TauNet Software Design Document

Version 1.0

November 11, 2015

TauNet v1

Rachael Johnson

Table of Contents

1.0 TauNet General Description Overview

TauNet is a program that allows multiple TauNet nodes to securely communicate with each other over the internet. There is one user per node, and one address per node.

1.1 TauNet Design Overview

The TauNet program will be divided into several different modules. TauNet will be written in Python, for its ease of use and handling of network protocols and strings. The python program will be broken into several different files for modularity.

2.0 User Interface

2.1 Main Screen

The main screen has three options for the user to choose:

- Send Message
- View Message
- Quit

2.1.1 Send Message

This will open the address book (see 2.2). After the user chooses an recipient, the user has to type a message. After the user types a message and presses “enter”, the message is sent to the address for the recipient that the user chose.

2.1.2 View Message

When the user selects view message, TauNet loads all messages that were stored from the server module. The messages displayed are according to protocol. They will be listed first received to last received. After the messages are listed the user is returned to the main menu.

2.1.3 Quit

When the user chooses to quit, the program displays a message “Program Quitting…” and the exits.

2.2 Address Book

The address book populates from a text file. The text file contains one line per recipient. Each line contains the recipient's username, followed by a space, followed by the recipient's address. The address can be a web address or an ip address.

When the user chooses to send a message, the address book displays a list of of recipients, each numbered sequentially. The user then types in the number of the recipient they wish to send the message too, and then the address book sends that information to the main program.

2.3 Log

When the user tries to send a message to a TauNet node that is offline, then TauNet will produce error code 111 and TauNet will store the message that was supposed to be sent in a log, along with the intended recipient and address.

When the user selects Log from the main menu, all messages that the user tried to send (but failed to be received by the recipient) are displayed, number sequently. Once the user selects a message, a prompt appears asking if the user wants to:

1. Send Message
2. Delete Message
3. Cancel

If the user selects send message, the message is sent and removed from the log. If the user selects Delete Message, the message is not sent, but is removed from the log. If the user selects Cancel, the log is not changed and the user is returned to the main menu.

3.0 TauNet file structure

3.1 TauNet.py

- Handles main screen
- starts server
    - server is a separate thread
- starts client to send a message
- exits program when user chooses to quit
- handles address book

3.2 server.py

- opens a port and starts listening
- once it receives a connection it opens a new thread for that connection
- receives 1024 bytes of data and then closes connection

3.3 client.py

- Opens a socket and sends the data passed to it to an address that was also passed to it

3.4 protocol.py

- contains the rs4/ciphersaber2 encryption algorithm
- contains encrypt function
    - returns a byte string
- contains a decrypt function
    - returns a string

3.5 ReadMe.md

- Describes project for broader audience