RFC 6962-bis and BCP 190

TL;DR

There is broad agreement in the TRANS working group that the currently defined mechanism is better than all alternatives considered, but this mechanism violates certain provisions of BCP 190.

We’re asking for an exception for RFC 6962-bis to ignore the requirements outlined in Section 2.3 of BCP190.

With such an exception, we would like to clear the outstanding DISCUSS on RFC 6962-bis related to BCP 190 compliance.

Background on CT

CT is a protocol for publicly logging TLS certificates, primarily at point of issuance, such that anyone can audit the issuance behaviors of Certification Authorities (CAs).

- https://tools.ietf.org/html/rfc6962
- https://tools.ietf.org/html/draft-ietf-trans-rfc6962-bis-32

CT-enforcing User Agents require proof that a TLS certificate has been logged in order to successfully validate the certificate.

Existing User Agent policy heavily dictates (limits) CT Log Server behavior.

Background on CT

CT is an adversarial, verifiable protocol that is specifically designed to limit the reliance on trusted third parties, including CAs and CT Logs.

- Client-side guarantees are obtained by restricting CT Logs to a limited, verifiable set of behaviors
- Granting additional flexibility/control to Log Operators would allow gamification of some of the real-world constraints applied to Log Operators

The technical specification is tightly coupled with user agent policy. The flexibility granted by altering CT to support BCP 190 would likely be forbidden by CT-enforcing user agent policies.

Proposed Approach for RFC 6962-bis

We propose continuing with the language specified in Draft 31 before the introduction of the /.well-known/ language in Draft 32.

“The <log server> prefix, which is one of the log's parameters, MAY include a path as well as a server name and a port.”

- POST https://<log server>/ct/v2/submit-entry
- GET https://<log server>/ct/v2/get-sth

Examples from existing Logs, which define distinct prefixes as allowed by the definition of <log server>:

- ct1.digicert-ct.com/log/
- ct.googleapis.com/logs/argon2019/
- ctlog-gen2.api.venafi.com/

Alternatives Considered - /.well-known/

RFC 5785 Section 1.1. Appropriate Use of Well-Known URIs

“… **well-known URIs are not intended for general information retrieval or establishment of large URI namespaces on the Web**. Rather, they are designed to facilitate discovery of information on a site when it isn't practical to use other mechanisms; for example, when discovering policy that needs to be evaluated before a resource is accessed, or when using multiple round-trips is judged detrimental to performance.”

Additionally, the /.well-known/ space is intentionally restricted by service providers, which would require exceptions for Log Operators under certain deployment models.

Alternatives Considered - Directories

CT is an adversarial protocol designed to protect clients against misbehavior by both CAs *and* CT Logs.

This approach would permit Logs to implement undesired behaviors like constantly changing their directory, which could hinder the ability for them to be audited, thus allowing them to hide misbehavior.

Directories force a bifurcation of Log metadata between the existing parameters (MMD, hash algorithm, key, ID) and the directory object.

Directories introduce ambiguities around refresh frequency.

Alternatives Considered - URL Templates

Would require clients to implement complicated client-side logic that is not otherwise required by the protocol. RFC 6962 (non-bis) has been implemented by CAs, Log Operators, and multiple User Agents with no concerns over this matter raised across several years.

Would discourage diverse implementations of CT Auditors, which is harmful to the goals of CT

If forced down this route, CT would likely ossify on a single URL structure that mirrors the existing implementation, which would cause the spec to no longer accurately represent how CT actually works in the wild.

- This isn't merely hypothetical: it happened with WEIRDS (https://mailarchive.ietf.org/arch/msg/apps-discuss/qDrcRMMavJxuOPqQ8h9oduYiz9A)

Protocols that don’t follow Section 2.3 of BCP 190

https://tools.ietf.org/html/rfc7808#section-5.1

https://tools.ietf.org/html/rfc8040

https://tools.ietf.org/html/rfc7482

https://github.com/tootsuite/documentation/blob/master/Using-the-API/API.md#accounts

https://github.com/git/git/blob/master/Documentation/technical/http-protocol.txt