# John Carroll jcc0044/902521946

# COMP 5370/6370 ASSIGNMENT 4

You are provided with two socket programs in C. One of them acts as a server and the other as a client (they will be uploaded on the Canvas).

Your homework consists of testing whether the server is vulnerable to buffer overflow attack. If not, modify the server to create such a vulnerability. If yes, modify the server to eliminate the vulnerability.

* If you are using latest gcc version to compile, use ‘–fno-stack-protector’ option while compiling the server and client.
* Run server and client on two different shell terminals.
* Figure out if server is prone to buffer attacks.
* Modify the program as necessary and explain those modifications in detail.

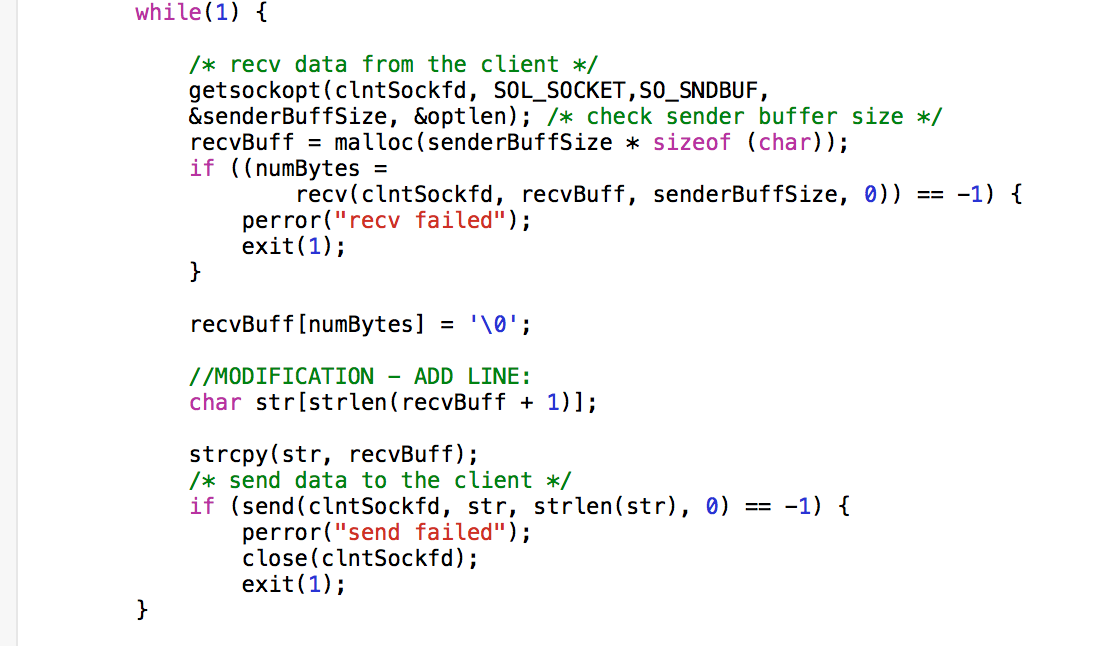
**NOTE**

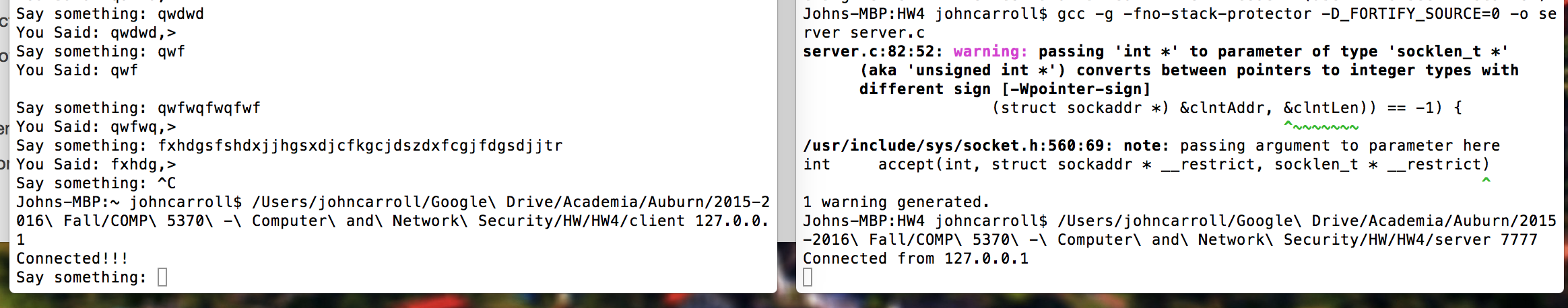
* Give screenshots of the working of your modified server and client.
* Highlight modified parts of code in a document.
* Your submission must contain
* Modified Server.c and client.c files
* Document that has screenshots of modified & working server.c and client.c
* Detailed explanation of the modifications you made.

**Submission**

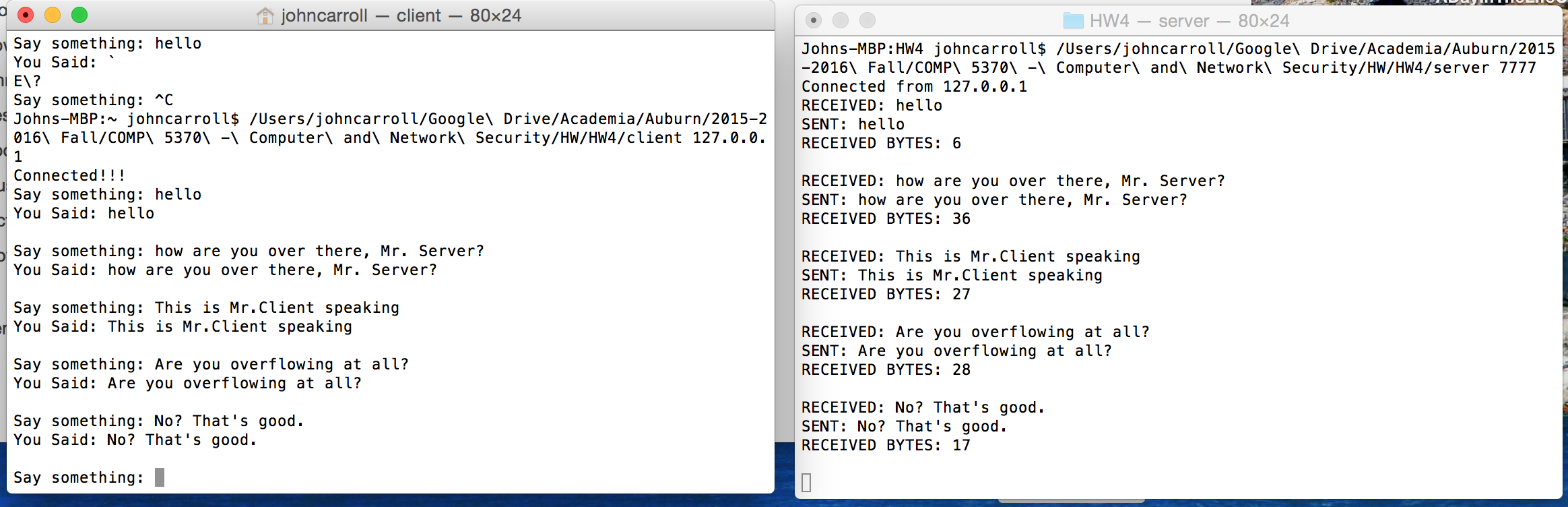
1. Modified server.c and client.c is in same zipped directory

Compiled using: gcc -g -fno-stack-protector -D\_FORTIFY\_SOURCE=0 -o server server.c  
Run using: ‘./server 7777’ and ‘./client 127.0.0.1’

1. This is the document containing screenshots of modified & working server.c and client.c  
   Modification within server.c is indicated with a comment shown below  
   Added code: char str[strlen(recvBuff +1)];  
   

Compilation Example:  


Working test run:



1. Yes the provided program has buffer overflow. Strncpy() combats buffer overflow by requiring you to put a length in it. Strcpy() depends on a trailing \0, which may not always occur. Strcpy() should generally be avoided in favor of strncpy(). (Note that "generally avoid" doesn't mean "never use"; it does mean don't use without CAREFULLY considering the circumstances.)

Using strncpy is helpful, but is not, in and of itself, a guarantee of safety. You still have to be careful about NULL termination issues when the src string is longer than, or equal in length to, the destination string. Use strlcpy if you can get away with it, but be careful of portability.   
With the modification overflow is fixed. This is due to resizing the char string array.