The Evolving Web Security Model

Ian Melven
@imelven

Who is this imelven character ?

* Employed by Mozilla as a security engineer
* Previously worked at Adobe, Symantec, McAfee, @stake
* I <3 the web
* Nothing in this talk should be construed as an official Mozilla position (except for maybe I <3 the web)

What is the web security model ?

* Is there a unified web security model ?
* No.
* There are lots of different parts with slightly different rules

A journey through time and space

* Same Origin Policy
* CORS
* Content Security Policy
* iframe sandbox
* HSTS
* SSL, CA's and cert pinning
* WebGL/DOMCrypt
* The future : web apps, APIs, frameworks, site isolation

The 'Sort-Of' Same Origin Policy

* Netscape Navigator 2 - frames
* scheme://host:port
* Some stuff is accessible cross domain (setting window. location), some stuff used to be but isn’t now (window. history)
* document.domain is weird and sketchy
* Imagine if SOP had been used to check loads - no mixed content !

XmlHttpRequest and Same Origin

* XHR first in IE 5.0 in 1999 [wikipedia]
* Initially enforced same origin policy
* Web developers started using XHR, immediately started working around SOP restriction using proxies
* Major hassle (and possibly insecure)

CORS

* Cross Origin Resource Sharing
* Flash Player 7 - crossdomain.xml (2003)
* Access-Control-Allow-Origin and friends
* CORS started as "Authorizing Read Access to XML Content Using the <?access-control?> Processing Instruction 1.0" in the W3C in 2005
* W3C candidate recommendation January 2013, also under development in WHATWG fetch spec

Put a CORS on it

* Good tool - can cover new use cases !
* CORS in Canvas - https://developer.mozilla.org/en-US/docs/HTML/CORS_Enabled_Image
* CORS in webgl - https://hacks.mozilla.org/2011/11/using-cors-to-load-webgl-textures-from-cross-domain-images/

Content Security Policy (CSP)

CSP - A Policy To Secure Content

* Basic idea is to restrict where a document can load various types of content from
* This includes whether inline scripts or the use of eval() are allowed
* This includes whether styles can be applied via <style> elements and attributes
* default-src: * ; script-src: unsafe-inline unsafe-eval ; style-src: unsafe-inline

Brief History of CSP : The Idea

* bsterne credits RSnake and Gerv for the idea, and Jeremiah Grossman for helping make it happen
* The RSnake and Gerv posts describing the idea are from 2007

Brief History of CSP : Towards a spec

* initial implementation in Firefox 4.0 using X-Content-Security-Policy - March 2011
* predates the spec which evolved CSP syntax from this initial first cut
* X-Webkit-CSP in Chrome 13 - August 2011 and Safari 6 (July 2012)
* Working Draft of spec - November 2011

Brief History of CSP : 1.0, 1.1 and beyond

* spec became Candidate Recommendation in November 2012
* Chrome has unprefixed Content-Security-Policy header support in Chrome 25, Firefox's close to landing, CSP sandbox support in IE10
* no need to send both headers or use different syntaxes !
* new directives and use cases being explored for CSP 1.1 currently !

2013 - The Year Of The ~~Linux Desktop~~ CSP

* Isaac Dawson's research (2012) - 79 out of top 1 million websites serve a CSP header of some kind >:O
* But momentum seems to be building ?
* Libraries like helmet and secureheaders / angular.js CSP
* "Large sites" are interested in using it or are using it in some places
* script-nonce & script-hash being discussed for CSP 1.1
* UserCSP addon - https://addons.mozilla.org/en-us/firefox/addon/newusercspdesign/

CSP - apps and extensions !

* Used by Firefox OS by default for certified and privileged apps
* Apps can apply their own CSP via their manifest
* Used by Chrome extensions, Chrome apps also have a default CSP
* Maybe interesting : frames don’t inherit CSP in browsers but do in apps

iframe sandbox

* Bugzilla Bug 341604 - Implement HTML5 sandbox attribute for IFRAMEs - 2006
* No plugins, unique origin, no form submission, different navigation rules (especially for top level windows), cannot open new windows, no scripts, no pointer lock
* Can opt back in using allow- keywords
* New keywords not backwards compatible :(

The Strange and Wonderful World of the iframe sandbox

* CSP sandbox - what's that about ?
* The weirdness of allow-same-origin
* Combine with seamless for another slightly different security model

You got your CSP in my iframe sandbox

* Both of these mechanisms aim to do the same thing - restrict what content can do
* Different approaches : CSP much more granular, sandbox more coarse grained
* iframe CSP attribute ? <meta> CSP ?
* Some sort of CSP JS API ?

HSTS (HTTP Strict Transport Security)

* ForceHTTPS addon - Collin Jackson, Adam Barth - 2008
* Original draft spec - Jeff Hodges, Collin Jackson, Adam Barth - September 2009
* RFC 6797 - published November 2012
* The preload list and problems with it - https://blog.mozilla.org/security/2012/11/01/preloading-hsts/

HSTS Quirks

* HSTS is an 'internal' redirect
* Interaction with CSP
* Interaction with mixed content
* HSTS does not allow override per spec

SSL/CA system/Certificate Pinning

* SSL/CA system - revocation
* CRL -> OCSP -> OCSP stapling
* Alternate systems : Perspectives, Convergence, Certificate Transparency
* The long game vs what can we do NOW ?
* CA pinning - http://tools.ietf.org/html/draft-ietf-websec-key-pinning-04

WebGL

* GPU access !
* Mitigations : ... blacklist drivers :(
* CORS as mentioned earlier
* Hash cracking on someone else’s machine from web content ?
* WebCL ???

DOM Crypt (JS Crypto)

* Crypto fully implemented in JS vs Crypto API exposed to JS
* getRandomValues
* Current debate about high and low level API
* Probably need both, but this is also problematic

The Future : “Web Apps”

* Web apps is an overloaded term - sites (e.g. gmail) vs packaged apps
* Chrome packaged web apps - http://developer.chrome.com/apps/about_apps.html
* Firefox OS : iframe mozapp and iframe mozbrowser - https://developer.mozilla.org/en-US/docs/DOM/Using_the_Browser_API

The Future : “Web Apps” spec/standard ?

* sysapps WG - http://www.w3.org/2012/sysapps/
* "define a runtime environment, security model, and associated APIs for building Web applications with comparable capabilities to native applications.”
* Security model - http://www.w3.org/wiki/System_Applications_WG:_Security_Model
* Compare to Firefox OS: https://wiki.mozilla.org/Apps/Security and https://wiki.mozilla.org/B2G/Architecture/Runtime_Security

The Future : WebAPI

* Scary from a security perspective !
* Not all apps should have access to all apis
* Web pages should not have access to all apis
* Different ‘classes’ of apps - packaged, certified, trusted
* Firefox OS : https://mxr.mozilla.org/mozilla-central/source/dom/apps/src/PermissionsTable.jsm

https://wiki.mozilla.org/WebAPI

The Future : Site Isolation

* Per process site (NOT origin) isolation in browsers
* Paper: "App Isolation : Get the Security of Multiple Browsers with Just One"
* IPC performance is critical, more processes = more memory (think mobile)
* Pretty close to process per packaged web app in Firefox OS et al?

The Future : Security conscious frameworks ?

* Paper: Privilege Separation in HTML5 Applications
* Idea is to use some of the mechanisms discussed earlier to build web apps with components of different privilege and a restricted interface between them
* Sandboxes inside of sandboxes inside of sandboxes

The Cycle of History Continues...

* ‘Web security model’ is constantly evolving as new attacks are discovered and new apis/capabilities are added to the web
* The web itself is constantly evolving - gmail in 2013 vs cgi-bin webmail in the mid 90’s
* What about ‘breaking the web’ ?
* We are making progress ! (I think?)
* We are recording our learnings in specs/standards !

To Shape The Future

* There is still much further to go (understatement)
* Browser implementors need feedback from people trying to use these mechanisms to protect websites or in apps
* What works ? What doesn’t ? what are the pain points ?
* w3c public-webappsec mailing list, IETF websec mailing list
* For Mozilla stuff: file bugs in bugzilla, discuss on mozilla.dev.security, ~~complain to me on twitter~~

A Final Thought

[from a 1/8/2013 post to the whatwg list discussing the need to account for webapis in the html 5 spec's security model]

"I suspect we'll need more of that sort of thing as time goes on.

Which means the security model will likely need to evolve.

Put another way, I think we have good evidence that the security model in the spec, as well as that in every browser, Gecko included, is wrong in the same sense that Newtonian mechanics is wrong. The problem is that we don't know what our equivalent of special relativity is yet."

- Boris Zbarsky

Shout outs

Thank you for attending !

* Mozilla Security Engineering (especially Sid Stamm) & Mozilla Security Assurance (especially Dan Veditz)

* Mike West

* Deneb Meketa

* Neil Matatall

* Brad Hill

* Boris Zbarsky

* Everyone working to improve web security in the past, present, and future

* Andy C, Ed Rush, Optical, Trace

Questions ?

Resources

https://www.frederik-braun.com/thesis/ Origin Policy Enforcement in Modern Browsers - A Case Study on Same Origin Policy Implementations - frederik braun, Ruhr Universitat Bochum

http://www.slideshare.net/BradHill2/w3-conf-hillhtml5securityrealities - Brad Hill

http://benvinegar.github.com/seamless-talk - Seamless iframes - Ben Vinegar

http://seclab.stanford.edu/websec/origins/fgo.pdf - Beware of Finer-Grained Origins - Collin Jackson and Adam Barth

http://lcamtuf.coredump.cx/postxss/ - Postcards from the post-XSS world - Michal Zalewski

Resources

http://slides.creativemisuse.com/ - Firefox OS Application Security - Paul Theriault

http://www.html5rocks.com/en/tutorials/security/content-security-policy/ - Mike West

http://deadliestwebattacks.files.wordpress.com/2013/02/javascript-security-html5.pdf - Javascript & HTML5 Security - Mike Shema (BSides SF 2013)

https://mikewest.org/2013/02/securing-the-client-side-devoxx-2012 - Securing the Client Side - Mike West

Resources

http://www.cs.berkeley.edu/~devdatta/papers/LeastPrivileges.pdf - Privilege Separation in HTML5 Applications - Devdatta Akhawe, Prateek Saxena, Dawn Song - UC Berkeley

http://www.html5rocks.com/en/tutorials/security/sandboxed-iframes/ - Mike West

http://www.veracode.com/blog/2012/11/security-headers-report/ - Security Headers On The Top 1,000,000 Websites - Isaac Dawson

http://www.cs.berkeley.edu/~afelt/ - Adrienne Porter Felt - several papers on permissions, particularly on Android