Course Outline

Secure .Net Web Application Development

Course Summary

Description

Securing .Net Web Applications is a lab-intensive, hands-on .Net security training course, essential for experienced enterprise developers who need to engineer, maintain, and support secure .Net-based web applications. In this course, students thoroughly examine best practices for defensively coding web applications, covering all the OWASP Top Ten as well as several additional prominent vulnerabilities (such as file uploads, CSRF and direct object references). Students will repeatedly attack and then defend various assets associated with fully functional web applications and services. This hands-on approach drives home the mechanics of how to secure .Net web applications in the most practical of terms.

A key component to our ***Best Defense IT Security Training Series***, this workshop is a companion course with several developer-oriented courses and seminars. Our bug hunting class introduces penetration testing , illustrating how hackers probe and exploit our applications. Our developing secure software class introduces various security measures that can be applied through the software lifecycle. The combination of ethical hacking, secure coding, and secure lifecycle training provides student with the complete experience in application security. This course merges these classes with a specific .Net orientation.

NOTE: Although this edition of the course is .Net-specific, it may also be presented using Java or other programming languages.

Objectives

After taking this course, students will be able to:

- Understand the concepts and terminology behind defensive, secure coding including the phases and goals of a typical exploit
- Develop an appreciation for the need and value of a multilayered defense in depth
- 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
- 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
- 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
- Detect, attack, and implement defenses against XSS and Injection attacks
- Understand the risks associated with XML processing, file uploads, and server-side interpreters and how to best eliminate or mitigate those risks
- Learn the strengths, limitations, and use for tools such as code scanners, dynamic scanners, and web application firewalls (WAFs)
- Understand techniques and measures that can used to harden web and application servers as well as other components in your infrastructure

Course Outline

Secure .Net Web Application Development

Course Summary (cont'd)

Topics

- Securing Applications Foundation
- Bug Stomping 101
- Bug Stomping 102
- Moving Forward

Audience

This course is designed for experienced .Net developers.

Prerequisites

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. 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 development practical experience.

Duration

Four days

Course Outline

Secure .Net Web Application Development

Course Outline

I. Securing Applications Foundation
- A. Removing Bugs
  1. Open Web Application Security Project (OWASP)
  2. OWASP Top Ten Overview
  3. Web Application Security Consortium
  4. CERT Secure Coding Standards
  5. Bug Hunting Mistakes to Avoid
  6. Tools and Resource
- 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
  7. Tutorial: Working with Visual Studio
  8. Lab: Case Study Setup and Review

II. Bug Stomping 101
- A. Unvalidated Data
  1. Buffer Overflows
  2. Integer Arithmetic Vulnerabilities
  3. Unvalidated Data: Crossing Trust Boundaries
  4. Defending Trust Boundaries
  5. Whitelisting vs Blacklisting
  6. Lab: Defending Trust Boundaries
- B. Injection
  1. Injection Flaws
  2. SQL Injection Attacks Evolve
  3. Drill Down on Stored Procedures
  4. Other Forms of Injection
  5. Minimizing Injection Flaws
  6. Lab: Defending Against SQL Injection
- C. Broken Authentication
  1. Quality and Protection of Authentication Data
  2. Handling Passwords on Server Side
  3. SessionID Risk Reduction
  4. HttpOnly and Security Headers
  5. Lab: Defending Authentication
- D. Sensitive Data Exposure
  1. Protecting Data Can Mitigate Impact
  2. In-Memory Data Handling
  3. Secure Pipes
  4. Failures in TLS/SSL Framework
  5. Lab: Defending Sensitive Data
- E. XML External Entities (XXE)
  1. XML Parser Coercion
  2. XML Attacks: Structure
  3. XML Attacks: Injection
  4. Safe XML Processing
  5. Lab: Safe XML Processing
  6. Lab: Dynamic Loading Using XSLT (Optional)
- F. Broken Access Control
  1. Access Control Issues
  2. Excessive Privileges
  3. Insufficient Flow Control
  4. Unprotected URL/Resource Access
  5. Examples of Shabby Access Control
  6. Sessions and Session Management
  7. Lab: Unsafe Direct Object References
  8. Lab: Spotlight on Verizon Exploit

III. Bug Stomping 102
- A. Security Misconfiguration
  1. System Hardening: IA Mitigation
  2. Application Whitelisting
  3. Least Privileges
  4. Anti-Exploitation
  5. Secure Baseline
- B. Cross Site Scripting (XSS)
  1. XSS Patterns
  2. Persistent XSS
  3. Reflective XSS
  4. DOM-based XSS
  5. Best Practices for Untrusted Data
  6. Lab: Defending Against XSS
- C. Deserialization/Vulnerable Components
  1. Deserialization Issues
  2. Identifying Serialization and Deserializations
  3. Vulnerable Components
  4. Software Inventory
  5. Managing Updates

Lab: Spotlight on Equifax Exploit

Course Outline

Secure .Net Web Application Development

Course Outline (cont'd)

D. Insufficient Logging and Monitoring
   1. Fingerprinting a Web Site
   2. Error-Handling Issues
   3. Logging In Support of Forensics
   4. Solving DLP Challenges
   5. Lab: Error Handling

E. Spoofing, CSRF, and Redirects
   1. Name Resolution Vulnerabilities
   2. Fake Certs and Mobile Apps
   3. Targeted Spoofing Attacks
   4. Cross Site Request Forgeries (CSRF)
   5. CSRF Defenses
   6. Lab: Cross-Site Request Forgeries

IV. Moving Forward

A. What Next?
   1. Common Vulnerabilities and Exposures
   2. CWE/SANS Top 25 Most Dangerous SW Errors
   3. Strength Training: Project Teams/Developers
   4. Strength Training: IT Organizations
   5. Lab: Spotlight on Capitol One Exploit

B. NET Issues and Best Practices
   1. Manage Code and Buffer Overflows
   2. .Net Permissions
   3. ActiveX Controls
   4. Proper Exception Handling
   5. Lab: Securing the Business Layer
   6. Web Service Attacks

Time Permitting:

C. Cryptography Overview
   1. Strong Encryption
   2. Message Digests
   3. Encryption/Decryption
   4. Keys and Key Management
   5. NIST Recommendations

D. .NET Cryptographic Services
   1. The role of cryptographic services
   2. Hash algorithms and hash codes
   3. Encrypting data symmetrically
   4. Encrypting data asymmetrically
   5. Lab: .Net Hashing (Optional)
   6. Lab: .Net Symmetric Encryption
   7. Lab: .Net Asymmetric Encryption (Optional)