

Securing J2EE Web Applications – Lifecycle (TT8325)

Duration: 4 days

Course Description: Securing J2EE Web Applications - Lifecycle is a lab-intensive, hands-on Java / J2EE security training course, essential for experienced enterprise developers who need to produce secure J2EE-based web applications. In addition to teaching basic programming skills, this course digs deep into sound processes and practices that apply to the entire software development lifecycle. In this course, students thoroughly examine best practices for defensively coding J2EE web applications, including XML processing and web services. Students will repeatedly attack and then defend various assets associated with a fully functional web application. This hands-on approach drives home the mechanics of how to secure J2EE web applications in the most practical of terms.

Audience This is an intermediate -level J2EE / web services programming course, designed for developers who wish to get up and running on developing well defended software applications. This course may be customized to suit your team’s unique objectives.

Prerequisites: Familiarity with Java and J2EE is required and real world programming experience is highly recommended. Ideally, students should have approximately 6 months to a year of Java and J2EE working knowledge. Students should have an understanding and a working knowledge in the following topics, or attend these courses as a pre-requisite: Core Java Programming for OO Developers (C++, etc) or TT2120 Java Fundamentals for Non-OO Programmers or TT5140 Core Java Programming for Server Side Developers New to OO or TT5100 Fast Track to J2EE Fundamentals (Servlets/JSPs, JDBC and more)

Topics

SESSION: FOUNDATION

- Find Security Defects In Web Application

MISCONCEPTIONS

- Thriving Industry of Identify Theft
- Dishonor Roll of Data Breaches
- TJX: Anatomy of a Disaster
- Heartland: What? Again?

SECURITY CONCEPTS

- Terminology and Players
- Assets, Threats, and Attacks
- OWASP
- CWE/SANS Top 25 Programming Errors
- Categories
- What they mean to your web applications

DEFENSIVE CODING PRINCIPLES

- Reality
- Recent, Relevant Incidents

SESSION: TOP SECURITY VULNERABILITIES

UNVALIDATED INPUT

- Description With Working Example
- Defenses
- Identifying Trust Boundaries
- Qualifying Untrusted Data
- Implementing An Effect, Layered Defense
- Designing An Appropriate Response
- Testing Defenses And Responses

OVERVIEW OF REGULAR EXPRESSIONS

- Regular Expressions
- Working with Regular Expressions in Java

Securing J2EE Web Applications – Lifecycle (TT8325)

BROKEN ACCESS CONTROL

- Description With Working Example
- Defenses
- Authorization Security Overview
- Defending Special Privileges Such As Administrative Functions
- Application Authorization Best Practices

BROKEN AUTHENTICATION AND SESSION MANAGEMENT

- Description With Working Example
- Defenses
- Multi-Layered Defenses Of Authentication Services
- Password Management Strategies
- Password Handling With Hashing Using JCE/JCA
- Mitigating Password Caching
- Testing Defenses And Responses For Weaknesses
- Alternative Authentication Mechanisms
- Best Practices For Session Management in J2EE
- Defending Session Hijacking Attacks
- Best Practices For Single Sign-On (SSO)

CROSS SITE SCRIPTING (XSS) FLAWS

- Description With Working Example
- Defenses
- Character Encoding Complications
- Blacklisting
- Whitelisting
- HTML/XML Entity Encoding
- Trust Boundary Definition
- Implementing An Effective Layered Defense
- Designing An Appropriate Response
- Cross-Site Request Forgeries (CSRF)
- Understanding CSRF
- Defending Against CSRF
- Output Encoding – Why
- Output Encoding – How
- Output Encoding – Best Practices

INJECTION FLAWS

- Description With Working Example

- Defenses
- Qualifying Untrusted Data
- JDBC, PreparedStatement, and StoredProcedures
- Hibernate Best Practices
- XML Best Practices
- Third Party API's
- Implementing An Effective Layered Defense
- Designing An Appropriate Response

ERROR HANDLING AND INFORMATION LEAKAGE

- Description With Working Example
- Defenses
- J2EE Application Exception Handling
- Error Response Best Practices
- Error, Auditing, And Logging Content Management
- Error, Auditing, And Logging Service Management
- Best Practices For Supporting Web Attack Forensics

INSECURE STORAGE

- Description With Working Example
- Defenses
- Data Leakage
- Risk Minimization
- Cryptography Overview
- JCS/JCE
- Data Encryption
- Partial/Complete
- Property/Deployment/Configuration Files

INSECURE MANAGEMENT OF CONFIGURATION

- Description with working example
- Defenses
- System hardening
- J2EE application server configuration "Gotchas!"
- Hardening software installation

DIRECT OBJECT ACCESS

- Description With Working Example

Securing J2EE Web Applications – Lifecycle (TT8325)

- Defenses
- Java Byte Code Verifier
- XML/DTD/Schema/XSLT Best Practices
- Message-Level Security
- WS-Security
- Attacks And Defenses

SPOOFING

- Description With Working Example
- Defenses
- Protecting Your Clients
- Defending Against Cross Site Request Forgeries
- Phishing Defenses

SESSION: BEST PRACTICES

BEST PRACTICES AND PRINCIPLES

- Security Is A Lifecycle Issue
- Minimize Attack Surface
- Manage Resources
- Application States
- Compartmentalize
- Defense In Depth - Layered Defense
- Consider All Application States
- Not Trusting The Untrusted
- Security Defect Mitigation
- Leverage Experience

JAVA BEST PRACTICES

- Code Obfuscation
- JAAS Usage
- Java 2 Security and Policy Files
- Signing JAR Files

SESSION: DEFENDING XML PROCESSING

DEFENDING XML

- Understanding Common Attacks And How To Defend
- Operating In Safe Mode
- Using Standards-Based Security
- XML-Aware Security Infrastructure
- JAXP Safe Mode

DEFENDING WEB SERVICES

- Security Exposures
- Transport-Level Security

DEFENDING AJAX

- Ajax Security Exposures
- Attack Surface Changes
- Injection Threats And Concerns
- Effective Defenses And Practices

SESSION: SECURE SOFTWARE DEVELOPMENT (SSD)

SSD PROCESS OVERVIEW

- CLASP Defined
- CLASP Applied
- Asset, Boundary, and Vulnerability Identification
- Vulnerability Response
- Design and Code Reviews
- Applying Processes and Practices
- Risk Analysis

SESSION: SECURITY TESTING

- Testing as Lifecycle Process
- Testing Planning and Documentation
- Testing Tools And Processes
 - Principles
 - Reviews
 - Testing
 - Tools

STATIC AND DYNAMIC CODE ANALYSIS

TESTING PRACTICES

- Authentication Testing
- Session Management Testing
- Data Validation Testing
- Denial Of Service Testing
- Web Services Testing
- Ajax Testing



Securing J2EE Web Applications – Lifecycle (TT8325)

SESSION: APPENDIX: SECURITY DESIGN PATTERNS

DESIGN PATTERNS INTRODUCTION

J2EE WEB APPLICATION SECURITY DESIGN PATTERNS

- Authentication Enforcer
- Authorization Enforcer
- Intercepting Validator
- Secure Base Action
- Secure Logger
- Secure Pipe
- Secure Service Proxy
- Intercepting Web Agent