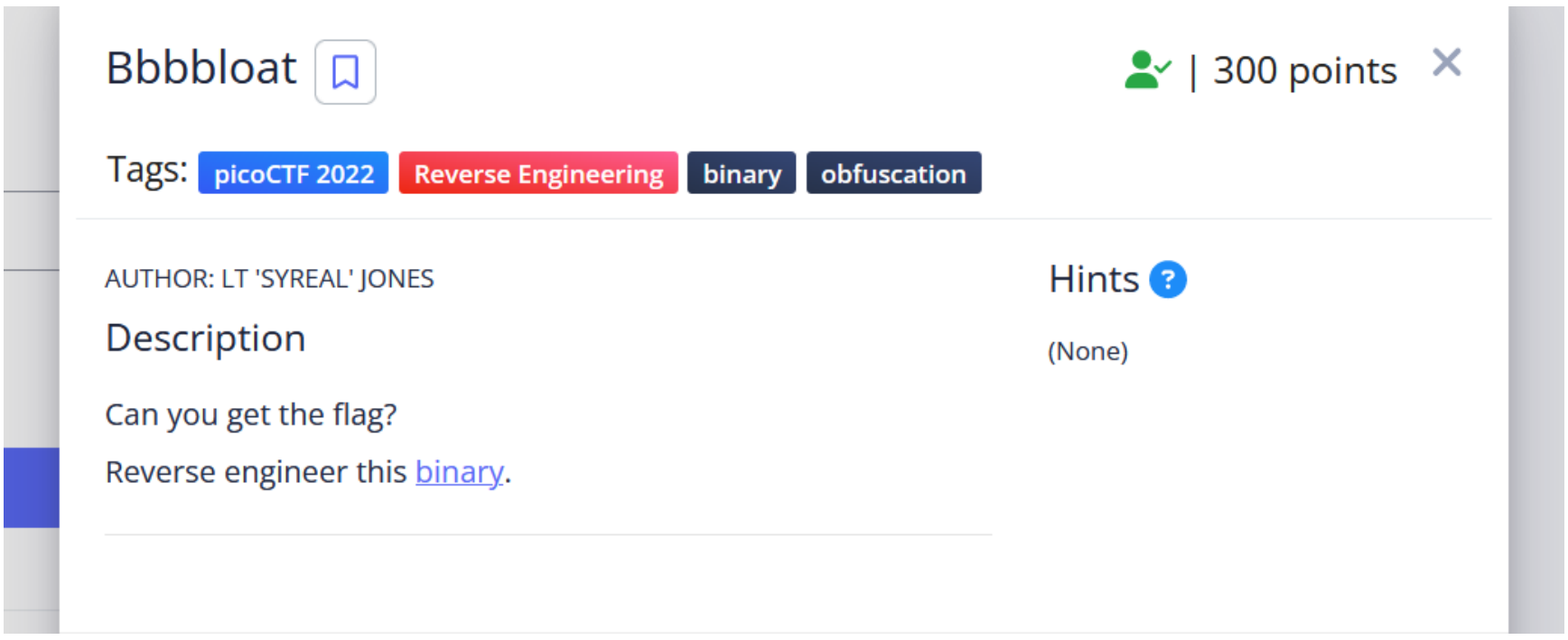
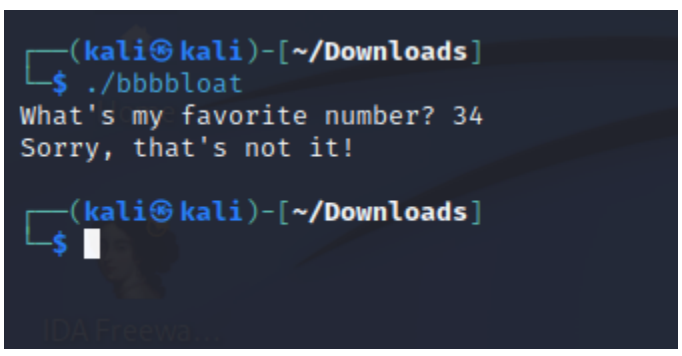
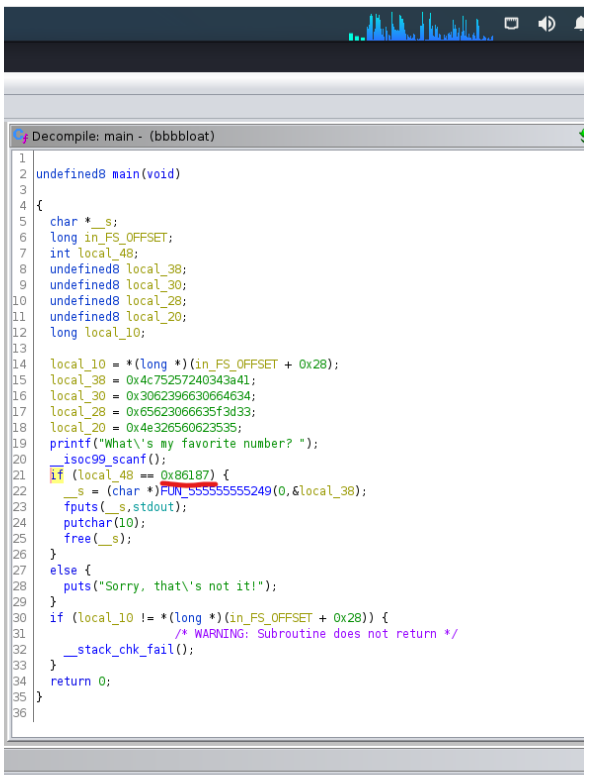
picoCTF Notes - Bbbbloat

-picoCTF is a computer security education program built on a capture-the-flag framework created by CMU security and privacy experts.



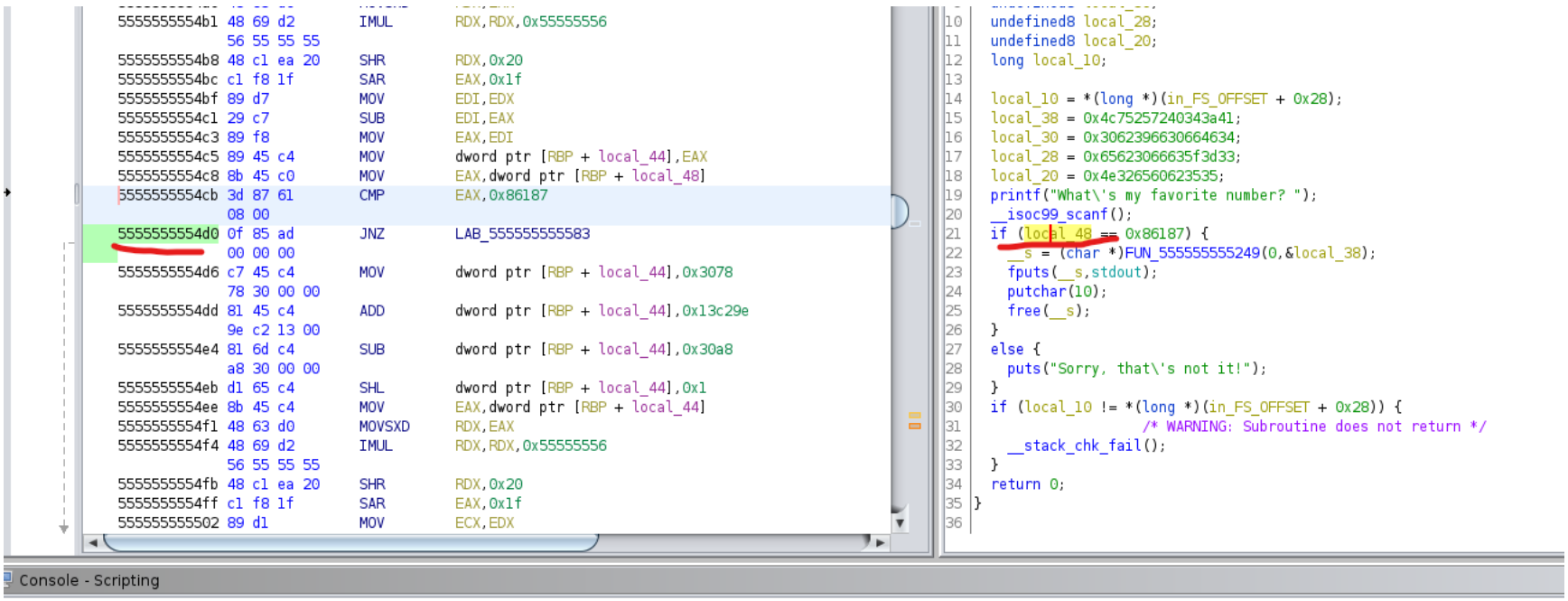
1. The program starts by prompting the user for a number and checks for the number in hex 0x861187 or 549255 in decimal and prints the flag to the screen if the user inputs the number correctly.





2. When looking at the disassembled code through ghidra, I found the comparison for the number as stated by the arrow. I can infer by the else statement that the if statement is the comparison for the number.

3. Then I used the gdb debugger to break on the if statement that compares the user input against 0x86187. You can look at the disassembly and it will mark the spot in the assembly language code that you can use to find the breakpoint in gdb. (Remember to set the base image to match gdb, otherwise, ghidra address will not match gdb.)




```
gef➤  info b
Num     Type           Disp Enb Address            What
2       breakpoint     keep y   0x00005555555554cb
        breakpoint already hit 1 time
gef➤
```

4. Then when the program gets to the cmp statement I set eax to 0x86187, which is where the user input is stored. I continued the program and was able to get the flag.

```
   0x5555555554c8                  mov    eax, DWORD PTR [rbp-0x40]
●→ 0x5555555554cb                  cmp    eax, 0x86187
   0x5555555554d0                  jne    0x555555555583
   0x5555555554d6                  mov    DWORD PTR [rbp-0x3c], 0x3078
   0x5555555554dd                  add    DWORD PTR [rbp-0x3c], 0x13c29e
   0x5555555554e4                  sub    DWORD PTR [rbp-0x3c], 0x30a8
   0x5555555554eb                  shl    DWORD PTR [rbp-0x3c], 1
──────────────────────────────────────────────────────────────────────────── threads ────
[#0] Id 1, Name: "bbbbloat", stopped 0x5555555554cb in ?? (), reason: BREAKPOINT
──────────────────────────────────────────────────────────────────────────── trace ────
[#0] 0x5555555554cb → cmp eax, 0x86187
[#1] 0x7ffff7c2920a → __libc_start_call_main(main=0x555555555307, argc=0x1, argv=0x7fffffffdf48)
[#2] 0x7ffff7c292bc → __libc_start_main_impl(main=0x555555555307, argc=0x1, argv=0x7fffffffdf48, init=<optimized
t>, fini=<optimized out>, rtld_fini=<optimized out>, stack_end=0x7fffffffdf38)
[#3] 0x55555555518e → hlt
──────────────────────────────────────────────────────────────────────────────────────
gef➤  set $eax=0x86187
gef➤  c
Continuing.
picoCTF{cu7_7h3_bl047_36dd316a}
[Inferior 1 (process 3532) exited normally]
```

Summary

Bbbbloat is a program that checks user input to see if it matches 0x86187 in hex or 549255 in decimal. I found this out by looking at the “if else statement” in disassembly to see that the compare function was only checking for one number. Then, when I ran the program in gdb, I set a breakpoint at the compare function. At the compare function, I set eax to 0x86187, and then the program prints the flag to the screen.