Secure Software Design

Course Summary

Description

At the core of the recent Heartbleed vulnerability was a violation of a basic principle of secure coding – validate untrusted data. According to research by the National Institute of Standards, 92% of all security vulnerabilities are considered application vulnerabilities and not network vulnerabilities. Retailers, financial institutions, government agencies, high-tech companies, and many others are paying the price for poor application security - financial losses and eroding trust.

PCI Compliant Developer Training

This secure coding training addresses common coding vulnerabilities in software development processes. This training is used by one of the principle participants in the PCI DSS. Having passed multiple PCI audits, this course has been shown to meet the PCI requirements. The specification of those training requirements are detailed in 6.5.1 through 6.5.10 on pages 55 through 59 of the PCI DSS Requirements 3.0 document dated November, 2013. This is not "checklist mentality" training as it integrates demonstrations, code flashes, and hands-on labs for vulnerabilities, defenses, and best practices in secure development lifecycle (SDL).

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
- 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

Topics

- Misconceptions
- Foundation
- Vulnerabilities
- Secure Development Lifecycle (SDL)
- Security Testing

Audience

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.

Prerequisites

Familiarity with software design and technologies is required and real world programming experience is highly recommended. Ideally, students should have approximately 6 months to a year of working knowledge of a programming language.

Duration

Four days

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

"Charting the Course ...

... to Your Success!"

Secure Software Design

Course Outline

I. **Introduction: 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. **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. Vulnerabilities
     1. Unvalidated Input
     2. Broken Authentication
     3. Cross Site Scripting (XSS/CSRF)
     4. Injection Flaws
     5. Error Handling, Logging, and Information Leakage
     6. Insecure Storage
     7. Direct Object Access
     8. XML Vulnerabilities
     9. Web Services Vulnerabilities
     10. Ajax Vulnerabilities
   - B. 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
   - C. Security Design Patterns
     1. Authentication Enforcer
     2. Authorization Enforcer
     3. Intercepting Validator
     4. Secure Base Action
     5. Secure Logger
     6. Secure Pipe
     7. Secure Service Proxy
     8. Intercepting Web Agent

IV. **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

V. **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

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically