draft-ietf-radext-radiusdtls-bis
Making RADIUS/(D)TLS a Proposed Standard

The finish line keeps moving away

IETF 123 in Madrid – radext | 21.07.2025

Janfred Rieckers | DFN-Verein

What happened since IETF122

- 3 new draft versions (each one hopefully a step closer to WGLC)
- 1st new draft version with fixes for most issues/PRs opened in Github in May
- Discussions on the mailing list on most open issues
- 2nd new draft version with some more fixes just before draft deadline
- Ongoing discussion on the mailing list
- 3rd new draft version just at the end of the hackathon

Resolved issues (hopefully)

- DTLS session tracking (5-tuple vs CID), added sentence with a „there be dragons“ regarding IP address based policies
  - Spec already allowed using a different, equivalent session tracking method than the 5-tuple, now more explicit
- Usage of CN for certificate validation, now only explicitly forbidden for servers, clients can use CN if they want (not if configured with FQDN or IP addr/range, then use appropriate SAN)
- Added sentence regarding use of different clients doing both RADIUS/UDP and RADIUS/(D)TLS behind the NAT
- Changed „4-tuple“ to „5-tuple“ with a fixed protocol (UDP) to be consistent with usual session tracking mechanisms

Open Issues

- DTLS retransmissions
- Retransmissions in general
- (Acct-Delay-Time vs Event-Timestamp)

DTLS retransmissions

- Current spec requires the server to re-process retransmissions of responses through DTLS stack -> Cache RADIUS response instead of DTLS Record with the response
- No specification on client behavior on retransmissions
- RFC9167 (DTLS 1.3), Section 4.2.1 mandates that retransmissions of lost messages MUST be done using the same epoch and keying material
  - Does this apply to RADIUS retransmissions as well?
    If so: current specified server behavior contradicts this
- Input from (D)TLS experts needed
  - would we need to throw away all cache entries? (unanswered pkt waiting for retransmission on client side, cached responses on server side)

Retransmissions in general

- Discussion on ML around retransmissions and their need
- Retransmission is forbidden in RADIUS/TLS (due to the reliability of the transport)
- RADIUS/DTLS must do retransmissions (esp. for proxies if the packet came via TLS)
- Re-sending the packet again after it completely timed out (“application layer retry”) should still be permissible
  - The information may still need to be transmitted, even if the RADIUS packet timed out
- Is the text in the current spec clear enough? Do we need to change something? Is this in scope?

Acct-Delay-Time vs Event-Timestamp

- Current spec discourages the use of Acct-Delay-Time
  - Causes problems if attribute is updated on every retransmission, because RADIUS packet changes and will not be deduplicated
    - Re-using the same ID for the new packet works only for max 2-hop proxy chains, since proxy may not re-use the same ID as well
  - Does not include delay on transport layer
- Instead, spec recommends use of Event-Timestamp
  - Stays constant across retransmits
  - Transport delays are not important
- Discussion on mailing list on necessity of this recommendation
- I don’t see any argument against leaving the current text in as it is.

What else before WGLC?

- What else have we missed before we can start the WGLC?
- Intended time for this milestone was January 2024, we should finish this document soon-ish

Discussion/Questions?

- Contact
  - Jan-Frederik Rieckers

Mail: rieckers@dfn.de
Phone: 0049 30 884299-339
Fax: 0049 30 884299-370

Address:
DFN-Verein, Geschäftsstelle
Alexanderplatz1
10178 Berlin