Faysal Aziz

Code Review and Secure Coding Practices

Practical, industry-centric techniques to discover, analyze, and mitigate vulnerabilities in real-world software in 3 days

FA-0137 · 3 days

Software Development · DevOps, Cloud & Infrastructure · Cybersecurity

Introduction

Security begins with the software you write and maintain. Teams that master secure code review and proactive vulnerability discovery dramatically reduce risk and build trustworthy systems faster.

This course takes you beyond academic theory into hands-on, industry-proven techniques for finding defects, assessing security posture, and integrating intelligence and automation into your development lifecycle. Over three days, you will learn how to review source code rigorously, leverage open-source intelligence, assess cloud configurations, and apply penetration-testing techniques (including AI-assisted approaches) to strengthen both Java and Python projects.

The instructor brings over 30 years of industry experience, focusing on real security challenges faced by organizations today. Instead of abstract exercises, you will work with relevant patterns, tools, and methodologies that mirror how top security teams operate.

Learning outcomes

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

- Conduct structured, effective code reviews for security defects in real Java and Python codebases.
- Apply OSINT techniques to gather contextual intelligence that informs security assessments.
- Assess and harden cloud security configurations using best practices.
- Integrate intelligence-led and automated penetration testing concepts into secure development.

- Use AI-assisted tools responsibly to support code review and vulnerability discovery.
- Communicate findings clearly to engineers and stakeholders to drive remediation.

Prerequisites

Participants should have:

- Working knowledge of software development in Java and/or Python.
- Basic understanding of software security concepts (e.g., OWASP Top Ten).
- Familiarity with version control (e.g., Git) and development workflows.
- Access to development environments for hands-on exercises.

Training outline

1. Code Review and Source Code Security

Fundamentals of Secure Code Review

- Objectives of code review
- Differences between functional and security reviews
- Security-oriented review checklists

Review Methodologies

- Manual review techniques
- Automated static analysis
- Combining manual and tool-based approaches

Language-Specific Patterns

- Java secure coding patterns and anti-patterns
- Python secure coding guidelines
- Common vulnerability classes (injection, insecure deserialization, etc.)

Tools for Code Review

- Static analysis tools for Java (e.g., SpotBugs, PMD)
- Static analysis tools for Python (e.g., Bandit, pylint)
- Integrating tools into CI/CD pipelines

2. Open Source Intelligence (OSINT) in Security Assessment

OSINT Fundamentals

- What OSINT is and how it supports security testing
- Ethical and legal considerations

Relevant OSINT Techniques for Code Review

- Discovering exposed endpoints and sensitive data
- Identifying leaked credentials or API keys
- Profiling open source dependencies and known exploits

Tools and Resources

- Search engines and special operators
- Public vulnerability databases (e.g., NVD, CVE)
- Community platforms for threat insights

3. Cloud Security Configuration Best Practices and Review

Cloud Security Basics

- Shared responsibility models
- Identity and access management principles

Secure Configurations

- Network security (VPC, security groups)
- Storage and data protection
- Logging and monitoring configurations

Assessing Cloud Environments

- Hands-on review: AWS, Azure, or GCP service configurations
- Identifying misconfigurations and risky defaults

Tools for Cloud Configuration Review

- Infrastructure as Code linters and scanners
- Continuous monitoring solutions

4. Intelligence-Led Penetration Testing

Foundations of Penetration Testing

- Goals and scope definition
- Ethical rules of engagement

Intelligence-Led Approaches

- Using OSINT to plan and prioritize tests
- Threat modeling based on real world data

Hands-On Test Scenarios

- Simulating advanced attacks against sample applications
- Correlating findings with code review insights

Reporting and Remediation

- Translating test results into actionable developer guidance
- Prioritizing fixes based on risk

5. AI-Assisted Penetration Testing and Code Security

Role of AI in Security Assessments

- What AI can and cannot do in testing and review
- Responsible and ethical use of AI tools

AI Tools for Code Review

- Integrating generative assistance for pattern recognition and suggestion
- Reviewing AI suggestions critically

AI-Driven Vulnerability Discovery

- Automated test case generation
- Using machine learning to identify anomalous patterns

Case Studies

- Examples of AI assist uncovering weaknesses in Java/Python code
- Lessons learned from real engagements

---

6. Capstone Integration

Combined Assessment Exercise

- Participants perform a secure code review of a sample application
- Apply OSINT, cloud configuration checks, and targeted testing
- Present findings and remediation strategy

© 2026 Faysal Aziz · Kuala Lumpur