Training: SCADEMY
CL-JSM Java and Web application security master course

TRAINING GOALS:

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.

Participants attending this course will:

- 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 client-side vulnerabilities and secure coding practices
- Have a practical understanding of cryptography
- Learn to use various security features of the Java development environment
- Learn about typical coding mistakes and how to avoid them
- Get information about some recent vulnerabilities in the Java framework
- Understand security concepts of Web services
- Learn about JSON security
- Learn about Hibernate security
- Understand security solutions of Java EE
- Learn about JSF and PrimeFaces security
- Learn about Spring security
- Learn about denial of service attacks and protections

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- Get practical knowledge in using security testing techniques and tools
- Get sources and further readings on secure coding practices

Audience:

Java EE developers, software architects and testers

CONSPECT:

- 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
- Web application security (OWASP Top Ten 2017)
  - OWASP Top Ten 2017
  - A1 - Injection
    - Injection principles
    - SQL injection
      - Exercise – SQL injection
      - Typical SQL Injection attack methods
      - Blind and time-based SQL injection
      - SQL injection protection methods
      - Effect of data storage frameworks on SQL injection
    - Other injection flaws
      - Command injection
      - Case study – ImageMagick
  - A2 - Broken authentication
    - Session handling threats
    - Session handling best practices
    - Session handling in Java
    - Setting cookie attributes – best practices
    - Cross site request forgery (CSRF)
      - CSRF prevention
      - CSRF prevention in Java frameworks

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- A3 - Sensitive data exposure
  - Sensitive data exposure
  - Transport layer security
    - Enforcing HTTPS
- A4 - XML external entity (XXE)
  - XML Entity introduction
  - 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
- A5 - Broken access control
  - Typical access control weaknesses
  - Insecure direct object reference (IDOR)
  - Exercise – Insecure direct object reference
  - Protection against IDOR
  - Case study – Facebook Notes
- A6 - Security misconfiguration
  - Security misconfiguration
  - Configuring the environment
  - Insecure file uploads
  - Exercise – Uploading executable files
  - Filtering file uploads – validation and configuration
- A7 - Cross-Site Scripting (XSS)
  - Persistent XSS
  - Reflected XSS
  - DOM-based XSS
  - Exercise – Cross Site Scripting
  - XSS prevention
  - XSS prevention tools in Java and JSP

- Web application security (OWASP Top Ten 2017)
  - A8 - Insecure deserialization
    - Serialization and deserialization basics
    - Security challenges of deserialization
    - Deserialization in Java

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- Denial-of-service via Java deserialization
- From deserialization to code execution
- POP payload targeting InvokerTransformer (Java)
- Real-world Java deserialization vulnerabilities
- Issues with deserialization – JSON
- Best practices against deserialization vulnerabilities

- A9 - Using components with known vulnerabilities
  - Vulnerability attributes
  - Common Vulnerability Scoring System – CVSS
- A10 - Insufficient logging and monitoring
  - Detection and response
  - Logging and log analysis

- Client-side security
  - JavaScript security
  - Same Origin Policy
  - Simple requests
  - Preflight requests
  - Exercise – Client-side authentication
  - Client-side authentication and password management
  - Protecting JavaScript code
  - Clickjacking
    - Clickjacking
    - Exercise – IFrame, Where is My Car?
    - 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

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

        - HTML5 clickjacking – content extraction
        - Form tampering
        - Exercise – Form tampering
        - Cross-origin requests
        - HTML proxy with cross-origin request
        - Exercise – Client side include
- Practical cryptography
    - Rule #1 of implementing cryptography
    - Cryptosystems
        - Elements of a cryptosystem
        - Java Cryptography Architecture / Extension (JCA/JCE)
        - Using Cryptographic Service Providers
    - Symmetric-key cryptography
        - Providing confidentiality with symmetric cryptography
        - Symmetric encryption algorithms
        - Modes of operation
        - Private (symmetric) key cryptography in Java
    - Other cryptographic algorithms
        - Hash or message digest
        - Hash algorithms
        - SHAttered
        - Hashing in Java: MessageDigest class
        - MAC and password-based encryption in Java: Mac class
        - Message Authentication Code (MAC)
        - Providing integrity and authenticity with a symmetric key
        - Random number generation
            - Random numbers and cryptography
            - Cryptographically-strong PRNGs
            - Weak and strong PRNGs in Java
            - Hardware-based TRNGs
            - Exercise RandomTest
            - Using random numbers in Java – spot the bug!
    - Asymmetric (public-key) cryptography
        - Providing confidentiality with public-key encryption
        - Rule of thumb – possession of private key
        - Combining symmetric and asymmetric algorithms

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- 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
  - The Java Keystore (JKS)
  - Java Certification Path (CertPath)
- Java security services
  - Java security services – architecture
- Foundations of Java security
  - The Java environment
  - 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
    - Allowing untrusted code to load arbitrary classes
  - Code signing in Java
    - Code signing
    - Code signing considerations and best practices
    - Package sealing
    - Exercise Jars – Code signing
- Secure communication in Java
  - SSL and TLS
  - Usage options
  - Security services of TLS
  - SSL/TLS handshake
- Common coding errors and vulnerabilities

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- 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
  - Path traversal vulnerability
    - Path traversal – weak protections
    - Path traversal – best practices
    - 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
- Security of Web services
  - Securing web services – two general approaches
  - SOAP - Simple Object Access Protocol
  - Security of RESTful web services
    - Authenticating users in RESTful web services
    - Authentication with JSON Web Tokens (JWT)
    - Authorization with REST
    - Vulnerabilities in connection with REST
  - XML security
    - Introduction
    - XML parsing
    - XML injection
      - (Ab)using CDATA to store XSS payload in XML
      - Exercise – XML injection

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- Protection through sanitization and XML validation
- XML bomb
- Exercise – XML bomb

- JSON security
  - Embedding JSON server-side
  - JSON injection
  - JSON hijacking
  - Case study – XSS via spoofed JSON element

- Common coding errors and vulnerabilities
  - Improper use of security features
    - Typical problems related to the use of security features
    - 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

- Hibernate security
  - Hibernate security overview
    - The Hibernate framework and ecosystem
    - Persistence-related security issues
  - Hibernate-specific vulnerabilities
    - HQL injection
    - Bypassing the HQL lexer
    - ASTification example
    - HQL injection exercise
    - Protection against HQL injection
    - Persisting tainted objects (XSS)
    - Hibernate Validator

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

        - Hibernate Validator example
        - The equals() and hashCode() problem
        - Other Hibernate related issues and recommendations
        - Other query possibilities in Hibernate
- Java EE security
    - Java EE multi-tier architecture
    - Roles and responsibilities
    - Java EE container-based security
    - Basic concepts: user, group, role, realm
    - Security of the Web/Presentation tier
        - Java EE authorization
        - Declaring security roles
        - Checking the caller's identity programmatically
        - Security constraints
        - Exercise WebSecurity – Using JDBC realm
    - Security of the EJB/Business tier
        - Propagating the identity
        - Declaring EJB security roles
        - EJB authorization – annotation
        - EJB authorization – deployment descriptor
        - Accessing the caller's context programmatically
        - Exercise EJBSecurity
        - Exercise EJBSecurity – Preparation
        - Exercise EJBSecurity – Run
        - Exercise EJBSecurity – Enforce authentication
- JSF and PrimeFaces security
    - The Java Server Faces (JSF)
    - PrimeFaces
    - JSF ViewState
        - ViewState (POST)
        - ViewState (GET)
    - Common mistakes
        - JSF/PrimeFaces attack surface
        - Preventing XSS
        - XSS Exercise
        - XSS Exercise – Apply escaping and validation in JSF

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- XSS Exercise – Fixes in Primefaces
- CSRF Exercise
- EL Injection (Expression Language Injection)
- Exercise – EL Injection

- Spring security
  - Spring security overview
    - What makes Spring different?
    - Spring modules
    - Inversion of Control
    - Aspect Oriented Programming (AOP)
    - Aspect Oriented Programming (AOP) and security
  - The Spring Security framework
    - Spring Security overview
    - Integrating the Spring Security framework
    - Spring Security - authentication
      - The authentication process
      - Authentication code example
      - CSRF protection in Spring
    - Spring Security - authorization
      - The authorization process
      - Page-level authorization
      - Customizing FilterSecurityInterceptor
      - Exercise: Spring Security authentication example
      - Defining and working with ACLs
      - SpEL integration with Spring Security access control
      - Custom authorization
      - Using a custom PermissionEvaluator
    - Method-level security
      - Method-level security
      - Method-level security using SpEL

- Common coding errors and vulnerabilities
  - Improper error and exception handling
    - Typical problems with error and exception handling
    - Empty catch block
    - Overly broad throws
    - Overly broad catch

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- Using multi-catch
- Returning from finally block – spot the bug!
- Catching NullPointerException
- Exception handling – spot the bug!
- Exercise ScademyPay – Error handling

- Time and state problems
  - Time and state related problems
  - Concurrency – spot the bug!
  - Calling Thread.run()
  - Race condition in servlets – spot the bug!
  - Race condition – spot the bug!
  - ArrayList vs Vector
- 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
  - Serialization – spot the bug!
  - Exercise Serializable Sensitive
  - Immutable String – spot the bug!
  - Exercise Immutable Strings
  - Immutability and security

- Denial of service
  - DoS introduction
  - Asymmetric DoS
  - Regular expression DoS (ReDoS)
    - Exercise ReDoS
    - ReDoS mitigation
    - Case study – ReDos in Stack Exchange
  - Hashtable collision attack
    - Using hashtables to store data
    - Hashtable collision
    - Hashtable collision in Java

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- Security testing techniques and tools
  - General testing approaches
  - Source code review
    - Code review for software security
    - Taint analysis
    - Heuristic-based
    - Static code analysis
      - Static code analysis
      - Exercise – Using static code analysis tools
  - Testing the implementation
    - Manual vs. automated security testing
    - Web vulnerability scanners
      - Exercise – Using a vulnerability scanner
      - SQL injection tools
      - Exercise – Using SQL injection tools
- 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

REQUIREMENTS:

Advanced Java and Web application development

Difficulty level



CERTIFICATE:

The participants will obtain certificates signed by SCADEMY (course completion).

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

TRAINER:

Authorized SCADEMY Trainer

ADDITIONAL INFORMATION:

Training come with a number of easy-to-understand exercises providing live hacking fun. By accomplishing these exercises with the lead 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.

SCADEMY together with online application security educational platform AVATAO (more about AVATAO www.avatao.com) for each of participant SCADEMYs authorized training adds the 30 days business AVATAO trial holds the following package:

- 30-day customized free trial

Compendium Centrum Edukacyjne Spółka z o.o.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Postępu 18B, 02-676 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007