SC470 Secure Development Principles

Kurzbeschreibung:

Der Workshop **SC470 Secure Development Principles** vermittelt Ihnen die theoretischen Grundlagen der sicheren Softwareentwicklung im professionellen Umfeld.

Neben der robusten Architektur steht vor allem die Business- und Threatmodellierung sowie die Risikobehandlung im Fokus. Sie lernen sämtliche Bausteine des Secure Development Lifecycle kennen: Requirement Gathering, Secure Design, Secure Implementation, Secure Testing und Deployment. Konkret werden die Themen Business- und Projektanforderungen, Threat Modeling und sicheres Design behandelt.

Der Workshop legt besonderen Wert auf aktuelle Konzepte, die durch zahlreiche interaktive Praxisbeispiele vertieft werden. Die Teilnehmer haben so die Möglichkeit, ihre Erfahrungen und Anforderungen aktiv miteinzubringen. Am Ende des Workshops werden die Teilnehmer ein solides Verständnis für sichere Softwareentwicklung im professionellen Umfeld erlangt haben und in der Lage sein, robuste und sichere Anwendungen zu planen und die Implementierung zu begleiten.

Der Kurs ist Teil des "qSkills Secure Software Quadrant", bestehend aus:

- SC460 Secure Architecture and Design
- SC470 Secure Development Principles
- SC475 OWASP Security Champion
- SC480 Secure Operations

Zielgruppe:

Das Training **SC470 Secure Development Principles** ist ideal geeignet für:

- Software Projektmanager / Product Owner
- Business Analysten / Requirements Engineers
- IT Consultants/Berater
- Junior Software-/Cloudarchitekten
- Junior Softwareentwickler

Voraussetzungen:

Um den Kursinhalten und dem Lerntempo im Workshop **SC470 Secure Development Principles** gut folgen zu können, ist Berufserfahrung in der Softwareentwicklung hilfreich. Programmierkenntnisse sind keine Voraussetzung.

Sonstiges:

Dauer: 4 Tage

Preis: 2850 Euro plus Mwst.

Ziele:

Der Kurs **SC470 Secure Development Principles** bietet:

- Erkennen von Schwachstellen in Konzepten und Architekturen
- Identifizieren von Business-kritischen Assets
- Entwickeln und Beschreiben von Angriffs-Vektoren

Inhalte/Agenda:

- ♦ **Einleitung**
  - ♦ ◊ Was ist Secure Coding und was ist es nicht?
    ◊ Begrifflichkeiten und Konzept der Schulung
- ♦ ◊
- ♦ **Requirement Gathering**
  - ♦ ◊ Business Requirements (Geschäftsfeld, Prozesse, Assets usw.)
    ◊ Project Requirements (Code-Reife, interne Funktionalitätsanforderungen, Budget, gesetzliche Anforderungen usw.)
    ◊ Threat Model (Schutz-Ziele, Identifikation von Angriffs-Vektoren, Risk Management, Mitigation Strategien)
- ♦ ◊
- ♦ **Secure Design**
  - ♦ ◊ Secure Design Principles (Bugchains, Security by Design, Viega's and Graw's Principle)
    ◊ Robust Architecture (Application Components, The Dependency Rule, Service Mesh)
    ◊ Robust Technology Design (Development Considerations, Supply Chain Considerations)
- ♦ ◊
- ♦ **Secure Implementation**
  - ♦ ◊ OWASP Top 10, CWE, Best Practices
    ◊ Authentication (Identification & Authentication, Broken Access Control)
    ◊ Processing (Input Parsing, Injection)
    ◊ Storage (Software & Data Integrity, Cryptographic Failures, Logging & Monitoring Failures)
- ♦ ◊
- ♦ **Testing**
  - ♦ ◊ Automated Testing (Test Cases, Test Setups, Tools)
    ◊ Penetration Testing (Concept, Methods, Tools)
    ◊ Chaos Engineering (Concept, Resilience, Case Study)
- ♦ ◊
- ♦ **Deployment & Maintenance**
  - ♦ ◊ Launch (Release Strategies, Hypercare)
    ◊ Longterm Support (Change Management, Feature Requests, Future Proof)
    ◊ Desaster Recovery (Backups, Supply Chain, Business Continuity)
- ♦ ◊
- ♦ **Lernstandskontrolle / Prüfung**
  - ♦