Training: SCADEMY
CL-ONF OWASP Top 10, C# Secure Coding Fundamentals

TRAINING GOALS:

Writing .NET web applications 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 C# 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 give you a holistic view on the security aspects of the .NET framework – such as making use of cryptography or Code Access Security – as well as common C# programming mistakes you need to be aware of. We also cover typical Web vulnerabilities with a focus on how they affect ASP.NET web apps on the entire stack – from the CLR to modern AJAX and HTML5-based frontends. 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
- Learn about typical coding mistakes and how to avoid them
- Get sources and further readings on secure coding practices

Audience:

- Web developers

CONSPECT:

- IT security and secure coding
  - Nature of security
  - What is risk?

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

- IT security vs. secure coding
- From vulnerabilities to botnets and cybercrime
  - Nature of security flaws
  - 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
    - Other injection flaws
      - Command injection
      - Command injection exercise – starting Netcat
    - HTTP parameter pollution
      - Cookie injection / HTTP parameter pollution
      - Exercise – Value shadowing
    - Broken authentication
      - Session handling threats
      - Session fixation
      - Exercise – Session fixation
      - Session handling best practices
      - Setting cookie attributes – best practices
      - Cross site request forgery (CSRF)
        - CSRF prevention
      - 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

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

- 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
  - Output encoding API in C#
  - XSS protection in ASP.NET – validateRequest
- 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
- Insecure deserialization
  - Serialization and deserialization basics
  - Security challenges of deserialization
  - Deserialization in .NET
  - From deserialization to code execution
  - POP payload targeting MulticastDelegate (C#)
  - Real-world .NET deserialization vulnerabilities
  - Issues with deserialization – JSON
  - Best practices against deserialization vulnerabilities
  - Case study – Windows PowerShell RCE
    - CVE-2017-8565 – Windows PowerShell RCE
    - Example XML triggering the RCE
  - Using components with known 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

- Vulnerability attributes
- Common Vulnerability Scoring System – CVSS
- Insufficient logging and monitoring
- Detection and response
- Logging and log analysis
- Intrusion detection systems and Web application firewalls

- 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(int.MinValue)?
- Integer problem – best practices
- 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 in .NET
- bcrypt and scrypt implementations in .NET
- Case study – the Ashley Madison data breach
- Typical mistakes in password management
- Exercise – Hard coded passwords
- Accessibility modifiers
- Accessing private fields with reflection in .NET
- Exercise Reflection – Accessing private fields with reflection
- Improper error and exception handling

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

        - Typical problems with error and exception handling
        - Empty catch block
        - Overly broad catch
        - Using multi-catch
        - Catching NullReferenceException
        - Exception handling – spot the bug!
        - Exercise – Error handling
    - Time and state problems
        - Concurrency and threading
        - Concurrency in .NET
        - Omitted synchronization – spot the bug!
        - Exercise – Omitted synchronization
        - Incorrect granularity – spot the bug!
        - Exercise – Incorrect granularity
        - Deadlocks
        - Avoiding deadlocks
        - Exercise – Avoiding deadlocks
        - Lock statement
    - Code quality problems
        - Dangers arising from poor code quality
        - Poor code quality – spot the bug!
        - Unreleased resources
        - Serialization – spot the bug!
        - Exercise – Serializable sensitive
        - Private arrays – spot the bug!
        - Private arrays – typed field returned from a public method
        - Class not sealed – object hijacking
        - Exercise – Object hijacking
        - Immutable string – spot the bug!
        - Exercise – Immutable strings
        - Using SecureString
- 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.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- Vulnerability databases
- .NET secure coding guidelines at MSDN
- .NET secure coding cheat sheets
- Recommended books – .NET and ASP.NET

REQUIREMENTS:

General C# and Web application development

Difficulty level



CERTIFICATE:

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

ADDITIONAL INFORMATION:

Related courses:

- CL-ANS Secure desktop application development in C#
- CL-NSM C# and Web application security master course
- CL-WSC Web application security
- CL-WTS Web application security testing

Note: 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.

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