# Web Security – Microsoft .Net Based – Beginner Course

This course provides students with the knowledge and skills that are needed to build Web applications by using security-enhanced coding techniques. Students will learn how to identify Web application security vulnerabilities and understand the trade-offs between functionality and performance when choosing the appropriate security mechanisms for their Web applications. Throughout this course, students will get hands-on experience in creating security-enhanced Web applications. Students will be able to:

- Define the basic principals of, and motivations for, Web security.
- Perform a threat analysis of Web-accessible assets.
- Use knowledge of authentication, Security Identifiers (SIDs), Access Control Lists (ACLs), impersonation, and the concept of running with least privilege to help ensure access to only those system resources that are necessary to accomplish normal request processing.
- Help protect file system data by using the features in Microsoft® Windows® 2000.
- Help protect the portion of a Web application that requires private communications by using Secure Sockets Layer (SSL).
- Use general security coding best practices to help ensure a security-enhanced Web application.
- Employ a structured approach to testing for Web application security.
- Use a systematic approach and knowledge of security best practices to help protect an existing Web application.

---

## Module 1: Introduction to Web Security

This module provides an overview of the terms and concepts of, along with the justification for, Web security.

- Why Build Security-Enhanced Web Applications?
- Using the STRIDE Model to Determine Threats
- Implementing Security: An Overview

After completing this module, students will be able to:

- Describe why security is an essential consideration in Web application development.
- Describe the basic methods of cryptography, hashing, and digital signing.

## Module 2: Planning for Web Application Security

This module describes the general process of incorporating security in the Web application planning and design process.

- A Design Process for Building Security-Enhanced Web Applications

After completing this module, students will be able to:

- Describe the iterative process of designing security into a Web application and be able to describe how each step relates to the other steps.

- Categorize and identify the most common types of attacks, the potential threat that those attacks pose to systems, services, and data within the organization, and the relationship between these threats.

## Module 3: Validating User Input

This module explains the methods that can be used for checking user input, along with a discussion of the consequences of not performing those checks.

- User Input
- Types of User Input Attacks
- Performing Validation
- Revealing as Little Information as Possible to the User

After completing this module, students will be able to:

- Identify the sources of user input in a Web application.
- Describe the security aspects of the client/server Web paradigm.
- Implement user input verification.
- Use communications analysis and coding best practices to avoid providing information to users that can be leveraged for security attacks.
- Use proper error handling to help ensure all fallback paths are expected, wanted, and do not suspend resource allocations.
- Reduce the impact of Denial of Service (DoS) attacks of varying types, such as application crashing, CPU starvation, resource starvation, and bandwidth choking.

## Module 4: Internet Information Services Authentication

The following topics are covered in this module:

- Introduction to Web Client Authentication
- Configuring Access Permission for a Web Server
- Selecting a Security-Enhanced Client Authentication Method
- Running Services As an Authenticated User

After completing this module, students will be able to:

- Describe all of the authentication methods that are supported by IIS and Windows 2000 Server and be able to select the best method for a given set of requirements.
- Use knowledge of Windows 2000 access control mechanisms and process identification to properly configure resource access for the identities that are defined for a Web application.

## Module 5: Securing Web Pages

This module covers security in the context of Web applications that are built by using the .NET framework.

- ASP Forms-Based Authentication
- .NET Code Access and Role-Based Security
- Overview of ASP.NET Authentication Methods
- Working with Windows-Based Authentication in ASP.NET security
- Working with ASP.NET Forms-Based Authentication

Students will be given the task of completing the implementation of an ASP.NET Web application and setting up the authentication and impersonation methods

After completing this module, students will be able to:

- Describe the elements that make up the core security model of the .NET Framework.
- Use security best practices and a complete understanding of the security model while implementing ASP.NET Web applications.

### Module 6: Securing File System Data

This module teaches a Web developer how to help protect file system data that is typically part of a Web application.

- Overview of Securing Files
- Windows Access Control
- Creating ACLs Programmatically
- Helping to Protect ASP.NET Web Application Files

After completing this module, students will be able to:

- Describe how the Windows access control mechanisms are used to help protect file system data.
- Use the features of Windows to help protect Web application data from tampering.
- Use ASP.NET Web.config files to restrict access to files that are located in an ASP.NET Web application.

### Module 7: Helping to Protect Communication Privacy and Data Integrity

This module teaches the mechanisms that can be used to help ensure Web communication privacy and message data integrity, along with the guidelines for their proper use. The guidelines are presented as an attempt to avoid the common implementation mistakes that can compromise security and performance.

- Introduction to Cryptography
- Working with Digital Certificates
- Management
- Using Secure Sockets Layer/Transport Layer Security Protocols
- Using Internet Protocol Security

After completing this module, students will be able to:

Help protect the portions of a Web application that require private communications by using SSL.

## Module 8: Encrypting, Hashing, and Signing Data

This module explains how to use the cryptographic functionality, supported by Microsoft platforms, to encrypt and sign data.

- Encryption and Digital Signing Libraries
- Using CAPICOM
- Using System.Security.Cryptography Namespace to Hash Data
- Using System.Security.Cryptography Namespace to Encrypt and Sign Data

After completing this module, students will be able to:

Use one of the Cryptographic Services classes of the System.Security.Cryptography namespace to transform a block of data to cyphertext.

## Module 10: Testing Web Applications for Security

This module will provide students with the skills and knowledge that are required to properly test a Web implementation for security.

- Testing Security in a Web Application
- Creating a Security Test Plan
- Performing Security Testing

After completing this module, students will be able to:

- Differentiate security testing from other types of testing.
- Create a security test plan.
- Successfully carry out a security test plan.

---

## Audience:

This course is intended for students who are responsible for the design and development of Web applications. These students typically have three to five years of experience in developing or designing distributed Web applications. Actual job role titles vary throughout the technology industry, and they may include, but are not limited to:

Web Developer: The Web developer is responsible for developing the logic, coding, testing, and debugging of Web applications and Web application software.
Solutions Architect: The Solutions Architect is responsible for the design of the technical architecture of Web applications and Web-based software applications

## Prerequisites:

Before attending this course, students must have:

- Familiarity with n-tier application architecture.
- Experience in developing or designing distributed Web applications.