Song Spring 2015	CS 161 Computer Security	Discussion 3

Instructions. We will break into groups to discuss the following questions. Please think of as many solutions as you can. Be original! Maybe you will come up with something no one has thought of yet. Be prepared to talk about your solutions with the rest of the section.

Question 1 ***Buffer Overflow Mitigations*** **(10 min)**

Buffer overflow mitigations generally fall into two categories: (1) eliminating the cause and (2) alleviating the damage. This question is about techniques in the second category.

Several requirements must be met for a buffer overflow to succeed. Each requirement listed below can be combated with a different countermeasure. With each mitigation you discuss, think about *where* it can be implemented—common targets include the compiler and the operating system (OS). Also discuss limitations, pitfalls, and costs of each mitigation.

(a) The attacker needs to overwrite the return address on the stack to change the control flow. Is it possible to prevent this from happening or at least detect when it occurs?

(b) The overwritten return address must point to a valid instruction sequence. The attacker often places the malicious code to execute in the vulnerable buffer. However, the buffer address must be known to set up the jump target correctly. One way to find out this address is to observe the program in a debugger. This works because the address tends to same across multiple program runs. What could be done to make it harder to accurately find out the address of the start of the malicious code?

(c) Attackers often store their malicious code inside the same buffer that they overflow. What mechanism could prevent the execution of the malicious code? What type of code would break with this defense in place?

Question 2 ***Arc Injection*** **(10 min)**

Imagine that you are trying to exploit a buffer overflow, but you notice that none of the code you are injecting will execute for some reason. How frustrating! You still really want to run some malicious code, so what could you try instead?

Hint: In a stack smashing attack, you can overwrite the return address with any address of your choosing.