IETF 110 - TEEP
Wednesday March 10, 2021 Session I
Chairs: Nancy Cam-Winget, Tirumaleswar Reddy

Notetakers:
Robin Wilton
Nancy Cam-Winget

Agenda bashing, Logistics – Chairs (5 mins)

Architecture – Dave Thaler (5 mins)
Draft: draft-ietf-teep-architecture
Issues: https://github.com/ietf-teep/architecture/issues (https://github.com/ietf-teep/architecture/issues)
Dave presents updates to Architecture version 14.
WGLC @IETF 109, version 14 now posted. Document with new pull request (version 15), needs shepherd review.
Tiru will do a review before putting shepherd writeup.
Dave will wait to do last (version 15) to capture Tiru's comments.
Dave doesn't anticipate the need for further changes to the architecture document at this stage.

TEEP over HTTP update – Dave Thaler (5 min)
Draft: https://datatracker.ietf.org/doc/draft-ietf-teep-otrp-over-http/ (https://datatracker.ietf.org/doc/draft-ietf-teep-otrp-over-http/)
Issues: https://github.com/ietf-teep/otrp-over-http/issues (https://github.com/ietf-teep/otrp-over-http/issues)
Dave presents updates in draft version 10.
Resolution found for the case where a TEEP Agent has to talk to multiple TAMs. Solution is to invoke "RequestPolicyCheck" re
Solution was to invoke "RequestPolicyCheck" repeatedly until empty response.
Tiru will do shepherd review after the architecture doc is dealt with.

Hackathon Report – Akira Tsukamoto(15 min)
The work was really more on both TEEP/SUIT Hackathon.
Hackathon goal was to adopt draft-05 and address the many Pull Requests since IETF109.
Hackathon established consensus that TEEP and TEEP-over-http drafts are at least mature; some work still to do between IETF 110 and IETF 111, but generally "implementable"
Next steps would be to test the security aspects and COSE tokens.

Hackathon Test 1 tested TAM to teep-device; one test superseded because the tested feature is already obsolete.
Two tests uncompleted (1 - create and upload SUIT manifest, and 2 - verify every entry in the manifest, in teep-device) (Slide: Result 1/3)

Also tested the processes for creating Pull Requests on github, and creating Issues on github. 7 test items addressed, of which 2 were merged in Github, one resolved, leaving4 for discussion in this IETF110 session. (Slide: Result 2/3)

Still for investigation/discussion:

- preventing rollback updates
- URI usage
  URI-handling is complex, especially since URI information and TC binaries could well come from different vendors

Brendan Moran:

- Rollback; version number testing should apply, but older payloads can be validly installed by using sequence number. This mechanism intentionally allows e.g. forced downgrades.
- SUIT allows an overridden URI to be delivered, using a 2ndary manifest that simply “asserts” the replacement URI and nothing else. This mechanism is designed to allow the two elements to be signed by different signers (doesn’t have to be the TAM). However, there is *also* the option of encapsulating one signed manifest inside another signed encapsulation. Use cases for this may include handling of encrypted manifests.
- “Delete” command; probably not desirable, as it is likely to break dependencies. Preference would be for an “unlink” or “garbage collect” function, with a decrementing reference counter.
- Sequence number handling; in principle, each TC should have and use a sequence number specific to itself. Otherwise any system with multiple TCs could give rise to very confusing rollback authority ‘orders of precedence’. The sequence number is the 2nd element inside the manifest body.

Hannes Tschoefenig: Do these clarifications belong in TEEP, SUIT, or where?
Dave Thaler: Issues 104 and 105 explanations are about SUIT manifest usage, so belong in SUIT rather than TEEP.
Not ruling out putting a “deep reference” into TEEP documentation to point to the relevant information in SUIT documentation. DT doesn’t see a need for this relating to 104 and 105, but one might exist relating to the “delete”/“unlink” command mentioned above.

Thomas Fossati: are any of the implementations referenced open source? (Asking for RATS…)
Akira Tsukamoto: not all, but Secom and Microsoft implementations referenced in the Hackathon are. (Not Lepidum and AIST implementations of teep-device.)
Thomas: asks if TEEP evidence is also provided in the implementation?
Dave: there are stubs in the Microsoft implementation, but maybe in the next Hackathon the RATs work could be incorporated

Kohei Isobe posted on chat one GitHub source for libteep: https://github.com/yuichitk/libteep (https://github.com/yuichitk/libteep)

Discussion on manifest representation: whether it should be in TEEP vs SUIT draft and handling of updates

Hannes Tschoefenig presenting Hackathon results on encryption-related tests. Both SUIT and TEEP docs point to COSE but don't give examples.
Hannes' conclusions: SUIT manifest spec has many options and few implementation details,which makes developers' task hard.Recommends profiling COSE to reduce complexity, code size, and risk of interoperability problems.

- Dave Thaler: do TEEP docs also need any changes?
- Hannes: More complete examples of the encryption-related functions would be helpful "Generic" COSE encryption handling is hard to achieve, e.g. because of the multiple possible combinations of crypto libraries and CBOR parsers.
  Over all: "virtual hackathons are challenging"
  Hannes will send examples to the list.

TEEP Protocol – Dave Thaler (45 min)
Draft: https://datatracker.ietf.org/doc/draft-ietf-teep-protocol/ (https://datatracker.ietf.org/doc/draft-ietf-teep-protocol/)
Issues: https://github.com/ietf-teep/teep-protocol/issues![(https://github.com/ietf-teep/teep-protocol/issues!%5B)]

Dave listed changes since IETF 109 TEEP and SUIT meetings.
Those still for discussion:
Issue #49: EAT Claims meeting TEEP Architecture requirements. draft-birkholz-rats-suit-claims refers; TEEP has a dependency here (which may be normative or informational), but this Birkholz draft is not (yet?) a RATS WG doc. There will be a discussion on whether the draft belongs in RATs or TEEP in the next RATs IETF 110 session.
Thomas Fossati: asks if the hardware identifiers need to be registered
Dave: I think that is for RATS to decide, not TEEP; TEEP's Architecture doc expresses our requirements.

Issue #67: Appendix now updated to include a sample EAT, but this is not implementable yet because 7/12 parameters are defined in draft-birkholz, so depend on RATS.

Issue #51: Draft-05 recommends minimising the definition of new error codes, such that error codes represent different *protocol* behaviours, but can be differentiated by human-readable error *messages* meaningful to humans.
Slide 7 gives informative examples of error *codes* and corresponding TAM behaviour (i.e. behaviours that are sufficiently distinct to justify a discrete error code).
Robin Wilton: examples for errors that don’t justify their own error code, but can be adequately distinguished through error messages where that is helpful (to the human)
Dave: wasn’t planning on adding one in the document; temporary vs. permanent errors

Issue #129: errors processing QueryRequest; as it stands, the spec describes an error condition that can’t be encountered in normal implementation. Slide 8 sets out three options; Dave T’s preference is for Option 1:
1 - Allow Error to be sent in response to QueryRequest
2 - Silently drop QR (not friendly for debugging!)
3 - Duplicate optional error fields into QR message (duplicative but harmless?)
Akira: OK with 1 or 3
No other comments seen

Issue #132:

- in response to feedbackfromRuss Housley, “HMAC” requirement relaxed to “MAC”
  Question to WG: Should section 9 specify whether future registrations should allow integrity-without-confidentiality ciphersuites?
  Russ: if rules aren’t placed explicitly, then IANA expert can enforce WG consensus
  Tiru: a lot of messages in the protocol have potential privacy impact (sensitive data)

Russ: isn’t advocating either way (e.g. expressing no opinion on whether confidentiality should be mandated)
Jonathan Hammell: what’s the MAC used for?
Dave T: reframing: what is the *ciphersuite* used for?

2 applications in principle;
TEEP message integrity,
Integ/conf of the component being delivered
The assumption here is that the payload has confidentiality, and that the question here is whether the TEEP messages should have confidentiality. Prima facie yes, since they might reveal e.g.updates to a blood pressure monitor…
So: are there use cases that don’t require condfidentiality for the TEEP messages?

Ben Kaduk: looks like TEEP messages are effectively COSE_Sign1 objects; COSE_Sign1 can be done with either a proper asymmetric signature or a symmetric-keyed MAC.

Russ: Yes, but also, do EATs need confidentiality?
Dave T: Understanding is that EATs can use COSE/JOSE to provide confidentiality for their contents.

Dave T: if there’s a use case that doesn’t require confidentiality, we could allow IANA expert the latitude to decide.

Nancy: Intent of the UCCS draft was to address EAT tokens/
claims that do not require confidentiality

Brendan: If integriy-only is to be allowed, it probably needs an entry in the security considerations detailing how much it exposes personal information.
Dave T: propose to update draft to allow for ability to not have confidentiality, but include in the security considerations the use cases where it is appropriate or perhaps state examples where this should not be allowed

Issue #41: SUIT manifest example(s); SUIT manifest spec has 6 other examples in Appx B.
Dave T proposes putting examples in TEEP protocol spec with both manifest and report
Agreement from: Hannes T, Brendan M, Sorin Faibish.

Possible examples:

- Manifest: install a TA that needs personalization data
- Report: partial success
- Manifest: remove a TA and its dependency
- Report: full success

Discussion on how TEEP uses Tokens (for freshness and state matching claims, aligning with RATs):
TEEP evidence/remediation flow, and its relation to RATS;
Verifier cares about:

- freshness of evidence
- currency/obsolescence of evidence values?

RP cares about:

- recent signature of AttestationResult by Verifier
- currency/obsolescence of claims values

Three potential methods:
1 - timestamps; claims are “fresh” based on a timestamp, requiring clock synchronization and secure time synchronization
2 - nonces; removes dependence on time sync, but would need to retain state of “nonce”
3 - epoch handles; non-predictable, “somewhat ephemeral” value stored by both sender(s) and receiver(s) allows for scalability but at the expense of freshness only being within the epoch window

Dave T’s preference would be for method 3, since it’s scalable and doesn’t require constrained endpoints to have clocks, but combinations (incl. what is optional or mandatory for which role) might also be desirable.

Thomas Fossati: with regards to nonce and scalability, I guess you could make it a verifier problem rather than TAM’s

Isobe Kohei: Can TAM work as handle distributor?
Dave T: Yes
Mike Richardson: RATS will change the term “handle” to … probably “epoch identifier”
Hannes: suggest we use what RATS adopts
Dave T.: RATS isn’t chartered to define protocol, and this is a protocol choice (RATs will have to define use of different methods)
Hannes: But RATS architecture doc does discuss ways to demonstrate “liveness” of tokens
Sorin F: prefers method 3 for scalability reasons
Dave T: can start with use of method 3 for now but we may need to consider defining more than one. The spec may support more than one method?
Brendan Moran: Epoch IDs raise security considerations on attackers delaying delivery (e.g. by DoSing the epoch-ID distributor; otherwise, an attack would have to delay equally at both the sender and receiver - or they would notice the difference between their epoch handles)

Conclusion: take this to the list, for further discussion of the trade-offs and implications.

Issue #131: Challenge when attestation bit is set
Should it be allowed to have the attestation bit set *but* the “challenge” field absent?
(Moving on, in the interest of time)

Issue #127: Use of token vs challenge in QueryRequest
Since the token and challenge cannot both be present, should they be merged under a single CBOR label?
Resolution already proposed by Ben K: preferably not, since they are semantically distinct.

Issue #40: Unsolicited QueryResponse
If Attestation is used with timestamps or handles, then no need for per-request token or nonce

Do we still need extra round trip to get QueryRequest?
(Deferred to the list)

Issue #83: How long should TAM keep token?
(Deferred to the list)