Recommendations for Implementing IPFIX over DTLS

draft-mentz-ipfix-dtls-recommendations-00

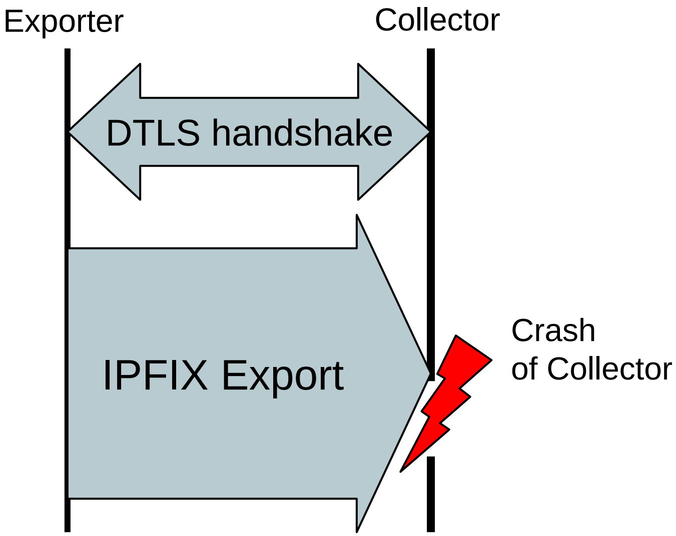
Daniel Mentz, Gerhard Münz, Lothar Braun

75th IETF Meeting, Stockholm, 2009

Background

- RFC 5101:
  - support of DTLS mandatory for IPFIX-over-SCTP and IPFIX-over-UDP for **security reasons**
- *draft-muenz-ipfix-compression-00* presented in Dublin (July 2008):
  - IPFIX Messages are highly compressible
  - DTLS provides built-in support for negotiation and use of **compression** algorithms ➔ no changes to IPFIX required
- Currently implementing DTLS support for VERMONT
  - http://vermont.berlios.de/
  - based on OpenSSL and patches of Michael Tüxen and Robin Seggelmann http://sctp.fh-muenster.de/dtls-patches.html
- Not only many implementation problems (bugs, missing features), but also open questions how to handle specific situations…

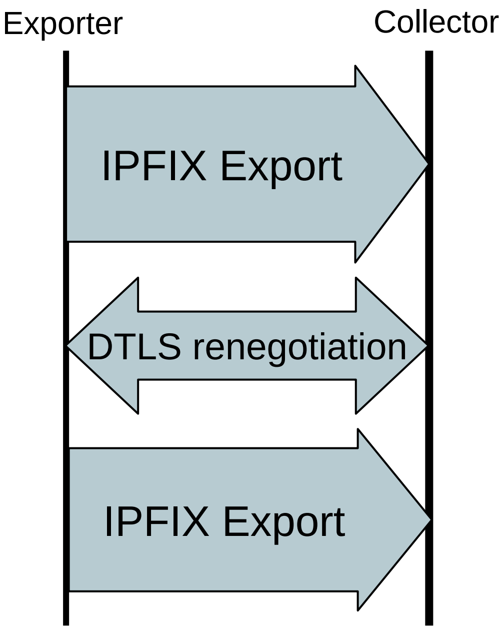
Problem with IPFIX-over-DTLS/UDP



- **Missing *“dead peer detection”***
  - Exporter unable to detect a crash of the Collector because IPFIX traffic is unidirectional
  - After reboot, Collector cannot decrypt/verify incoming IPFIX Messages due to lost DTLS state
- Possible solutions
  - **Exporter periodically initiates DTLS renegotiations**
    - if Collector does not respond, try to open new DTLS/UDP Transport Session
    - renegotiation is computationally complex and usually requires interruption of IPFIX export
  - **Exporter periodically opens new DTLS/UDP Transport Session to Collector**
    - “soft hand-off” of IPFIX export to new Transport Session after DTLS handshake is completed and Templates have been sent
    - in our opinion, best solution available today
  - Maybe available in the future: **DTLS Heartbeat Extension**
    - draft-seggelmann-tls-dtls-heartbeat-00 (July 2009)

Problem with IPFIX-over-DTLS/SCTP

- **DTLS renegotiation requires complete stall of IPFIX export**
  - According to *draft-ietf-tsvwg-dtls-for-sctp-01*, renegotiation cannot start before all previously exported IPFIX Messages are either
    - received and acknowledged by Collector or
    - discarded due to limited lifetime (PR-SCTP)
  - IPFIX export can only restart after renegotiation has finished



- Possible solutions
  - **Instead of DTLS renegotiation, Exporter opens a new DTLS/SCTP transport session to Collector**
    - “soft hand-off” of IPFIX export to new transport session after DTLS handshake is finished and Templates have been sent
    - this is a standard conform solution
  - **Collector keeps old keying material as long as necessary to decrypt IPFIX Messages exported before the renegotiation**
    - keeping old keying material is not covered by DTLS standard
    - IPFIX export does not have to be interrupted

Conclusion

- Opening a new IPFIX Transport Session solves both problems
  - Disadvantages:
    - frequent DTLS handshakes involve additional public key operations
      - session resumption should be supported (= reuse of old pre-master secret)
    - Templates and Options have to be resent on new Transport Session
    - IPFIX Transport Session represents a scope for IPFIX
      - Collector should be able to associate related Transport Sessions

- Alternative solutions not yet available
  - dead peer detection for DTLS/UDP
    - *DTLS Heartbeat Extension* will solve the DTLS/UDP problem
  - parallel usage of old and new keying material after DTLS renegotiation
    - not conform with *draft-ietf-tsvwg-dtls-for-sctp*

- Who else is working on IPFIX-over-DTLS?
  - Let’s share experience and perform interoperability tests!

- We think that an update of the *IPFIX Implementation Guidelines* will be useful.