

Software Test Plan

Version 0.1

December 8, 2015

TauNet

Copyright © 2015 Manpreet Bahl

Submitted in the partial fulfillment of the
requirements of CS 300 Software Engineering

Table of Contents

1.0)Introduction	3
1.1)Purpose.....	3
1.2)Scope	3
1.3)References	3
2.0)List of Features to be Tested	3
2.1)User Interface.....	3
2.2)TCP Client	4
2.3)TCP Server	4
2.4)RC4 Encryption and Decryption	4
2.5)Linear Linked List (Address Book) Features: Add, Remove, Display, Search, and Count	4
2.6)Loading contacts to Linear Linked List from CSV File	5
3.0)Test Schedule	5
4.0)Risks.....	5
4.1)Privacy Risk	5
4.2)Memory and Performance Risk.....	5

1.0) Introduction

1.1) Purpose

The purpose of this document is to outline all testing strategies to be used for TauNet. It will outline what features need to be tested and how they will be tested.

1.2) Scope

This document will mainly focus on testing the user interface, TCP (client side and server side), CipherSaber 2, address book data structure (linear linked list), and file reading/writing. Some uses of the file reading are related to the address book data structure, while the remaining features are not. This will be explicitly stated in the appropriate section.

1.3) References

1) TauNet Echo Server Implemented by Bart Massey:

<https://github.com/PSU-CS-300-Fall2015/taunet-utils>

2) CipherSaber 2 Implemented by Bart Massey:

<https://github.com/BartMassey/ciphersaber2>

3) TauNet Communications Protocol v0.2 (revision 1)

4) Test Plan Template:

<http://www.softwaretestingclass.com/wp-content/uploads/2013/01/Sample-Test-Plan-Template.pdf>

2.0) List of Features to be Tested

- 1) User Interface
- 2) TCP Client
- 3) TCP Server
- 4) RC4 Encryption and Decryption
- 5) Linear Linked List (Address Book) Features: Add, Remove, Display, Search, and Count
- 6) Loading contacts to Linear Linked List from CSV file

2.1) User Interface

The test plan for the user interface is to allow other developers, colleagues, and users run the program and test all the features of the menu. All testers will be

encouraged to enter wrong input whenever they choose as to test the error handling of the user interface.

2.2) TCP Client

The test plan for testing the functionality of the client side of TCP is to attempt to connect it with a server implemented by other TauNet developers and the echo server implemented by Bart Massey (Reference 1).

2.3) TCP Server

The test plan for testing the functionality of the server side of TCP is to have other TauNet developers attempt to connect to the server. An additional plan is to use the echo server implemented by Bart Massey (Reference 1) which will essentially by sending a message to the echo server and then receive the echo using the TCP server.

2.4) RC4 Encryption and Decryption

The test plan for RC4 encryption is to have the encrypted data displayed in the same format as the CipherSaber 2 implemented by Bart Massey in Haskell (Reference 2). If the hexadecimals are the same, then the test is successful. The test plan for RC4 decryption is to decrypt the data and see if the message obtained was the same as sent. The echo server implemented by Bart Massey (Reference 1) will also be used to test whether the RC4 implemented works as intended to. If the echo server returns the same message as it was sent, then RC4 encryption and decryption works as it should.

2.5) Linear Linked List (Address Book) Features: Add, Remove, Display, Search, and Count

The test plan for the Linear Linked List features is to implement each feature and then test them thoroughly.

Add: The plan is to add a contact to the list in alphabetical order. This will be verified after adding a node and then displaying the linked list.

Remove: The plan is to remove a contact from the list and then display the linked list to check whether the contact was removed or not.

Display: The plan is to test whether the linked list is being displayed correctly by utilizing the add feature and adding some contacts to the linked list in order to test the linked list.

Search: The plan is to give the feature a name to search the linked list and then display, with a test main, the contact name, IP, and port number. Note, that when testing is done, search will not display the contact's IP and port number, but instead pass them to other functions that depend on the information

Count: The plan is to have a linear linked list with a known number of nodes and then test this feature by using a test main and display the number of nodes in the linked list. If the number matches, then the feature is working. Various numbers of nodes will be tested to ensure that everything is working

2.6) Loading contacts to Linear Linked List from CSV file

The test plan is to utilize the linear linked list display feature and add feature. First, the data from the CSV file will be added to the linked list using the add feature. The display feature will be used to verify if the data was read in properly by displaying all the information.

3.0) Test Schedule

There is no specific test schedule as implementations will be tested thoroughly, following the test plans stated in Section 2.0.

4.0) Risks

3.1) Privacy Risk

There is a privacy risk where the message history from previous sessions is stored in a CSV file and can be viewed by anyone access to the user's device. The current solution is to delete the CSV file by running the program again after it has terminated and selection option 4, Delete Message History (Previous Sessions).

3.2) Memory and Performance Risk

The performance of TauNet is dependent on the number of contacts since the data structure used is a linear linked list. Currently, the number of contacts is bounded by minimum of 12 contacts and maximum of 300 contacts. However, the bounds are removed and the number of contacts is large, then there could be slowness of the program. All features of the address book are dependent on traversing through the linked list from the beginning.