

**C# and Web Application Security**

**Course Number:** SEC-122
**Duration:** 3 days

**Overview**

Accelebrate's C# and Web Application Security training teaches developers how to prevent common security issues in C# applications. Attendees go beyond core programming issues, exploring secure code pitfalls of the C# language and the .NET framework.

**Note:** To ensure ample one-on-one engagement with the instructor, this class is capped at 12 people, overriding Accelebrate’s default cap of 15.

**Prerequisites**

All secure coding students should have general C# and web application development experience.

**Materials**

All attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computer for this class. The class will be conducted in a remote environment that Accelebrate will provide; students will only need a local computer with a web browser and a stable Internet connection. Any recent version of Microsoft Edge, Mozilla Firefox, or Google Chrome will be fine.

**Objectives**

All students will:

* Get familiar with essential cyber security concepts
* Understand Web application security issues
* Gain a detailed analysis of the OWASP Top Ten elements
* Put Web application security in the context of C#
* Go beyond the low hanging fruits
* Manage vulnerabilities in third-party components
* Identify vulnerabilities and their consequences
* Learn the security best practices in C#

**Outline**

* Cyber security basics
	+ What is security?
	+ Threat and risk
	+ Cyber security threat types
	+ Consequences of insecure software
* Introducing the OWASP Top 10
* A1 - Injection
	+ Injection principles
	+ Injection attacks
	+ SQL injection
	+ SQL injection best practices
	+ Code injection
* A2 - Broken Authentication
	+ Authentication
	+ Password management
	+ Session management
* A3 - Sensitive Data Exposure
	+ Information exposure
	+ Exposure through extracted data and aggregation
	+ Case study – Strava data exposure
	+ System information leakage
	+ Information exposure best practices
* A4 - XML External Entities (XXE)
	+ DTD and the entities
	+ Entity expansion
	+ External Entity Attack (XXE)
* A5 - Broken Access Control
	+ Access control basics
	+ Failure to restrict URL access
	+ Confused deputy
	+ File upload
* A7 - Cross-site Scripting (XSS)
	+ Cross-site scripting basics
	+ Cross-site scripting types
	+ Case study – XSS in Fortnite accounts
	+ XSS protection best practices
* A8 - Insecure Deserialization
	+ Serialization and deserialization challenges
	+ Integrity – deserializing untrusted streams
	+ Integrity – deserialization best practices
	+ Property Oriented Programming (POP)
* A9 - Using Components with Known Vulnerabilities
	+ Using vulnerable components
	+ Assessing the environment
	+ Hardening
	+ Untrusted functionality import
	+ Importing JavaScript
	+ Case study – The British Airways data breach
	+ Vulnerability management
* A10 – Server-Side Request Forgery (SSRF)
	+ Server-side Request Forgery (SSRF)
	+ Case study – SSRF and the Capital One breach
* Web Application Security Beyond the Top Ten
	+ Client-side security
	+ Tabnabbing
	+ Frame sandboxing
* Common Software Security Weaknesses
	+ Input validation
	+ Integer handling problems
	+ Unsafe reflection
* Code quality
	+ Code quality and security
	+ Data handling
	+ Object-oriented programming pitfalls
* Conclusion
	+ Secure coding principles
	+ And now what?