Buffer Overflow Homework

## Server Denial of Service (5 pts)

* 1. **Download the server and client executables from** [http://www.cse.msu.edu/~cornwe19/cse825/](http://www.cse.msu.edu/~cornwe19/cse825/%20)
     1. Download instructions:

cd <your favorite directory>

wget www.cse.msu.edu/~cornwe19/cse825/server.out

wget www.cse.msu.edu/~cornwe19/cse825/client.out

chmod 755 ./server.out ./client.out

* + 1. These can be run in separate terminals by starting with the server in the first and sending messages to it via the client with:

client.out 127.0.0.1 <server port> “message”

* 1. **The server has a printf vulnerability** that can be exploited by the client. Find it and use it to crash the server. Attach a screenshot of both the client and server terminals during the crash to your homework submission.

## Spawning a shell (15 pts)

1. **Payload generation:**
   1. Download the buffer overflow payload assembly code at [shellspawn.s](http://www.cse.msu.edu/~cornwe19/cse825/shellspawn.s) and modify the comments of the code with answers to the questions it asks. Attach your answers to your homework submission.
2. **Spawning a shell:**
   1. Using byte code from the above assembly program (or the pre-provided byte code - [shellcode.txt](http://www.cse.msu.edu/~cornwe19/cse825/shellcode.txt)), spawn a shell from the vulnerable executable provided on our website.
      1. Provide the entirety of the input used to pull off the attack
      2. Provide a screen shot of the command line displaying the overflow happening
      3. Remember to turn off ASLR for your shell session:

sudo sh –c ‘echo 0 > /proc/sys/kernel/randomize\_va\_space’

* + 1. Hint: “\x90” represents a nop in shell hex-code
  1. *\*\*Note* that due to the debugging environment provided by GDB, it may be easier to spawn a shell from your exploitable program while debugging it. We will accept a shell spawned from GDB’s run time or from the console.

## Email homework submission to cornwe19@cse.msu.edu