

JVA-122. Secure Java Web Development

Version 7.0

This comprehensive course shows experienced developers of Java EE applications how to secure those applications and to apply best practices with regard to secure enterprise coding. Authentication, authorization, and input validation are major themes, and students get good exposure to basic Java cryptography for specific development scenarios, as well as thorough discussions of HTTPS configuration and certificate management, error handling, logging, and auditing.

Perhaps the most eye-opening parts of the course concern common web "hacks," or attack vectors. Students see how easy it is to leave an application unguarded against cross-site scripting (XSS), cross-site request forgery (CSRF), SQL injection, and other attack types -- and learn that it's also easy to fix such vulnerabilities and the importance of a secure development process.

In the last part of the course we move beyond the scope of traditional, interactive web applications to consider RESTful web services, single sign-on systems, and third-party authorization. Students learn to perform HMAC cryptography as a means of HTTP message-level authentication, and get introductions and hands-on exercise with SAML SSO and OAuth.

Course Length: 5 days

Prerequisites

Java programming experience is essential -- [Course 103](#) is excellent preparation.

Servlets programming experience is required -- [Course 111](#).

JSP page-authoring experience is recommended but not required -- again, [Course 111](#).

Understanding of RESTful web services as implemented in JAX-RS will be highly beneficial, but is not strictly required. Consider [Course 563](#).

Learning Objectives

Generally, be prepared to develop secure Java web applications and services, or to secure existing applications and services by refactoring as necessary.

Define security constraints and login configurations that instruct the web container to enforce authentication and authorization policies.

Guard against common web attacks including XSS, CSRF, and SQL injection. Validate user input aggressively, for general application health and specifically to foil injection and XSS attacks.

Configure a server and/or application to use one-way or two-way HTTPS.

Apply application-level cryptography where necessary.

Store sensitive information securely, hash user passwords, and understand the importance of salting and of using slow hashing algorithms and processes, to maximize the safety of stored credentials.

Use HMAC security as appropriate in RESTful web services.

Participate in SAML SSO systems, and be aware of the security concerns involved in single sign-on.

Implement server and client sides of the OAuth-2.0 initial flow in order to provide third-party authorization to resources in a secure manner.

Server Support: Tomcat

IDE Support: Eclipse Luna

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse Luna. Students can code, build, deploy, and test all exercises from within the IDE. See also our orientation to Using Capstone's Eclipse Overlays.

Course Outline

Chapter 1. Concerns for Web Applications

- Threats and Attack Vectors
- Server, Network, and Browser Vulnerabilities
- Secure Design Principles
- GET vs. POST
- Container Authentication and Authorization
- HTML Forms
- Privacy Under /WEB-INF
- HTTP and HTTPS
- Other Cryptographic Practices
- SOA and Web Services
- The OWASP Top 10

Chapter 2. Authentication and Authorization

- HTTP BASIC and DIGEST Authentication Schemes
- Declaring Security Constraints
- User Accounts
- Safeguarding Credentials in Transit
- Replay Attacks
- Authorization Over URL Patterns
- Roles
- FORM Authentication
- Login Form Design
- Session Fixation
- Protections
- Programmatic Security
- Programmatic Security in JSF

Chapter 3. Common Web Attacks

- Forceful Browsing
- Predictable Resource Locations
- Using Random Numbers
- Cross-Site Scripting
- Output Escaping

- Cross-Site Request Forgery
- Synchronizer Tokens
- Injection Attacks
- Protections in JDBC and JPA
- Session Management
- Taking Care of Cookies

Chapter 4. Input Validation •Validating User Input

- Validation Practices
- Regular Expressions
- Bean Validation (a/k/a JSR-303)
- Constraint Annotations
- Cross-Field Validation
- Built-In Support in Java EE
- Using a Validator
- Producing Error Responses
- JSF Validation

Chapter 5. HTTPS and Certificates

- Digital Cryptography
- Encryptio
- SSL and Secure Key Exchange
- Hashing
- Signature
- Keystores
- keytool
- Why Keys Aren't Enough
- X.509 Certificates
- Certificate Authorities
- Obtaining a Signed Certificate
- Configuring HTTPS
- Client-Side Certificates and Two-Way SSL
- PKCS #12 and Trust Stores
- CLIENT-CERT Authentication

Chapter 6. Application-Level Cryptography

- The Java Cryptography Architecture
- Secure Random Number Generation
- The KeyStore API
- Digital Signature
- Hashing
- Password Hashing
- Why Hashing Isn't Enough

- Salts
- Key Lengthening and Key Strengthening
- Slow Algorithms
- The Java Cryptography Extensions
- The SecretKey and KeyGenerator Types
- Symmetric Encryption
- Choosing Algorithms and Key Sizes
- Dangerous Practices
- Storing and Managing Keys

Chapter 7. REST Security Basics

- Security Concerns for REST Services
- HTTPS
- HTTP BASIC and DIGEST
- Authorization by URL Pattern
- Cross-Site Scripting
- Injection Attacks
- Cross-Site Request Forgery
- Common Countermeasures

Chapter 8. HMAC Security

- Use Case: Message Authentication
- Digital Signature
- Hashing as Signature: the HMAC
- Keyed Hashing
- The Hmac Utility
- Appropriate Salts
- Canonicalization
- Amazon S3
- Timestamps
- Signing and Verifying Messages
- XML Cryptography and Canonicalization
- Canonicalizing JSON

Chapter 9. SAML SSO

- The Challenge of Single Sign-O
- Federated Identity
- SAML 2.0
- The Web Browser SSO Profile
- Identity Providers and Service Providers •SAML Assertions
- SAML Protocol
- SAML Bindings
- Speaking "Through" the Browser

- The HTTP Redirect Binding
- Artifact and SOAP Bindings
- SAML Attributes
- Security Concerns in SSO Systems

Chapter 10. OAuth

- Use Case: Third-Party Authorization
- OAuth
- Initial Flow
- Grant Types
- Access Tokens
- The Google OAuth API
- Implementing Authorization and Resource Servers
- Implementing Clients
- Security Concerns with OAuth

Appendix A. Learning Resources