Java Web App Secure Coding

4-Day Course

Course Syllabus

AppSec Labs Ltd.
info@appsec-labs.com | https://appsec-labs.com | 10 HaTa'as St., Kfar Saba 44641 Israel | T: +972-9-7485005 | F: +972-9-7730595

Java secure coding

4-day course

Course description

Secure programming is the best defense against hackers. This multilayered hands-on course will demonstrate live real time hacking methods , analyze the code deficiency that enabled the attack and most importantly, teach how to prevent such vulnerabilities by adopting secure coding best practices in order to bullet-proof your J2EE application.

The methodology of the Cycle of knowledge is as follows: Understand, Identify, Prevent. This methodology presents the student with analytical tools to keep a deeper understanding of coding vulnerabilities and implement security countermeasures in different areas of the software development lifecycle. The hands on labs will enable the student to get a firsthand experience of the Hackers world and what could be done to stop him. Using sound programming techniques and best practices shown in this course, you will be able to produce high-quality code that stands up to attack.

The course covers major security principles in the Java framework, programming vulnerabilities, and specific security issues in J2EE web applications.

Target audience

Members of the software development team:

- Java developers in J2EE based applications
- Designers & architects.

Prerequisites

Before attending this course, students should be familiar with:

- Basic knowledge of the Java framework
- Apache/Tomcat, Databases (MySQL/Oracle) & SQL language

Course topics

Day 1

Authentication

- What is authentication?
- Password storage
- Securing passwords
- Brute force attacks
- Anti-automation
- User enumeration
- Authentication types
- SSO (Single Sign-On)
- Two-Factor Authentication
- Kerberos
- JAAS - Authentication
- **LAB**

Authorization

- Client side authorization
- Forceful browsing
- UI based security
- Parameter tampering
- Insecure direct object reference
- File authorization
- URL authorization
- ACL (Access Control List)
- RBAC (Role-Based Access Contorl)
- Java Security Manager
- JAAS – Authorization
- OAuth
- **LAB**

Input Validation

- Injection Flaws
- Integer Overflows & Underflows
- OS command injection
- SQL Injection
- Prepared statement
- Store procedure
- Xpath injection
- LDAP injection
- Data type conversion
- Black list vs. White list
- Regular expression
- Client-side validations
- **LAB**

Day 2

Output Encoding

- Cross-Site Scripting (XSS)
- What is encoding
- Encoding types
- ESAPI library
- XSS prevention cheat sheet
- XML Encoding
- Response Splitting
- **LAB**

Browser Manipulation

- Cross Site Request Forgery (CSRF)
- Anti CSRF solutions
- Open redirect
- Sandboxing
- Clickjacking
- Browser's cache
- Security Headers
- Session management
- Cookie's properties
- Session fixation
- **LAB**

File Handling

- Directory traversal
- Canonicalization
- Backdoors
- File extension handling
- Filenames threats
- Directory listing
- Isolation storage
- ACL
- **LAB**

Day 3

Data Confidentiality & Integrity

- Homemade algorithm
- Introduction to Crypto
- Privacy Standards
- Insecure storage
- Java Cryptography Architecture (JCA)
- Symmetric encryption
- A-Symmetric encryption
- Insecure communication
- Secure traffic enforcement
- Hash functions
- Digital signatures
- Key Management
- Certificates
- DPAPI
- Randomization
- **LAB**

Error Handling

- Information disclosure
- Fail securely
- Exceptions and stack trace
- Custom error pages
- **LAB**

Security Logging

- Logging technologies
- Events you should log
- Events you should not log
- Integration with exception management
- **LAB**

Business Logic

- Logical attacks
- Flow bypassing
- Replay attacks
- Abuse of functionality
- WSDL Information disclosure
- WS Message tampering
- WS security measures
- **LAB**

Day 4 – Advanced topics

Secure Design and Principles

- Implementing security as part of the SDLC
- Secure Design
- Common Security Principles
- Segmentation
- Layered Security

Java SE Security

- What is Java SE?
- Platform Security
- Bytecode Verification
- Secure Class Loading
- Crypto APIs
- Cryptographic Service Providers
- Authentication APIs
- Access Control Architecture
- Access Control Enforcement
- Secure Communication APIs
- Public Key Infrastructure (PKI)

Advanced Topics in Java Security

- Java Servlet Filter
- JSSE (Java Secure Socket Extension)
- JAX –RS: securing RESTful Web-Services
- Security Code Review: how to find security holes in your code
- Security Tools and Scanners
- Penetration Testing: do it yourself

Spring Security

- Security Concepts
- Authentication Types
- Run-As Replacement
- Authorization Manager
- Security Interceptor
- Roles and Expression-Based Access Control
- Spring Security Configuration
- @Pre / @Post Filters
- The Security Filter Chain
- Core Security Filters
- Remember-Me
- Session Management
- Spring Security Crypto Module
- Password Encoding