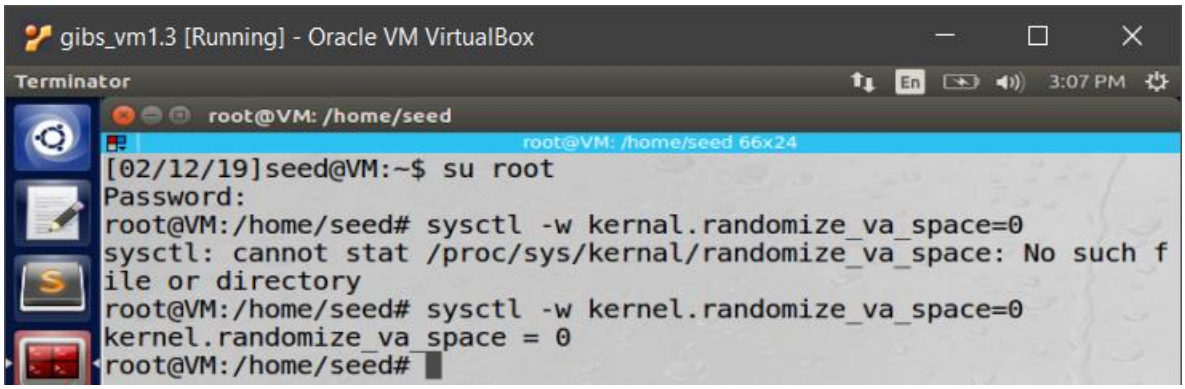
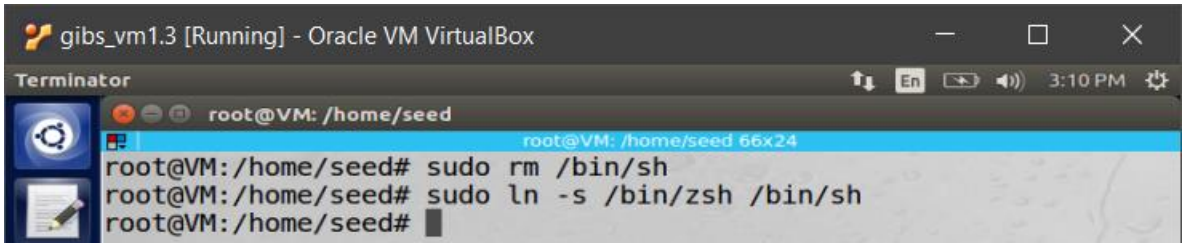
Gabriel Brehm
CS360

Lab 3

2.1 Initial Setup:

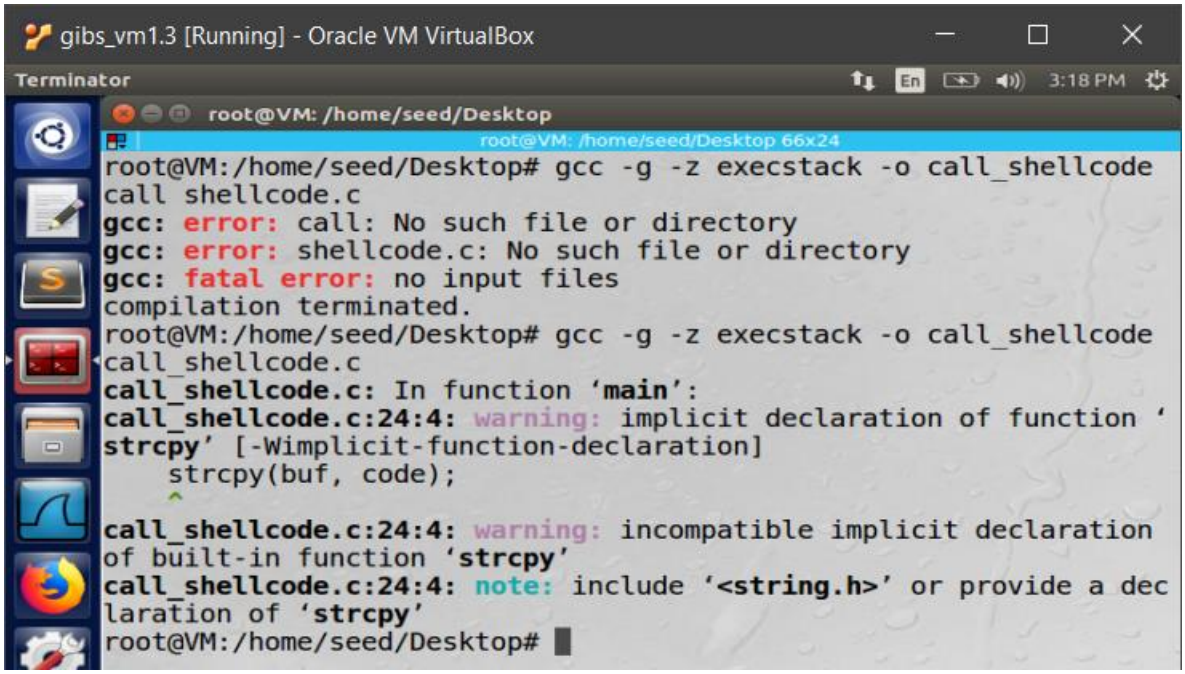


Here I disabled the address randomization.



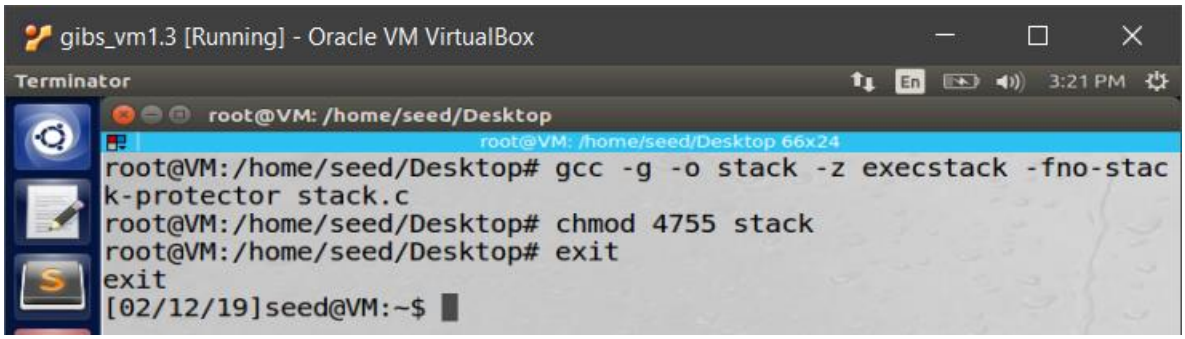
The above code was provided by my instructor. It removes the shell and replaces it with the zshell.

2.2 Shellcode:



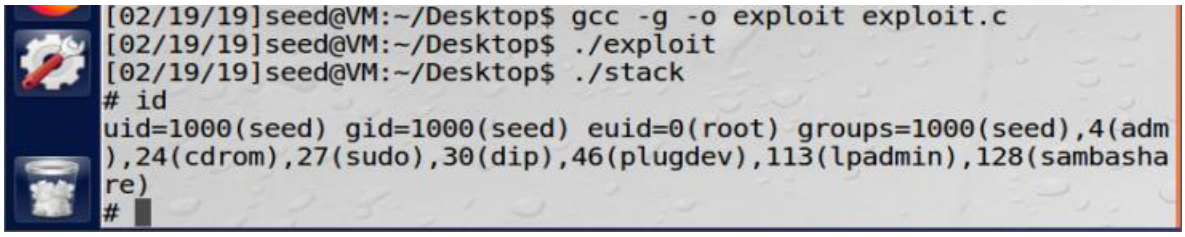
Here we are creating the executable shellcode.

2.3 The Vulnerable Program:



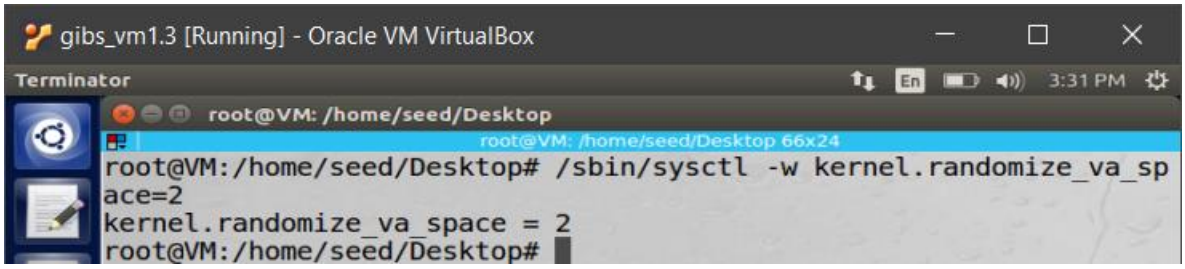
Here we compile the vulnerable program and make it a set-root-uid.

2.4 Task 1:

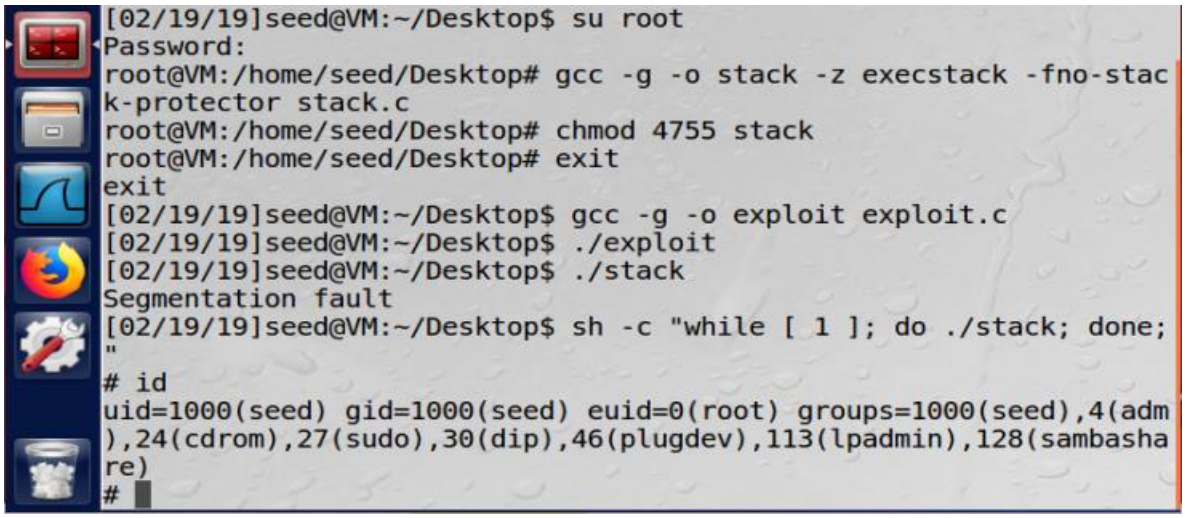


We now can run the exploit program and the stack program, and we get root access, as desired.

2.5 Task 2 Address Randomization:

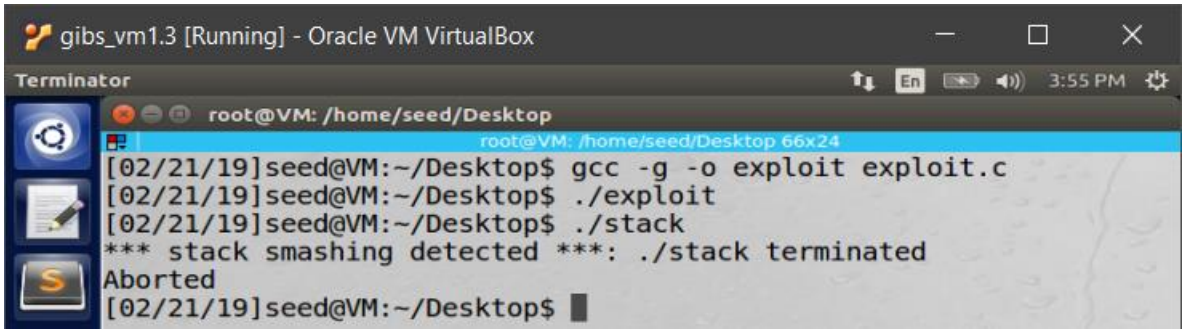


Here we turn on the randomization.



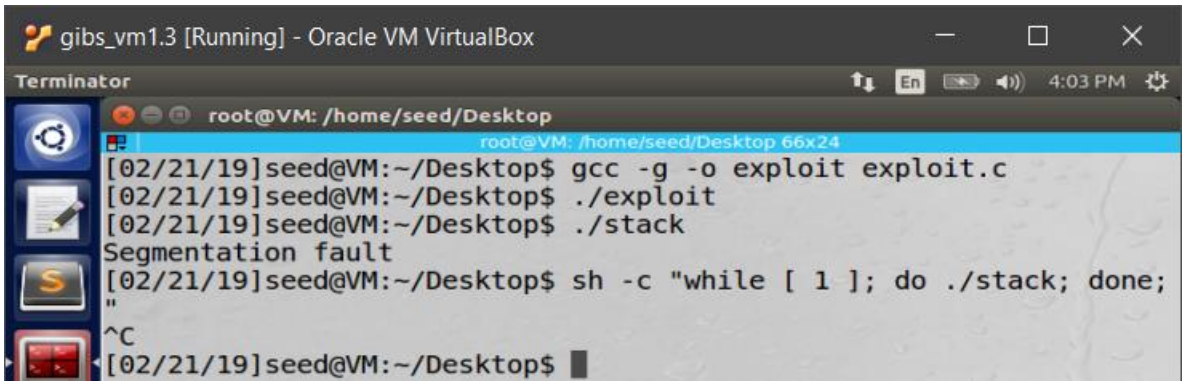
We then repeat all of the compilation of the code and run it all again, this time with the address randomization turned on. We can see that executing stack once results in a segmentation fault, but if we run a loop, we will eventually gain root access.

2.6 Task 3 Stack Guard:



When we reenable the stack guard, and recompile everything, we are not able to get root access. We get the error message, "*** stack smashing detected ***: ./stack terminated."

2.7 Task 4 Non-executable Stack:



For this part, we used "noexecstack" instead of "execstack" when performing Task 1 again. This resulted in a segmentation fault when we tried to execute the stack program. Address randomization was disabled at this point, but I wanted to see if it would work if we ran the stack program in a loop, but it did nothing for several minutes. I conclude that we cannot gain root access with the nonexecutable stack.