Specialized - Securing .Net Web Applications

Code: TT8320-N
Length: 4 days
URL: View Online

Securing .Net Web Applications is a lab-intensive, hands-on .Net security training course, essential for experienced enterprise developers who need to produce secure .Net-based web applications. In addition to teaching basic programming skills, this course digs deep into sound processes and practices that apply to the entire software development lifecycle.

- In this course, students thoroughly examine best practices for defensively coding .Net web applications, including XML processing and web services. Students 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 .Net web applications in the most practical of terms.

Skills Gained
- Understand potential sources for untrusted data
- Understand the consequences for not properly handling untrusted data such as denial of service, cross-site scripting, and injections
- Be able to 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
- Understand the vulnerabilities of associated with authentication and authorization
- Be able to detect, attack, and implement defenses for authentication and authorization functionality and services
- Understand the dangers and mechanisms behind Cross-Site Scripting (XSS) and Injection attacks
- Be able to detect, attack, and implement defenses for authentication and authorization functionality and services
- Understand the dangers and mechanisms behind Cross-Site Scripting (XSS) and Injection attacks
- Be able to detect, attack, and implement defenses against XSS and Injection attacks
- Understand the concepts and terminology behind defensive, secure, coding
- Understand the use of Threat Modeling as a tool in identifying software vulnerabilities based on realistic threats against meaningful assets
- Perform both static code reviews and dynamic application testing to uncover vulnerabilities in web applications
- Design and develop strong, robust authentication and authorization implementations within the context of .NET
- Understand the fundamentals of XML Digital Signature and XML Encryption as well as how they are used within the web services arena
- Be able to detect, attack, and implement defenses for XML-based services and functionality
- Understand techniques and measures that can used to harden web and application servers as well as other components in your infrastructure
Who Can Benefit
This is an intermediate-level .Net secure programming course, designed for developers who wish to get up and running on developing well defended software applications. This course may be customized to suit your team’s unique objectives.

- Familiarity with C# is required and real world programming experience is highly recommended. Ideally students should have approximately 6 months to a year of .Net application development practical experience.

Course Details

Foundation
Misconceptions

- Misconception #1
- Security: The Complete Picture
- TJX: Anatomy of a Disaster?
- So What is the Point?
- 2012 Attacks Continued to Evolve
- Causes of Data Breaches
- Heartland – Slipping Past PCI Compliance
- What's the Point?
- Verizon’s 2012 Data Breach Report
- 360M Down to 4M in 2010???
- US Secret Service Continued to Battle
- Verizon’s 2013 Data Breach Report
- The Numbers are Abstract, but…
- Are You Concerned Yet?
- Verizon AppSec Recommendations

Security Concepts

- Terminology and Players
- Assets, Threats, and Attacks
- OWASP
- WASC

Defensive Coding Principles

- Security Is a Lifecycle Issue
- Bolted on Versus Baked
- Minimize Attack Surface Area
- Examples of Minimization
- Defense in Depth
- Manage Resources
- Layers of Defense: Tenacious D
Reality

- Recent, Relevant Incidents
- Finding Security Defects In Web Applications

**Top Security Vulnerabilities**

Unvalidated Input

- OWASP/WASC Coverage
- Unvalidated Input: Description
- Buffer Overflows
- Format String Attacks
- Null Byte Injection Attacks
- Integer Arithmetic Vulnerabilities
- Unvalidated Input: From the Web
- Hidden Values in HTTP Communications
- Unvalidated Input: Symptoms and Detection
- Detection Through Fuzz Testing
- Unvalidated Input: Fixes
- Identifying Trust Boundaries
- Designing An Appropriate Response
- Testing Defenses And Responses

Overview of Regular Expressions

- Description with working example

Broken Access Control

- OWASP/WASC Coverage
- Access Control Issues
- Broken Access Control: Description
- Excessive Privileges
- Insufficient Flow Control/Forceful Browsing
- Primary Concerns in URL/Resource Access
- Unprotected URL/Resource Access: Fixes
- Protecting Sessions
- Addressing Client-Side Caching of Content
* Authorization Security Overview
* .Net authorization security overview
* Defending Special Privileges Such As Administrative Functions
* Application Authorization Best Practices

**Broken Authentication And Session Management**

* OWASP/WASC Coverage
* Quality of Authentication Credentials
* Multi-Layered Defenses Of Authentication Services
* Password Management Strategies
* Password Handling With Hashing
* Mitigating Password Caching
* Testing Defenses And Responses For Weaknesses
* Alternative Authentication Mechanisms
* Best Practices For Session Management
* Defending Session Hijacking Attacks

**Cross Site Scripting (XSS) Flaws**

* OWASP/WASC Coverage
* XSS Mechanisms
* Character Encoding Complications
* Blacklisting
* Whitelisting
* HTML/XML Entity Encoding
* Trust Boundary Definition
* Implementing An Effective Layered Defense
* Designing An Appropriate Response

**Injection Flaws**

* OWASP/WASC Coverage
* SQL Injection Continues to be Prevalent
* Injection Flaws: Description
* Injection Flaws: Symptoms and Detection
* SQL Injection Examples
* SQL Injection Attacks Evolve
* Attackers have a Variety of Tools
* SQL Injection: Drill Down on Stored Procedures
* SQL Injection: Drill Down on ORM
* Minimize SQL Injection Vulnerabilities
* Minimizing Injection Flaws
Error Handling And Information Leakage

- OWASP/WASC Coverage
- Four Dimensions of Designing Error Responses
- Error Response Best Practices
- Error, Auditing, And Logging Content Management
- Error, Auditing, And Logging Service Management
- Best Practices For Supporting Web Attack Forensics
- Information Leaks
- Data Loss Prevention (DLP)
- Solving DLP Challenges
- DLP: What and Where
- DLP: Best Practices

Insecure Data Handling

- OWASP/WASC Coverage
- Sony and Related Exploits
- Protecting Data can Mitigate Impact of Exploit
- Data Handling Concerns
- Unexpected Data Repositories
- In-Memory Data Handling
- Secure Pipes
- Transport-Level Security
- SSL
  - Recent Failures in SSL Framework
  - BEAST Attacks on SSL

Insecure Management of Configuration

- OWASP/WASC Coverage
- System hardening
- Server configuration “Gotchas!”
  - Hardening software installation

Direct Object Access

- OWASP/WASC Coverage
- Dynamic Loading Mechanisms
  - Race Conditions
Spoofing and Redirects

- Direct Object References

**Defending XML Processing**

Defending XML

- Understanding common attacks and how to defend
- Operating in safe mode
- Using standards-based security
- XML-aware security infrastructure
Defending Ajax

- Ajax Security Exposures
- Attack Surface Changes
- Injection Threats And Concerns
- Bridging and Potential Problems
- Managing Bridges
- Effective Defenses And Practices