Secure Software Engineering
Lehrplan zum Basiskurs

ASQF Secure Software Engineer SSE

Lehrplan Version: Draft V1

21.03.2019

Copyright und Nutzungsrechte

Autoren

Dr.-Ing. Armin Lunkeit
Prof. Dr. Friedrich Holl
Dipl.-Ing. (FH) Hauke Petersen
Prof. Dr. Ivo Keller
Dipl.-Ing. Konstantinos Dalamagkidis, PhD
Dr. Kristian Trenkel
Dipl.-Wi.-Math.Mareike Roth
Dipl.-Phys. Max Perner
MSc Olga Jaufman
Dr. Thomas Fehlmann
Dr.-Ing. Tobias Koal

Reviewer

Dipl.-Ing. Axel Gürtler
Dr. Kristian Trenkel
Dr.-Ing. Armin Lunkeit
Günter Jung
Torsten Schulz
Vera Gebhardt
Dipl.-Ing. Axel Wintsche

Änderungsübersicht

Version	Datum	Autor	Bemerkung
1.0	27.11.2018	Autoren und Reviewer	Erste freigegebene Version

Inhalt

1. Grundverständnis für SSE und dessen Ziele (3h)

1.1 Die Bedeutung der Grundbegriffe aus dem Sicherheitsmanagement

1.1.1 LO: Die Einordnung von Safety / Funktionale Sicherheit und Security / IT-Sicherheit kennen (K1, 5min)

- Safety (funktionale Sicherheit) schützt Menschen und Umwelt vor Device (Gerät)
- Security schützt Device vor Manipulation
- Safety fordert Security ein

1.1.2 LO: Definitionen Gefahr, Gefährdung, Risiko, Bedrohung kennen (K1, 5min)

(Verweis 3.1. dort als Beispiel)

- **Gefahr**
- **Gefährdung**: Szenario aus Funktionaler Sicherheit
- **Bedrohung**
- **Risiko**
- Bedrohungsszenario Cybersecurity

Quelle [1]

1.1.3 Notwendigkeit von Standards und Regelwerke erkennen (K1, 5min)

- Dies ist keine Normenschulung
- Es existieren spezifische Standards und Complianceanforderungen, die beachtet werden müssen.

1.2 Ziele

1.2.1 Die Security Triade kennen (K1, 5min)

- CIA (Vertraulichkeit, Integrität, Verfügbarkeit) als zentrale Security Ziele kennen
  - **Vertraulichkeit**: Kein unautorisierter Zugang ermöglicht
  - **Integrität**: Keine unautorisierten Modifikationen ermöglichen
  - **Verfügbarkeit**: von Diensten sicherstellen

1.2.2 LO: Die Bedeutung und das Verfahren des Secure SW Engineering verstehen (K2, 15 min)

- Verfahren:
  - Aus **regulatorischen** Anforderungen,
  - **Unternehmenszielen und Projektzielen** für SSE definieren,
  - **Qualitätsmerkmale** [2] [3] ableiten, im Detail: IT-Sicherheit
  - deren **Priorisierung** und
  - Ableitung von **Sicherheitsanforderungen** und **Architekturen** verstehen

1.3 Attribut Datenschutz

1.3.1 Den Begriff Datenschutz kennen (K1, 5min)

- Es ist notwendig, das Sammeln von sensiblen Daten zu minimieren und diese richtig zu verwalten.

1.3.2 LO: Sich an den Hauptmerkmalen der DSGVO (o.ä. BSI) Vorschriften erinnern können (K1, 5 min)

- Die DSGVO führt explizit folgende Grundsätze für die Verarbeitung personenbezogener Daten auf:
- Rechtmäßigkeit, Verarbeitung nach Treu und Glauben, Transparenz
- Zweckbindung (Verarbeitung nur für festgelegte, eindeutige und legitime Zwecke)
- Datenminimierung („dem Zweck angemessen und erheblich sowie auf das notwendige Maß beschränkt“)
- Richtigkeit („es sind alle angemessenen Maßnahmen zu treffen, damit [unrichtige] personenbezogene Daten unverzüglich gelöscht oder berichtigt werden“)
- Speicherbegrenzung (Daten müssen „in einer Form gespeichert werden, die die Identifizierung der betroffenen Personen nur so lange ermöglicht, wie es erforderlich ist“)
- Integrität und Vertraulichkeit („angemessene Sicherheit der personenbezogenen Daten, einschließlich Schutz vor unbefugter oder unrechtmäßiger Verarbeitung und vor unbeabsichtigtem Verlust, unbeabsichtigter Zerstörung oder unbeabsichtigter Schädigung“)
- Recht auf “Vergessen werden”

1.1.1 LO: Die Für Datenschutz wichtige Begriffe erkennen können (K1, 5 min)

- **Personally Identifiable Information (PII)** – Informationen, mit denen eine dritte Partei die betroffene Person identifizieren und kontaktieren kann. Dies betrifft nicht nur Telefonnummern, Emailadressen, Versicherungsnummern und Geburtstage.
- **Personal Health Information (PHI)** – Informationen die nicht nur Identifikation zulassen, sondern auch Diagnosen, Krankheiten, Prognosen und Behandlungen betreffen.
- **Personal Financial Information (PFI)** – Informationen die zusätzlich zur Identifikation auch Informationen über Investitionen, Kreditwürdigkeit, Schulden, Gehälter und Steuern liefern.
- **Privacy Policy**

1.2 Unternehmensübergreifende Sicherheit

1.2.1 LO: Einbettung des konstruktiven SSE in Lieferketten verallgemeinern (K2, 15min)

- Katalog-Bausteine und Sicherheitsstandards
- Aufgabenteilung in Lieferkette und Lebenszyklus

Leseempfehlung [4] Standards, Normen, Best Practices kennen

1.2.2 LO: Anwendbare Normen zusammenfassen (K2, 15min)

- ISO 27000 Series
- BSI-Standards [5]
  - BSI 200-1: [6] ISMS
  - BSI 200-2: [7] IT-Grundschutz Methodologie
  - BSI-200-3: [8] Risk Analysis based on IT-Grundschutz
- weitere Normen und Regelwerke wie NIST Common Criteria [9], IEC 62443 [10] [11], PCI DSS [12], SAE J 3061 [13]

1.3 SW Lifecycle und Prozesse – Ein Überblick

1.3.1 LO: Einbettung des SSE fürs Unternehmen und in die Prozesse verstehen (Impact) (K2, 15 min)

- Sicherheitseffektivität
- Sicherheitseffizienz
- Transformation der nichtfunktionalen Ziele zu funktionalen technischen Anforderungen im Prozess

1.4 Software-Lebenszyklus und IT-Sicherheit

1.4.1 LO: IT-Sicherheit als QM im SW Lebenszyklus verstehen (K2, 15min)

1.4.2 LO: Zusammenhänge der wesentlichen Aktivitäten und Einflussfaktoren auf SSE im Lebenszyklus verstehen (K2, 15min)

Quelle [14]

- Der Software-Lebenszyklus besteht aus (Anforderung => Analyse und Design => Implementierung => Test => Inbetriebnahme, Betrieb und Wartung, Deinstallation) Charakterisierung der jeweiligen Lebenszyklusphasen
- Es gibt Möglichkeiten des Einsatzes von Aspekten des Security Engineerings innerhalb der Softwareentwicklung
- Nicht-technische Einflussgrößen (Finanzen, Ressourcen, Image) nehmen Einfluss auf die Behebung von Sicherheitslücken im Kontext des Software-Lebenszyklus.
- Die Entwicklung und Dokumentation nach einem Prozess, der Secure Design und Secure Software Engineering umsetzt, ist eine Methode, um Regeltreue (“Compliance”) bezüglich anzuwendender Normen zu erreichen.

ENTWURF

2. Bedrohungsanalyse und Anforderungen 2h 40min

2.1.1 LO (Kapitel übergreifend): Einbindung in bestehende Softwareentwicklung kennen (K1, 5min)

- Verschiedene Standards empfehlen die Anwendung eines Softwareentwicklungsprozesses.
- Es wird kein Prozess vorgeschrieben, damit SSE sich in bestehende Entwicklungsvorgänge integrieren kann.
- Methode schafft Compliance (Wiederholung)

2.1.2 LO: Für SSE wichtige Qualitätsmerkmale kennen – mit Übung (K3, 60 min)

2.2 Begriffe

2.2.1 LO: Den Begriff Risiko kennen (K1, 5min)

- Es gibt unterschiedliche Standards, damit auch unterschiedliche Definitionen
- Hier: Risiko ist Schwere des möglichen Schadens multipliziert mit Wahrscheinlichkeit des Schadenseintritts [15]
- Die Anwendung der Risikoanalyse ermöglicht Priorisierung von Gefährdungen und Aufwand von Maßnahmen [16]

2.2.2 LO: Risikoanalyse als allgemeinen Begriff kennen (K1, 5min)

- Geht auch auf Elementare Gefährdungen ein
- Geht auf Angriffe außerhalb der Domäne der Software ein (Angriffe auf Serverraum, Social Phishing, etc.
- OWASP [17]
- als Beispiel nach BSI [8]

2.3 Bedrohungsanalyse in der Designphase

2.3.1 LO: Definition von Modell und Gründe / Vorteile für modellbasiertes Vorgehen kennen (K1, 5min)

- Modelle sind vereinfachte Darstellungen der Realität.
- Modelle stellen abstrakt da und betonen Aspekte

2.3.2 LO: Diagramme (Whiteboard) als einfache Modelle zum Finden von Vertrauensgrenzen und Angriffsflächen konstruieren können (K2, 15 min)

- Modell ist hier beispielsweise ein Diagramm (Zustands-, Use-Case-, Datenfluss-, …)
- Hier kann durch einzeichnen von Vertrauensgrenzen Handlungsbedarf sichtbar werden.
- Auch Angriffsflächen können so sichtbar werden.

2.4 Methoden der Bedrohungs- und Risikoanalyse

2.4.1 Unterschiedliche Herangehensweisen verstehen (K2, 15min)

- Eine Bedrohungsanalyse kann durchgeführt werden mit Fokus auf Asset (angreifbare Güter), Attacker (Wünsche/Ziele oder Vorgehensweisen des Angreifers), Vulnerabilities (welche Schwachstellen typischerweise ausgenutzt werden), Software-centric/Modellierung, Risk (Ursache der Risiken)

2.4.2 LO: Verstehen, wie man schützenswerten Güter identifiziert und priorisiert (K2, 15min)

- Frage nach einer Risikoanalyse
- Frage nach Kosten der Risikominderung
- Frage nach vertretbarem Restrisiko

2.4.3 LO: Verschiedene Methoden der Bedrohungsanalyse kennen (K1, 5min)

- Verschiedene Methoden kennen
  - z. B. Attack Trees, CVSS, Octave, Stride, Trike,

2.4.4 LO: Eine Methode (Attack Trees) zur Bedrohungsanalyse anwenden können (K3, 60 min)

  - am Beispiel Autonomes Fahrzeug und evtl. DSGVO als Anwendungsbeispiel

2.4.5 Risikoanalyse als Priorisierungsmaßstab kennen (K1, 5min)

- Risikoanalyse als Bewertungsmaßstab
- Aus Risikoanalyse folgen Security Requirements.
- Minderung des Risikos durch Maßnahmen und erneute Bewertung des Restrisikos, sowie Akzeptanz des Restrisikos

2.5 Anforderungen

2.5.1 LO: Grundlagen des Requirements-Engineerings verstehen (K1, 5min)

- RE dient der effizienten und fehlerarmen Entwicklung komplexer Systeme.
- Ziel ist, gemeinsames Verständnis über ein zu entwickelndes System zwischen Auftragnehmer und Auftraggeber zu erreichen.
- Es existieren Requirementsmanager und Requirementsengineer hierfür
- Es gibt einen CPRE-FL, der empfohlen wird.

2.5.2 LO: Qualitätsmerkmale Nachvollziehbarkeit und Umsetzbarkeit kennen (K1, 5min)

- Nachvollziehbarkeit bedeutet, dass sich jede Anforderung auf einen Anfordernden zurückverfolgen lässt.
- Jede Aufgabe ist mit den Zwischenschritten der Implementierung und nötigen Testfällen verknüpft.
- Umsetzbarkeit bedeutet, dass es technisch möglich ist, die Anforderung zu realisieren
- Es gibt weitere. Diese Zwei sind für SSE besonders wichtig

2.5.3 LO: Spezielle Bestimmungsmethoden für Anforderungen kennen (K1, 5min)

- z. B. Square [18], Common Criteria [9], OpenSamm [19], …

2.5.4 LO: Quellen weiterer Anforderungen kennen (K1, 5min)

- von Dritten
  - Kundenanforderungen
  - Unternehmensrichtlinien
  - Best Practices
- Gesetzliche Vorgaben
  - Produktanforderungen
- aus dem eigenen Prozess
  - Data classifications / Threat modeling
  - Funktionale Spezifikation / Use cases
  - Modellierung (z. B. Misuse cases)

2.5.5 LO: Verfolgbarkeitsketten/-graphen, Versionierung kennen (K1, 5min)

- Es gibt Professionelles Konfigurationsmanagement. Ein securer Prozess sollte diese Methoden nutzen.
- Bei der Erstellung, der Veränderung und Erfüllung von Anforderungen ist es nötig, diese zu versionieren.
- Verfolgbarkeitsketten/-graphen sind Methoden, die Auswirkungen von Änderungen sichtbar zu machen.

3. Engineering und Architektur (konstruktives SSE) 1h35min

3.1 Die Prinzipien des konstruktiven SSE

3.1.1 Moderne Softwarearchitektur verstehen (K2, 15min)

- Grober Überblick [20] über
  - Docker [21] [22] und Kubernetes [23] für Microservices [24]
  - Monitoring, Tracking, Tracing
  - Intrusion Detection mittels Pattern Recognition
  - Data Movement Encryption Methoden
  - Managing Encryption Keys

3.1.2 LO: Verstehen des Messens von Privacy (K2, 15 min)

- Bewertung des Wertes der Daten
- Bewertung der Maßnahmen des Schutzes

3.2 Modernes Sicherheitsdesign

3.2.1 LO: Modernes Sicherheitsdesign kennenlernen (K1, 5 min)

- Schnittstellen stellen Zugriff auf Daten und Funktionen bereit und müssen entsprechend geprüft werden.
- Das Monitoring der Dateigröße und der Schnittstellen liefert Hinweise auf unberechtigte Zugriffe.
- Geschützte Datenablage (z.B. TPM, HSM)

3.3 Zugriffsverwaltung

3.3.1 LO: Einbettung des konstruktiven SSE in Organisation verstehen (K2, 15min)

- Trennung von Daten und Funktionen
- Zugriffsverwaltung
- Rollenkonzept
- und Organisatorische Einbettung von Encryption Key Management

3.4 Etablierte Techniken

3.4.1 LO: Ansätze und Methodik verstehen (K2, 15min)

- Secure Design Principles
- Secure Design Patterns [25] [26] [27] [28] [29] [30]
- Secure Coding [31] [32]

  Hinweis auf Code Analyzer for Security Weaknesses im Kapitel Security Testing

Weitere Quellen [33]

4. Security Testing (analytisches SSE) 1h25min

4.1 Grundwissen zum Softwaretest

4.1.1 LO: Grundwissen zum Softwaretest kennen (K1, 5 min)

- Unterschied Black-Box / White-Box Testing
- Entwurfsmethoden für Testfälle (Äquivalenzklassenbildung, Grenzwertanalyse, …)
- Grundzüge des klassischen Softwaretests (Certified Tester – Foundation Level)
- Ablauf und Schwerpunkte beim Test von Hardware – Security Prozessor, Smart Cards, …

4.2 Typische Angriffsverfahren aus Testsicht

4.2.1 LO: Verstehen der typischen Angriffswege (K2, 15min)

Darstellung Anhand von (historischen) Beispielen.

- Darstellung typischer Angriffe
  - Code Modification = Virus
  - DLL Hooking
  - Man in the Middle

4.2.2 LO: Verstehen der typischen Fehler (K2, 15min)

Darstellung Anhand von (historischen) Beispielen.

- Darstellung typischer Angriffe
  - Code Modification = Virus
  - DLL Hooking
  - Man in the Middle
- Darstellung typischer Fehler, die zu Security-Problemen führen
  - Unzureichender Schutz von Passwörtern (Salting und Hashing)
  - Mangelnde Kapselung (Firewalls, Seitenwege, User Input Validation (Verhindern von z. B. SQL Injection))
  - Beispiele aus der Historie → Sensibilisierung

4.3 Analyse der Security Architektur

4.3.1 LO: Verstehen von Methoden zur Schwachstellenanalyse von Architekturen (K2, 15min)

- Statischen Analyse-Methoden zur Erkennung von Schwachstellen sind verfügbar

4.4 Methoden zum Auffinden von Schwachstellen in Architekturen

4.4.1 LO: Statische Analyse Techniken kennen (K1, 5 min)

- Statische Analyse Techniken → inklusive Tools

4.5 Testverfahren für Security Testing

4.5.1 LO: Systematisches Testing mittels Tools kennen lernen (K1, 5 min)

- Es existieren fortgeschrittene Testverfahren für den Security-Bereich (z. B. Fuzzing, Genetische Algorithmen, …)
- Es existiert eine umfangreiche Liste (Testverfahren, Tools, Herangehensweisen) beim BSI. S. Anhang, Darstellung verschiedener Arten von Werkzeugen und Nennung von Beispielen (z.B. Valgrind)

- Ziel ist zu verstehen, dass der Mensch nur Funktionalität testen kann. Da Security Requirements weiter gehen sind Tools nötig.

4.6 Erinnerung: Bewertung und Nachweis der IT-Sicherheit

4.6.1 LO: Bewertung und Nachweis der IT-Sicherheit kennen (K1, 5min)

- Risikoanalyse und Metriken
- Schemas und Vorgehens der Common Criteria
- Rolle formaler Modelle
- Rolle des sicherheitsorientierten Tests

5. SSE in Deployment und Betrieb (180 min)

5.1 Was ist Bestandteil von Deployment und Betrieb von Software?

5.1.1 LO: Bestandteil von Deployment und Betrieb von Software nennen können (K2, 15 min)

5.1.2 LO: Kernbegriffe von Deployment und Betrieb erklären können (K2, 15 min)

- Es gibt Elemente in einer Secure Operations Policy (SVM, Incident Response, Patch Management, Änderungsmanagement, Redundantes Design, etc.)
- Es gibt IaaS, SaaS, PaaS, Managed Services sowie Continuous Delivery und diese haben einen Zusammenhang zur IT-Sicherheit
- Es gibt die Begriffe DevOps und SecDevOps und dessen Hauptmerkmale

5.2 Die Notwendigkeit von IT-Sicherheit im Deployment und Betrieb

5.2.1 LO: Die Notwendigkeit von IT-Sicherheit im Deployment und Betrieb erkennen können (K2, 15min)

- Schutz der Prozessumgebung (Einsatzumgebung) als übergeordnetes Ziel
- Notwendigkeit die Software-Repositories und Buildumgebung auch zu schützen
- Vertrauliche Dokumente nach Gebrauch schreddern.
- Dokumentation des bestimmungsgemäßen Gebrauchs.

5.3 Sicheres Deployment

5.3.1 LO: Sicherem Deployment anwenden können (K3, 60 min)

Elemente von Sicherem Deployment auflisten, ihre jeweiligen Eigenschaften und Vorteile erklären und beispielhaft einer Bedrohung eine Sicherheitsmaßnahme zuordnen können

- Erklärung der Begriffe "Environment Hardening" und "Secure Defaults" anhand von Beispielen (Netzwerksegmentierung, Credential-Management, Reduzierung der installierten Anwendungen und Nutzer, Applikationsrechteverwaltung, Virtualisierung, usw.)
- Unterschiede zwischen
  - DAC Discretionary Access Control
  - MAC Mandatory Access Control
  - RBAC Role based Access Control
- Vorteile und Nachteile der Virtualisierung
  - Hypervisor kontrolliert Anwendungen während der Ausführung
  - Erhöhter Ressourcenverbrauch
- Techniken für die Gewährleistung der Integrität (z.B. Hashes, Signaturen, Jails, Immutable-Deployment) und Verfügbarkeit (z.B. Load-Balancing, Datenreplizierung, Redundanz)
- Hardware-basierte Technologien für Trusted Computing und für die Verwaltung von kryptographischem Material (wie TPM (Trusted Platform Module) and HSM (Hardware Security Module)), Vermitteln des Konzepts von Root of Trust und Chain of Trust

5.4 Systemüberwachung

5.4.1 LO: Begriff Systemüberwachung kennen (K1, 5min)

- Systemüberwachung ist nötig.
- Bedeutung der kontinuierlichen Systemüberwachung
- Mechanismen für Systemüberwachung (SIEM, IDS, Malware Scanners)
- SIEM Security Information and Event Management[1]

5.5 Patch Management und Vulnerability Management

5.5.1 LO: Die Hauptmerkmale von Patch-Management und Software-Vulnerability-Management auflisten können (K2, 15min)

- Definition "Patch Management" und "Software Vulnerability Management"
- Welche Informationen sind für SVM benötigt, und was für Quellen können dafür benutzt werden
- Was ist ein Patch und wie kann es die Security Eigenschaften von Software beeinflussen?
- Aktivitäten für einen sicheren Rollout neuer Funktionen und Bug Fixes

5.6 Incident Response

5.6.1 LO: Incident Response als wichtiger Geschäftsprozess beim Betrieb von Software kennen (K1, 5min)

- Die Bestandteile einer Richtlinie für "Incident Response" und die Eigenschaften eines effektiven Response-Teams
- Gründe für die Wichtigkeit die Grundursachen eines Ereignisses festzustellen
- Gründe für verantwortungsvolle Offenlegung von Sicherheitsproblemen und wie ein Unternehmen sie fördern kann

5.7 Beschaffung und Außerbetriebnahme

5.7.1 LO: Begriffe und Aktivitäten bezüglich Beschaffung und Außerbetriebnahme auflisten können (K1, 5min)

- Aktivitäten bei der Beschaffung von Software
- Begriffe ILM (Information Lifecycle Management, SLA (Service Level Agreement) und EoL (end of life)
- Wichtigste Bestandteile einer EoL-Policy
- Möglichkeiten für den Umgang mit Daten z.B. ordnungsgemäße und gesetzeskonforme Vernichtung von Datenträgern

[1] Verwendungsmöglichkeiten und Grenzen von IDS/IPS und Malware-Scanner
IDS Intrusion Detection System
IPS Intrusion Prevention System
führt zu weit. Netzwerksicherheit statt secure Software Engineering

6. SW Lifecycle und Prozesse – Eine Zusammenfassung (3,h)

6.1 Team-Entwicklung

6.1.1 LO: Verstehen, dass gemeinsame Entwicklung nicht aus Abnicken besteht (K2, 15 min)

- Teampsychologie und Softskills
- 4 Augen Prinzip als
  - Schutz vor versehentlicher Fehleingaben
  - Schutz vor absichtlichen Fehleingaben
- weitere Punkte

6.2 Vorgehensmodelle in Entwicklung und Test

6.2.1 LO: Anwenden des Security Development Life Cycle (K3, 60 min)

Der Referenzprozess des Security Development LifeCycle und Abbildung der Spezialitäten auf sequentiellen und iterativen/agilen Vorgehensmodellen soll am Beispiel angewendet werden

- Die Merkmale sicherheitsorientierter Vorgehensmodelle können benannt und innerhalb eines Beispielprozesses klar identifiziert und zugeordnet werden.
- Anhand exemplarischer Prozesse die Unterschiede zwischen sequentiellen und iterativen Vorgehensmodellen kennen und die jeweiligen Aktivitäten des Security Engineerings innerhalb dieser Vorgehensmodelle verstehen.
- Verstehen des Security Development LifeCycle als Referenzprozess

Quelle [14]

- Übung am Beispiel.

6.3 Vorgehensmodelle im Betrieb

6.3.1 LO: Verstehen des Einflusses von Sicherheitslücken auf die IT-Sicherheit des betriebenen Produktes (K2, 15 min)

- Klassifikationen und
- Metriken gemeldeter Sicherheitslücken
  - CVE,
  - OWASP
  - etc.

6.3.2 LO: Techniken zur Überwachung betriebener Systeme verstehen (K3, 60 min)

- Es gibt Techniken wie
  - Security Incident
  - Event Monitoring
  - ETSI Information Security Indicators
- Übung am Beispiel

7. Literaturverzeichnis

[1] ISO/IEC 27034-1:2011(E) 6.5.3.

[2] ISO 25000.

[3] CPIOT Kapitel 2.

[4] Joint Security Management, Faber, 2018.

[5] „BSI IT Grundschutz," BSI, [Online]. Available: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzStandards/ITGrundschutzStandards_node.html. [Zugriff am 10 11 2018].

[6] „BSI," [Online]. Available: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzStandards/Standard201/ITGStandard201_node.html. [Zugriff am 10 11 2018].

[7] „BSI," [Online]. Available: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzStandards/Standard202/ITGStandard202_node.html. [Zugriff am 10 11 2018].

[8] „BSI," [Online]. Available: https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzStandards/Standard203/ITGStandard203_node.html. [Zugriff am 10 11 2018].

[9] „nist.gov," 25 11 1998. [Online]. Available: https://www.nist.gov/publications/common-criteria-launching-international-standards. [Zugriff am 10 11 2018].

[10] „iec.ch>search>IEC62443," [Online]. Available: https://webstore.iec.ch/searchform&q=IEC%2062443. [Zugriff am 10 11 2018].

[11] „wikipedia / IEC 62443," [Online]. Available: https://de.wikipedia.org/wiki/IEC_62443. [Zugriff am 10 11 2018].

[12] „PCI Security Standards Council," [Online]. Available: https://www.pcisecuritystandards.org/. [Zugriff am 10 11 2018].

[13] „sae.org," [Online]. Available: https://www.sae.org/standards/content/j3061/. [Zugriff am 10 11 2018].

[14] „Microsoft SDL - Security Development Lifecycle," [Online]. Available: https://www.microsoft.com/en-us/sdl. [Zugriff am 10 11 2018].

[15] A. A. o. C. Estimators, Risk Analysis and Contingency Determination Using Range Estimating, Morgantown, WV: AACE International Recommended Practices, 2008-2.

[16] ISO 31000:2018, Risk management – Guidelines,, Geneva, Switzerland: ISO, 2018.

[17] „owasp.org," [Online]. Available: https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology. [Zugriff am 20 11 2018].

[18] „Carnegie Mellon University," [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=484884. [Zugriff am 20 11 2018].

[19] „opensamm," owasp, [Online]. Available: https://www.opensamm.org/. [Zugriff am 20 11 2018].

[20] S. Strobel, Firewalls und IT-Sicherheit, dpunkt.verlag.

[21] „redhat.com," [Online]. Available: https://www.redhat.com/de/topics/containers/what-is-docker. [Zugriff am 20 11 2018].

[22] „Docker auf github," [Online]. Available: https://github.com/docker-library. [Zugriff am 20 11 2018].

[23] „kubernetes," [Online]. Available: https://kubernetes.io/. [Zugriff am 20 11 2018].

[24] E. Wolff, Microservices, dpunkt.verlag.

[25] „owasp.org - Esapi-design-patterns.pdf," [Online]. Available: https://www.owasp.org/images/8/82/Esapi-design-patterns.pdf. [Zugriff am 20 11

2018].
[26] „owasp.org OWASP_Conference/2011/10.pdf,“ [Online]. Available: http://www.owasp.org.cn/OWASP_Conference/2011/10.pdf.
[27] C. H. Bob Blakley, Security Design Patterns. Technical Guide, The Open Group. , 2004.
[28] [Online]. Available: http://users.uom.gr/~achat/articles/sec_patterns.pdf. [Zugriff am 20 11 2018].
[29] K. S. R. C. S. D. S. K. T. Chad Dougherty, Secure Design Patterns. TECHNICAL REPORT, Carnegie Mellon University. CERT. , 2009.
[30] M. Schumacher, Security engineering with patterns, Berlin. , 2003.
[31] „sonarqube.org Dokumentation architecture-integration,“ [Online]. Available: https://docs.sonarqube.org/latest/architecture/architecture-integration/. [Zugriff am 20 11 2018].
[32] „OWASP - Security by Design Principles,“ [Online]. Available: https://www.owasp.org/index.php/Security_by_Design_Principles. [Zugriff am 20 11 2018].
[33] G. CodeSonar, Static Analysis for Automotive, 2018.