Web Application Security Training Syllabus

CorporateSpace Cybersecurity Academy

Hands-On Training for Developers and IT Professionals

Last Updated: August 2025

https://corporatespace.in/web-application-security-training

Contents

1 Course Overview

This Web Application Security Training program equips participants with the knowledge and skills to protect web applications from modern cyber threats. Through hands-on labs, real-world projects, and expert-led instruction, learners will master secure coding practices, vulnerability identification, and penetration testing techniques. The curriculum aligns with industry standards, including OWASP Top 10, and is designed to meet the needs of developers, IT security professionals, and corporate teams.

1.1 Course Objectives

- Understand web application vulnerabilities and their impact on business.
- Implement secure coding practices to mitigate common threats like XSS, SQL Injection, and CSRF.
- Conduct threat modeling, risk assessments, and penetration testing.
- Integrate security into the software development lifecycle (SDLC) and CI/CD pipelines.
- Perform a security audit and harden a web application through a real-time project.

1.2 Course Duration

- Total: 25–30 hours
- Format: Live sessions (onsite, online via Zoom/Google Meet, or hybrid)
- Schedule: Flexible, including weekend batches for corporate teams

1.3 Target Audience

- Web Developers and Software Engineers
- IT Security Professionals
- Students and Graduates pursuing cybersecurity careers
- Startups and Founders building secure applications

1.4 Prerequisites

- Basic knowledge of web development (HTML, JavaScript, or any server-side language) is helpful but not mandatory.
- Familiarity with web application operations (e.g., browser/server interaction) is recommended.

2 Module 1: Introduction to Web Application Security

2.1 Importance of Web Security

- Role of web applications in modern business
- Overview of cyber threats: Data breaches, hacking, and privacy violations
- Impact of vulnerabilities on user trust and business reputation

2.2 Web Security Fundamentals

- Understanding the web application stack: Frontend, backend, and database
- Common attack vectors: XSS, SQL Injection, CSRF, and more
- Introduction to OWASP Top 10 vulnerabilities[](https://www.sans.org/cyber-security-courses/application-security-securing-web-apps-api-microservices)

3 Module 2: OWASP Top 10 Vulnerabilities

3.1 Exploring Critical Vulnerabilities

- Broken Access Control: IDOR, privilege escalation
- Cryptographic Failures: Insecure encryption and data exposure
- Injection Attacks: SQL, Command, and Code Injection
- Insecure Design and Security Misconfiguration
- Vulnerable and Outdated Components
- Authentication and Session Management Failures
- Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF)
- Insecure Deserialization and Server-Side Request Forgery (SSRF)

3.2 Mitigation Strategies

- Best practices for secure coding and configuration
- Real-world examples of vulnerabilities and their fixes[](https://frontendmasters.com/courses/web security-v2/)

4 Module 3: Secure Coding Practices

4.1 Input Validation and Sanitization

- Validating and sanitizing user inputs to prevent injection attacks
- Using libraries and frameworks for secure input handling

4.2 Authentication and Authorization

- Implementing secure authentication (OAuth, SAML, SSO)
- Role-based access control and session management best practices
- Password-less authentication and multi-factor authentication (MFA)

4.3 Secure Data Handling

- Encrypting sensitive data with HTTPS and TLS
- Secure storage and transmission of data
- Using Content Security Policy (CSP) and HTTP security headers[](https://www.offsec.com/cyberver application-security/)

5 Module 4: Penetration Testing and Security Tools

5.1 Introduction to Penetration Testing

- Penetration testing methodologies: White box, black box, and gray box
- Setting up a penetration testing environment

5.2 Tools and Techniques

- Burp Suite: Intercepting and analyzing HTTP requests
- OWASP ZAP: Automated vulnerability scanning
- Manual exploitation techniques for advanced testing
- Configuring and using security testing tools in VS Code and GitHub

5.3 Hands-On Labs

- Simulating attacks: SQL Injection, XSS, and CSRF
- Identifying and fixing vulnerabilities in a sample application

6 Module 5: Securing APIs and RESTful Services

6.1 API Security Fundamentals

- Understanding REST, SOAP, and GraphQL APIs
- Common API vulnerabilities: Broken authentication, excessive data exposure

6.2 Securing APIs

- Implementing secure API authentication with JWT, OAuth
- Rate limiting, input validation, and error handling
- Protecting against API-specific attacks (e.g., parameter tampering)

6.3 Lab: API Security Testing

- Testing an API for vulnerabilities using Burp Suite
- Applying security fixes and verifying results

7 Module 6: Security Automation and CI/CD Integration

7.1 Integrating Security into SDLC

- Introduction to DevSecOps: Shifting left in security
- Embedding security checks in development pipelines

7.2 Automated Security Testing

- Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST)
- Integrating tools like OWASP ZAP into CI/CD pipelines
- Automating vulnerability scans with GitHub Actions

7.3 Lab: Automating Security Checks

- Setting up a CI/CD pipeline with security testing
- Reviewing automated scan reports and addressing findings

8 Module 7: Real-Time Project: Security Audit and Hardening

8.1 Project Overview

- Conducting a security audit on a sample web application
- Identifying vulnerabilities using manual and automated tools

8.2 Implementing Fixes

- Applying secure coding practices to fix vulnerabilities
- Configuring HTTPS, CORS, and CSP headers
- Validating fixes through re-testing

8.3 Project Deliverables

- Security audit report with findings and recommendations
- Hardened web application codebase hosted on GitHub

9 Module 8: Career Development and Certification

9.1 Building a Security Portfolio

- Crafting a resume with security project highlights
- Showcasing the project codebase on GitHub

9.2 Certification and Support

- Earning a CorporateSpace Web Application Security Certificate
- 30 days of post-training doubt-solving and project support
- Job and internship references (on request)

10 Course Materials and Tools

10.1 Provided Resources

- Comprehensive course slides and cheat sheets
- Access to recorded sessions for review
- Sample vulnerable application for practice
- Templates for security audit reports

10.2 Tools Covered

- Burp Suite (Community Edition)
- OWASP ZAP
- Visual Studio Code
- GitHub for version control and project hosting
- HTTPS and security header configurations

11 Program Delivery Options

11.1 Training Formats

- Onsite: At your company premises
- Online: Live sessions via Zoom/Google Meet
- Hybrid: Weekend batches for flexible learning

11.2 Customization for Corporate Teams

- Tailored modules based on company needs
- Flexible scheduling for team availability
- Integration with company-specific tech stacks

12 Instructor Profile

- Certified Security Expert and Developer with over 6 years of experience
- Delivered training to 15+ corporate teams globally
- Expertise in secure coding, penetration testing, and DevSecOps