Lehrveranstaltung: IT-Sicherheit und Funktionale Sicherheit

LV-Kurzbezeichnung: FS

Verantwortlicher: Prof. Dr. Jürgen Mottok

Fakultät: EI

Lehrende/r /Dozierende/r: Prof. Dr. Jürgen Mottok, Sebastian Renner

Angebotsfrequenz: in jedem Semester

Lehrform: Wissenschaftliches Seminar mit Übungsanteil (mit Teamarbeit)

Studiensemester gemäß Studienplan: 1 oder 2

Lehrumfang [SWS oder UE]: 4 SWS (2 SWS Vorlesung und 2 SWS Übung)

Lehrsprache: Deutsch

Arbeitsaufwand [ECTS-Credits]: 5

= 5 Leistungspunkte (150 Std.)

Zeitaufwand – Präsenzstudium 45 Std. (Seminar, Workshops, Vorträge, Präsentationen, ...)

Zeitaufwand - Eigenstudium: 105 Std.

Studien- und Prüfungsleistung: Siehe Studienplan!

Zugelassene Hilfsmittel für Leistungsnachweis: Siehe Studienplan!

Inhalte:

I. IT-Sicherheit (engl. IT-Security)

System

- Normen und Richtlinien (ISO 27001, BSI Grundsicherheitsrichtlinien, ...)
- Grundlagen
- Schutzziele und Bedrohungen
- Internet-(Un)Sicherheit
- Security Engineering
- Bewertungskriterien
- Sicherheitsmodelle
- Basistechnologien
- Kryptographische Verfahren (Symmetrische/ asymmetrische Verschlüsselung) und Krypto-Analyse
- Hashfunktionen und elektronische Signaturen
- Schlüsselmanagement
- **Post-Quantum Kryptographie (PQC)**
- **Quantum Key Distribution (QKD)**
- System- und Anwendungssicherheit
- Basis von Angriffstechniken und Systemhärtung
- Authentifikation
- Zugriffskontrolle

- Sicherheit in Netzen (Firewall, ...)
- Sichere und mobile drahtlose Kommunikation

Software (Basiswissen sichere Software)

- „Secure Software Engineering“
- Secure Design Pattern
- Krypto-Bibliotheken und Implementierungstechniken
- Sicheres Programmieren
- Software auf Sicherheit testen

II. Funktionale Sicherheit (eng. Functional Safety)

System

- Normen und Richtlinien (IEC 61508, ISO 26262, ...)
- Kenngrößen der Risiko- und Zuverlässigkeitsanalyse
- Maßnahmen zur Risikobestimmung (Gefahrenanalyse, Wahrscheinlichkeitsanalyse)
- Risikomatrix
- Risikograph
- FMEDA und Fehlerbaumanalyse
- Ereignisbaumanalyse
- Layer of Protection Analyse (LOPA)
- Zuverlässigkeitsblock-Analyse
- Markov-Modell
- Lebenszyklusbetrachtung eines Sicherheitssystems
- Common Cause Failure
- Proofstest

Software

- Normen und Richtlinien
- Entwicklungsprozess
- Patterns für fehlertolerante Software
- Diversität und Codierung
- Geeigneter Einsatz von Programmiersprachen
- Coding Guidelines (MISRA C, MISRA C++), Safety Bibliotheken

Lernziele/Lernergebnisse/Kompetenzen:

Die Studierenden sind nach der erfolgreichen Teilnahme an diesem Modul in der Lage, grundlegende Begriffe aus den Domänen IT-Security und Functional Safety zu interpretieren und zu erklären. Sie lernen etablierte wissenschaftliche Methoden zur Bestimmung von Systemrisiken und der Systemzuverlässigkeit kennen. Das Modul stellt außerdem relevante Standards und Normen aus dem Themenfeld Sicherheit (IT-Security, Functional Safety) vor und vermittelt den richtigen Umgang mit diesen in der Praxis. Studierende kennen zum Ende des Kurses die aktuell wichtigsten kryptographischen Verfahren und können formulieren, auf welchen mathematischen Problemen diese fußen. Ebenso werden die Konzepte der diversitären Redundanz vermittelt, die als Architekturprinzipien einer funktional sicheren Software- und Systemarchitektur angewandt werden.

In anwendungsnahen Übungen werden potentielle, durch Programmierfehler verursachte Sicherheitslücken untersucht. Das Verständnis für ausgewählte Algorithmen wird durch einfache, manuelle Berechnungen gestärkt. Anhand der durchgeführten Übungen, können zuvor in der Vorlesung besprochene Limitierungen bestimmter Sicherheitskonzepte anschaulich reflektiert werden.

Kenntnisse

- Verständnis grundlegender Begriffe aus den Bereichen IT-Security und Safety
- Methoden zur Evaluierung der Systemzuverlässigkeit
- Kennen wichtiger Normen aus dem Umfeld Safety/IT-Security
- Funktionsweise und mathematischer Hintergrund kryptographischer Verfahren
- Konzepte zum Aufbau von Vertrauen in öffentlichen Netzwerken
- Basistechniken zur Absicherung von Datenverkehr vor Angreifern

Fertigkeiten

- Durchführen einer Risikoanalyse für ein gegebenes System
- Auswählen und Anwenden sicherer Software- und Systemarchitekturen
- Auswählen und Anwenden passender sicherer Coding-Guidelines
- Bewerten kryptographischer Algorithmen hinsichtlich ihrer Eignung für unterschiedliche Einsatzzwecke
- Untersuchen von Quellcode hinsichtlich möglicher Schwachstellen oder Inkompatibilitäten mit gewählten Standards
- Verwenden von bestehenden Bibliotheken zur Integration kryptographischer Verfahren in umfangreichere Software-Projekte

Die Diskussion der Kompetenzen erfolgt entlang dem Kompetenzgitter nach Erpenbeck (Erpenbeck 2017). Der Schwerpunkt der Lernzielformulierung liegt in der Anwendung. Die angestrebten Kompetenzen dieses Moduls werden gemäß der Taxonomie der Lernziele nach Bloom geordnet: Wissen (1), Verständnis (2), Anwendung (3).

Die höherwertigen Lernziele Analyse (4) und Beurteilung und Synthese (5) werden im Rahmen der Masterausbildung ebenfalls verfolgt; Dem Europäische Qualifikationsrahmen für lebenslanges Lernen (EQR) folgend werden die Bildungsniveaus in Kenntnisse, Fertigkeiten und Kompetenzen unterschieden. Die Bildungsniveaus mit den Lernzielebenen Analyse (4) und Synthese (5) verfolgen das Masterstudium und anschließende Promotionsverfahren.

Fach- und Methodenkompetenz

- Evaluierung und Kontextualisierung der Ergebnisse von sicherheitsrelevanten Systemanalysen (3)

- Bedarfsorientierte Auswahl und Extraktion relevanter Richtlinien aus Standards der IT-Security/Functional Safety (4)
- Ausgewählte quantitative und qualitative Verfahren der Funktionalen Sicherheit und IT-Sicherheit ausführen (3)
- Anpassung und Anwendung eines Prozesses zur Etablierung eines sicheren Software-Entwicklungs- und Lebenszyklus (4)
- Erkennen potentieller, offensichtlicher Schwachstellen in Quellcode durch manuelle und automatisierte Analyse (3)
- Konzeptionierung grundlegender Maßnahmen zur Erhöhung der Systemsicherheit (4)
- Analytische Fähigkeiten und Konzeptionsstärke entwickeln (4)
- Beurteilungsvermögen zeigen (4)
- Projektmanagement und Planungsverhalten (3)
- Nachweis von im Studium erworbenen Fachkenntnissen (3)
- Fähigkeit zum systematischen und methodisch korrekten Bearbeiten eines begrenzten Themas (Systematisch-methodisches Vorgehen) (3)
- Nachweis der Selbständigkeit bei der Lösung einer vorgegebenen Aufgabe (Originalität von Lösungsideen) (4)
- Fähigkeit zur Problematisierung und (Selbst-)Kritik (Systematik in der Bewertung der Lösungen) (5)
- Qualität der Ergebnisse - Neuartigkeit, Güte, Zuverlässigkeit (4)
- Fähigkeit zur logischen und prägnanten Argumentation (Beispielsweise Wissenschaftliches Schreiben) (4)
- Formal korrekte Präsentation der Ergebnisse (4)

Personale Kompetenzen

- Entwicklung einer normativ-ethischen Einstellung (3)
- Hilfsbereitschaft in einem teamorientierten Arbeiten zeigen (3)
- Zuverlässigkeit im eigenen Team (3)
- Offenheit für veränderte Randbedingungen (3)
- In Selbstmanagement die eigene Arbeit gestalten (3)
- Mit Einsatzbereitschaft Ideen ins Team einbringen (3)

Aktivitäts- und Handlungskompetenz

- Entscheidungsfähigkeit bei mehreren Alternativen entwickeln (3)
- Tatkraft und Gestaltungswille im Forschungsdesign zeigen (3)
- Mit Innovationsfreudigkeit unterschiedliche neue Ideen annehmen (3)
- Zielorientiertes Führen in Teilaufgaben in einem Team (3)
- Ergebnisorientiertes Handeln entwickeln (3)
- In schwierigen Situationen Beharrlichkeit zeigen (3)
- Impulse in Workshops des Teams geben (3)
- Optimistische Grundhaltungen im Team sich aneignen (3)

Sozial- kommunikative Kompetenzen

- Konfliktlösungsfähigkeit zeigen (5)
- Integrationsfähigkeit zeigen und verschiedene Positionen bei der Aufgabenbearbeitung zuzulassen (5)
- Die eigene Teamfähigkeit weiter entwickeln (5)

- Die eigene Problemlösungsfähigkeit entwickeln (3)
- Verständnisbereitschaft zeigen im dialogischen Diskurs (5)
- Mit Experimentierfreude neue Ideen zulassen und ausprobieren (3)
- Die eigene Sprachgewandtheit im Team ausreifen (3)
- Beziehungsmanagement im Team entwickeln (5)
- Pflichtgefühl in den Aufgaben zeigen (3)

Angebotene Lehrunterlagen:

- Skript, Foliensatz, Literaturverweise

Lehrmedien:

- Übungsaufgaben
- Aktuelle Paper

Literatur:

Ein Quellenverzeichnis findet sich im digitalen Kursraum moodle.

Weitere Informationen zur Lehrveranstaltung:

Vorkenntnisse in Software Engineering und Programmierung (bspw. C/C++)