Agenda

• Declarative Web Security
  • Content Security Policy
  • HTTP Strict Transport Security

• Addressing the Web's top threats (spoiler: these aren't solved problems)
  • Outdated plugins
  • CSRF
  • Clickjacking
  • Better privacy support
Landscape of Threats

• The web faces a host of well-known but persistent threats
  • XSS
  • CSRF
  • MITM
  • Phishing
  • Overlay (“clickjacking”)

• Developers are aware of threats and mitigation strategies

• Rates of regression and bug discovery remain stable

• **Declarative security** mechanisms hold promise for reliable attack mitigation
Content Security Policy

• Addresses the threat of content injection, e.g. XSS:

• Fundamental problem:
  • Web client treats all content in server response with equal privilege
  • No way to differentiate legitimate content from injected content

• CSP provides a mechanism for sites to explicitly state which content is legitimate
  • Everything else can be dropped on the floor
A Line in the Sand...

- Script must come from external files served from white-listed hosts
  - No inline JavaScript, e.g. internal `<script>` nodes, `javascript:` URIs, event handling attributes

- No code from strings, a.k.a. `eval` is evil
  - Strings easily tainted by attacker-controlled data

- Only explicitly allowed content will load
  - Policies can be separately defined for other types of content too: images, audio/video, plugin content, stylesheets, etc.
Content Security Policy Directives

- allow
  - catch-all for unspecified content types
- img-src
- media-src
- script-src
- object-src
- frame-src
  - "what can be embedded here?"
- frame-ancestors
  - "what sites may embed me?"
- style-src
- report-uri
- policy-uri

Content Sources:

- Host expression
  - Hostname plus optional scheme and port
  - Wildcards are valid, e.g. *.example.com
- Keywords: 'self', 'none'

Example:

- X-Content-Security-Policy: allow 'self'
  my-cdn.com; frame-src ads.net;
  frame-ancestors 'self'
Content Security Policy – Side Benefits

• Clickjacking Protection
  • `frame-ancestors` policy allows site to specify where a resource may be embedded
  • Frame-busting not as effective as once thought

• Violation Reporting
  • `report-uri` – “Canary in the coal mine” – get notified when policy violations occur
  • Report-only mode also available
Content Security Policy Demo

- WebGoat protected by CSP
More information on Content Security Policy

• Getting started
  • http://mz.la/csp-info

• New in Firefox 4 (Early 2011)
  • Grab a beta: http://mz.la/csp-demo

• W3C standard planned
  • http://www.w3.org/2010/07/appsecwg-charter
  • Proposed spec: http://mz.la/csp-spec

• Mozilla looking for launch partners
  • We're willing to help! (bsterne@mozilla.com, @bsterne)
HTTP Strict-Transport-Security

- Addresses the threat of **Man-in-the-middle** attacks
  - Packet sniffing, session hijacking
  - See also: recent **Firesheep** controversy

- Fundamental problem:
  - Web sites don't fully implement SSL/TLS
  - Session tokens are passed insecurely

- HTTP Strict Transport Security allows sites to force all connections to be made over SSL/TLS
  - Insecure requests are automatically rewritten
HTTP Strict Transport Security

• Specification:

• Examples:
  • Strict-Transport-Security: max-age=60000
  • Strict-Transport-Security: max-age=60000; includeSubdomains

• Firefox 4
  • Fully implements the spec
  • Add-on for better UI: STS UI

• Firefox 3.6
  • Implemented by Force-TLS and NoScript add-ons
Plugin Checking Service

- Addresses the threat of outdated plugins
- **Major** source of security and stability risk for users
- Provides a way for users to see if their plugins are up to date
Plugin Checking Service

• Top 24 plugins currently checked – more being added

• Plugin check webpage also works in Safari 4, Chrome 4, and Opera 10.5

• Longer term the service will be integrated into Firefox
  • Updating process varies widely between plugins → confusing to users
  • Plugin vendors will have “self-service” panel for updating new versions as they are released
Fixed: CSS History Sniffing Attack

• Addresses the threat of **browser history leakage** via CSS

• Long and well-understood issue
  • [http://jeremiahgrossman.blogspot.com/2006/08/i-know-where-youve-been.html](http://jeremiahgrossman.blogspot.com/2006/08/i-know-where-youve-been.html)
  • [http://ha.ckers.org/weird/CSS-history-hack.html](http://ha.ckers.org/weird/CSS-history-hack.html)

• Fixed in **bug 147777**
  • Limited which properties can be styled using `:visited`
  • `GetComputedStyle()` “lies” to the webpage
Fixed: CSS History Sniffing Attack

Firefox 3.6

CSS History Hack
Originally found here but permanently hosted on ha.ckers.org with Jeremiah's permission.

Ha.ckers.org home || Jeremiah's blog
Firefox Only! (1.5 - 2.0) tested on WinXP.

Visited
- http://mail.google.com/
- http://mail.yahoo.com/
- http://www.amazon.com/
- http://www.bankofamerica.com/
- http://www.cgisecurity.com/
- http://www.chase.com/
- http://www.cm.com/
- http://www.ebay.com/
- http://www.etrade.com/
- http://www.flickr.com/
- http://www.google.com/
- http://www.yahoo.com/

Not Visited
- http://ha.ckers.org/blog/
- http://login.yahoo.com/
- http://my.yahoo.com/
- http://sla.ckers.org/forum/
- http://slashdot.org/
- http://www.aol.com/
- http://www.bankofamerica.com/
- http://www.blackhat.com/
- http://www.blogger.com/
- http://www.bofa.com/
- http://www.capitalone.com/
- http://www.cgisecurity.com/
- http://www.chase.com/
- http://www.citybank.com/
- http://www.cm.com/
- http://www.comerica.com/
- http://www.e-gold.com/
- http://www.oval.com/
- http://www.etrade.com/
- http://www.flickr.com/
- http://www.google.com/
- http://www.hsbc.com/
- http://www.icyq.com/
- http://www.live.com/
- http://www.microsoft.com/

Firefox 4

CSS History Hack
Originally found here but permanently hosted on ha.ckers.org with Jeremiah's permission.

Ha.ckers.org home || Jeremiah's blog
Firefox Only! (1.5 - 2.0) tested on WinXP.

Visited

Not Visited
- http://ha.ckers.org/blog/
- http://login.yahoo.com/
- http://mail.google.com/
- http://mail.yahoo.com/
- http://my.yahoo.com/
- http://sla.ckers.org/forum/
- http://slashdot.org/
- http://www.aol.com/
- http://www.bankofamerica.com/
- http://www.blackhat.com/
- http://www.blogger.com/
- http://www.bofa.com/
- http://www.capitalone.com/
- http://www.cgisecurity.com/
- http://www.chase.com/
- http://www.citybank.com/
- http://www.cm.com/
- http://www.comerica.com/
- http://www.e-gold.com/
- http://www.oval.com/
- http://www.etrade.com/
- http://www.flickr.com/
- http://www.google.com/
- http://www.hsbc.com/
- http://www.icyq.com/
- http://www.live.com/
- http://www.microsoft.com/
- http://www.msn.com/
- http://www.myspace.com/
- http://www.nn.com/
Looking Forward

• Prioritize list of biggest security threats to Web
  • Your input is invaluable
  • mozilla.dev.security, security@mozilla.org

• Always a spectrum of solutions
  • Compatibility/Usability ↔ Security

• How much can we reasonably break?
Top Threat: CSRF*

• Huge percentage of sites are vulnerable
  • Conservatively 21% per WhiteHat Fall 2010 report

• Mitigation strategies are apparently hard to implement

• One complete solution: RequestPolicy add-on
  • Breaks the Web for most users
  • Workaround: ship with known-good policy configurations and crowd source the rest

* author's humble opinion
CSRF – Another solution

• Origin header
  • Privacy improvements over Referer header
  • http://tools.ietf.org/html/draft-abarth-origin-00
  • “The user agent MAY include an Origin header in any HTTP request.”
  • Implemented in Chrome, only sent with POST
  • Browsers will send null from “privacy-sensitive contexts”

• Web is fraught with state-changing GETs
• Still requires servers to do the right thing with Origin data
CSRF – Fixing part of the problem

• Intranet hacking
  • Yes, an old and well-understood issue
    • “Hacking Intranet Websites...” - Grossman, Black Hat 2006
  • Not a trivial fix
    • Security context not always available (a common theme)
    • Web proxies
• 95% fixed in bug 354493
  • Ironing out testing infrastructure
Clickjacking

- Browsers have started to provide solutions
  - X-Frame-Options
  - CSP frame-ancestors

- Incomplete Solutions
  - Does prevent framing, does not prevent stolen mouse clicks
  - Sites want to be framed across domains, they just don't want to be clickjacked
Clickjacking – Potential Solution

• Prevent obfuscated elements from being clicked

• Heuristics are hard
  • Can't force an iframe to be unconditionally on top, could break the embedding site's layout
  • A strong definition of “clickable” would be a good start; force such elements to be 100% opaque and on top
  • Obscured by small size, similar background, etc.

• NoScript attempts to implement these heuristics
  • “Partially obstructed, transparent or otherwise disguised” elements are revealed before interaction
Privacy Improvements

• Anonymous Browsing Mode
  • Different from Private Browsing (protect against local attacker)
  • Minimize amount of identifying data sent to servers
    • Prevent tracking and fingerprinting (see Panopticlick)
  • Do everything Torbutton add-on does natively in the browser
• Exploratory work:
  https://wiki.mozilla.org/Security/Anonymous_Browsing
Privacy Improvements

• Double-keyed cookies
  • Third party cookies are only sent from the same embedding context
    • The Doubleclick cookie you got on example.com only gets sent when you're on example.com
  • Work in progress
    • https://wiki.mozilla.org/Thirdparty
    • 3rd party cookies downgraded to session \o/ (bug 565475) but pref'd-off (bug 570630) ¯\_\_\_¯
    • Key cookies by embedding context (bug 565965)
    • Political groundwork needed
Thank You