WebAppSec WG Update

TPAC 2015

Brad Hill

Scope Expansion

- **2013 had 3 Rec-track documents under active development**
- Content Security Policy
- User Interface Security
- Mixed Content

- **2015 has 13 Rec-track documents under active development**
- Content Security Policy Level 2
- User Interface Security
- Mixed Content
- Subresource Integrity
- Referrer Policy
- Secure Contexts
- CSP Pinning
- Upgrade Insecure Requests
- The Permissions API
- Credential Management Level 1
- Entry Point Regulation
- Clear Site Data
- Confinement with Origin Web Labels (COWL)

- **Plus 3 More Planned / Potential Rec-track documents:**
- Suborigin Namespaces
- CSP Embedded Enforcement
- https-transitional

Broad Themes

- **Enlisting the User Agent in Cooperative Policy Enforcement**
- Content Security Policy Level 2
- User Interface Security
- Subresource Integrity
- CSP Pinning
- Entry Point Regulation

- **Standardizing and Enabling HTTPS for Web Applications**
- Mixed Content
- Secure Contexts
- Upgrade Insecure Requests
- Referrer Policy

- **Security Related APIs**
- The Permissions API
- Credential Management Level 1
- Clear Site Data

- **Experiments in the Web Security Model / Same Origin Policy**
- Confinement with Origin Web Labels (COWL)
- Suborigin Namespaces

Rec-Track Highlights

Content Security Policy

- **Candidate Recommendation**
- 99.9% complete implementation of Level 2 in Chrome
- 95% in Firefox
- Level 1 support in Edge and Safari

An HTTP header or <meta> tag which allows restricting from where content may be loaded into a resource as well as other capabilities of the resulting document.

Developer adoption has been the major challenge of CSP. Although Chrome telemetry shows that CSP is used on over 16% of pageloads, making it one of the most popular features of the web platform, that is heavily driven by mandatory use in extensions and a few of the most popular websites in the world. In the long tail, CSP is used by only ~2% of the Alexa Top 1M websites.

To ease this developer adoption burden, key features of Level 2 included the ability to whitelist legacy inline content by hash or nonce.

Level 3 will target greater modularity and ability to cooperate with templating engines for a better developer experience.

Subresource Integrity (SRI)

- **Approximately 1 week from Candidate Recommendation**
- Complete implementations available in both Firefox and Chrome

Discussed for many years, SRI gives the ability to specify a hash attribute for script and CSS subresources. A compliant user agent will refuse to load a resource that does not match the expected content.

Primary use case today is to reduce trust in CDNs. Push more content to edge-caches in potentially untrusted POPs; mitigate some security "single points of failure" for much of the Web (e.g. the jQuery CDN)

Not a content-addressable cache (yet) – privacy and security issues still to be worked out. Also not applicable to downloads (yet) as UI issues still to be worked out.

We plan to learn from deployment of this very constrained feature set how the security / brittleness tradeoffs work out before expanding the ambition of the work.

Referrer Policy

- **First Public Working Draft**
- Relatively complete implementations available in both Firefox and Chrome, some features in Safari

Standardization and enhancement of behavior originally specified in WHATWG as <meta> referrer.

Allows control of outbound HTTP Referer (sic) header information.

Helps preserve privacy and security for capability URLs.

Of great use for link shortening / shimming services, as they can protect user privacy (e.g. of search results in the query string) and send Origin-granularity referrer information from https->http without requiring an additional redirect. Between deployment on Facebook, Google and Twitter, tens of billions of unnecessary redirects have already been eliminated since deployment earlier this year.

Mixed Content & Secure Contexts

- **Candidate Recommendation**
- Reflects longstanding behavior in most major user agents, w/recent adoption by Safari

Standardization and enhancement of widely-established behavior to block mixed content, setting clear, interoperable expectations for developers about how secure document environments will behave with regard to mixed content blocking or blocking of sensitive features in insecure contexts.

Upgrade Insecure Requests

- **Candidate Recommendation**
- Implemented by Chrome and Firefox, experimental deployment on w3.org

Eases the burden of sites wishing to enable HTTPS by instructing the user agent to transparently upgrade insecure references instead of blocking by default.

The group has worked with the W3C to determine what obstacles have prevented its own adoption of HTTPS, and hope that this will remove some of those for our own and other organizations. We would like to see W3C be a leader, not a laggard, in delivering a secure and trustworthy experience to our community.

Potential future work which may mesh with this includes a proposal for an “https-transitional” protocol mode that would allow “in-place” upgrade of URLs with an http scheme to the full guarantees of https. Please join us for a session on this at tomorrow’s plenary day if you are interested.

User Interface Security

- **Last Call Working Draft**
- No implementation activity, stalled since 2013

This specification aims to provide content authors with declarative policy mechanisms to protect embedded content from “Clickjacking” in a more functional way than the X-Frame-Options header allows.

A policy grammar and mechanism has been at last call for 2 years with no activity because the implementation approach didn’t meet necessary performance characteristics.

As of September 2015, the group welcomes Dan Kaminsky as an invited expert, who is presenting a new approach to the problem which addresses both clickjacking threats as well as providing a strong and accurate signal of “viewability” for content, a property highly desirable to the online advertising industry. We look forward to new activity on the spec as a result, and an experimental implementation for Chrome is under review.

Credential Management Level 1

- **First Public Working Draft**
- Experimental support in Chrome

Imperative API support for managing credentials in the user agent.

Improves password manager and federated sign-in scenarios.

Currently supports Origin-bound credentials, but API shape is extensible. Group has worked with Credentials CG and considered other work on the horizon (such as FIDO) to be able to accommodate post-password authentication scenarios.

Confinement with Origin Web Labels (COWL)

- **First Public Working Draft**
- Privately prototyped in Firefox

COWL defines an API for specifying privacy and integrity policies on data in the form of Origin labels, and a mechanism for confining code according to such policies.

COWL looks at new ways of constructing “mashup” applications with multiple security principals, allowing data sharing beyond the narrow Same Origin model while imposing restrictions on how such data may be further shared.

This is an early-stage and very ambitious project which will require lots of work and careful analysis but holds interesting potential for constructing complex applications while applying least-privilege and privilege separation techniques.

On the horizon:

- **Suborigin Namespaces (already in charter)**
- Allow resources to declare synthetic security principles more granular than a scheme/host/port Origin.

- **CSP Embedded Enforcement**
- Cooperative least-privilege for embedded content (e.g. ads)

- **https-transitional**
- Optimistic upgrades for http-schemed URLs

Thank you!

- http://www.w3.org/2011/webappsec/