Research Center for Information Security

Repairing HTTP authn. for Web security

- HTTP Mutual authentication proposal -

Yutaka OIWA (AIST Japan)

May 25, 2011

W3C Workshop on Identity in the Browser

AIST NATIONAL INSTITUTE OF ADVANCED INDUSTRIAL SCIENCE AND TECHNOLOGY (AIST)

Some Keywords yesterday and today…

- You can't get there from here directly
- Incremental adoption is important
- “Phishing is fun and profitable”
- Browsers should be an agent for user auth
- Bi-directional (mutual) authentication desired

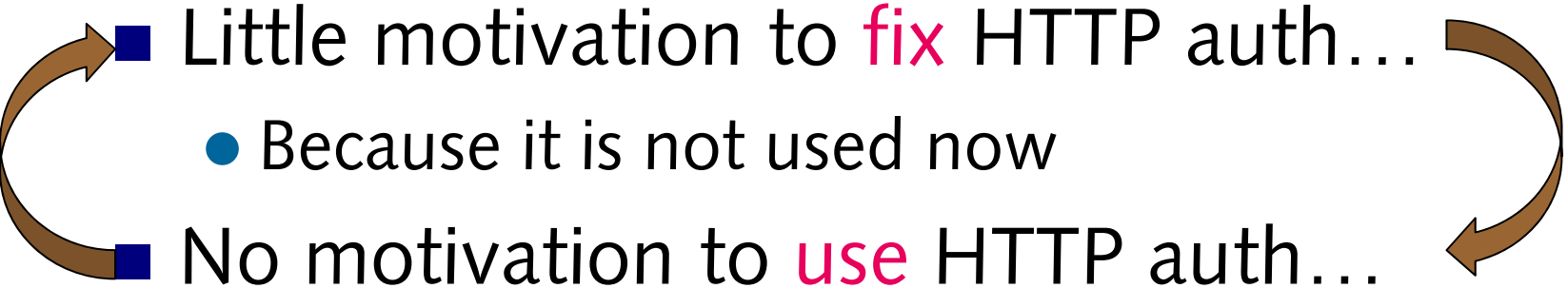
Problems so far (1)

- Form auth is insecure against forging!
  - Web pages have 100% control of behavior
    - Webpage script has full access to inputs
    - No measures introducible against phishing
      - Even if we had a "secure password field", phishers could always make a imitation using JavaScript
- HTTP auth: (only) *potentially* better
  - Browser will have a full control of auth process
    - It could protect user's passwords (e.g. Digest)
  - But…

Problems so far (2)

- HTTP auth is currently useless!
  - It is insecure now… Basic and Digest
  - More over, lacks applicability…
    - ugly modal dialog
    - no logout, no guest access
    - no session management possible

Chicken and egg problem…



- Little motivation to fix HTTP auth…
  - Because it is not used now
- No motivation to use HTTP auth…
  - Because it is hard to use
  - Because it is as insecure as Form

- We cannot fix Form auth…

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

Our proposal

- Password-based HTTP authentication which
  - Strongly protects the password from attackers
    - No eavesdropping, MITM, forwarding attack, etc.
    - Now "safe" to talk with Phishers! (no offline attack)
  - Provides *mutual* server-client authentication
    - Correct site & correct password ⇔ **auth success**
      - Phishing site || wrong password ⇔ **auth failure**
    - Users can make sure they talk to the "correct" site
      - "correct" := the site they have registered an account

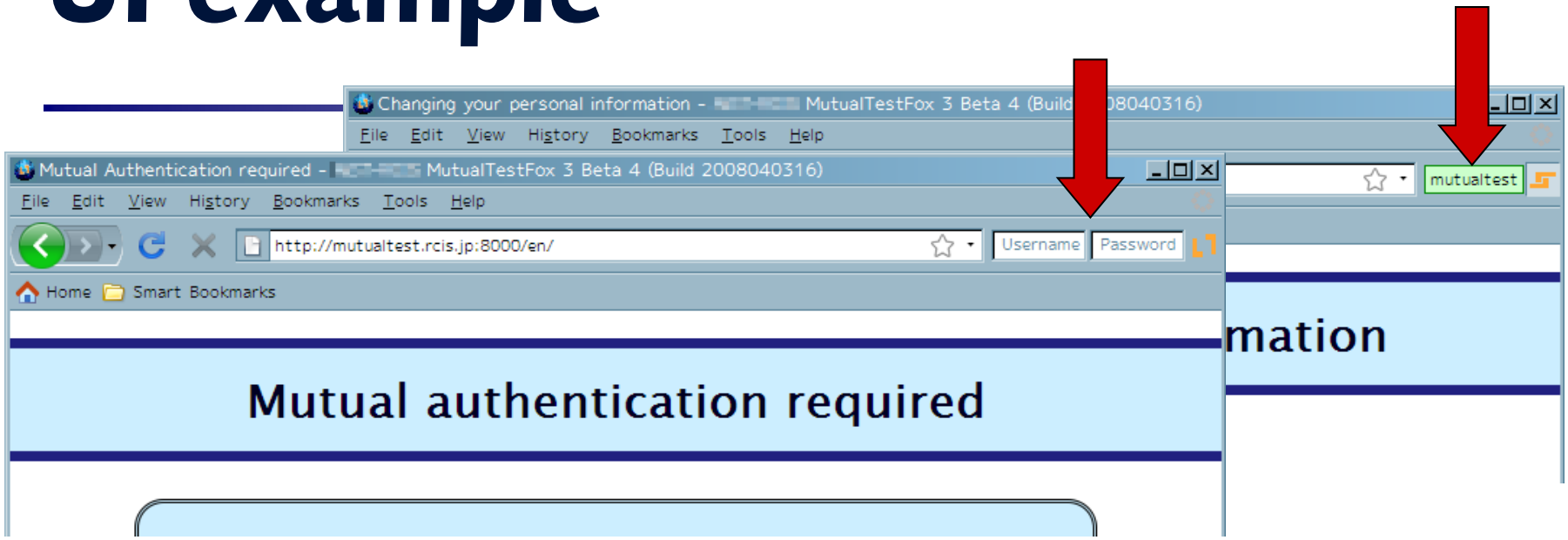
To overcome "usability" problem

- Support for recent Web application design
  - Non-modal authentication
  - Optional authentication
    - Guest users can be supported
  - Timed/server-initiated logout
  - log-on/log-off page redirection
- Gradually release possible
  - Coexist with Form auth. during transition period

UI issues

- Secure UI needed
  - To prevent password-stealing by imitation
  - Mutual auth result should be available to user
- "Non-modal" UI proposed
  - UI in a non-content (browser-controlled) area
  - not interrupting user's website experiences
    - Web site can design own log-in page
      - Except the input area itself
    - Guest page + login-UI is also possible

UI example

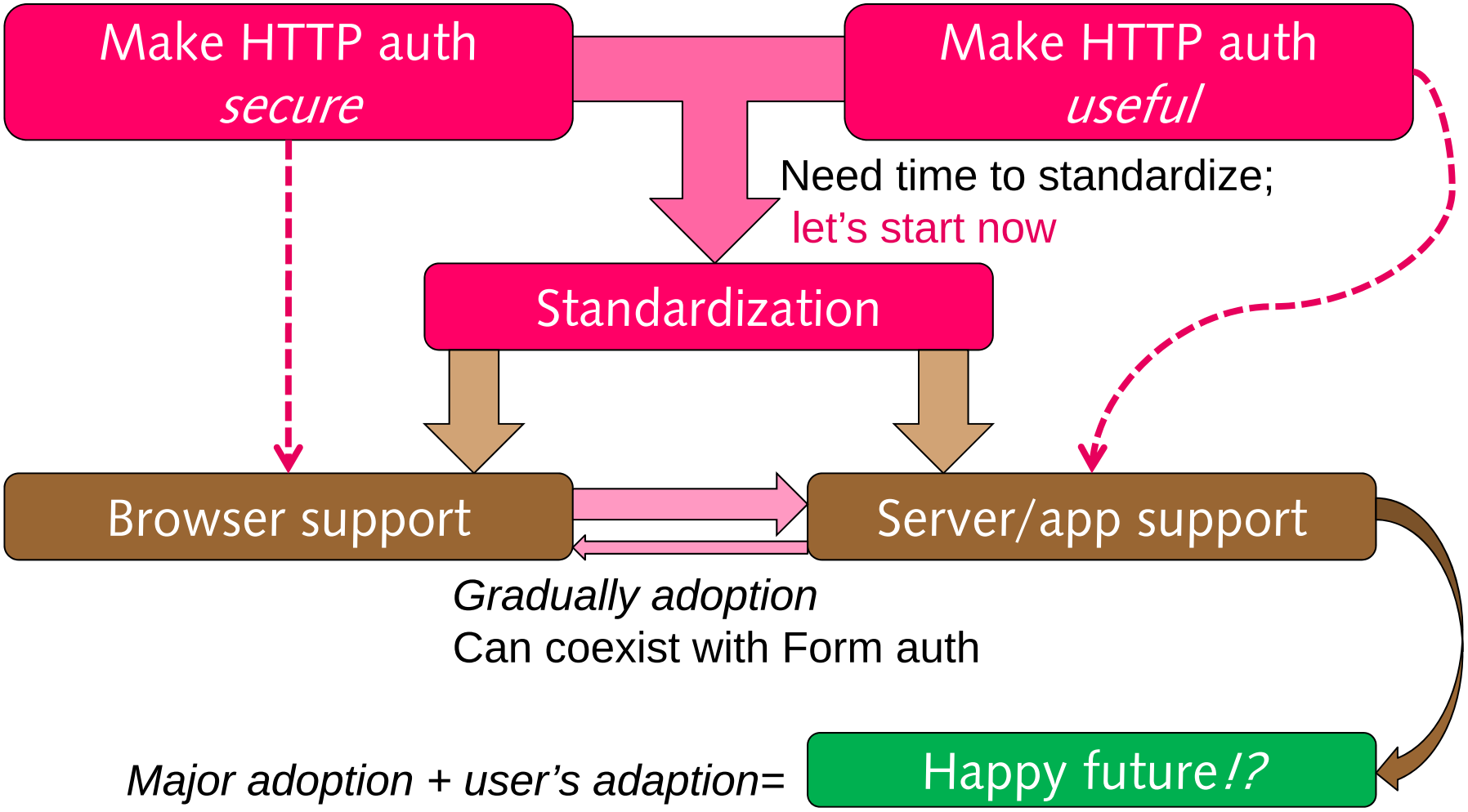


- Only "requirements" described in spec
  - Each browser will have an own UI
    - Can be integrated with local identity managements
  - Some "coordination" between browsers may needed
    - like padlock/RSS UI

Possible use cases

- As a stand-alone
  - Openly applicable to “any” website
- Combined with ID management
- With federated logins
  - Used for login to “initial” ID provider
    - Where “Phishing” will be a real problem

Our possible strategy



More resources

- Our project homepage: https://www.rcis.aist.go.jp/special/MutualAuth/
- IETF standardization effort
  - Mailing list http-auth @ ietf.org
  - **Need your assistance/involvement!**
- Draft:
  - Official: https://datatracker.ietf.org/drafts/draft-oiwa-http-mutualauth/
  - Some preliminary drafts (before submission) may be on our homepage