800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

Secure Web Application Engineering: OWASP & CWE in Practice

Applied Technology Academy

A two-day course on recognizing, preventing and responding to common web application weaknesses across the software lifecycle. It uses OWASP Top 10:2025 risk themes and CWE Top 25:2025 weakness patterns as a shared language, but organizes the learning around durable engineering practice rather than a checklist that ages quickly. Lecture, code analysis, demonstrations and case studies show both the vulnerable pattern and the practical defensive alternative across common web frameworks.

LEVEL	DURATION	DELIVERY
Foundational to Intermediate	**2 Days**	**Instructor-led**

Course Overview

- Two days of lecture, code analysis and follow-along demonstrations.
- OWASP Top 10:2025 and CWE Top 25:2025 used to communicate and prioritize risk.
- Examples span C#/ASP.NET Core, Java/Spring and Python/Flask - no single language is required.
- Covers the full arc: secure design, identity, access control, input handling, dependencies,
- configuration, integrity, logging and incident-ready engineering.

Course Outline

- **Day One: Foundations, Design, Identity and Input Security**
- **Module 1. Application Security Context and Ethical Bug Hunting**
  - Security terminology, the exploit lifecycle, remote code execution, web shells and supply-chain attacks.
  - Current breach patterns: vulnerability exploitation, credential misuse, ransomware and the human element.
  - Developer lessons from real incidents, including the MOVEit Transfer compromise.
  - Authorization, scope, privacy, professional conduct and safe testing practice.
- **Module 2. Fingerprinting and Risk Prioritization**
  - How attackers identify platforms, versions, configurations and database technologies.
  - Banner grabbing, injection probing, timing analysis, error analysis and metadata discovery.
  - Information-exposure risk, practical mitigations, and prioritizing with OWASP and CWE.
- **Module 3. Secure Design and Threat Modeling**
  - Security across the development lifecycle; n-tier architecture and framework security features.
  - Threat-modeling workflow and STRIDE analysis.

800.674.3550
Info@AppliedTechAc.com
AppliedTechnologyAcademy.com

  - Attack-surface analysis, trust boundaries, defense in depth, secure defaults and least privilege.
  - Demonstration: threat modeling a representative web application.
- **Module 4. Asset Management, Data Protection and Validation**
  - Asset discovery, data classification and regulatory considerations.
  - Cryptographic failures, hard-coded credentials, TLS configuration and certificate management.
  - Key and secret management; boundary validation and framework input/output validation patterns.
- **Module 5. Authentication and Session Management**
  - Secure authentication patterns, password storage, hashing and account recovery.
  - Multi-factor authentication, TOTP and WebAuthn/FIDO2.
  - Secure cookies, session timeout, rotation and fixation prevention; rate limiting and lockout.
- **Module 6. Access Control and Authorization**
  - Broken, missing and incorrect authorization; role-based and attribute-based patterns.
  - Forceful browsing, exposed administrative interfaces, backup files and test environments.
  - Insecure direct object references across pages, APIs, documents and file paths.
  - Cross-site request forgery flows, framework defenses and privilege-escalation prevention.
- **Module 7. Injection, Input Validation and Output Encoding**
  - SQL injection with parameterized query and ORM defenses; OS command injection.
  - Reflected, stored and DOM-based XSS; contextual encoding and Content Security Policy.
  - Path traversal and server-side request forgery.
  - Positive validation, encodings, rich text, file uploads and well-vetted regular expressions.
- **Day Two: Hardening, Supply Chain, Runtime Defense and Emerging Risk**
- **Module 8. Memory Safety, Resource Management and Error Handling**
  - Out-of-bounds access, buffer weaknesses, integer overflow and null-pointer risk.
  - Resource limits and protection against uncontrolled resource consumption.
  - Exception handling, safe user-facing errors and preventing sensitive-information exposure.
- **Module 9. Security Configuration and Application Hardening**
  - Security misconfiguration and the secure-defaults principle.
  - Framework hardening examples for Spring Boot and Flask; security headers.
  - Container security scanning, resource controls and reviewable configuration management.
- **Module 10. Dependency and Software Supply-Chain Security**
  - Supply-chain failures and the dependency-management lifecycle.
  - Software composition analysis and software bills of materials.
  - CVE, NVD, GitHub Security Advisories, OSV and commercial feeds; patch and rollback paths.
- **Module 11. Software and Data Integrity**
  - Integrity failures in code, updates, plugins, CI/CD pipelines and trusted data flows.

  - Code-injection risk from dynamic evaluation, and safer alternatives.
  - Insecure deserialization, constrained data binding and XML external entity risk.
- **Module 12. Security Logging, Alerting and Runtime Response**
  - Logging and alerting failures and their impact on detection, response and forensics.
  - What to capture: validation and authentication failures, authorization denials, unusual access.
  - Actionable alerts with identity, target, outcome, risk and correlation context.
  - Application-layer response: step-up authentication, session revocation, throttling and lockout.
- **Module 13. Tools, Compliance and Continuous Improvement**
  - Framework-specific security responsibilities and implementation review.
  - Static analysis, dynamic testing, dependency scanning, vulnerability scanning and penetration testing.
  - Using OWASP, CWE, CVE, NIST and vendor advisories to keep practice current.
- **Module 14. Emerging Security Technologies (optional)**
  - Quantum-computing timelines and risk to current public-key cryptography.
  - Post-quantum cryptography, NIST standards, cryptographic inventory and phased migration.
  - AI-assisted scanning, code review and incident response, and the limits of each.

Prerequisites

- No prior application-security, penetration-testing or secure-coding experience is required.
- Basic familiarity with web applications: browser-to-server requests, APIs, databases,
- authentication and sessions.
- Familiarity with at least one server-side language is useful; participants are not expected to
- write production code in class.
- A general understanding of software-development and deployment workflows is recommended.
- No special tooling is required for the standard lecture and demonstration delivery.

Who Should Attend

- Web and software developers, senior developers and technical leads.
- Software and solution architects.
- Application-security engineers, security analysts and security champions.
- QA engineers, test-automation specialists, DevOps, platform and site-reliability teams.
- Engineering managers, product owners and other stakeholders responsible for application risk.

What You'll Learn

By the end of this course, participants will be able to:

- connect attacker behavior to concrete defenses at each trust boundary

- use OWASP Top 10:2025 and CWE Top 25:2025 to communicate and prioritize risk
- run a threat-modeling workflow using STRIDE, attack-surface and data-flow analysis
- classify sensitive data and apply encryption at rest and in transit correctly
- implement secure authentication, MFA, session handling and anti-automation controls
- prevent broken access control, IDOR, forceful browsing and cross-site request forgery
- defend against SQL injection, command injection, XSS, path traversal and SSRF
- harden framework configuration and apply security headers and container controls
- manage dependency and supply-chain risk with SCA, SBOMs and advisory feeds
- design security logging and alerting that supports detection, response and forensics

Hands-On Work

Demonstrations participants can follow:

- threat modeling a representative web application
- asset classification and asset analysis
- weak and strengthened authentication flows
- insecure direct object reference and cross-site request forgery
- SQL injection and cross-site scripting
- dependency and package checking
- XML external entity exploitation and mitigation

Ready to enroll? Call 800.674.3550 or email Info@AppliedTechAc.com — private team cohorts, GSA MAS purchasing, military credentialing funding, VA GI Bill and VR&E and student financing available.