DENIM GROUP

build | integrate | secure

AJAX Security: Here We Go Again

OWASP Austin

August 29th, 2006

Overview

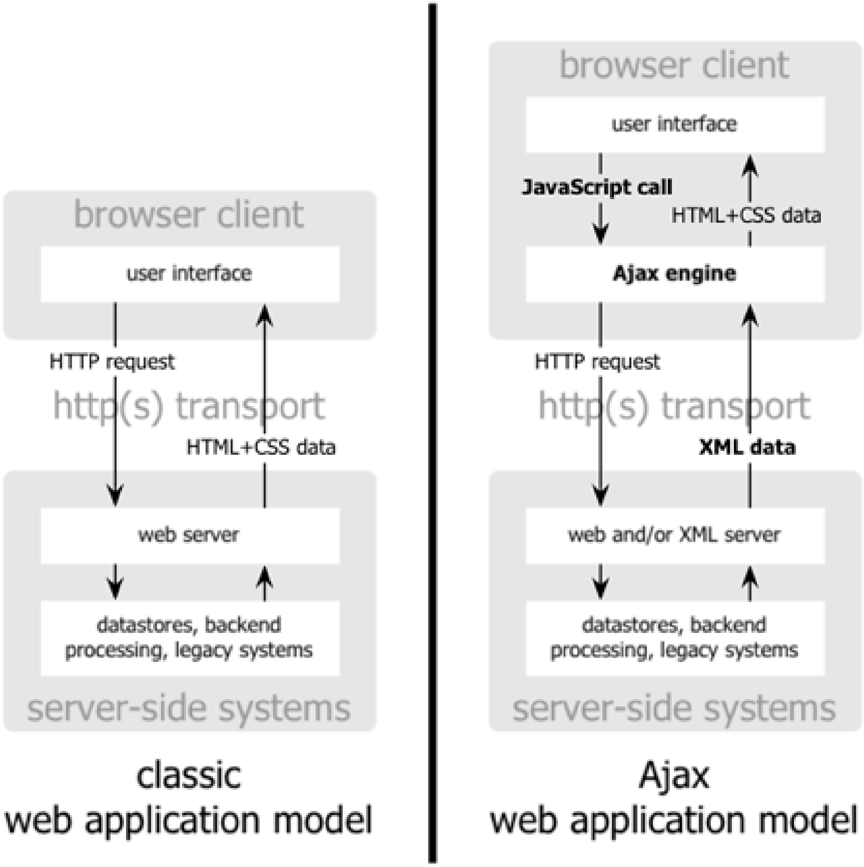
- Introduction
- Basics of AJAX
- AJAX and Application Security
- Sprajax Demo
- Conclusions

Basics of AJAX

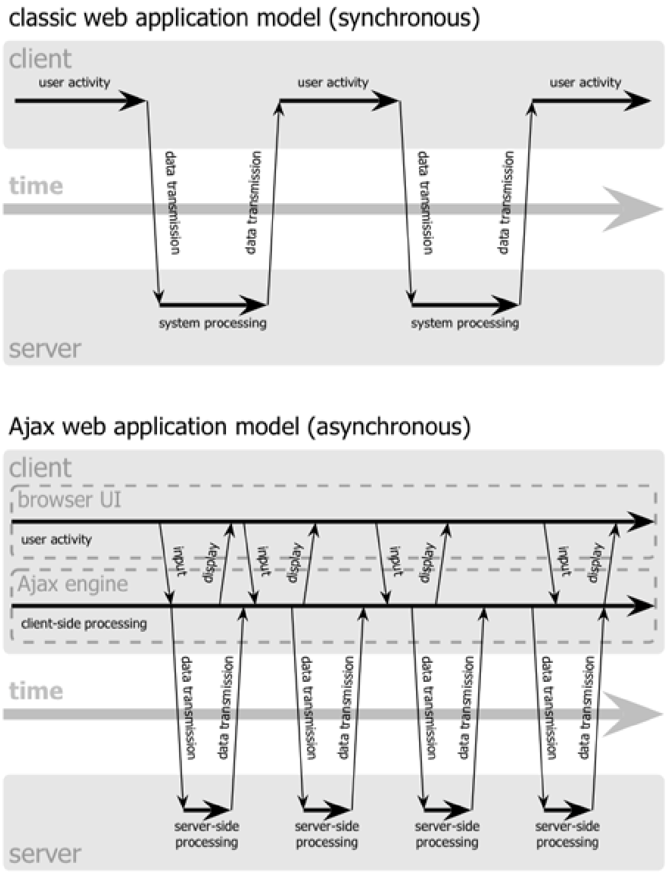
- Asynchronous JavaScript and XML
- Term coined by Jesse James Garrett
  - *http://www.adaptivepath.com/publications/essays/archives/000385.php*
  - *Illustrations from next two slides come from this URL*
- Relies on the XMLHttpRequest object accessible from JavaScript

- ALL THE SAME RULES FOR WEB APPLICATION SECURITY STILL APPLY
  - *Do not trust anything in the request*
    - Cookies
    - Parameters
    - HTTP Headers

AJAX Illustrated



AJAX Illustrated



AJAX Communications

- XMLHttpRequest
- Can use GET parameters
- Can POST arbitrary data
- Can POST XML
- Other approaches
  - *JavaScript Object Notation (JSON) http://www.json.org/*
    - This is used by Atlas

What Does AJAX Look Like?

- Demonstration in Web Proxy

AJAX and Application Security

- Old Favorites
  - *Authentication and Authorization*
  - *SQL Injection*
  - *Cookie Tampering*
  - *Parameter Tampering*
- New Stuff
  - *Additional State on Client*
  - *Cross Site Scripting*
  - *SOA Woes*

Old Favorites

- Authentication and Authorization
- SQL Injection
- Cookie Tampering
- Parameter Tampering

Authentication and Authorization

- XMLHttpRequests send along the same cookies as the browser
  - *Leverage the server-side session*
- Require authentication as with accessing any other web application
- Require authorization before allowing access to sensitive content or capabilities
- Continue to leverage platform Authentication and Authorization features

SQL Injection

- This works just like it does for normal web applications
- Creating SQL queries based on static text and unfiltered inputs

Cookie Tampering

- Cookies are passed along with AJAX XMLHttpRequests
- Malicious users can still modify cookies

Parameter Tampering

- Parameters are passed along with the XMLHttpRequests via a variety of means
  - *GET*
  - *POST*
    - Each framework has its own encoding scheme

New Stuff

- Additional State on the Client
- Cross Site Scripting
- “Service Oriented Architecture” woes

Additional State on the Client

- Bad in normal web applications
  - *Hidden form fields*
- *TERRIBLE* in AJAX applications
  - *Although handling things on the client is kind of the point of AJAX*
  - *No excuse for sloppy server-side validation*
  - *Never trust decisions made by JavaScript on the client side*
    - This can be hard to do in an AJAX environment where client-side JavaScript is a first-class development environment

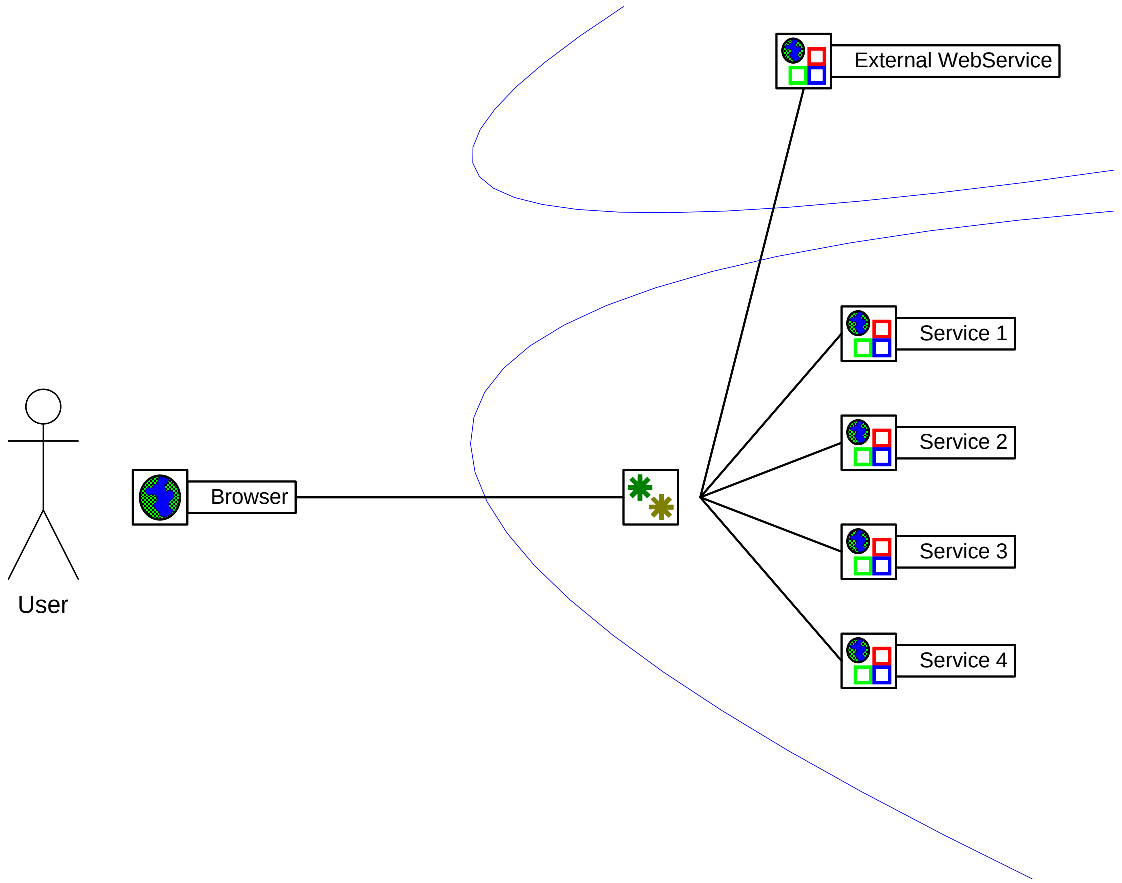
Cross Site Scripting

- Receiving HTML from 3rd party sources is just as dangerous to AJAX-enabled applications
- How much do you trust the folks you are talking to?
- Be careful of error handling using JavaScript "alert" calls

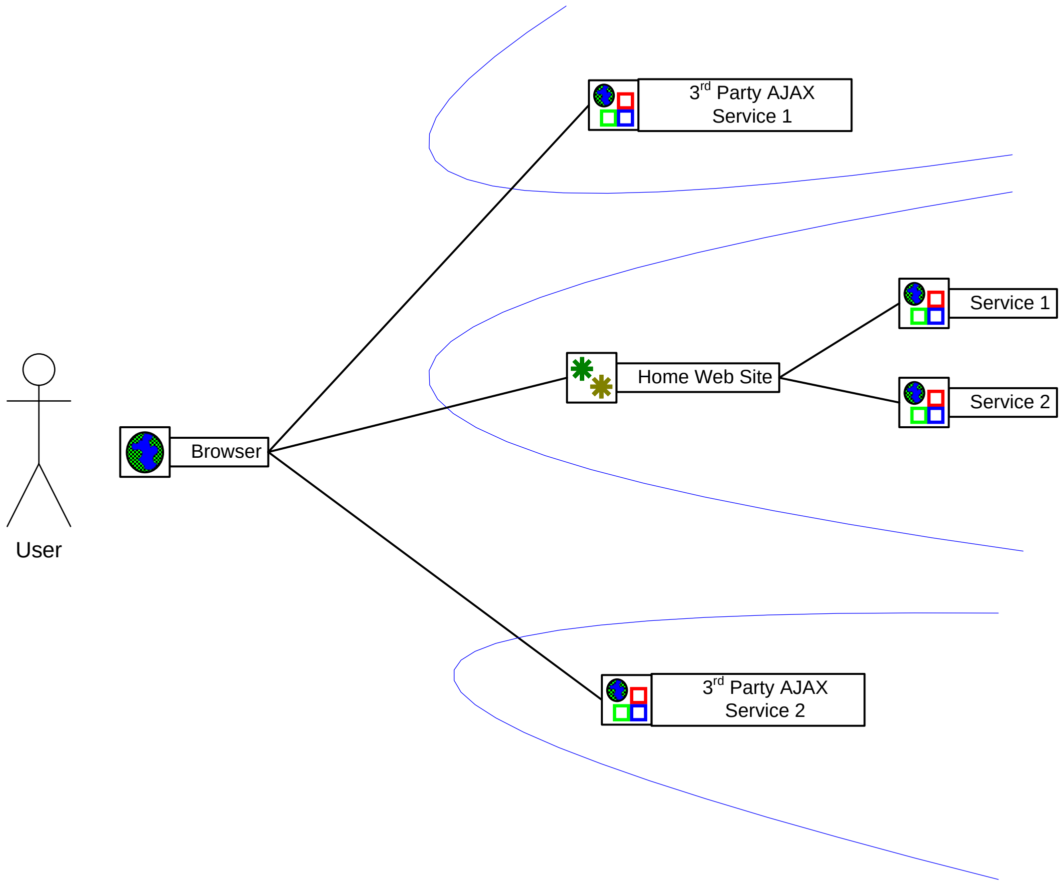
“Service Oriented Architecture” Woes

- Using AJAX services from multiple web applications is essentially creating a “service oriented architecture” where the controller is the client
- Recognize the difference between threat models dealing with in-house SOAs and publicly-accessible SOAs

Traditional SOA



AJAX SOA


3rd Party AJAX Service 1
Service 1
Home Web Site
Service 2
User
Browser
3rd Party AJAX Service 2

Sprajax Demo

- Sprajax Basics
- Actual Demonstration
- Sprajax Internals
- Future directions

Sprajax Basics

- Sprajax is a black-box assessment tool for AJAX-enabled applications
- Spiders and application searching for evidence of AJAX endpoints
  - *Detects common AJAX frameworks (well, Atlas at the moment)*
- Fuzzes AJAX endpoints
  - *Web Services (for Atlas)*
  - *Others in the future*
- Very similar to the web-facing capabilities of existing tools, but specific to AJAX interfaces
- 0.1 alpha level code (be gentle!)

Actual Demonstration

- Demo site needs some work, but you should get the idea

Sprajax Internals

- .NET 2.0 Windows application written in C#
- Makes liberal use of freely available components
- Jeff Heaton C# spider (LGPL)
  - *http://www.owasp.org/software/labs/cspider.html*
- Christian Weyer of Thinktecture's DynWSLib (Non-commercial use)
  - *http://www.thinktecture.com/Resources/Software/DynWsLib/default.html*
- These folks in no way endorse sprajax. All faults are my own.
- Updates to libraries given back to owners/the community

A Sprajax Session

- Spider the site based on a top-level URL
- Detect common AJAX frameworks
  - *Currently based on signatures checking JavaScript file includes*
  - *Could use other “evidence”*
- Detect AJAX endpoints
  - *Currently based on frameworks detected*
  - *Could parse JavaScript snippets*
- Fuzz the endpoints and store the results in a SQL Server database
  - *Currently checks for calls that throw exceptions*
  - *Currently logs either all calls or only calls with exceptions*
  - *Could do more sophisticated analysis and categorization of failures*

Future Directions (For a legit 0.9)

- Clean up data model and event model
- Support the detection of additional frameworks
  - *Google Web Toolkit in progress (detection works, fuzzing does not)*
  - *Direct Web Remoting (DWR) for Java/Spring applications*
  - *And so on. Many have "telltale" signs*
- Improve the UI
  - *Make it easier to navigate the findings from a session*
  - *Add in manual assessment tools, request replay capabilities*
- More real-world testing
  - *The current test application is a "proof of concept"*
- Include authentication capabilities
- Improve request/response signature detection capabilities
  - *Right now this is completely Exception-based*

Conclusions

- AJAX is a premiere “Web 2.0” technique/technology
- The same rules apply as in web application security
  - *Require proper authentication and authorization*
  - *Validate, filter and escape input*
  - *Escape output*
- AJAX can be made secure, but developers have to pay attention
- Tools like sprajax can help to automatically scan applications for some vulnerabilities

Contact

Dan Cornell

dan@denimgroup.com

Website: http://www.denimgroup.com/

Blog: http://denimgroup.typepad.com/

Sprajax Site: http://www.denimgroup.com/sprajax/