CS610 – Project #1

Overview:

This semester you will complete a semester-long programming project involving networking and security. I encourage you come up with your own project that interests you. If you do, you must have the project description and four deliverables (due dates for the deliverables are given in the syllabus and on Canvas) approved by me no later than the second week of classes.

In the past, some students have struggled to come up with a project and have asked me for a suggestion. For this reason, I provide a sample project with four deliverables to give you an idea of the size, scope, and content I expect from your project. If you don't want to come up with your own project and want to do the sample project that is fine – just let me know by the end of the second week of classes.

Two of the requirements I have for your project (besides that it have substantial networking and security components) are:

1. Your program(s) must run on stu.cs.jmu.edu (and, optionally, elsewhere).
2. You may use any programming language supported by stu (including, but not limited to: C/C++, Java, Perl, PHP, and Python) for your project.

Sample Project:

Create a Dropbox®-like client-server application (with a few security enhancements). Eventually, your application will support:

- Cloud storage of a user's files on the server
- Synching of files when a user logs in from multiple computers
- Multiple users
- File sharing among users
- Secure communication between clients and the server
- Strong encryption of all files before they are sent to the server so that if the server is compromised a user's files are not.

Description:

Develop a basic client and server program that communicate over the network. You do not need to implement most of the above functionality at this time so do not worry about authenticating users, multiple users, synching, or any type of security. However, give some thought to the functionality you will implement in the future and try to design your programs to be as robust and general as possible so that modifying them in subsequent projects will not require major changes to existing code.

The server should run on stu on a well-known port of your choice (Note: stu is inaccessible from off-campus except through SSH so if you want to be able to communicate with your server running on stu from an off-campus client you will need to setup a VPN connection to JMU). Your server must support two operations at this point:

1. A client should be able to upload any file to the server for storage
2. A client should be able to download any file stored on the server

If a user attempts to download a file that is not on the server then that is an error. If a user uploads a file with a name that is already on the server then the old file simply gets overwritten.

Deliverable #1:

Submit a design document and a tarred copy of your code using the "Project 1" link under "Assignments" in Canvas.

- Your design document should explain the high-level design of your programs. Make sure to describe any non-standard libraries you use and how you use them. You don't have to tell me about iostream, fstream, or any other standard programming libraries, but if you use socket, cryptographic, or other "specialized" libraries please describe them.
- Your tarfile should include a file named README that explains how to compile and run your programs, how to use the client, and any known bugs either contains. Leave your server running on stu until your project has been graded.