Faysal Aziz

Next-Gen Security for .NET:

5 days of Training and Migration

FA-0339 · 5 days

Software Development

Introduction

Welcome to the "Application Security Training" course! This five-day program is designed to provide you with a comprehensive understanding of application security, including the necessary prerequisites for effective learning and a section on migrating from .NET 4.5 to modern .NET. By the end of this course, you will be equipped with the knowledge and skills to secure applications effectively using the latest .NET technologies.

Learning outcomes

By the end of this course, you will be able to:

- Understand the fundamentals of C# and .NET.
- Implement authentication and authorization in .NET applications.
- Manage sessions and access control to prevent security vulnerabilities.
- Securely handle input and output data to prevent common attacks.
- Protect sensitive data using encryption and secure data management practices.
- Implement secure error handling and logging.
- Manage vulnerable dependencies and protect business logic.
- Handle untrusted files securely.
- Harden application configuration.
- Understand ISO27001 requirements and API security.
- Migrate from .NET 4.5 to modern .NET versions.

Prerequisites

Before attending this course, you should have:

- Basic understanding of programming concepts.
- Familiarity with basic web development.
- A laptop with internet connectivity and the ability to install software.

Training outline

1. C# Recap:

1. Overview of C# syntax and features.
2. Object-oriented programming principles in C#.
3. Key libraries and frameworks in the .NET ecosystem.

2. Migrating from .NET 4.5 to Modern .NET:

1. Challenges of using obsolete versions.
2. Step-by-step migration process.
3. Key features and improvements in modern .NET versions (e.g., .NET Core, .NET 5/6/7).

3. Dotnet Authentication and Authorization:

1. Implementing authentication through the login page.
2. Overview of .NET Core identity.
3. Role-based authorization.
4. Authentication and authorization in .NET Core Razor pages.

4. Session Management:

1. Importance of session management.
2. Techniques for managing sessions in .NET applications.
3. Best practices for secure session management.

5. Access Control:

1. Fundamentals of access control.

2. Preventing Cross-site Request Forgery (CSRF) attacks.
3. Implementing access control mechanisms in .NET.

6. Securely Handling Input and Output Data:

1. Input validation techniques.
2. Encoding output data to prevent XSS attacks.
3. Preventing SQL injection attacks with Entity Framework Core and ADO.NET.
4. Preventing operating system command injection.
5. Demo: Invoking encoders in code.

7. Protecting Sensitive Data:

1. Encrypting and decrypting sensitive data.
2. Secrets management in .NET applications.
3. Protecting data in the browser.
4. Sending sensitive data in HTTP request bodies.
5. Preventing data from being cached in the browser.
6. Data privacy best practices.
7. Implementing export and deletion of personal data.
8. Protecting data in transit.
9. Demo: Enforce use of HTTPS protocol.
10. Secure TLS configuration for outbound network connections.

8. Secure Error Handling and Logging:

1. ASP.NET Core logging and exception handling.
2. Redacting sensitive information before logging.
3. Logging relevant security events.
4. Preventing the leaking of sensitive information through error pages.

9. Managing Vulnerable Dependencies:

1. Understanding risks from dependencies.
2. Managing NuGet packages securely.
3. Tools and techniques for managing vulnerable dependencies.

10. Protecting the Business Logic:

1. Identifying business logic flaws.
2. Techniques for protecting business logic integrity.

11. Handling Untrusted Files:

1. Securely handling files in ASP.NET.
2. Preventing large file uploads.
3. Scanning uploaded files for viruses and malware.
4. Securely storing uploaded files.
5. Preventing server-side request forgery (SSRF) attacks.

12. Hardening Configuration:

1. Disabling debugging facilities.
2. Hiding system component information.
3. Validating HTTP request headers.
4. Setting HTTP security headers in responses.

13. ISO27001 Requirements:

1. Overview of ISO27001 requirements.
2. Implementing ISO27001 in .NET applications.
3. Best practices for compliance.

14. API Security:

1. Securing APIs with SSL.
2. Using service credentials to authenticate to the API.
3. Using the HTTP Basic Authorization method.

This training course is designed to be hands-on and interactive, ensuring that you not only learn theoretical concepts but also apply them in practical scenarios. By the end of these five days, you will be proficient in securing applications using the latest .NET technologies and best practices.

© 2026 Faysal Aziz · Kuala Lumpur