# .NET Secure Coding for Client-Server Applications

4-Day hands on Course

Course Syllabus

**AppSec Labs Ltd.**
info@appsec-labs.com | https://appsec-labs.com | 10 HaTa'as St., Kfar Saba 44641 Israel | T: +972-9-7485005 | F: +972-9-7730595

# .NET Secure Coding for Client-Server Applications 4-Day hands on Course

## Course description

Secure programming is the last line of defense against attacks targeted toward our systems. This course shows you how to identify security flaws & implement security countermeasures in different areas of the software development lifecycle and apply these skills to improve the overall quality of the products and applications.

Using sound programming techniques and best practices shown in this course, you can produce high-quality code that stands up to attack. This course covers major security principles in the .NET framework, programming vulnerabilities and specific security issues in desktop WPF application coupled with WS backend.

The objectives of the course are to acquaint students with security concepts and terminology, and to provide them with a solid foundation for developing software using the best practices in the .NET framework. By course completion, students will be proficient in secure programming and have learnt the basics of security analysis and design. Students should then be able to develop, design and maintain applications using security methods and techniques using the .NET framework advanced security features.

## Target audience

Members of the software development team:
- .NET Client Side and Server Side developers
- Designers & architects.

## Prerequisites

Before attending this course, students should be familiar with:
- .NET background using C# (preferred) or VB.NET
- IIS, Databases (SQL Server) & SQL language

## Course topics

Day 1:

### Introduction to Application Security

- Security should be considered during the entire application lifecycle
- Design Level Attacks
- Implementation Level Attacks
- Deployment Level Attacks
- Handling Risks
- Security assumptions
- Thinking like attackers
- Structural security
- Segmentation
- Layered security
- Stand-alone security
- Principle of least privilege
- Fail securely
- Transparency, ease of use
- Secure input validation
- Sanitize Application Output
- Data protection
- OS security mechanisms

### .NET Authentication

- Authentication scenarios and protocols
- Password based authentication
- Protecting user accounts passwords
- Implementing Windows authentication
- WS authentication scenarios
- Relationship between IIS and ASP.NET.
- Impersonation
- Delegation
- **Lab - Implementing authentication in .NET applications by using a variety of methods**

## .NET Authorization

- Introduction to authorization models
- Role Based Access Control (RBAC)
- OS ACL authorization
- WS authorization scenarios
- Membership provider & role manager
- Least privileged approach
- **Lab - Implementing authorization in .NET applications by using a variety of methods**

Day 2:

## .NET Input Validation

- What is considered Input?
- The need for input validation
- Client side VS. server side validations
- Black list VS. white list validation
- String manipulation and comparison
- Data type conversion
- Regular expressions
- WCF input validation
- **Lab - Validating user input in .NET applications**

## .NET Secure File Handling

- Path traversal
- Canonicalization problem
- Virtual path mapping using MapPath
- Sanitizing file names using GetFullPath
- Uploaded files backdoors
- File extension handling
- Isolated storage
- File ACLs
- **Lab – secure file handling**

## .NET Cryptography

- Introduction to cryptography
- Avoiding weak “encryption”
- 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.**

Day 3:

## Transport, Web Services & WCF Security

- REST & SOAP overview
- Security and web services
- Common WS threats and vulnerabilities
- WCF security mechanisms
- Sensitive information transmission
- Transport level security
- Message level security
- Validating certificates and avoiding man-in-the-middle
- SSL Pinning
- **Lab - Implementing security in WCF web services**

## Application Denial of Service Vulnerabilities

- Application / OS crash
- CPU starvation
- Memory starvation
- File system starvation
- Resource starvation
- Resource locking
- Triggering high network bandwidth
- User level DoS

- Exploiting a specific vulnerability to cause DoS
- **Lab – Preventing DoS attacks**

## .NET Secure Configuration Management

- Securing back-end communications
- Protecting connections strings
- Disable debugging
- Disable tracing
- Protecting server runtime environment
- Implementing least privileged approach for DB
- Secure compilation
- Secure deployment
- **Lab – secure configuration management**

Day 4:

## .NET Error Handling

- Why exposing detailed error messages is bad
- Structured Exception Handling – Try, Catch, Finally
- The Fail-Open VS. Fail-Close approach
- Configuring error handling in web.config
- Creating custom error pages
- HTTP error codes
- Level of application error handling
- Handling Runtime Security Errors
- Error handling strategies
- WCF error handling
- **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
- WCF logging options
- Windows event log

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

## EXE Reverse Engineering and code protection

- The problem of reversing & decompilation
- Assume attackers have source code
- Introduction to MSIL & the CLR
- Debugging
- Patching
- Unpacking
- Obfuscation
- Avoiding hard coded secrets
- Secure serialization
- **Lab**