Working Group Draft for TCPCLv4

Brian Sipos
RKF Engineering Solutions
IETF99
Overview

• Background
• Discussion of current state
• Way forward for TCPCL and BPbis
Motivations for Updates to TCPCL

1. During implementation of TCPCLv3, Scott Burleigh found an ambiguity in bundle acknowledgment and refusal.

2. For use in a terrestrial WAN, author has a need for TLS-based authentication and integrity. TCPCLv3 mentions TLS but does not specify its use.

3. Reduced sequencing variability from TCPCLv3

4. Allow an endpoint to positively reject a message (rather than simply ignoring it).
Goals for TCPCLv4

• Do not change scope or workflow of TCPCL!
  ◦ As much as possible, keep existing requirements and behaviors. The baseline spec was a copy-paste of TCPCLv3.
  ◦ Still using single-phase contact negotiation, re-using existing headers and message type codes.
  ◦ Allow existing implementations to be adapted for TCPCLv4.

• Re-use existing encoding, type and reason codes.
  ◦ New IANA registries are requested but where purpose is identical to TCPCLv3 the cde is re-used.
  ◦ Since workflow is preserved, majority of message types are retained.
Last Draft Edits

• Last questions resolved on the TCP CL:
  ◦ TLS is now mandatory-to-implement but optional-to-use at session time
  ◦ Separate IANA registries defined for TCPCLv4
    ◦ Avoids naming confusion, makes changes more obvious
  ◦ TCP CL contact header now supports protocol extensions, but does not define any currently
    ◦ Uses fixed-octet-width fields in TLV-type structure
  ◦ Edited message encoding to use octet-aligned fields
    ◦ Avoids bit-packing for message header content
    ◦ Avoids reserving flag bits for messages which do not use them
Way Forward for TCPCLv4

• Current specification draft is complete
  ◦ No new comments have been received to-date
  ◦ This means no issues but also no concurrence

• Working implementation exists and is available for interoperability testing
  ◦ Updated to current I-D content
  ◦ Implemented in scapy/python for ease of understanding
  ◦ Handles concurrent sessions
  ◦ Does not implement BP agent behavior, only CL behavior