HTTP Mutual auth: statuses and updates

Yutaka OIWA
26 Mar 2015 (IETF 92)
HTTPAUTH WG

Updates since IETF 90 (1)

- Adopted several drafts and RFCs
  - RFC 5987 for internationalized strings
  - httpbis-auth-info for Authentication-Info
  - PRECIS username profile for normalization
    - Currently called “saslprepbis”,
      but it is much more general than its name
    - Mandatory in Mutual
- Auth-realm string changed
  - `http://example.com:80` → `http://example.com`
    - Consistent with Web Origin’s string formation

Updates since IETF 90 (2)

- HTTP Auth Extensions:
  - Added an explicit “realm” for pre-auth status
    - Where multiple challenges are provided
  - Added “username”
    - Borrowed Michael’s proposal to Basic, into an experimental draft.

Updates from IETF 90 (3)

- HTTP Mutual Algorithm:
  - Small bug fix for possible DoS, related to handling of mathematically-invalid values
  - Elliptic curve choice issue
    - No change from pre-Toronto
      - No move from NIST curves at this moment

Current “official” issues

- All issues on the tracker are closed

Unofficial “request for comments”

- Sent to HTTPAUTH list on Mar 16
  - Subject: (mutual auth) possible discussions / call for opinions
- 18 questions
  - We think it’s OK, but
  - We want to have comments
- Several comments are already received
  - *Thank you very much!*
- Some of these questions follow.

(p1) use of RFC 5987

- ASCII encoding of internationalized strings in HTTP headers
  - E.g. The user name parameter
    - Renee of France →

      `username="Renee of France"`
    - Renée of France →

      `username*=UTF-8''Ren%C3%89e%20of%20France`

(p2) encoding of RFC 5987

- username*=UTF-8''Ren%C3%89e%20of%20France

charset

Optional language
(between single quotes)

- Fixing charset to “UTF-8”, language empty
  - Rationale: This is not a negotiable parameter
    - Used as binary blobs in many places
      - Recipient-side charset conversion not realistic
    - Make no sense for multi-value provisions
      - NG: username*=ASCII’en’OIWA,
        username*=Shift-JIS’ja’%91%E5%8A%E2

(p5) failure reasons

- Detailed information for clients from servers
  - Some discussion on the mailing list

(p6) Operation Parameters

- Session ID: min. 80 bits
- # of active nonces: min. 32
  - Upper bound for duplicate detections
  - Lower bound for parallel operations
    - Multiple connections and pipelines for HTTP/1.1
    - Multiple streams for HTTP/2.0
- Session key retention: min. 60 s
  - Only an advertisement:
    servers may still discard any keys

(p13) IANA Consideration

- Requirement level for new algorithms (cryptography, parameters)
  - “RFC Required” OK?
  - We provide range of private-use IDs (like those in SecSH protocol)
    - RFC versions MAY also use these if they want
      - Following recommendations in “X- considered harmful” BCP (RFC 6648, BCP 178)

(p15) Optional Authentication

- How is it be signaled?
  - My proposal: *a new header*
    - Guaranteed to be ignored by old clients
  - Alternative: *use* WWW-Authenticate: *with 200*
    - RFC 7235 says:
      *A server **MAY** generate a* WWW-Authenticate *header field in other response messages to indicate that supplying credentials (or different credentials) might affect the response.*
    - Behavior undefined for old clients
    - Some additional rules about header usage needed

(p16) parameter lengths

- Location-when-unauthenticated
  - …… is too long?
  - Possible: unauthed-URL
- Location-when-logout
  - Possible: logout-URL

(p18) IANA Consideration

- Requirement level for new client hints
  - “Specification required” OK?
  - Rationale: this is a catch-all extension point for (trivial) HTTP authentication extensions.
    - Intentionally defined to a loose requirement.

More comments?

- Skipped my questions: p3-4, 7-12, 14, 17
- Other points as well?

Next steps

- Reflect discussions and comments to the next draft.
- Refine the whole English text.
- Proceed to LC?