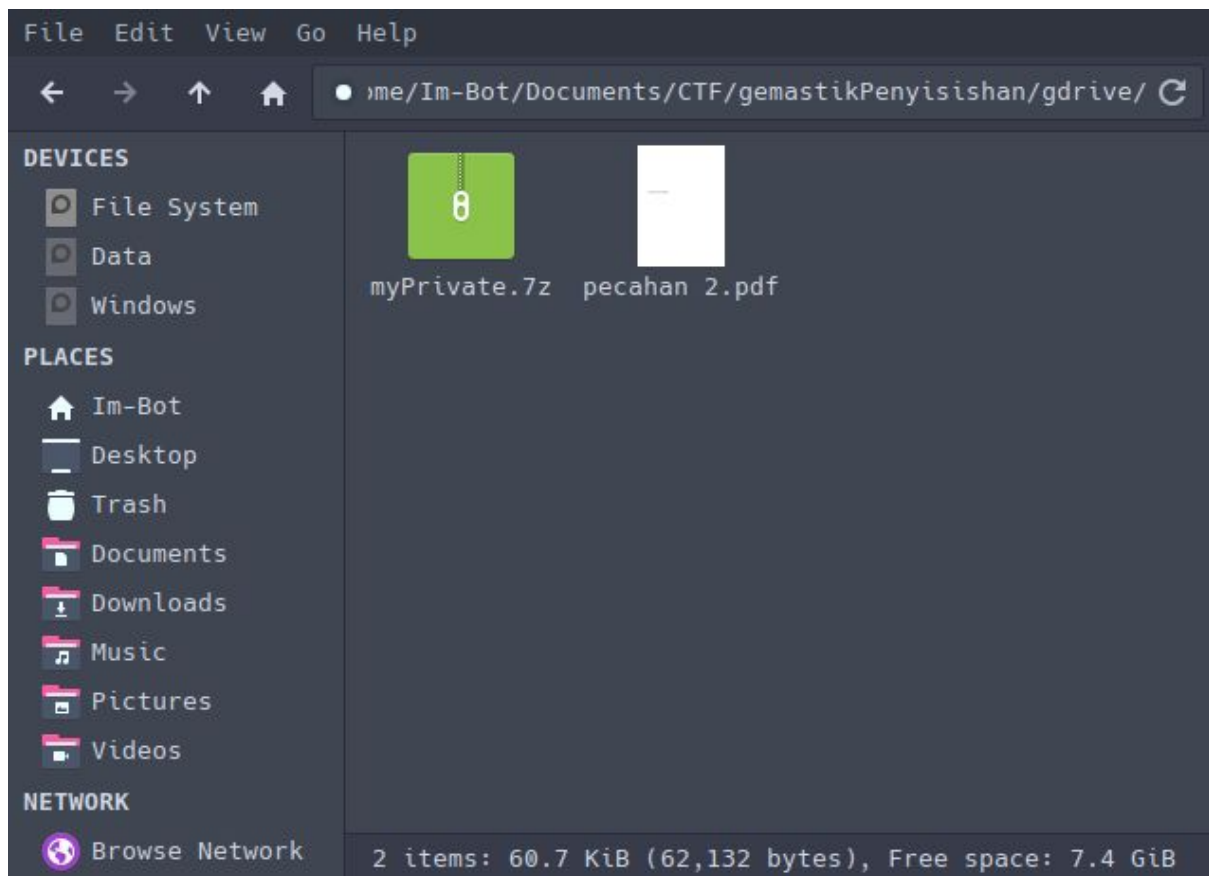
diberikan sebuah link discord, dimana panitia meminta peserta untuk masuk ke link discord tadi.



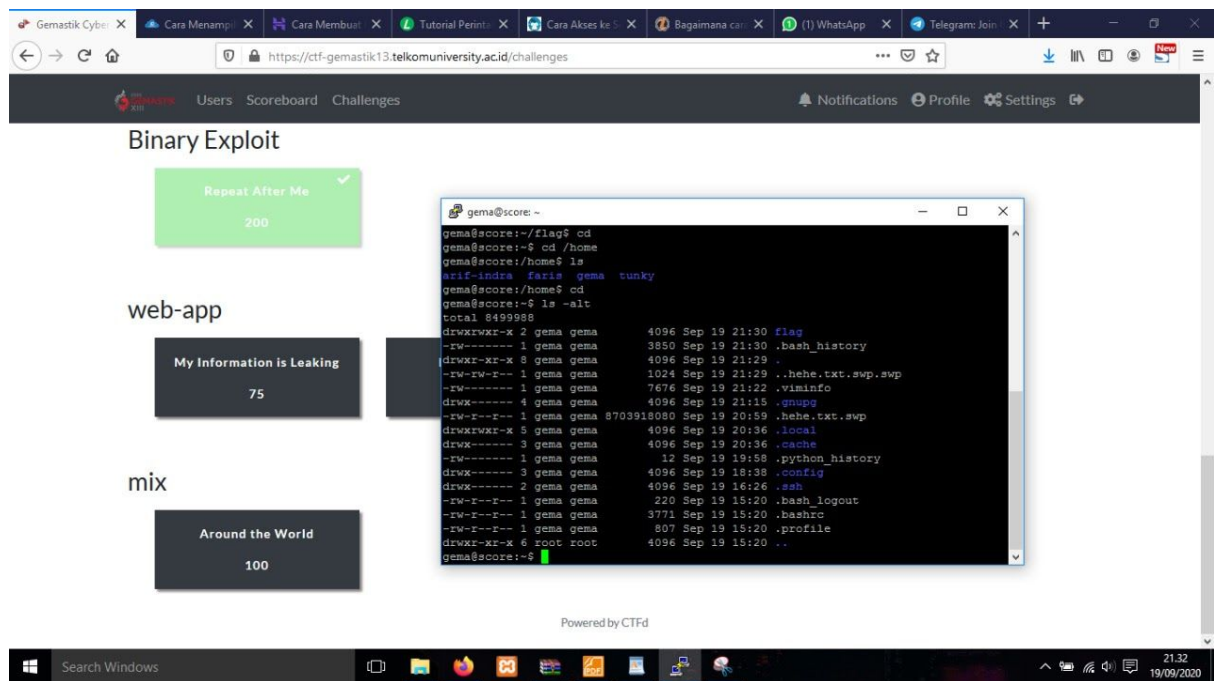
Kemudian di discord kami menemukan link untuk bergabung ke grup whatsapp dan di dekripsi kami menemukan pecahan ke-1 yaitu BeRsamA serta terdapat link ke google drive.



Setelah itu, kami mendownload file yang ada di google drive dan mengekstraknya terdapat dua file yaitu file berekstensi pdf dan 7z. Kami mencoba membuka file pdf yang ternyata ada pecahan kedua yaitu beRJuANG namun tidak terlihat karena teksnya berwarna putih serta terdapat ip untuk ssh dan password untuk mengekstrak file 7z yang berisi private key.



Kami pun mencoba mengaksesnya dengan putty, namun sayang flagnya tidak ada karena ada seseorang yang jail. Kami pun mencoba perintah `cat .bash_history` dan menemukan kode base64.



Dan setelah kami decode berisi pesan:

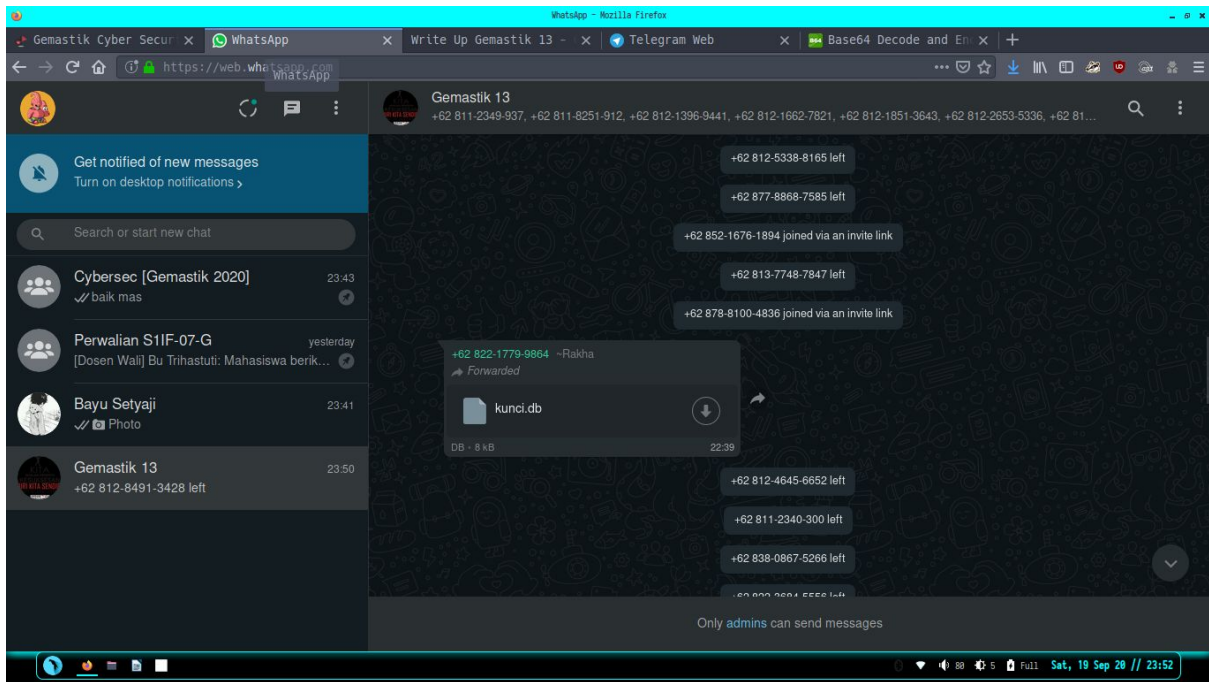
silahkan join group telegram berikut ini:

https://t.me/joinchat/KIHk-hx-4xizwwAcnuG_Zg

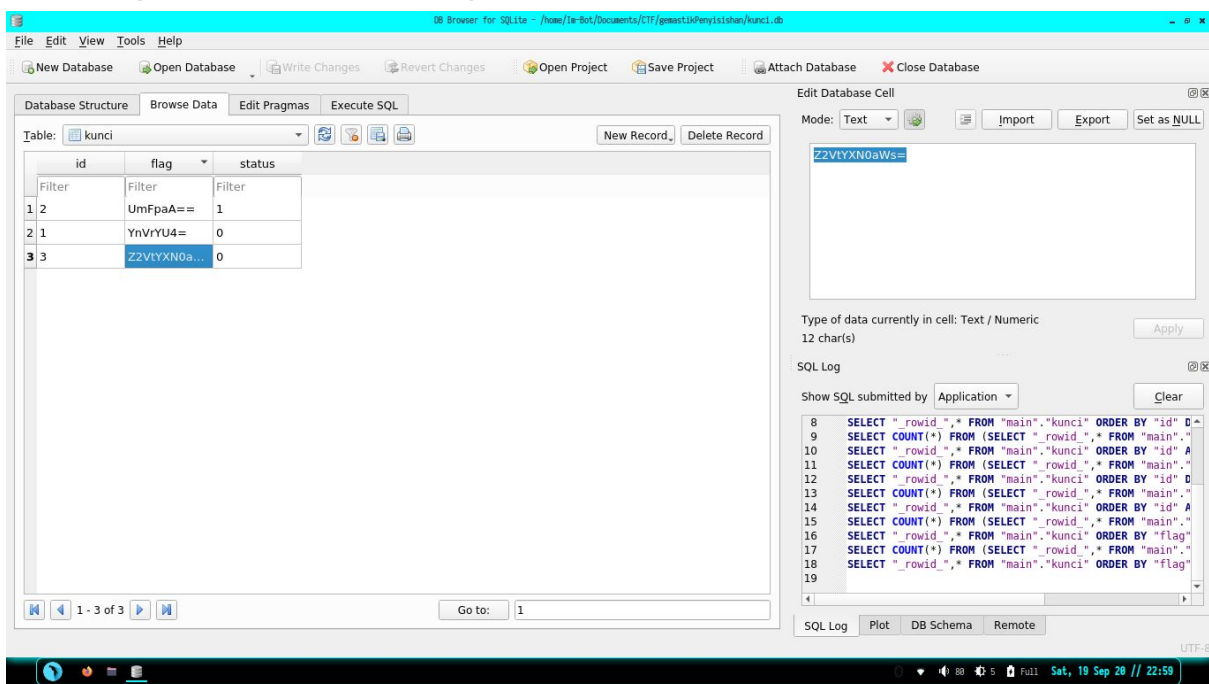
dan kami pun bergabung ke grup telegram tersebut dan menemukan encode base64 di bagian info:

cGVjYWWhhbiB0ZXJha2hpciBkYXJpIGZsYWcgaW5pIGFkYWxhaCB
rZU1FTkFOR2Fu

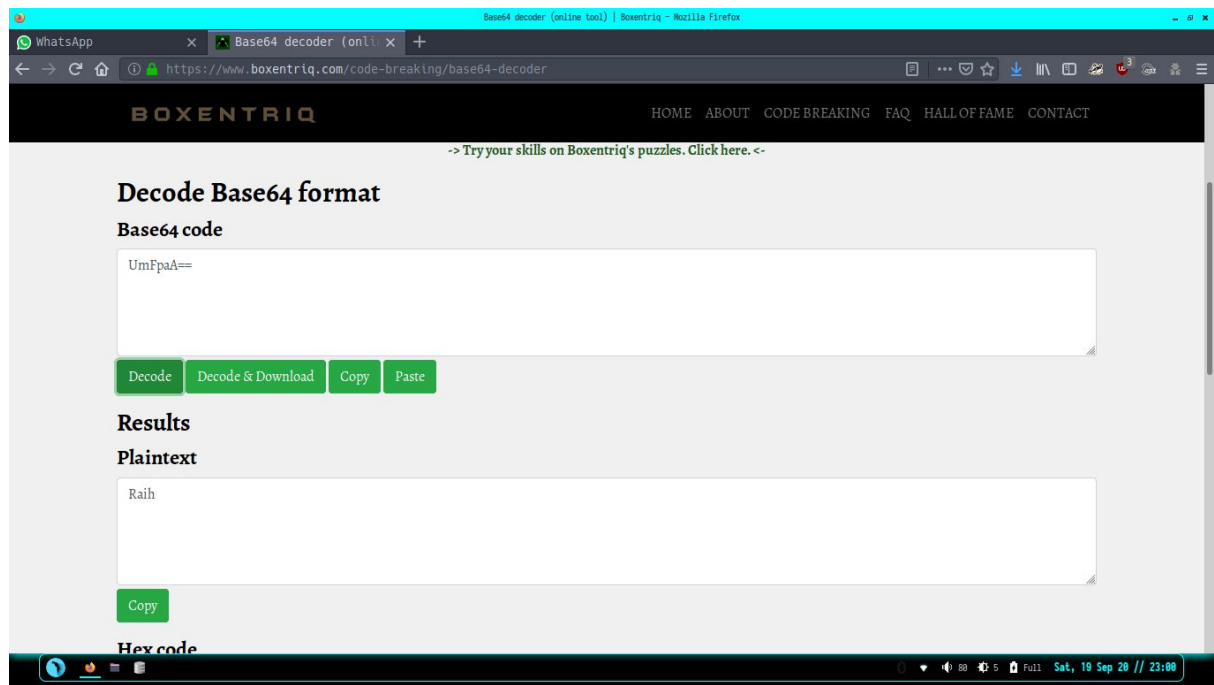
setelah didecode muncul pesan “pecahan terakhir dari flag ini adalah keMENANGAn”. Namun sayangnya pecahan ke-3 belum kami temukan karena ada seseorang yang jail tadi. Tetapi, setelah 30 menit panitia mengirim file ekstensi .db di grup Whatsapp



Setelah itu kami buka file tersebut dengan SQLite dan menampilkan informasi seperti di bawah



terdapat perbedaan pada field status yang mana hanya satu ada angka 1, maka kami decode baris nomor satu dan menemukan hasil seperti screenshot dibawah:



Flag : gemastik13{BeRsamA_beRJuaNG_Raih_keMENANGan}