Atom
Information Security

Application Security for HTML5

Chuck Ben-Tzur

October 2, 2012

Overview

- **Goal**

  Review (some of) HTML5 new features and the related application security attacks scenarios, risks and recommendations.

- **Agenda**
  - The History of HTML5 (The short version)
  - HTML5 Updates (The incomplete version)
  - Local Storage
  - Web Messaging
  - Web Sockets
  - Implementation Considerations

History of HTML5

- HTML4 was published as a W3C Recommendation in… **December 1997**
- 2002 - W3C recommended XHTML 1.0 as the “next step in the evolution of the Internet” – no plans for future HTML versions.
- 2004 - the Web Hypertext Application Technology Working Group (WHATWG) started working on the next version of HTML (not XHTML based).

  The WHATWG was founded by individuals from Apple, the Mozilla Foundation and Opera Software.
- 2009 - W3C allowed the XHTML 2.0 working group to expire and started working with WHATWG on the development of HTML5 (WHATWG HTML).
- Current target date for Recommendation status is 2014 for HTML 5.0 and 2016 for HTML 5.1 (better than 2022…)

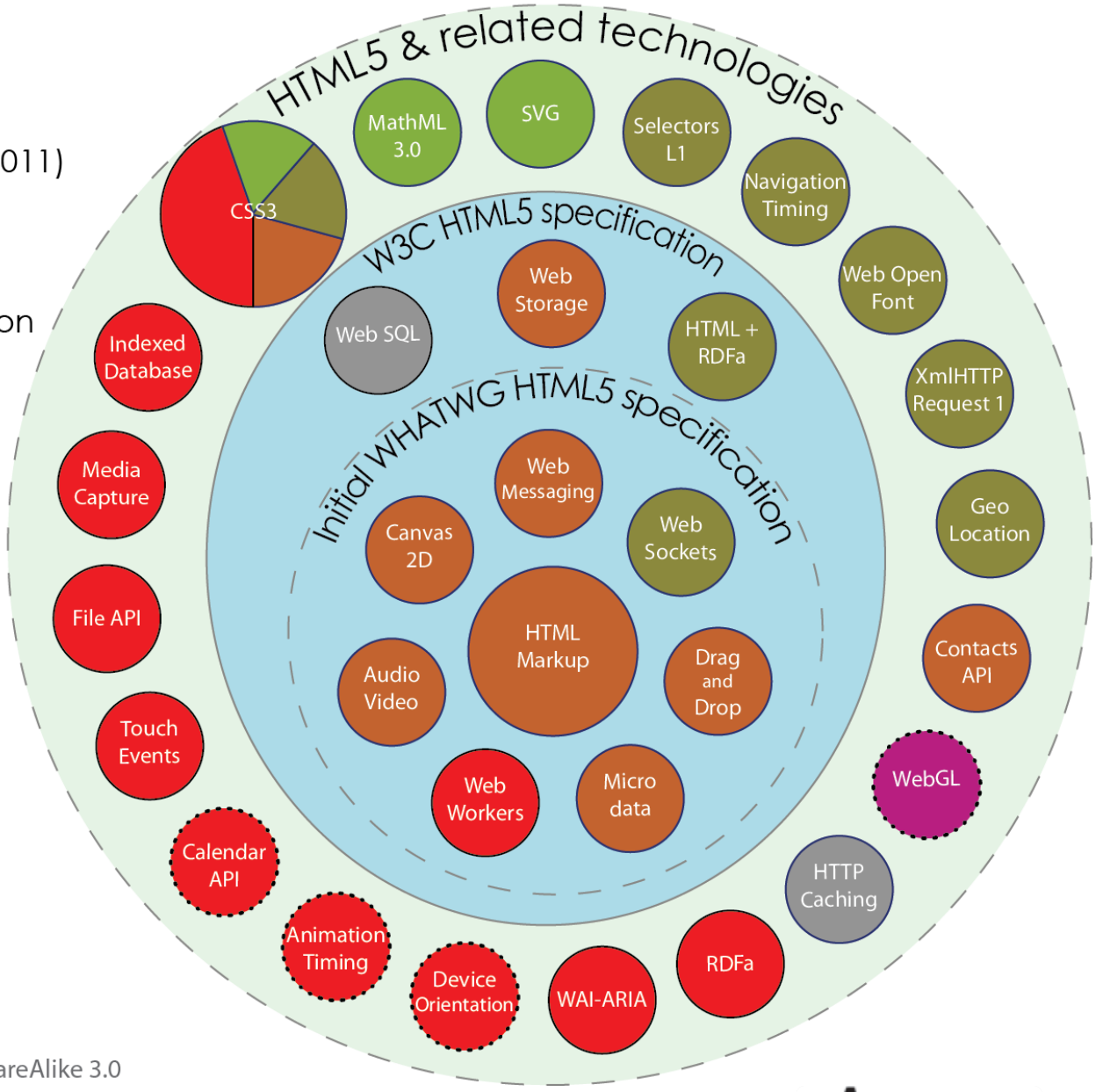
HTML5 Updates

- Markup
  - New tags such as <video>, <audio>, <canvas>
  - Deprecation of tags such as <frame>, <frameset>, <font>
- Backward compatibility with older versions of HTML.
- Scripting APIs - Offline Web Applications, Drag and Drop, Local Storage, 2D Drawing (Canvas), Web Sockets, Web Workers and more.
- Detailed rules for lexing and parsing (browsers will produce the same result in the case of incorrect syntax).
- Related Technologies (not part of the HTML5 specifications):
  Geo Locations, Microdata, File API, MathML

HTML5

Taxonomy & Status (December 2011)

- W3C Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated W3C APIs



Atom

Local Storage

- Started as part of HTML5 (not anymore)
- Like cookies but with better functionality:
  - Greater Storage Capacity (Cookies – 4K, Firefox and Chrome – 5MB, IE – 10MB)
  - Client Only access (Scripts) – not sent to the server (but may be leveraged as client side input!)
  - Better API as it is using structured data (Associative array – Key, Value pairs)
- Storage types
  - Local (per domain, persists after browser is closed)
  - Session (limited to the window's lifetime)
- Managed and accessed using client side scripts only

Demo Applications

- ACME Social Media Manager (www.acmesmm.com)
  - LocalStorage management (.htm)
  - Edit and Confirm post pages (.aspx) – Hiding keys with "_" prefix.
- AtomInfoSec Website (www.atominfosec.com:81)
  - Contains static script pages (.htm)
  - Pages background is light-blue
- C# (.NET 4.0)
- HTML5 + JavaScript
- Internet Explorer 9
- Google Chrome 22

Atom

Local Storage - Security

- Follows “Same Origin” principle
- Session Storage is not persistent

- Large storage area creates a bigger attack surface
- Not protected (e.g. encryption, object types)
- No expiration date on the LocalStorage or the stored information
- No HTTPOnly, secure or path attributes (similar to cookies functionality).
- Accessible via console and file (cleartext XML)
- Relies on client based security controls

- Do not use for sensitive information

Atom

Web Messaging

- Also known as “Cross Domain Messaging”.
- Allows messaging between documents from different origins (to allow communication between two non-hostile pages).
- messages can be posted to the following:
  - Other frames or iframes within the document
  - Windows explicitly opened through Javascript calls
  - The parent window
- Initiated and managed using client side scripts only.

Web Messaging - Security

- Relies on client based security controls

- Verify Origin:
  - Explicitly state the expected origin as the second argument to postMessage() method.
  - Check the origin attribute of the sender
- Both pages should only interpret the exchanged messages as data.
- Never evaluate received messages as code or insert it to a page DOM.
- Check the origin properly exactly to match the FQDN.

Atom

Web Socket

- Providing for bi-directional, full-duplex communications channels between client and server.
- Drastically reduces the amount of unnecessary traffic between server and browser.
- Easy to initiate and manage using scripts.
- Defines two new URI schemes (ws: and wss:) for unencrypted and encrypted connections .

Web Socket - Security

- Not supported in all browsers (e.g. Internet Explorer 10, IIS8)
- Using standard web ports (80, 443) impacting security controls such as Data Loss Prevention (DLP), Intrusion Prevention System (IPS) and Logging and Monitoring.
- In a WebSocket port, the packets lack the traditional headers. Requires Deep Packet Inspection solutions as security controls.

- Avoid backward compatibility - use only hybi-00 and RFC 6455 protocols.
- Server must check the origin of the WebSocket client .
- Leverage server side component to perform Input Validation.

Implementation/Security Considerations

- HTML5 Reduces the usage of Objects, ActiveX and other “out of browser” technologies.

- HTML5 standard and browser support are still evolving.
- HTML5 provides far more client side based activity and functionality, more access to the computer's resources (Upgrades the script kiddie “toolkit”)
- Impact on risk and compliance efforts not clear (such as OWASP Top 10 and Risk Assessments)
- Security tools are not up to date
- Check online Resources for latest updates and security risks
- Early adapters are usually the early victims

Atom

Resources - Specifications

- HTML5 (Working Draft)
  http://www.w3.org/TR/2011/WD-html5-20110525/
- Web Hypertext Application Technology Working Group (WHATWG)
  http://www.whatwg.org/
- Local Storage (Candidate Recommendation)
  http://www.w3.org/TR/webstorage/
- HTML5 Web Messaging (Working Draft)
  http://www.w3.org/TR/2010/WD-webmessaging-20101118/
- Web Socket
  http://www.websocket.org/aboutwebsocket.html

- HTML5 differences from HTML4
  http://dev.w3.org/html5/html4-differences/

Atom

Resources – HTML5 Security

- HTML5 Security Resources Repository
  http://html5security.org/
- HTML5 Security Cheat Sheet
  https://www.owasp.org/index.php/HTML5_Security_Cheat_Sheet

- This Presentation
  http://www.atominformationsecurity.com//Resources/HTML5AppSec.pdf
- Demo Code Used
  http://www.atominformationsecurity.com/Resources/HTML5AppSec.zip

Atom

Thank You

cbentzur@atominfosec.com