Recommendations for Implementing IPFIX over DTLS

draft-mentz-ipfix-dtls-recommendations-02

Daniel Mentz, Gerhard Münz, Lothar Braun

80th IETF Meeting, Prague, 2011
Recommendations for Implementing IPFIX over DTLS

Background

- **RFC 5101:**
  - support of DTLS mandatory for IPFIX-over-SCTP and IPFIX-over-UDP for **security reasons**

- Implemented DTLS support for our monitoring probe VERMONT
  - [http://vermont.berlios.de/](http://vermont.berlios.de/)
  - based on OpenSSL and patches of Michael Tüxen and Robin Seggelmann
    - [http://sctp.fh-muenster.de/dtls-patches.html](http://sctp.fh-muenster.de/dtls-patches.html)

- Implementation guidelines give limited advice on how to implement DTLS support

- Found several problems during implementation phase
Problem with IPFIX-over-DTLS/UDP

- **Missing “dead peer detection”**
  - problem
    - IPFIX traffic is unidirectional
    - DTLS requires shared state
  - Problem occurs on collector restart/crash
    - Collector looses state
    - state-loss cannot be detected by Exporter
    - Exporter continues to export encrypted Messages
    - results in Message loss

- **Recommended: DTLS Heartbeat Extension**
  - draft-seggelmann-tls-dtls-heartbeat-02 (February 2010)
  - problem: development in TLS-WG stalled

- **More workarounds in the draft**
  - trigger DTLS renegotiations periodically
  - open new DTLS/UDP transport association periodically
Problem with IPFIX-over-DTLS/SCTP

- DTLS renegotiation requires complete stall of IPFIX export
  - in case of DTLS renegotiation as defined in RFC 6083
  - renegotiation requires full stop of IPFIX export
  - Problem
    - buffers can fill up
    - Records/Messages can be lost

- Proposal:
  - avoid DTLS renegotiation for IPFIX Export
  - if new keying material is required
    - 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
Mutual Authentication via Pre-Shared Keys

- **Not a problem, more a nice to have**
  - reduces costs of association setup
  - simplifies DTLS/TLS setup

- **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 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

- **DTLS Heartbeat Extension should be used for DTLS/UDP**
  - however, no progress is made in the TLS group
  - do we want to push it?
  - is there a way for us to do this?

<table>
<thead>
<tr>
<th>Fix</th>
<th>Problem in RFC 5101/ Update Guidelines</th>
<th>Dead Peer UDP</th>
<th>Renegotiation SCTP</th>
<th>MTU UDP</th>
<th>Ciphers all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do noting</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Update Guidelines</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>State Problem in RFC 5101/ Update Guidelines</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Update RFC 5101/ Update Guidelines</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>