EAT Measured Components

draft-ietf-rats-eat-measured-component

IETF 122 Bangkok, RATS WG

Quick Recap

Extends the EAT Measurements claim

It can be used in addition to or as an alternative to CoSWID

"Measured component" ≡ any measurable object on a target environment:

- Firmware component loaded in memory at boot time (typical)
- Run-time integrity check (RTIC)
- File
- CPU register

Changes Since Adoption

Reviews and comments from Giri, Laurence, Carsten, Dionna, MCR, and Carl (🙏)

12 PRs merged, published `ietf-01` and `ietf-02`

Changes Since Adoption (cont)

- Technical
- Editorial
- Features
  - Added (limited) extensibility

Technical

- Remove extra `b64u`-wrapping for JSON in CBOR (Carsten) #17
- Add JSON examples #20
- Fix wrong `flags` size (Dionna) #26
- Describe behaviour when the profile is not known (Carl) #29

Editorial

- Clarification around `signers` by Laurence #13
- Editorial suggestions from Carl #27
- Text changes in the abstract #23
- Slightly rephrase the security considerations #24
- Add example using file system object as `component-id` #25

(Limited) extensibility

"The format SHOULD also allow a limited amount of extensibility to accommodate profile-specific semantics."

- Add a profile-dependent `flags` field #16

  ```
  flags-type = eat.JC<bytes8-b64u, bytes8>
  ```

- Bit mask
- Single value within a predetermined set of codepoints

(Unlimited) extensibility

A slightly more radical approach:

```
any-defined-by-profile-type = eat.JC<bytes-b64u, bytes>
```

Any profile-defined semantics, e.g.:

```
flags-type = bytes .cbor my-extension-map
flags-type = text .json my-extension-map
```

RATS: Do you object to proceeding this way?

Open Issues

- #21 Purely editorial (CDDL import)
- #10 Add SVN?

SVN

There are cases where the measurement includes a Security Version Number (SVN)[1].

Discussed in the context of EAT but not pursued.

[1] An anti-rollback counter utilized in certain attestation schemes, such as Intel TDX and AMD SEV-SNP.

SVN Options

A) As an alternative to `digest`:

```
(digest / svn)
```

B) Both `digest` and `svn` are possible, even one alongside the other:

```
non-empty<(?digest, ?svn)>
```

C) `digest` is mandatory and possibly complemented by `svn`:

```
(digest, ?svn)
```

Next Steps

Sort out the extensibility topic

Sort out the SVN topic

WGLC

FIN