**HOMEWORK 11**

**What are two different ways to succeed at a stack smashing attack described in the paper?**

A buffer overflow approach is used in this paper to succeed at a stack smashing attack.

Buffer overflow - Writing a code that overflows a buffer, thereby overwriting data on the stack, allowing you to call other code.

▪ *Jump to address stored in a register technique:* Overwriting code in a way that the function return will jump to an address specified. To do this, location of the stack pointer and size of the buffer should be known.

▪ *NOP Sled Technique:* Padding code with NOP sled. After the return statement is executed, the jump lands somewhere in NOP sled and executes all of the NOPs until it reaches the code. The NOP sled gives a broad range of memory addresses to jump that will result in the code being executed, increasing the likelihood that the code actually gets executed

**How does the paper recommend you find a buffer overflow vulnerability?**

By searching for functions for copying or appending strings that perform no boundary checking is a way of finding buffer flow vulnerability mentioned in the paper. These include

strcat()

strcpy()

sprintf()

vsprintf()

These functions operate on null-terminated strings, and do not check for overflow of the receiving string.

Other vulnerable functions are:

getc()

fgetc()

getchar()

If there are no explicit checks for overflows when using these functions in a while loop, programs are easily exploited.

**Do you know of any other ways to find this vulnerability?**

No

**What two questions do you have about stack smashing attacks that you can bring to class?**

1. How is it prevented?