Dylan Aegbuniwe

CSC 472

Final Lab

12/17/2021

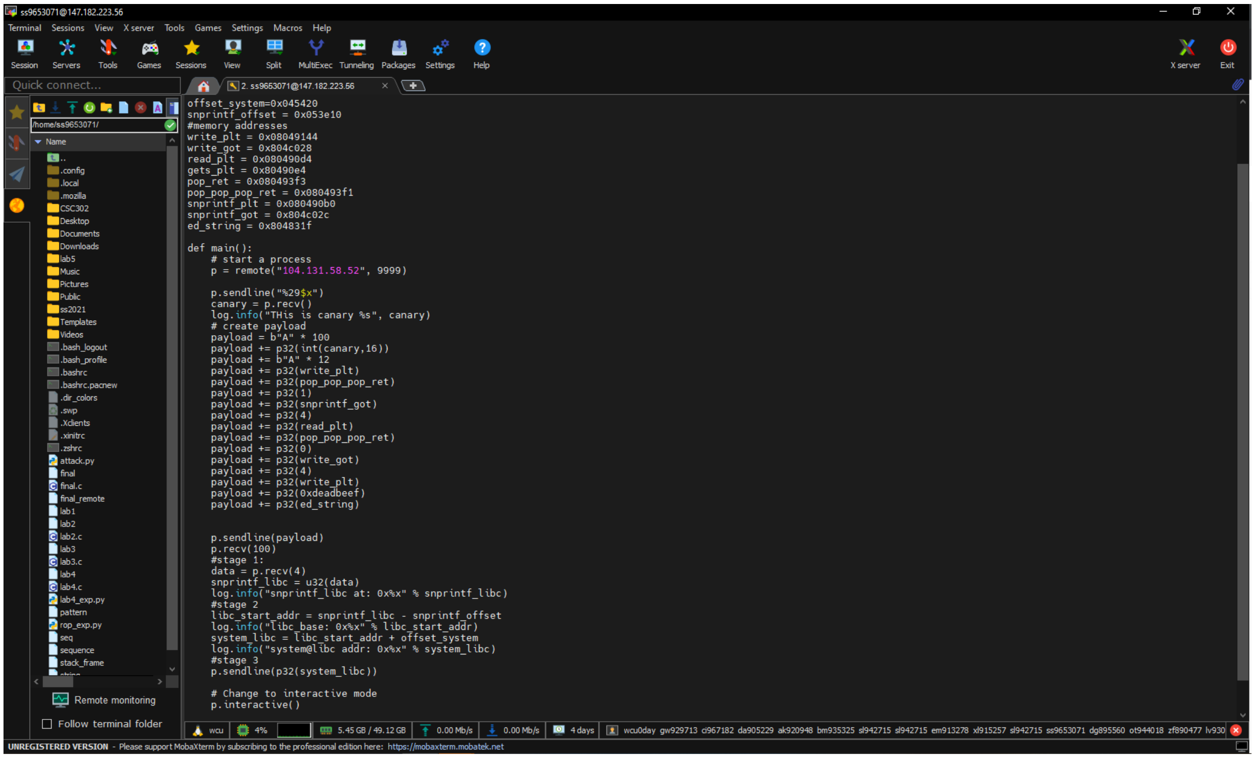
Introduction

In this lab, we are exploiting a vulnerable code with leaking data values for canary and buffer overflow using gdb and Python.

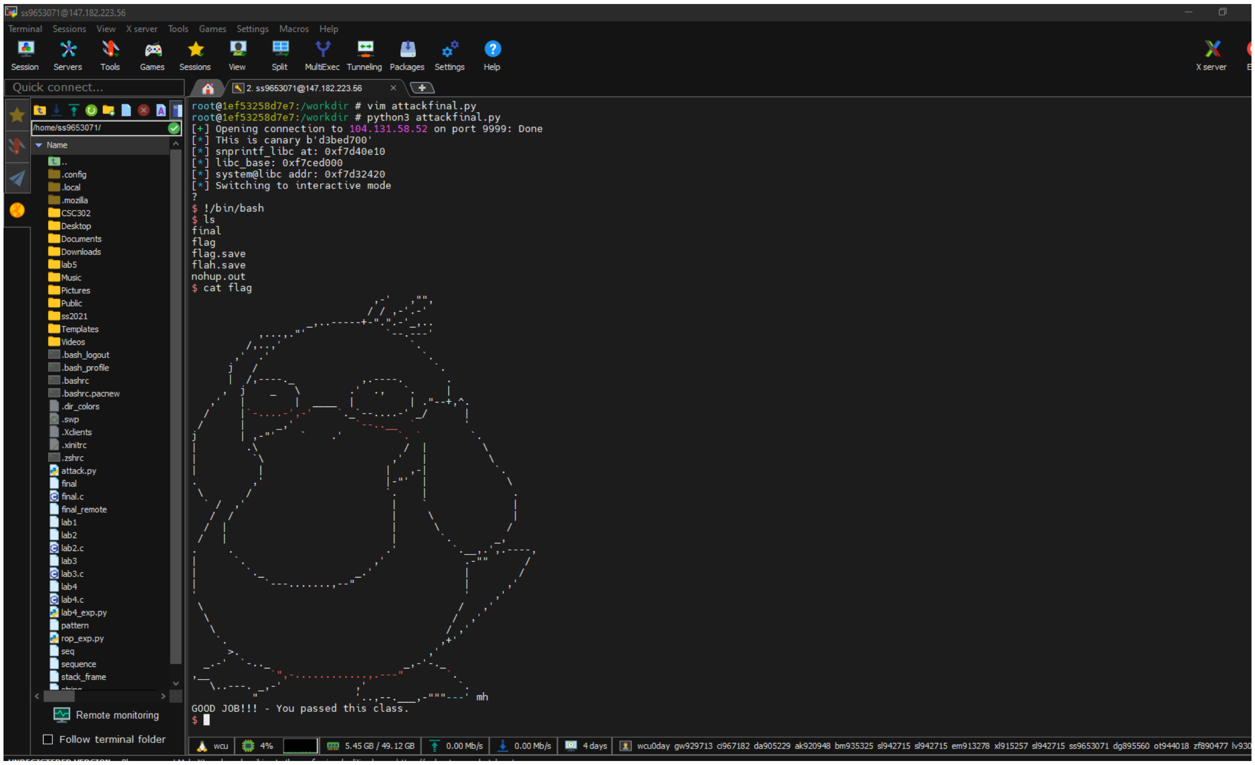
Analysis and Results

Dummy Character "A" * 100
Canary value
Dummy Character "A" * 12
write@plt
Address of pop, pop, pop, ret gadget
1
snprintf@got
4
read@plt
Address of pop, pop, pop, ret gadget
0
system@plt -> write@plt
0xdeadbeef
"ed" string

This table shows the order of the payload and how it matches each of the required values in the final.c code to pass through each function. It works off of a multi-stage process going buff overflow > obtain canary value > leak information > got overwrite > spawn shell.



This is the main portion of the payload, the edited data that hacks into the final.c file.



This is the output when running the exploit, getting into the new shell code and getting inside the flag.

Discussion and Conclusion

This lab satisfied the purpose of using the canary value and buffer overflow to hack a vulnerable code, as well as using the gdb debugger to spotlight certain addresses to see their vulnerabilities. The exploit was all done in Python. In the end, the result was what was wanted as the exploit successfully hacked into the final.c code, entered the shell and opened the flag.txt file.