A screenshot of a computer program

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect. Writup

(gdb) break main

Breakpoint 1 at 0x1201

(gdb) run

(gdb) disass

A screen shot of a computer program

AI-generated content may be incorrect.

(gdb) break \*0x0000555555555d8

(gdb) continue

Enter a password: dfgfnhh

Breakpoint 2, 0x00005555555552d8 in main ()

(gdb) set $rax=0

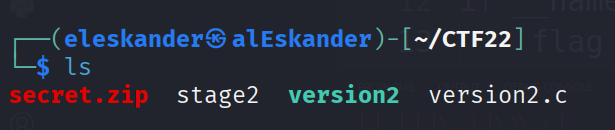
(gdb) continue

password is correct BUT!!!you want the flag ??

Oops, not everything you want comes easily try with the next stage

run ./stage2 to continue

\\ Now we have stage 2



Chmod +x stage2

A computer screen shot of a program

AI-generated content may be incorrect.

We see “Cyberus\_pass”

But it fake password

A screen shot of a computer code

AI-generated content may be incorrect.

We conclude that real password should be 8 chars and also should be correct

* Now using gdb

(gdb) break main

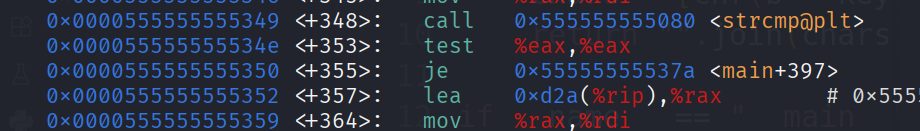
Breakpoint 1 at 0x11f1

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.



We have “strlen@plt” ,and two “strcmp@plt”

We will set $rax in strlen = 8

(gdb) break \*0x00005555555552b6

Breakpoint 2 at 0x5555555552b6

(gdb) break \*0x00005555555552ed

Breakpoint 3 at 0x5555555552ed

(gdb) break \*0x000055555555534e

Breakpoint 4 at 0x55555555534e

(gdb) continuing

Enter a password: dgfhjsf

Breakpoint 2, 0x00005555555552b6 in main ()

(gdb) set $rax=8

Breakpoint 3, 0x00005555555552ed in main ()

(gdb) set $rax=0

(gdb) continue

Continuing.

Try again, this is fake pass

Now we conclude that the first strcmp is just check for fake password so we will ignore it

(gdb) break \*0x00005555555552b6

Breakpoint 2 at 0x5555555552b6

(gdb) break \*0x000055555555534e

Breakpoint 3 at 0x55555555534e

(gdb) c

Continuing.

Enter a password: dsfg

Breakpoint 2, 0x00005555555552b6 in main ()

(gdb) set $rax=8

(gdb) c

Continuing.

Breakpoint 3, 0x000055555555534e in main ()

(gdb) set $rax=0

(gdb) c

Continuing.

Enter password for Stage 2:

// now we see another pass ? where is it



These two lines make “cmpl SRC, DEST”

So if we skip this step we will find the flag

Repeat all steps again then do these

(gdb) break \* 0x00005555555553eb

After write any pass

Breakpoint 5, 0x00005555555553eb in main ()

(gdb) jump \*0x555555555403

Continuing at 0x555555555403.

flag not here

flag on secret.zip

command hint: unzip file.zip

password: r3v3rs3\_pass\_enc

[Inferior 1 (process 123889) exited normally]

We have the password of secret.zip

Contains:

1. dec.py
2. encrypted\_flag
3. private.pem