PAKE-based mutual HTTP authentication for preventing phishing attacks

Research Center for Information Security
National Institute of Advanced Industrial Science and Technology
Our Proposal

Our proposal
new mutual authentication protocol for Web systems against phishing attacks

Some of design goals are
- Secure,
- Easy to use, and
- Generic.

… details follow.
Design Goals

Secure

- detecting phishing websites reliably
  - Both users and servers are authenticated
- no password information leaks for false websites
- offline dictionary attack impossible

(↔DIGEST auth, PwdHash:
  >20 chars required for password secrecy)
Design Goals

Easy to use
- using human-memorable passwords only
- no need for personal secret storage
  (±TLS client auth., password reminders)

Generic
- no whitelist (±EV SSL)
- no blacklist (±IE/Firefox phishing warnings)
- not site-specific
Design Goals

Aiming for long-term solution:

- future replacement for form-based auth.

- Requires server modifications
  - Installation of the new authentication module.

- Requires client modifications
  - Browsers must be modified to support this algorithm.
Adopting PAKE for Web authentication
- Mutual auth. with weak secret (password)
- Password information is not leaked at all
  - Offline dictionary attack impossible

Naturally extending RFC2617
- Drop-in replacement for BASIC/DIGEST
- Replacement for form-based authentication in web applications
- Relying on TLS for secrecy of payload
  - Assume transport/DNS security

Host-name based detection of phishing
- avoiding man-in-the-middle phishing
Based on ISO-defined variant of PAKE protocol (ISO 11770-4 KAM3)

- Password is combined with hostname as “weak secret” to prevent MIM attack.
  \[ \pi = H(\text{password, host}) \]

- Computational cost similar to TLS
  - Single public-key op. for 1\textsuperscript{st} access
  - A few hash op. for 2\textsuperscript{nd} access & more
Some features

- **Session management**
  - reuse negotiated key for several requests
    - reducing computational overhead

- **“Optional” authentication**
  - Support guest/authenticated accesses in same URI
    - Like Yahoo! top-page and many other websites
Sample Communication

GET / HTTP/1.1
Host: www.example.com

HTTP/1.1 401 Authentication required
WWW-Authenticate: Mutual algorithm=iso11770-4-ec-p256,
validation=host, realm="Protected Contents",
state=0
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 5163

GET / HTTP/1.1
Host: www.example.com
Authorization: Mutual algorithm=iso11770-4-ec-p256,
validation=host, user=foobar,
w=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

HTTP/1.1 401 Authentication required
WWW-Authenticate: Mutual sid=yyyyyyyy,
w=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx,
nc-max=1024, nc-window=64, time=300, path="/"
Content-Length: 0

GET / HTTP/1.1
Host: www.example.com
Authorization: Mutual sid=yyyyyyyy, nc=0,
oa=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

HTTP/1.1 200 OK
Authentication-Info: Mutual sid=yyyyyyyy,
ob=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 7043

First request
(w/o authentication)

Key exchange phase

Authentication
confirmation phase
GET / HTTP/1.1
Host: www.example.com

HTTP/1.1 401 Auth. required
WWW-Authenticate: Mutual
  algorithm=iso11770-4-ec-p256,
  validation=host,
  realm="Protected Contents",
  stale=0
Content-Type: text/html;
  charset="ISO-8859-1"
Content-Length: 5163

First request without auth.
- 401 response
  (as in usual HTTP auth.)
- Specify crypto. algorithm
  from server
- Content body displayed
  - Requesting login
- ID/Pass requested
GET / HTTP/1.1
Host: www.example.com

HTTP/1.1 200 OK
Optional-WWW-Authenticate: Mutual
   algorithm=iso11770-4-ec-p256,
   validation=host,
   realm="Protected Contents",
   stale=0
Content-Type: text/html;
   charset="ISO-8859-1"
Content-Length: 5163

Variant: 200 + optional auth.
    New introduction in ours

“Guest Contents” displayed with Optional-WWW-auth header

    User can continue to be a guest
    Or
    User can enter ID/Pass to login to the site
GET / HTTP/1.1
Host: www.example.com
Authorization: Mutual
    algorithm=iso11770-4-ec-p256,
    validation=host, user=foobar,
    wa=xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    xxxxxxxxxxxxxxxxxxxxxxxxxxx

HTTP/1.1 401 Authentication required
WWW-Authenticate: Mutual
    sid=yyyyyyyy,
    wb=zzzzzzzzzzzzzzzzzzzzzzzzz
    zzzzzzzzzzzzzzzzzzzzzzzzzz,
    nc-max=1024, nc-window=64,
    time=300, path="/
Content-Length: 0

- PAKE Key exchange ongoing
  - Client/server exchanges key materials wa & wb, twisted by password-based weak secrets

- Session ID (sid) established

- Temporary key shared between client & server, *only if the weak secrets are generated from the same password*
Final authentication

GET / HTTP/1.1
Host: www.example.com
Authorization: Mutual
  sid=yyyyyyyy, nc=0,
  oa=wwwwwwwwwwwwww

HTTP/1.1 200 OK
Authentication-Info: Mutual
  sid=yyyyyyyy,
  ob=vvvvvvvvvvvvvv
Content-Type: text/html;
  charset="ISO-8859-1"
Content-Length: 7043

- Checking proper key exchange using shared-key and hash functions
- Both clients and servers are authenticated (oa, ob)
- Client MUST check the validity of ob
GET /logo.png HTTP/1.1
Host: www.example.com
Authorization: Mutual
  sid=yyyyyyyy, nc=1,
  oa=uuuuuuuuuuuuuuuuuuuuuuuu

HTTP/1.1 200 OK
Authentication-Info: Mutual
  sid=yyyyyyyy,
  ob=ssssssssssssssssssssssss
Content-Type: image/png
Content-Length: 15082

- Shared key can be reused for multiple requests
- Nonce prevents replay
- Number of reuse are limited by nonce counter limit
UI consideration

- Entry field must be protected from image-based forgeries
  - no popup dialog (↩ BASIC/DIGEST auth.)
  - e.g. use toolbar area (see above)

- Auth. status must be indicated
  - to prevent imitated auth. success
Current status

- Plugin for Apache server implemented
- Test Firefox extension implemented
  - Full implementation is about to start
- Internet-Draft in preparation
Future Plans

- Field test in a part of Yahoo! Japan
- Distribution of open-source modules
- Submitting Internet-Draft
Related Work

- EV-SSL … relies on central authorities
- Passpet … requires private key storage
- PwdHash
  - Similar hostname-based mangling
  - Weak against offline attacks
Page Transition Examples
Websites not supporting authentications

Status: 200 OK
No login field displayed
Visiting Mutual-auth sites as guest

- Status: 200 OK
- Optional-WWW-Authenticate: Mutual ...
- ID/Password box appears
Switching Tabs

Login field displayed/hidden, according to the status of each tab
Login!

Enter ID/password, click the button…
Login...

Waiting for authentication response
Logged in

Status: 200 OK
Authentication-info: Mutual ……
ID indicated with green background
Switching Tabs

Login status display changes still according to the status of each tab
Logout

Click button to logout
Logout

Status: 200 OK

Optional-WWW-Authenticate: Mutual ...

Contents reloaded, back to the guest state