Szkolenie: SCADEMY

CL-NJS Node.js and Web application security

Cel szkolenia:

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? What if they could steal away your database and sell it on the black market?

This Web application security 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.

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 about Node.js security
- Have a practical understanding of cryptography
- Understand security concepts of Web services
- Learn about JSON security
- Learn about MongoDB security
- Learn about typical coding mistakes and how to avoid them
- Get information about some recent vulnerabilities in the Java framework
- Learn about denial of service attacks and protections
- Get sources and further readings on secure coding practices

Audience:

Web developers, architects, and testers

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.
Oddział w Warszawie
Nr konta: PL 20 1600 1462 1853 6451 3000 0001

Plan szkolenia:

- 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
  - Classification of security flaws
    - Landwehr’s taxonomy
    - 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
    - Other injection flaws
      - Command injection
      - Command injection exercise – starting Netcat
      - Case study – ImageMagick
  - Broken authentication
    - Session handling threats
    - Session handling best practices
    - Cross site request forgery (CSRF)
      - CSRF prevention
  - Sensitive data exposure
    - Sensitive data exposure
    - Transport layer security
  - XML external entity (XXE)
    - XML Entity introduction

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.
Oddział w Warszawie
Nr konta: PL 20 1600 1462 1853 6451 3000 0001

- XML external entity attack (XXE) – resource inclusion
- XML external entity attack – URL invocation
- XML external entity attack – parameter entities
- 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
  - Node.js encoding and escaping libraries
- Insecure deserialization
  - Serialization and deserialization basics
  - Security challenges of deserialization
  - Issues with deserialization – JSON

- 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 – Do you Like me?
    - Protection against Clickjacking
    - Anti frame-busting – dismissing protection scripts
    - Protection against busting frame busting
  - AJAX security

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.
Oddział w Warszawie
Nr konta: PL 20 1600 1462 1853 6451 3000 0001

- 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
  - Cross-origin requests
  - HTML proxy with cross-origin request
  - Exercise – Client side include

- Node.js security
  - Node.js introduction
  - Typical attack surface for a Node.js app
  - Node.js security architecture and features
    - Creating your Web server
    - Express.js security considerations
    - Using cookie-session and helmet to secure headers and cookies
    - Authentication
    - Authorization
    - Node Package Manager pitfalls
    - The rimrafall malicious package
    - The Node Security Project
  - Common coding mistakes in Node.js
    - Escaping and filtering difficulties
    - Node.js Buffer issues
    - Buffer information leakage in request v2.67.0 and earlier
    - String null termination problem in Node.js
    - Global namespace pollution – guess the output!
    - Eval is evil
    - Server side code injection
    - Dangerous JS constructs – spot the bug!
    - Using strict mode

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.
Oddział w Warszawie
Nr konta: PL 20 1600 1462 1853 6451 3000 0001

- Vulnerabilities in the Node.js framework
  - Hardcoded CA certificates
  - Abusing methodOverride
  - CVE-2013-7451...7454 – Bypassing XSS filters
  - CVE-2015-5380 – heap memory corruption in Unicode decoder
  - CVE-2016-3956 – npm bearer token vulnerability
- 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 number generation
      - Random numbers and cryptography
      - Cryptographically-strong PRNGs
      - Weak PRNGs in Node.js
      - 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
- Security of Web services
  - Securing web services – two general approaches

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.
Oddział w Warszawie
Nr konta: PL 20 1600 1462 1853 6451 3000 0001

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

- MongoDB security
  - MongoDB introduction
  - MongoDB security architecture and features
    - Authentication and access control
    - Document validation in MongoDB
    - Securing MongoDB communication via TLS
    - Secure configuration and hardening
  - Typical MongoDB security issues
    - NoSQL injection in MongoDB
    - Exercise – MongoDB NoSQL injection
    - Preventing NoSQL injection – Mongoose
    - Case studies: some past MongoDB weaknesses and vulnerabilities
      - Unsecured MongoDB instances exposed to the Internet
      - CVE-2014-3971 – Crash when processing malformed certificate
      - CVE-2015-1609 – Denial of service via Unicode string

- Common coding errors and vulnerabilities
  - Input validation
    - Input validation concepts
    - Integer problems
      - Representation of negative integers
      - Integer overflow
      - Integer issues in Javascript
      - Parsing integers in Javascript
      - Integer issues in Javascript – guess the output!
      - Integer problem – best practices

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.
Oddział w Warszawie
Nr konta: PL 20 1600 1462 1853 6451 3000 0001

- Path traversal vulnerability
  - Path traversal – weak protections
  - 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
  - Password management
    - Exercise – Weakness of hashed passwords
    - Password management and storage
    - Special purpose hash algorithms for password storage
    - Argon2 and PBKDF2 implementations for Node.js
    - bcrypt and scrypt implementations for Node.js
    - Case study – the Ashley Madison data breach
    - Typical mistakes in password management
- Improper error and exception handling
  - Typical problems with error and exception handling
  - Error and exception handling in Node.js
  - Empty catch block
  - Returning from finally block – spot the bug!

- 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
- 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

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.
Oddział w Warszawie
Nr konta: PL 20 1600 1462 1853 6451 3000 0001

- Vulnerability databases

Wymagania:

General Web application development

Poziom trudności



Certyfikaty:

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

Prowadzący:

Authorized SCADEMY Trainer

Informacje dodatkowe:

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.
Oddział w Warszawie
Nr konta: PL 20 1600 1462 1853 6451 3000 0001