

Assignment #5 - Exploiting Buffer Overflows  
CMPSC443 - Introduction to Computer and Network Security  
Spring 2009 - Prof. McDaniel

**Due date: May 7, 2009 – 8:00AM - NO LATE EXCEPTIONS**

In this assignment you will launch a buffer overflow attack on a target application. This requires you generate attack code and document the process. Making the latter clear and easy to understand is key to doing well in the project. Note that assignment requires a working knowledge of Intel hardware, assembly, and C programming. For this reason, you should allow an appropriate amount of time to complete the assignment.

1. **This assignment must be completed on a Linux operating system.** No exceptions. If you cannot locate a Linux system for your own use, you can obtain a virtual machine from Professor McDaniel via email.
2. You are to compromise the following program using a buffer overflow attack:

```
// Assignment #5: testme.c
// CMPSC443 - Spring 2009 (Professor McDaniel)

#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 );
}
```

You can grab this code from the website at:

<http://www.cse.psu.edu/~mcdaniel/cmpsc443-s09/docs/testme.c>

3. Exploiting buffer overflows is an subtle art. You should work from one of the better descriptions on how to do this given at:

<http://mixter.void.ru/exploit.txt>

I would recommend you read these instructions very carefully and try to use them as a cookbook for how to proceed. *You are are allowed to copy source code from this document to aid in this assignment.*

4. You are create a simple C program <yourname> that prints your name, the class name, and the current date and time to standard output. For example, if your last name is McDaniel, then the program should called mcdaniel.

5. You are to create a program `exploit` that calls the `testme` program using the `execlp` as shown in the example program in section 5a. You should copy the code from the example program and modify it to perform a buffer overflow operation. However, instead of calling `"/bin/sh"` (as in the document), you should call your program `<yourname>`.

**Note:** when running many versions of Linux, you need to use the `-fno-stack-protector` option when on the `gcc` compiler to prevent locally compiled programs from intercepting buffer overflows. If you see a "stack smashing detection" message when you try to overrun the buffer, the system has this feature enabled (you did not use the `compile` option, or did so improperly).

**Note:** when running many versions of Linux, you may need to disable some address randomization. See the following link for instructions:

<http://gcc.gnu.org/wiki/Randomization>

6. You are to create a document `README` that documents this process. You should include a description of how you determined the appropriate addresses and offsets, as well as the tools you used to perform the analysis. The document should give an annotated description of the stack at the point you launch the attack, as well as detail how the attack works.

**Note:** *Do not* simply regurgitate the text from the webpage referenced above, but provide detailed insight into how you approached assignment and how you performed it. The more detail, the better.

7. Create a gzipped tar file containing the commented code and `Makefile` for all the program, as well as the `README`. The tar file `<psu-id>-assign5.tgz` should contain a single directory `<lastname>-assign5` with all the submission files.

8. Attach the gzipped tar file to an email with the subject 'Assignment #5' and addressed to `mcdaniel@cse.psu.edu` and `buz107@psu.edu` by 8:00PM on the due date. **There will be no late projects accepted.**

**Note:** With the exception of the above exemptions, like all assignments in this class **you are prohibited from copying any content from the Internet or sharing ideas, code, configuration, text or anything else or getting help from anyone in or outside of the class.** Consulting online sources is acceptable, but under no circumstances should *anything* be copied. Failure to abide by this requirement will result dismissal from the class.