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
- Have a practical understanding of cryptography
- Understand essential security protocols
- Understand some recent attacks against cryptosystems
- Understand security concepts of Web services
Learn about JSON 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

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 (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
    - Other injection flaws
      - Command injection
      - Case study – ImageMagick
  - A2 - Broken authentication
    - Session handling threats
    - Session handling best practices
Setting cookie attributes – best practices
Cross site request forgery (CSRF)
  Login CSRF
  CSRF prevention

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
  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
  Configuring the environment
  Insecure file uploads
  Exercise - Uploading executable files
  Filtering file uploads - validation and configuration

Web application security (OWASP Top Ten 2017)
A7 - Cross-Site Scripting (XSS)
  Persistent XSS
  Reflected XSS
  DOM-based XSS
  Exercise - Cross Site Scripting
  Exploitation: CSS injection
  Exploitation: injecting the tag
  XSS prevention

A8 - Insecure deserialization
• Serialization and deserialization basics
• Security challenges of deserialization
• Issues with deserialization - JSON

• 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
  • Intrusion detection systems and Web application firewalls

• 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
  • 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
Symmetric-key cryptography
Providing confidentiality with symmetric cryptography
Symmetric encryption algorithms
Modes of operation
Other cryptographic algorithms
Hash or message digest
Hash algorithms
SHAtered
Message Authentication Code (MAC)
Providing integrity and authenticity with a symmetric key
Random number generation
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
The RSA algorithm
Introduction to RSA algorithm
Encrypting with RSA
Combining symmetric and asymmetric algorithms
Digital signing with RSA
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 protocols
○ The TLS protocol
  ○ SSL and TLS
  ○ Usage options
  ○ Security services of TLS
  ○ SSL/TLS handshake
○ Protocol-level vulnerabilities
  ○ BEAST
  ○ FREAK
  ○ FREAK – attack against SSL/TLS
  ○ Logjam attack
○ Padding oracle attacks
  ○ Adaptive chosen-ciphertext attacks
  ○ Padding oracle attack
  ○ CBC decryption
  ○ Padding oracle example
  ○ Lucky Thirteen
  ○ POODLE
○ 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
○ XML Signature
  ○ XML Signature introduction
XML Signature structure
- Hash collision with XML Digital Signature
- XML canonicalization
- Signing XML documents – spot the bug!
- XML Signature Wrapping (XSW) attack
- XML Signature Wrapping – countermeasures

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
  - Brute forcing
  - Special purpose hash algorithms for password storage
  - Case study – the Ashley Madison data breach
  - Typical mistakes in password management

Insufficient anti-automation
- Captcha
- Captcha weaknesses

Denial of service
- DoS introduction
- Asymmetric DoS
- Case study – Denial-of-service against ICDs
  - Denial-of-service: battery drain
- 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
- 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](http://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