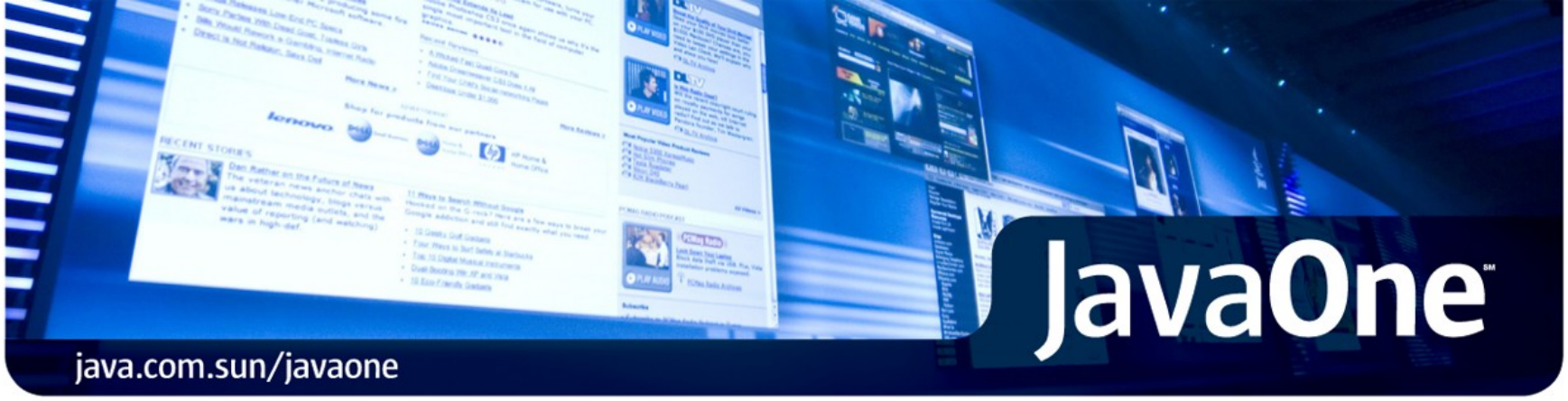

ADDRESSING TOMORROW'S SECURITY REQUIREMENTS IN ENTERPRISE APPLICATIONS

Ben Alex, Principal Software Engineer

TS-6348

spring source

Java

Sun microsystems

Learn what's coming in enterprise application security, and how to achieve it easily today.

GOAL

Agenda

- A Quick Landscape Review
- Simple Web Application Security
- Beyond Simple Web Application Security
- Adding Method Authorization
- Enterprise Connectivity
- AJAX Clients and Web 2.0
- Final Thoughts

Approaching AAAA

- Security considerations revolve around AAAA
  - Authentication: who are they?
  - Authorization: what can they do?
  - Accounting: what resources did they consume in doing it?
  - Auditing: what exactly did they do?
- Java™ Servlets and JAAS
- Third party products
- Build your own

Java™ Servlet Security

- Solid foundation provided by the Servlet API
- HttpServletRequest methods
  - boolean isUserInRole(String role)
  - String getRemoteUser()
  - Principal getUserPrincipal()
- Configured in web.xml
  - <security-constraint>
  - <login-config>
  - <security-role>
- Self registration requirements being considered in JSR 315

JAAS

- Java Authentication and Authorization Service (JAAS)
- Optional in Java™ 1.3, and included in SDK from Java™ 1.4

- LoginContext
- CallbackHandler
- LoginModule
- Subject
- Principal

- Configured in policy files
- Many Java™ servers use JAAS

Agenda

- A Quick Landscape Review
- Simple Web Application Security
- Beyond Simple Web Application Security
- Adding Method Authorization
- Enterprise Connectivity
- AJAX Clients and Web 2.0
- Final Thoughts

What Are "Simple Requirements"?

- Login form
- Authentication against common backends
- Web URL authorization
- Determining who is logged in
- Programmatic authorization
- Logout mechanism
- Externalization of deployment configuration
- Transport-level protection
- Maintaining authentication scope during a session

Demonstration Software

- My demos today will use Spring Security 2
  - Open source project from SpringSource, the company behind Spring
  - Formerly known as “Acegi Security”, and publicly available since 2003
- Builds upon the Java™ platform
  - Integrates with Java™ Servlet Security container authentication
  - Integrates with JAAS login modules
  - External porting efforts underway to .Net, Python and other platforms
- Spring Security supports all technologies discussed today
- Widely used in global banking, defense, government etc

Implementing Simple Requirements
(in 10 minutes or less)

DEMO

Agenda

- A Quick Landscape Review
- Simple Web Application Security
- Beyond Simple Web Application Security
- Adding Method Authorization
- Enterprise Connectivity
- AJAX Clients and Web 2.0
- Final Thoughts

Component, State and Transition Security

- We're moving towards component-based web frameworks
  - For example, Java™ Server Faces
  - Reducing development time, and increasing modularity
- Spring Web Flow provides a JSF model and authorization of
  - States
  - Flows
  - Transitions
- Spring Web Flow unifies JSF and Spring Security

CAPTCHA Overview

- Completely Automated Public Test to tell Computers and Humans Apart
- Useful for mitigating denial of service and IP infringement
- Popular Java™ solutions include JCaptcha and reCAPTCHA
- Consider accessibility issues
  - http://tinyurl.com/3ypzck (Matt May, “Escape from Captcha”, 2004)
- Captchas are often machine decipherable

  - http://www.cs.sfu.ca/~mori/research/gimpy/

Single Sign On and Federated Identity

- NTLM for Microsoft® Windows® intranet apps
  - Works with Mozilla® Firefox® and Microsoft® Internet Explorer®
  - Implement using Samba JCIFS
- JA-SIG Central Authentication Service (CAS) for intranet apps
  - Java™ (for both client and server, plus full Spring Security support)
  - Other clients include .Net, PHP, Perl, Apache etc
- OpenID for Internet apps
  - Sun®, IBM®, Microsoft®, Google®, Yahoo®, AOL®, Yahoo®, Blogger™
  - Implement using OpenID4Java

Agenda

- A Quick Landscape Review
- Simple Web Application Security
- Beyond Simple Web Application Security
- **Adding Method Authorization**
- Enterprise Connectivity
- AJAX Clients and Web 2.0
- Final Thoughts

Protecting Methods

- Strongly recommended in preference to web authorization
  - Can still be used concurrently with web URL authorization
- Before a method invocation
  - Is the user authorized in view of the signature and arguments?
- After a method invocation
  - Was the user authorized in view of the returned object?
  - Should the returned object be modified in some manner?
- Typically used on the services or domain layer
  - Thus all client layers are secured (web, rich client, JMS etc)
  - Of course, proper encapsulation and layering should remain a priority

JSR 250 Method Security Metadata

- JSR 250: “Common Annotations for the Java™ Platform™”
- Annotate classes
  - @RunAs(“someRole”)
  - @RolesAllowed(“someRole”)
  - @PermitAll()
  - @DenyAll()
  - @DeclareRoles(“someRole”)
- Annotation interaction defined in JSR 250, Section 2.11
- Spring Security supports JSR-250, plus @Secured, <protect-pointcut>, <protect-method> and custom strategies

Domain Access Control

- Domain access control considers
  - Who the user is
  - Which method they are invoking on which Java™ type
  - Which domain object instance they are invoking the method on
- Major considerations
  - Performance and normalization level of the ACL database
  - Appropriate tier to perform filtering (eg database-side or Java™)
  - Potential propagation of user identity from Java™ to database

Method Authorization

Agenda

- A Quick Landscape Review
- Simple Web Application Security
- Beyond Simple Web Application Security
- Adding Method Authorization
- Enterprise Connectivity
- AJAX Clients and Web 2.0
- Final Thoughts

Basic Authentication

- Great for RESTful paradigms and remote clients
- RFC 1945, Section 11.1
- Stateless HTTP header based
  - Header name: “Authorization”
  - Header value: “Basic” + “ “ + Base64(username + “:” + password)
  - Some SSO solutions permit session tokens to be presented over Basic
- Recommended, but only if used with HTTPS
  - Consider Digest authentication if only HTTP is available
  - See RFC 2617 Section 4 for a comparison of Basic and Digest
  - Be aware of cross-site request forgery risks

WSS (formerly WS-Security) and REST POX

- WSS provides key security functionality for SOAP
- Use XWSS for Java™ WSS
  - Visit https://xwss.dev.java.net/
  - XWSS version 2.0 implements OASIS WSS Specification 1.0
  - XWSS version 3.0 implements OASIS WSS Specification 1.1
  - Part of Project Metro (https://metro.dev.java.net/) and Glassfish™
- Spring Web Services integrates XWSS and Spring Security
- Securing Plain Old XML using the RESTful paradigm
  - Option 1: Use HTTP Basic authentication
  - Option 2: Use XPath to extract username/password from XML payload

Securing JMS and ESBs

- JMS 1.1 does not provide message integrity or privacy
  - Refer to JMS 1.1 Specification (JSR 914) Section 2.7
  - Implementations are expected to provide such features
- Destination authorization may be provided (eg ActiveMQ)
  - Read from destination; write to destination; admin destination
- ESBs may provide implementation-specific capabilities
  - Endpoint authorization; channel authorization; security translation
- Spring Integration addresses security in its Q2 2008 roadmap

Enterprise Connectivity

Agenda

- A Quick Landscape Review
- Simple Web Application Security
- Beyond Simple Web Application Security
- Adding Method Authorization
- Enterprise Connectivity
- AJAX Clients and Web 2.0
- Final Thoughts

Securing GWT

- GWT offers no authentication-specific features
  - Option 1: ServiceDefTarget.setServiceEndPoint(...) with jsessionid
  - Option 2: Establish a token, then present it as a cookie and in RPC calls
- Key considerations
  - Session and/or token timeout issues
  - How to robustly logout
  - Cross-site request forgery
  - Transition of existing server-side authentication to GWT client
- Useful GWT security advice
  - http://tinyurl.com/3akc6y (GWT wiki) offers GWT security advice
  - http://tinyurl.com/2ynd26 (GWT incubator) includes Spring Security

AJAX Clients and Web 2.0

Agenda

- A Quick Landscape Review
- Simple Web Application Security
- Beyond Simple Web Application Security
- Adding Method Authorization
- Enterprise Connectivity
- Remote Clients and Web 2.0
- Final Thoughts

Enterprise Application Security Tips

- Use a proven security framework; don't roll your own
- Start simply, and add complexity incrementally
- Consider user registration requirements
- Plan for federated identity, particularly involving OpenID
- For in-house applications, consider NTLM and CAS
- Employ Captcha techniques to mitigate DoS attacks
- Favor method authorization over web authorization
- Annotations-based authorization metadata is quick and easy
- Very carefully consider any domain object instance security
- Prefer Basic authentication for RESTful, HTTPS interactions
- Leverage WSS for transport-independent SOAP

THANK YOU

Ben Alex, Principal Software Engineer

TS-6348