scademy
secure coding academy

Secure desktop application development in C#

CL-ANS | Onsite / Virtual classroom | 3 days

Audience: C# desktop application developers, software architects and testers
Preparedness: General C# 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
- Common coding errors and vulnerabilities
- .NET security architecture and services
- Practical cryptography
- Desktop application security
- Data access security in .NET
- Windows Communication Foundation security
- Denial of service
- Principles of security and secure coding
- Knowledge sources

scademy
secure coding academy

Participants attending this course will:

Understand basic concepts of security, IT security and secure coding
Learn about typical coding mistakes and how to avoid them
Learn to use various security features of the .NET development environment
Have a practical understanding of cryptography
Understand security concepts of Web services
Learn about XML security
Learn about denial of service attacks and protections
Get sources and further readings on secure coding practices

Related courses:

- CL-NWA - C# and Web application security (Onsite / Virtual classroom, 3 days)
- CL-NSM - C# and Web application security master course (Onsite / Virtual classroom, 5 days)
- CL-WSC - Web application security (Onsite / Virtual classroom, 3 days)
- CL-WTS - Web application security testing (Onsite / Virtual classroom, 3 days)
- CL-NSM - C# and Web application security master course (Onsite / Virtual classroom, 5 days)

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
  - From an infected computer to targeted attacks
- Classification of security flaws
  - Landwehr's taxonomy
  - The Seven Pernicious Kingdoms
  - OWASP Top Ten 2017

Common coding errors and vulnerabilities

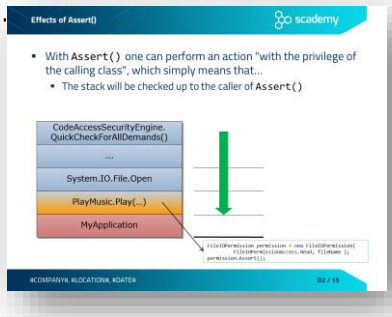
- Input validation
  - Input validation concepts
  - Injection
    - Injection principles
    - SQL injection
    - Command injection
    - Case study – ImageMagick
  - Integer problems
    - Representation of negative integers
    - Integer overflow
    - Exercise IntOverflow
    - What is the value of Math.Abs(int.MinValue)?
    - Integer problem – best practices
    - Case study – Integer overflow in .NET
  - Path traversal vulnerability
    - Path traversal – weak protections
    - Path traversal – best practices
  - Unsafe native calls
    - Exercise – Unsafe unmanaged code

- Unsafe reflection
  - Implementation of a command dispatcher
  - Unsafe reflection – spot the bug!
  - Mitigation of unsafe reflection
- Log forging
  - Some other typical problems with log files

Day 2

.NET security architecture and services

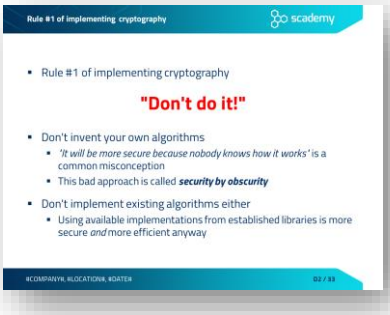
- .NET architecture
- Code Access Security
  - Full and partial trust
  - Evidence classes
  - Permissions
  - Code access permission classes
  - Deriving permissions from evidence
  - Defining custom permissions
  - .NET runtime permission checking
  - The Stack Walk
  - Effects of Assert()
  - Class and method-level declarative permission
  - Imperative (programmatic) permission checking
  - Exercise – sandboxing .NET code
  - Using transparency attributes
  - Allow partially trusted callers
  - Exercise – using transparency attributes



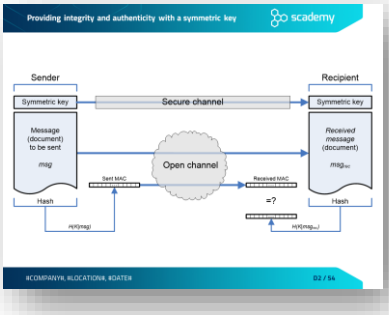
- Role-based security
  - Principal-based authorization
  - Exercise – adding role-based authorization
  - Impersonation

Practical cryptography

- Rule #1 of implementing cryptography
- Cryptosystems
  - Elements of a cryptosystem
  - .NET cryptographic architecture
- Symmetric-key cryptography
  - Providing confidentiality with symmetric cryptography



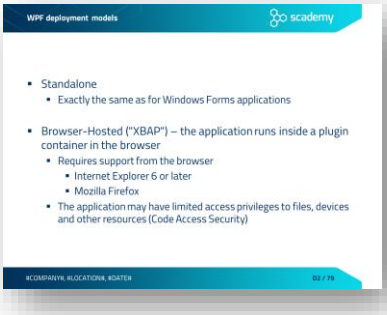
- Symmetric encryption algorithms
  - Modes of operation
  - Encrypting and decrypting (symmetric)
- Other cryptographic algorithms
  - Hash or message digest
  - Hash algorithms
  - SHAttered
  - Hashing
  - Message Authentication Code (MAC)
  - Providing integrity and authenticity with a symmetric key.....



- 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 algorithms in .NET
    - Exercise Sign
    - Exercise – using .NET cryptographic classes
- 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

Desktop application security

- Windows Presentation Foundation
  - Introduction to WPF
  - Extensible Application Markup Language (XAML)
  - WPF deployment models.....



- Common security issues with desktop .NET applications
  - Resource hijacking in WPF applications
  - Exercise – LibHijack
- Protecting .NET code
  - Authenticode
  - Exercise – Using Authenticode

- XML security
  - XML injection
    - (Ab)using CDATA to store XSS payload in XML
    - Exercise – XML injection
    - Protection through sanitization and XML validation
  - Abusing XML Entity
    - XML Entity introduction
    - XML bomb
    - Exercise – XML bomb
    - 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

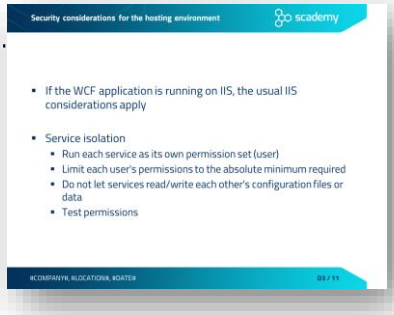
Day 3

Data access security in .NET

- Working with databases in .NET

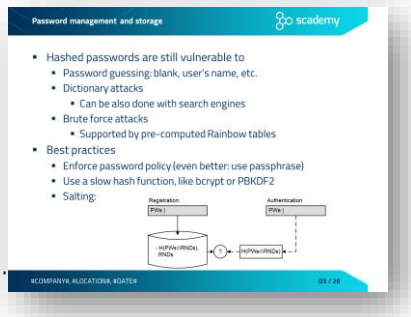
Windows Communication Foundation security

- Introduction to WCF
- WCF architecture and security considerations
  - WCF architecture
  - Security considerations for the hosting environment
  - WCF security terminology
  - Transport layer security
  - Transport layer security – client authentication
  - Message level security
  - Authorization options



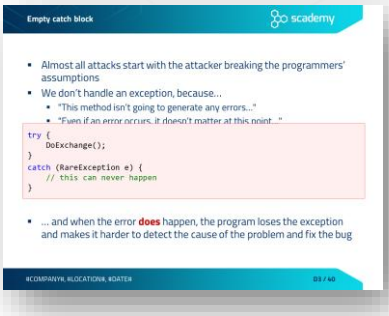
Common coding errors and vulnerabilities

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

- Improper error and exception handling
  - 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

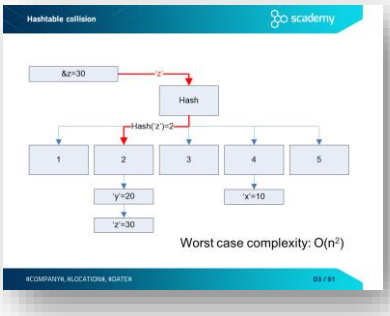


- Time and state problems
  - Concurrency and threading
  - Concurrency in .NET
  - Omitted synchronization – spot the bug!
  - Exercise – Omitted synchronization
  - Incorrect granularity – spot the bug!
  - Exercise – Incorrect granularity
  - Deadlocks
  - Avoiding deadlocks
  - Exercise – Avoiding deadlocks
  - Lock statement
- Code quality problems
  - Dangers arising from poor code quality
  - Serialization – spot the bug!
  - Exercise – Serializable sensitive
  - Class not sealed – object hijacking
  - Exercise – Object hijacking
  - Immutable string – spot the bug!
  - Exercise – Immutable strings
  - Using SecureString

scademy
secure coding academy

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 ASP.NET



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