

Training: Capstone Courseware 107 Java Development for Secure Systems



TRAINING GOALS:

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.

CONSPECT:

- Chapter 1. Java SE Security
 - Holistic Security Practices
 - $\circ~\mbox{Threats}$ to the User
 - $\circ\,$ The Class Loader and Bytecode Verifier
 - System Classes and the Core API
 - SecurityManager and AccessController
 - Permissions
 - \circ Implication
 - $\circ \ {\rm CodeSources}$
 - Policies
 - Configuring Java SE Security

www.compendium.pl





- Dynamic Policies
- Privileged Actions
- $\circ\,$ 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
 - $\circ~$ Code Injection
 - Final Classes and Methods
 - Singletons, Factories, and Flyweights
 - $\circ\,$ Methods, Collections, and Data Hiding
 - Sealing JARs
 - Code Obfuscation
 - Object Serialization
- Chapter 4. Cryptography
 - $\circ\,$ 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
 - $\circ~$ HTTP and JSSE
- Chapter 5. JAAS
 - Pluggable Authentication Logic
 - \circ JAAS
 - Packages and Interfaces
 - Subjects and Principals
 - $\circ~\mbox{ANDs}$ and ORs
 - Impersonation Methods

www.compendium.pl





- Permissions for JAAS Use
- LoginContext and LoginModule
- Configuring JAAS
- CallbackHandler and Callbacks
- Implementing a JAAS Client
- $\circ~$ Implementing a LoginModule
- Chapter 6. Java EE Security
 - $\circ\,$ Java EE Servers as Code Hosts
 - $\circ\,$ Tomcat Security Configuration
 - Declaring Roles
 - Securing URLs
 - HTTP Authentication Schemes
 - Securing EJBs
 - Programmatic Security
 - $\circ\,$ JAAS in Java EE
 - Realms and LoginModules
 - JAAS in Tomcat
 - \circ JACC
 - Certifying a Java EE Application
 - HTTPS Configuration
- Chapter 7. Secure Development Practices: Java EE
 - Presentation-Tier Vulnerabilities
 - User Accounts
 - $\circ~$ MVC and Security
 - Validating User Input
 - SQL Injection
 - Cross-Site Scripting
 - Reflected XSS
 - Defeating XSS
 - OWASP
 - Penetration Testing
 - $\circ\,$ Error Handling and Information Leakage
 - Logging and Auditing

REQUIREMENTS:

www.compendium.pl





- 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 Overview of Java EE Development, which offers a one-day overview of Java EE development, including architecture and working examples.

Difficulty level

CERTIFICATE:

The participants will obtain certificates signed by Capstone Courseware.

TRAINER:

Authorized Capstone Courseware Trainer.



page 4 of 4