CECS 478 Lab 2 - Exploitation (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. This assigment will be difficult–but not impossible–to complete on a modern operating system, as there are canaries built-in to modern shells (and kernels) to prevent such a thing from occurring. Review the article Smashing the Stack for Fun and Profit for a very good, detailed introduction on how to perform a stack smashing attack. Your textbook also has an excellent tutorial in Chapter 3 on how to perform a buffer overflow exploit on a modern operating system.

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*. For full credit, 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. Don't start the exploit as root–use your own user account to do this). The *vuln.c* code must be compiled in its own, separate program and must not be altered from its original state. You might consider running your exploit using a Perl wrapper to inject the appropriate assembly code. See here for an example of running Perl within C.

```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 okay
   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: the fun stuff
   return(0); }
```

Note: when running Linux, you will probably 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. 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.