Assignment #2 - Exploiting Buffer Overflows   Deadline: 1st Feb 2024   [20Pt]

In this assignment you need to launch a buffer overflow attack on a target application. This requires you to generate attack code and document the process. Make the document clear and easy to understand which would be key to doing well in this work.

This assignment must be completed on a Linux operating system. Disable address randomization ($ sysctl –w kernel.randomize-va-space=0) if requires. Compile with No stack protection (-fno-stack-protector).

You are to compromise the following program using a buffer overflow attack:

```
// Assignment #1: testme.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 );
}
```

1. Exploiting buffer overflows is a subtle art which we discussed in class. Create a simple C program/ function <yourname> that prints your name, the class name, and the current date and time to standard output.
2. Create a program exploit that calls the testme program to perform a buffer overflow operation which should call your program <yourname>.
NB: 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.

3. 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.

NB: Do not simply regurgitate the text from the internet, but provide detailed insight into how you approached assignment and how you performed it. The more detail, the better.

4. upload the documented code and Makefile for all the programs, as well as the README/ Manual textfile.