Secure .Net Application Development Lifecycle (SDL)

Course Summary

Description

Secure .Net Coding – Lifecycle is a hands-on, lab-intensive .Net security, code-level training course that teaches students the best practices for designing, implementing, and deploying secure programs in .Net. Students will take an application from requirements through to implementation, analyzing and testing for software vulnerabilities. This course explores well beyond basic programming skills, teaching developers sound processes and practices to apply to the entire software development lifecycle. Perhaps just as significantly, students learn about current, real examples that illustrate the potential consequences of not following these best practices. This course is short on theory and long on application, providing students with in-depth, code-level labs.

Objectives

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

- Understand the concepts and terminology behind defensive coding
- Understand and use Threat 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 Modeling to identify potential vulnerabilities in a real life case study
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in .Net applications
- Understand the vulnerabilities of the .Net programming language and the runtime environment as well as how to harden both
- Understand and work with .Net platform security to gain an appreciation for what is protected and how
- Understand the basics of Cryptography and Encryption and where they fit in the overall security picture
- Work with the .Net Cryptographic services
- Examine how role-based security works in .Net and use it to control access
- Examine how Code Access Security (CAS) works and use it to control access
- Understand and work with the mechanics of isolated storage
- Understand the fundamentals of XML Digital Signature and XML Encryption
- 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
- .Net Security
- .Net Security Fundamentals
- Cryptography Overview
- Defending XML and Services
- Secure Development Lifecycle (SDL)
- Security Testing

Audience

This is an intermediate-level Java programming course designed for application project stakeholders who wish to get up and running on developing well defended Java applications.

Prerequisites

Familiarity with the Java programming language is required, and real world programming experience is highly recommended.

Duration

Four days

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Secure .Net Application Development Lifecycle (SDL)

Course Outline

I. Introduction: Misconceptions
- A. Security: The Complete Picture
  1. TJX: Anatomy of a Disaster?
  2. Causes of Data Breaches
  3. Heartland – Slipping Past PCI Compliance
  4. Target's Painful Christmas
  5. Meaning of Being Compliant
  6. Verizon's 2013 and 2014 Data Breach Reports

II. 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
- C. Vulnerabilities
  1. Unvalidated Input
  2. Broken Access Control
  3. Broken Authentication And Session Management
  4. Cross Site Scripting (XSS) Flaws
  5. Injection Flaws
  6. Error Handling And Information Leakage
  7. Insecure Storage
  8. Insecure Management of Configuration
  9. Direct Object Access
  10. Spoofing and Redirects
- D. Understanding What's Important
  1. Common Vulnerabilities and Exposures
  2. OWASP Top Ten for 2013
  3. CWE/SANS Top 25 Most Dangerous SW Errors
  4. Monster Mitigations
  5. Strength Training: Project Teams/Developers
  6. Strength Training: IT Organizations

III. .Net Security

IV. .Net Security Fundamentals
- A. .Net Security Overview
  1. Services Provided
  2. Code Protections
  3. Data Protections
- B. .NET Assembly Security
  1. The role of Application Domains
  2. Protecting assemblies from tampering
  3. Using obfuscation
  4. Using publisher certificates
  5. Using FxCop.exe

V. Cryptography Overview
- A. Strong Encryption
  1. Message digests
  2. Keys and key management
  3. Certificate management
  4. Encryption/Decryption
- B. .NET Cryptographic Services
  1. The role of cryptographic services
  2. Hash algorithms and hash codes
  3. Encrypting data symmetrically
  4. Encrypting data asymmetrically
- C. Understanding Role Based Security
  1. Using role based security
  2. Creating and administering roles
  3. Principals, identity and roles
  4. Determining role membership
  5. Restricting actions based on roles
- D. Code Access Security
  1. What is Code Access Security (CAS)
  2. CAS components
  3. Using CAS to secure applications"
  4. Interacting with CAS
- E. Isolated Storage
  1. The purpose of Isolated Storage
  2. Levels of isolated storage
  3. Using isolated storage administrative tools
  4. Working with isolated storage programmatically

Secure .Net Application Development Lifecycle (SDL)

Course Outline (cont'd)

VI. Defending XML and Services
- 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

VII. Secure Development Lifecycle (SDL)
- A. SDL Process Overview
  1. Software Security Axioms
  2. Security Lifecycle – Phases
- B. Applying Processes and Practices
  1. Awareness
  2. Application Assessments
  3. Security Requirements
  4. Secure Development Practices
  5. Security Architecture/Design Review
  6. Security Code Review
  7. Configuration Management and Deployment
  8. Vulnerability Remediation Procedures
- C. Risk Analysis
  1. Threat Modeling Process
  2. 1. Identify Security Objectives
  3. 2. Describe the System
  4. 3. List Assets
  5. 4. Define System/Trust Boundaries
  6. 5. List and Rank Threats
  7. 6. List Defenses and Countermeasures

VIII. 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