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

IETF 121 in Dublin – radext | 04.11.2024

Janfred Rieckers | DFN-Verein

What happened since IETF120

- Discussion on the ML on some key issues
- Reviews came in \o/ (Thanks to all reviewers for their comments)
- PRs on Github from Fabian, Ethan, et al.
- Interim meeting in October
  - Discussions on open issues with the draft

Results of the interim

- Proxying / load balancing considerations are deemed out of scope for this document
  - This includes „what is the same server“
  - Open question: Text for Server Name Indication?
  - Problem discussion maybe in a new „RADIUS Lessons Learned“ document
- Dealing with unwanted packets (i.e. Accounting) needs some more thought
  - out of scope or not out of scope?
    - Issue is not unique for RADIUS/(D)TLS, but RADIUS/UDP has different ports for auth and acct, so maybe still worth saying something about
- Trust decisions and certificate validation text was updated

Selfie attack – looping back the TLS traffic

- Attack precondition: RADIUS/(D)TLS has an aligning configuration for client/server
  - Either the same TLS-PSK for both directions (for TLS-PSK) the server configuration would accept its own client certificate and vice versa (for TLS-X.509-*)
  - Latter scenario could happen in cases of dynamic discovery
- Attacker could mirror TLS traffic back to the same server and cause a loop.
  - More sophisticated loop if traffic is not looped back to the same server
- RADIUS itself has no automatic loop prevention, additional means are available, but may not be used

Selfie attack – proposed countermeasures

- Option 1: Check if the server certificate presented is the own server certificate
  - Objections regarding implementability of the approach and operational considerations
- Option 2: Check client random received in TLS handshake if equal to some own client random
  - May not be forward compatible with future TLS versions
  - Implemented in radsecproxy during hackathon by Fabian, easy with current openssl callbacks, not many lines of code
- Option 3: Use Status-Server message to send a peer ID, as first packet
  - peer ID can be generated at random on each startup, check is easy
  - server could send its own peer-ID back, or an error message if check fails
  - clients would wait for server to answer before sending other RADIUS packets
  - backward compatible with old clients (they would just omit the peer ID attribute in the Status-Server packet)

Hot off the press (thanks Hackathon-Team): DTLS records and RADIUS packet

- Current document has no specification how RADIUS packets are put into TLS records
  - implicit assumption: for TLS treat it as a stream without boundries, for DTLS treat each record as one RADIUS packet (specification on the receiver's side)
- We should add explicit specification how to send RADIUS packets for DTLS (one RADIUS packet per DTLS record). TLS is a stream anyway, no need to specify.
- Do we have experience with existing DTLS implementations? Are they doing it? Do the TLS libraries provide this interface or are they doing some obscure magic?

Next steps

- Include feedback from reviews and from interim
- Fix remaining TODOs in the document
  - My plan: until end of this year, definitely before next IETF
- What else is needed for WGLC?
  - Lots of changes from RFC6614/7360. Implementors should look it over and see if their implementation needs updates
    - Put all needed updates for configuration in Appendix

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