Faysal Aziz

Source Code Security Review

Find real security vulnerabilities in Java and Python code using OWASP-driven, industry-proven review techniques

FA-0415

Software Development · Cybersecurity · Data & Analytics

Introduction

Most critical security failures are not the result of exotic attacks but of ordinary code written under pressure and reviewed without a security mindset. This course is built for professionals who need to **consistently identify security vulnerabilities directly in source code**, before scanners, penetration tests, or attackers do.

Rather than abstract theory or generalized secure design guidance, this training focuses exclusively on **how vulnerabilities manifest in real Java and Python codebases** and how experienced reviewers uncover them. The material aligns with **OWASP standards** and reflects how modern security teams actually perform source code reviews in enterprise and product environments.

The instructor brings **over 30 years of industry experience**, using examples and techniques drawn from real assessments, breach investigations, and secure development programs. The emphasis is on **what matters in practice**, not academic completeness.

Learning outcomes

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

- Perform structured, security-focused source code reviews on Java and Python applications
- Identify OWASP Top 10 vulnerabilities directly from source code
- Recognize insecure coding patterns and dangerous assumptions
- Evaluate authentication, authorization, and data handling logic for security flaws
- Use static analysis tools effectively without relying on them blindly

- Document security findings in a way developers can act on immediately

---

Prerequisites

Participants should have:

- Practical programming experience in Java and/or Python
- Familiarity with web application concepts and APIs
- Basic understanding of security terminology and risk concepts
- Experience reading and navigating unfamiliar codebases

Training outline

1. Foundations of Secure Source Code Review

- Purpose and goals of security-focused code review
- Threat-oriented mindset versus functional correctness
- Understanding attacker perspectives while reviewing code
- Mapping application architecture to review priorities
- Scoping a security code review effectively
- Common reviewer mistakes and blind spots

2. OWASP Standards Applied to Source Code Review

- Overview of OWASP Top 10 from a code reviewer’s perspective
- Translating OWASP categories into concrete code-level indicators
- Identifying trust boundaries in code
- Data flow analysis for security review
- Entry points, sinks, and sensitive operations

3. Injection Vulnerabilities

- Input sources and tainted data identification

  SQL injection patterns in Java and Python
  - ORM misuse and false sense of safety
- Command injection through system calls
- LDAP, XPath, and expression language injections

- Unsafe string construction patterns
- Defensive coding indicators reviewers should expect to see

4. Authentication and Identity Handling

- Password handling logic and storage review
- Authentication workflow analysis
- Token generation, validation, and lifecycle flaws
- Session management mistakes in code
- Hardcoded credentials and backdoor logic
- Common framework misuse patterns

5. Authorization and Access Control Failures

- Missing authorization checks in business logic
- Insecure direct object references
- Role and permission enforcement flaws
- Inconsistent authorization across layers
- Trusting client-side or upstream checks
- Reviewing authorization logic in controllers, services, and helpers

6. Sensitive Data Exposure

- Identifying sensitive data in code paths
- Insecure handling of PII, credentials, and secrets
- Improper use of cryptographic APIs
- Weak or deprecated algorithms
- Key management issues visible in source code
- Logging of sensitive information

7. Cross-Site Scripting and Output Handling

- Reflected, stored, and DOM-based XSS from a code perspective
- Unsafe rendering patterns in templates
- Improper output encoding
- Framework-specific escaping pitfalls
- JSON and API-related XSS vectors

8. Insecure Deserialization and Unsafe Object Handling

- Identifying deserialization entry points
- Java serialization risks and patterns
- Python pickle and object reconstruction risks
- Trust assumptions around serialized data
- Defensive patterns reviewers should verify

9. Security Misconfiguration as Seen in Code

- Dangerous default configurations embedded in code
- Debug and test flags left enabled
- Improper error handling and information leakage
- Stack trace exposure patterns
- Feature flags and environment-based security failures

10. Using Components with Known Vulnerabilities

- Dependency review from source and configuration files
- Identifying outdated or vulnerable libraries
- Understanding transitive dependency risks
- Mapping vulnerable components to actual code usage
- Limits of automated dependency scanners

11. Static Analysis Tools as Review Aids

- Role of SAST tools in security review
- Tool selection for Java and Python
- Interpreting findings correctly
- Validating tool results manually
- Managing false positives and false negatives
- Integrating tools into a reviewer workflow

12. Documenting and Communicating Findings

- Writing actionable security findings
- Risk rating based on exploitability and impact

- Including code references and proof points
- Providing remediation guidance developers accept
- Aligning findings with OWASP categories
- Review reports used in real organizations

13. End-to-End Secure Code Review Practice

- Reviewing an unfamiliar codebase methodically
- Prioritizing high-risk areas quickly
- Combining manual review with tool output
- Producing a professional security review summary
- Common patterns seen across real-world projects

© 2026 Faysal Aziz · Kuala Lumpur