Java Secure Coding for Client-Server Applications

3-Day hands on 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 for Client-Server Applications 3-Day hands on 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
- Store password securely
- Hashing
- Brute force
- Dictionary attack
- Anti-automation
- CAPTCHA
- Account lockout
- User enumeration
- Basic & Digest authentication
- Windows integration
- Form based 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 ACL)
- **LAB**

Input Validation

- OS command injection
- SQL Injection
- Prepared statement

- Store procedure
- Xpath injection
- LDAP injection
- Data type conversion
- Black list
- White list
- **LAB**

Day 2:

File Handling

- Directory traversal
- Canonicalization
- File extension handling
- Filenames threats
- Directory listing
- **LAB**

Data Confidentiality & Integrity

- Homemade algorithm
- Insecure communication
- Secure traffic enforcement
- Insecure storage
- Symmetric encryption
- A-Symmetric encryption
- Java Cryptography Architecture (JCA)
- Hash functions
- Digital signatures
- **LAB**

Application Denial of Service Vulnerabilities

- Application / OS crash
- CPU starvation
- Memory starvation
- File system starvation
- Resource starvation
- Resource locking
- Triggering high network bandwidth
- User level DoS
- Exploiting a specific vulnerability to cause DoS
- **LAB**

Day 3:

Code Protection

- Reverse engineering techniques
- Obfuscation
- Native compilers
- Jar protection - Signed jar, Sealed jar
- Digitally signed applets
- Secure object serialization
- **LAB**

Error Handling

- Information disclosure
- Exceptions and stack trace
- Default error pages
- **LAB**

Security Logging

- Logging technologies
- Events you should log
- Events you should not log
- **LAB**

Business Logic

- Logical attacks
- Flow bypassing
- Replay attacks
- Abuse of functionality
- **LAB**