

# Invitation for IETF http-auth discussion

---

Yutaka OIWA

Thu, 20 Oct 2011

IIW #13

# Agenda

---

- How we work with http authentication?
- A short introduction of my proposal
- Discussions

# HTTP/Web authentication

---

- Many peoples might agree it is broken
  
- How?
  - People continues to use Form/plaintext auth.
  - People does not use HTTP authentication although there is it
  - We failed to solve any kind of existing problems
    - ◆ Phishing...
    - ◆ Hardness of using any cryptography...


# Things what we have now

---

- HTTP authentication schemes
  - Basic
  - Digest... more or less died
  - NTLM, Negotiate, ... limited usage
- TLS
  - Client authentication ... very limited usage
- Form authentication
  - Very widely used
  - Causes LOTS of problems

# Problem with federated Auth/authz

- Users have to input passwords in a redirect page
  - How we can make sure it is not a phishing page?

 その他のOpenIDでログインする  
 OpenIDを以下のフォームに入力して、「ログイン」ボタンをクリックしてください。



Yahoo! JAPANへログインしてください

 **フィッシングの危険を回避**  
 ログインシールを設定しましょう。  
 ログインシールとは？

Yahoo! JAPAN ID:

パスワード:

次回からIDの入力を省略  
 共用のパソコンではチェックを外してください。



Can you carefully check identity of this form every time without mistake?

# True cause of problem

---

- HTTP etc. has provided *no usable solutions*
  - Recent Web application evolved to provide lots of security-related application features
  - Most of these hard to be implemented on HTTP/TLS authentication
  
- people has difficulty/distaste of using HTTP authentication, prefers Form-based auth

# True cause of problem

---

- (Incomplete) list of modern features implemented by using application-level auth
  - ◆ Complex timeout management of log-in status
  - ◆ Forced/user-originated log-out
  - ◆ Persistent log-in
  - ◆ Site-wide single-sign-on
  - ◆ Federated log-in
  - ◆ Multiple authentication realms (user-name spaces)

# Basic/Digest HTTP auth

---

- Rarely used now
  - Hard to use with recent application
    - ◆ Ugly dialog, mandatory auth design
    - ◆ No (direct) support for log out etc.
  - Not enough security/usability
    - ◆ Basic: as insecure as Form auth
    - ◆ Digest: interoperability problem, improper implementations, off-line password attacks



# TLS Client auth

---

- Secure, but...
  - Usability even worse than HTTP Basic/Digest
    - ◆ Auth before HTTP access
      - = site-wide single authentication space enforced
    - ◆ Site design choices severely limited
  - Hard to manage certificates etc.

# Application-level auth... drawbacks

---

- No protection of passwords to the server
  - Form and server-provided HTML have full control of what is inputted
  - Plaintext always available (often sent) to the server (on TLS, though)
  - No cryptographic protection against fraudulent servers
    - ◆ So-called “Phishing”, many variations

# Chicken and Egg problem

---

- “Improving HTTP-auth is boring, if people does not use those instead of Form auth.”
  - Or, “Why they do not use this incredibly-secure solution existing now?”
    - ◆ *it often does not meet application/business requirements*
- “If there is only HTTP-Basic useful, no one have good reasons to throw Form auth. away.”

# So what we need?

---

- We need to cut the Gordian knots
  - We must provide enough-Secure mechanisms to address existing security problems
  - We must, *at the same time*, provide enough useful mechanisms so that people can move to the new things

# Possible auth means

---

- Passwords
  - Most simple, easy-to-understand credential
  - Starting point for other authentication methods
    - HTTP Mutual authentication proposal
- Certificates, keys in smart cards
- Two-factor authentications (e.g. HW token)
- Federated Authentications
- Use existing backend (SASL, Kerberos etc.)
- Auth for Non-Web backend

# What to discuss

---

- Discussion on the “Problem space”
  - What we should solve from this year
  - What we are required to solve
  - What we can use now
- Discussion on the time scope
  - Possible future timeline/schedule?
- “Cloud/association” of people interested
  - We need friends to work with

# Current document

---

## ■ Problem statement

- [draft-oiwa-http-auth-problem-statement-00](#)

- ◆ Preparing -01 draft now
- ◆ Inputs/suggestions really needed!

## ■ Some technical proposals

- HTTP Mutual auth by us

- ◆ [draft-oiwa-http-mutualauth-09](#)

- ◆ [draft-oiwa-http-mutualauth-algo-00](#)

- GSS-REST by Nico

# Discussion place

---

- *http-auth* mailing list
  - [http-auth@ietf.org](mailto:http-auth@ietf.org)
  - <https://www.ietf.org/mailman/listinfo/http-auth>
- Face-to-face meetings
  - Now 😊
  - In past
    - ◆ Side meeting in IETF Prague (2011-03)
  - In plan
    - ◆ Side meeting again in Taipei (2011-11)?
    - ◆ Seeking opportunity for BoF in Paris (2012-03)



# Related Activities

---

- W3C Federated Social Web ID in Browser activity
  - In session 5, 15:30– today
- Related activities in IETF
  - ABFAB ... authentication beyond Web
  - WEBSEC ... Web security modulo authentication
  - OAUTH ... Authorization for federation
  - Etc.

# A (relatively) short description of HTTP Mutual authentication

---

Yutaka OIWA

Thu, 20 Oct 2011

IIW #13

# Goal

---

- A better authentication which will enable
  - Password-based authentication
  - Strong protection of password, even if it is either eavesdropped or phished
    - ◆ Note: hash is not enough strong against password-crack on recent computers
  - Prevent that *phishing site to make authentication succeed, or even pretend it succeeded*
  - Works well with recent web applications design
- *Mid-/Long-term solution: very secure, but requires both client/server implementation changes*

# HTTP “Mutual” auth.

---

- 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 (phishing) 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 Augmented 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
  - (⇔ Client Certificate, pwd-mgr + auto-gen etc.)

# To overcome “usability” problem

---

- Support for recent Web application design
  - ◆ To solve several current issues with HTTP auth: covers reasons to use Form-based auth.
  - Optional authentication
    - ◆ Single URI can serve both auth/unauth contents
    - ◆ Support for sites like Slashdot, Google or Yahoo
  - Timed/server-initiated logout
  - log-on/log-off page redirection
    - ◆ More to be needed?
      - I need a feedback for that, too

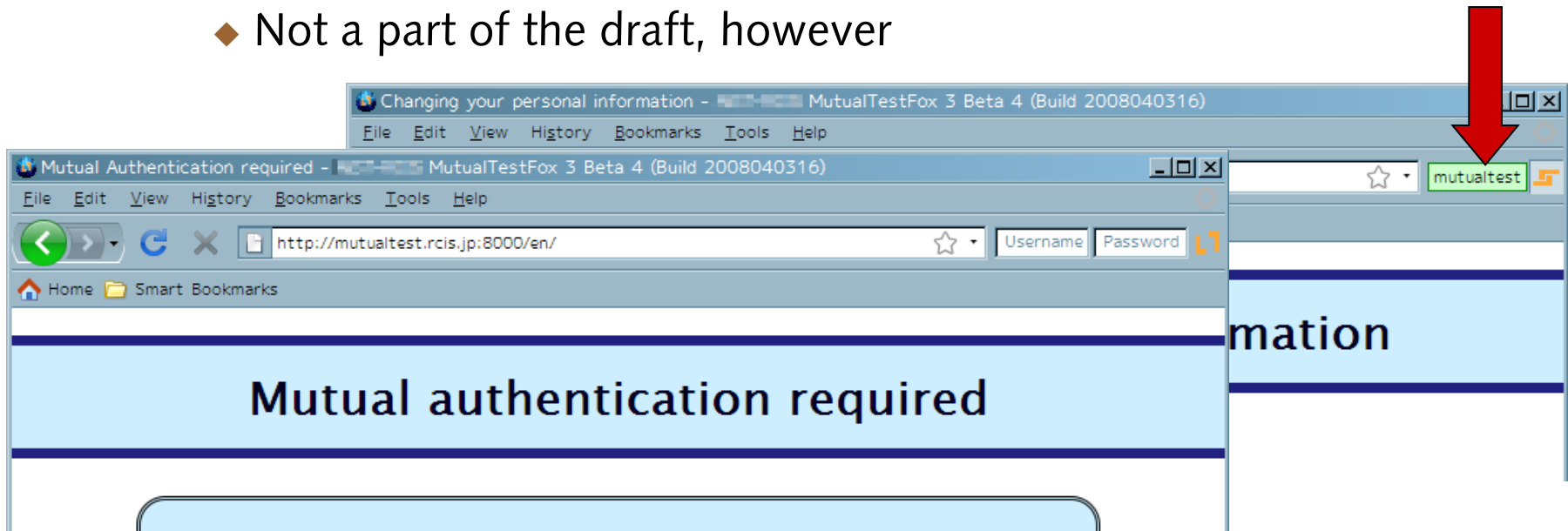
# Draft organization

---

- As of draft-09:
  - 1.: Introduction
  - 2.-9.: Core part
    - ◆ message syntax, state machines, session caching
    - ◆ Single-sign-on treated in 5.
  - 10.: Authentication-Control header
    - ◆ Extensions to make it usable with Web apps.
    - ◆ “application” peoples comments needed
  - 11.: Authentication Algorithms
    - ◆ All boring mathematics 😊
      - Crypto is now separated to another draft
    - ◆ “security” people’s comments needed
  - 12-16.: all finish-ups
    - ◆ IANA, security consideration, references etc.

# UI consideration

- Trusted display for mutual authentication result will be needed
  - We propose new UI for this auth scheme
    - ◆ Uses browser chrome area
    - ◆ Not a part of the draft, however





# Current status

---

- Spec draft: draft-oiwa-http-mutualauth-09
  - ...-algo-00
- Draft Implementations
  - Server-side: Apache, Ruby webrick
  - Client-side:
    - ◆ Mozilla-based implementation (Open-source)
    - ◆ Pure-Ruby reference implementation (to appear)
    - ◆ IE-based implementation (closed-source)
  - Available from project homepage:  
<https://www.rcis.aist.go.jp/special/MutualAuth/>
    - ◆ Trial website there!

# Thank you

---

## ■ More resources

- Our project homepage:

<https://www.rcis.aist.go.jp/special/MutualAuth/>

- Draft:

- ◆ Official: <https://datatracker.ietf.org/drafts/draft-oiwa-http-mutualauth/>
- ◆ Some preliminary drafts (before submission) on our homepage