Nathan Reed
Project Evaluation Report
TauNet Communication Network v.1.2
CS300 Fall 2015
Dec. 3, 2015

1. PROJECT

The TauNet Communication project was a joint design effort of Bart Massey's CS300 software engineering course. The project was directed by Bart Massey and was an exercise in the process of software development: gathering requirements, creating a project protocol and requirements specification, editing the specification, and the various phases of design, testing and implementation. I conclude that my TauNet implementation accomplishes the minimum requirements as outlined in the requirement specification and not more.

2. OBJECTIVES

The major aim of the TauNet project was to develop a platform on which users could send encrypted messages to each other over a public internet connection. The protocol specified a TCP connection to be used along with the cipher saber 2 encryption method which is a version of arc four encryption.

3. METHODS AND LIMITATIONS

I will stick to a very general description of challenges I faced in the implementation and save the details of the implementation for the Software Design Document.

I chose Python3 as the language for this project because rumor had it that it had a good socket library. Had I used Python before, I might have chosen version 2 since it is more ubiquitous, but I figured I might as well use the latest version. While the extensive libraries and the high level make Python easy to use they also made some tasks challenging. I started downloading scapy because I thought it would make things easier, but learned that scapy was really intended for more advanced operations than the scope of this project and required basic knowledge of sockets which I didn't have. The socket library made the connecting part of the problem not too bad. Luckily there were many good examples and instructional videos online that I could work on to get a basic idea of how sockets work and how to implement them. On the encryption it took some time to figure out that byte to string conversions and back are not a thing. So I ended up working with bytes exclusively. Byte arrays were a useful convention as well as byte objects. This is one part of the program where I could see a c implementation being more intuitive. This is also due to my lack of Python prowess.

I borrowed all of my algorithms for the encryption and decryption from Bart Massey's cipher saber 2 repository on github. The socket code came from a variety of sources including python docs, youtube instructional videos and debugging help from class mates and Bart. The sockets were not so easy to wrap my head around and parts of it still seem a bit magical.

4. SUCCESSES AND THINGS THAT COULD BE BETTER

In the end, I was able to send and receive messages both to the echo server and to my classmates. When I send a message the program enters a private dialogue of sorts. When the user hits the enter key they exit and return to the main menu. This is nice but I did it a bit crudely. If a user is having a dialogue any other person can interrupt. To leave one conversation and move to another the user needs to hit the enter key and select the new person they want to chat with. It's a bit hackey but with a little practice the design really isn't so bad to use. If I had more time I would have made this feature a bit more robust but I decided to mess with it late in the project and have other things to get done. Overall, I think it meets or exceeds the minimum viable product. And I haven't gotten the latest version to crash yet!

5. CONCLUSION

Given more time it would have been nice to add additional features. By the time people started talking about a time stamp and looking for in-network users who were listening, I had done all of my testing and did not want to add any more to the project. This project is finished and additional features can be deferred till the next version.

This project was a good learning experience. I got some good exposure to sockets and building a software project from the ground up. I also learned about the need for security. My pi was owned by some malicious agent of doom because I never changed it from the default password I believe. This was a bad idea. Luckily everything was backed up on my home machine so it wasn't that big of a deal. So I also learned that valuable lesson.c ..