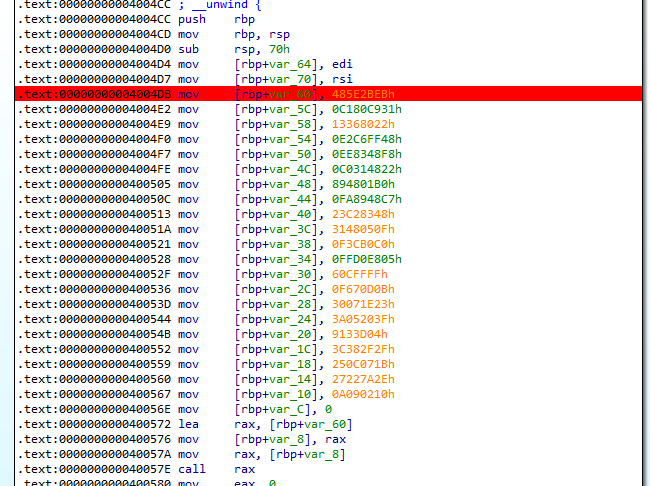
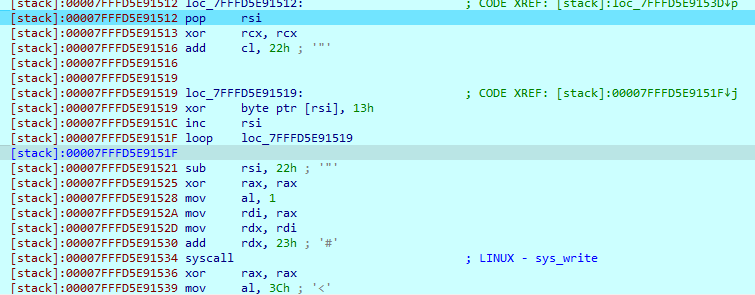
We got a x64 ELF file

Throw it into IDA



first at main, it write bunch of code into stack start with rbp+var\_60

Then it call at that address to execute the code



kind of simple code

simple pseudocode:

while(cl != 0){

[rsi] = [rsi] ^0x13;

rsi++;

}

(loop instruction in nasm use cl register as counter)

After that it print the string after enxor to screen, which is a rubbish

-A hint hear is the name of this challenge - wrong byte

-Because all things is so simple, write byte into stack, execute this code, this code is print a string after enxor with 0x13

-So i guess the wrong byte is 0x13

-So how to correct it, well, we know the form of the flag is FLAG-......, so i try some first bytes stored at RSI then enxor it with FLAG (first with F, second with L, so on,..) and the result is the same, 0x4a

-> t patch the 0x13 into 0x4a, then let the program run on

- I get the FLAG

