TAUNET TEST PLAN

1. INTRODUCTION

1.1. Overview

TauNet is a simple, encrypted messaging application. On the client side it is able to view a .csv file consisting of contacts and let the user select one of those contacts. It then makes a socket connection with the user selected contact and allows the user to input a message to send to that contact. TauNet then encrypts the message and sends it to the connected contact. On the server side, TauNet will listen for incoming socket connections, accept an incoming connection, and then accept the incoming byte stream. It will decrypt the byte stream and save it to a text file that is also accessible by the client for viewing. In addition to that it will write connection information to a server log text file.

2. SCOPE

This Document is intended to illustrate how TauNet will be tested.

3. TESTING STRATEGY

3.1. Overview

The basic strategy is as follows:

1. Verify functionality of all available units by first using them as intended.
    a. Select all menu options and verify they return the expected result.
    b. Make a connection with the echo server and verify that the protocol is met, verify that the encryption worked as intended, and verify that the returned message is properly accepted and saved.
2. If intended input does not produce desired output, fix it.
3. Once intended use is verified, feed the application erroneous input.
    a. Select menu options that do not exist.
    b. Send messages that do not meet protocol to an instance of the program.
4. Implement error checking and exception handling wherever bad input causes failures or crashes.

3.2. Unit Testing

1. A simple client and server were the first units to be built. They were tested by having the client request a socket connection with the server, then sending messages to verify basic functionality.
2. Further units were tested with the manual regression testing basic strategy outlined.

3.3. Manual Regression Testing

The basic testing strategy must be repeated after every new atomic unit is added to ensure overall functionality is valid.

3.4. User Acceptance Testing

Allow non-tech savvy users to test the system in order to verify usability.

4. FEATURES TO BE TESTED

1. Sending a message.
2. Viewing messages
3. Listing/Viewing contacts
4. Selecting contact
5. Receiving messages.
6. Encryption.

5. FEATURES NOT TO BE TESTED

5.1. Multiple simultaneous connections.

6. DEPENDENCIES

6.1. Echo Server

6.2. Classmates Nodes