DTLS over SCTP

draft-ietf-tsvwg-dtls-over-sctp-bis-05

Magnus Westerlund
Claudio Porfiri
John Preuß Mattsson
Overview

- Status
- Open Issues
  1. Dealing with SCTP-AUTH limitations
  2. DTLS 1.3
  3. DTLS messages demultiplexing vs DTLS records containing user messages
- Next Steps
Status

- The latest version -05 addresses a lot of Martin Thomson’s review comments
- Still a small number of issues remain beyond what will be discussed today
  - Some are highly dependent on which DTLS versions to support
Dependency on SCTP-AUTH

- [https://github.com/gloinul/draft-westerlund-.tsvwg-dtls-over-sctp-bis/issues/183](https://github.com/gloinul/draft-westerlund-.tsvwg-dtls-over-sctp-bis/issues/183)
- Like RFC 6083 this document is depending on SCTP-AUTH for certain security services
  - Ensure no replay, verify authenticity of DTLS records part of a user message to ensure user message integrity
  - Any replay or insertion of DATA chunks will result in either:
    - A DTLS integrity failure resulting in the DTLS record being discarded
      - Leading to SCTP Association closure per our specified rules as user message integrity has failed
    - A successful replay of a complete DTLS Record would result in a undetected user message corruption
      - SCTP Association availability failure due to other chunks types being replayed?
- Conclusion: Replay or reflection must be prevented to the probability levels the crypto can provide
SCTP-AUTH Mitigations

- Relection Attack
  - Require directional SCTP-AUTH keys
    - We can specify how to derive directional keys
    - Changes SCTP-AUTH implementations to support directional keys including APIs
- Replay Attack
  - Require that SCTP-AUTH keys have been retired before $2^{32}$ TSN have been used

- Are these mitigations sufficient and implementable?
SCTP-AUTH Next Steps

● DTLS over SCTP is complex and has a lot of corner cases just to avoid direct SCTP implementation impact
  – For better security and likely simpler solution should we look at alternative as we will have impact on the SCTP implementation anyway?
  – We authors are willing to draft an alternative solution for consideration by the WG

● Please perform your own security analysis to determine your view
  – Are suggested mitigation or additional that you may propose sufficient?
  – Do we need to find an alternative solution?
DTLS 1.3 Only

- We raised the question on the mailing list about requiring supporting DTLS 1.3 exclusively
- Benefits
  - Improved interoperability with only one DTLS version
  - Better security with the subsetting of ciphers and no possibilities to encounter DTLS 1.2 weakness
  - Lesser specification work to address DTLS 1.2 requirements further
- Potential downsides
  - Question about availability of DTLS 1.3 implementation including required functionalities
    - Connection ID
    - Support for RFC 8449 for negotiating record size or full 16k DTLS records
DTLS 1.3 Availability

- The availability of DTLS 1.3 stacks is more limited than DTLS 1.2 so far. Some examples:
  - One available stack we know of: WolfSSL (announced beta)
  - Working on it: Mozilla NSS (No Connection IDs)
- However it is the additional requirements that makes it hard to find implementations
  - Connection ID
  - Turning off replay
  - Large record sizes and/or RFC 8449 negotiation of maximum record size

- Additional Input?
- Accept that we will have to support DTLS 1.2?
DTLS Message Demultiplexing

- https://github.com/gloinul/draft-westerlund-tsvwg-dtls-over-sctp-bis/issues/139

- During a DTLS connection, some DTLS messages not containing protected application data are sent
  - Handshake
  - Errors
  - Close Notify

- DTLS expects them to be sent in order as they are produced
  - RFC 6083 required in order delivery on stream 0
  - We change this to any stream any user message, including interleaved with data

- Martin Thomson raised some potential issues with this:
  - Hard to optimize DTLS implementation for record processing and internal protocol messages in separate paths
DTLS Message Demultiplexing

- One issue is that DTLS/SCTP adaptation layer can’t identify these for DTLS 1.3
  - No content type in plain text
- Thus, the DTLS stack must process and may consume it as it is the target
  Likely results in some DTLS stack API output
- Two directions:
  - Keep them hard to identify, but possibly negatively impact processing
  - Make them identifiable and put them in their own user messages in order
    - Security implication that the handshake for example can be targeted by on-path attacker
- One alternative for identification is using a PPID
● Timeline for completing is delayed as either of these need to be done:
  – SCTP-AUTH fix
  – Alternative DTLS solution

● Target updated drafts before new year