www.appsec-labs.com

ASP.NET MVC Secure Coding 4-Day hands on Course

Course Syllabus

ASP.NET MVC Secure Coding 4-Day hands on Course

Course description

Secure programming is the best defense against hackers. This multilayered Hands on course will demonstrate live real time hacking methods , analyze the code deficiency that enabled the attack and most importantly teach how to prevent such vulnerabilities by adopting secure coding best practices in order to bullet-proof your ASP.NET MVC application. The methodology of the Cycle of knowledge is as follows: Understand, Identify, Prevent. This methodology presents the student with analytical tools to keep a deeper understanding of coding vulnerabilities and implement security countermeasures in different areas of the software development lifecycle. The hands on labs will enable the student to get a firsthand experience of the Hackers world and what could be done to stop him. Using sound programming techniques and best practices shown in this course, you will be able to produce high-quality code that stands up to attack. The course covers major security principles in the .NET framework, programming vulnerabilities, and specific security issues in ASP.NET MVC 5 web applications, .NET 4.5 client-side applications and WebAPI Web Services.

Target audience

Members of the software development team:

- .NET developers in ASP.NET MVC / WS backend developers
- Designers & architects

Prerequisites

Before attending this course, students should be familiar with:

- Basic knowledge of the .NET framework
- IIS, Databases (SQL Server) & SQL language

Course topics

Day 1

Introduction to application security

- Why web application risks occur?
- How is application security different from network security?
- Web application exploits & vulnerabilities
- OWASP Top 10
- Live hacking examples

.NET authentication

- Authentication scenarios
- Weak Passwords
- Implementing Forms authentication
- Mitigating Brute Force attacks
- The CAPTCHA mechanism
- Implementing Windows authentication
- Relationship between IIS and ASP.NET.
- External Authentication Scenarios
- Katana (OWIN)
- ASP.NET Identity
- ASP.NET MVC 5 Authentication
- Impersonation
- Delegation
- **Lab - Implementing authentication in .NET applications by using a variety of methods.**

.NET authorization

- Authorization models
- URL authorization
- File authorization
- Role based access control (RBAC)
- Using Least Privileged DB User Accounts
- Working with Identities
- Claim Bases Authorization
- Role Based Access Control (RBAC)
- Role manager
- MVC 5 New Protection Approach
- MVC 5 Authorization filters
- **Lab - Implementing authorization in .NET applications by using a variety of methods**

Day 2

Performing Input Validation

- Injection Flaws
- OS Command Injection
- Preventing SQL Injection
- Preventing SQL Injection with nHibernate and EntityFramework
- Using Parameterized queries to prevent SQL Injection
- Stored procedures
- Preventing XPATH Injection
- Mitigating LDAP Injection
- Using Strong typing
- Blacklist VS. Whitelist validation
- Regular expressions (Regex)
- Model Validation for Web API services
- MVC 5 Mass Assignment Vulnerability
- **LAB – handling malicious input and performing secure input validation**

Output Encoding

- Preventing HTML Injection
- Understanding Cross Site Scripting (XSS) attacks
- MVC 5 Html Encoding
- The Anti-XSS built-in functionality
- ASP.NET MVC 5 Request Validator
- **LAB – output encoding**

Browser Manipulation

- Cross Site Request Forgery (CSRF)
- Anti CSRF token
- Preventing CSRF attack for MVC/Web API controllers
- CSRF Protection for XHR
- The dangers of open redirect mechanisms
- Index based redirection
- **LAB: preventing browser manipulation attacks**

Day 3

File Handling

- Path traversal attacks
- Canonicalization
- Virtual path mapping using MapPath
- Sanitizing file names using GetFullPath
- Uploaded files backdoors
- File extension handling
- Directory listing
- **LAB – secure file handling**

Cryptography - Data Confidentiality & Integrity

- Introduction to cryptography
- Avoiding weak “encryption”
- Implementing Encryption using the System.Security.Cryptography namespace
- Symmetric Encryption
- Asymmetric Encryption
- Hashing
- Digital signatures
- Certificates
- The certificate store
- Transport Level Encryption
- Storage Level Encryption
- DB Encryption
- Protecting sensitive strings with SecureString
- Key derivation
- Password Vault
- Using DPAPI (Data Protection API)
- **Lab- Implementing cryptography in .NET applications.**

Application Denial of Service vulnerabilities

- Application / OS crash
- CPU starvation
- Memory starvation
- File system starvation
- Resource starvation
- Triggering high network bandwidth
- User level DOS
- Exploiting a specific vulnerability to cause DoS
- **Lab – Preventing DoS attacks**

Day 4

Secure Session & Cookie Management

- Session management techniques
- Session State Options in .NET
- Avoiding session hijacking
- Cookie based session management
- Cookie information leakage
- Secure Cookie Attributes - Expire, Secure, HttpOnly, Domain, Path
- Attack Scenarios on session management
- Referrer based decisions
- Mitigating CSRF (Cross Site Request forgery)
- ViewState integrity validation
- Preventing ViewState reply attacks
- ViewState changes in ASP.NET 4.5
- Session management common vulnerabilities
- Session management for MVC/Web API controllers
- Cryptography Changes in Session Management in MVC 5
- **LAB – session & cookie management**

.NET Secure error handling

- Why exposing detailed error messages is bad
- Structured Exception Handling – Try, Catch, Finally
- The Fail-Open VS. Fail-Close approach
- Configuring ASP.NET error handling in web.config
- MVC filters and attributes
- Creating custom error pages
- HTTP error codes
- Handling errors using HttpModule
- Page level VS. Application level handling
- Handling Runtime Security Errors
- Error handling strategies
- **Lab- how to securely handle runtime errors using the .NET framework and Windows mechanisms**.

.NET auditing & logging

- Importance of Logging
- What should we audit?
- Event message structure
- Logging best practices
- Built-in logging technologies in .NET MVC
- ASP.NET trace and System.Diagnostics.Trace

- Windows event log
- Performance Monitor
- Windows Management Instrumentation (WMI)
- The logn4net framework
- **Lab- Implementing auditing in .NET applications by using a variety of methods**

.NET configuration management

- Secure connection to remove services
- Protecting connections strings
- Disable debugging
- Disable tracing
- **Lab – secure configuration management**