

Foundations of Modern Cybersecurity

Guided labs in code review, configuration and AI-assisted analysis

FA-0618 · 1 day

Cybersecurity

Practise small defensive security exercises using prepared code, configuration and evidence, with careful review of AI suggestions.

Introduction

This one-day lab-based course introduces practical defensive security checks across code, dependencies, public-information exposure and cloud configuration. Participants use prepared examples to record findings, consider risk and review possible controls.

To keep the scope realistic, coding exercises use one selected familiar language, with comparative Java/Python examples and short demonstrations for other topics. AI assistants may help organise or propose findings, but suggestions are checked against evidence and authoritative guidance.

Activities use synthetic data, authorised training repositories and simulated or isolated environments. No real credentials, client information or unauthorised testing targets are used.

Learning outcomes

The course teaches participants to:

- Apply selected input-validation and secret-handling checks to a small code sample.
- Review a prepared repository and distinguish dependency findings from code-analysis results.
- Document exposure findings from permitted public or prepared information.
- Identify selected IAM, storage and network-configuration issues in an isolated example.
- Create a small threat model and qualitative risk register.
- Compare AI-generated analysis with manual evidence and identify unsupported suggestions.

- Explain authorisation, privacy and scope boundaries for security testing.

Prerequisites

- Basic programming skills in Java or Python.
- Familiarity with command-line basics.
- Interest in cybersecurity fundamentals.

A prepared lab environment with a familiar Java or Python toolchain and access to the selected review tools. Cloud and AI activities depend on approved training accounts, permissions and service availability.

Use dummy secrets and non-confidential sample inputs. Cloud review may be read-only or simulated where live lab access is unavailable.

Training outline

1. 1. Scope and Responsible Practice

- Course goals, rules of engagement
- What constitutes ethical vs unethical actions
- How we will use AI tools safely

2. 2. Secure-Coding Exercise

Input Validation

- Write Java/Python programs that validate user input
- Use safe libraries and avoid unsafe parsing

Authentication Example

- Review a prepared authentication example rather than build a production login system.
- Use an appropriate password-hashing library; discuss Argon2id for new systems and bcrypt's legacy limitations. Never store plaintext passwords.

Sensitive Data and Secrets

- Compare masking, encryption and access-control purposes using dummy values; masking is not encryption.
- Discuss why secrets should not be hard-coded or sent to AI services.

Dependencies

- Inspect a prepared dependency scan, for example OWASP Dependency-Check where appropriate to the project; confirm affected versions and false-positive possibilities.
- Document supported findings and a proposed fix or investigation; a clean scan is not proof of security.

AI-Supported Review

- Ask ChatGPT / Gemini / Manus AI to suggest secure coding improvements
- Compare AI recommendations with secure coding checklists

3. 3. Structured Code Review

- Confidentiality, input validation, error handling
- Review selected snippets from a provided Java repository.
- Compare equivalent Python snippets, or use Python as the primary exercise language where appropriate.
- Document findings in a report format

AI-Assisted Comments

- Use AI to generate review comments
- Validate AI suggestions manually

4. 4. Defensive OSINT Exercise

- Use prepared search examples or permitted queries for a training target; do not collect unnecessary personal information.
- Analyse metadata in non-confidential sample public content.
- Review selected defensive OSINT tools and their scope; no intrusive scanning or access attempts.
- Document what types of information are visible and why that matters

AI-Supported Organisation

- Use AI prompts to organize OSINT findings
- Compare manual vs AI summaries

5. 5. Cloud Configuration Review

- Shared Responsibility Model Walkthrough
- Work with a sandbox cloud environment
- Identify misconfigurations (IAM roles, public storage, network rules)
- Compare sample settings with a relevant checklist and propose justified changes; only change settings inside an explicitly authorised sandbox.

Reviewing AI Checklists

- Ask AI for configuration checklists
- Interpret sample warnings and verify AI explanations against actual settings and official documentation.

6. 6. Threat Model and Risk Register

- Identify assets, threats, and controls
- Use a simple qualitative impact/likelihood scale and document uncertainty.
- Risk register creation

AI-Suggested Scenarios

- AI prompts to expand threat scenarios
- Validate output against framework templates

7. 7. Security-Testing Concepts and Simulation

- Review prepared vulnerability-test cases at awareness level.
- Use a short, instructor-led simulation in a permitted training environment.
- Ask AI to explain potential vulnerabilities
- Discuss AI-supported analysis and why automated suggestions need verification and controlled permissions.

Testing is limited to authorised simulations; this module does not provide a professional penetration-testing qualification.

8. 8. Reflection and Findings

- Group sharing of lab results

- Discussion on ethical & legal boundaries
- Q&A on AI tools and secure practice

Training Tools

- Languages: Java, Python
- Optional AI-assistant examples: ChatGPT, Gemini or Manus, subject to approved access and data-handling rules.
- Cloud Sandbox: Local or hosted environment (read-only or simulated)
- Dependency analysis: OWASP Dependency-Check where suitable; separate linters/static-analysis tools for source-code checks.