# Secure Java Application Development

# Course Summary

**Description**

This application security training workshop / seminar is essential for developers who need to produce secure Java applications, integrating security measures into the development process from requirements to deployment and maintenance. This course explores well beyond basic programming skills, teaching developers sound processes and practices to apply to the entire software development lifecycle. Perhaps just as significantly, students learn about current, real examples that illustrate the potential consequences of not following these best practices. This course is short on theory and long on application, providing students with in-depth, code-level demonstrations and walkthroughs.

Security experts agree that the least effective approach to security is "penetrate and patch". It is far more effective to "bake" security into an application throughout its lifecycle. After spending significant time trying to defend a poorly designed (from a security perspective) web application, developers are ready to learn how to build secure web applications starting at project inception. The final portion of this course builds on the previously learned mechanics for building defenses by exploring how design and analysis can be used to build stronger applications from the beginning of the software lifecycle.

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

- Understand the concepts and terminology behind defensive coding
- Understand and use Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Learn the entire spectrum of threats and attacks that take place against software applications in today's world
- Use Threat Modeling to identify potential vulnerabilities in a real life case study
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in Java applications
- Understand the vulnerabilities of the Java programming language and the JVM as well as how to harden both
- Understand and work with Java 2 platform security to gain an appreciation for what is protected and how
- Understand the role that Java Authentication and Authorization Service (JAAS) has in Java applications
- Use JAAS in conjunction with a Java application for both authentication and authorization
- Understand the basics of Java Cryptography (JCA) and Encryption (JCE) and where they fit in the overall security picture
- Understand the fundamentals of XML Digital Signature and XML Encryption
- Understand and implement the processes and measures associated with the Secure Software Development (SSD)
- Acquire the skills, tools, and best practices for design and code reviews as well as testing initiatives
- Understand the basics of security testing and planning
- Work through a comprehensive testing plan for recognized vulnerabilities and weaknesses

# Secure Java Application Development

## Course Summary (cont'd)

### Topics

- Defensive Coding Overview
- Vulnerabilities
- Java Security Fundamentals
- Cryptography Overview
- Code Level Security
- User-Based J2SE Security
- Java Network Security
- Code Level Security Best Practices
- Defending XML Processing
- Secure Software Development (SSD)
- Security Testing

### Audience

This is an intermediate-level course designed for application project stakeholders who wish to get up and running on developing well defended Java applications.

### Prerequisites

Familiarity with the Java programming language is required, and real world programming experience is highly recommended.

### Duration

Two days

ProTech
protechtraining.com

*"Charting the Course ...*

*... to Your Success!"*

# Secure Java Application Development

## Course Outline

**I. Defensive Coding Overview**
- A. Misconceptions
  1. Thriving Industry of Identify Theft
  2. Dishonor Roll of Data Breaches
  3. TJX: Anatomy of a Disaster
  4. Heartland: What? Again?
- B. Security Concepts
  1. Terminology and Players
  2. Assets, Threats, and Attacks
  3. OWASP
  4. CWE/SANS Top 25 Programming Errors
  5. Categories
  6. What they mean to your applications
- C. Defensive Coding Principles
  1. Security Is A Lifecycle Issue
  2. Minimize Attack Surface
  3. Manage Resources
  4. Application States
  5. Compartmentalize
  6. Defense In Depth - Layered Defense
  7. Consider All Application States
  8. Not Trusting The Untrusted
  9. Security Defect Mitigation
  10. Leverage Experience
- D. Reality
  1. Recent, Relevant Incidents
  2. Find Security Defects In Web Application

**II. Vulnerabilities**
- A. Unvalidated Input - XSS/CSRF, Injection, and Others
- B. Broken Authentication and Authorization
- C. Information Leakage - Error Handling, Logging, Insecure Storage and Others
- D. Spoofing - Protecting Your Users and Your Applications

**III. Java Security Fundamentals**
- A. Perimeter Defenses
- B. Java Security Architecture
- C. JVM Defenses
- D. Extending the Defenses

**IV. Cryptography Overview**
- A. Cryptography Defined
- B. Strong Encryption
- C. Ciphers and Algorithms
- D. Message Digests
- E. Keys and Key Management
- F. Types of Keys
- G. Key Management
- H. Certificate Management
- I. Encryption/Decryption
- J. Working with JCE and JCA
- K. Current Best Practices

**V. Code Level Security**
- A. Java 2 Security
- B. Working With Java 2 Security
- C. Signing Code
- D. Trusted Code
- E. Java Permission Management
- F. Extending Java Permissions

**VI. User-Based J2SE Security**
- A. JAAS Overview
- B. JAAS Authentication
- C. Extending JAAS Authentication
- D. JAAS Authorization

**VII. Java Network Security**
- A. SSL Support
- B. Https
- C. GSS
- D. SASL Protocols

**VIII. Code Level Security Best Practices**
- A. What Java Security Provides For
- B. Preventing Remote Hacking
- C. Preventing Accessing Of Restricted Resources
- D. Retaining Credibility with Java Code

**IX. Defending XML Processing**
- A. Defending XML
- 1. Understanding Common Attacks And How To Defend
- 2. Operating In Safe Mode
- 3. Using Standards-Based Security
- 4. XML-Aware Security Infrastructure

# Secure Java Application Development

## Course Outline (cont'd)

**X. Secure Software Development (SSD)**
- A. SSD Process Overview
  1. CLASP Defined
  2. CLASP Applied
- B. Asset, Boundary, and Vulnerability Identification
- C. Vulnerability Response
- D. Design and Code Reviews
- E. Applying Processes and Practices
- F. Risk Analysis

**XI. Security Testing**
- A. Testing as Lifecycle Process
- B. Testing Planning and Documentation
- C. Testing Tools and Processes
  1. Principles
  2. Reviews
  3. Testing
  4. Tools
- D. Static and Dynamic Code Analysis
- E. Testing Practices
  1. Authentication Testing
  2. Data Validation Testing
  3. Denial of Service Testing