Training: SCADEMY
CL-ITELC Secure coding master course for telecoms

TRAINING GOALS:

Telecommunication is the foundation on which critical infrastructures are built – all modern technologies heavily rely on its security. The continued adoption of IP-based telecommunication and the exposure of telecom equipment opens additional paths for attackers, with a massive increase in security incidents against telecom systems.

The security of customer premises equipment and the prevalence of Internet of Things (IoT) devices is especially critical, as these devices are directly exposed to attackers and potentially resulting in large-scale attacks against users and companies alike. IoT security is recognized by the industry as a critical area of telecommunications, as evidenced by the efforts of the GSM Alliance to produce a set of IoT security guidelines for network operators.

The threats against telecom systems run the gamut from simple-yet-effective DDoS attacks to large-scale exploitation of vulnerabilities in communication equipment. Ultimately, the best way 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 and network engineers.

This training program exclusively targets engineers developing software applications or network equipment for the healthcare sector. Our dedicated trainers share their experience and expertise through hands-on labs, and give real-life case studies from the telecom industry – engaging participants in hands-on labs to realize the harsh consequences of insecure coding.

Our dedicated trainers share their experience and expertise through hands-on labs, and give real-life case studies from the telecom industry – engaging participants in hands-on labs to realize the harsh consequences of insecure coding.

Participants attending this course will:

- Understand basic concepts of security, IT security and secure coding
- Understand special threats in the telecom 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
- Learn about network attacks and defenses at different OSI layers
- Learn about typical coding mistakes and how to avoid them

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

- Get information about some recent vulnerabilities in the Java framework
- Learn to use various security features of the Java development environment
- Understand security concepts of Web services
- Realize the severe consequences of unsecure buffer handling in native code
- Understand the architectural protection techniques and their weaknesses
- Realize the severe consequences of unsecure buffer handling
- Learn about denial of service attacks and protections
- Get sources and further readings on secure coding practices

Audience:

Developers working in the telecom sector

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
- Special threats in the telecom sector
  - Threats in telecom
  - Attacker model
  - Attack response
- Regulations and standards
  - ITU-T X.805 RecommendationSecurity architecture for systems providing end-to-end communications
    - Overview
    - Security dimensions
    - Security layers
    - Security planes
    - Architecture
    - Security threats (X.800)
  - GSMA IoT Security Guidelines for Network Operators
    - Overview and scope

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

- Network security considerations for IoT
- Service security considerations for IoT
- Technical Guideline on Security measures for Article 4 and Article 13a<
  - D1-2
  - D3-4
  - D5-7

- 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
      - Case study – ImageMagick
  - XML external entity (XXE)
    - XML Entity introduction
    - XML external entity attack (XXE) – resource inclusion
    - Exercise – XXE attack
    - Preventing entity-related attacks
    - 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
  - Cross-Site Scripting (XSS)
    - Persistent XSS
    - Reflected XSS
    - DOM-based XSS
    - Exercise – Cross Site Scripting
    - XSS prevention
  - Insecure deserialization

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

- Serialization and deserialization basics
- Security challenges of deserialization
- Deserialization in Java
- Denial-of-service via Java deserialization
- From deserialization to code execution
- POP payload targeting InvokerTransformer (Java)
- Real-world Java deserialization vulnerabilities
- Issues with deserialization – JSON
- Best practices against deserialization vulnerabilities

- Client-side security
  - JavaScript security
  - Same Origin Policy
  - Simple requests
  - Preflight requests
  - Clickjacking
    - Clickjacking
    - Exercise – IFrame, Where is My Car?
    - Protection against Clickjacking
    - Anti frame-busting – dismissing protection scripts
    - Protection against busting frame busting

- Practical cryptography
  - Rule #1 of implementing cryptography
  - Cryptosystems
    - Elements of a cryptosystem
    - Java Cryptography Architecture / Extension (JCA/JCE)
    - Using Cryptographic Service Providers
  - Symmetric-key cryptography
    - Providing confidentiality with symmetric cryptography
    - Symmetric encryption algorithms
    - Modes of operation
    - Symmetric encryption with OpenSSL: encryption
    - Symmetric encryption with OpenSSL: decryption
    - Private (symmetric) key cryptography in Java
  - Other cryptographic algorithms
    - Hash or message digest
    - Hash algorithms

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

- SHAttered
- Hashing with OpenSSL
- Hashing in Java: MessageDigest class
- MAC and password-based encryption in Java: Mac class
- 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 C and C++
  - Stronger PRNGs in C
  - Generating random numbers with OpenSSL
  - Weak and strong PRNGs in Java
  - Hardware-based TRNGs
  - Exercise RandomTest
  - Using random numbers in Java – spot the bug!
- 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 with OpenSSL
    - Digital signatures with OpenSSL
    - Exercise Sign
- 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
  - The Java Keystore (JKS)
  - Java Certification Path (CertPath)
- Network security
  - Overview

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

- The TCP/IP stack
- Data Link layer
  - Sniffing attacks
    - What is a sniffer?
    - A revision on hubs and switches
    - MAC flooding
  - Spoofing
    - Spoofing attacks
    - Address Resolution Protocol (ARP)
    - ARP spoofing
    - Dynamic Host Configuration Protocol (DHCP)
    - DHCP starvation
  - Man-in-the-Middle
    - Man-in-the-Middle
    - Man-in-the-Middle with ARP poisoning
    - Rogue DHCP server
  - Attacks against VLANs
    - VLANs, Native VLANs, DTP
    - VLAN hopping, Switch spoofing
    - Double tagging
  - Data Link layer protections
    - Segmentation
    - Detecting sniffing tools
    - VLAN security
    - Port Security
    - DHCP snooping
    - Dynamic ARP Inspection (DAI)
    - Private VLANs
- Network layer
  - IP address spoofing
  - Maximum Transmission Unit
  - Fragmentation attack
  - ICMP attacks
    - Internet Control Message Protocol (ICMP)
    - Smurf attack
    - Ping of death

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

- Route hijacking
- Network layer protections
  - Ingress filtering, Egress filtering
  - IP Source Guard
  - Firewalls
  - Packet filtering firewalls
  - Intrusion Detection/Prevention Systems
- Transport layer
  - Transmission Control Protocol (TCP)
    - Transmission Control Protocol
    - SYN flood
    - TCP session hijacking
  - User Datagram Protocol (UDP)
    - User Datagram Protocol
    - UDP flooding
  - Routing protocols
    - Routing protocols
  - Fingerprinting and service detection
    - connect() scan
    - SYN scan
    - FIN scan
    - X-mas scan
  - Transport layer protection
    - SYN proxy
    - SYN cookies
    - Stateful firewalls
    - Routing protocol security
- 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
      - Special purpose hash algorithms for password storage
      - Argon2 and PBKDF2 implementations in C/C++
      - bcrypt and scrypt implementations in C/C++

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

- Argon2 and PBKDF2 implementations in Java
- bcrypt and scrypt implementations in Java
- Case study – the Ashley Madison data breach
- Typical mistakes in password management
- Exercise – Hard coded passwords

- Accessibility modifiers
  - Accessing private fields with reflection in Java
  - Exercise Reflection – Accessing private fields with reflection
- Exercise ScademyPay – Integrity protection weakness

- Foundations of Java security
  - The Java environment
  - Low-level security – the Java language and environment
    - Java language security
    - Type safety
    - Automatic memory management
    - Java execution overview
    - Bytecode Verifier
    - Class Loader
    - Protecting Java code
  - High-level security – access control
    - Protection domains
    - Security Manager and Access Controller
    - Permission checking
    - Effects of doPrivileged
    - Allowing untrusted code to load arbitrary classes
  - Code signing in Java
    - Code signing
    - Code signing considerations and best practices
    - Package sealing
    - Exercise Jars – Code signing
- Secure communication in Java
- Java security services
  - Java security services – architecture
  - XML security
    - Introduction
    - XML parsing

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 injection
  - Exercise – XML injection
  - Protection through sanitization and XML validation
  - XML parsing in C++
  - Exercise – XML bomb
  - XML bomb

- x86 machine code, memory layout and stack operations
  - Intel 80x86 Processors – main registers
  - Intel 80x86 Processors – most important instructions
  - Intel 80x86 Processors – flags
  - Intel 80x86 Processors – control instructions
  - Intel 80x86 Processors – stack handling and flow control
  - The memory address layout
  - The function calling mechanism in C/C++ on x86
  - Calling conventions
  - The local variables and the stack frame
  - Function calls – prologue and epilogue of a function
  - Stack frame of nested calls
  - Stack frame of recursive functions
- Buffer overflow
  - Stack overflow
    - Buffer overflow on the stack
    - Overwriting the return address
    - Exercises – introduction
    - Exercise BOFIntro
    - Exercise BOFShellcode
  - Protection against stack overflow
    - Specific protection methods
    - Protection methods at different layers
    - The protection matrix of software security
    - Stack overflow – Prevention (during development)
    - Stack overflow – Detection (during execution)
    - Fortify compiler option (FORTIFY_SOURCE)
    - Exercise BOFShellcode – Using the Fortify compiler option
  - Stack smashing protection
    - Stack smashing protection variants

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

- Stack smashing protection in GCC
- Exercise BOFShellcode – Stack smashing protection
- Effects of stack smashing protection

- Address Space Layout Randomization (ASLR)
  - Randomization with ASLR
  - Practical weaknesses and limitations to ASLR
  - Circumventing ASLR: NOP sledding
- Non executable memory areas – the NX bit
  - Access control on memory segments
  - The Never eXecute (NX) bit
- Return-to-libc attack – Circumventing the NX bit protection
  - Circumventing memory execution protection
  - Return-to-libc attack
- Return oriented programming (ROP)
  - Exploiting with ROP
  - ROP gadgets

- Some additional native code-related vulnerabilities
  - Boundary violation
    - Array indexing – spot the bug!
    - Off-by-one and other null termination errors
    - The Unicode bug
  - Code quality problems
    - Poor code quality – spot the bug!
    - Unreleased resources
    - Type mismatch – Spot the bug!
    - Exercise TypeMismatch
    - Memory allocation problems
      - Smart pointers
      - Zero length allocation
      - Double free
      - Mixing delete and delete[]
  - Race condition
    - Serialization errors
    - Exercise TOCTTOU
  - Case study - the Shellshock bash vulnerability
    - Shellshock – basics of using functions in bash

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

- Shellshock – vulnerability in bash
- Exercise - Shellshock
- Shellshock fix and counterattacks
- Exercise – Command override with environment variables

- Common coding errors and vulnerabilities
  - Input validation
    - Input validation concepts
    - Integer problems
      - Representation of negative integers
      - Integer ranges
      - Integer overflow
      - Integer problems in C/C++
      - The integer promotion rule in C/C++
      - Arithmetic overflow – spot the bug!
      - Exercise IntOverflow
      - What is the value of Math.abs(Integer.MIN_VALUE)?
      - Signedness bug – spot the bug!
      - Integer truncation – spot the bug!
      - Integer problem – best practices
      - Case study – Android Stagefright
    - Path traversal vulnerability
      - Path traversal – weak protections
      - Path traversal – best practices
    - Unvalidated redirects and forwards
    - Unsafe native calls
      - Unsafe JNI
      - Exercise Unsafe JNI
    - Log forging
      - Some other typical problems with log files
  - Improper error and exception handling
    - Typical problems with error and exception handling
    - Empty catch block
    - Overly broad throws
    - Overly broad catch
    - Using multi-catch
    - Returning from finally block – spot the bug!

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

- Catching NullPointerException
- Exception handling – spot the bug!
- Exercise ScademyPay – Error handling
- Exercise – Error handling
- Case study – "e;#iamroot"e; authentication bypass in macOS
  - Authentication process in macOS (High Sierra)
  - Incorrect error handling in opendirectoryd
  - The #iamroot vulnerability (CVE-2017-13872)

- Code quality problems
  - Dangers arising from poor code quality
  - Private arrays – spot the bug!
  - Private arrays – typed field returned from a public method
  - Exercise Object Hijack
  - Public method without final – object hijacking
  - Serialization – spot the bug!
  - Exercise Serializable Sensitive
  - Immutable String – spot the bug!
  - Exercise Immutable Strings
  - Immutability and security

- 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
    - Hashtable collision in Java

- 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

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

- Java secure coding sources
- Recommended books – C/C++
- Recommended books – Java

REQUIREMENTS:

Advanced Java, Web and C/C++ 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:

- 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