

Course 913: Secure Software Design (3 days)

Course Description...

According to research by the National Institute of Standards, 92% of all security vulnerabilities are now considered application vulnerabilities and not network vulnerabilities. This intense hands-on workshop is essential for software application designers and architects who need to build secure Java and J2EE applications.

Throughout the course, students learn the best practices for designing and architecting secure programs in Java and J2EE. Students will take an application from requirements through to implementation, analyzing and testing for software vulnerabilities. This approach builds a strong appreciation for why software needs to be designed from the ground up in a secure fashion.

Attendees leave the course armed with the required skills to recognize software vulnerabilities (actual and potential) and design defenses for those vulnerabilities. This course quickly introduces developers to the various types of threats against their software.

The concept and process of Threat Risk Modeling is introduced as a key enabler for architecting effective and appropriate security for software and information assets.

Hands-On Workshops...

During this course, students will be led through a series of progressively advanced topics, where most topics consist of lecture, group discussion, comprehensive hands-on lab exercises, and lab review.

This workshop is about 40% hands-on lab and 60% lecture. Multiple complete “mini-projects” are laced throughout the course, designed to reinforce fundamental skills and concepts learned in the lessons. Because these lessons, labs and projects are presented in a building block fashion, students will gain a solid understanding of not only the core concepts, but also how all the pieces fit together in a complete application.

At the end of each lesson, developers will be tested with a set of review questions to ensure that he/she has fully understands that topic.

Learning Objectives...

- Understand the concepts and terminology behind defensive coding.
- Understand and use Threat Risk 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 Risk Modeling to identify potential vulnerabilities in a real life case study.
- Understand and implement the processes and measures associated with the security development lifecycle (SDL)
- Acquire the skills, tools, and best practices for design 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

Who should attend...

This is an intermediate level software design course, designed for architects and stakeholders who wish to get up and running on building well defended software applications. This course may be customized to suit your team's unique objectives.

Familiarity with software design and technologies is required, and real world programming experience is highly recommended.

Prerequisites...

Students should have basic development skills and a working knowledge in the following topics, or attend these courses as a pre-requisite:

- Understanding Internet Architectures
- Essential Java Programming
- Building J2EE Web Applications

See next page for a detailed course outline...



Course Outline...

Defensive Coding Overview

- Security Concepts
- Principles of Defensive Coding
- Threat Risk Modeling
- Threat Risk Modeling of Case Study

Vulnerabilities

- Security Attacks
- Information Attacks
- System Attacks
- Data Attacks
- Threat Risk Modeling Revisited

Defensive Coding Applied

- Basic Principles Revisited
- Defensive Coding

Security Design Patterns

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

Security Development Lifecycle (SDL)

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

Security Testing

- Testing as Lifecycle Process
- Testing Planning and Documentation
- Testing Tools
- Approaches for Testing:
 - Information Leakage
 - Business Logic
 - Authentication
 - Session Management
 - Input Data Validation
 - Denial of Service
 - Web Services