Manifest Requirements from TEEP WG

draft-ietf-teep-architecture

Dave Thaler

TEEP WG dependency on SUIT WG

- TEEP provisions code+config into a Trusted Execution Environment
- Brendan presented SUIT manifest at TEEP interim meeting earlier this year
- TEEP decided to take a dependency on SUIT WG manifest

Summary of SUIT Manifest Requirements

1. Ability to list one or more TAM URIs for a dependency
   - This is different from a URI to download the binary
2. Install steps/dependencies for Security Domain on GP devices
3. Ability to update a file that isn't a binary executable
4. Ability to indicate which TEE a binary should be installed to

Background slides follow, used if there are questions

Dependencies on/from TAs (1/2)

- Issues:
  - #13: Is it in scope: TA depends on another TA and related installation?
  - #34: Version dependencies between TA and normal world app
  - #35: Coordinate TA updates with UA
- Had previous WG consensus to use SUIT manifest for dependencies from TA's
  - Draft already has text talking about Untrusted App manifest expressing dependencies on TA's
- Proposal (pull request #75):
  - Add reference to SUIT manifest and explain it expresses dependencies from TAs
  - Add discussion of compatibility issues when updating a dependency

Dependencies on/from TAs (2/2)

Separate from the Client App's manifest, this framework relies on the use of the manifest format in [I-D.ietf-suit-manifest] for expressing how to install the TA as well as dependencies on other TEE components and versions. That is, dependencies from TAs on other TEE components can be expressed in a SUIT manifest, including dependencies on any other TAs, or trusted OS code (if any), or trusted firmware. Installation steps can also be expressed in a SUIT manifest.

…

Updating a TA may cause compatibility issues with any Untrusted Applications or other components that depend on the updated TA, just like updating the OS or a shared library could impact an Untrusted Application. Thus, an implementation needs to take into account such issues.

Security Domains (1/2)

- Issues:
  - #7: Clarify meaning of Security Domain
    - Also part of #70 Juergen's feedback
  - #62: Editorial update for SD full removal
- Had previous WG consensus to remove formal concept
- Proposal (pull request #72, and part of #75)
  - Remove from API discussion
  - Remove OTrP message name
  - Remove explicit entry from terminology section
  - **Depend on SUIT manifest to express security domain dependency**
  - Include SD informatively in example text (next slide)

Security Domains (2/2)

Separate from the Client App's manifest, this framework relies on the use of the manifest format in [I-D.ietf-suit-manifest] for expressing how to install the TA as well as dependencies on other TEE components and versions. That is, dependencies from TAs on other TEE components can be expressed in a SUIT manifest, including dependencies on any other TAs, or trusted OS code (if any), or trusted firmware. Installation steps can also be expressed in a SUIT manifest.

For example, TEE's compliant with Global Platform may have a notion of a "security domain" (which is a grouping of one or more TAs installed on a device, that can share information within such a group) that must be created and into which one or more TAs can then be installed. It is thus up to the SUIT manifest to express a dependency on having such a security domain existing or being created first, as appropriate.

Keeping secrets from the TAM (1/2)

- Issue:
  - #64 End to end security between a SP and TEE for confidential IP
- Desire to get an encrypted binary that the TAM cannot decrypt
- One proposal was:
  - Encrypted binary can be delivered just like an unencrypted binary
  - Decryption key is delivered separately, but **by a different (SP's) TAM**
- Current text in doc:
  - For any client app, there should be **only a single TAM** for the TEEP Broker to contact.
  - This is also the case when a Client App uses multiple TAs, or when one TA depends on another TA in a software dependency ...
  - The reason is that the SP should **provide each TAM** that it places in the Client App's manifest **all the TAs** that the app requires.

Keeping secrets from the TAM (2/2)

- Option 1) Agent uses a single TAM for TA and all dependencies
  - TAM URI of every dependency is assumed to be same as for the depending TA
  - SP must host TAM for all TAs that depend on its secret
- Option 2) Agent can use a separate TAM per dependency
  - Every dependency can have its own TAM URI in the manifest file
  - If a TA depends on another TA, the dependent TA might even have its own SUIT manifest
  - If TAM URI for a dependency is different from the depending TA, can invoke RequestTA recursively

Multiple TAM URIs for a TA

- Issue:
  - #14 Multiple TAMs for a single Client App
- Ming proposes:
  - A TA binary can be carried by multiple TAMs, similar to application stores for client apps.
  - We propose to allow multiple TAM URIs in a manifest file for a TA where it can be downloaded.
  - Only one of the TAMs needs to be contacted and others can be used as failover TAM if the primary fails to respond.

Trust Anchor Update

- Issue:
  - #32 Trust Anchor lifecycle management
    - Also part of #70 Juergen's feedback
- Current text:
  - It is **out of the scope** in this document to specify how the trust anchors should be updated when a new root certificate should be added or existing one should be updated or removed.
  - **A device manufacturer is expected to provide its TEE trust anchors live update or out-of-band update** to Device Administrators.
- Can't/shouldn't you use TEEP to update trust anchors in the TEE?
- Proposal:
  - A manufacturer may have a Trust Anchor Manager TA, with trust anchors in the configuration data for that TA