

SC SCR 101: SOURCE CODE REVIEW FOR JAVA/JEE – PCI COMPLIANCE

Taking this class will empower students to perform code reviews that are compliant with Payment Card Industry Data Security Standard (PCI DSS) Requirements 6.3.7 & 6.6. Students will learn to review existing applications and discover vulnerabilities.

Application security attacks are an increasingly prevalent risk in today's technological landscape. Experts are recognizing that the most cost effective strategy to mitigate this risk is to eliminate defects in code. Organizations can achieve this goal by identifying and remediating defects or creating more secure code from the start. Leverage Security Compass's extensive knowledge of Java security along with its innovative training programs to help deliver highly trusted applications. Students will be given extensive experience reviewing real open source applications for vulnerabilities within the Open Web Application Security Project (OWASP) Top 10 and beyond. Students learn attacks in-depth and how to find vulnerabilities that static source code analysis tools can't find.

DURATION & INTENDED AUDIENCE

- Duration: 3 days
- Intended Audience:
 - Java developers, testers, architects and business analysts
 - Information security analysts who perform source code reviews

PRE-REQUISITE KNOWLEDGE

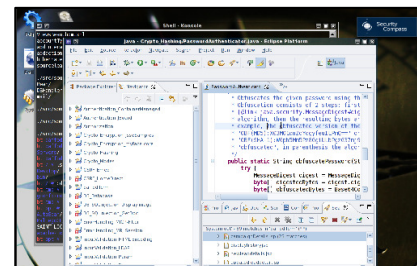
- Strong knowledge of Java Enterprise Edition programming mandatory
- Basic knowledge of Apache Struts recommended
- Basic knowledge of Unix/Linux recommended
- Knowledge of web application security **not** required

LEARNING OBJECTIVES

- Understand how the Payment Card Industry Data Security Standards (PCI DSS) affect Java applications
- Identify Java source code that leads to non-compliance with PCI
- Identify adequate security controls in Java source code
- Prioritize security review findings by risk)

LEARNING ENVIRONMENT & TAKEAWAYS

- **Every section of days 2 and 3 feature hands-on examples**
- 4 GB bootable Backtrack Linux flash drive that students keep
- Eclipse, Tomcat, My SQL, and source code from over 30 different **real** open source applications
- Course book containing printouts of each slide along with detailed notes in paragraph form
- Hard-copy and soft copy of a source code review checklist to assist in performing source code reviews in the real world



CONTACT INFO

- Phone: 1-888-777-2211 x 1
- Email: training@securitycompass.com

DETAILED OUTLINE

DAY 1: INTRO

Authentication

- Factors of Authentication
- User Enumeration
- Password Reset
- Brute Force
- Password Sniffing

Authorization

- Privilege Escalation
- Page Authorization
- Functional Authorization
- Data Authorization

Session Management

- Session Hijacking
- Content Caching

Cryptography

- Random Numbers
- Hashes
- Rainbow Tables
- Symmetric Key Encryption
- Asymmetric key Encryption
- Cryptographic Flaws

Input Validation

- Input Validation Overview
- Parameter Manipulation
- XSS & CSRF
- HTTP Response Splitting
- SQL Injection
- XML Parsers & Validators
- XML Attack

DAY 2: SOURCE CODE REVIEW

Source Code Review Background

- Automated approaches
- Manual approaches
- The combined approach
- Point of entry review

Authentication and Authorization

- Identifying authentication & authorization mechanisms
- Common authentication errors in code
- Common authorization errors in code
- Reviewing applications for authentication & authorization

Session Management

- Identifying session management mechanisms
- Identifying session management vulnerabilities in code
- Reviewing applications for session management vulnerabilities

Input Validation

- Identifying validation mechanisms
- Identifying encoding mechanisms
- Common frameworks: Apache Struts input validation

DAY 3: SOURCE CODE REVIEW

Input Validation Continued

- Identifying input validation vulnerabilities in code
- Reviewing applications for input validation

Database Issues

- Identifying database code
- Identifying SQL injection
- Database connection strings
- Access rights of application in database
- Reviewing applications for database security

Assorted Topics

- Identifying logging mechanisms
- Identifying logging injection vulnerabilities
- Identifying error handling mechanisms
- Identifying gaps in error handling
- Identifying cryptographic controls
- Identifying gaps in cryptographic controls
- Reviewing applications for cryptographic controls