CECS 378 Lab 3 - Buffer Overflow

60 points

Assignment Description. This assignment focuses on buffer overflow attacks and how they can be carried out on poorly-programmed system programs. You may not be able to complete this assignment on a modern operating system, as there are canaries built-in to modern shells (and kernels) to prevent such a thing from occurring. I would recommend using an older Linux distribution in a virtual machine for this assignment (something prior to Linux kernel version 2.4), but you are welcome to attempt this on a modern OS and see if you can get it to work. Review the article Smashing the Stack for Fun and Profit for a very good, detailed introduction on how to perform a stack smashing attack.

Assignment. Given the following C code file, perform a stack smash on the *vuln.c* code file using a C program that you create named *exploit.c*. Your program should attempt to open up a reverse shell on the attacked program as root by exploiting the buffer (you can verify this by typing the command 'whoami' on the resulting terminal). The *vuln.c* code must be compiled in its own, separate program and must not be altered from its original state.

```c
//vuln.c
#include <stdio.h>
#include <string.h>
int main(int argc, char **argv) {
   // Make some stack information
   char a[100], b[100], c[100], d[100];
   // Call the exploitable function
   exploitable(argv[1]);
   // Return: everything is OK
   return(0); }

int exploitable(char *arg) {
   // Make some stack space
   char buffer[10];
   // Now copy the buffer
   strcpy(buffer, arg);
   printf("The buffer says .. [%s/%p].\n", buffer, &buffer);
   // Return: everything fun
   return(0); }
```

Note: when running many versions of Linux, you may need to disable some address randomization.

Deliverables. Submit your *exploit.c* to Beachboard Dropbox (*no compression!*) along with a writeup of how you attempted the stack smashing attack and screenshots of the output or result of a successful attack. If you are not able to succeed with the attack due to OS constraints, detail that in your writeup and explain how you would go about performing such an attack on this system (along with your C code). Make sure that all code is **commented** with your own explanations or it will not be graded and you will receive zero points for the lab.