Secure Software

Allgemeine Angaben

Kürzel	SSO
Modulverantwortliche	Prof. Dr. Jan Stehr
Dozenten	Prof. Dr. Stehr, Prof. Dr. Nüßer, Prof. Dr. Schumann
Lehrsprache	Deutsch
Semester	5
ECTS-Punkte	5
Kontaktstunden	40
Selbststudium	85
Dauer	1 Semester
Art	Pflicht
Häufigkeit	Jedes Studienjahr
Gewichtung	5/180
Prüfungsleistung	K**R**S90

Stichwörter

- Safety, Security, Resilienz, Zuverlässigkeit
- Security by Design
- Monitoring

Zugangsvoraussetzungen

- Aktuelle Trends in der Programmierung
- SW Engineering und Advanced Software Engineering
- Betriebssysteme und Netzwerk-Infrastrukturen

Verwendbarkeit

- Entwicklungsprojekte

Qualifikations- und Kompetenzziele

Die Studierenden kennen die bestimmenden Prinzipien sicherer Software, können die einzelnen Aspekte differenzieren und praktisch damit umgehen. Sie sind für die technische, wirtschaftliche und ethische Problematik der Zuverlässigkeit von Systemen in ihrem Arbeitsumfeld sensibilisiert und kennen den Unterschied zwischen den verbreiteten organisatorischen Vorgehensmodellen und der tatsächlichen technischen Implementierung robuster – komplett neuer oder bestehender – Software. Studierende sind qualifiziert, systematisch Softwarekomponenten abzusichern oder eine existierende Absicherung sowohl konzeptionell als auch programmiertechnisch zu prüfen

Lehr- und Lernmethoden

Präsenzveranstaltungen, Eigenstudium, Fallstudienarbeit.

Besonderheiten

Übungsaufgaben, Literaturstudium, praktische Beispiele auf Basis aktueller Technologie-Stacks, Diskussion aktueller Schwachstellen und Exploits (Gründe, Voraussetzungen, Behebung, künftige Vermeidung).

Inhalte

- Aspekte der Sicherheit
    - Risiken und Zuverlässigkeit heutiger Software-Systeme
    - Safety: Begriff, Prinzipien & Ziele
    - Security: Begriff, Prinzipien & Ziele
    - Standards, Vorgehens- und Lebenszyklusmodelle
    - Ethische Aspekte sicherer Software
- Sicherheit und Software
    - Missionskritische Systeme
        - Eigenschaften
        - Beispiel Medizintechnik
        - Beispiel Luftfahrt
    - Schutzbedarf konventioneller Systeme
        - Verteilte Anwendungen und Dienste
        - Legacy Software
- Design sicherer Software
    - Eigenschaften einer sicheren Software
    - Security + Safety by Design
    - Härtung existierender Systeme
- Implementierung sicherer Software
    - Programmiersprachen und Programmierrichtlinien
    - Externe Einflüsse
        - Schnittstellen und die Laufzeitumgebung
        - Abhängigkeiten von Frameworks und Bibliotheken
    - Prüfen und Testen von Sicherheit
    - Monitoring/Überwachung im laufenden Betrieb

Grundlegenden Literaturhinweise

VIEGA, J. And MCGRAW, G., 2006. *Building Secure Software*. Boston; Addison Wesley.

FOWLER, K. (Ed.), 2010. *Mission-critical and Safety-critical Systems Handbook*. Oxford: Newnes.

Ergänzende Literaturempfehlungen

MERKOW, M. and RAGHAVAN, L., 2010. *Secure and Resilient Software Development*. London: Taylor&Francis.

PFLEEGER et al., 2015. *Security in Computing*. Upper Saddle River: Prentice Hall.