

Copyright © 2015 Luke Kazmierowicz

CS 311

12/07/15

Project Evaluation Report

Overall I think this project was very successful. The submitted version meets all the hard requirements and works reasonably well. I was able to send messages back and forth between classmates and the EchoServer. It was really cool to be able to send and receive messages based on an agreed upon protocol and I was surprised at how well it worked. Throughout my testing I discovered and fixed many potential issues and feel fairly confident in its ability to meet the requirements.

One known issue with my TauNet software is the rate at which it can receive messages. If two messages are sent to it at the same time there is a chance that it won't receive one of them. To the user who sent the message that wasn't received, it will appear as though my system was offline when they sent the message.

Another minor problem that currently exists involves determining if a TauNet user is available to receive a message. The issue occurs when the program checks to see if a user is available by sending them an empty message. Because there is currently no timeout, occasionally it will take a long time to determine if there is a connection or not and makes the user wait while it does so. I submitted an issue report for these known problems in GitHub so that any potential users will know that these issue exist and that the developer (myself) is aware of them.

What I learned from this Project

First off, this project was awesome. I was worried that it would be too difficult for me based on my very limited (basically none) experience with encryption, networking, and multithreading prior to taking this class. What I discovered as the term progressed was that I

really enjoyed learning these new skills. I didn't expect to learn stuff like that in a software engineering course so it was a total bonus to come out with not only experience designing software but building it too.

One of the places I felt that I did well in this project was prototyping. Because I did not know much about sockets, encryption or multithreading I didn't have any idea where to begin on designing a system. I realized that prototypes don't always have to be for the entire project. You can build prototypes for the subsystems as well. I started by implementing and testing the encryption algorithm separately from the rest of the project. Then I built a simple client and server program that allowed me to test opening a socket and sending a text message from one machine to another. Once I had the main components figured out I built a small multithreading prototype to get an idea of how I could use it to allow my program listen for messages all time.

At this point, aside from the encryption algorithm, I had no code that I was actually planning on using in my final project. Even so, I still felt like I was almost finished. The prototypes had helped me to learn how each part of the system was going to work before actually building them. This allowed me to create an intelligent design that leveraged that knowledge.

Another big thing that I learned while working on this project was the power of Git. I had used Git before but it wasn't until this project that I realized how helpful it can be if used correctly. Towards the beginning of the implementation process I found an issue with my CipherSaber2 algorithm. I was convinced that my prior test had proven its correctness but I had deleted the tests since then. I had made two big mistakes. The first was to create a temporary test and not a permanent one that I could refer to later. The second was that it had been over a week and many hours of work since my last commit.

From that moment on I committed my changes after each session. This automatically gave me history of what I worked on, when, and exactly what I changes I made. I honestly don't know how I ever programmed without using Git.

Overall I think this project went very well. If the goal was to learn something and create something useful while doing so, I definitely think I succeeded in both!