Secondary Certificate Authentication of HTTP servers

draft-egorbaty-httpbis-secondary-server-certs

Eric Gorbaty, Mike Bishop
HTTPBIS
IETF 117, July 2023, San Francisco
Background

• Based on an older draft that the WG has previously discussed
  • draft-ietf-httpbis-http2-secondary-certs-06
• TLS Exported Authenticators (RFC 9261) allow the ability to send and receive X.509 certificates at the application layer
• Proposal: Define support for HTTP/2 and HTTP/3 servers to send secondary certificates to clients, and make themselves authoritative for different origins
  • New frame type on stream 0 / control stream to carry the exported authenticators
• As per usual, we’ll just ignore HTTP/1.1
Basic authentication example

After trusting the inbound certificates, the client can send requests for the newly authorized origins to the server.

The server can send secondary certificates anytime for domains it is also authoritative for:
- server1.example
- server2.example
With Forward / Reverse proxy ("Hybrid Proxy")

Client

(Stream 4)
:method = CONNECT
:protocol = connect-udp
:scheme = https
:authority = relay.example
:path = /server.example/
[+ Encapsulated Request]

(MASQUE)
https://relay.example

This proxy forwards
server.example, and has
cached / reverse proxies
server1.example and
server2.example

[ Encapsulated Request ]

(Stream 0 / Control Stream)
CERTIFICATE
server1.example
server2.example

(Stream 8)
:method = GET
:scheme = https
:authority = server1.example
:path = /

[ Encapsulated Response ]

(Stream 8)
:status = 200 OK
[+ Encapsulated Response ]
Why do we want this?

- Connection reuse is important
  - Helps servers that host content from multiple origins
- Useful for reverse proxies, which are very common for CDNs
  - “Hybrid Proxies” - Forward proxies like MASQUE can cache and reverse proxy a subset of origins for performance and load balancing benefits
- Privacy / Security
  - Servers could make particular origins only accessible for certain users
    - Can combine with client auth mechanisms, like unprompted authentication
  - Excluded users wouldn’t be able to know what origins it serves
What has changed from last time?

- Defined support for both HTTP/2 and HTTP/3
- Only currently includes unprompted server authentication
  - Probably want to tackle client / server authentication separately
- TLS Exported Authenticators are now RFC 9261
- Implementation interest
  - Apple has been recently exploring uses for this mechanism as it pertains to relays / reverse proxies
Open issues / discussion preview

• Is a SETTINGS parameter to advertise support necessary?
  • New frame type (CERTIFICATE) would be dropped by non-supported clients
  • If focused on server authentication, SERVER_CERTIFICATE could be a better name for the frame
  • Backwards Compatibility: Servers might want behave differently for connections with clients that are not known to support this mechanism, especially if they have strict interpretations of the ORIGIN frame
    • For clients that drop CERTIFICATE frames and use ORIGIN to scope coalescing, ORIGIN frames with names not in the initial cert might be considered malicious
  • Synchronization issues between streams over HTTP/3
Open issues / discussion preview

• There are a number of good reasons for the client to prompt certificates from the server
  • ORIGIN frames have less overhead and more clear semantics than certificate frames alone
  • Alt-Svc
  • Exported authenticators specifies support for authenticator requests (RFC 9261 Section 4)
• Adds quite a lot of complexity, as well as potential privacy concerns
  • Probably better as an extension / separate draft, if desired
Open issues / discussion preview

• Currently proposed frame types expect a full authenticator in one frame
• Will have issues with HTTP/2 size limitations and post quantum certificates
• Possibly reintroduce a certificate ID field
  • Use to gather authenticator fragments in over multiple frames in HTTP/2 (and HTTP/3?)
  • Relevant for cases where the server might require the client to indicate which certificate was used for a request
• Can be randomly generated, or sequential
Closing Remarks

- Having a clear focus on server authentication can help us get the ball rolling
- Implementation and experimentation can drive this
  - Interest certainly exists
- Seeking adoption in HTTPBIS