Discovery of Network-designated OSCORE-based Resolvers: Problem Statement

draft-lenders-core-dnr

Martine S. Lenders (martine.lenders@tu-dresden.de), Christian Amsüss,
Thomas C. Schmidt, Matthias Wählisch

IETF 123, CoRE WG Meeting, 2025-07-22

Goal

- SvcParam definitions to bootstrap CoAP security
- Example Use Cases:
    - Use DNS to find security context
    - Use DDR (RFC9462) or DNR (RFC9463) to find local recursive DoC resolver, e.g.:
        - Use DNS to find DoC resolver (RFC9462)
        - Use Neighbor Discovery to find DoC resolver (Encrypted DNS Option, RFC9463)
        - Use DHCP to find DoC resolver (Encrypted DNS Option, RFC9463)
    - This may include:
        - Configure EDHOC credentials for CoAP server
        - Configure ACE details for CoAP server
        - Find (D)TLS raw public keys when not using WebPKIs
- Problem: What appropriate SvcParams are needed?

TODOs from IETF 122 for draft-lenders-core-dnr

Sharpen into two (possibly independent) main topics:

a: Lead DoC client to do object security

b: Help DoC client to find security context

- Example for $a \land \neg b$: EDHOC with WebPKI (“TLS-like EDHOC”)

TODOs from IETF 122 for draft-lenders-core-dnr

Sharpen into two (possibly independent) main topics:

a: Lead DoC client to do object security

b: Help DoC client to find security context

- Example for $a \wedge \neg b$: EDHOC with WebPKI (“TLS-like EDHOC”)

Not addressed in `-core-dnr` yet!

What Happened Elsewhere: draft-ietf-core-transport-indication

Defines the following SvcParams:

- `cred`: Provide COSE credentials (i.e., addresses topic *b*)
- `edhoc-info`: Provides 1 `APP_PROF_SEQ` (EDHOC Application Profile, draft-ietf-lake-app-profile) for `.well-known/edhoc` (i.e., addresses topic *a*)
- `oauth-hints`: Provides AS Request Creation hints (i.e., configures ACE details, potentially addressing *b*).

Which CoAP transport to use? Folded into ALPN:

- `coap` CoAP over TLS (RFC8323)
- `co` CoAP over DTLS (draft-ietf-core-coap-dtls-alpn)
- `COAP` CoAP over (unsecured) TCP (new)
- `CO` CoAP over (unsecured) UDP (new)

What Happened Elsewhere: draft-ietf-lake-app-profiles

Defines the following SvcParams:

- `edhocpath`: Provides 1 or more paths to do EDHOC (beyond `.well-known/edhoc`)
- `edhoc-app-prof`: Provides 1 or more `APP_PROF_SEQ` (EDHOC Application Profile) for each `edhocpath` (i.e., addresses topic *a*)

What Happened Elsewhere: draft-ietf-lake-app-profiles

Defines the following SvcParams:

- `edhocpath`: Provides 1 or more paths to do EDHOC (beyond `.well-known/edhoc`)
- `edhoc-app-prof`: Provides 1 or more `APP_PROF_SEQ` (EDHOC Application Profile) for each `edhocpath` (i.e., addresses topic *a*)

- `edhoc-app-prof` somewhat duplicates `edhoc-info`, see mailinglist discussion
  - ⇒ Conclusion @ Hackathon: `edhoc-info` will be removed in future versions of `transport-indication` (PR#21)

What is missing?

- $a \land \neg b$: EDHOC with WebPKI

What is missing?

- $a \wedge \neg b$: ~~EDHOC with WebPKI~~ Could be an EDHOC App Profile

What is missing?

- $a \land \neg b$: ~~EDHOC with WebPKI~~ Could be an EDHOC App Profile
- Find (D)TLS security contexts.

What is missing?

- $a \land \neg b$: ~~EDHOC with WebPKI~~ Could be an EDHOC App Profile
- Find (D)TLS security contexts. Could be folded into `creds`…

What is missing?

- $a \wedge \neg b$: ~~EDHOC with WebPKI~~ Could be an EDHOC App Profile
- Find (D)TLS security contexts. Could be folded into `creds`… Do we want that?

What is missing?

- $a \land \neg b$: ~~EDHOC with WebPKI~~ Could be an EDHOC App Profile
- Find (D)TLS security contexts. Could be folded into `creds`… Do we want that?
  - Maybe DANE (TLSA records) would be the better route here?

Is there still a problem
that needs stating?

Hackathon Report

- Synced with Marco on SvcParam formats for `edhocpath`&`edhoc-app-prof`
- DoC in Unbound: Continue to work on draft PR
  - ☑ Make OSCORE credentials non-constant and configurable
  - ☑ Make CoAP resource path non-constant and configurable
  - □ Find out why libcoap sends a piggybacked ACK-message for late responses instead of the correct CON-message
  - □ Reuse TLS-PKI for DTLS
- Shipping `aiodnsprox` for Fedora
  - ☑ Make OSCORE credentials non-constant and configurable
  - ☑ Make CoAP resource path non-constant and configurable
- A usable DTLSv1.3 implementation for embedded systems, CoAP, and Python would be great!