Training: SCADEMY
CL-PYS Python security

TRAINING GOALS:

The Python language is used in many different settings – from command-line tools to complex Web applications. Many of these Python programs are exposed to attack, either by being directly accessible through the Internet or by directly processing user-provided data in a server environment. Developers must therefore be extremely cautious in how to use different technologies securely, and should also have a deep understanding in secure coding techniques and potential pitfalls.

This course covers the most critical security issues in Python applications. We cover vulnerabilities from the OWASP Top Ten list for the web as they concern Python web applications as well as the Django framework. The course also encompasses the most significant security issues for Python code in general (including many Python-specific issues such as function hijacking), while also presenting security solutions provided by the Python ecosystem – such as authentication, access control and encryption.

Understanding the security solutions provided by Python as well as the various security issues and vulnerabilities is a must for all programmers using these technologies to develop web, desktop or server applications.

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
- Understand security concepts of Web services
- Learn about JSON security
- Learn about Python security architecture
- Have a practical understanding of cryptography
- Learn about typical coding mistakes and how to avoid them
- Learn about denial of service attacks and protections
- Get sources and further readings on secure coding practices

Audience:

Python 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.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

CONSPECT:

- 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 (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
      - Database access in Python
      - ORM libraries in Python
    - Other injection flaws
      - Command injection
      - Command injection exercise – starting Netcat
      - Case study – ImageMagick
  - A2 - Broken authentication
    - Session handling threats
    - Session handling best practices
    - Sessions in Django
    - Additional cookie security considerations
    - Setting cookie attributes – best practices
    - Cross site request forgery (CSRF)
      - CSRF prevention
      - CSRF prevention in Django
  - A4 - XML external entity (XXE)

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

- 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

- 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
- A7 - Cross-Site Scripting (XSS)
  - Persistent XSS
  - Reflected XSS
  - DOM-based XSS
  - Exercise – Cross Site Scripting
  - XSS prevention
  - XSS prevention in Python
- A8 - Insecure deserialization
  - Serialization and deserialization basics
  - Security challenges of deserialization
  - Security issues when using Pickle
  - Code injection via overriding __reduce__ in Pickle
  - Code injection via YAML deserialization
  - Issues with deserialization – JSON

- Client-side security
  - JavaScript security
  - Same Origin Policy
  - Simple requests
  - Preflight requests
  - Exercise – Same-Origin Policy
  - JavaScript usage
  - JavaScript Global Object
  - Dangers of JavaScript

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

- 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

- XML security
  - Introduction
  - XML parsing
  - XML parsing in Python
  - XML bomb
  - Exercise – XML bomb
  - JSON security
    - Introduction
    - Embedding JSON server-side
    - JSON injection
    - JSON hijacking

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

- Case study – XSS via spoofed JSON element

- Python security architecture
  - Python architecture
  - Python applications and their attack surfaces
  - Authentication and authorization
    - Authentication in Python
    - Authorization in Python
    - Authentication and authorization in Django
    - Authentication and authorization in Flask
  - Code protection in Python
    - Python bytecode
    - Obfuscation
    - Modifying the Python runtime
    - Weaknesses in the techniques
    - Other protection methods
    - Related security issues
    - Sandboxing

- Practical cryptography
  - Rule #1 of implementing cryptography
  - Cryptosystems
    - Elements of a cryptosystem
    - Cryptographic libraries in Python – overview
  - Symmetric-key cryptography
    - Providing confidentiality with symmetric cryptography
    - Symmetric encryption algorithms
    - Modes of operation
    - Symmetric encryption
  - Other cryptographic algorithms
    - Hash or message digest
    - Hash algorithms
    - SHAttered
    - Hashing
    - Message Authentication Code (MAC)
    - Providing integrity and authenticity with a symmetric key
    - Random number generation
      - Random numbers and cryptography

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

- Cryptographically-strong PRNGs
- Weak PRNGs in Python
- Random numbers
- 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
    - Asymmetric encryption
- 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

- Common coding errors and vulnerabilities
  - Input validation
    - Input validation concepts
    - Integer problems
      - Representation of negative integers
      - Integer overflow
      - Integers in Python
      - Integer problems in Python
      - Exercise IntOverflow
    - Path traversal vulnerability
      - Path traversal vulnerability
      - Path traversal – weak protections
      - Path traversal – best practices
      - Path traversal mitigation
    - Unvalidated redirects and forwards
      - Redirects in Django
    - Log forging
      - Some other typical problems with log files

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

- Executing user-controlled code in Python
  - Dangerous code execution – eval() and exec()
  - Dangerous code execution – input()
- String formatting issues in Python
  - Format string problems in Python
  - Information leakage – can you guess the output?

- 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
    - PBKDF2 and scrypt implementations in Python
    - Argon2 and bcrypt implementations in Python
    - Case study – the Ashley Madison data breach
    - Typical mistakes in password management
  - Dangers of reflection in Python
    - Python introspection and reflection – features and risks
    - Dynamic loading
    - Module injection
    - Monkey patching
    - Monkey patching example
    - Function hijacking
    - Exercise – Module injection in Python

- Improper error and exception handling
  - Typical problems with error and exception handling
  - Exception handling in Python
  - Empty except block
  - Overly broad except
  - Using multi-except
  - Returning from finally block – spot the bug!
  - Exercise ErrorHandling – spot the bug!
  - Exercise – Error handling
  - Relying on assertions for error checking – spot the bug!
- Time and state problems
  - Concurrency and threading

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

- Concurrency issues in Python
  - Concurrency modules in Python
  - The threading module
  - Synchronization options
  - The Global Interpreter Lock
  - Performance and the GIL
  - GIL management
  - Bypassing the GIL
  - A quote from Guido van Rossum
  - Pools
  - Exercise – GIL performance
  - Exercise – Concurrency issues
- Time-of-check-to-time-of-use (TOCTTOU)
  - Serialization errors
  - Preventing file I/O TOCTTOU in Python
  - Exercise - TOCTTOU

- Code quality problems
  - Dangers arising from poor code quality
  - Immutability
    - Immutability
    - Immutability – guess the output!
  - Context managers
    - Resource management in Python
    - Releasing resources – spot the bug!
    - An even better solution
    - Behind the with statement
    - @contextmanager decorator – spot the bug!
    - @contextmanager decorator – handling error

- Denial of service
  - DoS introduction
  - Asymmetric DoS
  - Regular expression DoS (ReDoS)
    - Exercise ReDoS
    - ReDoS mitigation
    - Case study – ReDos in Stack Exchange
  - Hashtable collision attack

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

        - Using hashtables to store data
        - Hashtable collision
        - Hash tables in Python
    - 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
        - Python secure coding resources
        - Recommended books – Python security

REQUIREMENTS:

General Python development

Difficulty level



CERTIFICATE:

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

TRAINER:

Authorized SCADEMY Trainer

ADDITIONAL INFORMATION:

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:

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

- 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.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007