TEEP Tues Nov 7, 2018 Session 1 - IETF 103 Bangkok

Update on interim WG and work sessions (Mingliang Pei)
https/::github.com/ietf-teep created to file and track issues (18 filed), and as a repository for draft docs

draft-ietf-teep-architecture-01 published, with a number of editorial and issue-related changes
(Example of editorial change: "Secure Boot" removed as a term of art, after much discussion on list, and recognition that some target devices do not support it; consensus from IETF102 was that architecture should be 'secure boot agnostic')
(Example of issue-related change: architecture doc now expresses explicit requirement for algorithm and attestation agility.)

Issue #5 - Don't use Secure Boot
Definitions have been clarified, making trusted firmware optional
And secure boot is removed
D. Thaler to close this (and other) issue(s) on GitHub by Noon today.

Issue #6 Attestation Agility
From architecture point of view it is easy
In protocol, we need to look forward to using EAT/RATS
We will close this issue, because the requirement is clear in the arch document in section 8

v01 now includes text to define and distinguish between
- Trust Anchor
- Root of Trust

Eric Norrmark: over the course of the device lifecycle, there are times when more than one trust anchor will be present (for example, when transitioning from one trust anchor to its replacement). Follow-up question: does the architecture define any mechanism other than an expiry date in a certificate, to indicate when a trust anchor should be replaced.

Russ H: suggests adding text to define 'trust anchor fingerprint' and its role as part of the validation/replacement process for trust anchors.
Nancy C-W where would this term be used in the documentation as a whole?
Dave T: rather than add as a discrete term definition, append this text to the definition of 'trust anchor'.
General “nod’ from the participants in the room

Issue #29 Device Admin vs Device Owner; almost ready to close, though there was a further suggestions of defining three roles:

- device owner (sets policy defining what can be done with the device)

- device admin (executes/?enforces that policy)
- device operator

Simpler update is to replace all instances of “device owner” to “device admin”. This issue remains open until the updates are made.

Robin W asks (in the notes so as to not interrupt note taking) : in the diagram, Dave W said any comms between the device and a TAM is initiated by the device (not the TAM). Is that implicit, in the diagram, in the unidirectional arrow from the device (specifically, the TEEP Broker) to the TAM? If so, do the unidirectional arrows between other components in the diagram represent the same implicit rule (i.e. that the origin of the arrow denotes the element that always initiates communication)?
Nancy CW: Yes, that is the intent. Although, we should confirm this with the authors, so we can open an issue to track this.

Dave T to resolve any confusion over numbering of issues #18 (believed closed) and #32.

Issue #3 (Trusted App Distribution) needs to be complemented by a further issue statement about how to coordinate version management between TEE and application (i.e. so that it's clear which element is responsible for detecting/managing any mismatches or invalid configurations).

Issue #7 - clarify meaning of Security Domain; may need to be supplemented with description of how to address the management use-case - specifically, could it sensibly be addressed by defining
1 - mandatory security domain
2 - optional security domain, but with a requirement to provision a public key per Security Domain] -- this is what the doc attempts to define currently.
3 - security domain optional, with possibility of avoiding the public key provisioning round-trip -- this option is not currently defined/documented
4 - remove concept of SD from the document

Intuition of how the Security Domain works is really needed. Management is a potential useful case for SDs.
Dave W: agree it's a substantive issue, but it really needs more discussion before decision.
Dave T/Erik N: this boils down to defining what a Security Domain is really used for; is it a management component, or an isolation mechanism?

Issue #13: couple of use-cases described for dependencies among TAs, e.g.:
- one TA that acts as a launcher for one or more other TAs
- one or more TAs that require access to, say, a crypto library which is deployed as a discrete binary (perhaps because it's from a different vendor/maniuacturer)

Nancy C-W: recommend adding a couple of example use cases to the document for illustrative purposes, Update from Dave T on OTrP hackathon

Update from Mingliang Pei on OTrP
https://datatracker.ietf.org/doc/draft-ietf-teep-opentrustprotocol/

Hannes: is there a good way to ensure we document the transport/workflow options, and is it best done in the OTrP spec itself, or in separate documents?
Nancy C-W/Dave T: charter would allow for it to be done as discrete docs, e.g.:
- OTrP architecture (components, relationships, message definitions)
- http transport spec
- COAP transport spec

(Geof Cooper: regarding this point, the TLS spec defines a message layer, which is then mapped into the actual transport. This is what we did in SDO. It might be useful here. Didn't get a chance to get in the queue).

Dave T: TEEP should monitor the work being done in SUIT and evaluate how it affects TEEP's thinking about binary distribution options (particularly, for example, whether/how TEEP caters for direct distribution of binaries by a cilent).