Secure Software Development Lifecycle (SSDLC) with Hands-On Coding

CYBER SCHOOL | PROFESSIONAL TRAINING FOR ADULTS

A 4.5-hour, hands-on workshop for software professionals who want to build security into every line of code.

Security that is bolted on at the end of a project is expensive, fragile, and frequently too late. This workshop teaches participants to integrate security into every phase of the software development lifecycle - from the first requirements meeting to production monitoring - and to prove it with working code. Participants analyze real-world breaches such as MOVEit and T-Mobile, write and fix vulnerable code, build a threat model for a realistic e-commerce system, and run industry-standard security testing tools against a live application.

Detail	Information
Duration	4.5 hours of instructor-led training across six sessions
Audience	Software developers, security engineers, QA testers, IT professionals
Format	Instructor-led sessions combined with hands-on labs and group exercises
Language	English
Prerequisites	Working familiarity with at least one programming language and basic web application concepts

CYBER SCHOOL™

Learning Outcomes and Instructional Approach

Learning Outcomes

Upon successful completion of this workshop, participants will be able to:

1	**SDLC Security Integration** Explain why security must be incorporated into every phase of the SDLC and quantify the cost of leaving it out.
2	**Security Requirements** Define security requirements alongside functional requirements, using use cases and misuse cases.
3	**Threat Modeling** Build a threat model for a realistic system and propose concrete, prioritized mitigations.
4	**Vulnerability Remediation** Identify and fix common vulnerability classes in code, including SQL injection, XSS, CSRF, and buffer overflows.
5	**Security Testing Tools** Operate security testing tools (SAST, DAST, penetration testing) against a running application.
6	**DevSecOps Pipeline** Design a DevSecOps pipeline that makes security continuous rather than episodic.

Instructional Approach

Every topic in this workshop follows the Cyber School learning arc:

1	2	3
Learn	**Practice**	**Challenge**
Structured theory framed by real-world context, with measurable objectives for every module.	Guided, hands-on exercises with clear steps and success criteria defined in advance.	Independent tasks that prove mastery: live code fixes, tool runs, and a team deliverable presented to the group.

CYBER SCHOOL™

Course Schedule - Part 1

Session 1 - Introduction to SSDLC (30 minutes)

Topics: The traditional SDLC phases; why secure development matters; the CIA Triad; Principle of Least Privilege; Defense in Depth.

Case studies: The MOVEit file-transfer breach and the T-Mobile data breach.

Activity: Group analysis of recent breaches - identifying the SDLC phase where each vulnerability was introduced.

Session 2 - Phase-by-Phase Breakdown with Code (60 minutes)

Topics: Security requirements and misuse cases; threat modeling at design time; secure coding against the OWASP Top 10; SAST, DAST and IAST testing; hardened deployment and monitoring; continuous maintenance.

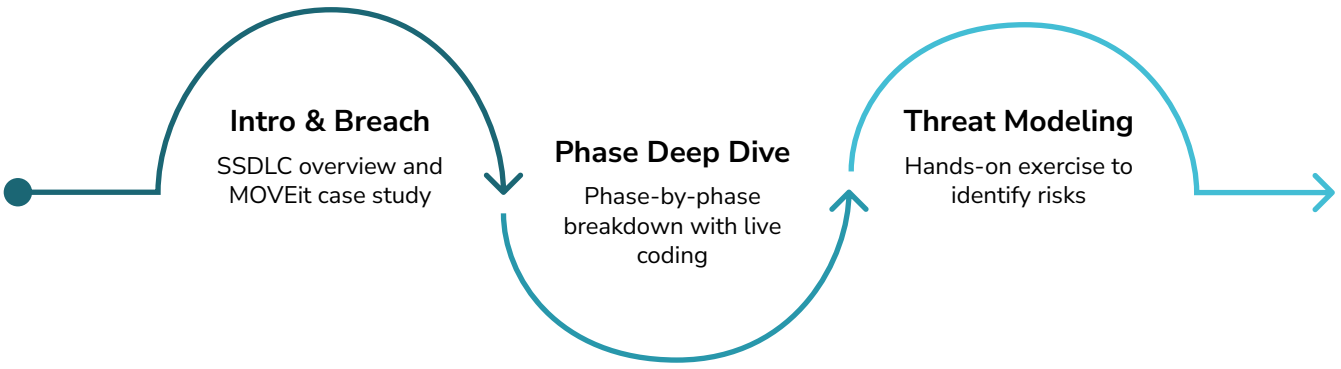
Live code: Preventing SQL injection with parameterized queries; escaping output against XSS; secure logging in Python.

Tools: Microsoft Threat Modeling Tool, OWASP Threat Dragon, SonarQube, Dependabot.

Session 3 - Hands-On Threat Modeling (45 minutes)

Scenario: Participants receive the system diagram of an e-commerce platform and identify potential threats such as data exposure and weak authentication.

Deliverable: A team threat model presented with suggested mitigations.



Part 1 totals 135 minutes and establishes the foundational knowledge and threat modeling skills that underpin all subsequent hands-on work.

CYBER SCHOOL™

Course Schedule - Part 2

Session 4 - Secure Coding Practices Workshop (45 minutes)

Topics: Buffer overflows in C; injection attacks; XSS; CSRF protection with tokens.

Challenge: Participants receive insecure code snippets and must identify and correct the security flaws.

Session 5 - Security Testing Hands-On (45 minutes)

Topics: Automating SAST, DAST and IAST in the development workflow.

Tool lab: Dynamic testing of a deliberately vulnerable web application with OWASP ZAP or Burp Suite.

Session 6 - Integrating DevSecOps and Wrap-Up (45 minutes)

Topics: Security inside the CI/CD pipeline; automated scanning; real-time monitoring with ELK Stack or Datadog.

Design exercise: Participants design a DevSecOps pipeline for their own organization.

Wrap-up: Recap of the SSDLC phases, open Q&A, and next steps.

Full Workshop at a Glance

#	Session	Duration	Key Activity	Tools / Output
1	Introduction to SSDLC	30 min	Breach analysis group exercise	Case studies: MOVEit, T-Mobile
2	Phase-by-Phase with Code	60 min	Live coding: SQL injection, XSS, logging	SonarQube, Dependabot, Threat Dragon
3	Hands-On Threat Modeling	45 min	Team threat model for e-commerce platform	MS Threat Modeling Tool, Threat Dragon
4	Secure Coding Workshop	45 min	Fix insecure code snippets	Vulnerable code challenge set
5	Security Testing Hands-On	45 min	Dynamic testing of vulnerable app	OWASP ZAP, Burp Suite
6	DevSecOps and Wrap-Up	45 min	Design a DevSecOps pipeline	ELK Stack, Datadog, CI/CD design

Assessment, Materials and References

Assessment and Completion Criteria

This workshop is assessed on demonstrated competence, not attendance alone. To complete the workshop, participants must:

- Actively participate in all hands-on exercises
- Submit the team threat modeling deliverable
- Successfully identify and correct the vulnerable code snippets
- Pass the end-of-workshop knowledge check quiz

Materials and Tools

Provided: A demo environment with deliberately vulnerable web applications (OWASP Juice Shop), all tools required for the duration of the workshop, and the full workshop textbook.

Required: A laptop capable of running a modern web browser.

Toolset used:

OWASP ZAP	**Burp Suite**
SonarQube	**MS Threat Modeling Tool**
OWASP Threat Dragon	**Dependabot**

References and Further Reading

SAFECode

Fundamental Practices for Secure Software Development (2018)

OWASP Top 10

The Ten Most Critical Web Application Security Risks

Ponemon Institute

Cost of a Data Breach Report (2021)

CERT Secure Coding Standards

CERT Secure Coding Standards (2020)

Tool Documentation

Microsoft Threat Modeling Tool and OWASP ZAP documentation

Participants who complete all assessment criteria will receive a Cyber School certificate of completion for this workshop.

CYBER SCHOOL™