Security Engineering, Assignment 2

Due date: March 14, 2015

Introduction to Buffer Overflow Vulnerabilities (Maximum points: 40)

1. Write a shellcode in x86_64 assembly (nasm, GNU AS, etc.) which prints "Hello world!" on the screen and exits. You could use the shellcode posted on the course website for reference. [Hint: you would need to employ the write() system call to print messages.] Figure out what is the system call used for identifying the write() system call and how you could pass arguments via the registers. **Bonus points:** For scoring the bonus points you need to be able to handle the errors returned by write().
2. Write a C program which has a large buffer. The buffer should have the shellcode you wrote for (1) above. The return address on the stack should be modifiable such that it points to the beginning of the large buffer.
3. Function call flow diversion – Modify the following program in such a way that the function call flow in such a way that the call to function '`function1()`' should never return to the '`main()`' function. The call flow must be directed to '`target()`'. **Bonus points:** For scoring the bonus point, you need to supply input in the '`main()`' in such a way that the return from '`function1()`' is redirected to '`target()`'.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void function1(unsigned char *ptr)
{
  printf("Inside function1. Must not return to main()\n");
  printf("String:%s\n",ptr);

}


void target()
{
 printf("This is the target function\n");
 exit(0);
}


int main(int argc, unsigned char **argv)
{
 unsigned char buff[256];
 memset(buff,0,256);
 printf("Input string:");
 gets(buff);
 function1(buff);
 printf("back to main!\n");
 return 0;
}
```

What you are supposed to submit:

1. C and Assembly source code for the aforementioned programs.
2. Makefile through which one could compile these programs.
3. A short write up of what is the logic behind your shellcode, C program and the function call diversion.

How you would be graded:

1. Successful compilation using Makefile – 10 points.
2. Correct execution of the assembly shellcode. – 10 points. **Bonus points**: **5 if you are able to handle the errors when write() system call fails.**
3. Correct execution of the C program – 10 points.
4. Program flow diversion – 10 points. **Bonus points: 10 points if you are able to supply input via the** ‘`gets()`’ **function such that the flow is directed from** ‘`function1()`’ **to** ‘`target()`’.