In this course, you will examine best practices for defensively coding JEE web applications, including XML processing and web services. You 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 JEE web applications in the most practical of terms.

### Skills Gained

- Potential sources for untrusted data
- Consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections
- 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
- Vulnerabilities of associated with authentication and authorization
- Detect, attack, and implement defenses for authentication and authorization functionality and services
- Dangers and mechanisms behind Cross-Site Scripting (XSS) and Injection attacks
- Detect, attack, and implement defenses against XSS and Injection attacks
- Concepts and terminology behind defensive, secure, coding
- Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against 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
- Fundamentals of XML Digital Signature and XML Encryption as well as how they are used within the web services arena
- Detect, attack, and implement defenses for XML-based services and functionality
- Techniques and measures that can used to harden web and application servers as well as other components in your infrastructure

### Who Can Benefit

Developers who wish to develop secure applications

### Prerequisites

- Familiarity with Java and JEE is required
- Programming experience is highly recommended
- At least six months of Java and JEE working knowledge recommended
- You should have a working knowledge in the following topics or attend these courses as a prerequisite:
- Building JEE Web Applications (TT5100-JEE)
- Java 7 SE Programming for OO Experienced Developers (TT2100-J7)
- Java 7 Essentials for Object Oriented (OO) Developers (TT2101-J7)
- Java Web Essentials for OO Developers (TT5140)

Course Details

Introduction: Misconceptions
- Security: The Complete Picture
- TJX: Anatomy of a Disaster?
- Causes of Data Breaches
- Heartland - Slipping Past PCI Compliance
- Target's Painful Christmas
- Meaning of Being Compliant
- Verizon's 2013 Data Breach Report

Foundation
- Security Concepts
- Motivations: Costs and Standards
- Open Web Application Security Project
- Web Application Security Consortium
- CERT Secure Coding Standards
- Assets are the Targets
- Security Activities Cost Resources
- Threat Modeling
- System/Trust Boundaries
- Principles of Information Security
- Security Is a Lifecycle Issue
- Minimize Attack Surface Area
- Layers of Defense: Tenacious D
- Compartamentalize
- Consider All Application States
- Do Not Trust the Untrusted

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

**Best Practices**
• Cryptography Overview
• Strong Encryption
• Message digests
• Keys and key management
• Certificate management
• Encryption/Decryption
• Understanding What's Important
• Common Vulnerabilities and Exposures
• OWASP Top Ten for 2013
• CWE/SANS Top 25 Most Dangerous SW Errors
• Monster Mitigations
• Strength Training: Project Teams/Developers
• Strength Training: IT Organizations

**Defending XML, Services, and Rich Interfaces**
• Defending XML
• XML Signature
• XML Encryption
• XML Attacks: Structure
• XML Attacks: Injection
• Safe XML Processing
• Defending Web Services
• Web Service Security Exposures
• When Transport-Level Alone is NOT Enough
• Message-Level Security
• WS-Security Roadmap
• XWSS Provides Many Functions
• Web Service Attacks
• Web Service Appliance/Gateways
• Defending Rich Interfaces and REST
• How Attackers See Rich Interfaces
• Attack Surface Changes When Moving to Rich Interfaces
• Bridging and its Potential Problems
• Three Basic Tenets for Safe Rich Interfaces
• OWASP REST Security Recommendations