SessionLock
Securing HTTP Sessions from Eavesdroppers

Ben Adida
Harvard University

WWW 2008 - Beijing, China
24 April 2008
Can we improve web security without upgrading the browser?
wi-fi typically unencrypted, easy to sniff.
Web Browser

Web Server

SetCookie: uid=12345; tok=87654

Cookie: uid=12345; tok=87654

/login?username\&password
Web Browser

Web Server

SetCookie: uid=12345;tok=87654

Cookie: uid=12345;tok=87654

/login?username&password

/inbox

Cookie: uid=12345;tok=87654
Web Browser

Web Server

/login?username&password

SetCookie: uid=12345;tok=87654

/inbox

Cookie: uid=12345;tok=87654

contents of inbox

Cookie: _----- uid=12345;tok=87654

uid=12345;tok=87654
Web Browser

Web Server

SetCookie: uid=12345; tok=87654

Cookie: uid=12345; tok=87654

Eve

SetCookie: uid=12345; tok=87654

Cookie: uid=12345; tok=87654

/inbox

contents of inbox
Web Browser

Web Server

SetCookie: uid=12345; tok=87654

Cookie: uid=12345; tok=87654

/inbox

Contents of inbox

Cookie: uid=12345; tok=87654

Eve

/send?to=billg

Cookie: uid=12345; tok=87654

/username&password

uid=12345; tok=87654

contents of inbox
SSL?

- Content Distribution Networks
- All-or-Nothing Content
- All-or-Nothing Download
- Caching
SSL?

- Content Distribution Networks
- All-or-Nothing Content
- All-or-Nothing Download
- Caching

**Even with server-side horsepower, SSL is significantly more sluggish.**
Encryption is not the Issue

- add a few lines to a wiki
- check a few notifications on gmail
- send a few pages to my bookmarks aggregator
Encryption is not the Issue

- add a few lines to a wiki
- check a few notifications on gmail
- send a few pages to my bookmarks aggregator

Risk: complete impersonation!
Encryption is not the Issue

- add a few lines to a wiki
- check a few notifications on gmail
- send a few pages to my bookmarks aggregator

Risk: complete impersonation!

Shoulder-surfing my email is not the end of the world. Impersonating me to my webmail is much worse.
Can We Do Something **Now?**
Can We Do Something Now?

HTTP
Can We Do *Something* Now?
Can We Do *Something* Now?
Can We Do *Something* Now?

- Application Code
- HTML & JavaScript
- HTTP
Can We Do *Something* Now?

- The web is a (limited) platform

- Application Code
- HTML & JavaScript
- HTTP
Can We Do *Something* Now?

- The web is a (limited) platform
- Can we build better security in the application layer?
Can We Do *Something* Now?

- The web is a (limited) platform
- Can we build better security in the application layer?
- Maybe by hijacking certain features for security purposes? (Active Cookies, Subspace, BeamAuth...)
Can We Do *Something* Now?

- The web is a (limited) platform
- Can we build better security in the application layer?
- Maybe by hijacking certain features for security purposes? (Active Cookies, Subspace, BeamAuth...)

Goal: “good enough” security for basic sessions
Current Login Approach

Web Server

Web Browser
Current Login Approach

Web Browser

Web Server

/login?username&password

Set-Cookie

Cookie:

uid=12345;tok=87654

/inbox

contents of inbox
The SessionLock Idea
The **SessionLock** Idea

- Use the SSL login to send a **SECRET** to the web browser.
The SessionLock Idea

- Use the SSL login to send a SECRET to the web browser.
- “Pass” SECRET to the non-SSL side.
The **SessionLock** Idea

- Use the SSL login to send a **SECRET** to the web browser.
- “Pass” **SECRET** to the non-SSL side.
- Keep **SECRET** in scope, use it to sign every HTTP request.
The **SessionLock** Idea

- Use the SSL login to send a **SECRET** to the web browser.
- “Pass” **SECRET** to the non-SSL side.
- Keep **SECRET** in scope, use it to sign every HTTP request.
- Never send **SECRET** in plaintext.
Passing via URL Fragment

http://site.com/page#paragraph
Passing via URL Fragment

http://site.com/page#paragraph

- used to designate a portion of a page
  browser scrolls to the appropriate location.
Passing via URL Fragment

http://site.com/page#paragraph

- used to designate a portion of a page browser scrolls to the appropriate location.

- never sent over the network but accessible from JavaScript.
Passing via URL Fragment

http://site.com/page#paragraph

- used to designate a portion of a page
  browser scrolls to the appropriate location.

- *never sent* over the network
  but accessible from JavaScript.

- navigation between fragments
  does not cause a page reload.
Passing Secret via Fragment

http://webmail/#SECRET

[BeamAuth, Webkeys]
Passing Secret via Fragment

http://webmail/#SECRET

```javascript
var hash = document.location.hash;

if (hash != '') {
    // parse the hash, get username and token
    store_secret(hash);

    // clear the hash from the URL
    document.location.replace('/');
}
```

[BeamAuth, Webkeys]
Signing with HMAC
Signing with HMAC

- Assume a single-page web application (for now).
Signing with HMAC

- Assume a single-page web application (for now).
- Intercept all XHR JavaScript requests, append timestamp, HMAC the request line, append HMAC result.
Signing with HMAC

- Assume a single-page web application (for now).
- Intercept all XHR JavaScript requests, append timestamp, HMAC the request line, append HMAC result.
- 50ms on a 3-year-old laptop for each request (much faster on recent laptop with FF3.)
Web Browser

Web Server

Cookie: bd5c

302 /mail#[ac8f]

/login?username&password
Web Browser

Web Server

Cookie: bd5c

302 /mail#[ac8f]

/launch

/login?username&password
Web Server

Web Browser

Cookie: bd5c

Load #[ac8f] from fragment and store in JavaScript scope.

/load?username&password

302 /mail#[ac8f]

/launch

JS+HTML interface
Load #[ac8f] from fragment and store in JavaScript scope.
Load #[ac8f] from fragment and store in JavaScript scope.
Load #[ac8f] from fragment and store in JavaScript scope.
Recovery

Web Server

/launch

JS+HTML interface

Cookie: bd5c

Load #[ac8f] from fragment and store in JavaScript scope.
Recovery

Cookie: bd5c

302 /mail#[ac8f]

Cookie: ac8f

Load #[ac8f] from fragment and store in JavaScript scope.
Recovery

Web Browser

Cookie: bd5c

Load #[ac8f] from fragment and store in JavaScript scope.

Web Server

302 /mail#[ac8f]

/launch

JS+HTML interface

/inbox?date&HMAC

list of emails in inbox

Cookie: ac8f

/load?username&password
Recovery

http://webmail/

missing secret token
Recovery

open dynamic iframe

http://webmail/

https://webmail/recover

tok = cookie('tok');
location =
'http://webmail/recover#' + tok;
Recovery

http://webmail/

tok = get_fragment();
window.parent.
set_tok(tok);

recover token
Recovery

http://webmail/

token recovered, close frame, make AJAX requests
Multi-Page Apps

Web Server

Web Browser
Multi-Page Apps

Web Server

/login?username&password

302 /mail#ac8f
Multi-Page Apps

Web Browser

Web Server

/login?username&password

302 /mail#[ac8f]

/launch

Cookie: bd5c
Multi-Page Apps

Web Server

Web Browser

Cookie: bd5c

/login?username&password

302 /mail#[ac8f]

/launch

launch page

# [ac8f]
Multi-Page Apps

Web Server

Web Browser

Cookie: bd5c

Load #[ac8f] from fragment use to HMAC and patch links.
Multi-Page Apps

Web Server

/login?username&password

302 /mail#[ac8f]

/load

launch page

/inbox?date&HMAC

Cookie: bd5c

Load #[ac8f] from fragment use to HMAC and patch links.
Multi-Page Apps

Web Server

Cookie: bd5c

Web Browser

/load?username&password

302 /mail#[ac8f]

/launch

launch page

/inbox?date&HMAC

Load #[ac8f] from fragment use to HMAC and patch links.
Multi-Page Apps

Web Server

Web Browser

Cookie: bd5c

Load #[ac8f] from fragment use to HMAC and patch links.
Multi-Page Apps

Web Server

Web Browser

Cookie: bd5c

Load #[ac8f] from fragment used to patch links with HMAC.

Load #[ac8f] from fragment use to HMAC and patch links.
Better Page-to-Page Transfer & Recovery

/*
 * Browser-Local Storage
 */

// store the token as soon as it is obtained.
globalStorage['webmail.com'].session_key = '...';

// On opening up a new browser window, get token.
var session_key = globalStorage['webmail.com']
Summary
Summary

- fragment identifier as a client-only channel between HTTPS and HTTP.
Summary

- fragment identifier as a client-only channel between HTTPS and HTTP.

- AJAX single-page application is easy.
Summary

- fragment identifier as a client-only channel between HTTPS and HTTP.
- AJAX single-page application is easy.
- Multi-page apps:
Summary

- fragment identifier as a client-only channel between HTTPS and HTTP.
- AJAX single-page application is easy.
- Multi-page apps:
  - keep the secret around by frag id.
Summary

- fragment identifier as a client-only channel between HTTPS and HTTP.
- AJAX single-page application is easy.
- Multi-page apps:
  - keep the secret around by frag id.
  - small JS library that patches HTML.
Summary

- fragment identifier as a client-only channel between HTTPS and HTTP.

- AJAX single-page application is easy.

- Multi-page apps:
  - keep the secret around by frag id.
  - small JS library that patches HTML.
  - writing your app with SessionLock in mind.
Summary
Summary

Intuition matches threat model more closely.

What you browse is visible to an eavesdropper. What you don’t browse is safe.

Actions with side-effects are safe.
Questions?

ben_adida@harvard.edu

http://ben.adida.net/projects/sessionlock/