

**Java and Web Application Security**

**Course Number:** SEC-124  
**Duration:** 3 days

**Overview**

Accelebrate's Java and Web Application Security course teaches developers common security vulnerabilities in Java applications and how to write hardened, more secure code. Attendees learn the OWASP Top Ten and typical Web vulnerabilities, focusing on how these issues affect Java development. Participants also learn the security pitfalls of the Java language and the runtime environment.

**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 students must have Java and web development experience.

**Materials**

All Secure Coding training attendees receive comprehensive courseware.

**Software Needed on Each Student PC**

Attendees will not need to install any software on their computers 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 work well.

**Objectives**

* Become familiar with essential cyber security concepts
* Understand how cryptography supports security
* Use cryptographic APIs correctly in Java
* Understand web application security issues
* Analyze the OWASP Top Ten elements
* Implement web application security in the context of Java
* Go beyond “low hanging fruit” into more in-depth challenges
* Manage vulnerabilities in third-party components

**Outline**

* Introduction
* Cybersecurity Basics
  + What is security?
  + Threat and risk
  + Cyber security threat types, the CIA triad
  + Consequences of insecure software
* The OWASP Top 10 2021
* A01: Broken Access Control
  + Access control basics
  + Confused deputy
    - Insecure direct object reference (IDOR)
    - Path traversal
    - Path traversal best practices
    - Authorization bypass through user-controlled keys
  + File upload
    - Unrestricted file upload
    - Good practices
  + Open redirects and forwards
    - Unvalidated redirect at Epic Games
    - Best practices
* A02: Cryptographic Failures
  + Information exposure
    - Exposure through extracted data and aggregation
    - Strava data exposure
  + Cryptography for developers
    - Cryptography basics
    - Java Cryptographic Architecture (JCA) in brief
    - Elementary algorithms
    - Confidentiality protection
* A03: Injection
  + Injection principles
  + Injection attacks
  + SQL injection
    - SQL injection basics
    - Lab – SQL injection
    - Attack techniques
    - Content-based blind SQL injection
    - Time-based blind SQL injection
  + SQL injection best practices
    - Input validation
    - Parameterized queries
    - Using prepared statements
    - Additional considerations
    - Hacking Fortnite accounts
  + Code injection
    - OS command injection
  + HTML injection – Cross-site scripting (XSS)
    - Cross-site scripting basics
    - Cross-site scripting types
    - Stored XSS
    - Reflected XSS
    - XSS in Fortnite accounts
    - XSS protection best practices
* A04: Insecure Design
  + The STRIDE model of threats
  + Secure design principles of Saltzer and Schroeder
    - Economy of mechanism
    - Fail-safe defaults
    - Complete mediation
    - Open design
    - Separation of privilege
    - Least privilege
    - Least common mechanism
    - Psychological acceptability
  + Client-side security
    - Frame sandboxing
* A05: Security Misconfiguration
  + Configuration principles
  + XML entities
    - DTD and the entities
    - Entity expansion
    - External Entity Attack (XXE)
* A06: Vulnerable and Outdated Components
  + Using vulnerable components
  + Assessing the environment
  + Hardening
  + Untrusted functionality import
  + Vulnerability management
    - Patch management
    - Vulnerability databases
    - Finding vulnerabilities in third-party components
    - DevOps, the build process and CI / CD
    - Dependency checking in Java
    - Detecting vulnerable components
* A07: Identification and Authentication Failures
  + Authentication
    - Authentication basics
    - Multi-factor authentication
    - Authentication weaknesses
    - PayPal 2FA bypass
  + Password management
    - Inbound password management
* A08: Software and Data Integrity Failures
  + Integrity protection
    - Message Authentication Code (MAC)
    - Digital signature
  + Subresource integrity
    - Importing JavaScript
    - Importing JavaScript
    - The British Airways data breach
  + Insecure deserialization
    - Serialization and deserialization challenges
    - Integrity – deserializing untrusted streams
    - Using readObject
    - Integrity – deserialization best practices
    - Look ahead deserialization
    - Property Oriented Programming (POP)
* A09: Security Logging and Monitoring Failures
  + Logging and monitoring principles
  + Insufficient logging
  + Plaintext passwords at Facebook
  + Logging best practices
* A10: Server-Side Request Forgery (SSRF)
  + Server-side Request Forgery (SSRF)
  + SSRF and the Capital One breach
* Conclusion
  + Secure coding principles
    - Principles of robust programming by Matt Bishop
  + And now what?
    - Software security sources and further reading
    - Java resources