

TauNet System Requirements Specification

Abstract

This document describes the hardware and software requirements for the TauNet system.

Status of This Memo

This document is provided as part of the requirements for TauNet, as specified in Portland State University CS 300 Section 1 Fall 2015.

Copyright Notice

Copyright (c) 2015.

1. Introduction

TauNet is a project conceived as a class project for PSU CS 300 Section 1 Fall 2015. It is intended to allow a pre-defined network of self-contained communication nodes to communicate amongst themselves in a fashion secure against outside observation.

1.1 Purpose

The purpose of this document is to define the hardware and software requirements of the TauNet system.

1.2 Scope

The scope of this project is limited to the development of the TauNet software. TauNet shall allow encrypted text messages to be sent between clients. The TauNet software shall conform to the specification in this document.

2. Overall Description

2.1 Hardware Environment

The TauNet system shall be engineered to operate on the Raspberry Pi platform.

2.2 Software Environment

The TauNet system shall be engineered to run on the Raspbian distribution of the linux operating system.

3 Functional Requirements

The TauNet system shall encrypt messages before transmission. Messages shall conform to the protocol defined in the TauNet Communications Protocol v0.1. The system shall store a cryptographic key. The system shall store a list of usernames and IP addresses.

3.1 System Features

3.1.1 Messaging

The system shall support messaging between clients.

3.1.2 Encryption

The system shall encrypt messages in accordance with the TauNet Communications Protocol v0.1.

3.1.3 Persistent Destinations and Key

The system shall maintain copies of the destination list and cryptographic keys between uses.

3.2 Use Cases

3.2.1 Messaging

Description	User A messages User B
Actors	User A, User B
Preconditions	<ul style="list-style-type: none">• User A and User B have exchanged IPs and the cryptographic key• User B is listening for communications
Steps	<ol style="list-style-type: none">1. User A starts the TauNet messenger<ol style="list-style-type: none">a. The System displays the main menu2. User A selects 'Set Destination' on the main menu<ol style="list-style-type: none">a. The System displays an enumerated list of destinations3. User A selects User B from the destination list<ol style="list-style-type: none">a. The System sets the destinationb. The System displays the main menu4. User A selects 'Start Messenger'<ol style="list-style-type: none">a. The System displays the messenger

	<ol style="list-style-type: none"> 5. User A enters a message and hits return <ol style="list-style-type: none"> a. The System encrypts the message b. The System sends the message to User B 6. User B receives the message
Postconditions	<ul style="list-style-type: none"> • User B has received a message

3.2.2 Starting the TauNet messenger without going to the main menu

Description	User A wants to start TauNet messenger foregoing the main menu
Actors	User A
Preconditions	<ul style="list-style-type: none"> • TauNet is not running
Steps	<ol style="list-style-type: none"> 1. User A starts TauNet with the -msg flag <ol style="list-style-type: none"> a. The System starts the messenger directly using the settings from the previous session
Postconditions	<ul style="list-style-type: none"> • The TauNet messenger is running

4 Non-Functional Requirements

4.1 Usability

The system shall be usable with basic linux experience.

4.2 Maintainability

The software shall be implemented in a maintainable manner (object oriented).

4.3 Scalability

The system shall scale to several hundred users per TauNet.