SC460 Secure Architecture and Design

Kurzbeschreibung:

Software-Entwickler, Software- und Cloud-Architekten erhalten eine praxisnahe Einführung in Secure Architecture and Design. Vermittelt werden Best Practices für sichere Anwendungsarchitekturen und Methoden zur Reduzierung der Angriffsfläche. Behandelt werden Bedrohungsperspektiven, Threat Modeling zur Identifikation von Schwachstellen sowie Übungen, um Sicherheitsmaßnahmen gezielt anzuwenden und zu festigen.

Zielgruppe:

Das Training **SC460 Secure Architecture and Design** ist ideal geeignet für:

- Software Entwickler
- Software Architekten
- Cloud Architekten

Voraussetzungen:

Um den Kursinhalten und dem Lerntempo im Workshop **SC460 Secure Architecture and Design** gut folgen zu können, sollten Sie folgende Voraussetzungen mitbringen:

- grundlegende IT-Kenntnisse
- gundlegende IT-Security-Begriffe

Sonstiges:

Dauer: 5 Tage

Preis: 3450 Euro plus Mwst.

Ziele:

Das Training **SC460 Secure Architecture and Design** hat folgende Kursziele:

- Kenntnis und Anwendung gängiger Security Design Prinzipien
- Fähigkeiten einen Threat Model Workshop durchzuführen
- Wissen über gängige Design-Schwachstellen und deren Behebung

Inhalte/Agenda:

- ♦ **Security-Design-Prinzipien – Einführung, Anwendung und Messbarkeit**
- ♦ **Vertrauen / Trust-Principals**
  - ♦ ◊ „Never trust the Client“
  - ◊ „Zero Trust“
  - ◊ „Trusted 3rd Party“

♦ **Secure◊Design - Authentication**

- ♦ ◊ Sichere Identifier / Identities
  - ◊ Password-Based Authentication
  - ◊ Sichere Einsatz von Krypto-Verfahren
  - ◊ Kerkhof’s Principal

♦ **Secure◊Design Prinzipien: Autorisierung**

- ♦ ◊ „Segregation of Duties“
  - ◊ „Least Privilege“
  - ◊ „Avoid Broadly generic functions“
  - ◊ “Authorize close to the source”
  - ◊ „Extension of Kerkhoffs-Principle“

♦ **Weitere◊ Prinzipien im Überflug**

- ♦ ◊ „Do not be Chatty“
  - ◊ „Encrypt High“
  - ◊ „Decrease visibility“

♦ **Secure◊Design – Input / Output / Communication**

- ♦ ◊ „Input-Validierung“
  - ◊ „Output- Validierung“
  - ◊ „Black-Listing / White-Listing“
  - ◊ „Do not interpret – discard“
  - ◊ “Intercept – do not process”
  - ◊ “Don’t call me – I call you!”
  - ◊ “Resilient Design”

♦ **Secure◊Design – “Reste-Rampe”**

- ♦ ◊ “Visibility”
  - ◊ “Default is tight”
  - ◊ “Fail save”
  - ◊ “Double book-keeping”
  - ◊ “No Filesystem”

♦ **Threat-◊Modelling**

- ♦ ◊ Einführung, Anwendung, Historie, Grundlagen

♦ **Threat-◊Modelling-Methoden**

- ♦ ◊ Misuse-Cases
  - ◊ Attack-Trees
  - ◊ STRIDE
  - ◊ EoP-Card-Game
  - ◊ Tools
  - ◊ Application-Level-Threat-Modelling

♦ **Schwach◊stellen und Praxis-Übung**

- ♦ ◊ Identity Management
  - ◊ Authentication
  - ◊ Authorization

♦ **Schwach◊stellen**

- ♦ ◊ Kommunikation
  - ◊ Speicher
  - ◊ Input-Attacks
  - ◊ Angriffe durch privilegierte Benutzer
  - ◊ Angriffs-Erkennung
  - ◊

- Nachvollziehbarkeit
- Angriffe über die Infrastruktur
- Datenschutz
- Open-Source-Security
- Angriffe auf Software-Lebenszyklus
- Angriffe auf Krypto
- Angriffe auf Fehler-Situation

- **Bewertung von Schwachstellen**
  - Angriffs-Vektor
  - CVSS
  - Risiko-Bewertung

- **Workshops durchführen und dokumentieren**

- **Moderation eines Workshops**

- **Abschluss-Übung**
  - Threat Modelling
  - Risiko-Bewertung der Findings
  - Secure Design-Maßnahmen analysieren

- **Viele praktische Übungen zu den einzelnen Modulen**