INTERNET STANDARDS AND WEB SECURITY

Jeff Hodges
PayPal
August 2012
The Current Web Has Some Holes

(some) Help is On The Way

Venues

What can you do to help?

[ includes bonus references on the last slide! ]
The Current Web Has Some Holes
Sort of Like This:
IT IS HARD TO DO EVEN SIMPLE THINGS SAFELY

- Include an ad on your site
- Use third-party Site-Analytics
- Allow user input (“Rich” or otherwise)
- Uniform use of HTTPS
WHAT ARE SOME OF THE HOLES?

- Cross Site Request Forgery (CSRF)
- Cross Site Scripting (XSS)
- Clickjacking
- Malvertising
- TLS/SSL Man In The Middle (MITM)
  - For example - sslstrip
WHY DO THESE ATTACKS EXIST?

- Core protocol/technology weaknesses
- Too much required of each and every developer
- Lack of Security Policy Mechanisms
Cookies are broken:
  - Their scope rules are broken
  - “Secure” Flag doesn’t really mean the same thing everywhere
  - “HTTPonly” and “Secure” only partially effective
  - Network MiTM attacker can overwrite cookies by spoofing..
    http://www.example.com
    ..to overwrite real “secure cookies” for..
    https://www.example.com
  - Practically anything can be interpreted as JavaScript
  - Browsers default to HTTP first (Not HTTPS)
• To Implement a “Strong” Security Policy……

• Every Cookie has to have HTTPOnly and Secure Flag

• Every link generated has to have the right scheme (HTTP vs. HTTPS)

• Every page must have the right content encoding

  • This is TOO HARD
A Developer or WebSite Administrator has *no coherent way* to say, for example:

- Treat all my cookies “Securely”,
- Only load HTTPS Content,
- And don’t frame my site.
(Some) Help Is On The Way
“RECENT” WEB SECURITY STANDARDS

- Cookies aka “HTTP State Management” [RFC6265]
- The Web Origin Concept [RFC6454]
- X-Frame-Options (de-jure)
- Cross Origin Resource Sharing (CORS) [W3C]
  - Still a “working draft”
“EMERGING” WEB SECURITY STANDARDS

- HTTP Strict-Transport-Security (HSTS)  [IETF I-D]
- Content Security Policy (CSPv1)  [W3C WD]
- Content Sniffing controls
  - E.g., X-Content-Type-Options: nosniff
CSPv1.1

- Adds new directives:
  - Form-action, Script-nonce, plugin-types, frame-options (so far)
  - Frame-Options directive intended as successor to x-frame-options

- User Interface Safety directives for CSP
  - More fine-grained framing control with input protections from click, keypress, touch, and drag events
LEVERAGABLE, EMERGING SECURITY STANDARDS

- Secure DNS (aka DNSSEC)
- TLSA (aka DANE) [RFC6698]
Venues for the Foregoing Work
VENUES FOR THE FOREGOING WORK

• We actively contributed to the creation of, and participate in, these working groups:

• IETF HTTP State Working Group
  • Dec-2009 to May-2011
  • RFC6265 “HTTP State Management Mechanism”
  • (The WG successfully achieved its goal and was closed)

• IETF WebSec Working Group (Feb 2011)
  • Web Origin [RFC6454]
  • HTTP Strict Transport Security (HSTS) [RFC-to-be-soon]
  • X-Frame-Options (a real spec, retroactively)
  • Web Security Framework Requirements
VENUES FOR THE FOREGOING WORK CONT’D

- **W3C WebAppSec** (Jun 2011) and WebApps Working Groups
  - Key specs underway:
    - CORS “Cross Origin Resource Sharing”
    - CSP “Content Security Policy”
    - UI Safety

- Related:
  - IETF DANE WG
    - Now working on applying these techniques to other protocols, e.g., IPSEC
• Use of third-party script
  • E.g., include arbitrary ads on your site or third-party Site-Analytics
  • Note: (draft) ECMAScript v5 “strict mode” plus Caja is a promising solution here
• Allow arbitrary user input (“Rich” or otherwise)
• Automatic uniform use of HTTPS
• A Coherent Web Security Policy framework
  • Still inventing new one-off HTTP headers for specific issues, e.g., HSTS
  • CSP is step in right direction, e.g., in how the UI Safety spec leverages CSP for policy conveyance
SOME GOALS FOR APPROACHES

• Should not rely on every developer (and user) “getting it right” 100% of the time

• Security mechanisms should be “declarative policy and configuration”
  • separate from “code”

• Reduce the need for new individual HTTP headers for each specific issue

• Overall – create security mechanisms that allow/enforce the concept of Least Privilege
Common Security User-Interfaces
- Browsers presently display security issues differently
- Also have differing approaches to dealing with issues
- Is an area of active research & experimentation so standardizing is perhaps premature (W3C eventually?)

Fixing the Certificate Authority (CA) Situation
- Multitude of CAs in browser & OS “Trust Anchor Repositories (TARs)”
- All trusted equally
- Each can certify any domain name
- Large attack surface
- Whither the CA/Browser Forum?
- BOF session at IETF-85 Atlanta (Nov 2012) on “Web PKI Operations” (WPKops) ? (a step in a useful direction)
WHAT CAN YOU DO TO HELP?

• Participate in the IETF and W3C Working Groups, and other such cross-industry orgs

• Deploy your website uniformly via HTTPS

• Use HSTS and CSP in your web application

• Provide feedback to the working groups
QUESTIONS?

For more details:

- **The Need for Coherent Web Security Policy Framework(s)**
- W3C Web App Security Working Group
- IETF WebSec Working Group
  [https://datatracker.ietf.org/wg/websec/charter/](https://datatracker.ietf.org/wg/websec/charter/)
- IETF DANE (TLSA et al) Working Group
  [https://datatracker.ietf.org/wg/dane/charter/](https://datatracker.ietf.org/wg/dane/charter/)
- WPKops (non-working group, exploratory) mailing list
  [https://www.ietf.org/mailman/listinfo/wpkops](https://www.ietf.org/mailman/listinfo/wpkops)

- Jeff Hodges ([Jeff.Hodges@paypal.com](mailto:Jeff.Hodges@paypal.com))
- Brad Hill ([bhill@paypal.com](mailto:bhill@paypal.com))
- Andy Steingruebl ([asteingruebl@paypal.com](mailto:asteingruebl@paypal.com))