

**Comprehensive C# and Web Application Security**

**Course Number:** SEC-128
**Duration:** 5 days

**Overview**

Accelebrate's Comprehensive C# and Web Application Security training takes attendees through the common Web application security issues following the OWASP Top Ten and beyond. This security course is taught in C# and discusses core programming issues, including the security pitfalls of the C# language and the ASP.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**

Students should have solid C# and web application development skills.

**Materials**

All secure coding 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#
* Learn input validation approaches and principles
* Understand how cryptography can support application security
* Learn how to use cryptographic APIs correctly in C#
* Understand security testing methodology and approaches
* Get familiar with common security testing techniques and tools

**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
	+ NoSQL injection
	+ SQL injection best practices
	+ SQL injection protection and ORM
	+ Parameter manipulation
	+ Code injection
	+ Script injection
	+ General injection best practices
	+ Storing account passwords
	+ Password in transit
	+ Dictionary attacks and brute forcing
	+ Salting
	+ Adaptive hash functions for password storage
* A2 - Broken Authentication
	+ Authentication
	+ Password management
	+ Session management
	+ Using tokens
	+ Cookie security
* A3 - Sensitive Data Exposure
	+ Information exposure
	+ Exposure through extracted data and aggregation
	+ Case study – Strava data exposure
	+ Privacy violation
	+ System information leakage
	+ Information leakage through side channels
	+ Information exposure best practices
* A4 - XML External Entities (XXE)
	+ DTD and the entities
	+ Attribute blowup
	+ Entity expansion
	+ External Entity Attack (XXE)
* A5 - Broken Access Control
	+ Access control basics
	+ Failure to restrict URL access
	+ Confused deputy
	+ File upload
* A6 - Security Misconfiguration
	+ Configuration principles
	+ Server misconfiguration
	+ ASP.NET and IIS configuration best practices
	+ AWS configuration best practices
* A7 - Cross-site Scripting (XSS)
	+ Cross-site scripting basics
	+ Cross-site scripting types
	+ 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 - Insufficient Logging & Monitoring
	+ Logging and monitoring principles
	+ Insufficient logging
	+ Case study – Plaintext passwords at Facebook
	+ Logging best practices
	+ Monitoring best practices
* XML Security
	+ XML validation
	+ XML injection
* JSON Security
	+ JSON validation
	+ JSON injection
	+ Dangers of JSONP
	+ JSON/JavaScript hijacking
	+ Best practices
	+ Case study – ReactJS vulnerability in HackerOne
* Web Application Security Beyond the Top Ten
	+ Client-side security
	+ Tabnabbing
	+ Reverse tabnabbing
	+ Frame sandboxing
* API security - Input validation
	+ Integer handling problems
	+ Open redirects and forwards
	+ Files and streams
	+ Unsafe reflection
	+ Unsafe native code
* Time and state
	+ Race conditions
* Errors
	+ Error and exception handling principles
	+ Error handling
	+ Exception handling
* Code quality
	+ Code quality and security
	+ Data handling
	+ Object-oriented programming pitfalls
* Denial of Service
	+ Flooding
	+ Resource exhaustion
	+ Sustained client engagement
	+ Denial of service problems in C#
	+ Infinite loop
	+ Economic Denial of Sustainability (EDoS)
	+ Denial of service
	+ Algorithm complexity issues
* Cryptography for Developers
	+ Cryptography basics
	+ Crypto APIs in C#
* Elementary Algorithms
	+ Random number generation
	+ Hashing
* Confidentiality Protection
	+ Symmetric encryption
	+ Asymmetric encryption
	+ Combining symmetric and asymmetric algorithms
	+ Key exchange and agreement
* Integrity Protection
	+ Authenticity and non-repudiation
	+ Message Authentication Code (MAC)
	+ Digital signature
* Public Key Infrastructure (PKI)
	+ Some further key management challenges
	+ Certificates
* Security testing
	+ Security testing methodology
	+ Security testing techniques and tools
* Conclusion
	+ Secure coding principles
	+ And now what?