IETF 125 RegExt Session, Shenzhen, March 17th, 2026

Status update on draft-ietf-regext-epp-https

M. Loffredo
IIT-CNR/Registro.IT

External Reviews and Current Status

- Last September, J. Gould requested a review from httpbis
  - We received and addressed feedback from the authors of rfc6265bis regarding the impacts on EoH
- M. Nottingham invited authors to also request a review from the HTTP Directorate. We did so, and the feedback was:
  - «This draft violates many aspects of BCP56, and needs substantial revision to address that»
  - «recommend mapping the semantics of EPP more faithfully to HTTP»
  - «Failing that, if the authors wish to tunnel, they should do so using CONNECT rather than over HTTP semantics (such as POST)»
- HTTP Directorate feedback is currently being discussed

BCP56 Applicability

- BCP56 provides guidance for designing application protocols on top of HTTP

  «Applications often use HTTP as a substrate to create HTTP-based APIs. This document specifies **best practices for writing specifications that use HTTP to define new application protocols**. It is written primarily to **guide IETF efforts to define application protocols using HTTP** for deployment on the Internet but might be applicable in other situations.»

- EoH does not define a new application protocol using HTTP but instead a transport mapping for an existing protocol

Transport Mappings over HTTP: DoH, OHTTP, EoH

Aspect	DoH (RFC 8484)	OHTTP (RFC 9458)	EoH
Application Protocol	DNS	HTTP (encapsulated)	EPP
Payload	DNS wire format	Encrypted HTTP request	EPP XML messages
HTTP role	Transport for DNS messages	Transport for encapsulated request	Transport for EPP messages
HTTP methods	GET and POST	POST	GET and POST
GET usage	DNS query encoded in URL	Not used	Start of the EoH connection (HTTP session)
POST usage	DNS message in request body	Encapsulated encrypted request	EPP XML message in request body
HTTP status codes	Not specialized	Not specialized	Not specialized
Application result	DNS response code in payload	HTTP response inside payload	EPP result code in payload

BCP56 vs. RFC5730

- BCP56 – Section 4.10
  - «Specifications that use HTTP need to carefully consider how application state is handled. HTTP is designed to be **stateless**;..»
  - «Applications are encouraged to **make state explicit in the request** (…) rather than relying upon implicit server-side session state.»
- RFC5730 – Section 2
  - «EPP is a **stateful** XML protocol that can be layered over multiple transport protocols.»
  - «The transport mapping MUST preserve the **stateful** nature of the protocol.»

Challenges in Mapping EPP to HTTP Methods

- RFC5730 requires a clear separation between the EPP semantics and the transport layer
  - Attempting to map EPP commands directly to HTTP methods would require defining a new resource model for EPP objects
  - HTTP status codes are used to signal transport-level success or failure of the HTTP request itself, while EPP result codes convey the outcome of the EPP command
  - EPP is session-oriented. A client establishes a session using the <login> command, and subsequent commands are processed within the context of that session until the client issues a <logout> command

Operational Challenges of Using CONNECT

- CONNECT is often not supported by HTTP load balancers
- It is generally incompatible with the reverse-proxy architecture commonly used in cloud infrastructures
- CONNECT tunnels make the traffic opaque to the surrounding HTTP infrastructure, preventing inspection and policy enforcement
- The CONNECT method may be blocked or restricted by network security policies
- CONNECT tunnels can reduce load-balancing flexibility, since the entire session must typically remain bound to a single backend
- Using CONNECT may require additional operational configuration and complexity compared to standard HTTP request handling

Proposed Next Steps

- Confirm that BCP56 does not apply to EPP transport mappings and leave the document as-is
- Optionally, add an explanatory text clarifying the architectural choice

Discussion

Thanks!