Szkolenie: SCADEMY
CL-IHEAL Secure coding master course for healthcare

<table>
<thead>
<tr>
<th>FORMA SZKOLENIA</th>
<th>MATERIAŁY SZKOLENIOWE</th>
<th>CENA</th>
<th>CZAS TRWANIA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stacjonarne</td>
<td>Tradycyjne</td>
<td>10750 PLN NETTO*</td>
<td>0 dzień</td>
</tr>
<tr>
<td>Stacjonarne</td>
<td>Cyfrowe</td>
<td>10750 PLN NETTO*</td>
<td>0 dzień</td>
</tr>
<tr>
<td>Stacjonarne</td>
<td>Tablet CTAB</td>
<td>11150 PLN NETTO*</td>
<td>0 dzień</td>
</tr>
<tr>
<td>Metoda dlearning</td>
<td>Tradycyjne</td>
<td>10750 PLN NETTO*</td>
<td>0 dzień</td>
</tr>
<tr>
<td>Metoda dlearning</td>
<td>Cyfrowe</td>
<td>10750 PLN NETTO*</td>
<td>0 dzień</td>
</tr>
<tr>
<td>Metoda dlearning</td>
<td>Tablet CTAB</td>
<td>10750 PLN NETTO*</td>
<td>0 dzień</td>
</tr>
</tbody>
</table>

* (+VAT zgodnie z obowiązującą stawką w dniu wystawienia faktury)

LOKALIZACJE

Kraków - ul. Tatarska 5, II piętro, godz. 9:00 - 16:00
Warszawa - ul. Bielska 17, godz. 9:00 - 16:00

Cel szkolenia:

The past few years have seen a massive increase in attacks, data breaches and medical identity theft targeting the healthcare industry; there have also been various ransomware attacks paralyzing healthcare computer networks as well as the various medical devices connected to them. The rise of mobile devices used in the industry needs to be addressed as well: there is a huge growth of medical software applications for mobiles and tablets that connect the patient with the organization – carrying and storing personally identifiable information (PII).

Healthcare is one of the business domains where security is absolutely crucial. Vulnerability is not an option when working with life-saving devices. There is also significant compliance pressure – if you want to stay a trusted and reliable vendor, your systems and applications need to comply with Health Information Portability and Accountability Act (HIPAA) requirements. To deal with these challenges, you need motivated secure coders with the right skills and the right attitude to fight security problems: a skilled team of software engineers as well as network administrators.

This training program exclusively targets engineers developing applications or maintaining networks for the healthcare sector. Our dedicated trainers share their experience and expertise through hands-on labs, and give real-life case studies from the healthcare industry – engaging participants in live hacking fun to reveal all consequences of insecure coding.

Participants attending this course will:

- Understand basic concepts of security, IT security and secure coding
Understand special threats in the healthcare sector
Understand regulations and standards
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 the requirements of secure communication
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 practical knowledge in using security testing techniques and tools
Get sources and further readings on secure coding practices

Audience:

Developers working in the healthcare sector

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

Special threats in the healthcare sector
  Threats in healthcare - trends and numbers
Attacker model
Most significant targets
Industry and regulatory response to threats
How is cybersecurity different for medical devices?
Attacker tools and vectors

Regulations and standards
HIPAA
What is HIPAA?
Amendments
Who needs to be regulated by HIPAA?
General safety requirements
Implementation requirements
Administrative safeguards
Physical safeguards
Technical safeguards

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

○ Web application security (OWASP Top Ten 2017)
  ○ A5 - Broken access control
    ○ Typical access control weaknesses
    ○ Insecure direct object reference (IDOR)
    ○ Exercise – Insecure direct object reference
    ○ Protection against IDOR
    ○ Case study – Molina Healthcare
      ○ Exposed patient records
    ○ Case study – Facebook Notes
  ○ A6 - 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
    ○ 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
  - JavaScript usage
  - JavaScript Global Object
  - Dangers of JavaScript
  - Exercise – Client-side authentication
  - Client-side authentication and password management
  - Protecting JavaScript code
  - Exercise – JavaScript obfuscation
  - 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
  - MySpace worm
  - AJAX security guidelines

- HTML5 security
  - New XSS possibilities in HTML5
  - Client-side persistent data storage
  - 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

- Security architecture
  - (platform and technology dependent topics)
  - Application level access control
    - (permissions, sandboxing)
  - User level access control
    - Authentication
    - Authorization

- Requirements of secure communication
  - Security levels
  - Secure acknowledgment
    - Malicious message absorption
      - Feasibility of secure acknowledgment
      - The solution: Clearing Centers
    - Inadvertent message loss
  - Integrity
    - Error detection - Inadvertent message distortion (noise)
      - Modeling message distortion
      - Error detection and correction codes
    - Authenticity - Malicious message manipulation
      - Modeling message manipulation
      - Practical integrity protection (detection)
  - Non-repudiation
    - Non-repudiation
  - Summary
    - Detecting integrity violation

- Confidentiality
  - Model of encrypted communication
  - Encryption methods in practice
  - Strength of encryption algorithms

- Remote identification
- Requirements of remote identification
  - Anonymity and traffic analysis
    - Model of anonymous communication
    - Traffic analysis
    - Theoretically strong protection against traffic analysis
    - Practical protection against traffic analysis
  - Summary
    - Relationship between the requirements

- 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

Crypto libraries and APIs

Security protocols
- The TLS protocol
  - SSL and TLS
  - Usage options
  - Security services of TLS
  - SSL/TLS handshake
- Protocol-level vulnerabilities
  - BEAST
- Padding oracle attacks
  - Adaptive chosen-ciphertext attacks
  - Padding oracle attack
  - CBC decryption
  - Padding oracle example
  - POODLE

Input validation
- Input validation concepts
- Integer problems
  - Representation of negative integers
  - Integer overflow
  - Integer problem – best practices
    - 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
  - Case study – B. Braun SpaceCom
    - Space: an infusion pump management system
    - CVE-2017-6018: Open redirect issue in SpaceCom module
- Log forging
  - Some other typical problems with log files
  - (some additional platform and technology dependent topics)
○ 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
    ○ Embedding JSON server-side
    ○ JSON injection
    ○ JSON hijacking
    ○ Case study – XSS via spoofed JSON element
  ○ 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
    ○ Case study – the Ashley Madison data breach
      ○ The loginkey token
      ○ Revealing the passwords with brute forcing
    ○ Typical mistakes in password management
      ○ Sensitive info in memory - minimize the attack surface
  ○ Object-relational mapping (ORM) security
  ○ Improper error and exception handling
    ○ Typical problems with error and exception handling
- Time and state problems
- Code quality problems
- Denial of service
  - DoS introduction
  - Asymmetric DoS
  - Case study - Denial-of-service against ICDs
    - Denial-of-service: battery drain
    - Denial-of-service: RF crash
  - Case study - ReDos in Stack Exchange
- Hashtable collision attack
  - Using hashtables to store data
  - Hashtable collision
- 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
  - Testing the implementation
    - Manual vs. automated security testing
    - Penetration testing
    - Stress tests
    - Fuzzing
      - Automated security testing - fuzzing
      - Challenges of fuzzing
  - Proxy servers and sniffers
    - Testing with proxies and sniffers
    - Packet analyzers and proxies
    - Exercise - Testing with proxy
    - Proxying HTTPS traffic
    - Case study - The Lenovo Superfish incident
- 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
- Healthcare cybersecurity resources

Wymagania:
Advanced desktop and 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