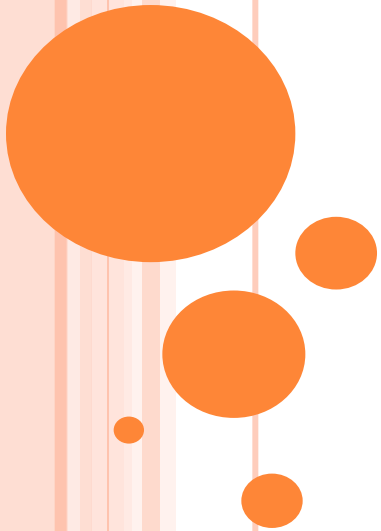


Application Security Enhancements in Java EE 6

SRINI PENCHIKALA

Austin Java User Group Meeting

October 26, 2010

About the Speaker

- Security Architect
- Certified Scrum Master
- Author, Editor (InfoQ)
- IASA Austin Chapter Leader
- Detroit Java User Group Leader
- Working with Java since 1996, JEE (2000), SOA (2006) & PPT since 01/2010
- Current: Agile Security Architectures, Domain-Driven Design, Architecture Enforcement, MDD
- Future: Role of DSL in Architecture Enforcement

Goals for This Presentation

- Overview of new authentication and authorization features in Java EE 6 release
- Best practices in using the new security features

Scope

- Covered:
  - Web Profile
  - Authentication in Web Tier
  - Authorization
  - Transport Security
- Not Covered:
  - EJB/WebServices Security
  - Message Security
  - Other security aspects (Cryptography etc)
- Assumptions:
  - Familiarity with Servlet 3.0 annotations

Format

- New Security Features Overview
- Demos
- Interactive
- Q & A
- Duration: 1 hour

Before We Start

- How many use other security frameworks?

Agenda

- Application Security Model
- Java EE 6 Security Enhancements
- Web Module Security – Authentication
- Authorization
- Enforcing Transport Security
- New Security Features in Java EE 7
- Best Practices
- Conclusions

Agenda

- Application Security Model
- Java EE 6 Security Enhancements
- Web Module Security – Authentication
- Authorization
- Enforcing Transport Security
- New Security Features in Java EE 7
- Best Practices
- Conclusions

Security Architecture Goals

- Portability
- Transparency
- Isolation
- Extensibility
- Flexibility
- Abstraction
- Independence
- Compatibility Testing
- Secure Interoperability

***Source:** Java EE 6 Specification

Application Security Model

- Authentication
- Authorization (Access Control)
- Data integrity
- Confidentiality or Data Privacy
- Non-repudiation
- Auditing

Source: Java EE 6 Specification

Security Architecture Framework

- Security Architecture provides for:
  - Availability of systems and data through:
    - Identification
    - Authorization and access control
  - Integrity of data
  - Confidentiality of data and system information
  - Accountability, which includes:
    - Requirements that actions of an entity can be traced to that entity
    - Non-repudiation, deterrence, fault isolation, and intrusion detection / prevention.
    - Audit
    - Assurance

Source: NIST Special Publications 800-33 and 800-53

Application Security in Java EE

- Prior to Java EE 6:
  - Security in web tier was optional
  - Non-standard implementations

Application Security in Java EE

- Prior to Java EE 6:
  - Security in web tier was optional
  - Non-standard implementations
- Java EE 6:
  - Common support for security-constraint processing
  - Integration of custom authentication mechanisms
  - Track authentication information at the servlet container level

Java EE 6 at a Glance

- Profiles
- Pruning
- More support for Annotations

Java EE 6 Technologies

- Java Platform, Enterprise Edition 6 (JSR 316)
- Enterprise JavaBeans 3.1 (includes Interceptors 1.1) (JSR 318)
- Java Servlet 3.0 (JSR 315)
- JavaServer Faces 2.0 (JSR 314)
- Contexts and Dependency Injection for Java (Web Beans 1.0) (JSR 299)
- Dependency Injection for Java 1.0 (JSR 330)
- Java API for RESTful Web Services (JAX-RS) 1.1 (JSR 311)

Java EE 6 Technologies (2)

- Bean Validation 1.0 (JSR 303)
- Java EE Connector Architecture 1.6 (JSR 322)
- Java Persistence 2.0 (JSR 317)
- Common Annotations for the Java Platform 1.1 (JSR 250)
- Java Message Service API 1.1 (JSR 914)

Management and Security Technologies

- Java Servlet 3.0 (JSR 315)*
- Enterprise JavaBeans 3.1 (JSR 318)
- Java Authentication Service Provider Interface for Containers (JSR 196)
- Java Authorization Contract for Containers 1.3 (JSR 115)

*Focus of this presentation

Agenda

- Application Security Model
- Java EE 6 Security Enhancements
- Web Module Security – Authentication
- Authorization
- Enforcing Transport Security
- New Security Features in Java EE 7
- Best Practices
- Conclusions

Java EE 6 Web Profile

- What is it?
  - Designed for modern web application development for creating small- to medium-sized enterprise web applications
- Provides J2EE Services:
  - Transaction Processing
  - Security
  - Persistence
- Easy to move to the Full Platform

Security Enhancements in Java EE 6

- Web Tier (Servlet 3.0 specification)
  - Support for Authentication and Authorization in web tier
  - Programmatic Security
  - Declarative Security
- EJB Tier
- Web Services
- Application Client Container (ACC)

Java Security Packages

- javax.annotation.security
- javax.ejb
- javax.servlet.annotation
- javax.servlet.http
- javax.ws.rs.core
- javax.interceptor

Agenda

- Application Security Model
- Java EE 6 Security Enhancements
- Web Module Security – Authentication
- Authorization
- Enforcing Transport Security
- New Security Features in Java EE 7
- Best Practices
- Conclusions

Authentication - Programmatic Security

- When to use:
  - Need to control security programmatically
  - Declarative security alone is not sufficient
- New methods in HttpServletRequest (Servlet 3.0)
  - authenticate
  - login
  - logout
- Other methods to access security information about the user
  - getRemoteUser
  - isUserInRole
  - getUserPrincipal

Demo Sample Application Details

- Use Case:
  - Loan processing application
  - Different user roles for different business functions
- Architecture:
  - Servlet 3.0
  - CDI, EJB 3.1
- Technologies:
  - GlassFish Server v3.1
  - LDAP authentication (OpenDS Server)
- Tools:
  - NetBeans 6.9 IDE
  - Maven

User Roles

Role	Function	Permission
Borrower	Loan Application	RW
Loan Officer	Loan Processing	RW
Loan Officer	Loan Underwriting	R
Underwriter	Loan Underwriting	RW
Funding Processor	Funding	RW
Funding Processor	Loan Underwriting	R

Demo 1: Programmatic Security

- Demo of authenticate() method
  - AuthenticationServlet
  - **URL:** http://localhost:8080/jeesecurityapp/auth
- login() and logout() methods – code samples
  - LoginServlet
- Security using Servlet Filter
  - Example: LoginFilter

Declarative Security

- When to use:
  - Need to define security model external to application code
  - Annotation driven
- New annotations for authentication & authorization
  - @ServletSecurity
  - @HttpConstraint , @HttpMethodConstraint
  - @WebFilter
  - @DeclareRoles
  - @RunAs
- Transport Security:
  - ServletSecurity.EmptyRoleSemantic
  - ServletSecurity.TransportGuarantee

Agenda

- Application Security Model
- Java EE 6 Security Enhancements
- Web Module Security – Authentication
- Authorization
- Enforcing Transport Security
- New Security Features in Java EE 7
- Best Practices
- Conclusions

Authorization - Declaring Security Roles

- role-name defined in web.xml deployment descriptor

```
<security-role>
    <role-name>Underwriter</role-name>
</security-role>
```

Declaring Security Roles (2)

- security-role-ref element of deployment descriptor
- Example:

```
<servlet>
...
    <security-role-ref>
        <role-name>Underwriter</role-name>
        <role-link>Underwriter</role-link>
    </security-role-ref>
...
</servlet>
```

Role Based Access Annotation

```
@WebServlet(name="UnderwritingServlet",
  urlPatterns={"/UnderwritingServlet"})
@ServletSecurity(@HttpConstraint(
rolesAllowed = { "Underwriter“, “” }))
public class UnderwritingServlet extends
  HttpServlet {
```

Demo 2: Authorization

- "rolesAllowed" attribute of @HttpConstraint attribute of @ServletSecurity annotation
- **Example:** UnderwritingServlet

Agenda

- Application Security Model
- Java EE 6 Security Enhancements
- Web Module Security – Authentication
- Authorization
- Enforcing Transport Security
- New Security Features in Java EE 7
- Best Practices
- Conclusions

Enforcing Transport Security

- Enforce use of SSL when user is accessing managed resources
- "user-data-constraint" and "transport-guarantee" elements in web.xml
- "transportGuarantee" attribute of @HttpConstraint annotation
- TransportGuarantee values:
  - NONE
  - INTEGRAL
  - CONFIDENTIAL

Transport Security Example – Deployment Descriptor

```
<security-constraint>
...
    <user-data-constraint>
        <transport-guarantee>
            CONFIDENTIAL
        </transport-guarantee>
    </user-data-constraint>
...
</security-constraint>
```

Transport Security Example - Annotation

- Code Example:

```
@ServletSecurity(
value=@HttpConstraint(
transportGuarantee=ServletSecurity.TransportGuar
  antee.CONFIDENTIAL),
httpMethodConstraints={
    @HttpMethodConstraint(value="GET",

  transportGuarantee=ServletSecurity.TransportGu
  arantee.CONFIDENTIAL, rolesAllowed = {
  "Underwriter" })
    }
```

Agenda

- Application Security Model
- Java EE 6 Security Enhancements
- Web Module Security – Authentication
- Authorization
- Enforcing Transport Security
- New Security Features in Java EE 7
- Best Practices
- Conclusions

Future Direction of Java EE Security

- Auditing
- Instance-based Access Control
- User Registration

Application Security Features Wish List

- Built-in “Remember Me” Feature
- Self-Registration Use Case
- Forgot User Id/Forgot Password
- Multi-Factor Authentication
- EL based authorization
- API for managing the ACL's

Auditing

- Protect application resources from illegal use by authorized users
- GlassFish support for auditing
- Fine-grained auditing
- Example: CDI and Interceptor based solution

Auditing - Code Sample

- Using Custom Annotation
- Annotation: SecurityAuditLog
- Implementation: SecurityAuditLogInterceptor
- Used by UnderwritingBean

Agenda

- Application Security Model
- Java EE 6 Security Enhancements
- Web Module Security – Authentication
- Authorization
- Enforcing Transport Security
- New Security Features in Java EE 7
- Best Practices
- Conclusions

Security Best Practices

- Separation of Security and Business Logic
- Authentication Logic in Controller Layer
- Authorization Logic in Service or Controller classes
- Access decision model to be data driven than code driven
- Access Control: Whitelisting v. Blacklisting
- Auditing
- Application Security in SOAP and REST Web Services
- Use user data constraint with Basic or Form authentication

Agenda

- Application Security Model
- Java EE 6 Security Enhancements
- Web Module Security
- Enforcing Transport Security
- EJB Module Security
- Securing Web Services
- Other Security Features in Java EE
- Best Practices
- Conclusions

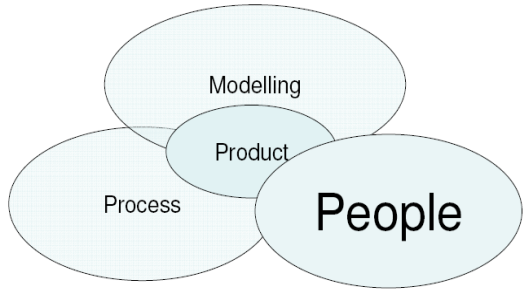
Summary: Security by Architecture Layer

- Domain Layer
  - Access Control Lists (ACLs)
  - Data Encryption / Method Interception
- Service/Facade
  - Declarative (@RolesAllowed)
  - Programmatic
- Controller
  - Declarative (@ServletSecurity, @RolesAllowed)
  - Programmatic (authenticate, login, logout)
- Presentation
  - URL Pattern Access Control
- Transport Layer
  - CONFIDENTIAL, INTEGRAL, NONE

Conclusions

- Web Profile in Java EE 6
- Authentication and Authorization in Web Tier
- Declarative and Programmatic Security
- Transport Layer Security
- Annotation Driven Security Implementation

Q & A



References

- JSR 316: Java EE 6.0 Specification (http://www.jcp.org/en/jsr/detail?id=316)
- JSR 315: Java Servlet 3.0 Specification (http://jcp.org/en/jsr/detail?id=315)
- JSR 318: Enterprise JavaBeans 3.1 (http://jcp.org/en/jsr/detail?id=318)
- JSR 196: Java Authentication Service Provider Interface for Containers (http://jcp.org/en/jsr/detail?id=196)
- JSR 115: Java Authorization Contract for Containers (http://jcp.org/en/jsr/detail?id=115)
- JSR 250: Common Annotations for the Java Platform (http://jcp.org/en/jsr/detail?id=250)

References (2)

- Java EE 6 Tutorial, (http://docs.sun.com/app/docs/doc/820-7627/gijrp?l=en&n=1&a=view)
  - Part VII - Security
    - Chapter 24, Introduction to Security in the Java EE Platform
    - Chapter 25, Getting Started Securing Web Applications
    - Chapter 26, Getting Started Securing Enterprise Applications
- SecureJavaEE6App Tutorial - NetBeans Wiki (http://wiki.netbeans.org/SecureJavaEE6App)
- Java EE 6: Application Security Enhancements (http://www.infoq.com/news/2010/07/javaee6-security)
- GlassFish Security, Masoud Kalali, Packt Publishing
- Getting Started with Java EE Security, DZone Refcardz

References - Tools

- GlassFish Server (https://glassfish.dev.java.net/downloads/v3-final.html)
- OpenDS Server (https://opends.dev.java.net/public/downloads_index.html)
- NetBeans (http://netbeans.org/downloads/)
- Maven (http://maven.apache.org)

References - Frameworks

- Spring Security 3 (http://static.springsource.org/spring-security/site/index.html)
- JBoss PicketLink (http://jboss.org/picketlink)
- Seam Security 3 (http://anonsvn.jboss.org/repos/seam/modules/security/trunk/)

Thank You

- Thank you for your attention
- Session feedback
- Contact Information:
  - Domain-Driven Design, Security and Enterprise Architecture articles on InfoQ
  -  website: http://www.infoq.com
  -  srinipenchikala@gmail.com
  -  @srinip
  -  http://srinip2007.blogspot.com

Extra Slides

Web Profile Technologies

- JSR 315: Java Servlet 3.0
- JSR 314: JavaServer Faces (JSF) 2.0
- JSR 245: JavaServer Pages 2.2 and Expression Language (EL) 1.2
- JSR 52: A Standard Tag Library for JavaServer Pages 1.2
- JSR-45: Debugging Support for Other Languages 1.0
- JSR 299: Contexts and Dependency Injection for the Java EE Platform 1.0
- JSR 330: Dependency Injection for Java
- JSR 318: Enterprise JavaBeans 3.1 (EJB Lite)
- JSR 317: Java Persistence API 2.0
- JSR 250: Common Annotations for the Java Platform 1.1

EJB Module Security

- Declarative
  - @DeclareRoles
  - @RunAs
  - @PermitAll
  - @DenyAll
  - @RolesAllowed
- Programmatic
  - EJBContext
  - isCallerInRole()
  - getCallerPrincipal()

Web Services Security

- SOAP Web Services
  - SAML
  - WS-Security
- RESTful Web Services
  - JAX-RS Authentication

Web Services Security in JEE6

- Web Modules
  - Dispatch client API
  - Authenticator
- EJB Modules
  - Stateless Session Bean

Web Services Security Demo

- REST Web Services Security Demo
- Jersey JAX-RS Authentication

Application Client Security

- Application Client Container
- Callback handler to collect user credentials
- Example:

```
<callback-handler>
    org.test.CustomCallBackHandler
</callback-handler>
```

Connector Architecture

- Container managed sign-on
- Component managed sign-on
- Resource adapter security

Other Security Features

- Java Authentication SPI
- JACC

Other Security Frameworks

- Spring Security 3
- JBoss PicketLink
- Seam Security