Maslak Mahallesi, Maslak Meydan Sk. No:5. Spring Giz Plaza. Maslak/İstanbul
+90 212 282 7700 - info@bilginc.com

Java and Web Application Security

Learn via: **Classroom**

Duration: **3 Days**

Overview

Writing web applications in Java can be rather complex – reasons range from dealing with legacy technologies or underdocumented third-party components to sharp deadlines and code maintainability. Yet, beyond all that, what if we told you that attackers were trying to break into your code right now? How likely would they be to succeed?

This course will change the way you look at your Java code. We'll teach you the common weaknesses and their consequences that can allow hackers to attack your system, and – more importantly – best practices you can apply to protect yourself. We cover typical Web vulnerabilities with a focus on how they affect Java web apps on the entire stack – from the Java runtime environment to modern AJAX and HTML5-based frontends. In addition, we discuss the security aspects of the Java platform itself as well as typical Java programming mistakes you need to be aware of. We present the entire course through live practical exercises to keep it engaging and fun.

Writing secure code will give you a distinct edge over your competitors. It is your choice to be ahead of the pack – take a step and be a game-changer in the fight against cybercrime.

Topics include:

- IT security and secure coding
- Web application security
- Client-side security
- Foundations of Java security
- Practical cryptography
- Java security services
- Common coding errors and vulnerabilities
- Principles of security and secure coding
- Knowledge sources

Prerequisites

General Java development skills are required.

What You Will Learn

- Understand basic concepts of security, IT security and secure coding
- Learn Web vulnerabilities beyond OWASP Top Ten and know how to avoid them
- Learn about XML security
- Learn how to set up and operate the deployment environment securely
- Learn client-side vulnerabilities and secure coding practices
- Learn to use various security features of the Java development environment
- Have a practical understanding of cryptography
- Learn about typical coding mistakes and how to avoid them
- Get information about some recent vulnerabilities in the Java framework
- Get sources and further readings on secure coding practices

Note: This course comes with a number of easy-to-understand exercises providing real-time ethical hacking fun. By accomplishing these exercises with the support of the trainer, participants can analyze vulnerable code snippets and commit attacks against them in order to fully understand the root causes of certain security problems. All exercises are prepared in a plug-and-play manner by using a pre-set desktop virtual machine, which provides a uniform development environment.

Outline

Day 1

IT security and secure coding

- Nature of security
- What is risk?
- IT security vs. secure coding
- From vulnerabilities to botnets and cybercrime
- Nature of security flaws
- Reasons of difficulty
- From an infected computer to targeted attacks
- The Seven Pernicious Kingdoms
- OWASP Top Ten 2017

Web application security

- Injection
- Injection principles
- SQL injection
- Exercise – SQL Injection
- Exercise – SQL injection
- Typical SQL Injection attack methods
- Blind and time-based SQL injection
- SQL injection protection methods
- Other injection flaws
- Command injection
- Case study – ImageMagick
- Broken authentication
- Session handling threats
- Session handling best practices
- Session handling in Java
- Setting cookie attributes – best practices
- Sensitive data exposure
- Transport layer security
- Enforcing HTTPS
- XML external entity (XXE)
- XML Entity introduction
- XML bomb
- Exercise – XML bomb
- XML external entity attack (XXE) – resource inclusion
- XML external entity attack – URL invocation
- XML external entity attack – parameter entities
- Exercise – XXE attack
- Preventing entity-related attacks
- Case study – XXE in Google Toolbar
- Broken access control
- Typical access control weaknesses
- Insecure direct object reference (IDOR)
- Exercise – Insecure direct object reference
- Protection against IDOR
- Case study – Facebook Notes
- Exercise – Authorization bypass
- Security misconfiguration
- Configuration management
- Hardening
- Patch management
- Configuring the environment
- Insecure file uploads
- Exercise – Uploading executable files
- Filtering file uploads – validation and configuration
- Cross-Site Scripting (XSS)
- Persistent XSS
- Reflected XSS
- DOM-based XSS
- Exercise – Cross Site Scripting
- Exploitation: CSS injection
- Exploitation: injecting the <base> tag
- Exercise – HTML injection with base tag
- XSS prevention
- XSS prevention tools in Java and JSP
- Insecure deserialization
- Deserialization basics

- Security challenges of deserialization
- Deserialization in Java
- From deserialization to code execution
- POP payload targeting the Apache Commons gadget (Java)
- Real-world Java examples of deserialization vulnerabilities
- Issues with deserialization – JSON
- Best practices against deserialization vulnerabilities

Day 2

Client-side security

- JavaScript security
- Same Origin Policy
- Cross Origin Resource Sharing (CORS)
- Exercise – Client-side authentication
- Client-side authentication and password management
- Protecting JavaScript code
- Exercise – JavaScript obfuscation
- Clickjacking
- Exercise – Do you Like me?
- Protection against Clickjacking
- Anti frame-busting – dismissing protection scripts
- Protection against busting frame busting
- AJAX security
- XSS in AJAX
- Script injection attack in AJAX
- Exercise – XSS in AJAX
- XSS protection in Ajax
- Exercise CSRF in AJAX – JavaScript hijacking
- CSRF protection in AJAX
- HTML5 security
- New XSS possibilities in HTML5
- HTML5 clickjacking attack – text field injection
- HTML5 clickjacking – content extraction
- Form tampering
- Exercise – Form tampering
- Cross-origin requests
- HTML proxy with cross-origin request
- Exercise – Client side include

Foundations of Java security

- The Java environment
- Java security
- Low-level security – the Java language and environment
- Java language security
- Type safety
- Automatic memory management
- Java execution overview
- Bytecode Verifier
- Class Loader
- Protecting Java code
- High-level security – access control
- Protection domains
- Security Manager and Access Controller
- Permission checking
- Effects of doPrivileged

Practical cryptography

- Cryptosystems
- Elements of a cryptosystem
- Symmetric-key cryptography
- Providing confidentiality with symmetric cryptography
- Symmetric encryption algorithms
- Block ciphers – modes of operation

- Other cryptographic algorithms
- Hash or message digest
- Hash algorithms
- SHAttered
- Message Authentication Code (MAC)
- Providing integrity and authenticity with a symmetric key
- Random numbers and cryptography
- Cryptographically-strong PRNGs
- Hardware-based TRNGs
- Asymmetric (public-key) cryptography
- Providing confidentiality with public-key encryption
- Rule of thumb – possession of private key
- Combining symmetric and asymmetric algorithms
- Public Key Infrastructure (PKI)
- Man-in-the-Middle (MitM) attack
- Digital certificates against MitM attack
- Certificate Authorities in Public Key Infrastructure
- X.509 digital certificate
- Exercise Jars – Granting permission to signed code

Java security services

- Java security services – architecture
- Java Cryptographic Architecture
- Java Cryptography Architecture / Extension (JCA/JCE)
- Using Cryptographic Service Providers
- Engine classes and algorithms
- Exercise Sign – Generating and verifying signatures

Day 3

Common coding errors and vulnerabilities

- Input validation
- Input validation concepts
- Integer problems
- Representation of negative integers
- Integer overflow
- Exercise IntOverflow
- What is the value of Math.abs(Integer.MIN_VALUE)?
- Integer problem – best practices
- Java case study
- Path traversal vulnerability
- Path traversal mitigation
- Case study – Insufficient URL validation in LastPass
- Unvalidated redirects and forwards
- Unsafe native calls
- Unsafe JNI
- Exercise Unsafe JNI
- Unsafe reflection
- Implementation of a command dispatcher
- Unsafe reflection – spot the bug!
- Mitigation of unsafe reflection
- Log forging
- Some other typical problems with log files
- Improper use of security features
- Typical problems related to the use of security features
- Insecure randomness
- Weak PRNGs in Java
- Exercise RandomTest
- Using random numbers in Java – spot the bug!
- Password management
- Exercise – Weakness of hashed passwords
- Password management and storage
- Special purpose hash algorithms for password storage
- Argon2 and PBKDF2 implementations in Java
- bcrypt and scrypt implementations in Java
- Case study – the Ashley Madison data breach
- Typical mistakes in password management

- Exercise – Hard coded passwords
- Accessibility modifiers
- Accessing private fields with reflection in Java
- Exercise Reflection – Accessing private fields with reflection
- Exercise ScademyPay – Integrity protection weakness
- Improper error and exception handling
- Typical problems with error and exception handling
- Empty catch block
- Overly broad throws
- Overly broad catch
- Using multi-catch
- Catching NullPointerException
- Exception handling – spot the bug!
- Exercise ScademyPay – Error handling
- Exercise – Information leakage through error reporting
- Code quality problems
- Dangers arising from poor code quality
- Poor code quality – spot the bug!
- Unreleased resources
- Private arrays – spot the bug!
- Private arrays – typed field returned from a public method
- Exercise Object Hijack
- Public method without final – object hijacking
- Immutable String – spot the bug!
- Exercise Immutable Strings
- Immutability and security
- Serialization – spot the bug!
- Exercise Serializable Sensitive

Principles of security and secure coding

- Matt Bishop's principles of robust programming
- The security principles of Saltzer and Schroeder

Knowledge sources

- Secure coding sources – a starter kit
- Vulnerability databases
- Java secure coding sources
- Recommended books – Java