scademy
secure coding academy

Secure coding for healthcare

CL-VHEAL | Classroom | 3 days

Audience: Developers working in the healthcare sector
Preparedness: General desktop and Web application development
Exercises: Hands-on

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.

Outline:

- IT security and secure coding
- Special threats in the healthcare sector
- Regulations and standards
- Web application security (OWASP Top Ten 2017)
- Client-side security
- XML security
- JSON security
- Denial of service
- Practical cryptography
- Security protocols
- Principles of security and secure coding

Knowledge sources

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
Learn about JSON security
Learn about denial of service attacks and protections
Have a practical understanding of cryptography
Understand essential security protocols
Get sources and further readings on secure coding practices

Related courses:

- CL-JWA - Java and Web application security (Classroom, 3 days)
- CL-JWE - Java EE and Web application security (Classroom, 4 days)
- CL-ANS - Secure desktop application development in C# (Classroom, 3 days)
- CL-NWA - C# and Web application security (Classroom, 3 days)
- CL-JNW - Combined Java, C# and Web application security (Classroom, 3 days)
- CL-PYS - Python security (Classroom, 3 days)
- CL-WSC - Web application security (Classroom, 3 days)
- CL-WTS - Web application security testing (Classroom, 3 days)

Note: Our classroom trainings 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.

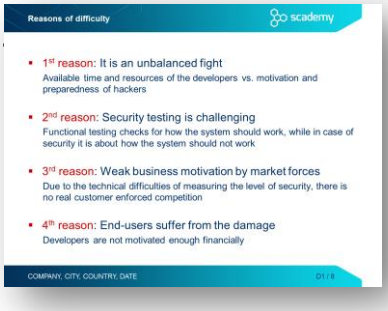
scademy
secure coding academy

Detailed table of contents

Day 1

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

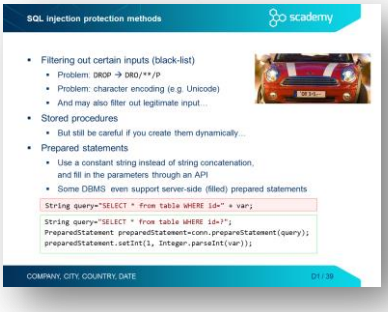
Regulations and standards

- HIPAA
  - What is HIPAA?
  - Amendments
  - Who needs to be regulated by HIPAA?
  - General safety requirements
  - Implementation requirements

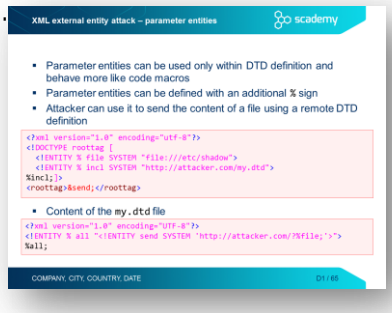
scademy
secure coding academy

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 .....
    - Effect of data storage frameworks on SQL injection
  - 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 – Molina Healthcare
    - Exposed patient records
  - Case study – Facebook Notes

scademy
secure coding academy

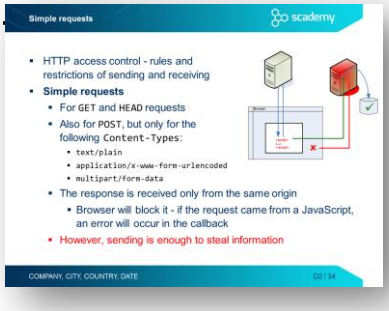
Day 2

Web application security (OWASP Top Ten 2017)

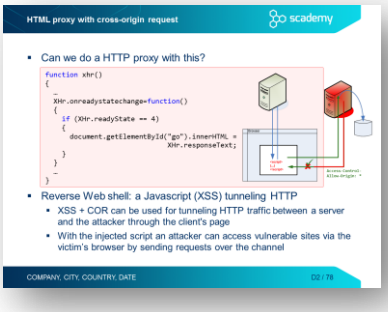
- A6 - Security misconfiguration
  - Configuring the environment
  - Insecure file uploads
  - Exercise – Uploading executable files
  - Filtering file uploads – validation and configuration
- A7 - Cross-Site Scripting (XSS)
  - Persistent XSS
  - Reflected XSS
  - DOM-based XSS
  - Exercise – Cross Site Scripting
  - XSS prevention
- A8 - Insecure deserialization
  - 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

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

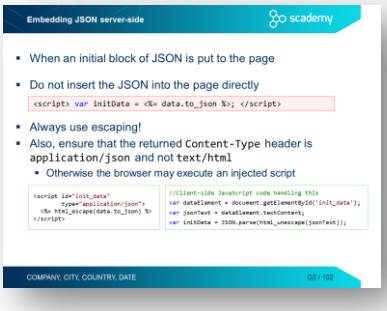


XML security

- 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

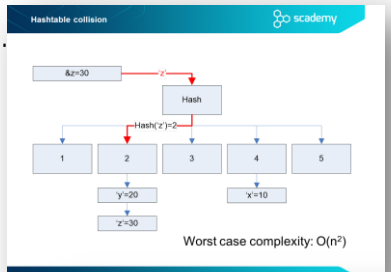
- Introduction
- Embedding JSON server-side.....



- JSON injection
- JSON hijacking
- Case study – XSS via spoofed JSON element

Denial of service

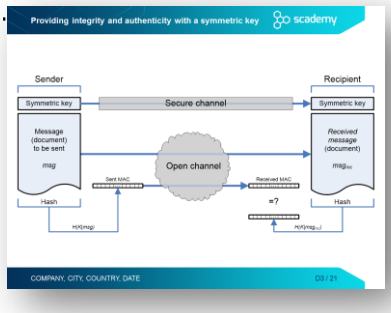
- DoS introduction
- Asymmetric DoS
- Case study – Denial-of-service against ICDs
  - Denial-of-service: battery drain
  - Denial-of-service: RF crash
- Case study – ReDos in Stack Exchange
- Hashtable collision attack
  - Using hashtables to store data
  - Hashtable collision



Day 3

Practical cryptography

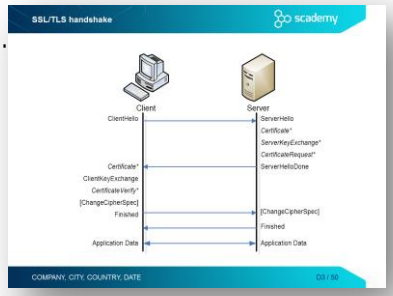
- Rule #1 of implementing cryptography
- Cryptosystems
  - Elements of a cryptosystem
- Symmetric-key cryptography
  - Providing confidentiality with symmetric cryptography
  - Symmetric encryption algorithms
  - Modes of operation
- Other cryptographic algorithms
  - Hash or message digest
  - Hash algorithms
  - SHAttered
  - Message Authentication Code (MAC)
  - Providing integrity and authenticity with a symmetric key
  - 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



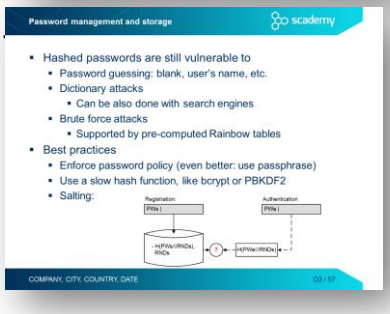
- The RSA algorithm
  - Introduction to RSA algorithm
  - Encrypting with RSA
  - Combining symmetric and asymmetric algorithms
  - Digital signing with RSA
- Public Key Infrastructure (PKI)
  - Man-in-the-Middle (MitM) attack
  - Digital certificates against MitM attack
  - Certificate Authorities in Public Key Infrastructure
  - X.509 digital certificate

Security protocols

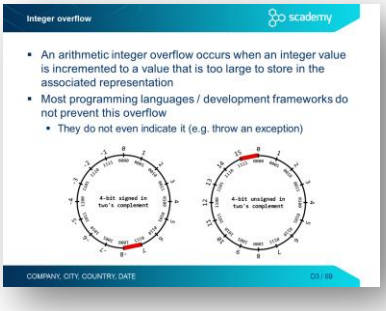
- Secure network protocols
- Specific vs. general solutions
- SSL/TLS protocols
  - Security services
  - SSL/TLS handshake .....



- Improper use of security features
  - Typical problems related to the use of security features
  - Insecure randomness
  - 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
    - Typical mistakes in password management
- Input validation
  - Input validation concepts
  - Integer problems
    - Representation of negative integers
    - Integer overflow .....



    - Integer problem – best practices
  - Path traversal vulnerability
    - Path traversal – best practices
    - Case study – Insufficient URL validation in LastPass
  - Unvalidated redirects and forwards
    - Case study – B. Braun SpaceCom
  - Log forging
    - Some other typical problems with log files

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