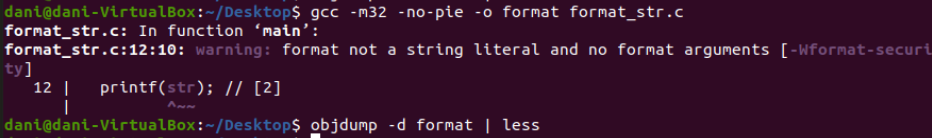
**Problem 1**

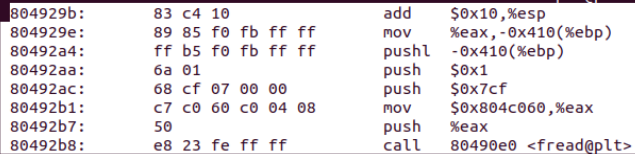
**1**

Here is how I compiled the program:

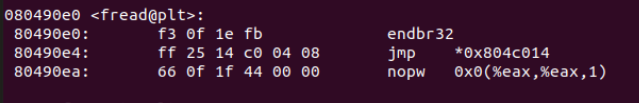


**2**

Here is the part that is responsible for passing in arguments:



Here is the disassembled code for fread:



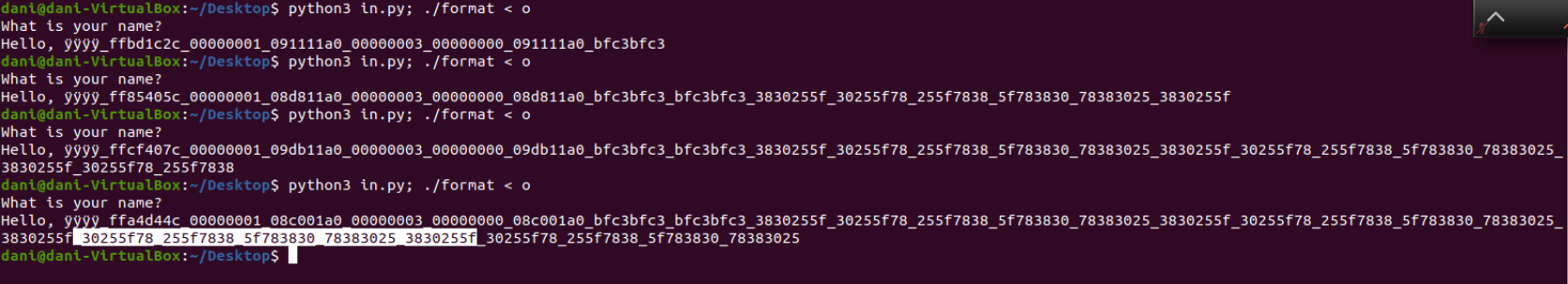
So as seen in the first screenshot, we see that the arguments get passed into the function from right to left. So the first pushl is used to pass in the memory address of where the file that pointer f is referring too. The next push 0x1 passes in the argument 1 to the function fread and similarly push 0x7cf pushes 1999 to fread. We see that the value being moved into eax and then subsequently being pushed to the stack will have to be the memory address of passwd hence, we have the location in memory where passwd is located.

**3**

0x804c060 since that is the last argument passed into the function fread.

**4**

I added “\_%08x” until I began to see a pattern:



So the repeated string is:

\_30255f08\_255f7838\_5f783830\_78383025\_3830255f

FINISH THE LAST SENTENCE

**5**

We essentially just change the line printf(str); to printf(“%s”, str); this will elimate the vulnerability that an attacker is able to use the format string attack and hence just print out what is inputted rather than looking at any memory. The attack no longer works since we provide the formatting of the string so the only thing passed in to printf is a string rather a formatting of the string and what to print out eliminating the extra step that is taken when using the first printf.

**Problem 2**

**1**