SC470 Secure Coding

Kurzbeschreibung:

Einführung in das Secure Coding.

Zielgruppe:

- Software Entwickler (Web)
- Software Architekten
- Tester
- Cloud Architekten
- DevOps

Voraussetzungen:

Allgemeine Programmierkenntnisse, Grundlagen Webentwicklung.

Sonstiges:

Dauer: 4 Tage

Preis: 2790 Euro plus Mwst.

Ziele:

- Vermittlung des Security-Gedankens über den gesamten Lebenszyklus eines Software-Produkts
- Erkennen und Vermeiden von Schwachstellen bei der Software-Entwicklung
- Aufzeigen von häufig gemachten sicherheitsrelevanten Fehlern
- Vermittlung von Best Practices zur Vermeidung sicherheitsrelevanter Fehler

In diesem Training werden Grundlagen für die Entwicklung von sicherer Software sowie Aufzeigen und Vermeiden von Sicherheitslücken beim Software-Development vermittelt. Zahlreiche praktische Übungen helfen dabei, das erworbene Wissen direkt anzuwenden und zu festigen.

Inhalte/Agenda:

- Intro: Warum sichere Entwicklung wichtig ist?
  - ♦ Vorstellung von großen Sicherheitslücken und Data Breaches
  - ♦ Risiken für Unternehmen

- Einführung in Risk Management
  - ♦ Identify risks
  - ♦ Classify risks
  - ♦ Plan
  - ♦ Monitor
  - ♦ Action
  - ♦ Communicate

- Essentials Do's and Don'ts in der Softwareentwicklung

- Software Development Lifecycle (SDL)
  - ♦ Secure Software Concepts
  - ♦ Secure Software Requirements
  - ♦ Secure Software Design
  - ♦ Secure Software Implementation/Programming
  - ♦ Secure Software Testing
  - ♦ Software Lifecycle Management
  - ♦ Software Deployment, Operations and Maintenance
  - ♦ Supply Chain and Software Acquisition

- Source Code Review
  - ♦ Best Practices im Source Code Review

- Web and Embedded Developer Threats and Vulnerabilities
  - ♦ OWASP Top 10
    - ◊ Kurzeinführung in die häufigsten Klassen von Vulnerabilites
  - ♦ Introduction to Web Interception Proxies
    - ◊ Kurze Einführung in Burp Suite in der vorinstallierten Umgebung
  - ♦ Tools of the trade (sqlmap, dirbuster,)
    - ◊ Was können bestimmte "Hacking"-Tools und wie nutze ich sie (erforderlich für "Hands on"-Teile)
  - ♦ Hands on Hacking (Insecure Example Application)
    - ◊ Juice Shop Introduction, Challenges und CTF (teilweise frei von den Teilnehmern gestaltbar aber auch geführte Angriffe mit dem Instructor)
  - ♦ Finding Bugs in Open Source Applications (Real World Hacking)
    - ◊ 0day Hunting in Open Source Application
  - ♦ Enumeration techniques (DNS, application mapping,)
    - ◊ Möglichst viel über eine Anwendung herausfinden
    - ◊ "Geheime" Anwendungsteile finden
  - ♦ OSINT (Open Source Intelligence) Using public information to attack software
    - ◊ Beispiele von geleakten Secrets (github commits), Repos auf Webservern, Fragen in Foren und vieles mehr
  - ♦ Using insecure dependencies to attack secure software
    - ◊ Third Party Module/Libraries
  - ♦ Supply Chain attacks in the wild
    - ◊ Manipulation von OpenSource Projekten um sichere Anwendungen anzugreifen, demonstriert anhand von Beispielen aus der echten Welt
- Penetrationstest
  - ♦ Vorstellung einiger anonymisierter Penetrationtest Reports
  - ♦ Analyse der gefunden Lücken und Empfehlungen für Fixes

- Web Developer - Prevention
  - ♦ Best Practices for secure web applications (Abhängig von Programmiersprache)
  - ♦ Static code analysis
  - ♦ Dynamic code analysis
  - ♦ Spotting bugs in application logic
  - ♦ Third Party Libraries and dependency management
  - ♦ Keeping the impact low

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