TCPINC & Framing protocol

Tero Kivinen
IETF 92, TCPINC
2015-03-26

Reasons

- As we are not protecting TCP header bits, to maximize the compatibility with middleboxes, do as little changes to the outer TCP as possible.
  - We most likely need to do some kind of TCPINC negotiation using TCP options during the connection establishment phase
  - After that move everything inside the tcp stream, so middleboxes cannot mess up the things that easily.

TLV protocol

- i.e. add TLV style protocol to be run inside the tcp stream:
  - Type, length, data
    - With data being encrypted and maced after key exchange is finished.
  - The actual format of the TLV protocol depends on final tcpinc protocol.

Features needed

- Ability to do some kind of key agreement / establishment at first.
- Encapsulate the real tcp stream and encrypt and MAC the tcp stream.
- Implementations can try to keep the tcp segments and framing protocol packets in sync
  - But middleboxes can mess up with this by splitting or merging the tcp segments, so needs to work even if not staying in sync.

Open Issues

- Do we need to replicate some of the tcp features inside the framing protocol.
  - Most functionality does not matter, as using outer tcp header is enough when no active attackers present.
  - Some are more problematic, like urgent pointer, as now we have some extra stuff inside the tcp stream, so what does urgent pointer mean.
    - Just make urgent data separate framing protocol record, and put the length of that (including overhead) to the urgent pointer (i.e. it points to start of next record).
    - Do the same but use separate record type for urgent data, i.e. urgent pointer value outside does not really matter.
    - Or we can just ignore the urgent data issue.