Java Development for Secure Systems v6.0

Course Summary

Description

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.

Objectives

At the end of this course, students will be able to:

- 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.

Topics

- Java SE Security
- Code Signature and Key Management
- Secure Development Practices: Java SE
- Cryptography
- JAAS
- Java EE Security
- Secure Development Practices: Java EE
- Appendix A. Learning Resources

Prerequisites

- Solid Java programming experience is assumed
- Though extensive practical experience with Java EE development is not necessary, some knowledge of Java EE architecture and development is also recommended

Duration

Three days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Java Development for Secure Systems v6.0

Course Outline

I. Java SE Security
- A. Holistic Security Practices
- B. Threats to the User
- C. The Class Loader and Bytecode Verifier
- D. System Classes and the Core API
- E. SecurityManager and AccessController
- F. Permissions
- G. Implication
- H. CodeSources
- I. Policies
- J. Configuring Java SE Security
- K. Dynamic Policies
- L. Privileged Actions

II. Code Signature and Key Management
- A. Encryption and Digital Signature
- B. Keystores
- C. Keys and Certificates
- D. Certificate Authorities
- E. The KeyStore API
- F. Signing JARs
- G. Signed CodeSources
- H. Additional Policy Semantics

III. Secure Development Practices: Java SE
- A. Code Injection
- B. Final Classes and Methods
- C. Singletons, Factories, and Flyweights
- D. Methods, Collections, and Data Hiding
- E. Sealing JARs
- F. Code Obfuscation
- G. Object Serialization

IV. Cryptography
- A. Threats to Identity and Privacy
- B. The Java Cryptography Extensions
- C. The Signature Class
- D. SignedObjects
- E. The Java Cryptography Extensions
- F. SecretKeys and KeyGenerator
- G. The Cipher Class
- H. Dangerous Practices
- I. HTTP and JSSE

V. JAAS
- A. Pluggable Authentication Logic
- B. JAAS
- C. Packages and Interfaces
- D. Subjects and Principals
- E. ANDs and ORs
- F. Impersonation Methods
- G. Permissions for JAAS Use
- H. LoginContext and LoginModule
- I. Configuring JAAS
- J. CallbackHandler and Callbacks
- K. Implementing a JAAS Client
- L. Implementing a LoginModule

VI. Java EE Security
- A. Java EE Servers as Code Hosts
- B. Tomcat Security Configuration
- C. Declaring Roles
- D. Securing URLs
- E. HTTP Authentication Schemes
- F. Securing EJBs
- G. Programmatic Security
- H. JAAS in Java EE
- I. Realms and LoginModules
- J. JAAS in Tomcat
- K. JACC
- L. Certifying a Java EE Application
- M. HTTPS Configuration

VII. Secure Development Practices: Java EE
- A. Presentation-Tier Vulnerabilities
- B. User Accounts
- C. MVC and Security
- D. Validating User Input
- E. SQL Injection
- F. Cross-Site Scripting
- G. Reflected XSS
- H. Defeating XSS
- I. OWASP
- J. Penetration Testing
- K. Error Handling and Information Leakage
- L. Logging and Auditing

VIII. Appendix A. Learning Resources

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically