Checking Resource Consistency with HTTP Mirrors

Benjamin Beurdouche, Matthew Finkel, Tommy Pauly, Steven Valdez, Chris Wood

IETF 119 - PRIVACYPASS - 2024-03
Agenda

- Open Issues
- Client Fetching Patterns/Authenticity
- Next Steps
Binary HTTP? (#25)

- Draft uses Binary HTTP (RFC 9292) for the content, to encapsulate the entire target resource response in a binary format for the client to check.
- Issue suggests that we should only be validating the content of the resource, and some specific header fields (like content type) rather than the entire response.
- Do we need to be checking anything other header fields with the mirror?
Config rotation (#31)

- If a mirror is caching the resource across responses, its cache might be behind a recently-rotated resource that the client now has.
- How should clients handle inconsistency when the mirror and the resource are temporarily out of sync? Are clients responsible for retrying?
- Should clients be able to tell mirrors to refresh their view?
Thundering herd at expiration (#8)

- If a resource expires and all clients get a new copy, they can create a “thundering herd” to the mirror when they all need to check consistency again.

Mitigations:
- Only check the mirror on-demand for infrequently used resources.
- “Ladder” strategy, where clients get multiple versions of the resource – i.e., the current key and the next key, so they can delay updating.
Client IP leakage (#15)

- If clients directly access a mirror, the mirror can see the client IPs and log the pool of IPs seen
- One mitigation is to spread mirror requests out among a larger set of mirrors
Client Fetching Patterns/Authenticity (#27)

- Brought up on adoption call.
- Currently the document doesn’t call out that the consistency fetch only establishes consistency and not authenticity.
- Some applications should require an additional direct fetch.
- Other applications (privacypass) may be able to rely on the content being provided through other parts of the protocol.

Potential Resolutions:
- Default behavior of the draft requires a “direct fetch”?
- Addition to security considerations that this doesn’t resolve authenticity and allow per-application.
Other Issues

● Consistency/Lifetime Clarifications
  ○ #5 - Consistency at end of key validity.
  ○ #26 - Cache lifetime on mirrored resources.
  ○ #28 - Consistency period

● Batch/Multiple Resource Fetching
  ○ #7 - Batched fetching
  ○ #33 - All valid keys

● #23 - Freshness vs Age
● #29 - Defection probability
● #32 - Content negotiation
● #34 - Mention CORS
Next Steps

● Resolve open issues based on this discussion.
● Other comments
  ○ Consistency endpoint could provide hashes/digest instead.
  ○ Question about interest in a push-based/polling-based model.
● Implementations?