Multilegged Auth for HTTP/2.0

http://tools.ietf.org/html/draft-montenegro-httpbis-multilegged-auth/

HTTP-Auth BoF, IETF 85, Atlanta, 7 November, 2012

Jonathan Silvera
Matthew Cox
Ivan Pashov
Osama Mazahir
→Gabriel Montenegro
(Microsoft)

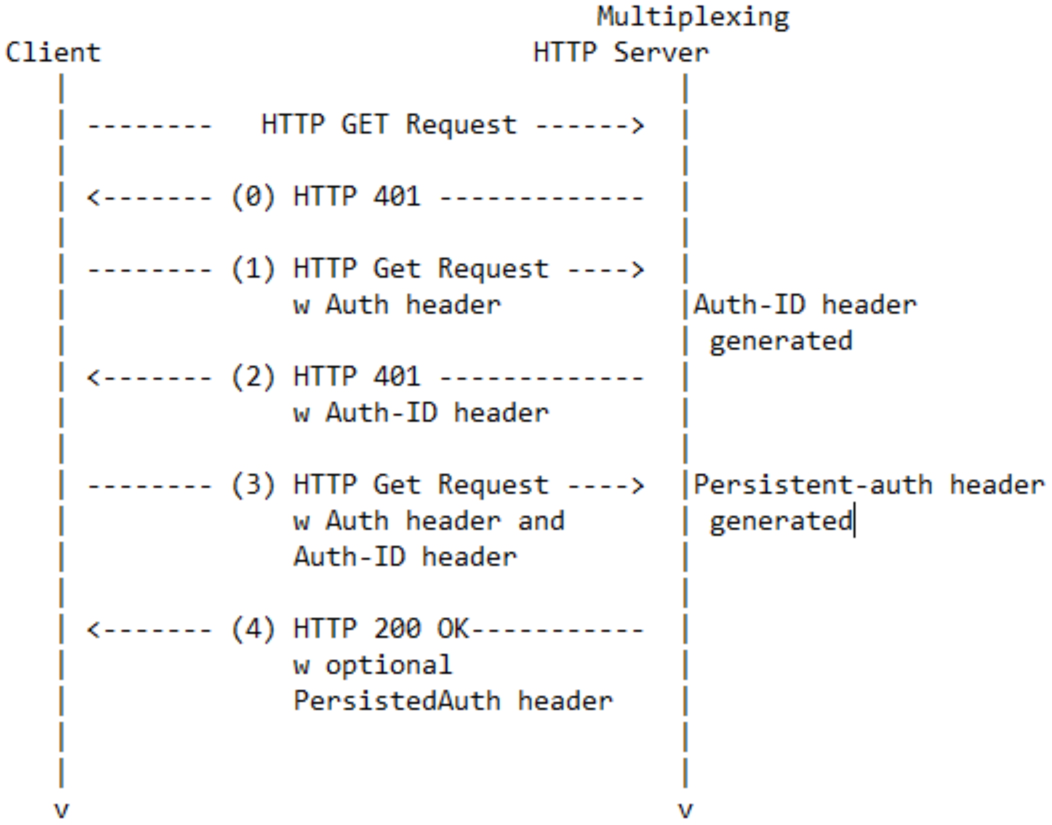
Our Goals in HTTP 2.0 Auth

- Enable HTTP 1.X clients to migrate to HTTP 2.0.
  - There are many multilegged authentication clients (e.g., using Kerberos or negotiate)
  - Initially it was not clear how this would work with multiplexing in HTTP 2.0
- Move whatever state is required into HTTP 2.0 session/connection layer
  - Avoid implicit state from other layers: don't use the fact that the exchanges go over a given TCP connection
  - Use explicit state: exchange explicit headers

Proposal

- Associate separate Requests/Responses as part of the same multilegged authentication exchange
  - *Auth-ID* header
  - Client queues other requests until the first multilegged auth is complete (in case a *Persistent-Auth* header is forthcoming)
- Further optimization: distinguish between per-connection and per-request authentication
  - *Persistent-Auth* header
  - If TRUE: Client may elide authentication of other requests on other streams

Authentication Flow



```
                                     Multiplexing
Client                           HTTP Server
   |                                      |
   | --------   HTTP GET Request ------>  |
   |                                      |
   | <------- (0) HTTP 401 -------------  |
   |                                      |
   | -------- (1) HTTP Get Request ---->  |
   |              w Auth header           |Auth-ID header
   |                                      | generated
   | <------- (2) HTTP 401 -------------  |
   |              w Auth-ID header        |
   |                                      |
   | -------- (3) HTTP Get Request ---->  |Persistent-auth header
   |              w Auth header and       | generated
   |              Auth-ID header          |
   |                                      |
   | <------- (4) HTTP 200 OK-----------  |
   |              w optional              |
   |              PersistedAuth header    |
   |                                      |
   |                                      |
   v                                      v
```

Proxies

- Remote-http-version header added by proxy:
  - HTTP version of the remote host
- Allows client to determine if these headers are used.