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.

```
2 undefined8 main(void)
   char *__s;
   long in_FS_OFFSET;
   int local_48;
   undefined8 local_38;
   undefined8 local_30;
   undefined8 local_28;
   undefined8 local_20;
    long local_10;
    local_10 = *(long *)(in_FS_0FFSET + 0x28);
    local_38 = 0x4c75257240343a41;
    local_30 = 0x3062396630664634;
    local_28 = 0x65623066635f3d33;
    local_20 = 0x4e326560623535;
    printf("What\'s my favorite number? ");
     _isoc99_scanf();
   if (local_48 == 0x86187) {
       s = (char *)FUN_555555555249(0,&local_38);
      fputs(_s,stdout);
     putchar(10);
     free(__s);
   else {
     puts("Sorry, that\'s not it!");
   if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
                     /* WARNING: Subroutine does not return */
   __stack_chk_fail();
}
   return 0;
```

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.)

```
5555555554b1 48 69 d2
                                                      RDX, RDX, 0x5555556
                                                                                                                         undefined8 local_28;
                                                                                                                         undefined8 local_20;
                          56 55 55 55
             5555555554b8 48 cl ea 20
                                           SHR
                                                      RDX, 0x20
                                                                                                                         long local_10;
             5555555554bc cl f8 lf
                                                      EAX, 0x1f
             5555555554bf 89 d7
                                           MOV
                                                      EDI, EDX
                                                                                                                         local_10 = *(long *)(in_FS_0FFSET + 0x28);
                                                                                                                         local_38 = 0x4c75257240343a41;
local_30 = 0x3062396630664634;
             5555555554c1 29 c7
                                           SUB
                                                      EDI, EAX
             5555555554c3 89 f8
             55555555554c5 89 45 c4
                                           MOV
                                                      dword ptr [RBP + local_44],EAX
                                                                                                                         local_28 = 0x65623066635f3d33;
             5555555554c8 8b 45 c0
                                                                                                                         local 20 = 0x4e326560623535;
                                           MOV
                                                      EAX, dword ptr [RBP + local_48]
             5555555554cb 3d 87 61
                                                                                                                        printf("What\'s my favorite number? ");
                                                                                                                        08 00
             5555555554d0 Of 85 ad
                                           JNZ
                                                      LAB 55555555583
                                                                                                                          fputs(__s,stdout);
putchar(10);
free(__s);
             5555555554d6 c7 45 c4
                                           MOV
                                                      dword ptr [RBP + local_44],0x3078
                          78 30 00 00
             5555555554dd 81 45 c4
                                                      dword ptr [RBP + local_44],0x13c29e
                                                                                                                   26
27
                          9e c2 13 00
             5555555554e4 81 6d c4
                                                      dword ptr [RBP + local 44].0x30a8
                                          SUB
                                                                                                                         else {
                                                                                                                   28
                                                                                                                          puts("Sorry, that\'s not it!");
             5555555554eb d1 65 c4
                                                                                                                   29
30
31
                                           SHI
                                                      dword ptr [RBP + local_44],0x1
                                                                                                                        if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
             5555555554ee 8b 45 c4
                                                      EAX, dword ptr [RBP + local_44]
                                           MOV
             5555555554f1 48 63 d0
                                           MOVSXD
                                                      RDX, EAX
                                                                                                                                            /* WARNING: Subroutine does not return */
             5555555554f4 48 69 d2
                                          IMUL
                                                      RDX, RDX, 0x5555556
                                                                                                                   32
33
                                                                                                                           __stack_chk_fail();
                          56 55 55 55
             5555555554fb 48 cl ea 20
                                                       RDX, 0x20
                                                                                                                         return 0;
                                                                                                                   35 }
36
             5555555554ff cl f8 lf
                                                      EAX, 0x1f
             555555555502 89 d1
                                                      ECX, EDX
Console - Scripting
```

```
info b
                       Disp Enb Address
Num
                                                    What
                       keep y 0×00005555555554cb
2
        breakpoint
        breakpoint already hit 1 time
```

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.

```
0×555555554d0
                               jne
                                     0×55555555583
  0×555555554d6
                                     DWORD PTR [rbp-0\times3c], 0\times3078
                               mov
                                     DWORD PTR [rbp-0×3c], 0×13c29e
  0×555555554dd
                               add
  0×555555554e4
                                     DWORD PTR [rbp-0×3c], 0×30a8
                               sub
  0×555555554eb
                                     DWORD PTR [rbp-0×3c], 1
                               shl
[#0] Id 1, Name: "bbbbloat", stopped 0×555555554cb in ?? (), reason: BREAKPOINT
[#0] 0×555555554cb →
[#2] 0×7fffff7c292bc → __libc_start_main_impl(main=0×5555555555307, argc=0×1, argv=0×7fffffffffff48, init=<optimized
t>, fini=<optimized out>, rtld_fini=<optimized out>, stack_end=0×7fffffffdf38)
[#3] 0×5555555518e →
gef> set $eax=0×86187
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.