TEEP Protocol

draft-ietf-teep-protocol-08
Hannes Tschofenig, Ming Pei, David Wheeler,
Dave Thaler, Akira Tsukamoto
Items updated as agreed at IETF 112 (1/2)

• #40: can send QueryResponse without QueryRequest if:
  • the last QueryRequest contained no token or challenge, AND
  • the TEEP Broker didn’t send call ProcessError since the last QueryRequest received, AND
  • TAM key/cert is cached and still valid

• #166: When is token included in Update message?
  • Explicitly say up to TAM implementation to decide whether to insert a token
  • TEEP Agent must ensure suit-report-nonce present if Update contains a nonce
Items updated as agreed at IETF 112 (2/2)

• #158: Three types of SUIT manifests in Update messages
  4.4.1. Example 1: Having one SUIT Manifest pointing to a URI of a Trusted Component Binary
  4.4.2. Example 2: Having a SUIT Manifest include the Trusted Component Binary
  4.4.3. Example 3: Supplying Personalization Data for the Trusted Component Binary
  4.4.4. Example 4: Unlinking Trusted Component

• (details as presented by Akira at IETF 112 where conclusion was to put them in the TEEP protocol spec)
Use of SUIT
#170: Sample SUIT Reports

- Added examples directly from Brendan’s IETF 111 slides
- Not sure I got the text descriptions right though
  - Need Brendan to check

```plaintext
/ suit-report-records / 4: {
  / suit-record-manifest-id / 1:1,
  / suit-record-manifest-section / 3: 1,
  / suit-record-section-offset / 3: 66,
  / suit-record-dependency-index / 5: 0,
  / suit-record-dependency-resolution / 5: 404
}

107{
  authentication-wrapper,
  / manifest / 3:<{
    / manifest-version / 1:1,
    / manifest-sequence-number / 2:3,
    common,
    dependency-resolution,
    install, validate, run, text
  }>>,
}
```
#168: Removing (unlinking) a component

- Addressed by adding SUIT manifest example from IETF 112 of unlinking a trusted component using directive-unlink to decrement a refcount
- Details left to SUIT documents per IETF 112 discussion
- Added reference to draft-ietf-suit-trust-domains
#167: Simplify Ciphersuites

- Per IETF 112, now reuses COSE Algorithms Registry instead of creating TEEP original ciphersuites

<table>
<thead>
<tr>
<th>Type</th>
<th>Algorithm</th>
<th>TAM</th>
<th>TEEP Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>teep-cose-sign-algs</td>
<td>ES256 (ECDSA w/ SHA-256)</td>
<td>MUST</td>
<td>MUST support at least one of</td>
</tr>
<tr>
<td></td>
<td>EdDSA</td>
<td>MUST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PS256 (RSASSA-PSS w/ SHA-256)</td>
<td>MAY</td>
<td>MAY</td>
</tr>
<tr>
<td></td>
<td>PS384 (RSASSA-PSS w/ SHA-384)</td>
<td>MAY</td>
<td>MAY</td>
</tr>
<tr>
<td></td>
<td>PS512 (RSASSA-PSS w/ SHA-512)</td>
<td>MAY</td>
<td>MAY</td>
</tr>
<tr>
<td></td>
<td>RSAES-OAEP w/ SHA-256</td>
<td>MAY</td>
<td>MAY</td>
</tr>
<tr>
<td></td>
<td>RSAES-OAEP w/ SHA-512</td>
<td>MAY</td>
<td>MAY</td>
</tr>
<tr>
<td>teep-cose-encrypt-algs</td>
<td>AES-CCM-16-64-128 (AES-CCM mode 128-bit key, 64-bit tag, 13-byte nonce)</td>
<td>MUST</td>
<td>MUST</td>
</tr>
<tr>
<td>teep-cose-mac-algs</td>
<td>HMAC 256/256 (HMAC w/ SHA-256)</td>
<td>MUST</td>
<td>MUST</td>
</tr>
</tbody>
</table>
SUIT Ciphersuites

- Proposal from IETF 112 SUIT meeting:
  - MUST implement HSS-LMS (RFC 8778)
    - Quantum resistant, faster verification, but private key requires maintenance
  - SHOULD implement ECDSA
    - Mature tooling
  - MAY implement others: RSA, SHA-512?, SHA3?

- Should TEEP add HSS-LMS to MUST for TAM and drop ES256 to a SHOULD?
  - What about TEEP Agent’s “at least one” set?
EAT Profile
#171 EAT Profile Discussion

• draft-ietf-rats-eat section 7 covers requirements for an EAT Profile:
  • Profile label
  • Use of JSON, CBOR or both
  • CBOR Map and Array Encoding
  • CBOR String Encoding
  • CBOR Preferred Serialization
  • COSE/JOSE Protection
  • COSE/JOSE Algorithms
  • Detached EAT Bundle Support
  • Verification Key Identification
  • Endorsement Identification
  • Freshness
  • Required Claims
  • Prohibited Claims
  • Additional Claims
  • Refined Claim Definition
  • CBOR Tags
  • Manifests and Software Evidence Claims
Principle used in draft-08: better to put something even if wrong, so people can tell us what’s wrong

- profile-label: The profile-label for this specification is the URI https://datatracker.ietf.org/doc/html/draft-ietf-teep-protocol-08. (RFC-editor: upon RFC publication, replace string with "https://www.rfc-editor.org/info/rfcXXXX" where XXXX is the RFC number of this document.)
#171 EAT Profile Discussion

- Use of JSON, CBOR or both: CBOR only.
- CBOR Map and Array Encoding: Only definite length arrays and maps.
- CBOR String Encoding: Only definite-length strings are allowed.
- CBOR Preferred Serialization: Encoders must use preferred serialization, and decoders need not accept non-preferred serialization.
- CBOR Tags: CBOR Tags are not used.
- Freshness: See Section 9.
- Detached EAT Bundle Support: DEB use is permitted.
#171 EAT Profile Discussion

- COSE/JOSE Protection/Algorithms: See Section 8.
- Detached EAT Bundle Support: DEB use is permitted.
- Verification Key Identification: COSE Key ID (kid) is used, where the key ID is the hash of a public key (where the public key may be used as a raw public key, or in a certificate).
- Endorsement Identification: Optional, but semantics are the same as in Verification Key Identification.
#171 EAT Profile Discussion

- Required Claims: None.
- Prohibited Claims: None.
- Additional Claims: Optional claims are those listed in *next slides*.
- Refined Claim Definition: None.
- Manifests and Software Evidence Claims: The sw-name claim for a Trusted Component holds the URI of the SUIT manifest for that component.
#165: Dependency on draft-birkholz-rats-suit-claims

- Discussed in RATS/SUIT/TEEP interim meeting
- TEEP TAM uses claims in Attestation Results (evidence is opaque)
- Claims in Attestation Results might be copied from evidence
- EAT can be used for both, so claims can apply to either one
## EAT Claims

<table>
<thead>
<tr>
<th>Requirement from arch draft</th>
<th>draft -07</th>
<th>draft-08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor of the device</td>
<td>vendor-identifier [Birkholz]</td>
<td>oemid [EAT]</td>
</tr>
<tr>
<td>Class of the device</td>
<td>class-identifier [Birkholz]</td>
<td>class-identifier [Birkholz]</td>
</tr>
<tr>
<td>Device unique identifier</td>
<td>device-identifier [Birkholz]</td>
<td>ueid [EAT]</td>
</tr>
<tr>
<td>TEE hardware type</td>
<td>chip-version [EAT]</td>
<td>chip-version [EAT]</td>
</tr>
<tr>
<td>TEE hardware version</td>
<td>chip-version [EAT]</td>
<td>chip-version [EAT]</td>
</tr>
<tr>
<td>TEE firmware type</td>
<td>component-identifier [Birkholz]</td>
<td>sw-name [EAT]</td>
</tr>
<tr>
<td>TEE firmware version</td>
<td>version [Birkholz]</td>
<td>sw-version [EAT]</td>
</tr>
<tr>
<td>Freshness proof</td>
<td>nonce [EAT]</td>
<td>nonce [EAT]</td>
</tr>
</tbody>
</table>
Next steps

• Update draft with feedback from hackathon & this meeting
• Initiate WGLC on next rev?