Recommendations for Implementing IPFIX over DTLS

draft-mentz-ipfix-dtls-recommendations-01

Daniel Mentz, Gerhard Münz, Lothar Braun

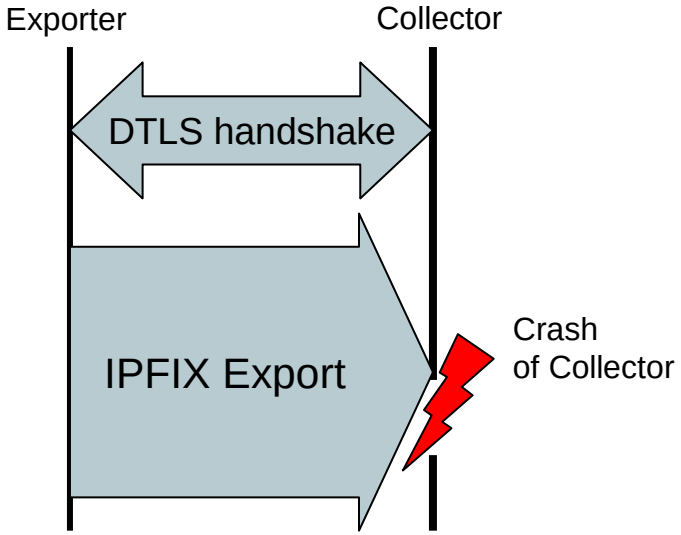
77th IETF Meeting, Anaheim, 2010

Introduction

- RFC 5101:
  - support of DTLS mandatory for IPFIX-over-SCTP and IPFIX-over-UDP for **security reasons**
- Implemented 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
- Implementation guidelines give limited advice on how to implement DTLS support
- Found four issues that should be addressed

Problem (1) 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



- **Recommended Solution:**
  - **DTLS Heartbeat Extension**
    - draft-seggelmann-tls-dtls-heartbeat-02 (February 2010)
- **Alternative Workarounds:**
  - **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

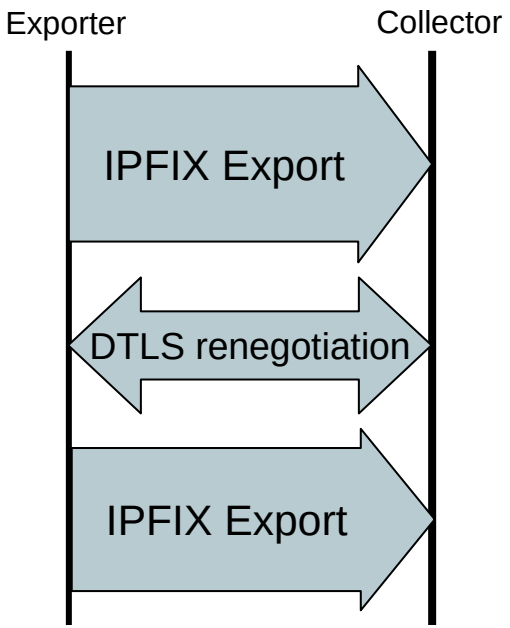
Problem (2): Incorrect PMTU on IPFIX-over-DTLS/UDP

- **Exporter must not generate Messages larger than PMTU**
  - Either by configuration or by discovery
  - Problem on discovery:
    - PMTU discovery required DF bit set
    - PMTU estimate update only after packet loss
    - ICMP “fragmentation needed and DF set“ messages might be filtered by firewalls
  - Consequences:
    - Loss cannot be identified by the Exporter
    - Exporter keeps incorrect PMTU estimate

- **Recommendation:**
  - **Use heartbeat extension from draft-seggelmann-tls-dtls-heartbeat-02**
    - Variable sized heartbeat messages
    - Heartbeat message size is reduced if message is not acknowledged

Problem (3) with IPFIX-over-DTLS/SCTP

- **DTLS renegotiation requires complete stall of IPFIX export**
  - According to draft-ietf-tsvwg-dtls-for-sctp-04, DTLS renegotiation cannot start before all previously exported IPFIX Messages are acknowledged by the Collector
  - IPFIX export can only restart after renegotiation has finished



- **Recommendation:**
  - **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

Annotation (4): Mutual Authentication via Pre-Shared Keys

- **RFC 5101 requires mutual authentication with X.509 certificates**
  - PKI is necessary
  - Maintaining a PKI may be disproportionate for small environments
  - Costly public key operations on handshake/renegotiation

- **RFC 4279 defines a set of new ciphersuites that use pre-shared keys**
  - Pre-configured keys on the monitoring device
  - No asymmetric keys, no costly public key operations or PKI needed
  - Problem:
    - Does not conform to RFC 5101

Discussion

- An update of the *IPFIX Implementation Guidelines* will be useful
- DTLS Heartbeat Extension should be used for DTLS/UDP
  - Solves the “dead peer problem”
  - Can help to discover PMTU
  - Needs support in the TLS group
- Allowing pre-shared keys as per RFC 4279 could be useful
- Who else is working on IPFIX-over-DTLS?
  - Let’s share experience and perform interoperability tests!