Software Test Plan

TauNet Messaging System

Date: 05 December 2015

Prepared By: RJ Russell

TABLE OF CONTENTS

1 INTRODUCTION

1.1 PURPOSE

The purpose of this document is to outline a test plan that will effectively test each piece of implementation for the TauNet messaging system.

1.2 SCOPE

This document will cover the major components of the TauNet messaging system. It will not cover testing the connections, nor will it cover proper Raspberry Pi (or other Linux platform) setup.

The following is a breakdown of the components and the test plans in place.

2 RC4 / CIPHERSABER2

Test	Test Method
Ensure zero based list is initialized	Initialize and display the list
Ensure no errors after key scheduling and key stream coding is finished as per algorithm	Run program. Pass in a string with ord mapped to it, and a key with ord mapped to it. Display result.
Ensure decrypt can take in a message, pull the IV off and append to the key.	Run code and ensure no errors.
Ensure decrypt can decode.	Read in a test file, pass it to the rc4 method and display result.
Ensure encrypt method can generate 10 random bytes and append to key.	Run code, and ensure there are no errors. Display result.
Test encrypt and decrypt together	Pass a message to encrypt and catch the returned result. Take the returned result and pass to decrypt. Result should be original message.
Test decrypt with all other test files.	Read in all test files. Change rounds and key to the appropriate values based on test file. Display results.
Test encrypt/decrypt with other TauNet users	After server/client is active. Ensure messages sent/received to other users are encrypted/decrypted properly.

3 SERVER

Test	Test Method
Ensure socket calls are made properly	Test by inspection.
Ensure server is properly listening on the correct port.	Using netstat –tlupn verify the correct port is listening.
Test message can be received	Using telnet, connect to server address and send message to server.
Ensure when encrypted message is sent to the server, it is being decrypted and displayed properly.	Send encrypted message to server. Make sure message displays properly.
Ensure when server is exited with Cntrl-C that the error handling catches it and displays appropriate message.	Exit server with Cntrl-C. Inspect to make sure proper exit is displayed and no errors are displayed.

4 CLIENT

Test	Test Method
Ensure socket calls are made properly	Test by inspection
Ensure client can send a message	Hard code message and send to server (with encryption/decryption turned off)
Ensure socket timeout and try/catch block display appropriate messages.	Enter address that is offline, wait to see if the except block catches the timeout and displays appropriately.
Ensure client can properly send encrypted message	Encrypt message and send to server and ensure the message is properly decrypted
Ensure socket closes properly	Display socket close message when closing sockets.

5 CLIENT INTERFACE

Test	Test Method
Ensure address book is read in properly	Read in address book and display it
Ensure recipient can be chosen	Prompt and choose recipient, display the recipient.
Ensure address is properly assigned.	Choose recipient and display the name and address. Verify by inspection
Ensure message is read in and header is appended.	Prompt for message, and display result after header is appended.
Ensure all flags work properly	Enter in the flag, and verify correct behavior.
Ensure program loops correctly until user exits	Run program repeatedly ensuring program behaves properly.
Ensure message can be encrypted and sent	Send message to server, verify correctness.
Ensure program exits properly	Exit program and check for any errors.

6 CONCLUSION

Ensure each test is conducted appropriately. Fix any errors or program behaviors that are inadequate to program performance.