NET302 - DEFENDING .NET

Course Learning Objectives

Defending .NET is designed for experienced C# and Web Application Developers and provides the latest security guidance for .NET. This course covers five areas of .NET security, beginning with identifying and addressing common vulnerabilities, extending the ASP.NET identity system, protecting stored data, and protecting transient data. The second part covers protecting and hardening services, and authenticating safely with a Blazor client. The third part explores logging in ASP.NET, important security events to look out for, and defining log review processes. The fourth part then discusses defending against attacks by setting password policies, preventing users from reusing passwords, and stopping phishing attacks.inally, the last part covers logging in ASP.NET, important security events to look out for, and defining log review processes.

Description

By the end of this course, learners will be able to identify common vulnerabilities in .NET web applications, and learn how to address them, defend against attacks that target the people who use your web applications, use logging effectively in an ASP.NET web application, protect services you build with ASP.NET Web API, and ensure that the sensitive information you handle during development stays secret.

Audience

C# Developers

Web Application Developers

Time Required

Tailored learning - 75 minutes approx.

NET302 - DEFENDING .NET

Course Outline

1. Best Practices for Security

- The big picture
- The ASP.NET identity system
- Using a custom DbContext
- Using a custom IdentityUser
- Using a custom data store
- Avoid writing custom auth. code
- Subtle weaknesses
- Additional features
- Extensibility
- Manual encryption of stored data
- Key management
- Performance
- New failure point
- Use built-in encryption features
- Use SQL server always encrypted
- Protecting transient data
- The data protection APIs
- Security is hard

2. Securing ASP.NET Web APIs and Blazor

- Understanding Web API security
- API methods are endpoints
- Broken access control
- Solution
- Hardening methods
- Denial-of-service attacks
- Rate limiting
- Hardening methods against DoS
- Web API authentication
- Token-based authentication
- Who issues the tokens?
- The authentication server
- The advantage of a third-party service
- Using IdentityServer
- An example with Blazor
- Testing the Blazor client with token-based authentication

3. Safely Handling Developer Secrets

- What are developer secrets?
- Key stealing on GitHub
- Secrets in a configuration file
- Accidental checkins and .gitignore
- Using the Secret Manager
- Enabling the Secret Manager
- Storing a secret
- Visual Studio support
- Where are secrets stored?
- The Secret Manager in a production environment
- Migrate to a cloud-based secret store
- Storing secrets in Azure Key Vault
- Best practices for secrets management

4. Keeping Users Secure

- What are user-focused attacks?
- Arbitrary password policies
- Password length
- Implementing passphrases
- Allowing pwned passwords
- Have I Been Pwned
- Testing a password for pwnage
- Testing pwned passwords with a library
- Phishing attacks
- Two-factor authentication (2FA)
- Enabling 2FA for a user
- 2FA registration in ASP.NET
- ASP.NET classes
- QR code registration
- QR codes in ASP.NET
- Requiring 2FA
- FIDO2
- FIDO2 in ASP.NET

5. Detecting Security Issues with Logging

- Security and the SDLC
- Security exploits
- Logging with Serilog
- Configuring Serilog
- Additional Serilog features
- Identifying important security events
- Log reviews
- Best practices for logging

SecurityCompass