scademy
secure coding academy

Java EE and Web application security

CL-JWE | Classroom | 4 days

Variants: Java EE, Spring

Auxiliary topics: JMS/ActiveMQ, JSF, Vaadin

Audience: Java EE developers, software architects and testers
Preparedness: Advanced Java and Web application development
Exercises: Hands-on

As a developer, your duty is to write bulletproof code. However…

What if we told you that despite all of your efforts, the code you have been writing your entire career is full of weaknesses you never knew existed? What if, as you are reading this, hackers were trying to break into your code? How likely would they be to succeed?

This advanced course will change the way you look at code. A hands-on training during which we will teach you all of the attackers' tricks and how to mitigate them, leaving you with no other feeling than the desire to know more.

It is your choice to be ahead of the pack, and be seen as a game changer in the fight against cybercrime.

Outline:

- IT security and secure coding
- Web application security
- Client-side security
- Foundations of Java security
- Practical cryptography
- Java security services
- Secure communication in Java
- Security of Web services
- Java EE security
- Denial of service
- Common coding errors and vulnerabilities
- Principles of security and secure coding
- Knowledge sources

scademy
secure coding academy

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
Learn to use various security features of the Java development environment
Have a practical understanding of cryptography
Understand security concepts of Web services
Learn about JSON security
Understand security solutions of Java EE
Learn about denial of service attacks and protections
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

Related courses:

- CL-JWA - Java and Web application security (Classroom, 3 days)
- CL-JSM - Java and Web application security master course (Classroom, 5 days)
- CL-WSC - Web application security (Classroom, 3 days)
- CL-WTS - Web application security testing (Classroom, 3 days)
- CL-CJW - Combined C/C++, Java and Web application security (Classroom, 4 days)
- CL-JSM - Java and Web application security master course (Classroom, 5 days)

Note: Our classroom trainings 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
secure coding academy

Detailed table of contents

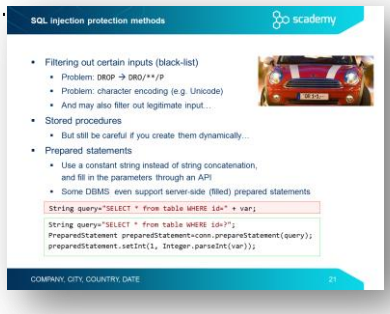
Day 1

IT security and secure coding

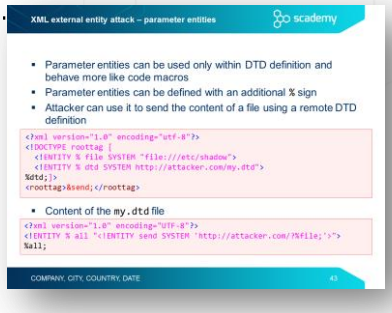
- Nature of security
- What is risk?
- IT security vs. secure coding
- From vulnerabilities to botnets and cybercrime
  - Nature of security flaws
  - Reasons of difficulty
  - From an infected computer to targeted attacks
  - The Seven Pernicious Kingdoms
  - OWASP Top Ten 2017

Web application security

- Injection
  - Injection principles
  - SQL injection
    - Exercise – SQL injection
    - Typical SQL Injection attack methods
    - Blind and time-based SQL injection
    - SQL injection protection methods
    - Effect of data storage frameworks on SQL injection in Java
  - 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
  - Cross site request forgery (CSRF)
    - CSRF prevention
    - CSRF prevention in Java frameworks



- 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
- Broken access control
  - Typical access control weaknesses
  - Insecure direct object reference (IDOR)
  - Exercise – Insecure direct object reference
  - Protection against IDOR
  - Case study – Facebook Notes
- Cross-Site Scripting (XSS)
  - Persistent XSS
  - Reflected XSS
  - DOM-based XSS
  - Exercise – Cross Site Scripting
  - 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

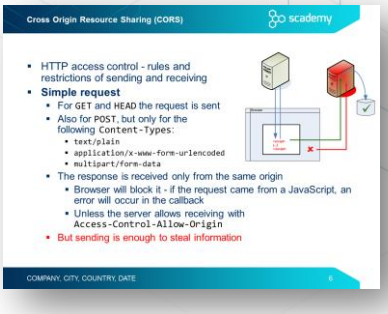


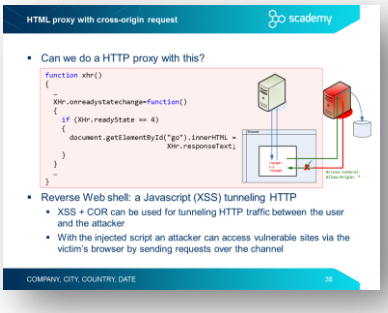
Day 2

Client-side security

- JavaScript security
- Same Origin Policy

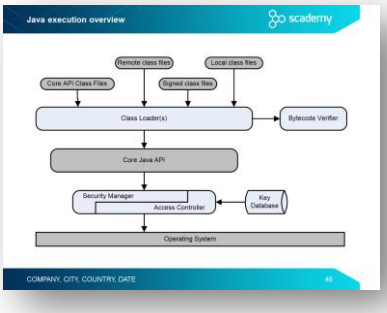
- Cross Origin Resource Sharing (CORS)
- Exercise – Client-side authentication
- Client-side authentication and password management
- Protecting JavaScript code
- Clickjacking
  - Exercise – Do you Like me?
  - Protection against Clickjacking
  - Anti frame-busting – dismissing protection scripts
  - 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





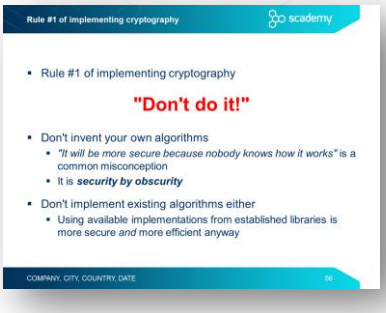
Foundations of Java security

- The Java environment
- 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

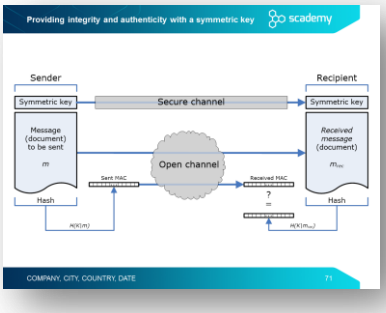


- Security Manager and Access Controller
- Permission checking
- Effects of doPrivileged

Practical cryptography



- Rule #1 of implementing cryptography.....
- Cryptosystems
  - Elements of a cryptosystem
- Symmetric-key cryptography
  - Providing confidentiality with symmetric cryptography
  - Symmetric encryption algorithms
  - 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

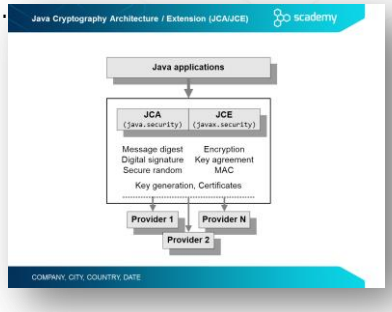


scademy
secure coding academy

Day 3

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



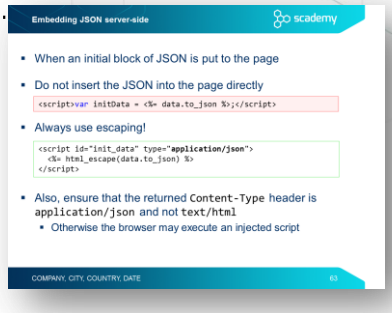
Secure communication in Java

- Security services
- SSL/TLS handshake
- Java Secure Socket Extension (JSSE)
- Exercise Https – Switching from HTTP to HTTPS
- Public Key Infrastructure support
  - The Java Keystore (JKS)
  - Java Certification Path (CertPath)

Security of Web services

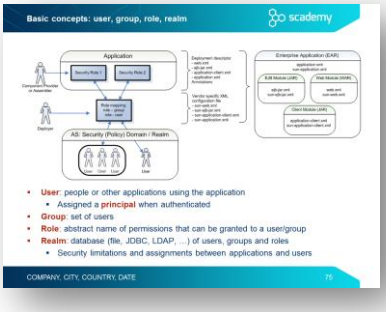
- Securing web services – two general approaches
- SOAP - Simple Object Access Protocol
- Security of RESTful web services
  - Authenticating users in RESTful web services
  - Authentication with JSON Web Tokens (JWT)
  - Authorization with REST
  - Vulnerabilities in connection with REST
- XML security
  - Introduction
  - XML parsing
  - XML injection
    - (Ab)using CDATA to store XSS payload in XML
    - Exercise – XML injection
    - Protection through sanitization and XML validation
    - XML bomb
    - Exercise – XML bomb

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



Java EE security

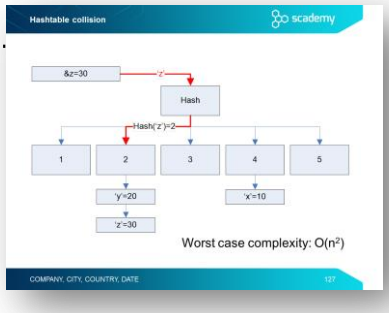
- Java EE multi-tier architecture
- Roles and responsibilities
- Java EE container-based security
- Basic concepts: user, group, role, realm.....
- Security of the Web/Presentation tier
  - Java EE authentication
  - HTTP basic/digest authentication
  - Form-based authentication
  - HTTPS client authentication
  - Java EE authorization
  - Declaring security roles
  - Checking the caller's identity programmatically
  - Security constraints
  - Exercise WebSecurity
  - 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



scademy
secure coding academy

Denial of service

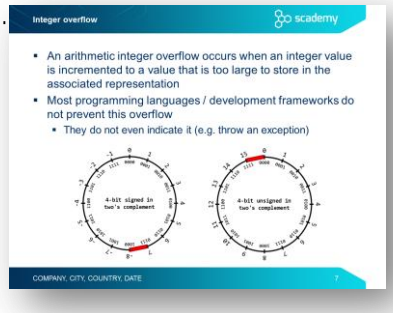
- DoS introduction
- Asymmetric DoS
- SSL/TLS renegotiation DoS
- Asymmetric DOS with JSON deserialization
- Regular expression DoS (ReDoS)
  - Exercise ReDoS
  - ReDoS mitigation
  - Case study – ReDos in Stack Exchange
- Hashtable collision attack
  - Using hashtables to store inputs
  - Hashtable collision
  - Hashtable collision in Java



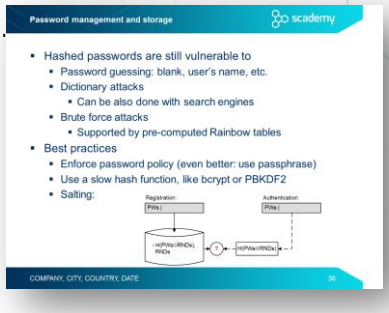
Day 4

Common coding errors and vulnerabilities

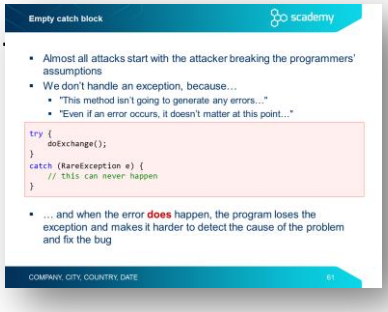
- Input validation
  - Input validation concepts
  - Integer problems
    - Representation of negative integers
    - Integer overflow
    - Exercise IntOverflow
    - What is the value of Math.abs(Integer.MIN_VALUE)?
    - Integer problem – best practices
  - Path traversal vulnerability
    - Path traversal – best practices
  - Unvalidated redirects and forwards
  - 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



- 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

scademy
secure coding academy

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