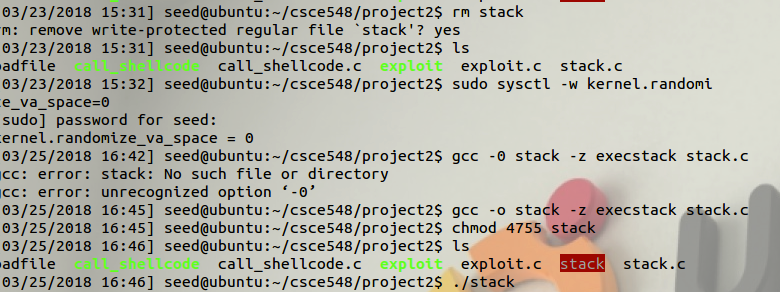
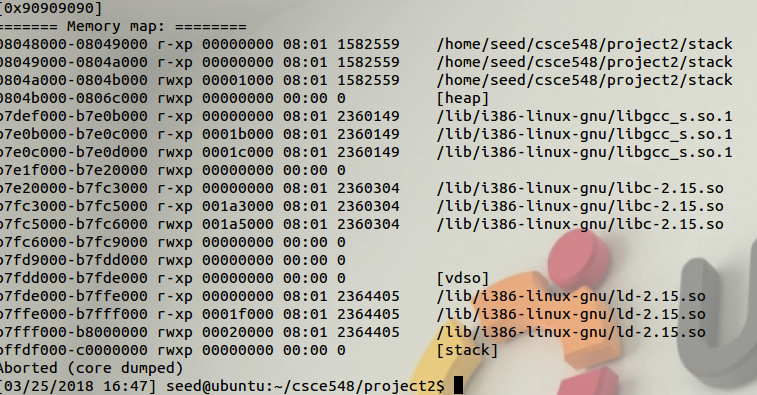
﻿Lab 2 Report

Andrew Cox, Sebastian Martin, Joy Ray, Xiyuan Zheng

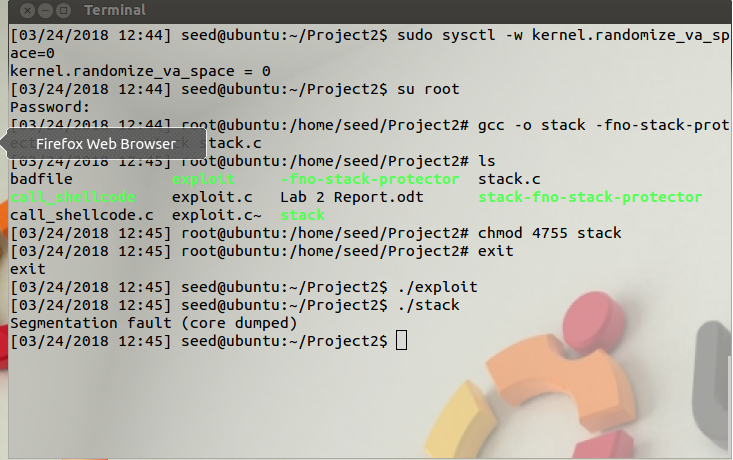
Task 2:

Task 3: No, I am not able to get the shell. In order to perform this task, I need to turn off address randomization defense. Then I repeated the buffer overflow attack again with GCC’s Stack Guard. Stack Guard enhances the code by detecting buffer overflow attacks against stack. It allows us to prevent the attempt of changing address before the function returns. Therefore, the program was terminated, since the smash attack was detected.





Task 4:No, I cannot get to the shell when using “noexecstack.” When I make the addresses in the stack non-executable it prevents the exploit.c code because all the writable addresses in the stack are now non-executable. Since the addresses now cannot be executed the code in exploit.c will not work because it is dependent on being able to get to the return address of the stack. Although this makes it difficult to cause a buffer overflow it does not prevent the buffer overflow completely. The “return-to-libc” attack would get around the “noexecstack” and this is done because it calls functions that are in libc and do not reside in the stack.



Group Contributions:

Andrew Cox -

Sebastian Martin -

Joy Ray – Completed Task 4 and wrote the observations of what happens when the addresses of the stack are non-executable

Xiyuan Zheng