TEEP HTTP Transport

draft-ietf-teep-otrp-over-http-14
Approved-announcement to be sent::AD Followup

• IANA OK - No Actions Needed
• SECDIR Stefan Santesson: Ready
• INT Erik Kline: No Objection
• SEC Roman Danyliw: No Objection
• TSV Zaheduzzaman Sarker: No Objection
• ART Murray Kucherawy: No Objection
• ART Francesca Palombini: No Objection
• RTG John Scudder: No Objection
INT: Erik Kline

S5.6, S6.4 (Error handling)
> If any local error occurs where the TEEP/HTTP Client cannot get a message buffer (empty or not) back from the TEEP Agent,

Q: Is there any discussion that can be referenced for how to set “reasonable” timeouts? Even though the HTTP transport layer may be up and functioning, how long is too long to wait for message processing before an error should be declared by one of the layers? I'm sure the timeout needs might vary according to many different factors, but does the desired timeout need to be conveyed by some mechanism to the TEEP/HTTP {Client,Server} layer?

A: added
“Note that no timeout check is used at the TEEP/HTTP Client layer; any timeout would be handled inside the TEEP Agent.”
(Actually even a step later than that, inside a SUIT implementation, but the transport spec only refers to the TEEP protocol spec directly.)
SECDIR: Stefan Santesson

"It is strongly RECOMMENDED that implementations use HTTPS."

This brings my thought to other interesting alternatives to spice requirements as defined in RFC 6919 like "OUGHT TO"? ;)
But jokes aside, I'm not sure "strongly" is appropriate next to "RECOMMENDED".
But other than that I find no issues with the document.

Response: Removed “strongly”

"It is RECOMMENDED that implementations use HTTPS."

SEC: Roman Danyliw

➤ Section 5. Starting in this section there is the introduction of the concept of “message buffers” being exchanged. Is there some more formal description of that idea in this context. I didn’t find that term defined in draft-ietf-teep-protocol or draft-ietf-teep-architecture.

It’s synonymous with what draft-ietf-teep-protocol calls passing back a message. Dropped the word “buffer” and just used “message”, for consistency.

➤ Section 8. The protocol interaction model has URI being exchanged and followed. Consider providing a reference to Section 7 of RFC 3986 to provide generic guidance about de-referencing URIs.

Section 8 is just an example. Added such a reference to the Security Considerations section, since Section 7 of RFC 3986 is specifically about security considerations.
TSV: Zaheduzzaman Sarker

> and a "Trusted Application Manager (TAM)" on the server side) SHOULD themselves run inside a TEE

Q: why is it necessary to use normative language here? is this something this spec describing first for the TEEP architecture? It is however not the intention of this specification to define TAM placement, or?

A: Updated to use text agreed on with Zahed:

“To be secure against malware, a TEEP implementation (referred to as a TEEP "Agent" on the client side is expected to run inside a TEE, and a "Trusted Application Manager (TAM)" on the server side) might or might not run inside a TEE.”

> Since POST responses without explicit freshness information are uncacheable (see Section 9.3.3 of [RFC9110]), no Cache-Control header is needed.

Q: Should this not say “Since POST responses without explicit freshness information are uncacheable (see Section 9.3.3 of [RFC9110]), hence Cache-Control header MUST NOT be used”. I.e. use normative language to avoid the use of that particular header? Also explains if a Cache-Control header would generate error.

A: Updated.
Section 4:
> See Sections 4.4.2 and 6 of [RFC9205] for more discussion of additional security considerations that apply in this case.

Q: Which part of 4.4.2 of 9205 is relevant here?
A: Removed, as section 6 is a sufficient reference.

> See [BCP195] for additional TLS recommendations and [RFC7925] for TLS recommendations related to IoT devices.
Q: The BCP195 reference should be updated to point to 9325, which obsoletes 7525
A: Done

Section 6.2:
> If the TAM passes back an empty buffer, the TEEP/HTTP Server sends a successful (2xx) response with no body. It SHOULD be status 204 (No Content).
Q: As Carsten asks, what alternative to 204 is there? What would be the reason to deviate from the SHOULD?

A: From response to Carsten last fall:
Some implementations might not implement the HTTP response code itself but rely on some underlying middleware. If the middleware instead just uses 200, it's not non-compliant, just less specific than it ought to be. The receiver just relies on a 2xx success as noted in existing text. No change since I'm not sure what would be useful to say so just responding in email.
ART: Murray Kucherawy

➤ Section 1 is "Introduction" yet makes a normative ("SHOULD") assertion, which is peculiar. If this is a new assertion, it should be in its own later section; if it's imported from another TEEP document, I suggest expressing it some other way ("generally required" or suchlike) so that this document isn’t seen as the normative one on this point.

Covered earlier (for Zahed)

➤ The SHOULD near the top of Section 3 is bare, in the sense that it presents a choice to implementers but no guidance about when it might be legitimate to deviate from the recommended behavior. I suggest adding at last a brief discussion of this. (At least I presume this is a new SHOULD; at first I read this as if it's importing a SHOULD from Section 4.13 of RFC 9205, but there's no SHOULD in that section.)

A: Changed SHOULD to MUST

➤ I have similar comments about the SHOULD in Section 6.2. What if I use some other response code?

Covered earlier.

➤ I have the same question as Erik about timeouts.

Covered earlier.
RTG: John Scudder

>> There are two topological scenarios in which TEEP could be deployed:
> The quoted sentence implies that the list (Agent behind NAT/firewall, TAM behind NAT/firewall) is exhaustive. But surely both TAM and Agent on the open Internet would work too?
> For that matter, there's a large body of work (in which I am not at all expert) that aims to accommodate both client and server being behind a NAT or firewall. I assume you don't want to go to the additional labor to cater for this scenario, and therefore it really is one in which TEEP could *not* be deployed.

Updated sentence to say:
“There are two topological scenarios (among others) in which TEEP could be deployed:”

since the point of the text this prefixes is just to motivate the scope of the document being Agent-initiated connections.
Any other questions?