ASP.NET Core 6 Secure Coding

Duration: 56 Hours

Day 1

Secure Coding with OWASP Resources

* Introduction to OWASP
* OWASP ASVS
* OWASP ASVS: Architecture

Input validation

1. [Enabling whitelist validation using validation attributes](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-24)
2. [Whitelist validation using the FluentValidation library](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-28)
3. [Syntactic and semantic validation](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-34)
4. [Input sanitization](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-39)
5. [Input sanitization using the HTMLSanitizer library](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-43)
6. [Output encoding](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-47)
7. [Output encoding using HtmlEncoder](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-48)
8. [Output encoding using UrlEncoder](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-52)
9. [Output encoding using JavascriptEncoder](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-56)
10. [Protecting sensitive data using the Data Protection API](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_01_Final_NM_ePub.xhtml#_idParaDest-60)

Injection Flaws

1. [What is SQL injection?](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_02_Final_NM_ePub.xhtml#_idParaDest-67)
2. [Fixing SQL injection with Entity Framework](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_02_Final_NM_ePub.xhtml#_idParaDest-68)
3. [Fixing SQL injection in ADO.NET](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_02_Final_NM_ePub.xhtml#_idParaDest-73)
4. [Fixing NoSQL injection](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_02_Final_NM_ePub.xhtml#_idParaDest-78)
5. [Fixing command injection](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_02_Final_NM_ePub.xhtml#_idParaDest-83)
6. [Fixing LDAP injection](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_02_Final_NM_ePub.xhtml#_idParaDest-88)
7. [Fixing XPath injection](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_02_Final_NM_ePub.xhtml#_idParaDest-92)

Broken Authentication

1. [Fixing the incorrect restrictions of excessive authentication attempts](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_03_Final_NM_ePub.xhtml#_idParaDest-99)
2. [Fixing insufficiently protected credentials](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_03_Final_NM_ePub.xhtml#_idParaDest-104)
3. [Fixing user enumeration](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_03_Final_NM_ePub.xhtml#_idParaDest-108)
4. [Fixing weak password requirements](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_03_Final_NM_ePub.xhtml#_idParaDest-112)
5. [Fixing insufficient session expiration](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_03_Final_NM_ePub.xhtml#_idParaDest-116)

Day 2

Sensitive Data Exposure

1. [Fixing insufficient protection of data in transit](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_04_Final_NM_ePub.xhtml#_idParaDest-122)
2. [Fix missing HSTS headers](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_04_Final_NM_ePub.xhtml#_idParaDest-126)
3. [Fixing weak protocols](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_04_Final_NM_ePub.xhtml#_idParaDest-131)
4. [Fixing hardcoded cryptographic keys](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_04_Final_NM_ePub.xhtml#_idParaDest-135)
5. [Disabling caching for critical web pages](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_04_Final_NM_ePub.xhtml#_idParaDest-140)

XML External Entities

1. [Technical requirements](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_05_Final_NM_ePub.xhtml#_idParaDest-145)
2. [Enabling XML validation](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_05_Final_NM_ePub.xhtml#_idParaDest-146)
3. [Fixing XXE injection with XmlDocument](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_05_Final_NM_ePub.xhtml#_idParaDest-151)
4. [Fixing XXE injection with XmlTextReader](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_05_Final_NM_ePub.xhtml#_idParaDest-156)
5. [Fixing XXE injection with LINQ to XML](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_05_Final_NM_ePub.xhtml#_idParaDest-160)

Broken Access Control

1. [Fixing IDOR](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_06_Final_NM_ePub.xhtml#_idParaDest-166)
2. [Fixing improper authorization](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_06_Final_NM_ePub.xhtml#_idParaDest-170)
3. [Fixing missing access control](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_06_Final_NM_ePub.xhtml#_idParaDest-175)
4. [Fixing open redirect vulnerabilities](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_06_Final_NM_ePub.xhtml#_idParaDest-179)

Security Misconfiguration

1. [Disabling debugging features in non-development environments](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_07_Final_NM_ePub.xhtml#_idParaDest-185)
2. [Fixing disabled security features](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_07_Final_NM_ePub.xhtml#_idParaDest-189)
3. [Disabling unnecessary features](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_07_Final_NM_ePub.xhtml#_idParaDest-193)
4. [Fixing information exposure through an error message](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_07_Final_NM_ePub.xhtml#_idParaDest-197)
5. [Fixing information exposure through insecure cookies](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_07_Final_NM_ePub.xhtml#_idParaDest-201)

Day 3

Cross-Site Scripting

1. [Fixing reflected XSS](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_08_Final_NM_ePub.xhtml#_idParaDest-207)
2. [Fixing stored/persistent XSS](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_08_Final_NM_ePub.xhtml#_idParaDest-211)
3. [Fixing DOM XSS](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_08_Final_NM_ePub.xhtml#_idParaDest-215)

Insecure Deserialization

1. [Fixing unsafe deserialization](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_09_Final_NM_ePub.xhtml#_idParaDest-221)
2. [Fixing the use of insecure deserializers](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_09_Final_NM_ePub.xhtml#_idParaDest-227)
3. [Fixing untrusted data deserialization](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_09_Final_NM_ePub.xhtml#_idParaDest-232)

Using Components with Known Vulnerabilities

1. [Fixing the use of a vulnerable third-party JavaScript library](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_10_Final_NM_ePub.xhtml#_idParaDest-239)
2. [Fixing the use of a vulnerable NuGet package](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_10_Final_NM_ePub.xhtml#_idParaDest-246)
3. [Fixing the use of a library hosted from an untrusted source](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_10_Final_NM_ePub.xhtml#_idParaDest-251)

Insufficient Logging and Monitoring

1. [Fixing insufficient logging of exceptions](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_11_Final_NM_ePub.xhtml#_idParaDest-258)
2. [Fixing insufficient logging of DB transactions](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_11_Final_NM_ePub.xhtml#_idParaDest-262)
3. [Fixing excessive information logging](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_11_Final_NM_ePub.xhtml#_idParaDest-265)
4. [Fixing a lack of security monitoring](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_11_Final_NM_ePub.xhtml#_idParaDest-268)

Miscellaneous Vulnerabilities

1. [Fixing the disabled anti-Cross-Site Request Forgery protection](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_12_Final_NM_ePub.xhtml#_idParaDest-274)
2. [Preventing Server-Side Request Forgery](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_12_Final_NM_ePub.xhtml#_idParaDest-279)
3. [Preventing log injection](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_12_Final_NM_ePub.xhtml#_idParaDest-284)
4. [Preventing HTTP response splitting](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_12_Final_NM_ePub.xhtml#_idParaDest-289)
5. [Preventing clickjacking](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_12_Final_NM_ePub.xhtml#_idParaDest-294)
6. [Fixing insufficient randomness](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_12_Final_NM_ePub.xhtml#_idParaDest-299)

Best Practices

1. [Proper exception handling](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_13_Final_NM_ePub.xhtml#_idParaDest-306)
2. [Using security-related cookie attributes](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_13_Final_NM_ePub.xhtml#_idParaDest-311)
3. [Using a Content Security Policy](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_13_Final_NM_ePub.xhtml#_idParaDest-315)
4. [Fixing leftover debug code](https://learning.oreilly.com/library/view/aspnet-core-5/9781801071567/B17201_13_Final_NM_ePub.xhtml#_idParaDest-320)

Day 4

Authentication

* ASP.NET Core Identity
* Demo: Implement User Authentication
* Password Hashing
* Demo: Password Strength Policy
* Demo: Sign-up and Account Confirmation
* Demo: Password Reset
* Demo: Two-factor Authentication
* Demo: Rate Limiting Failed Logins

Session Management

* ASP.NET Core Sessions
* Demo: Cookie-based Session Parameters
* Demo: Step-up Authentication
* Demo: Logout

Day 5

Access Control

* ASP.NET Core Authorization
* Demo: Role-based Access Control
* Demo: Claims-based Access Control
* Demo: Rendering UI Based on Level of Access
* Preventing Cross-site Request Forgery (CSRF) Attacks

Securely Handling Input and Output Data

* Input Validation
* Demo: Input Validation
* Demo: Redirect URL Validation
* Encoding Output Data
* Demo: Invoking Encoders in Code
* Injection Attacks
* Demo: Prevent SQL Injection Attacks with Entity Framework Core and ADO.NET
* Demo: Prevent Operating System (OS) Command Injection
* Insecure Deserialization
* Demo: Prevent XML External Entity (XXE) Attack

Protecting Sensitive Data

* ASP.NET Core Data Protection
* Demo: Encrypt and Decrypt Sensitive Data
* Demo: Secrets Management
* Protecting Data in the Browser
* Demo: Send Sensitive Data in HTTP Request Body
* Demo: Prevent Data from Being Cached in the Browser
* Data Privacy
* Demo: Implement Export and Deletion of Personal Data
* Protecting Data in Transit
* Demo: Enforce Use of HTTPS Protocol
* Demo: Secure TLS Configuration for Outbound Network Connections

Day 6

Secure Error Handling and Logging

* ASP.NET Core Logging and Exception Handling
* Demo: Redact Sensitive Information Before Logging
* Demo: Log Relevant Security Events
* Demo: Prevent Leaking Sensitive Information through Error Pages

Managing Vulnerable Dependencies

* Risks from Dependencies
* Demo: NuGet Packages
* Demo: Find Vulnerable and Deprecated Dependencies
* Demo: Generate Sub-Resource Integrity (SRI) Hashes for JavaScript

Protecting the Business Logic

* Business Logic Flaws
* Demo: Protecting Business Logic Integrity
* Demo: Protecting Business Logic Flows
* Demo: Implement Rate Limiting

Handling Untrusted Files

* Handling Files in ASP.NET Core
* Demo: Prevent Large File Upload
* Demo: Scan Uploaded Files for Viruses and Malware
* Demo: Securely Store Uploaded Files
* Preventing Server-side Request Forgery (SSRF) Attacks
* Demo: Prevent SSRF Attacks

Day 7

Hardening Configuration

* Hardening ASP.NET Core Applications
* Demo: Disable Debugging Facilities
* Demo: Hide System Component Information
* Demo: Validate HTTP Request Headers
* Demo: Set HTTP Security Headers in Responses

API Security

* Web Application Security Versus API Security
* Demo: Using Bearer Tokens
* Demo: Implementing CORS