

SECURE CODING TRAINING: DEVELOPMENT AND TECHNICAL LEADERSHIP



GENTEX[®]
TRAINING CENTER



Introduction

Secure Coding Training: Development and Technical Leadership Training Course equips developers and technical leads to build software that resists attack by design. Participants learn how vulnerabilities enter code, how attackers exploit them, and how to prevent them at the source. The program covers the OWASP Top 10, secure design principles, code review practice, and the leadership habits that make security a team norm. As a result, learners leave able to write safer code and guide others to do the same. Furthermore, the course strengthens threat modelling, secure testing, and DevSecOps collaboration.

Secure Coding Training: Development and Technical Leadership Course Objectives

- Identify and remediate the OWASP Top 10 web application vulnerabilities.
- Apply secure design principles such as least privilege, defence in depth, and fail-safe defaults.
- Conduct threat modelling using STRIDE and data flow diagrams.
- Write input validation, output encoding, and parameterized queries to prevent injection.
- Implement safe authentication, session management, and authorization controls.
- Apply cryptographic best practices using approved algorithms and key management.
- Handle secrets, configuration, and sensitive data securely.
- Integrate SAST, DAST, and SCA tools into the development pipeline.
- Perform secure code reviews and interpret CWE and CVSS findings.
- Lead security culture through standards, coaching, and secure SDLC governance.

Course Methodology

- Hands-on labs with vulnerable applications and remediation exercises.

LEARN BOLD. LEAD BEYOND

GENTEX Training Center LLC | Orlando - FL, USA
Info@gentextraining.com



- Threat modelling workshops using STRIDE and data flow diagrams.
- Live code review sessions with real vulnerable code samples.
- Guided practice with SAST, DAST, and dependency scanning tools.
- Capture-the-flag style challenges on common vulnerability classes.
- Facilitator coaching, peer review, and secure coding standards development.

Who Should Take This Course

- Software developers, engineers, and programmers.
- Technical leads, architects, and development managers.
- Application security engineers and penetration testers.
- QA and test engineers involved in security testing.
- DevOps and platform engineers building CI/CD pipelines.
- IT managers responsible for secure development governance.

Secure Coding Training: Development and Technical Leadership Course Outlines

Day 1: Security Foundations and the Attacker Mindset

- Why secure coding matters and the cost of vulnerabilities.
- The CIA triad: confidentiality, integrity, and availability.
- The OWASP Top 10 overview and common attack patterns.
- Understanding the attacker mindset and attack surfaces.
- Secure design principles: least privilege, defence in depth, fail-safe defaults.
- Secure SDLC phases and security gates.
- Case study: analysing a major application breach.





Day 2: Threat Modelling and Secure Design

- Threat modelling process: scope, decompose, identify, mitigate.
- STRIDE methodology: spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege.
- Data flow diagrams and trust boundaries.
- Attack trees and abuse cases.
- Security requirements and misuse stories.
- Designing layered controls and secure architecture patterns.
- Workshop: threat modelling a web application.

Day 3: Preventing Injection and Input Flaws

- Injection vulnerabilities: SQL, NoSQL, command, and LDAP injection.
- Input validation and allow-list strategies.
- Parameterized queries and prepared statements.
- Cross-site scripting (XSS): stored, reflected, and DOM-based.
- Output encoding and content security policy (CSP).
- Cross-site request forgery (CSRF) and same-site protections.
- Lab: exploiting and fixing injection and XSS flaws.

Day 4: Authentication, Access Control, and Data Protection

- Broken authentication and session management risks.
- Password storage with bcrypt, scrypt, and Argon2.
- Multi-factor authentication and identity standards (OAuth 2.0, OIDC).

LEARN BOLD. LEAD BEYOND

GENTEX Training Center LLC | Orlando - FL, USA
Info@gentextraining.com



- Broken access control and insecure direct object references (IDOR).
- Cryptographic best practices and key management.
- Secure handling of secrets, configuration, and sensitive data.
- Lab: hardening authentication and authorization flows.

Day 5: Secure Testing, Tooling, and Technical Leadership

- Static application security testing (SAST) and dynamic testing (DAST).
- Software composition analysis (SCA) and dependency risk.
- Interpreting CWE and CVSS scores and prioritization.
- Secure code review checklists and peer review practice.
- Integrating security into CI/CD (DevSecOps) pipelines.
- Building a secure coding standard and coaching developers.
- Capstone: presenting a secure development improvement plan.

Conclusion

By successfully completing Secure Coding Training: Development and Technical Leadership, participants will gain a practical command of secure coding practice and the leadership habits that sustain it. They will acquire skills in threat modelling, vulnerability remediation, secure authentication and data protection, and security tooling across the pipeline. Gentex Training Center designs the program to help teams ship software that is secure by design, not patched after damage.

FAQs





Q1 What is the "Secure Coding Training: Development and Technical Leadership" course about?

It teaches developers and technical leads how to prevent vulnerabilities at the source rather than fixing them after release. The course covers the OWASP Top 10, threat modelling, safe input handling, secure authentication, security tooling, and the leadership practices that embed security in daily work.

Q2 What are the key benefits of the "Secure Coding Training: Development and Technical Leadership" course?

You gain the ability to write code that resists common attacks and to review others' code with confidence. The course reduces remediation costs, lowers breach risk, improves audit and compliance readiness, and helps technical leads build a security culture instead of relying on last-minute fixes.

Q3 What skills will I gain from the "Secure Coding Training: Development and Technical Leadership" course?

You will develop four core skills: threat modelling and secure design, vulnerability identification and remediation across the OWASP Top 10, secure authentication, access control, and data protection, and security tooling with secure code review and technical leadership.



Q4 What tools, methods, or standards are covered in the "Secure Coding Training: Development and Technical Leadership" course?

The course covers the OWASP Top 10, STRIDE threat modelling, data flow diagrams and trust boundaries, attack trees, parameterized queries, output encoding and CSP, CSRF and same-site protections, bcrypt/scrypt/Argon2 password storage, OAuth 2.0 and OIDC, SAST, DAST, software composition analysis (SCA), CWE and CVSS scoring, and DevSecOps CI/CD integration.

Q5 How is the "Secure Coding Training: Development and Technical Leadership" course applied in real-world practice?

Participants apply it to real work such as threat modelling a new service, fixing injection flaws in an existing codebase, reviewing a pull request for security defects, and adding security gates to a CI/CD pipeline. The same practices support secure design reviews, dependency management, and incident root-cause analysis.