Origin-Signed Exchanges

draft-yasskin-http-origin-signed-responses-05

Jeffrey Yasskin, Chromium

HTTPWG, IETF 104

March 2019

Use Cases

From draft-yasskin-webpackage-use-cases:

- Privacy-preserving prefetch
  - This, with other changes, lets Google Search treat AMP and non-AMP content alike.
- Avoiding Slashdot effect
- Censorship evasion
- Cross-CDN Push (maybe with double-keyed caching?)
- Offline P2P site sharing (with bundling)

Structure

- We sign HTTP request URL + response
  - Request headers seem to always express content negotiation => Variants response header
- TLS-like certificate + CanSignHttpExchanges extension
- Sign(
  - Format version
  - Request URL
  - Response headers
  - SHA-256(leaf certificate)
  - Timestamp range the signature is valid
  - Signature-update URL ("validityUrl"), same-origin with exchange
  - digest/mi-sha256-03 (or name of other header that guards response payload's integrity)
  -)

Chrome 73 shipping SXG-b3

- Only the `application/signed-exchange;v=b3` format
  - Not PUSH or the Signature header
- Security risks are opt-in for websites:
  - New X.509 certificate extension to distinguish from TLS.
    DigiCert is issuing certificates with this extension.
  - CAA requirement
- Request URL is at a fixed offset, so
- We can drop support for older versions by redirecting to the Request URL.

Security/privacy risks

- All off-path risks of CERTIFICATE frame.
- Replay attacks: 0RTT allows replaying requests;
  signed-exchanges allow replaying responses.
  - Mostly a problem for signed personalized content.
- Downgrade attacks: Within an exchange's signature's validity, attacker can push an old, vulnerable or inaccurate version.

Mitigations

- Off-path
  - CAA reduces mis-issuance
  - Fetching validityUrl under TLS identifies stolen private keys but helps surveillance.
- Replay
  - Cookie and authentication headers are blocked.
  - Servers are advised to strip request authentication before processing a to-be-signed exchange, and to only sign Cache-Control:public responses.
  - Could enforce Cache-Control:public but currently don't.
- Downgrade
  - Signature validity capped to 7 days (=OCSP validity). Servers can choose shorter expirations.
  - Fetching validityUrl under TLS would give a weak liveness guarantee but helps surveillance.

Questions

- Do you have ideas for automatically blocking personalized content? Systematically helping servers prevent it?
- How do we trade off security vs anti-surveillance?
- Identify double-keyed HTTP caches?

Discuss!

Backup Slides

Use Cases for non-origin signed exchanges

- Subresource Integrity
- Presence in a Binary Transparency log (*B)
- Appstore-like static analysis (*B)

(*B) With bundling.