Buffer Overflow Homework

1. Server Denial of Service (5 pts)

a. **Download the server and client executables from** http://www.cse.msu.edu/~cornwe19/cse825/

   i. 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
```

   ii. 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"
```

b. **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.

2. Spawning a shell (15 pts)

a. **Payload generation:**

   i. Download the buffer overflow payload assembly code at shellspawn.s and modify the comments of the code with answers to the questions it asks. Attach your answers to your homework submission.

b. **Spawning a shell:**

   i. Using byte code from the above assembly program (or the pre-provided byte code - shellcode.txt), spawn a shell from the vulnerable executable provided on our website.

      i. Provide the entirety of the input used to pull off the attack

      ii. Provide a screen shot of the command line displaying the overflow happening

      iii. Remember to turn off ASLR for your shell session:

```
sudo sh –c 'echo 0 > /proc/sys/kernel/randomize_va_space'
```

      iv. Hint: "`\x90`" represents a nop in shell hex-code

   ii. ***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