



Capstone Courseware, LLC

33 Boylston Street  
Jamaica Plain, MA 02130

877-227-2477  
capstonecourseware.com

## 107. Java Development for Secure Systems

### Version 6.0

This course exposes students to the broad range of challenges and techniques that is "Java security." Secure coding practice for Java incorporates techniques for Java SE and Java EE, and increasingly EE applications are using SE techniques such as policy files and JAAS authentication. This course spends some time on each platform, so that students will be exposed to SE basics such as access controller, permissions, and policies; and also traditional EE techniques such as web-security declarations and the EJB authorization model. Best-practice chapters wrap up coverage of each platform.

The course emphasizes hands-on exercise, and students will spend more than half of their classroom time solving specific security problems. Most labs are organized as scenarios in which a security breach of existing software is possible - students begin by hacking the system in some way. Then the work of the lab is to tighten up the software to eliminate the threat: set a secure policy, sign a file, clean up overexposed parts of an API, require user login, etc.

This version of the course targets Java SE 6 and Java EE 5, but it is largely applicable to Java SE 5 and J2EE 1.4 as well, and groups looking for Java training who know they'll be using those earlier platforms are encouraged to use this course. For training within the J2SE 1.4 environment, please see version 1.4 of this course.)

### Prerequisites

- Solid Java programming experience is assumed -- Course 103 is excellent preparation.
- Though extensive practical experience with Java EE development is not necessary, some knowledge of Java EE architecture and development is also recommended -- consider Course 108, which offers a one-day overview of Java EE development, including architecture and working examples.



## Learning Objectives

- Design and implement security policies for Java applications, servers, and components.
- Manage keys and certificates for a Java application, and sign code sources as necessary.
- Practice secure design and coding, and balance usability with security in UI and API.
- Sign and verify application data and messages using the JCA, and encrypt/decrypt using the JCE.
- Incorporate JAAS authentication into an application.
- Implement a JAAS LoginModule to connect to your own application data.
- Secure Java EE applications by URL and role, and integrate JAAS authentication.
- Avoid common pitfalls of Java web applications, including SQL injection and cross-site-scripting attacks.

**Timeline: 3 days.**

## IDE Support: Eclipse WTP 2.0

In addition to the primary lab files, an optional overlay is available that adds support for Eclipse WTP 2.0. Students can code and build all exercises from within the IDE. Some exercises can be tested from within the IDE as well, though most must be tested from the command line to take advantage of various scripted configuration tasks such as building keystores and signed JARs. 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. Java SE Security**

- Holistic Security Practices
- Threats to the User
- The Class Loader and Bytecode Verifier
- System Classes and the Core API
- SecurityManager and AccessController
- Permissions
- Implication
- CodeSources
- Policies
- Configuring Java SE Security
- Dynamic Policies
- Privileged Actions

## **Chapter 2. Code Signature and Key Management**

- Encryption and Digital Signature
- Keystores
- Keys and Certificates
- Certificate Authorities
- The KeyStore API
- Signing JARs
- Signed CodeSources
- Additional Policy Semantics

## **Chapter 3. Secure Development Practices: Java SE**

- Code Injection
- Final Classes and Methods
- Singletons, Factories, and Flyweights
- Methods, Collections, and Data Hiding
- Sealing JARs
- Code Obfuscation
- Object Serialization

## **Chapter 4. Cryptography**

- Threats to Identity and Privacy
- The Java Cryptography Extensions
- The Signature Class
- SignedObjects
- The Java Cryptography Extensions





- SecretKeys and KeyGenerator
- The Cipher Class
- Dangerous Practices
- HTTP and JSSE

### **Chapter 5. JAAS**

- Pluggable Authentication Logic
- JAAS
- Packages and Interfaces
- Subjects and Principals
- ANDs and ORs
- Impersonation Methods
- Permissions for JAAS Use
- LoginContext and LoginModule
- Configuring JAAS
- CallbackHandler and Callbacks
- Implementing a JAAS Client
- Implementing a LoginModule

### **Chapter 6. Java EE Security**

- Java EE Servers as Code Hosts
- Tomcat Security Configuration
- Declaring Roles
- Securing URLs
- HTTP Authentication Schemes
- Securing EJBs
- Programmatic Security
- JAAS in Java EE
- Realms and LoginModules
- JAAS in Tomcat
- JACC
- Certifying a Java EE Application
- HTTPS Configuration

### **Chapter 7. Secure Development Practices: Java EE**

- Presentation-Tier Vulnerabilities
- User Accounts
- MVC and Security
- Validating User Input
- SQL Injection





Cross-Site Scripting  
Reflected XSS  
Defeating XSS  
OWASP  
Penetration Testing  
Error Handling and Information Leakage  
Logging and Auditing

## Appendix A. Learning Resources

### System Requirements

<b>Hardware Requirements (Minimum)</b>	500 MHz, 256 meg RAM, 500 meg disk space.
<b>Hardware Requirements (Recommended)</b>	1.5 GHz, 512 meg RAM, 1 gig disk space.
<b>Operating System</b>	Tested on Windows XP Professional. Course software should be viable on all systems which support a Java 6 Developer's Kit.
<b>Network and Security</b>	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.

