Inspector

First, I ran checksec and determined that canaries and PIE were disabled. This means we can write past the buffer and jump to function addresses that will remain unchanging. I wrote garbage into the buffer and set the return address above rbp to be the address of “pop rdi, ret”. Then, I wrote the next value to be the value I wanted to set rdi to. I repeated this for the registers rsi, rax, and rdx. This works because the function returns to the section of code where “pop register, ret” is occurring and since the next value on the stack is the value we want to put on the register, that gets popped and set as the value of the register. Then, the program returns to the next value in the stack, which causes the program to start executing from “pop next_register, ret”. Rinse and repeat. Using this, I set up the registers for an execve call and then made the program return to a part of the binary that is executing syscall. Thus, I was able to obtain the flag.