Portland State University CS 300 Section 1 Fall 2015
22 November 2015

TauNet Communications Protocol v0.2 (revision 1)

Abstract

This document describes a protocol layered atop TCP/IP for the exchange of encrypted messages between TauNet nodes.

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. While it is formatted in vague accordance with IETF RFC format, it is currently in no way associated with the RFC process.

Copyright Notice

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 (Raspberry Pis in the initial implementation, although any sort of node might be used) to communicate amongst themselves in a fashion secure against outside observation. The TauNet protocol described in this document shall be used for message interchange between communication nodes.

This document describes only the TauNet protocol. Issues of key distribution, etc. are outside the scope of this work.

1.1 Definitions

TCP: A standard Internet stream communication protocol, layered atop IP, a standard datagram protocol. Description of this protocol and its usage is outside the scope of this document. See e.g. the book TCP/IP Illustrated, Vol. 1: The Protocols by Richard Stevens for a tutorial introduction.

IP Address: A standard Internet host endpoint address as used by TCP. An IPv4 address is the format used by version 4 of TCP.

NAT: Network Address Translation is a technique used by firewalls to allow hosts inside a network to share an Internet-facing IP Address.

IETF: The Internet Engineering Task Force, the governing body of the Internet.

RFC: An IETF “Request For Comments” document.

2. Assumptionsn

Messages shall be sent and received using the TCP protocol. A single encryption key shall have been distributed securely to every TauNet node in advance of use. Every TauNet user shall have been pre-assigned a username consisting of 3-30 seven-bit ASCII characters, each either an uppercase letter, a lowercase letter, a digit or a dash symbol “-”. In addition, a table will have been distributed to every TauNet node: this table shall contain a username and either a corresponding IPv4 address or a fully domain-qualified hostname for every TauNet node on the network. A standard table for the TauNet project is being maintained at https://docs.google.com/a/pdx.edu/spreadsheets/d/1wejIg4qrmQGQPMKZIV2pQrmhsNSX1GI5db6-7GW1xkE. All TauNet IP addresses will be statically routable (i.e., not NAT) and will be capable of TCP connection on port 6283 (τ to four decimal places).

3. Protocol

The TauNet protocol is a layered protocol. A plaintext message shall be marked with header information: the header and message body shall be encrypted using RC4 encryption, and the resulting ciphertext sent to its destination via a TCP connection created for this purpose and closed upon completion.

As a special case, a message of length 0 is a valid TauNet message (used to identify open connections) and shall be immediately discarded by the recipient.

3.1. Encryption

Messages will be encrypted using CipherSaber-2 as described by the document at http://github.com/BartMassey/ciphersaber2. Exactly 20 rounds of key scheduling will be used. (The number of rounds and the key scheduling algorithm is a change from v0.1 of the TauNet protocol.)

3.2 Message Format

Each TauNet message shall begin with a header section. Each header will consist of an identifying keyword, a colon, a single space, and the header payload. There are several standard headers, all shown here, which shall all appear in the order given:

version: 0.2
from: sender’s TauNet username
to: receiver's TauNet username

The header names shall be sent in all lowercase. The ASCII characters CR and LF (code 13 and code 10) shall be used as a line ending indicator. Headers other than those shown shall not be used. (There is currently no “date” header, but one is likely to be added in a later protocol version.)

The end of the header section will be marked by a blank line (containing no characters). Following this, an arbitrary byte sequence of message characters may be transmitted. The message’s end will be marked by the termination of the TCP stream carrying the message. Sent messages must be no more than 1024 bytes, including headers. (Since a maximum-size header is 90 characters, a maximum message body length of 934 bytes is implied.) Receivers must be able to process messages up to this length, and may process larger messages.

A sample message:

version: 0.2
from: anne
to: bill

Hello neighbor

Notes

Old version: https://github.com/JVMartin/cs300-protocol

https://docs.google.com/document/d/1juKX1KE8FnpVBpb9S6RHwLm2DpCJv0RnuKHOfOK-h7Y/edit