Secure Java Web Application Development Lifecycle (SDL)

Course Summary

Description

The ***Best Defense™ Security Training Series*** is a suite of developer-oriented, application security courses that provide complete coverage of the CWE/SANS ***Top 25 Most Dangerous Programming Errors*** the OWASP Top Ten for 2013, the Verizon 2014 Data Breach Report, and the WASC Threat Classifications. These errors, as determined by a consortium of cyber security organizations, enable cyber espionage and crime. Our comprehensive application security and secure coding classes address each of these critical issues head-on, as our courses, seminars and workshops explicitly:

- Teach programmers what these errors are
- Demonstrate, in real terms, the potential impact of each of these errors
- Provide experience in how to recognize and properly address these errors
- Teach stakeholders how to defend against the potential consequences of security breaches in other parts of their IT infrastructure.
- Cross-reference materials, vulnerabilities, and attacks that are covered with both the OWASP Top 10 and the WASC Threat Classifications
- Covers the latest security trends and developments, including the Verizon Data Breach Report and the latest from the National Vulnerabilities Database

Secure Java Web Application Development Lifecyle (SDL) is a lab-intensive, hands-on Java / JEE security training course, essential for experienced enterprise developers who need to engineer, maintain, and support secure JEE-based web applications. In addition to teaching basicsecure 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 web applications, including XML processing, rich interfaces, and both RESTful and SOAP-based web services. Students will repeatedly attack and then defend various assets associated with fully-functional web applications and web services. This hands-on approach drives home the mechanics of how to secure JEE web applications in the most practical of terms.

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.

A key component to our ***Best Defense IT Security Training Series***, this workshop is a companion course with several developer-oriented courses and seminars. Although this edition of the course is Java-specific, it may also be presented using .Net or other programming languages

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

- Understand potential sources for untrusted data
- Understand the consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections
- Be able to test web applications with various attack techniques to determine the existence of and effectiveness of layered defenses
- Prevent and defend the many potential vulnerabilities associated with untrusted data
- Understand the vulnerabilities of associated with authentication and authorization
- Be able to detect, attack, and implement defenses for authentication and authorization functionality and services
- Understand the dangers and mechanisms behind Cross-Site Scripting (XSS) and Injection attacks

Secure Java Web Application Development Lifecycle (SDL)

Course Summary (cont'd)

- Be able to detect, attack, and implement defenses against XSS and Injection attacks
- Understand the concepts and terminology behind defensive, secure, coding
- Understand the use of Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in Java-based web applications
- Design and develop strong, robust authentication and authorization implementations within the context of JEE
- Understand the fundamentals of XML Digital Signature and XML Encryption as well as how they are used within the web services arena
- Be able to detect, attack, and implement defenses for both RESTful and SOAP-based web services and functionality
- Understand techniques and measures that can used to harden web and application servers as well as other components in your infrastructure
- 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

Topics

- Introduction: Misconceptions
- Foundation
- Vulnerabilities
- Defending XML, Services, and Rich Interfaces
- Secure Development Lifecycle (SDL)
- Security Testing

Audience

This is an intermediate -level JEE / 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 JEE is required and real world programming experience is highly recommended. Ideally students should have approximately 6 months to a year of Java and JEE working knowledge.

Duration

Five days

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Secure Java Web Application Development Lifecycle (SDL)

Course Outline

I. Misconceptions
- A. Security: The Complete Picture
- B. TJX: Anatomy of a Disaster?
- C. Causes of Data Breaches
- D. Heartland – Slipping Past PCI Compliance
- E. Target's Painful Christmas
- F. Meaning of Being Compliant
- G. Verizon's 2013 and 2014 Data Breach Reports

II. Session: Foundation
- A. Security Concepts
  1. Motivations: Costs and Standards
  2. Open Web Application Security Project
  3. Web Application Security Consortium
  4. CERT Secure Coding Standards
  5. Assets are the Targets
  6. Security Activities Cost Resources
  7. Threat Modeling
  8. System/Trust Boundaries
- B. Principles of Information Security
  1. Security Is a Lifecycle Issue
  2. Minimize Attack Surface Area
  3. Layers of Defense: Tenacious D
  4. Compartmentalize
  5. Consider All Application States
  6. Do NOT Trust the Untrusted

III. Vulnerabilities
- A. Unvalidated Input
  1. Buffer Overflows
  2. Integer Arithmetic Vulnerabilities
  3. Unvalidated Input: From the Web
  4. Defending Trust Boundaries
  5. Whitelisting vs Blacklisting
- B. Overview of Regular Expressions
  1. Regular Expressions
  2. Working With Regexes in Java
  3. Applying Regular Expressions
- C. Broken Access Control
  1. Access Control Issues
  2. Excessive Privileges
  3. Insufficient Flow Control
  4. Unprotected URL/Resource Access
  5. Examples of Shabby Access Control
  6. Session and Session Management
- D. Broken Authentication
  1. Broken Quality/DoS
  2. Authentication Data
  3. Username/Password Protection
  4. Exploits Magnify Importance
  5. Handling Passwords on Server Side
  6. Single Sign-on (SSO)
- E. Cross Site Scripting (XSS)
  1. Persistent XSS
  2. Reflective XSS
  3. Best Practices for Untrusted Data
- F. Injection
  1. Injection Flaws
  2. SQL Injection Attacks Evolve
  3. Drill Down on Stored Procedures
  4. Other Forms of Injection
  5. Minimizing Injection Flaws
- G. Error Handling and Information Leakage
  1. Fingerprinting a Web Site
  2. Error-Handling Issues
  3. Logging In Support of Forensics
  4. Solving DLP Challenges
- H. Insecure Data Handling
  1. Protecting Data Can Mitigate Impact
  2. In-Memory Data Handling
  3. Secure Pipes
  4. Failures in the SSL Framework Are Appearing
- I. Insecure Configuration Management
  1. System Hardening: IA Mitigation
  2. Application Whitelisting
  3. Least Privileges
  4. Anti-Exploitation
  5. Secure Baseline
- J. Direct Object Access
  1. Dynamic Loading
  2. Race Conditions
  3. Direct Object References
- K. Spoofing, CSRF, and Redirects
  1. Name Resolution Vulnerabilities
  2. Fake Certs and Mobile Apps
  3. Targeted Spoofing Attacks
  4. Cross Site Request Forgeries (CSRF)
  5. CSRF Defenses are Entirely Server-Side
  6. Safe Redirects and Forwards

Secure Java Web Application Development Lifecycle (SDL)

Course Outline (cont'd)

L. Cryptography Overview
   1. Strong Encryption
   2. Message digests
   3. Keys and key management
   4. Certificate management
   5. Encryption/Decryption

M. Understanding What's Important
   1. Common Vulnerabilities and Exposures

N. OWASP Top Ten for 2013
   1. CWE/SANS Top 25 Most Dangerous SW Errors
   2. Monster Mitigations
   3. Strength Training: Project Teams/Developers
   4. Strength Training: IT Organizations

IV. Defending XML, Services, and Rich Interfaces

A. Defending XML
   1. XML Signature
   2. XML Encryption
   3. XML Attacks: Structure
   4. XML Attacks: Injection
   5. Safe XML Processing

B. Defending Web Services
   1. Web Service Security Exposures
   2. When Transport-Level Alone is NOT Enough
   3. Message-Level Security
   4. WS-Security Roadmap
   5. XWSS Provides Many Functions
   6. Web Service Attacks
   7. Web Service Appliance/Gateways

C. Defending Rich Interfaces and REST
   1. How Attackers See Rich Interfaces
   2. Attack Surface Changes When Moving to Rich Interfaces
   3. Bridging and its Potential Problems
   4. Three Basic Tenets for Safe Rich Interfaces
   5. OWASP REST Security Recommendations

V. Secure Development Lifecycle (SDL)

A. SDL Process Overview
   1. Software Security Axioms
   2. Security Lifecycle – Phases
   3. Lesson: Applying Processes and Practices
   4. Awareness
   5. Application Assessments
   6. Security Requirements
   7. Secure Development Practices
   8. Security Architecture/Design Review
   9. Security Code Review
   10. Configuration Management and Deployment
   11. Vulnerability Remediation Procedures

B. Risk Analysis
   1. Threat Modeling Process
   2. Identify Security Objectives
   3. Describe the System
   4. List Assets
   5. Define System/Trust Boundaries
   6. List and Rank Threats
   7. List Defenses and Countermeasures

VI. Security Testing

A. Testing Tools and Processes
   1. Security Testing Principles
   2. Black Box Analyzers
   3. Static Code Analyzers
   4. Criteria for Selecting Static Analyzers

B. Testing Practices
   1. OWASP Web App Penetration Testing
   2. Authentication Testing
   3. Session Management Testing
   4. Data Validation Testing
   5. Denial of Service Testing
   6. Web Services Testing
   7. Ajax Testing