HTTP Mutual authentication protocol proposal

Yutaka OIWA
RCIS, AIST
New access authentication method for HTTP

Secure (↔ HTTP Basic/Digest, HTML Form)
- No offline password dictionary attack possible from received/eavesdropped traffic

Easy to use (↔ TLS client certificates)

Provides *Mutual authentication*:
- clients can check server’s validity
- Authentication will ONLY succeed with servers possessing valid authentication secrets
- Rogue servers can’t make authentication to succeed
Basic design

- Implemented on top of RFC2617
  - Standard WWW-auth/Auth-info headers used
- Password-based Mutual authentication
  - Using PAKE as underlying crypto primitive
- Authentication only
  - Can be used both with HTTP and HTTPS
  - Encryption/integrity provided by HTTPS
- No long-term storage required
Support for recent Web application design

- Optional authentication
  - Single URI can serve both auth/unauth contents
  - Support for sites like Slashdot, Google or Yahoo
- Timed/server-initiated logout

To solve several current issues with HTTP auth:
- covers reasons to use Form-based auth.
- More features currently under testing:
  will appear in draft-05 (or 06)
UI consideration

- Trusted display for mutual authentication result will be needed
- We propose new UI for this auth scheme
  - Uses browser chrome area
Current status

Spec draft: draft-oiwa-http-mutualauth-04
- 04 draft has solved an IPR issue requested
  - “once becomes Internet Standard” clause removed

Draft Implementations

Server-side: an Apache module

Client-side:
  - Mozilla-based implementation (Open-source)
  - IE-based implementation (closed-source)

Available from project homepage:
https://www.rcis.aist.go.jp/special/MutualAuth/
  - Trial website there!
Included in the current draft:
- Overview
- Detailed protocol description
- Security considerations

NOT included in the current draft:
- UI design description and guidelines
- Design background, decisions & considerations
- Comparisons (Related work)
  - Things which is not suitable for protocol standards
  - We’re preparing a paper for describing those
FAQ: why on HTTP? (or: why not TLS-SRP?)

Answer: Web authentications requires finer controls from Web applications

- Only part of pages in server require auth/authz.
- Two or more “realms” on the same server
  - The above possible with RFC2617 / not by TLS
- Application-initiated logout
- Authed/unauthed contents on single URI
  - Possible with our proposal (or form/cookie)

- How to implement those on TLS/SRP elegantly?
FAQ: why on HTTP? (or: why not TLS-SRP?)

More answer:

For some apps, transport auth is OK.
- If transport’s duration is equal to app’s duration
  - One user per connection, one connection per user
- Examples: IMAP, POP3, FTP, VPN, SVN etc.

However, Web auth. is not so simple

- An “authenticated session” involves several requests
- Multiple independent requests on one connection
- Multiple authentication realms on one server
  - Including “unauthenticated” realm
- So, authentication should be tied to each request, not to each transport
Thank you

More resources

Our project homepage:
https://www.rcis.aist.go.jp/special/MutualAuth/

Draft:

- Some preliminary drafts (before submission) may be on our homepage