



Capstone Courseware, LLC

33 Boylston Street  
Jamaica Plain, MA 02130

877-227-2477  
capstonecourseware.com

## 562. Securing Java Web Services

### Version 5.0

This advanced course introduces Java developers to key concepts and technology for developing secure web services and securing enterprise software architecture. Though consensus is forming, and standards have largely taken shape, this is still a broad and challenging field. We focus on a few well-defined approaches: XML cryptography, the WS-Security and WS-SecurityPolicy standards, and the Security Assertions Markup Language, or SAML. We also look XACML for authorization policies, and at trust and federation -- not only as envisioned by SAML but also through the WS-Trust and WS-Federation specifications.

These approaches do overlap, and through our primary case studies we present a single, coherent story of assuring confidentiality, integrity and non-repudiation, user authenticity, and proper request authorization with a blend of policy-driven WS-Security, SAML, and even some application-coded digital signature. We also investigate the web-application end of SAML, with an in-depth study of single sign-on and federated identity.

Although for practical purposes this course relies on a specific platform, which is Java EE, the great majority of the course content teaches interoperable specifications, and would be equally useful to developers working on other web-service-capable platforms such as .NET -- or to those who work with multiple platforms, and do need to understand the interoperable pieces in detail but perhaps don't need to delve into implementation strategies. In fact, customizations are available that essentially leave out the Java to stick more strictly to the XML.

### Prerequisites

- Solid Java programming experience is essential; Course 103 provides excellent preparation.
- Experience developing Java Web services is likewise a hard requirement: labs will assume understanding of both SAAJ and JAX-WS. Course 561 is strongly recommended.
- Students are expected to be able to read and write XML fluently, and have some familiarity with XML Schema. Consider courses 501 and 517.



## Learning Objectives

- Understand the unique challenges in securing interoperable XML-based services.
- Apply W3C standards to digitally sign and encrypt XML fragments and documents.
- Understand the importance of the WS-Security specifications to interoperably secure messaging.
- Use state-of-the-art tools to configure or implement signature, encryption, and various WS-Security header content for Java web services.
- Drive such WSS implementations from WS-SecurityPolicy documents.
- "Vouch for" a user across domains to achieve request authorization without sharing credentials.
- Exchange security information between servers, applications, and components, using SAML assertion and protocol models.
- Understand the role of XACML in policy management and decision-making.
- Understand the WS-Trust and WS-Federation architectures for developing the trust relationships that enable service federations and service-oriented architectures.
- Build web applications that participate in SAML federation and single sign-on.

**Timeline: 5 days.**

## IDE Support: Eclipse Ganymede

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse Ganymede. Students can code and compile all Java sources from within the IDE. Due to the advanced nature of the coding tasks, and the non-standard tools that must be used to implement many of our security strategies, there is no integrated server management, deployment, or debugging on the server; the Ant builds baked into all the lab exercises and described in the coursebook are used to deploy services, test them, and to perform certain administrative tasks. See also our orientation to Using Capstone's Eclipse Overlays, and please be advised that this is an optional feature; it is not a separate version of the course, and the course itself does not contain explicit Eclipse-specific lab instructions.





## **Chapter 1. Securing the Service-Oriented Enterprise**

- Security for Web Services
- Threats
- CIA Goals
- Solution Levels: W3C, OASIS, Java EE
- Scenario: Secure Multi-Party Conversation
- Cryptography
- WS-Security and WS-SecurityPolicy
- Scenario: Sharing Security Information
- SAML and XACML
- Scenario: Multiple User Realms
- Scenario: Single Sign-On
- Technology Stacks: WS-Federation and Liberty Alliance
- The WS-I Basic Security Profile

## **Chapter 2. Transport Security**

- Use Case: Secure Transport
- HTTP Authentication Schemes
- HTTP BASIC
- HTTP DIGEST
- Securing Web-Service URLs
- HTTPS
- JAX-WS Support
- Axis Support

## **Chapter 3. XML Signature**

- Use Case: Non-Repudiation
- XML Digital Signature
- Cryptography Backgrounder
- Canonical XML
- Enveloped, Enveloping, and Detached Signatures
- SignedInfo and References
- The Java Cryptography Architecture
- Keystores
- Why Keys Aren't Enough
- X.509 Certificates and Certificate Chains
- The KeyStore API
- Java XML Digital Signature API
- Steps to Sign and Verify XML Content
- JAX-WS Message Handlers





Foiling the Man in the Middle

#### **Chapter 4. XML Encryption**

- Use Case: Confidentiality
- XML Encryption
- EncryptedData
- Element vs. Content Encryption
- Key Wrapping
- The Java Cryptography Extensions
- Apache XML Security
- Steps to Encrypt and Decrypt XML Content
- Choosing Algorithms and Key Sizes

#### **Chapter 5. WS-Security**

- Use Case: Secure Message Exchange
- Use Case: User Login
- The WS-Security Specifications
- Security Token Types
- Timestamps
- Username Tokens
- Signature and Encryption
- Tools for WS-Security
- XWSS and JAAS
- Foiling Replay Attacks

#### **Chapter 6. WS-SecurityPolicy**

- Use Case: Sharing Metadata
- WS-Policy
- Normalized vs. Compact Form
- Policy Attachment
- Policy Scopes
- WS-SecurityPolicy
- Protection Assertions
- Token Assertions
- Supporting and Endorsing Tokens
- Bindings
- Metro and WSIT
- Implementing Callbacks
- Integrating Security Frameworks





## **Chapter 7. Introduction to SAML**

- History of SAML
- Assertions
- Protocol
- Bindings
- Profiles
- Using OpenSAML
- SAML and Web Services

## **Chapter 8. SAML Assertions**

- Use Case: "Vouching for" a User
- The Assertions Schema
- Extensibility
- Assertions and Subjects
- NameID Types
- Conditions
- Subject Confirmation
- Confirmation Methods
- AuthnStatement
- Authentication Contexts
- AttributeStatement
- Attribute Profiles
- AuthzDecisionStatements
- Actions and Evidence
- WS-Security and SAML Tokens
- OpenSAML Assertions Model
- Creating XML Objects
- Marshalling and Unmarshalling

## **Chapter 9. SAML Protocol**

- Use Case: Back-Channel Queries
- Requests, Queries, and Responses
- Status and StatusCode
- AuthnQuery
- AttributeQuery
- AuthzDecisionQuery
- Other Request and Response Types
- OpenSAML Protocol Model
- SAML and XML Signature





SAML and XML Encryption

### **Chapter 10. XACML**

- Use Case: Back-Channel Authorization
- Use Case: Sharing Authorization Policies
- Policies, Policy Sets, and Targets
- Rules
- Combining Algorithms
- Policy Context
- Request and Response Types
- The SAML Profile of XACML
- Authorization Decisions via XACML

### **Chapter 11. Securing Federated Services**

- Publish, Find, Bind ... Execute!
- UDDI
- WS-BPEL
- The Trust Problem
- WS-Trust
- The Security Token Service
- Messaging Model: RST and RSTR
- Derived Keys
- WS-SecureConversation
- Secure Conversation Metrics
- WS-Federation
- Value Proposition

### **Chapter 12. SAML Bindings**

- Use Case: Speaking "Through" the Browser
- The SOAP Binding
- SAML Over HTTP
- The Browser as Messenger
- The Redirect, POST, and Artifact Bindings
- The PAOS Binding
- The URI Binding

### **Chapter 13. Federated Identity**

- What is Federation?
- Problems for Identity Federation





SAML 2.0 Federations  
Single Sign-On  
Account Linking and Persistent Pseudonyms  
Transient Pseudonyms  
Name ID Mapping  
Federation Termination  
OpenSSO  
Fedlets

### **Appendix A. Learning Resources**

### **Appendix B. Web-Service Security Prefixes and Namespaces**

#### **System Requirements**

<b>Hardware Requirements (Minimum)</b>	1.5 GHz, 1 gig RAM, 2 gig disk space.
<b>Hardware Requirements (Recommended)</b>	2 GHz, 2 gig RAM, 3 gig disk space.
<b>Operating System</b>	Tested on Windows XP Professional. Course software should be viable on all systems which support the Java EE 5.0 SDK.
<b>Network and Security</b>	An internet connection is required for each student and instructor machine. Limited privileges required -- please see our standard security requirements at <a href="http://capcourse.com/Guides/Security.html">http://capcourse.com/Guides/Security.html</a> .
<b>Software Requirements</b>	All free downloadable tools.

