Jürgen Dürrwang

Steigerung der Betriebssicherheit von Personenkraftwagen durch Bedrohungsanalysen für die Informationssicherheit

Dissertation

Technische Universität München

Fakultät für Informatik
der Technischen Universität München

Steigerung der Betriebssicherheit von Personenkraftwagen durch Bedrohungsanalysen für die Informationssicherheit

Jürgen Dürrwang

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende:	Prof. Dr. Claudia Eckert
Prüfer der Dissertation:	
1.	Prof. Dr. Alexander Pretschner
2.	Prof. Dr. Reiner Kriesten, Hochschule Karlsruhe - Technik und Wirtschaft

Die Dissertation wurde am 30.11.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 15.03.2022 angenommen.

Danksagung

An dieser Stelle möchte ich all jene erwähnen, die mich auf meinem Weg der Anfertigung dieser Dissertation unterstützt und motiviert haben.

Zuerst gebührt mein Dank Herrn Prof. Dr. Alexander Pretschner und Herrn Prof. Dr. Reiner Kriesten, die meine Dissertation betreut und begutachtet haben. Ich bedanke mich zum einen ausdrücklich für die konstruktive Kritik, interessanten Debatten und wertvollen Ideen, durch welche ich mich fachlich weiterentwickeln konnte. Zum anderen bedanke mich für die Begleitung als Mentoren, welche mich auch persönlich hat wachsen lassen. Auch wenn einzelne Aufgaben herausfordernd für mich waren, haben sie maßgeblich dazu beigetragen, dass die Dissertation heute in dieser Form vorliegen kann.

Ebenfalls möchte ich mich herzlich bei meinen Kollegen Jens Köhler, Florian Sommer, Patrick Gründer, Marcel Rumez und Felix Müller bedanken, die mir mit viel Geduld, Interesse und Hilfsbereitschaft zur Seite standen. Hierbei möchte ich mich insbesondere bei Jens Köhler und Florian Sommer für die etlichen und konstruktiven Diskussionen bedanken, die mir stets den richtigen Weg für die Arbeit aufzeigten.

Für das Korrekturlesen meiner Dissertation möchte ich erneut den Kollegen Jens Köhler, Florian Sommer und Felix Müller danken, die mit jeder ihrer Rückmeldung die Qualität der Arbeit erhöhten.

Ein Dankeschön gilt außerdem den Teilnehmenden an meiner Studie, ohne die diese Arbeit nicht vollständig gewesen wäre. Ihre Offenheit zu meinen Ideen sowie ihre interessanten Anregungen waren eine Hilfe.

Abschließend möchte ich mich bei meiner Freundin bedanken, die mit ihrer selbstlosen Unterstützung einen besonderen Beitrag für diese Arbeit geleistet hat. So hat sie mir insbesondere in schweren Zeiten Kraft gegeben. Ebenso danke ich meiner Familie, die mich von Anfang an motivierte und unterstützte, den Weg der Promotion zu gehen.

Zusammenfassung

Verletzungen der Betriebssicherheit (Safety) durch Cyber-Angriffe können bei modernen Fahrzeugen zu Gefährdungssituationen führen. Aufgrund dessen müssen diese möglichst frühzeitig in der Fahrzeugentwicklung identifiziert werden. Etablierte Analyseansätze zeigen allerdings nur eine geringfügige Einbindung der Betriebssicherheit in die Security-Analyse. Die vorliegende Arbeit präsentiert daher die Security Guidewords Method (SGM), welche die Safety-Analyse mit Security-Kausalfaktoren erweitert, um Security-Bedrohungen identifizieren zu können, welche die Betriebssicherheit verletzen. Das Vorgehen basiert auf der anerkannten Hazard and Operability Study (HAZOP)-Technik und ermöglicht Safety-Artefakte in einer strukturierten und geführten Weise wiederzuverwenden, sodass Security-Bedrohungen identifiziert werden können. Eine empirische Evaluation der SGM mit einer Testgruppe aus Safety- und Security-Ingenieuren zeigt, dass das strukturierte Vorgehen Analysten bei ihrer Tätigkeit unterstützt und die Methodik sinnvoll anzuwenden ist. Neben der empirischen Evaluation der SGM wird diese außerdem in einen Leitfaden für Penetrationstests eingebunden, um Testfälle zu modellieren. Bei einer Anwendung dieser Koppelung auf ein Airbag-System wurde eine safety-kritische Schwachstelle festgestellt, welche das böswillige Auslösen der Airbag-Zündkapseln in zahlreich zugelassenen und sich derzeitig im Straßenverkehr befindlichen Fahrzeugtypen erlaubt. Neben der Identifizierung von Bedrohungen steht in dieser Arbeit ebenso deren Priorisierung im Fokus. Letzteres ist elementar, um Ressourcen möglichst optimal einzusetzen. Eine adäquate Priorisierung vorzunehmen, stellt allerdings eine Herausforderung dar. Zur Lösung dieses Problems wird ein modelbasierter Ansatz vorgestellt, der anhand von Komponenteneigenschaften des Fahrzeuges und einer Schwachstellendatenbank Angriffspfade erzeugt. Letztere beschreiben, ausgehend von einem Eintrittspunkt, die möglichen Angriffsschritte, die ein Angreifer in der vorliegenden Fahrzeugvernetzung vollziehen kann. Jeder Schritt entspricht dabei dem Ausnutzen einer Schwachstelle, was für den Angreifer mit einem Aufwand verbunden ist. Dieser setzt sich aus dem Aufwand für den Zugriff auf die Schwachstelle und dem Schwierigkeitsgrad für die Ausnutzung zusammen. Für letzteres wird auf das Common Vulnerability Scoring System (CVSS) zurückgegriffen, welches sich in der Informationstechnik (IT) bewährt hat. Für die Erreichbarkeit wird hingegen ein Angreifermodell vorgestellt, was den notwendigen physischen Zugriff des Angreifers bewertet. Basierend auf diesen Metriken kann jedem Angriffspfad eine Eintrittswahrscheinlichkeit zugeordnet und mit dem Schadenswert aus der Safety-Analyse auf einen Risikowert abgebildet werden. Die Arbeit schließt mit einer Evaluation der vorgestellten Konzepte in Form eines Softwareprototyps und mit einem Ausblick anknüpfender Arbeiten ab.

Abstract

Violations of operational safety caused by cyber attacks can lead to hazardous situations in modern vehicles. For this reason, they have to be identified as early as possible in the vehicle development process. However, established analysis approaches show only a slight integration of operational safety into the security analysis. This thesis presents the SGM, which extends the safety analysis with security causal factors in order to identify security threats that violate operational safety. The approach is based on the well-known HAZOP technique and allows the reuse of safety artifacts in a structured and guided way so that safety threats can be identified. An empirical evaluation of the SGM with a test group of safety and security engineers shows that the structured approach supports analysts in their work and the methodology can be applied in a meaningful way. In addition to the empirical evaluation of the SGM, it is also integrated into a penetration testing guideline in order to model test cases. When applying this combination to an airbag system, a safety-critical vulnerability was identified, which allows the malicious triggering of airbag detonators in numerous vehicle types that are currently on the road. Besides the identification of threats, this thesis also focuses on their prioritization. The latter is elementary in order to use resources as efficiently as possible. However, it is a challenge to adequately prioritize them. To solve this problem, a model-based approach is presented, which generates attack paths based on component properties of the vehicle and a vulnerability database. These paths describe, starting from an entry point, the possible attack steps that an attacker can perform in the existing vehicle network. Each step corresponds to the exploitation of a vulnerability, which is associated with an effort for the attacker. This is calculated from the effort required to access the vulnerability and the degree of complexity of its exploitation. To determine the latter, the CVSS is used, which has proven itself in the IT. For the accessibility, an attacker model is presented, which evaluates the necessary physical access of the attacker. Based on these metrics, each attack path can be assigned a probability of occurrence and mapped to a risk value using the damage value from the safety analysis. Finally, the thesis concludes with an evaluation of the presented concepts in the form of a software prototype and with an outlook on further work.

Inhaltsverzeichnis

Abkürzungsverzeichnis

ABS Anti-lock Braking System

ACL Additional Communication Line

ASIL Automotive Safety Integrity Level

ASTMT Automotive Security-Threat Modelling Tool

ATM Air Traffic Management

AUTOSAR AUTomotive Open System ARchitecture

BMS Batterie Managment System

BSI Bundesamt für Sicherheit in der Informationstechnik

BT Bluetooth

CAN Controller Area Network

CC Common Criteria

CD Compact Disc

CHASSIS Combined Harm Assessment of Safety and Security for Information Systems

CIA Confidentiality, Integrity, and Availability

CIAA Confidentiality, Integrity, Availability, Authenticity

CPS Cyber-physisches System

CTL Computation Tree Logic

CVE Common Vulnerabilities and Exposures Enumeration

CVSS Common Vulnerability Scoring System

CWE Common Weakness Enumeration

DFS Depth-first search

DREAD Damage, Reproducibility, Exploitability, Affected users, Discoverability

DVD Digital Versatile Disc

ECU Electronic Control Unit

ENISA Agentur der Europäischen Union für Cybersicherheit

ESCL Electronic Steering Column Lock

ESP Electronic Stability Control

ETA Event Tree Analysis

EVITA E-safety Vehicle Intrusion Protected Applications

FI Fields Initiated

FMEA Failure Mode and Effect Analysis

FMVEA Failure Mode, Vulnerabilities and Effect Analysis

FOTA Firmware Over-the-Air

FP False Positives

FSD Failure-Sequence-Diagram

FTA Fault Tree Analysis

GAL Guarded Action Language

GPS Global Positioning System

GT Goal Tree

GTST Goal Tree Success Tree

HARA Hazard and Risk Analysis

HAZOP Hazard and Operability Study

HEAVENS HEAling Vulnerabilities to ENhance Software Security and Safety

HSM Hardware Security Module

I2C Inter-Integrated Circuit

IFD Information Flow Diagram

IL Impact Level

IT Informationstechnik

Lidar Light Detection and Ranging

LIN Local Interconnect Network

MITM Man-in-the-Middle

MLD Master Logic Diagram

MMS Mensch-Maschine-Schnittstelle

MUC Misuse-Case

MUCD Misuse-Case-Diagram

MUSD Misuse-Case-Sequenz-Diagram

NF Number of Fields

NRC Negative Response Code

NVD National Vulnerability Database

OBD On-board Diagnostics

OCTAVE Operationally Critical Threat, Asset, and Vulnerability Evaluation

OEM Original Equipment Manufacturer

OWASP Open Web Application Security Project

OWASP Open Web Application Security Project

P2P Peer-to-Peer

PC Personal Computer

PCU Pyrotechnic Control Unit

PCU Pyrotechnical Control Unit

PHA Preliminary Hazard Analysis

PPV Präzision

PRO Produktivität

PTES Penetration Testing Execution Standard

Radar Radio Aircraft Detection and Ranging

RAM Random-Access Memory

RL Rechtelevel

SA Security Access

SAE Society of Automotive Engineers

SAHARA Security-Aware Hazard and Risk Analysis

SCADA Supervisory Control And Data Acquisition

SD Sequence-Diagram

SDL Security Development Lifecycle

SecL Security-Level

SeSaMo SEcurity and SAfety MOdelling

SGM Security Guide-word Method

SGM Security Guidewords Method

SIL Safety Integrity Level

SL Security-Level

SPI Serial Peripheral Interface

SPL Scrapping Program Module Loader

SPM Scrapping Program Module

SQL Structured Query Language

SSM Six-Step Model

ST Success Tree

STAMP Systems-Theoretic Accident Model and Processes

STPA-Sec Systems- Theoretic Process Analysis for Security

STPA Systems Theoretic Process Analysis

STRIDE Spoofing, Tampering, Information Disclosure, Denial of Service and Elevation of Privilege

T-MUC Textuell-Misuse-Case

T-UC Textuell-Use-Case

TARA Threat and Risk Analysis

TL Threat Level

TMT Threat Modeling Tool

TOE Target of Evaluation

TP True Positives

TS Transitionssystem

TVRA Threat, Vulnerability And Risk Assessment

UCD Use-Case-Diagram

UDS Unified Diagnostic Services

UML Unified Modeling Language

USB Universal Serial Bus

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

VDA Verband der Automobilindustrie

WLAN Wireless Local Area Network

1

Einleitung

Bei der Entwicklung von Kraftfahrzeugen stehen Ingenieure vor der Herausforderung, eine hohe Zuverlässigkeit (engl. dependability) ihres Endproduktes erreichen zu müssen [113]. Die Zuverlässigkeit beschreibt die Fähigkeit eines Systems, einen Dienst zu liefern, dem man zu Recht trauen kann [25] und ist somit ein entscheidendes Qualitätsmerkmal bei der Fahrzeugentwicklung. Das Merkmal Zuverlässigkeit setzt sich aus Teilmerkmalen wie der Betriebssicherheit (engl. safety) zusammen [25, 191]. Diese entspricht der Wahrscheinlichkeit für das Nichteintreten eines Unfalls über eine bestimmte Zeitperiode [140, 216] und nimmt einen besonderen Stellenwert in der Entwicklung von Fahrzeugen ein. Wird die Betriebssicherheit verletzt, kann der Fahrzeughersteller für den entstandenen Schaden in Haftung genommen werden. Aufgrund dessen und wegen des Ziels, die Sicherheit der Kunden zu gewährleisten, sind Hersteller gewillt, einen hohen Grad an Betriebssicherheit zu erreichen. Hierzu wird in einer frühen Phase die funktionale Sicherheit (engl. functional safety) eines Fahrzeuges betrachtet. Diese ist im Allgemeinen durch die Norm IEC 61508 [47] beschrieben und im Speziellen für Kraftfahrzeuge mit der Norm ISO 26262 [99] abgeleitet. Letztere beschreibt Prozesse und Maßnahmen zur Sicherstellung der funktionalen Sicherheit von sicherheitsbezogenen elektrischen/elektronischen/programmierbaren Systemen (E/E-Systeme) in Kraftfahrzeugen.

Das Umsetzen dieser Prozesse und Maßnahmen wird indirekt durch den § 4 ProdHaftG gefordert. So müssen Hersteller im Schadensfall nachweisen, dass bei der Entwicklung des Produktes eine anerkannte Methode zur sicheren Produktentwicklung und Produktherstellung angewandt wurde [118]. Ein Schadensfall liegt insbesondere dann vor, wenn ein oder mehrere elektronische Systeme eines Fahrzeuges ausfallen, das Fahrzeug aufgrund dessen vom spezifizierten Verhalten abweicht und Personenschäden entstehen. Um dieses Risiko zu mindern, wird zu Beginn der E/E-Systementwicklung eine Gefahrenanalyse durchgeführt. Ziel hierbei ist das Auffinden möglicher Gefährdungen (engl. hazards), die durch ein Fehlverhalten entstehen können, sowie der dazugehörigen

Ursachen. Basierend auf diesen Ergebnissen wird anschließend ein Risikowert bestimmt, der zur Priorisierung der nachfolgenden Entwicklungsschritte dient. Konkret werden durch deduktive oder induktive Analysemethoden potenzielle Verhaltensabweichungen und die dazugehörigen Ursachen in E/E-Systemen identifiziert und bewertet. Diese Methoden berücksichtigen allerdings nur Verhaltensabweichungen die auf Fehlern beruhen, wohingegen Cyber-Angriffe eine böswillige Intention darstellen. Zahlreiche Angriffe auf Fahrzeuge haben dabei gezeigt, dass die Betriebssicherheit (Safety)[1] von modernen Fahrzeugen negativ beeinflusst werden kann. Cyber-Security ist daher ein weiterer Aspekt der Zuverlässigkeit, der aber vom Schutz gegen intelligente Angreifer und nicht von zufälligen Fehlern abhängt. Aufgrund dessen sind Safety-Methoden nicht direkt übertragbar. Dies erfordert neue Methoden, welche die Zuverlässigkeit von Systemen auch gegen systematische Angreifer sicherstellen können.

1.1 Problemstellung

Zur Sicherstellung der Betriebssicherheit ergibt sich die Notwendigkeit, neben Quellen für ein unbeabsichtigtes Abweichen von der Spezifikation (Safety) auch Quellen für ein beabsichtigtes Abweichen (Cyber-Angriff) betrachten zu müssen. Diese Aussage wird anhand der in der Forschung gezeigten Cyber-Angriffe und den in dieser Arbeit durchgeführten Untersuchungen [5] untermauert. So konnte in dieser Arbeit ein safety-kritisches Fahrzeugsystem (Airbag-System) angegriffen und eine Gefährdung für die Fahrzeuginsassen erreicht werden [7], was die Abhängigkeit der Safety von Security zeigt. Die Abhängigkeiten sind hierbei vielschichtig, sodass sich Safety und Security manchmal ergänzen, in einem anderen Fall gegenseitig einschränken können. Ein Beispiel hierfür ist die Maßnahmenumsetzbarkeit, bei der eine Safety-Maßnahme eine Security-Maßnahme aussetzt oder deren Wirksamkeit verringert. So schützt beispielsweise der Einklemmschutz des elektrischen Fensterhebers vor einem tödlichen Einklemmen zwischen Fenster und Türrahmen. Dabei wird allerdings das vollständige Schließen des Fensters verhindert, was sich ein Angreifer für einen Zugang zum Fahrzeuginneren zunutze machen kann. Safety- und Security-Abhängigkeiten können somit Zielkonflikte hervorrufen, die häufig nicht vermeidbar sind und aufgelöst werden müssen. Demgegenüber stehen allerdings auch Synergieeffekte, die ausgenutzt werden sollten. So können beispielsweise Safety-Maßnahmen Security-Angriffe verhindern oder abschwächen. Konkret kann etwa eine Plausibilisierung von Sensorsignalen das böswillige Manipulieren der übertragenen Signale aufdecken und den Cyber-Angriff abwenden [8].

Ähnlich verhält es sich mit jenen Situationen, die in der Anforderungsanalyse betrachtet werden, eine Gefährdung für Leib und Leben darstellen und grundsätzlich durch Verletzen der Safety oder Security ausgelöst werden können. Damit diese Situationen

[1] Im Folgenden wird der Begriff Safety als Synonym für die Betriebssicherheit und der Begriff Security als Synonym für die Angriffssicherheit verwendet.

nur einmalig und nicht mehrfach analysiert werden, ist eine kombinierte Betrachtung beider Aspekte erforderlich. Die Analyse der Zielkonflikte und Synergieeffekte kann dabei im gesamten Entwicklungszyklus der Cyber-physischen Systeme (CPS) erfolgen: Anforderungsanalyse, Konzeption, Implementierung und Testing. Im Rahmen dieser Arbeit wird allerdings ein Fokus auf die Anforderungsanalyse und konkret auf gemeinsame Gefährdungs- und Bedrohungsanalysen (engl. threat analysis) gelegt. Letztere werden in der traditionellen Informationstechnik verwendet, um Cyber-Bedrohungen zu analysieren. Ziel ist hierbei, das Auffinden und Bewerten potenzieller Situationen oder Handlungen, die einen Vermögenswert (engl. asset) – wie beispielsweise Safety – negativ beeinflussen können. Durch die Identifikation der Cyber-Bedrohungen ist es im Anschluss möglich, Gegenmaßnahmen zu ergreifen und die Wahrscheinlichkeit für das Eintreten der Cyber-Bedrohungen, mit den daran geknüpften Konsequenzen, zu minimieren.

Das wissenschaftliche Umfeld wartet bereits mit Ansätzen zu gemeinsamen Gefährdungs- und Bedrohungsanalysen auf. Eine Analyse dieser Ansätze zeigt allerdings, das eine umfassende Wiederverwendung von Safety-Analyse-Ergebnissen für Security-Analysen in der Fahrzeug-Domäne zum Zeitpunkt dieser Arbeit nicht gegeben ist (Abschnitt 3.4.1). Begründet hierauf und auf Erkenntnissen eigener Cyber-Angriffe auf Fahrzeuge [3, 5, 7] lässt sich schließen, dass weitere Synergieeffekte ausgenutzt werden können, um weitere safety-relevante Cyber-Bedrohungen identifizieren zu können und damit die Betriebssicherheit zu erhöhen. Dies führt zu der übergeordneten Frage:

Hauptforschungsfrage (HF)**.** *Wie lassen sich Bedrohungsanalysen aus der Informationssicherheit (IT) mit Gefährdungsanalysen kombinieren, um die Betriebssicherheit von Personenkraftwagen zu steigern?*

Zur Beantwortung dieser Fragestellung sind mehrere Aspekte von Relevanz. Zum einen müssen gewachsene Strukturen aus der Safety beachtet werden, um eine kombinierte Methodik etablieren zu können. Zum anderen kann angenommen werden, dass Security-Analysten nur wenige Berührungspunkte mit der Safety-Domäne haben, sodass Safety-Ziele oft nicht identifiziert werden. So werden Cyber-Bedrohungen typischerweise zuerst mit dem Fokus auf die Verletzung informationstechnischer Schutzziele – wie Integrität, Authentizität, Geheimhaltung, Verfügbarkeit, etc. – identifiziert. Erst anschließend wird betrachtet, ob ebenso eine Beeinflussung der Safety vorliegt. In diesem Fall obliegt es dem Security-Analysten zu entscheiden, ob die vorliegende Cyber-Bedrohung safety-relevant ist oder nicht. Eine kombinierte Safety-Security-Bedrohungsanalyse müsste somit in der Lage sein, neben der Identifikation klassischer Cyber-Bedrohungen auch safety-relevante Cyber-Bedrohungen identifizieren zu können. Die zugrundeliegende Methodik muss demnach feststellen können, welche Cyber-Bedrohungen zu einem safety-kritischen Fahrzeugverhalten führen können, wodurch sich die Frage stellt:

Teilforschungsfrage 1 (TF1)**.** *Wie können safety-relevante Cyber-Bedrohungen systematisch identifiziert werden?*

Bereits der Aufwand für eine eigenständige Security-Analyse ist hoch und die Koppelung mit Safety-Fragestellungen lässt die Komplexität weiter ansteigen. Die Anzahl von safety- und security-relevanten Bedrohungen hängt dabei von der Kombinatorik vorhandener Systeme und deren Informationsaustausch ab. Aufgrund dessen kann je nach Größe des betrachteten Systemverbundes eine große Anzahl von Bedrohungen identifiziert werden. Im Hinblick auf Deadlines und beschränkte Ressourcen in der automotiven Domäne stellt die gemeinsame Betrachtung von Safety- und Security-Situationen somit einen Zielkonflikt dar. Es ist daher erforderlich, dass die kombinierte Methodik mit der gesteigerten Anzahl von Cyber-Bedrohungen umgehen kann. Hierzu muss ein Vorgehen entwickelt werden, welches den Aufwand für den Anwender in einem praktikablen Umfang hält, was zur Frage führt:

Teilforschungsfrage 2 (TF2)**.** *Wie kann der gesteigerte Aufwand für den Anwender reduziert werden, der sich durch die gemeinsame Betrachtung von Safety und Security ergibt?*

Zeitliche Fristen und Ressourcenknappheit stellen nicht nur bei Bedrohungsidentifikation einen Zielkonflikt dar, sondern ebenso bei der Umsetzung von Maßnahmen. Letztere verhindern oder schwächen die Realisierung einer Cyber-Bedrohung (Cyber-Angriff) ab und sind das primäre Ergebnis des Security-Konzeptes, das an die Bedrohungsanalyse anknüpft. Für die Behandlung der Cyber-Bedrohungen muss allerdings eine strukturierte Entscheidungsgrundlage gegeben werden, um jene Bedrohungen auszuwählen, welche die höchste Relevanz aufweisen. Eine neue Methodik muss somit in der Lage sein, die identifizierten Cyber-Bedrohungen priorisieren zu können, um verfügbare Ressourcen optimal einzusetzen. Hierzu muss die Methodik die Safety-Relevanz der Cyber-Bedrohungen in die Priorisierung einbeziehen, womit sich die letzte Frage stellt:

Teilforschungsfrage 3 (TF3)**.** *Wie lässt sich eine Priorisierung und Risikobewertung identifizierter Cyber-Bedrohungen hinsichtlich Safety verwirklichen?*

1.2 Lösungsansatz

Für die Beantwortung von **TF1** und der Fragestellung, wie safety-relevante Cyber-Bedrohungen systematisch identifiziert werden können, steht insbesondere die gegenseitige Beeinflussung von Safety und Security im Fokus. So lässt sich durch böswillige Manipulationen der verbauten CPS das Fahrzeugverhalten negativ beeinflussen, wodurch die Betriebssicherheit gefährdet ist. Diese Abhängigkeit wird in Abbildung 1.1 dargestellt und zeigt sogleich eine Schnittstelle zwischen beiden Studienfeldern. So können Abweichungen des Systemverhaltens (a) entweder durch Fehlfunktionen (b)

oder durch Cyber-Angriffe (c) ausgelöst werden. Cyber-Angriffe können somit die gleichen Verhaltensabweichungen wie Fehlfunktionen auslösen. In der Kombination mit der Gegebenheit, dass in einer Safety-Analyse die Verhaltensabweichungen identifiziert werden, können diese als Eingangsinformation für eine kombinierte Cyber-Bedrohungsanalyse dienen.

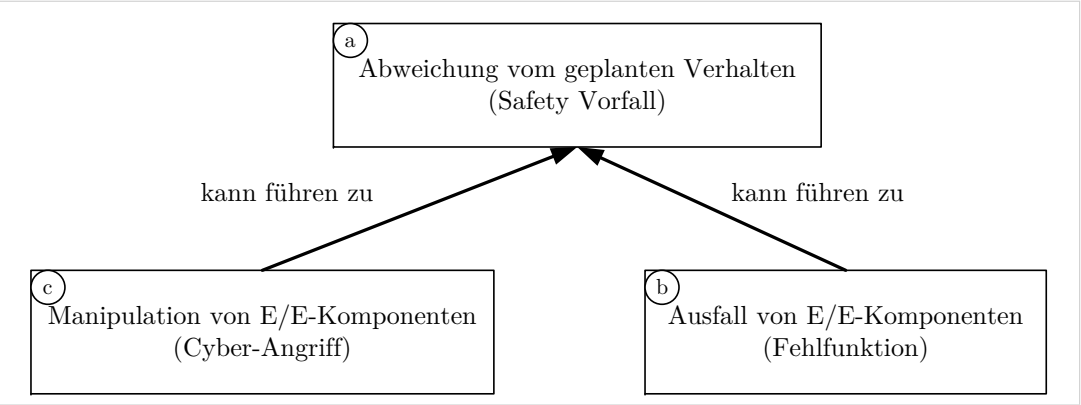


Abbildung 1.1: Gemeinsame Safety-Konsequenzen durch Cyber-Angriffe oder Fehlfunktionen.

Diese Gefährdungen können in die Bedrohungsanalyse übernommen werden, um zu überprüfen, durch welche Cyber-Angriffe sie ausgelöst werden. Der Ansatz entspricht einem deduktiven Vorgehen, da von bestehenden Auswirkungen – den Gefährdungen – aus die zugehörigen Ursachen (Cyber-Bedrohungen) gesucht werden. Basierend auf diesem Konzept können Arbeitsergebnisse aus einer Safety-Analyse für eine kombinierte Bedrohungsanalyse weiterverwendet werden. Die Arbeitsergebnisse können hierbei in zwei Formen vorliegen: Informationssammlungen wie Entwicklungsdokumente oder Datenpakete sowie die Erfahrung, die ein Analyst bei der Durchführung von Betriebssicherheitsanalysen sammelt. Das sind unter anderem:

- Die E/E-Architektur des Fahrzeuges mit den jeweiligen Steuergeräten,
- die Funktionalitäten des Systemverbundes und deren Auswirkungen auf die Safety,
- der Informationsfluss zwischen den Systemkomponenten,
- die identifizierten Gefährdungen und deren Ursachen,
- die Erfahrung über Wirkketten, die sich durch Interaktion der Komponenten ergeben.

Von besonderer Bedeutung ist hierbei das Verständnis über die Wirkketten zwischen den Systemkomponenten. Anhand dieser lassen sich die Auswirkungen durch Manipulationen einzelner Systeme auf den restlichen Systemverbund identifizieren. Um diese

Artefakte erfassen und nutzen zu können, wurde in dieser Arbeit ein semiformales Framework entwickelt, das einen Ingenieur beim Auffinden von safety-relevanten Cyber-Bedrohungen unterstützt. Das Framework nennt sich Security Guidewords Method (SGM) [2] und führt den Ingenieur anhand eines strukturierten Vorgehens durch den Analyseprozess. Es versetzt Safety-Ingenieure in die Lage, neben dem Fehlerverhalten auch böswillige Manipulationen durch Verletzungen der Security zu identifizieren und etabliert eine gemeinsame Schnittstelle zwischen Safety und Security. Die Unterstützung ist in der Weise gestaltet, dass Begrifflichkeiten aus der Security-Domäne in die Terminologie der Safety-Domäne übersetzt werden und damit eine Schnittstelle zwischen beiden Welten entsteht. Das Framework bereitet hierzu die in der Safety-Analyse identifizierten Gefährdungen sowie die dazugehörigen Ursachen für eine anknüpfende Cyber-Bedrohungsanalyse auf. Damit versetzt die Methodik Safety-Ingenieure in die Lage, Cyber-Bedrohungen zu identifizieren, welche die Safety gefährden können. Dies bedeutet jedoch nicht, dass Security- durch Safety-Ingenieure ersetzt werden. Stattdessen unterstützen Safety-Ingenieure mit der SGM die Bedrohungsanalyse, indem sie safety-relevante Cyber-Bedrohungen aufdecken. Außerdem werden die in der Safety-Analyse identifizierten Informationsflüsse formalisiert und für die Security-Analyse vorbereitet. Eine Evaluation der SGM in Form eines Experiments [2] zeigt hierbei, dass Safety-Ingenieure in der Lage sind, Cyber-Bedrohungen hinsichtlich der Safety zu identifizieren, wenn die SGM eingesetzt wird (siehe Kapitel 5). Darüber hinaus zeigt die Analyse der Ergebnisse, dass für Security-Ingenieure die Identifikation und Bewertung von safety-bezogenen Cyber-Bedrohungen eine Herausforderung darstellen kann. So zeigten Security-Ingenieure tendenziell schlechtere Ergebnisse im Experiment als die Safety-Ingenieure, die das Framework einsetzten. Dies unterstützt die These, dass Safety-Ingenieure einen sinnvollen Beitrag mit der SGM leisten können.

Für die Beantwortung von **TF2** und der Reduzierung des Aufwands durch die gesteigerte Anzahl von Cyber-Bedrohungen stehen Synergieeffekte und eine Automatisierung der Analyse im Fokus. Bereits die Wiederverwendung und Formalisierung von Safety-Ergebnissen für die Security-Analyse – durch die SGM – stellt einen Synergieeffekt und die Reduzierung des Arbeitsaufwands dar. Die SGM erleichtert außerdem die Durchführung der Cyber-Bedrohungsanalyse, indem sie den Analysten strukturiert durch die Analyse leitet. So zeigt eine Auswertung des Experiments [2], dass sich Safety-Ingenieure mit der Unterstützung des Frameworks weniger gefordert fühlten, obwohl sie im Durchschnitt eine höhere Produktivität und Sensitivität im Experiment erreichten (Anhang B.2.2). Dieser Synergieeffekt kann in einer weiteren Entwicklungsphase der CPS ebenso eine Reduktion des Aufwands ermöglichen. So zeigt die Einbettung der SGM in einen Penetrationstest [3], dass safety-kritische Testfälle strukturiert identifiziert werden können (Kapitel 6). Die Ausführung der abgeleiteten Testfälle auf ein Airbagsystem, welches in vielen gängigen Fahrzeugen eingesetzt wird, deckte eine safety-kritische Schwachstelle [52, 7] auf, die das Zünden von Airbags ermöglicht.

Bezogen auf eine Automatisierung der Analyse, wird eine automatisierte Priorisierung der Cyber-Bedrohungen vorgestellt. So ist das Framework in der Lage, anhand von Informationen aus einer gegebenen E/E-Architektur und deren Datenflüssen, die Eintrittswahrscheinlichkeiten für die Cyber-Bedrohungen zu berechnen. Hierzu wird die Systembeschreibung, die für die Safety-Analyse bereits benötigt wurde, durch den Safety-Analysten mit bekannten Informationsflüssen angereichert und ein Datenflussdiagramm erzeugt. Dazu werden Entitäten aus der automotiven Domäne, wie Steuergeräte und Netzwerktechnologien, in Form einer Bibliothek bereitgestellt. Jedes verfügbare Element ist dabei eine Instanz des definierten Modells, was die Analyse für den Anwender vereinfacht. Darauf basierend generiert das SGM-Framework automatisiert Angriffspfade (Kapitel 7). Letztere repräsentieren jene Wege, die ein Angreifer in der vorliegenden E/E-Architektur wählen kann, um eine Gefährdung auszulösen.

Um **TF3** und die Fragestellung nach einer safety-bezogenen Priorisierung der Cyber-Bedrohungen zu beantworten, formalisiert und verwendet die SGM weitere Safety-Arbeitsergebnisse. Konkret wird die Schwere des Safety-Schadens (engl. severity) für die Priorisierung weiter verarbeitet. Die Bestimmung des Schadenswertes für eine Cyber-Bedrohung folgt damit analog der Bewertung in der Safety-Analyse und ermöglicht eine identische Schadensbewertung für den Security-Prozess, die mit ISO 21434 [97] konform ist. Außerdem können so safety-bezogene Schadenswerte in Security-Analysen einfacher nachvollzogen werden, was ein gemeinsames Verständnis zwischen den Safety- und Security-Ingenieuren etabliert. Hinsichtlich der Bestimmung von Eintrittswahrscheinlichkeiten für Cyber-Bedrohungen können keine vergleichbaren Artefakte übernommen werden. Das ist mit der Tatsache zu begründen, dass Metriken zur Bestimmung von Eintrittswahrscheinlichkeiten in der Safety- und Security-Domäne grundlegend verschieden sind. So werden für Safety-Probleme Metriken wie die Ausfallrate von Komponenten verwendet, die rein hardwarebasiert sind. Für Cyber-Bedrohungen hingegen wird die Gesamtkomplexität für die Durchführung des Angriffs verwendet [100]. Zu dessen Bestimmung greift das SGM-Framework auf die erzeugten Angriffspfade und ein Angreifermodell zurück. Die Angriffspfade repräsentieren hierbei eine Verkettung von Schwachstellen, die der Angreifer ausnutzen muss. Das Angreifermodell dagegen beschreibt den Aufwand – für den Angreifer – zur Erreichung der jeweiligen Schwachstellen. Für die Eintrittswahrscheinlichkeiten der Schwachstellen sind Metriken des standardisierten Schwachstellen-Bewertungssystems CVSS für die automotive Domäne adaptiert. Diese werden einmalig bestimmt, um den Ingenieuren die Anwendung der adaptierten Metriken weiter zu vereinfachen. Die Werte werden anhand bekannter Angriffe abgeleitet, mittels einer Taxonomie [13] einheitlich beschrieben und stehen in Form einer Datenbasis [5] zur Verfügung (Abschnitt 2.5.4). Mit diesem Vorgehen und der Wiederverwendung des Schadenswerts für die ausgelöste Gefährdung können für Cyber-Bedrohungen einheitliche Eintrittswahrscheinlichkeiten vergeben werden.

1.3 Verwandte Arbeiten und Abgrenzung

Die gemeinsame Analyse von Safety und Security ist kein neues Forschungsfeld, sodass sich mehrere Ansätze in verschiedenen Domänen etabliert haben. Die Untersuchung der bestehenden Ansätze in Kapitel 3 zeigt jedoch deutliche Unterschiede bei der Modellierung von Safety und Security-Artefakten für kombinierte Analysen. Analysetechniken aus dem Studienfeld der CPS weisen dabei den höchsten Detailgrad bei der Modellierung des Systems und der Bedrohungen auf. Insbesondere die STPA-SafeSec-Methodik [70] zeigt einen hohen Detaillierungsgrad der Analyse, die auf einen systemtheoretischen Ansatz setzt und Safety-Probleme durch eine Abstraktion des Systems auf Regelkreise bestehend aus Sensoren, Aktoren und Controllern, identifiziert (Abschnitt 3.2.1). Sie entspricht außerdem der einzigen Methode, die erzeugte Safety-Artefakte für die Security-Analyse weiter verwendet. Diesem stehen jedoch ein hoher Aufwand bei der Bedrohungsmodellierung und ein hoher Detailgrad der notwendigen Eingangsinformationen gegenüber. Speziell die Notwendigkeit von detaillierten Eingangsinformationen, die in frühen Entwicklungsphase nicht gegeben sind, verhindern die Anwendung der Methodik in der frühen Entwicklung von Fahrzeugsystemen. Ein Einsatz der hier vorgestellten SGM ist hingegen früher möglich, da ein geringerer Detailgrad der Eingangsinformationen erforderlich ist. Darüber hinaus orientiert sich die SGM an einem bekannten Vorgehen aus der Safety-Domäne, was die Anwendung der Methodik erleichtert (Abschnitt 4.3). In ähnlicher Weise geht der Combined Harm Assessment of Safety and Security for Information Systems (CHASSIS)-Ansatz [157] bei der Analyse vor (Abschnitt 3.2.2). Hier wird der Analyst anhand von bekannten Techniken durch die Bedrohungsanalyse geleitet. Der Ansatz eignet sich für frühe Entwicklungsphasen. Allerdings werden bei CHASSIS die Safety und Security getrennt voneinander analysiert. Im Zuge dessen werden identifizierte Gefährdungen nicht als Eingangsinformation in die Security-Analyse übernommen. Das kann zur Situation führen, dass nicht geprüft wird, ob Gefährdungen ebenfalls durch Verletzungen der Security ausgelöst werden können. Diese Eigenschaften zeigen ebenfalls jene Methoden, die für die Analyse von Fahrzeugsystemen vorgeschlagen sind (Abschnitt 3.3). Im Gegensatz dazu übernimmt der in dieser Arbeit vorgeschlagene Ansatz die Gefährdungen aus der Safety-Analyse und analysiert, inwieweit durch Cyber-Bedrohungen die Safety beeinflusst werden kann, was dem **ersten Beitrag** der Arbeit entspricht und mit dem Papier [2] veröffentlicht wurde. Daneben ermöglicht die Kombination aus spezifischen Leitwörtern und der bereitgestellten Vorlage für die Bedrohungsanalyse eine weitere Erfassung security-relevanter Artefakte. So identifizieren die vorgeschlagenen Leitwörter, welche Angriffstechniken eingesetzt werden können. Die SGM-Vorlage erfasst außerdem die verwundbaren Stellen im Systemverbund und die betroffenen Systeme. Diese Informationen fließen anschließend in die nachgelagerte Risikobewertung ein und ermöglichen es erste Security-Anforderungen abzuleiten.

Bei der Betrachtung, ob bestehende Ansätze in der Lage sind, mehrstufige Angriffe zu identifizieren, zeigte sich, dass Methoden aus dem CPS-Umfeld grundsätzlich dazu fähig sind. Bei den dem Autor bekannten Methoden aus dem Fahrzeugumfeld ist hingegen nur der EVITA-Ansatz [84] fähig, mehrstufige Angriffe zu identifizieren (Abschnitt 3.3.1). Bezogen auf die Gegebenheit, dass bekannte Cyber-Angriffe auf Fahrzeuge überwiegend mehrere Angriffsschritte aufweisen, ist dies als nachteilig zu bewerten. Bei der weiteren Untersuchung jener Methoden, die zur Identifikation mehrstufiger Angriffe fähig sind, fällt auf, dass keine Werkzeuge zur Unterstützung bereitgestellt werden. Dies erfordert vom Analysten eine manuelle Erzeugung mehrstufiger Angriffspfade, was zu einem erhöhten Aufwand bei der Bedrohungsanalyse führt. Demgegenüber steht das in dieser Arbeit vorgestellte Framework zur automatisierten Generierung mehrstufiger Angriffspfade. Erreicht wird dies durch ein formales Modell, das aus erfassten Eingangsinformationen und einem Metamodell aufgebaut wird und sich mit formalen Methoden überprüfen lässt. Die hierzu eingesetzten Konzepte basieren auf der eigenen Veröffentlichung [1]. Sie beziehen die Steuergerätevernetzung, die Systemkomponenten, die Ergebnisse der SGM sowie Security-Schwachstellen und deren Auswirkungen auf informationstechnische Vermögenswerte in die Analyse mit ein. Außerdem werden bekannte Schwachstellen auf Fahrzeuge eingebunden. Ein Überprüfen des erzeugten Modells auf die Verletzung bestimmter Security-Eigenschaften mit Model-Checking Techniken ermöglicht das automatisierte Erzeugen mehrstufiger Angriffspfade (Abschnitt 7.4). Diese sind bezogen auf das geprüfte Model vollständig und können wiederholbar erzeugt werden. Das präsentierte Framework mindert damit den Analyseaufwand, was dem **zweiten Beitrag** der Arbeit entspricht.

Neben der Bereitstellung von Werkzeugen zur Unterstützung der Bedrohungsmodellierung trägt auch das Wiederverwenden von historischem Wissen zur Reduktion des Analyseaufwands bei. Die Untersuchung der Methoden auf dieses Kriterium zeigt jedoch, das keine der analysierten Methodiken bekannte Security-Probleme in die Analyse mit einbindet. Bezogen auf die Situation, dass in der Fahrzeug-Domäne Teilsysteme über mehrere Generationen hinweg eingesetzt werden, ist das fehlende Wiederverwenden von bekannten Security-Problemen ein Defizit. Konkret erhöht es den Aufwand bei der Durchführung der Bedrohungsanalyse, weil bekannte Security-Probleme wiederkehrend identifiziert und damit keine Synergieeffekte ausgenutzt werden. Bezogen auf jene Ansätze, die eine umfassende Bedrohungsmodellierung (STPA-SafeSec [70], CHASSIS [84], Six-Step Model [170], EVITA [84]) aufzeigen – allerdings einen hohen Aufwand erfordern – kann das Einbeziehen von bekannten Security-Problemen den Aufwand reduzieren und die Analyseergebnisse vollständiger gestalten. Das hier vorgestellte Framework bindet daher bekannte Schwachstellen in die Bedrohungsanalyse ein. Hierzu wird eine Sammlung [5] und Klassifikation [13] von Security-Schwachstellen in Fahrzeugsystemen präsentiert (Abschnitt 7.2). Erstere dient beim Erzeugen des formalen Modells als Datenbasis für bekannte Schwachstellen und letztere, um zukünf-

tige Schwachstellen einheitlich in die Datensammlung [5] einpflegen zu können. Die Klassifikation wird außerdem aktiv in der Security-Gemeinschaft eingesetzt, die Agentur der Europäischen Union für Cybersicherheit (ENISA) verweist beispielsweise in einem Leitfaden [60, Seite 77] für Security in intelligenten Fahrzeugen auf diese. Damit leistet das vorgestellte Framework einen Beitrag für das Einbinden von bekannten Security-Schwachstellen in Bedrohungsanalysen, was den **dritten Beitrag** der Arbeit darstellt.

Neben der Bedrohungsmodellierung ist auch die Risikoanalyse ein zentraler Punkt einer gesamtheitlichen Bedrohungsanalyse. Diese ermöglicht die Bewertung und Priorisierung identifizierter Bedrohungen für nachfolgende Entwicklungsphasen. Eine Untersuchung der Ansätze aus dem Umfeld der CPS zeigt hingegen, dass alleinig die Failure Mode, Vulnerabilities and Effect Analysis (FMVEA) [41] einen Vorschlag für eine Risikoanalyse und die dazu notwendigen Metriken präsentiert (Abschnitt 3.2.4). In der Fahrzeugdomäne präsentieren zwei Ansätze (EVITA [84] und HEAling Vulnerabilities to ENhance Software Security and Safety (HEAVENS) [146]) eine Risikoanalyse mit adaptierten Metriken aus der ISO 15408 [100]. Ein weiterer Ansatz (Security-Aware Hazard and Risk Analysis (SAHARA) [128]) schlägt seine eigene Metrik für die Bedrohungsanalyse vor (Abschnitt 3.3.3). Bei einer kritischen Auseinandersetzung mit den präsentierten Metriken zeigt sich jedoch, dass die dort verwendeten Kategorien einen größeren Diskussionsspielraum aufweisen. So kann beispielsweise für die Kategorie *Angreifer Motivation* nur mit hoher Unsicherheit bestimmt werden, welche Motivation ein möglicher Angreifer verfolgen wird. Die Metriken weisen damit einen hohen Grad an Subjektivität auf, was eine Vergleichbarkeit von Risikoanalyseergebnissen erschwert. Dem gegenüber steht das Vorgehen bei der integrierten Risikoanalyse des SGM-Frameworks, welches auf das CVSS-Konzept [51] setzt. Hierbei werden Schwachstellen einmalig und in einheitlicher Form bewertet. Das Vorgehen hierzu ist in der Informationstechnik am weitesten verbreitet und anerkannt [87, 135, 136]. Als Ergebnis ergibt sich eine Eintrittswahrscheinlichkeit, die an die Schwachstelle angeheftet ist, sich nicht mehr ändert und in einer Datenbank abgelegt wird. Somit kann das SGM-Framework stets die identischen Schwachstellen den automatisiert erzeugten Angriffspfaden zuordnen. Abschließend wird das Risiko, durch Einbeziehung der Schadenswerte aus der Gefährdungsanalyse und der Eintrittwahrscheinlichkeit für den gesamten Angriffspfade bestimmt (Abschnitt 7.6.3). Da das SGM-Framework bei identischen Eingaben (ausgefüllte SGM-Vorlage, E/E-Architektur und Schwachstellendatenbank) stets identische Risikowerte berechnet, ergibt sich ein Vorteil zu Methoden die keine einheitliche Bewertung von Schwachstellen aufweisen und durch manuelle Einflüsse die Möglichkeit zu abweichenden Risikowerten aufweisen. Mit dem vorgestellten SGM-Framework ist es somit möglich, bei sich wiederholenden Analysen vergleichbare Risikoergebnisse zu erzeugen, was dem **vierten Beitrag** der Arbeit entspricht.

Den **fünften Beitrag** der Arbeit stellt die durchgeführte empirische Studie im Bereich Safety und Security sowie die Übertragung des Konzepts zur Bedrohungsmodellierung auf Penetrationstests dar. Erstere zeigt – neben der Evaluierung der Microsoft Spoofing, Tampering, Information Disclosure, Denial of Service and Elevation of Privilege (STRIDE) Methodik [176] – als einzige Studie in der Literatur eine empirische Auswertung einer Fallstudie (Kapitel 5). Hiermit wird es möglich, den gezeigten SGM-Ansatz umfassender bewerten zu können.

Darüber hinaus wurde das in dieser Arbeit vorgeschlagene Vorgehen zur Bedrohungsmodellierung in einem Penetrationstest eingesetzt, da hier ebenfalls eine Bedrohungsmodellierung sinnvoll ist, was mit dem Papier [3] veröffentlicht ist. Genauer gesagt wird gezeigt, wie mit der SGM Bedrohungen in der Phase der Informationsbeschaffung modelliert werden können, die ein Teil eines Penetrationstests ist. Hierzu zeigt Kapitel 6, wie der SGM-Ansatz mit dem Penetration Testing Execution Standard (PTES) [200] zur Durchführung eines Penetrationstests kombiniert wird. Eine Evaluierung dieses Vorgehens mit einem Airbag-Steuergerät deckte eine safety-kritische Security-Schwachstelle auf, die ein unerlaubtes Auslösen der Airbag-Ladungen ermöglicht und mit [7] der Forschungsgemeinschaft präsentiert wurde, sowie eine CVE-Kennung [52] erhielt, was dem **sechsten Beitrag** der Arbeit entspricht.

Zusammengefasst leistet die Arbeit die folgenden Beiträge:

1. In der Literatur wurde bisher keine ausführliche Gegenüberstellung kombinierter Safety- und Security-Analysemethodiken gezeigt. Diese Arbeit schließt die Lücke mit einer Analyse und Bewertung kombinierter Safety- und Security-Analysen in Kapitel 3.

2. Die Literatur zeigt bis heute keine detaillierte Aufarbeitung der Abhängigkeiten zwischen Safety und Security für CPS. Diese Arbeit schließt die Lücke durch ein aufgestelltes Metamodell in Kapitel 4.

3. Mit denen in der Literatur gezeigten Bedrohungsanalyse-Methodiken ist es nicht möglich, Safety-Artefakte für Security-Analysen umfänglich wiederzuverwenden. Diese Arbeit schließt die Lücke mit der vorgestellten Bedrohungsanalyse-Methodik in Kapitel 4.

4. Die Literatur zeigt bis zu diesem Zeitpunkt, kein Vorgehen wie eine kombinierte Safety und Security Bedrohungsanalyse-Methodik empirisch bewertet werden kann. Diese Lücke wird durch eine empirische Bewertung der vorgestellten Methodik in Kapitel 5 geschlossen.

5. In der Literatur wird bis zu diesem Zeitpunkt nicht betrachtet, wie die Informationsbeschaffung bei Penetrationstest durch neue Ansätze gesteigert werden kann. Die Arbeit schließt diese Lücke durch die Integration der vorgestellten

Bedrohungsanalyse-Methode in einen Penetrationstest, mit der Aufdeckung einer Schwachstelle in Airbag-Systemen, in Kapitel 6.

6. Die Literatur stellt zum aktuellen Zeitpunkt keine Klassifizierung von Cyber-Angriffen auf Fahrzeuge zur Verfügung. Diese Lücke wird durch die Bereitstellung einer Taxonomie von Cyber-Angriffen auf Fahrzeuge und der dazugehörigen Sammlung von Angriffen in Kapitel 7 geschlossen.

7. Mit den in der Literatur gezeigten Ansätzen ist es zurzeit nicht möglich eine automatisierte Risikobewertung, safety-kritischer Schwachstellen für Fahrzeuge durchführen zu können. In Kapitel 7 wird dieses Lücke mit einem Ansatz und einem Prototyp zur automatisierten Modellierung von Cyber-Bedrohungen in Fahrzeugen und deren Risikobewertung geschlossen.

1.4 Aufbau der Arbeit

Die Struktur der vorliegenden Arbeit besteht aus sieben inhaltlichen Teilen, welche mit Abbildung 1.2 grafisch dargestellt sind.

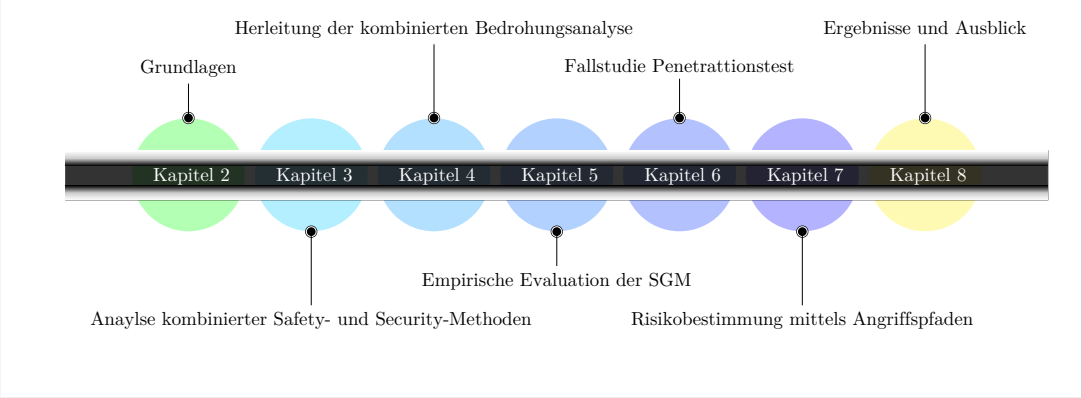


Abbildung 1.2: Struktur der vorliegenden Arbeit mit den sieben Teilabschnitten und der Zuordnung zu den jeweiligen Kapiteln. Die blauen Kreise repräsentieren Kapitel, die auf Konzeption und Erweiterung der Methodik bezogen sind.

In Kapitel 2 werden die relevanten Begrifflichkeiten definiert, das technische Umfeld der Arbeit aufgezeigt und die notwendigen Entwicklungsprozesse in der automotiven Domäne erläutert. Letzteres ist wesentlich, um die Arbeit in den korrekten Kontext der Fahrzeugentwicklung einordnen zu können. Im Anschluss wird in Kapitel 3 der Stand der Technik erläutert. Hier werden die etablierten Cyber-Bedrohungsanalysen für automotive Systeme und Cyber-physischen Systeme (CPS) aufgezeigt und beschrieben, welche Lücken in der Literatur durch diese Arbeit geschlossen werden. Die Methoden werden außerdem in Abschnitt 3.4.1 anhand festgelegter Kriterien bewertet und in Beziehung zur in dieser Arbeit vorgeschlagenen Methodik gesetzt. Die gewonnenen Ergebnisse dienen anschließend in Kapitel 4 als Grundlage für die Konzeption der SGM. In Kapitel 5, wird die SGM Methode empirisch durch ein Experiment mit Safety und

Security Ingenieuren evaluiert und die Ergebnisse werden diskutiert. Kapitel 6 zeigt anknüpfend eine Integration der Methodik in einen Penetrationstest. Hierbei dient der Ansatz zur Modellierung der Angriffsfläche des Testgegenstands, um Testfälle abzuleiten. An dieser Stelle werden Synergieeffekte der SGM ausgenutzt, um den Aufwand des Penetrationstests zu verringern. Als Ergebnis dieser Testfälle für eine Airbag-Steuergerät aus der Serie, konnte eine safety-kritische Schwachstelle identifiziert werden, die das Auslösen der Airbags ermöglicht. Mit Kapitel 7 wird anschließend ein Ansatz zur Risikobewertung mittels Angriffspfaden vorgestellt. Das Vorgehen vereint dabei die in den vorherigen Kapiteln erzielten Ergebnisse und erweitert die Bedrohungsanalyse-Methodik durch ein Vorgehen zur computergestützten Risikobewertung sowie einer Evaluierung dieser anhand eines Prototypen. Kapitel 8 schließt mit einer Zusammenfassung der Arbeit ab und gibt einen Ausblick für zukünftige Forschungsarbeiten.

1.5 Publikationen

Im Rahmen dieser Arbeit sind mehrere wissenschaftliche Papiere und ein Datensatz entstanden. Im Folgenden sollen diese Beiträge der Arbeit zu geordnet und der jeweilige Anteil des Autors beschrieben werden.

Für die Ausarbeitung der vorliegenden Thesis flossen die Veröffentlichungen [1, 2, 3, 4, 5, 7, 9, 10, 13] direkt ein. So bestand der Beitrag des Autors bei den Papieren [9] und [10] bei einer Sammlung und Klassifizierung von Cyber-Angriffen auf Fahrzeuge sowie einer Konzeption einer Schwachstellendatenbank für Fahrzeuge. Bei den Arbeiten [1, 13] war der Autor im gleichen Maßen wie der Erstautor beteiligt, was in den Papieren gekennzeichnet ist. Dies gilt ebenso für den Datensatz [5], der einer umfassenden Sammlung von Angriffen und Schwachstellen in Fahrzeugen entspricht. Darüber hinaus entsprechen die Veröffentlichungen [6, 8, 12] weiteren Beiträgen, die bei anknüpfenden Fragestellungen entstanden sind. Für das Papier [6] unterstützte der Autor bei der Konzeption der gezeigten Methodik sowie bei den Abschnitten zum Penetrationtesting. Bei der Veröffentlichung [12] handelt es sich ebenfalls um eine geteilte Erstautorenschaft, die im Papier gekennzeichnet ist.

[1] Kristian Beckers, Jürgen Dürrwang und Dominik Holling. „Standard Compliant Hazard and Threat Analysis for the Automotive Domain“. In: *Information* 7.3 (2016), Seiten 1–36. ISSN: 2078-2489. DOI: 10.3390/info7030036 (siehe Seiten 29, 33, 86, 92, 100, 115).

[2] Jürgen Dürrwang, Kristian Beckers und Reiner Kriesten. „A Lightweight Threat Analysis Approach Intertwining Safety and Security for the Automotive Domain“. In: *International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*. Springer. 2017, Seiten 305–319. DOI: 10.1007/978-3-319-66266-420 (siehe Seiten 26, 28, 33, 72, 92, 94, 100, 107, 179).

[3] Jürgen Dürrwang, Johannes Braun, Marcel Rumez, Reiner Kriesten und Alexander Pretschner. „Enhancement of Automotive Penetration Testing with Threat Analyses Results“. In: *SAE International Journal of Transportation Cybersecurity and Privacy* 1.2 (2018). ISSN: 2572-1054. DOI: 10.4271/11-01-02-0005 (siehe Seiten 23, 26, 31, 33, 72, 94, 125, 136, 167, 169, 179, 180).

[4] Jürgen Dürrwang, Florian Sommer und Reiner Kriesten. „Automation in automotive security by using attacker privileges“. In: *19th escar Europe : The World's Leading Automotive Cyber Security Conference (Konferenzveröffentlichung)*. 2021. DOI: 10.13154/294-8357 (siehe Seiten 33, 143, 148, 169).

[5] Sommer Florian und Dürrwang Jürgen. *IEEM-HsKA/AAD: Automotive Attack Database (AAD)*. Herausgegeben von Institut für Energieeffiziente Mobilität. https://github.com/IEEM-HsKA/AAD. 2019 (siehe Seiten 22, 23, 27, 29, 30, 33, 73, 77, 94, 140, 142, 148, 167, 173, 181).

[6] Sommer Florian, Dürrwang Jürgen, Wolf Marius, Juraschek Hendrik, Ranert Richard und Kriesten Reiner. „Automotive Network Protocol Detection for Supporting Penetration Testing“. In: *SECURWARE 2019*, Seiten 114–119. ISBN: 978-1-61208-746-7 (siehe Seite 33).

[7] Dürrwang Jürgen, Johannes Braun, Rumez Marcel und Kriesten Reiner. „Security Evaluation of an Airbag-ECU by Reusing Threat Modeling Artefacts“. In: *Proceedings of the 2017 International Conference on Computational Science and Computational Intelligence*. Herausgegeben von IEEE Computer Society. 2017, Seiten 37–43. ISBN: 978-1-5386-2652-8. DOI: 10.1109/CSCI.2017.7 (siehe Seiten 22, 23, 26, 31, 33, 86, 137).

[8] Dürrwang Jürgen, Rumez Marcel und Johannes, Braun and Reiner, Kriesten. „Security Hardening with Plausibility Checks for Automotive ECUs“. In: *VEHICULAR 2017*. Band 6, Seiten 38–41 (siehe Seiten 22, 33, 181).

[9] Ring Martin, Dürrwang Jürgen, Sommer Florian und Kriesten Reiner. „Building an Automotive Vulnerability Database: Survey and Tools“. In: *ESCAR* 2015 (2015) (siehe Seite 33).

[10] Ring Martin, Dürrwang Jürgen, Sommer Florian und Kriesten Reiner. „Survey on Vehicular Attacks -- Building a Vulnerability Database“. In: *2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES2015)*. 2015, Seiten 208–212. DOI: 10.1109/ICVES.2015.7396919 (siehe Seiten 33, 94).

[11] Marcel Rumez, Jürgen Dürrwang, Johannes Braun und Reiner Kriesten. „Security Hardening of Automotive Networks Through the Implementation of Attribute-Based Plausibility Checks“. In: *International Journal on Advances in Security* 11 (2018). ISSN: 1942-2636 (siehe Seite 181).

[12] Marcel Rumez, Jürgen Dürrwang, Tim Brecht, Timo Steinshorn, Peter Neugebauer, Reiner Kriesten und Eric Sax. „CAN Radar: Sensing Physical Devices in CAN Networks based on Time Domain Reflectometry“. In: *2019 IEEE Vehicular Networking Conference (VNC)*. IEEE, 12019, Seiten 1–8. ISBN: 978-1-7281-4571-6. DOI: 10.1109/VNC48660.2019.9062819 (siehe Seiten 33, 181).

[13] Florian Sommer, Jürgen Dürrwang und Reiner Kriesten. „Survey and Classification of Automotive Security Attacks“. In: *MDPI Information* 10.4 (2019), Seite 148. ISSN: 2078-2489. DOI: 10.3390/info10040148 (siehe Seiten 27, 29, 33, 73, 140–143, 163, 170, 171, 174, 175, 180, 291).

2

Grundlagen

In der vorliegenden Arbeit wird eine Bedrohungsanalysemethodik entwickelt, welche die Betriebssicherheit von Fahrzeugen steigert. Hierzu werden Konzepte und Begrifflichkeiten aus den Bereichen der Betriebs- und Angriffssicherheit verwendet. Darüber hinaus zeigen die präsentierten Beispiele automobil-spezifische Technologien und Prozesse. Aufgrund dessen sind in den folgenden Abschnitten die notwendigen Grundlagen für ein fortlaufendes Verständnis der Arbeit aufgeführt. Zu Beginn werden die unterschiedlichen Vernetzungstopologien und Bussysteme erläutert, die sich in den präsentierten Beispielen finden. Anschließend wird ein kurzer Einblick in Entwicklungsphasen der Fahrzeugentwicklung gegeben, um die hier entwickelte Methodik einordnen zu können. Zu den Entwicklungsphasen gehören ebenso Prozesse, welche die Betriebssicherheit eines Fahrzeuges gewährleisten. Hier wird insbesondere auf die « funktionale Sicherheit » eingegangen. Zur Erreichung dieser werden Analysemethoden eingesetzt, die in die Konzeption der neuen Bedrohungsanalysemethodik einfließen. Im Anschluss wechselt der Fokus von der Betriebssicherheit zur Angriffssicherheit im Fahrzeug. Beginnend mit einer allgemeinen Einführung und den relevanten Begrifflichkeiten werden fahrzeug-spezifische Aspekte und Methoden präsentiert.

2.1 Cyber-physisches System (CPS) in Kraftfahrzeugen

Im Kontext von Kraftfahrzeugen können CPS als ein Verbund aus Controllereinheiten (a), Aktoren (b) und Sensoren (d) verstanden werden (siehe Abbildung 2.1). Mittels Sensoren erfassen die Systeme die physische Umwelt und die Interaktion des Fahrzeugs mit dieser. Durch die vom Controller angesteuerten Aktoren (b) kann das Verhalten (c) des Fahrzeugs in der Umwelt beeinflusst werden und bei Fehlverhalten eine Gefährdung entstehen.

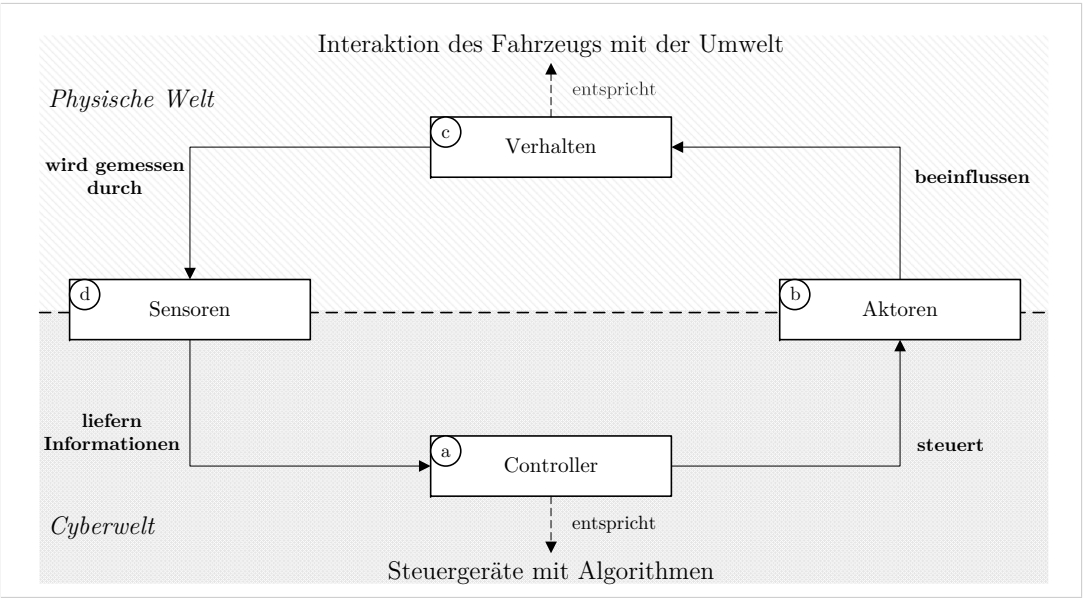


Abbildung 2.1: Cyber-physischen Systeme (CPS) im Kontext von Kraftfahrzeugen, basierend auf den Arbeiten von Parnas und Madey [151] sowie Thompson et al. [203]. Das CPS-Modell zeigt einen Controller (a), Aktuatoren (b), Sensoren (d) und das Verhalten (c) des Fahrzeugs in der Umwelt. Das Fahrzeugverhalten wird durch Aktoren beeinflusst, die wiederum durch Algorithmen auf einem Controller gesteuert werden.

CPS bestehen aus Hard- und Softwarekomponenten, wobei ein Teil der Softwarekomponenten von Algorithmen repräsentiert ist, die in Steuergeräten implementiert werden und durch Steuersignale die angebundenen Aktuatoren ansteuern. Hierzu nehmen die Algorithmen Eingangsinformationen von Sensoren, anderen Systemen aus dem Systemverbund oder von Schnittstellen zur Außenwelt auf. Insbesondere diejenigen, die Teil des internen Fahrzeugnetzes sind und mit der Umwelt kommunizieren, öffnen das Fahrzeug nach außen. Zu diesen zählen beispielsweise On-board Diagnostics (OBD), Universal Serial Bus (USB) oder Bluetooth. Es entsteht die Situation, dass die bis vor Jahren noch isolierten internen Fahrzeugnetze mit der Außenwelt kommunizieren können. Die ausgetauschten Informationen können neben dem gewünschten Empfänger, grundsätzlich auch von Drittparteien, unerlaubt gelesen oder verändert werden. Folglich ist es möglich, dass böswillig manipulierte oder nicht erwünschte Informationen von außen in das fahrzeug-interne Netzwerk gelangen können. Bezogen auf die in Abbildung 2.1 gezeigten Elemente und die Zuordnung möglicher Manipulationen durch eine böswillige Instanz (Angreifer) ergibt sich die in Abbildung 2.2 dargestellte Situation.

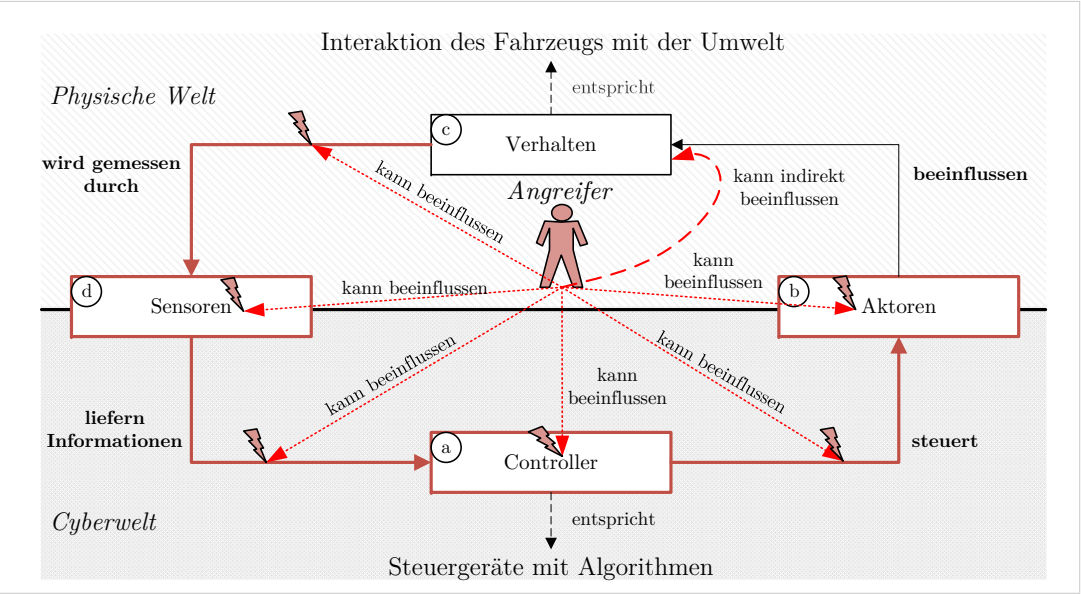


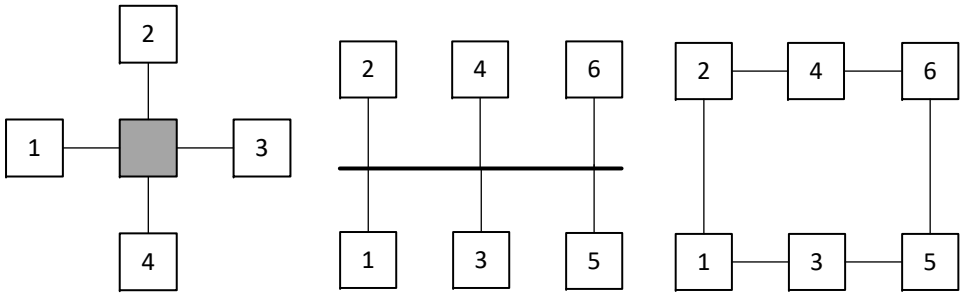
Abbildung 2.2: CPS Modell aus Abbildung 2.1, das durch einen Angreifer erweitert ist und dessen Beeinflussungsmöglichkeiten (direkt und indirekt) aufzeigt. Bestehend aus dem Controller (a), Aktuatoren (b), Sensoren (d) und dem Verhalten (c) des Fahrzeugs in der Umwelt. Eine Beeinflussung des Verhaltens kann erreicht werden, indem der Angreifer entweder die Funktionen der Teilsysteme (a, b, d) oder den Informationsfluss zwischen Letzteren manipuliert.

Durch die in Abbildung 2.2 hervorgehobenen Angriffspunkte kann ein Angreifer die vom Sensor (d) erfassten oder die zur Controllereinheit (a) übertragenen Informationen manipulieren. So kann ein Angreifer beispielsweise ausgestrahlte Radarwellen überlagern [215], um dem Radarsenor einen falschen Abstand zum vorausfahrenden Fahrzeug vorzuspielen. Ebenso sind kameragestütze Assistenzsysteme anfällig für Manipulationen in der Umwelt. Ein Beispiel hierfür sind manipulierte Verkehrsschilder [190]. Kombiniert mit der Tatsache, dass die im Controller implementierten Algorithmen die manipulierten Eingangsinformationen verarbeiten, kann ein Fehlverhalten der angesteuerten Aktuatoren erzielt werden. Es entsteht die Problematik, dass eine böswillige Manipulation von Informationen oder Softwarekomponenten (Cyber-Angriff) Verhaltensabweichungen erzeugen und Gefahrensituationen auslösen können. Diese Wirkkette zeigt damit die eindeutige Beeinflussung der Betriebssicherheit (engl. Safety) durch ein Verletzen der Angriffssicherheit (engl. Security) [36].

2.2 Vernetzungstechnologien im Fahrzeug

In der vorliegenden Arbeit werden in Kapitel 4 bis 7 « laufende Beispiele » präsentiert, die unterschiedliche Technologien aufweisen. Im Folgenden werden daher die verwendeten Begrifflichkeiten, Vernetzungs- und Verbindungstechnologien sowie der Ablauf der Diagnose im Fahrzeug erläutert. Letzterer dient zur Feststellung von Fehlern in Steuergeräten und nutzt Verbindungstechnologien, um erkannte Fehler an das Werkstattequipment (Diagnosetester) übertragen zu können.

Eine Topologie beschreibt, auf welche Weise Bus-Teilnehmer miteinander verbunden sind und wie sich daraus ihre Kommunikation ableitet. Grundsätzlich lassen sich die unterschiedlichen Topologien in drei Klassen einteilen, die Stern-, Bus- und Ring-Topologie [162, Seite 7]. Diese sind mit Abbildung 2.3 exemplarisch dargestellt.



(a) Stern-Topologie (b) Bus-Topologie (c) Ring-Topologie

Abbildung 2.3: Beispielhafte Darstellung der a) Stern-, b) Bus- und c) Ring-Topologie. Die Stern-Topologie weist außerdem einen Sternkoppler in der Mitte des Netzwerks auf.

Die **Stern-Topologie** zeichnet sich dadurch aus, dass eine zentrale Einheit den Kommunikationsverkehr zwischen den einzelnen Teilnehmern verteilt (Abbildung 2.3a). Diese Einheit wird als Sternkoppler bezeichnet und jeder Teilnehmer ist durch eine Punkt-zu-Punkt Verbindung mit diesem Element verbunden. Der Vorteil, der sich dadurch ergibt, ist, dass sich die Teilnehmer ihre Bandbreite nicht mit weiteren Teilnehmern teilen müssen, was hohe Übertragungsraten ermöglicht. Nachteilig ist allerdings der hohe Verdrahtungsaufwand, da von jedem Teilnehmer eine Leitung zum zentralen Verteiler benötigt wird. Außerdem muss Letzterer in der Lage sein, die eingehenden Datenpakete intelligent zu verteilen. Darüber hinaus fällt der gesamte Verbund aus, wenn die zentrale Einheit nicht mehr verfügbar ist. Ein Beispiel hierfür ist ein Ethernet-Netzwerk mit einem zentralen Switch.

Bei der **Bus-Topologie** sind alle Teilnehmer mit einem gemeinsamen Bus verbunden. Hierdurch stehen allen Teilnehmern zu jeder Zeit alle Daten auf dem Bus zur Verfügung. Dies kann von Vorteil sein, wenn ein Datenpaket von einem Teilnehmer zu mehreren Empfängern gesendet werden soll, da das Paket nur einmal versendet werden muss. Dem gegenüber steht die Aufteilung der Bandbreite durch die Anzahl der Teilnehmer am Bus, da stets nur ein Teilnehmer ein Datenpaket versenden kann. Letzteres muss durch Mechanismen gewährleistet werden, um Kollisionen auf dem Bus zu verhindern. Ein Vorteil dieses Konzeptes ist allerdings, dass der Ausfall eines Teilnehmers nicht zum Ausfall des gesamten Busses führt.

Bei der **Ring-Topologie** hingegen führt der Ausfall eines Teilnehmers zum Ausfall des gesamten Verbundes, da die Teilnehmer einen Ring formen, der damit unterbrochen

ist. Vorteilhaft ist hierbei der geringere Verkabelungsaufwand, da ausschließlich eine Verbindung von Teilnehmer zu Teilnehmer gezogen werden muss.

2.2.1 Controller Area Network (CAN)

Einer der bekanntesten Vertreter für die Bus-Topologie im Fahrzeug ist der CAN-Bus [73], der 1986 von der Firma Bosch entwickelt wurde [162, Seiten 14–20]. Dieser wurde mit der Norm ISO 11898 [93] spezifiziert und ist – wie die nachfolgenden Protokolle – ebenso auf die Schichten des ISO/OSI-Modell [96] bezogen. So sind die physikalischen Eigenschaften sowie der Betrieb als « Low-Speed-CAN » mit bis zu 125 kBit/s und als « High-Speed-CAN » mit bis zu 1 MBit/s spezifiziert. Das eingesetzte Protokoll ist nachrichtenorientiert und erfordert einen eindeutigen Identifier für die jeweilige Botschaft. Jeder Bus-Teilnehmer prüft anhand diesem, ob die Nachricht für ihn relevant ist oder nicht. So kann jeder Netzknoten entscheiden, ob eine Botschaft vom Bus übernommen werden soll oder nicht. Damit sind die Knoten am Bus gleichberechtigt, was einem Multi-Master-System entspricht. Mittels einer Prüfsumme und Fehlererkennung wird die Integrität der Botschaftsdaten gegen zufällige Übertragungsfehler abgesichert. Eine Absicherung hinsichtlich böswilliger Manipulationen (Cyber-Angriff) zeigt der CAN allerdings nicht.

2.2.2 Local Interconnect Network (LIN)

Das Single-Master-System LIN [94] wurde im Jahre 1998 durch einen Zusammenschluss mehrerer Hersteller entwickelt. Diese zielt auf eine preisgünstige Vernetzung ab und stellt eine Bandbreite von 20 kBit/s zur Verfügung [162, Seiten 20–23]. Als Übertragungsmedium kann eine Eindrahtleitung nach ISO 9141 verwendet werden. Die aktuelle Spezifikation beschreibt neben der physikalischen und Sicherungsschicht (Schicht 1 und 2 des OSI-Modells [96]) ebenso Festlegungen für die Diagnose. Auch LIN setzt auf eine Bus-Topologie, verwendet allerdings einen Master- und mehrere Slave-Knoten. Ähnlich zum CAN weist auch LIN zur Sicherung der Datenintegrität der Nachrichten eine Prüfsumme vor. Letztere sichert allerdings alleinig gegen Übertragungsfehler ab, eine Absicherung hinsichtlich Cyber-Angriffen fehlt.

2.3 Betriebssicherheit (Safety)

Da sich die hier vorliegende Arbeit mit der Steigerung der Betriebssicherheit von Kraftfahrzeugen auseinandersetzt, soll dieser Abschnitt die grundlegenden Aspekte zu diesem Themengebiet aufzeigen. Im Einleitungskapitel wurde dazu bereits erläutert, das Fehlfunktionen in Kraftfahrzeugen zu Personen- oder Sachschäden führen können. Damit steht die Gewährleistung der Betriebssicherheit im Vordergrund der Fahrzeugentwicklung. Für die generelle Entwicklung von sicherheitskritischen Systemen, existiert die internationale Norm IEC 61508 [57] die einen ganzheitlichen Ansatz verfolgt und sieben Teile aufweist [162, Seite 254]. Allerdings enthält die IEC 61508

Vorgaben, die eine Anwendung in der Fahrzeugentwicklung erschwert. Infolge dessen wurde für die Automobilindustrie die Norm ISO 26262 [99] entwickelt, die sich an der IEC 61508 orientiert. Ein Unterschied zwischen der IEC 61508 und der ISO 26262 ist beispielsweise, das die IEC 61508 das Sicherheitsintegritätsniveau (engl. safety integrity level) auf Safety-Funktionen bezieht, im Fahrzeug hingegen, den Systemen das Sicherheitsintegritätsniveau zugeordnet wird. Der wohl größte Unterschied zwischen beiden Normen ist allerdings, dass bei der IEC 61508 angenommen wird, dass ein System vollständig von einer Organisation entworfen und implementiert wird. Die notwendige Betrachtung der Lieferkette mit zahlreichen Zulieferern fehlt hier.

2.3.1 Funktionale Sicherheit nach ISO 26262

Die ISO 26262 [99] übernimmt das Konzept der Safety Integrity Level (SIL) aus der IEC 61508 [57], passt diese jedoch mit dem Automotive Safety Integrity Level (ASIL) an die Automobilwelt an. Ein elementarer Unterschied ist hierbei die Kategorie « Kontrollierbarkeit für den Fahrer » (engl. controllability), welcher beschreibt, inwieweit ein Fahrer mit einer auftretenden Gefahrensituation umgehen kann. Die Norm definiert die Betriebssicherheit als funktionale Sicherheit, was als die Abwesenheit von nicht zu akzeptierenden Risiken in elektrischen und elektronischen (E/E) Systemen verstanden wird. Hierzu stellt die Norm einen Sicherheitszyklus bereit, der sich auf Management, Entwicklung, Produktion, Betrieb, Service und Außerbetriebnahme bezieht.

Aufgrund einer Betrachtung aus der Safety-Sicht ist es ebenso relevant, das Verhalten des Analysegegenstandes und seine Betriebsszenarien zu erfassen. Anknüpfend an diesen Schritt wird der Safety-Lebenszyklus initiiert. Dieser bezieht sich stets auf genau ein *Item*, sodass bei mehreren *Items* ebenso der Lebenszyklus mehrmals durchlaufen werden muss. An dieser Stelle muss jedoch unterschieden werden, ob es sich bei dem *Item* um eine Neuentwicklung handelt oder nicht. Bei Ersterem muss der gesamte Lebenszyklus durchlaufen werden. Im zweiten Falle muss hingegen geprüft werden, inwieweit Modifikationen den Durchlauf aller Schritte des Lebenszyklus erfordern. Hierbei müssen nur jene Schritte der ISO 26262 durchlaufen werden, die durch die Modifikation beeinflusst werden. Um dies zu entscheiden, kann beispielsweise eine Änderungsauswirkungsanalyse (engl. Change Impact Analysis) [142] durchgeführt werden. Diese folgt einem systematischen Ansatz für die Bewertung von Veränderungen an einem System (*Item*). Modifikationen können hierbei die Hard- und Software betreffen und sind aufgrund von Optimierungen, Erweiterung des Funktionsumfangs oder geänderten Anforderungen notwendig. Ist die Initiierung des Sicherheitslebenszyklus abgeschlossen, folgt die Gefährdungs- und Risikoanalyse. Die Aufgabe dieser Phase ist es zu bestimmen, welche Gefährdungen durch das *Item* für die Umwelt und Insassen entstehen können und wie diese qualitativ zu bewerten sind. Hierzu fließen unterschiedliche Informationen in die Analyse ein. Neben der Definition des Anwendungsbereichs (*Item*) sind das bestehende Analyseergebnisse in Form bekannter

Gefährdungen, die Betriebs- und Fahrbedingungen sowie Kataloge mit potenziellen Fehlern, um die Analyse zu unterstützen. Basierend auf diesen Informationen werden jene Ausfälle aufgedeckt, die Auswirkungen auf der Fahrzeugebene zeigen und damit die Betriebssicherheit gefährden können. Hierzu stehen unterschiedliche Techniken für die Durchführung der Gefährdungsanalyse zur Verfügung, die in Abschnitt 2.4 näher erläutert werden. Sind die Gefährdungen identifiziert, werden diese anhand von drei Klassen qualitativ bewertet. Diese Klassen sind [99, Seiten 9–10]:

- Der Schweregrad des Schadens (engl. Severity S)
  - Keine Verletzungen (S0)
  - Leichte und mittlere Verletzungen (S1)
  - Schwere und lebensbedrohliche Verletzungen, Überleben wahrscheinlich (S2)
  - Lebensbedrohliche Verletzungen, Überleben unwahrscheinlich (S3)
- Häufigkeit der Fahrsituation (engl. Probability of Exposure E)
  - Nicht vorstellbar (E0)
  - Extrem niedrige Wahrscheinlichkeit (E1)
  - Geringe Wahrscheinlichkeit (E2)
  - Mittlere Wahrscheinlichkeit (E3)
  - Hohe Wahrscheinlichkeit (E4)
- Kontrollierbarkeit der Situation durch den Fahrer (engl. Controllability C)
  - Im Allgemeinen kontrollierbar (C0)
  - Kontrollierbar (C1)
  - Normalerweise kontrollierbar (C2)
  - Unkontrollierbar (C3)

Die Klassifizierung bezieht sich somit auf die Auswirkungen einer bestimmten Betriebssituation des Fahrzeugs. Ein Beispiel hierfür wäre eine ungewollte Airbag-Detonation, wenn der Fahrer mit hoher Geschwindigkeit in die Kurve fährt. Tabelle 2.1 stellt diese Kategorien gegenüber und weist einen ASIL-Wert zu.

Hierbei entspricht ASIL-D dem höchsten und QM dem geringsten Risikowert. Für Letzteren müssen keine Maßnahmen ergriffen werden, sodass ausschließlich für Level A bis D Maßnahmen implementiert werden. Dies schließt allerdings nicht aus, dass ein Hersteller ebenfalls für das QM-Level Maßnahmen implementieren kann. Die Level werden auch als Safety-Ziele bezeichnet und sollen die Risiken von Gefährdungen

Tabelle 2.1: Matrix zur ASIL Bestimmung [99, Seiten 9–10][75, Seite 30].

Schweregrad S	**Häufigkeit der Fahrsituation** E	**Kontrollierbarkeit** C		
		Einfach kontrollierbar (C1)	Normalerweise kontrollierbar (C2)	Unkontrollierbar (C3)
Leichte und mittlere Verletzungen (S1)	Extrem niedrig (E1)	QM	QM	QM
	Gering (E2)	QM	QM	QM
	Mittel (E3)	QM	QM	A
	Hoch (E4)	QM	A	B
Schwer und lebensbedrohlich, Überleben wahrscheinlich (S2)	Extrem niedrig (E1)	QM	QM	QM
	Gering (E2)	QM	QM	A
	Mittel (E3)	QM	A	B
	Hoch (E4)	A	B	C
Lebensbedrohlich, Überleben unwahrscheinlich (S3)	Extrem niedrig (E1)	QM	QM	A
	Gering (E2)	QM	A	B
	Mittel (E3)	A	B	C
	Hoch (E4)	B	C	D

abmildern. Für ASIL-D muss hierbei die effektivste Maßnahme definiert werden, wobei auch mehrere weniger effektive Maßnahmen kombiniert werden können, um das gleiche Risikolevel abmildern zu können [86]. An dieser Stelle sei angemerkt, dass in dieser Phase noch keine technischen Details oder Realisierungen der Maßnahmen festgelegt werden.

2.4 Gefährdungs- und Risikoanalyse Methoden für die Safety-Domäne

Im vorherigen Abschnitt wurden der Safety-Lebenszyklus der ISO 26262 [99] aufgezeigt und die darin geforderte Gefährdungsanalyse erläutert. Zur Durchführung dieser Analyseart können unterschiedliche Methoden eingesetzt werden, die sich in zwei Klassen einteilen lassen. Induktive und deduktive Methoden, die sich in der Art des Vorgehens unterscheiden (Abbildung 2.4).

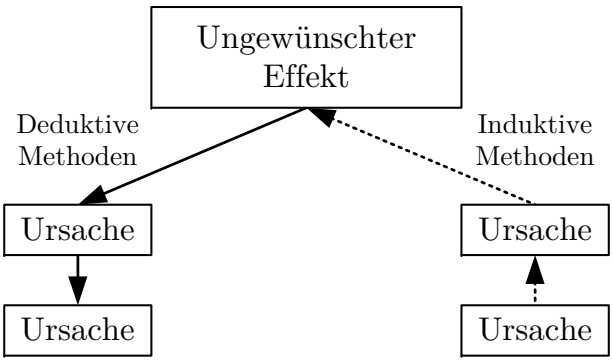


Abbildung 2.4: Konzeptionelle Unterscheidung zwischen induktiven und deduktiven Methoden zur Gefährdungsidentifikation. Hierbei folgen deduktive Methoden einem Top-Down- und induktive einem Bottom-Up-Ansatz [62, Seite 50].

Bei einem induktiven Vorgehen werden Schlussfolgerungen vorgeschlagen, die mehr Informationen enthalten als die Beobachtung oder die Erfahrung, auf der sie basieren [62, Seiten 48–50]. Beispielsweise war jedes jemals gesehene Allgäuer Braunvieh braun, und daher sind alle Kühe dieser Gattung braun. Der Wahrheitsgehalt dieser Schlussfolgerung ist nur im Hinblick auf zukünftige Erfahrungen überprüfbar und Gewissheit ist nur erreichbar, wenn alle möglichen Fälle untersucht wurden [62, Seite 49]. Diese Form wird als « Was-wäre-wenn » Analyse gesehen. Das bedeutet, dass ein Analyst beispielsweise eine Gefährdung aus begrenzten Informationen des *Items* herleitet und einen Informationszugewinn erzeugt. Aufgrund dessen eignet sich diese Methodenart für eine frühe und allgemeine Gefährdungsidentifikation. Hierbei kann ein Analyst allerdings nicht sicher sein, dass seine gezogenen Schlussfolgerungen schlüssig sind, da diese auf Erfahrung und dem begrenzten Kenntnisstand beruhen.

Deduktive Methoden hingegen leiten Schlussfolgerungen ab, die aus einer Reihe von Annahmen getroffen werden und die nicht mehr Informationen enthalten als die getroffenen Annahmen [62, Seiten 48–50]. Zum Beispiel sind alle Amseln Vögel. Dies ist eine Amsel, somit ist sie ein Vogel. Die Wahrheit der Schlussfolgerung hängt von den Prämissen ab und die Schlussfolgerung kann nicht falsch sein, wenn die Prämissen, auf denen sie beruhen, wahr sind [62, Seite 48]. Somit werden diese als « Wie-kann » Analysen bezeichnet. Im Zuge dessen leitet diese Methodenart nur Schlussfolgerungen ab, die aus den vorliegenden Informationen gezogen werden können. Allerdings werden spezifische Kausalfaktoren gesucht, welche eine Schlussfolgerung stützen. Ein Beispiel hierfür ist die Analyse der konkreten Ursachen für den Ausfall einer Bremse.

Auf die Safety bezogen werden deduktive Methoden typischerweise mit induktiven Analysen gekoppelt, um in erster Linie Gefährdungen zu identifizieren und anschließend deren Ursachen (Kausalfaktoren) aufzudecken. Tabelle 2.2 zeigt einen Auszug etablierter Methoden zur Gefährdungsidentifikation, die im Laufe dieser Arbeit betrachtet werden.

Tabelle 2.2: Auszug etablierter Methoden zur Gefährdungsidentifikation, die in induktive und deduktive Klassen eingeteilt sind. Eine umfassende Auflistung ist in [62, Seite 34] sowie [79, 125] gegeben.

Induktive Methoden	Deduktive Methoden
Preliminary Hazard Analysis (PHA)	Preliminary Hazard Analysis (PHA)
Hazard and Operability Study (HAZOP)	Fault Tree Analysis (FTA)
Failure Mode and Effect Analysis (FMEA)	Event Tree Analysis (ETA)
...	...

Hazard and Operability Study (HAZOP)

Eine weitere induktive Methode ist die HAZOP [62, Seite 365]. Diese Technik dient zur Identifizierung und Analyse von Gefährdungen operationeller Aspekte eines Analysegegenstandes. HAZOP zeigt dabei einen sehr strukturierten und methodischen Prozess zur Durchführung der Gefährdungsidentifikation. Die Anwendbarkeit ist breitflächig und reicht von der Konzeptphase bis hin zur Ausmusterung des betrachteten Systems. Zur Analyse werden Leitwörter, Systemdiagramme oder Designdarstellungen eingesetzt, um Systemgefährdungen zu identifizieren. Der Prozess für die Durchführung sowie die Eingangs- und Ausgangsartefakte ist in Abbildung 2.5 dargestellt.

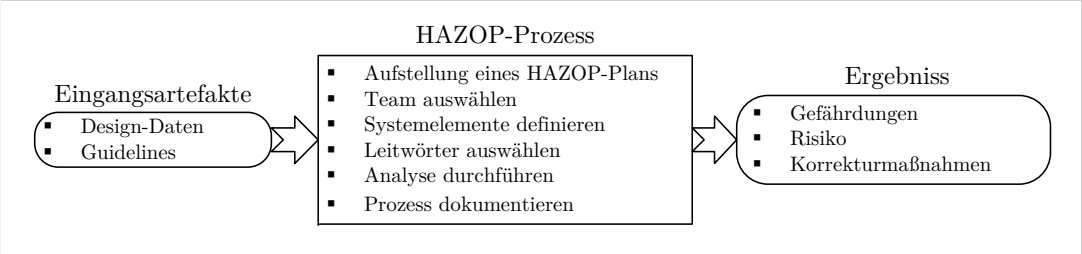


Abbildung 2.5: Überblick über den HAZOP-Prozess mit seinen Eingangs- und Ausgangsartefakten [62, Seite 368].

Damit Abweichungen identifiziert werden können, werden Leitwörter eingeführt. Diese stellen sich als Adjektive wie « mehr », « wenig », oder « zu hoch » dar und werden mit den System- beziehungsweise Prozessbedingungen kombiniert. Hiermit ist es möglich Situationen aufzudecken, die vom geplanten Systemverhalten abweichen. Beispielsweise könnte mit « *wenig* Druck im Bremssystem » die Gefährdung einer ausfallenden Fahrzeugbremse aufgedeckt werden. Durchgeführt wird die HAZOP mit einem Experten-Team, das eine Brainstorming-Sitzung durchführt und von einem Leiter geführt wird. Entscheidende Aspekte bei der HAZOP sind [62, Seite 365]:

- eine strukturierte Analyse, die einem systematischen Ablauf folgt,

- eine Anwendung mit einem multidisziplinären Experten-Team,
- die gezielte Verwendung von Design-Darstellungen,
- eine Verwendung von sorgfältig ausgewählten Systemeinheiten und Leitwörtern für die Gefährdungsidentifikation.

HAZOP gehört wie PHA zu den Arten von Analysen, die sich für vorläufige Systemdesigns eignen. Allerdings analysiert HAZOP das System detaillierter als die PHA. Aus diesem Grund ist die Methode für alle Arten von Systemen mit dessen Teilsystemen sowie Komponenten, die aus Soft- und Hardware bestehen können, geeignet. Das bedeutet, dass die HAZOP-Analyse auf unterschiedlichen Abstraktionsebenen angewendet werden kann. Durch das systematische Vorgehen ermöglicht sie eine umfassende und gründliche Identifizierung von Gefährdungen [62, Seite 366]. Deshalb wurde die Methode bereits erfolgreich in unterschiedlichen Domänen angewendet und wird in Normen für die Analyse von Gefährdungen empfohlen [99, Seite 14].

2.5 Angriffssicherheit (Security)

Nachdem im vorherigen Abschnitt die Betriebssicherheit im Allgemeinen und spezifisch für die Fahrzeug-Domäne erläutert wurde, wird an dieser Stelle ein Fokus auf die Angriffssicherheit bei Fahrzeugen gelegt. Diese ist in den letzten Jahren stärker in den Fokus der Fahrzeughersteller gerückt, da Fahrzeuge einen stetigen Zuwachs an Kommunikationsschnittstellen erfuhren. Durch diese kommunizieren moderne Fahrzeuge mit der Außenwelt, was Angriffsflächen schafft und Cyber-Angriffe ermöglicht. Im Folgenden werden die hierzu relevanten Aspekte aufgezeigt.

2.5.1 Schutzziele

Ein übergeordneter und vom Fahrzeug unabhängiger Security-Aspekt sind « Schutzziele ». Sie beschreiben, welche Eigenschaften von Informationen geschützt werden müssen. Informationen können einen Vermögenswert (engl. asset) für einen Stakeholder darstellen und sind daher schützenswert [58, Seiten 7–8]. Ein Beispiel hierfür ist ein Manipulationsschutz elektronischer Überweisungen, sodass kein drittes Subjekt (Angreifer) den Empfänger der Überweisungen ändern und damit auf sich umleiten kann. Ist dies nicht gegeben, so kann dem Sender der Überweisung ein finanzieller Schaden entstehen. Es ist daher wichtig den Zugriff auf Informationen ausschließlich autorisierten Subjekten zu erlauben. Damit dieses gewährleistet werden kann, müssen differenzierte Schutzziele beschrieben werden. Ein Teil dieser sind die Vertraulichkeit (engl. confidentiality) der Daten, die Datenintegrität (engl. integrity), die Verfügbarkeit (engl. availability) der Daten und die Authentizität (engl. authenticity). Für eine detaillierte Beschreibung der jeweiligen Schutzzieles ist wird auf [58, Seiten 7–8] verwiesen.

2.5.2 Notationen für Security-Angriffe

Der Vorgang um ein Schutzziel zu verletzen wird als Security-Angriff bezeichnet. Letzterer kann allerdings komplex sein und zahlreiche Teilschritte aufweisen, die abhängig oder unabhängig voneinander durchgeführt werden müssen. Damit diese grafisch erfasst werden können, haben sich zwei Konzepte etabliert. So können entweder Angriffsgraphen oder Angriffsbäume für die Beschreibung eines Security-Angriffs verwendet werden. Beide Notationen zeigen Unterschiede in ihrer Darstellung und Semantik, die in nachfolgenden Unterabschnitten beschrieben werden.

Angriffsgraphen

Ein Angriffsgraph ist eine Darstellung aller Angriffspfade, die in einem System denkbar sind. Dieser entspricht einem gerichteten Graphen $G = (V, E)$ mit den Knoten V und den Kanten E und dient der Modellierung mehrstufiger Cyber-Angriffe [207]. Ein Vorteil dieser Notation ist die anwenderfreundliche Handhabung, die durch eine visuelle und intuitive Darstellung entsteht. Darüber hinaus weist diese Notation formale Semantiken auf, die mit Algorithmen für Analysen eingesetzt werden können. Die Knoten $v \in V$ des Graphen repräsentieren den Zustand eines Systems während des Angriffs [114, Seite 33]. Dies kann beispielsweise das Ausweiten von Benutzerrechten im System sein. Die Kanten $e \in E$ beschreiben hingegen Zustandsübergänge, die durch Aktionen des Angreifers ausgelöst werden. Dies kann als das Ausnutzen von Schwachstellen und damit als ein Angriffsschritt angesehen werden. Eine beispielhafte Darstellung eines Angriffsgraphen ist in Abbildung 2.6 geben.

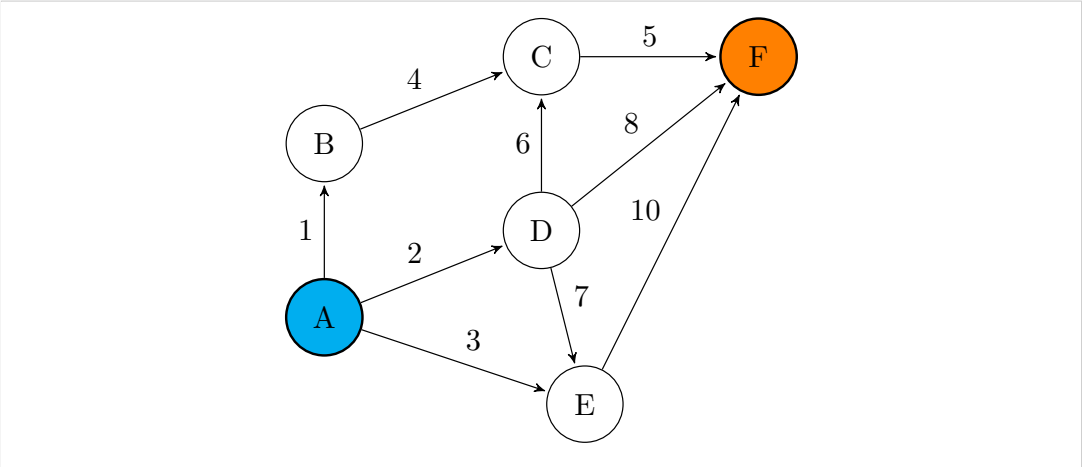


Abbildung 2.6: Gerichteter azyklischer Angriffsgraph, der die Knoten A bis E und die Kanten 1 bis 10 besitzt.

Hierbei wird Knoten A – mit keiner eingehenden Kante – als der Startpunkt des Angreifers verstanden und der Knoten F als das Ziel des Angreifers. Ein Unterschied, der zu Angriffsbäumen besteht, ist die nicht eindeutig definierte Erfüllungsbeziehung. So obliegt es dem Ersteller, ob die Beziehung eingehender Kanten ausschließlich disjunktiv

oder konjunktiv verstanden wird. Für den Knoten C bedeutet dies, dass entweder alle eingehenden Kanten (UND-Verknüpfung) erfüllt sein müssen oder nur eine (ODER-Verknüpfung) dieser. So sehen beispielsweise Wang et al.[207] die Erfüllungsbeziehung als grundsätzlich konjunktiv an, Jha et al.[101] modellieren hingegen disjunktiv.

Angriffsbäume

Angriffsbäume wurden erstmalig von Salter et al. [174] im Jahre 1998 erwähnt und anschließend von Schneier [181] in grafischer Notation präsentiert. Sie können als das Security-Pendant zum FTA gesehen werden und stellen sich damit als deduktive Analysemethode dar. Die Darstellungsform ist weit verbreitet und eine Analyse von Kordy et al. [114] zeigt auf, dass mehr als 30 Ansätze zur Modellierung von Cyber-Angriffen darauf basieren. So werden Angriffsbäume neben der IT-Security, bereits für Security-Analysen von Supervisory Control And Data Acquisition (SCADA)-Systemen [37, 199] und ebenso im automotiven Umfeld eingesetzt [15, 83]. Sie können für die Darstellung von mehrstufigen Angriffen aber auch in Bedrohungsanalysen eingesetzt werden.

Ein Angriffsbaum stellt sich als azyklisch gerichteter Graph dar, der einen Wurzelknoten, mehrere Zwischenknoten und Endknoten besitzt [133]. Letztere werden auch als die Blätter des Baums und untergeordnete Knoten (Teilziel 1 und 2) als Kinder bezeichnet. Der Wurzelknoten ist durch fehlende Eingangskanten definiert. Die Zwischen- und Endknoten weisen hingegen einen Eingangsgrad von genau einer Kante auf. Das bedeutet, dass ein Angriffsbaum mit n Knoten genau $n - 1$ Kanten aufweist. Der Wurzelknoten entspricht dabei dem Ziel des Angreifers, welches durch die darunter folgenden Knoten kontinuierlich verfeinert wird, bis grundlegende Aktionen (Basisaktionen) erreicht werden. Letzteres ist gegeben, wenn sich Angriffsschritte nicht weiter aufteilen oder konkretisieren lassen. Übertragen auf den gesamten Baum bedeutet dies, das mit jeder nachfolgenden Knotenebene die Angriffsschritte konkreter werden. In Abbildung 2.7 zeigt einen beispielhaften Angriffsgraphen mit drei Basisaktionen.

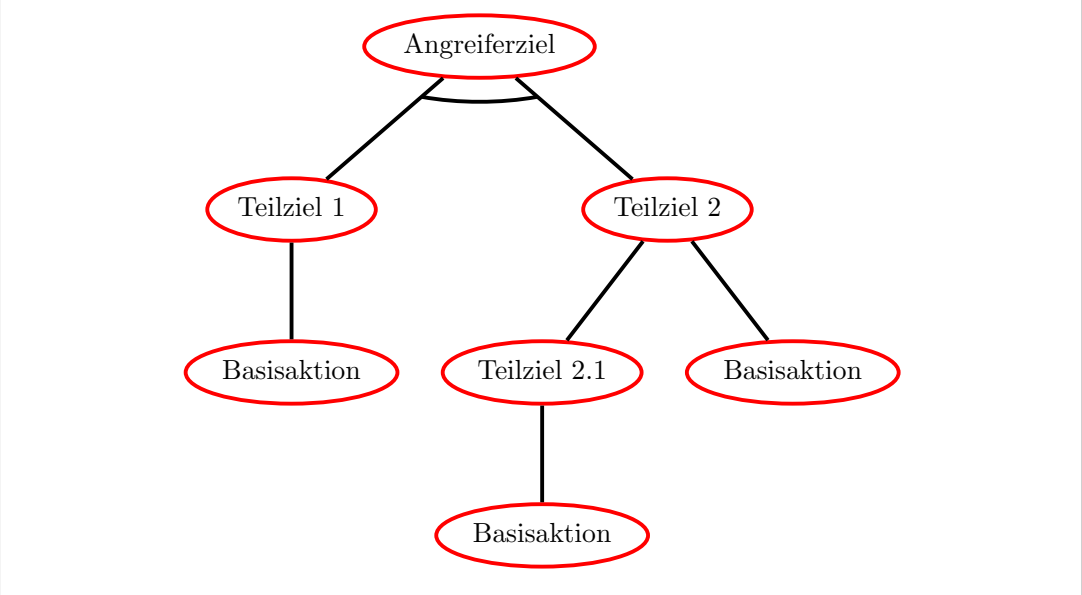


Abbildung 2.7: Beispielhafter Angriffsbaum.

Der Bogen zwischen den ausgehenden Linien am Wurzelknoten stellt ein konjunktive Beziehung der darunter liegenden Knoten dar (*UND*-Verknüpfung). Dies kennzeichnet, dass sowohl Teilziel 1 als auch 2 erfolgreich durchgeführt werden müssen, um das Angreiferziel zu erreichen. Die verbliebenen Kanten zeigen diesen Bogen nicht, was als disjunktive Beziehung (*ODER*-Verknüpfung) interpretiert wird, bei der nur einer der Zweige erfolgreich sein muss. Hierbei kann grundsätzlich angenommen werden, das Knoten mit *ODER*-Verknüpfungen, einfacher auszunutzen sind, da weniger Bedingungen erfüllt sein müssen.

Neben der Vorschrift, dass Zwischenknoten und Blätter einen Eingangsgrad von eins aufweisen müssen, besteht ein weiterer Unterschied zu einem Angriffsgraphen. So weisen die Kanten keine Bedingungen für die Erreichung des Folgeknotens auf und stellen sich damit ausschließlich als Ursache-Folge-Relation dar.

2.5.3 Bedrohungsanalyse und Risikobestimmung

Der vorherige Abschnitt zeigte Notationen für die Beschreibung von Cyber-Angriffen. Es ist jedoch wünschenswert, Cyber-Angriffe gar nicht erst zu ermöglichen. Hierzu muss während der Entwicklung eines Systems analysiert werden, ob Cyber-Bedrohungen existieren, die einen Cyber-Angriff ermöglichen könnten. Cyber-Bedrohungen stellen sich somit als die potenziell denkbaren Cyber-Angriffe dar. Sind diese Situationen bekannt, können Gegenmaßnahmen ergriffen werden, um deren Realisierung (Cyber-Angriff) abzuschwächen oder zu verhindern.

Das hierzu eingesetzte Konzept wird als « Bedrohungsanalyse » bezeichnet und dient der Identifikation und Priorisierung von Bedrohungen bezogen auf einen Analysegegenstand. Bedrohungen stellen sich dabei als Verletzungen von Sicherheitszielen dar, die in einem

Schaden für Vermögenswerte (engl. assets) resultieren können. Ein generelles Vorgehen hierzu ist in Abbildung 2.8 mit sieben Teilschritten dargestellt.

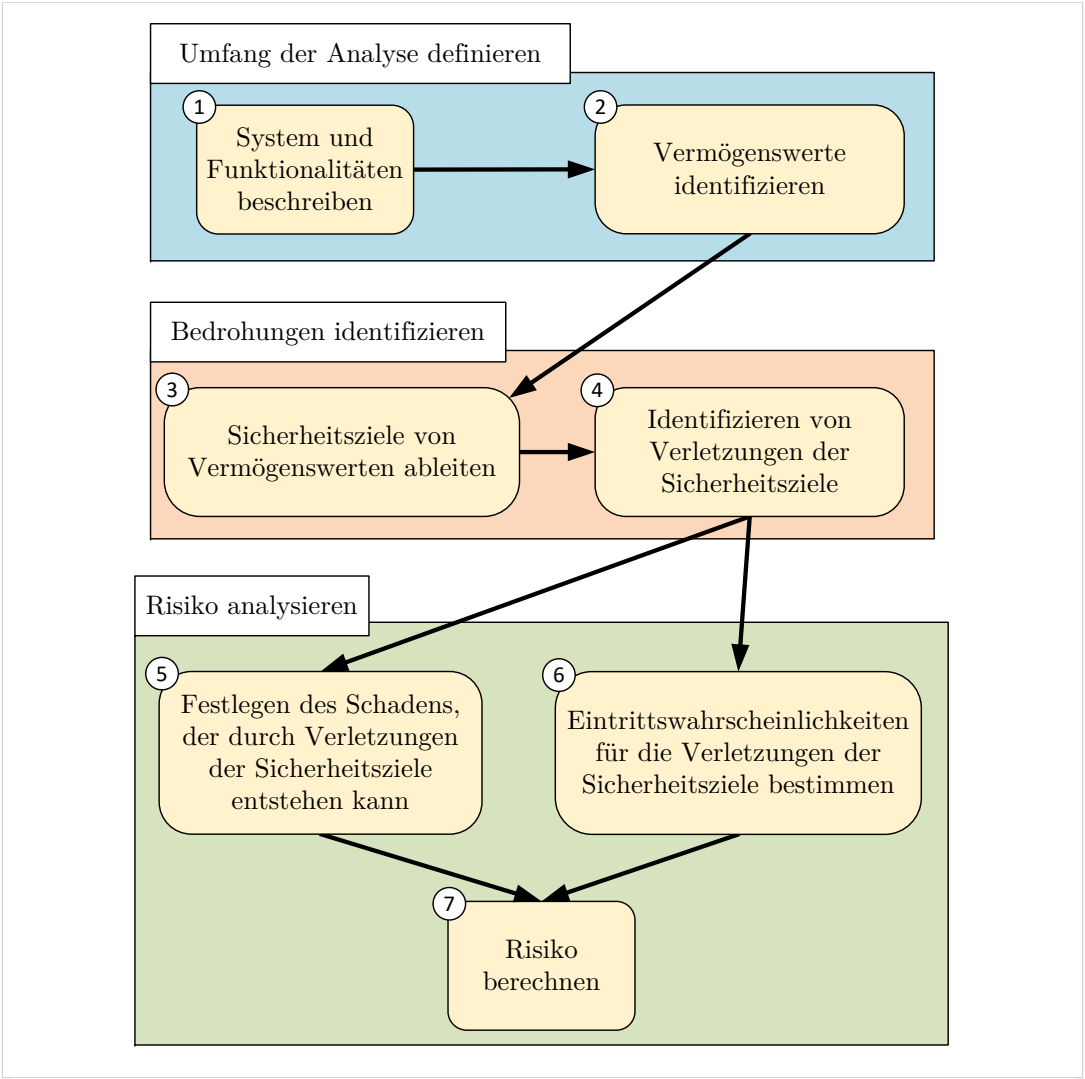


Abbildung 2.8: Generischer Ablauf einer Bedrohungsanalyse, aufgeteilt in drei Phasen: Festlegen des Analyse-Umfangs, Identifizieren der Bedrohungen und dem Analysieren des Risikos. Die Pfeile in der Abbildung repräsentieren die Abfolge der Analyse-Schritte.

In der ersten Phase wird der Analyseumfang definiert. Hierzu werden in Schritt ① die Grenzen des zu betrachteten Systems festgelegt und die Funktionalitäten identifiziert. Bei komplexen Systemen bietet es sich an, das System zu zerlegen und die gewünschten Systemfunktionalitäten auf die jeweiligen Teilsysteme herunterzubrechen. Basierend auf diesen Informationen werden in Schritt ② Vermögenswerte aufgedeckt. Sie stellen jene Objekte dar, die für eine Entität von Wert sind. Je nach Stakeholder kann dies die einwandfreie Funktion eines Dienstes bedeuten oder dem Schutz von geistigem Eigentum entsprechen. Nach der Auflistung aller Vermögenswerte werden in der zweiten Phase die Bedrohungen identifiziert. Hierzu werden in Schritt ③ Sicherheitsziele von Vermögenswerten abgeleitet, die auf informationstechnischer Ebene beschrieben sind.

Konkret sind das beispielsweise die Vertraulichkeit, Integrität, Authentizität und Verfügbarkeit von Daten.

Mittels der identifizierten Sicherheitsziele werden in Schritt ④ jene Situationen gesucht, die zu einer Verletzung der Sicherheitsziele führen können. Aufgrund der Tatsache, dass diese Situationen nicht erwünscht sind, können diese auch als Missbrauchsfälle beschrieben werden (engl. Misuse-Cases [189]). Um diese aufzudecken wird betrachtet, wie ein Angreifer Eigenschaften des Systems ausnutzen kann, um eine Verletzung von Sicherheitszielen zu erreichen. Die dabei aufgedeckten Aspekte können als die – abstrakten – Schwachstellen des Systems verstanden werden.

In der letzten Phase wird das Risiko bestimmt, indem der Schaden für die relevanten Vermögenswerte in Schritt ⑤ und die Eintrittswahrscheinlichkeiten in Schritt ⑥ bestimmt werden. Letzteres stellt sich als eine Herausforderung im Security-Bereich dar, insofern keine empirischen Daten zu Eintrittswahrscheinlichkeiten zur Verfügung stehen und aufgrund dessen Abschätzungen mittels Metriken eingesetzt werden müssen. Hierzu gibt es zahlreiche Ansätze, die aus Teilmetriken die Eintrittswahrscheinlichkeit bestimmen wie beispielsweise die IEC 15408 [100] zeigt. Mit Schritt ⑦ und der Berechnung des Risikos anhand der Kombination von Schaden und Eintrittswahrscheinlichkeit schließt die Bedrohungsanalyse ab. Risikowerte können dabei in qualitativer oder quantitativer Form bestimmt werden und dienen in den nachfolgenden Entwicklungsphasen zur Priorisierung von Abschwächungsmaßnahmen (Gegenmaßnahmen). Letzteres ist insbesondere von Bedeutung, wenn eine Vielzahl von Bedrohungen identifiziert wurde und entschieden werden muss, welche vorrangig behandelt werden sollten. Eine Bedrohungsanalyse ist daher ein essenzieller Baustein jedes Security-Engineering Prozesses, unabhängig von der betrachteten Domäne. Ein für die Fahrzeugdomäne zugeschnittener Security-Engineering-Prozess – der eine Bedrohungsanalyse inkludiert – wird im nachfolgenden Abschnitt näher erläutert.

2.5.4 Eintrittswahrscheinlichkeiten mit dem CVSS

Im vorherigen Abschnitt wurde erläutertet, dass für die Bildung des Risikos ein Wert für den Schaden und ein Wert für die Eintrittswahrscheinlichkeit benötigt wird. In dieser Arbeit wird für die Bestimmung der Eintrittswahrscheinlichkeit das Common Vulnerability Scoring System (CVSS) in der Version v3.0 eingesetzt. Dieses Bewertungssystem ist verbreitet und erlaubt Security-Analysten numerische Werte für Schwachstellen zu vergeben [39]. Hierzu dienen drei Gruppen, *Base Metric Group*, *Temporal Metric Group* und *Environmental Metric Group*, die einen Wertebereich zwischen 0,0 und 10,0 aufweisen [69, Seite 10]. Jede Gruppe wird dabei aus mehreren Teilwerten gebildet, die einen qualitativen Wertebereich annehmen können. Die Basisgruppe (*Base Metric Group*) repräsentiert die grundlegenden Schwachstelleneigenschaften, die über die Zeit und die Umgebung konstant sind. Die zeitliche Metrik (*Temporal Metric*) bezieht sich hingegen auf den aktuellen Stand der Verfügbarkeit von Techniken, die zum Ausnutzen

der Schwachstelle dienen, der Verfügbarkeit von Schadcode, das Vorhandensein von Patches und inwieweit der Schwachstellenbeschreibung vertraut werden kann [69, Seite 12]. Darüber hinaus ermöglicht die dritte Gruppe (*Environmental Metric Group*) den CVSS-Wert auf spezifische Umgebungen anzupassen. So kann beispielsweise die Bedeutung der betroffenen IT-Vermögenswerte angepasst werden [69, Seite 14]. Alle drei Gruppen sind in Abbildung 2.9 mit den farbigen Rechtecken dargestellt.

Die *Base Metric Group* zeigt dabei zwei Untergruppen, welche die *Exploitability metrics* und *Impact metrics* darstellen. Erstere beschreibt, wie leicht die Schwachstelle ausgenutzt werden kann, was unter Einbezug der technischen Mittel beschrieben ist. Der darin enthaltene Zugriffsvector (*Access Vector*) erfasst aus welcher Distanz auf die Schwachstelle zugegriffen werden kann. Dies kann physikalisch, lokal, über ein Subnetz oder ein Netzwerk wie dem Internet möglich sein. Die Angriffskomplexität (*Attack Complexity*) beschreibt hingegen, welche besonderen Bedingungen erfüllt sein müssen, damit die Schwachstelle ausgenutzt werden kann. Hierbei sind besondere Bedingungen jene die der Angreifer nicht kontrollieren kann. Die dritte Untergruppe *Privileges Required* bezieht sich auf Rechte, die ein Angreifer erlangt haben muss, bevor er die Schwachstelle ausnutzen kann. Dies kann mit den drei quantitativen Werten keine, gering und hoch festgelegt werden. Darüber hinaus wird mit der Kategorie *User Interaction* erfasst, ob neben dem Angreifer ebenfalls der Anwender eine gewisse Handlung vollziehen muss, um den Angriff zu ermöglichen.

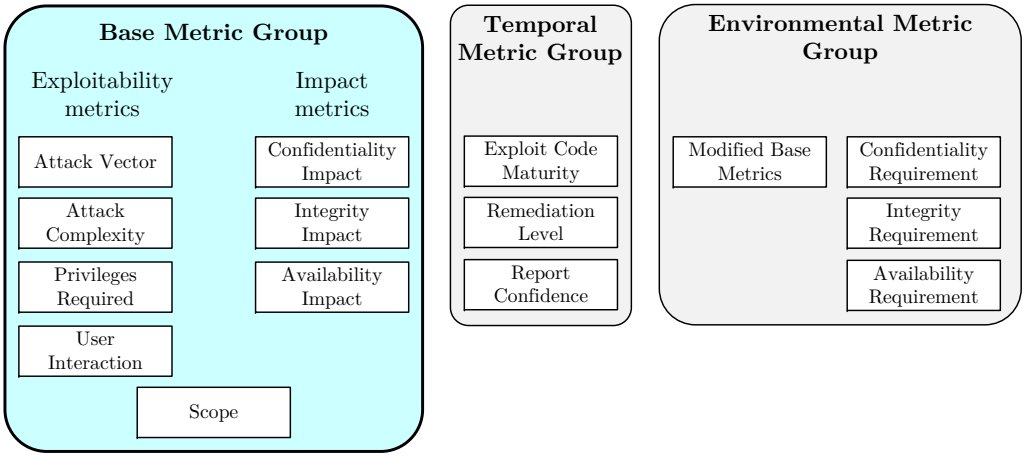


Abbildung 2.9: Die drei Kategorien, welche den CVSS-Wert bilden [69]. Gezeigt sind die Basisgruppe (*Base Metric Group*), die zeitliche Metrik (*Temporal Metrics*) sowie die auf die Umwelt bezogene Gruppe (*Environmental Metric Group*). Die blaue Färbung des linken Rechtecks hebt dabei jene Kategorie hervor, die in dieser Arbeit verwendet wurde.

Die Auswirkungsmetriken (*Impact Metrics*) erfassen außerdem die Konsequenzen für die Schutzziele: Vertraulichkeit (*Confidentiality Impact*), Integrität (*Integrity Impact*) und Verfügbarkeit (*Availability Impact*). Hierbei kann für jede der drei Klassen zwi-

schen den qualitativen Werten: Keine, gering und hoch unterschieden werden. Als letzte Einordnung erfasst die Basisgruppe mit dem Wert Geltungsbereich (*Scope*), ob die Ausnutzung einer Schwachstelle ebenso Einfluss auf andere Ressourcen hat. Dies entspricht beispielsweise einem Ausbruch aus einer Sandbox und dem anschließenden Löschen von Anwenderdateien [69, Seite 9]. Dies fließt neben den Werten der *Exploitability metrics* und *Impact metrics* in eine Berechnungsvorschrift [69, Seiten 18–19] ein, die den Wert der *Base Metric Group* formt.

2.6 Security-Engineering-Prozess für Fahrzeuge

Nachdem allgemeine Security-Attribute aufgezeigt und der Ablauf einer Bedrohungsanalyse bezüglich Security-Vermögenswerten präsentiert wurden, sind die nachfolgenden Unterabschnitte auf die Fahrzeug-Domäne fokussiert. Hierbei wird ein Security-Engineering-Prozess für Fahrzeuge präsentiert, der aus dem Cyber-Security Leitfaden der Society of Automotive Engineers (SAE) J3061 [173] entnommen ist. Der Leitfaden wird im Folgenden zusammengefasst und das auf Security erweiterte V-Modell wird präsentiert.

2.6.1 SAE J3061

Im Januar 2016 veröffentlichte die Society of Automotive Engineers (SAE) den Leitfaden J3061 [173], um die Cyber-Security in Fahrzeugen zu betrachten. Dieser bezieht neben kommerziellen Fahrzeugen wie Lastwagen oder Bussen ebenso militärische Fahrzeuge in die Betrachtung mit ein. Der Leitfaden basiert auf bewährten Verfahren (engl. best-practices) und definiert einen gesamtheitlichen Security-Lebenszyklus, welcher die Konzept-, Produktions-, Betriebs- und Ausmusterungsphase betrachtet [173, Seite 5]. Außerdem wird eine abstrakte Anleitung zur Verfügung gestellt, um Cyber-Security in einen bestehenden Entwicklungsprozess integrieren zu können. Darüber hinaus werden Methoden und Werkzeuge aufgezeigt, die beispielsweise für Bedrohungsanalysen [173, Seiten 70–101] oder Security-Testing [173, Seiten 127–128] verwendet werden können. Der Lebenszyklus basiert hierbei auf dem der ISO 26262 [99] und zeigt damit, dass die Safety beim Leitfaden mit einbezogen wird (Abbildung 2.10). Das ist damit zu begründen, das der Leitfaden safety-kritische Systeme als eine Teilmenge der security-kritischen Systeme sieht.

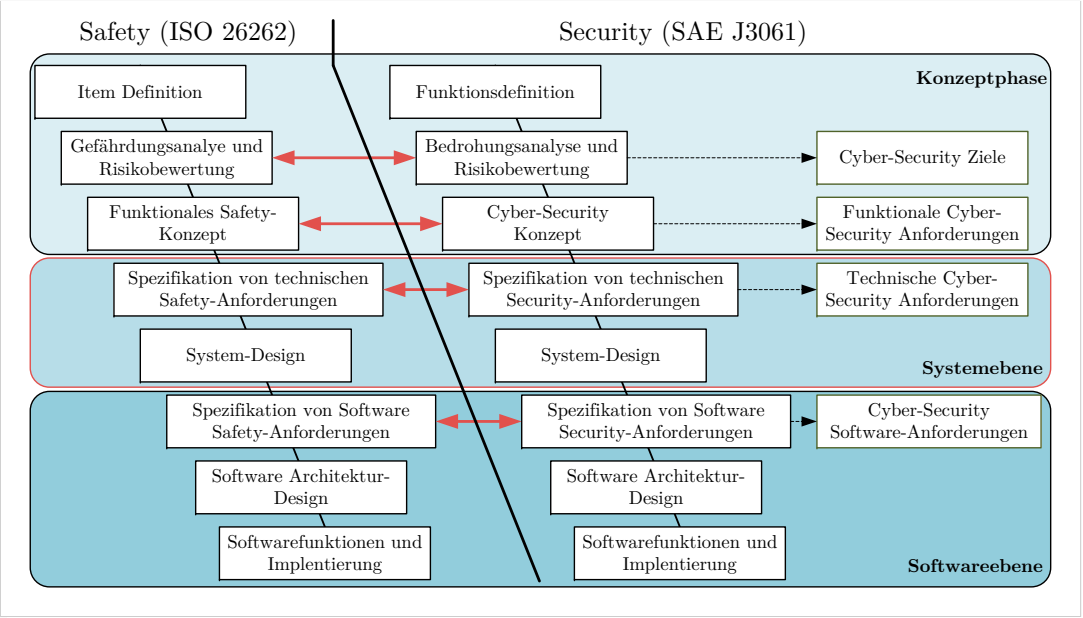


Abbildung 2.10: Gegengenüberstellung der linken Seite des V-Modells für die ISO 26262 (links) und die SAE J3061 (rechts), basierend auf [24] und [173, Seiten 40–43].

Dies zeigt ebenso die Gegenüberstellung von Safety und Security durch die jeweilige linke Seite des V-Modells für die ISO 26262 [99] (links) und der SAE J3061 (rechts). Außerdem zeigen die vier rechts gelegenen Rechtecke relevante Ergebnisartefakte, die auf die Security bezogen sind. In der Konzeptphase sind das die *Cyber-Security Ziele*, die bei der Bedrohungsanalyse identifiziert werden. Außerdem ergeben sich aus dem *Cyber-Security Konzept* die *funktionalen Cyber-Security Anforderungen*, die für die Spezifikation der technischen Security-Anforderungen benötigt werden.

Als letztes Ergebnisartefakt sind die *Cyber-Security Software-Anforderungen* aufgeführt, die zu konkreten Softwarefunktionen führen. An dieser Stelle sei angemerkt, dass ein Teil dieser Phasen ebenfalls für die Hardwareentwicklung durchlaufen werden muss. Konkret sind das alle Phasen nach der Konzeptphase. Neben den genannten Phasen zeigen rote Pfeile in Abbildung 2.10, mögliche Kommunikationspfade zwischen Safety- und Security-Artefakten. Leider fehlt es jedoch an einer genauen Beschreibung, wie ein Austausch umgesetzt beziehungsweise welche Artefakte sinnvoll auszutauschen sind [173, Seiten 40–43].

3

Analyse verwandter Arbeiten

In diesem Kapitel werden etablierte Methoden und Ansätze vorgestellt, die sich das Ziel setzen, Safety und Security in kombinierter Weise zu analysieren. Hierzu fasst das Kapitel die in der Literatur genannten Methoden zusammen und diskutiert die jeweiligen Vor- und Nachteile. Darüber hinaus können mit den Ausführungen in Anhang A die Methoden im Detail nachvollzogen werden.

Vor der Diskussion der verwandten Arbeiten werden im Folgenden die möglichen Integrationsmöglichkeiten von Gefährdungs- und Bedrohungsanalyse aufgezeigt. Dies soll es ermöglichen, die unterschiedlichen Ausprägungen für die Kombination von Safety- und Security-Analysen einordnen zu können. Eine gemeinsame Betrachtung beider Studienfelder kann nur erreicht werden, wenn beide Analysen nicht getrennt voneinander durchgeführt werden. Außerdem sollten Ergebnisse der Gefährdungsanalyse als Eingangsartefakte in die Security-Analyse übernommen werden, um gegenseitige Beeinflussungen erkennen zu können. Abbildung 3.1 zeigt hierzu die grundsätzlichen Kombinationen bei der Durchführung der Analysen.

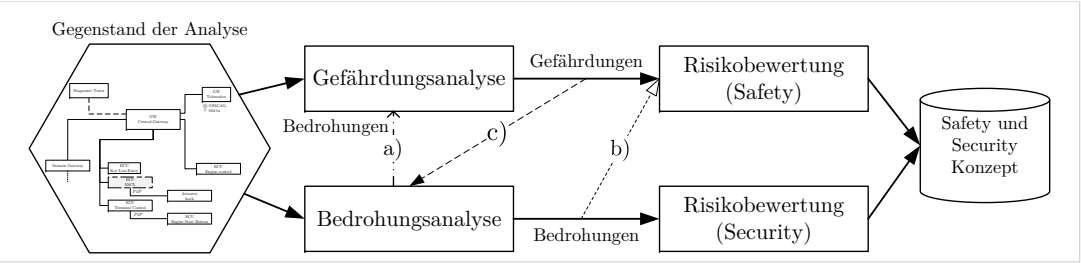


Abbildung 3.1: Grundsätzliche Methodenschritte bei Gefährdungs- und Bedrohungsanalysen, mit dem Analyse-Gegenstand als Eingangsartefakt und dem anschließenden Safety- und Security-Konzept. Weiterhin sind die möglichen Reihenfolgen beider Analyse-Techniken aufgezeigt.

Eine erste Möglichkeit zur gemeinsamen Betrachtung von Safety und Security ist das Weiterleiten von identifizierten Bedrohungen an die Gefährdungsanalyse und einer Betrachtung, ob Erstere ein safety-kritisches Verhalten erzeugen können. Dieses

Vorgehen entspricht Pfeil a) in Abbildung 3.1 und erfordert, dass Safety-Ingenieure die übernommenen Cyber-Bedrohungen verstehen und analysieren können. Aufgrund unterschiedlicher Terminologien in den Domänen ist dies i.A. nicht gegeben.

Eine zweite und mit Pfeil b) dargestellte Möglichkeit ist eine Übergabe identifizierter Cyber-Bedrohungen an die Safety-Risikobewertung und die Zuweisung eines safety-bezogenen Risikowertes, um den Safety-Bezug herzustellen (Abbildung 3.1). An dieser Stelle hängt die Qualität der Analyse allerdings von der Fähigkeit der Security-Ingenieure ab, safety-kritische Bedrohungen identifizieren zu können. Hiermit besteht der Nachteil, dass während der Bedrohungsidentifikation – ohne einen Safety-Fokus – Cyber-Bedrohungen übersehen werden könnten, die safety-relevant sind. Das ist mit der Tatsache zu erklären, dass beide Analyse-Typen erfahrungsorientiert sind. Es hängt somit stark von der Erfahrung des Analysten ab, welche Problem-Art (Gefährdung/Bedrohung) und wie viele unterschiedlichen Probleme identifiziert werden. Darüber hinaus ist es fraglich, wie eine Risikobewertung von Cyber-Bedrohungen mittels Safety-Metriken durchgeführt werden kann.

Die letzte und mit Pfeil c) dargestellte Variante ist das Weiterleiten identifizierter Gefährdungen an die Bedrohungsanalyse mit der Analyse, ob jene Gefährdungen durch Cyber-Bedrohungen ausgelöst werden können (Abbildung 3.1). Hierbei werden erzeugte Safety-Ergebnisse (Gefährdungen) für die Security-Analyse weiterverwendet. Außerdem wird ein implizierter Fokus auf die Safety gesetzt, da in der Bedrohungsanalyse überprüft wird, wie Gefährdungen durch Cyber-Bedrohungen ausgelöst werden können. Dies erfordert jedoch, dass eine gemeinsame Sprache etabliert ist, die den Zusammenhang zwischen Safety-Konsequenzen und Security-Verletzungen aufzeigt.

3.1 Auswahl verwandter Arbeiten

Für die Auswahl potenzieller Kandidaten wurde eine umfassende Literaturrecherche durchgeführt. Hierbei wurden die Ansätze ausgewählt, die sich das Ziel setzen in gemeinsamer Weise Safety und Security in Cyber-physischen Systeme (CPS) und Fahrzeugsystemen zu analysieren. Aus dieser Menge werden anschließend jene ausgewählt, die in mindestens zwei Veröffentlichungen ihren Ansatz darlegen. Dieses Vorgehen ist mit der Annahme zu begründen, dass bei fortlaufenden Publikationen eine Methodik aktiv weiterentwickelt wird und es sich nicht um einen einmaligen Vorschlag handelt. Neben der Auswahl relevanter Papiere, die in diesem Umfeld publiziert wurden, ist außerdem der für die Fahrzeugentwicklung vorgestellte Security-Leitfaden SAE J3061 [173] miteinbezogen worden. Der Leitfaden führt folgende Methoden, als mögliche Orientierungspunkte an, die potenziell auf die automotive Bedürfnisse angepasst werden können: EVITA, HEAVENS, TVRA und OCTAVE. Von diesen wurden ausschließlich die ersten beiden Ansätze weiter betrachtet. Zu begründen ist damit, dass beispielsweise das Threat, Vulnerability And Risk Assessment (TVRA) [65] für

die Anwendung in Telekommunikationsnetzen entwickelt wurde und ungeeignet für vernetzte CPS ist [173, Seite 75]. So eignet sich die Methodik nur für die Bedrohungsanalyse von Telekommunikationsschnittstellen und bindet darüber hinaus die Safety nicht in die Analyse mit ein, wie von den Forschern Islam et al. [92] und Macher et al. [126] angemerkt. Der Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE)-Ansatz [18] wurde ausgeschlossen, weil er für Security-Bewertungen von Unternehmensstrukturen entwickelt wurde und sich nicht auf Fahrzeugsysteme übertragen lässt [126, 153].

Schlussendlich wurden aus dem CPS-Umfeld vier Kandidaten ausgewählt, von denen drei bereits beispielhaft an Fahrzeugen-Systemen angewendet wurden. Aus der Fahrzeug-Domäne wurden hingegen drei Methoden ausgewählt. Jeder dieser Ansätze wird im Folgenden detailliert beschrieben. Anhand eines Diskussionsabschnitts wird auf nicht abgedeckte Bereiche hingewiesen. Die für jeden Ansatz relevanten Veröffentlichungen sind in Tabelle 3.1 aufgezeigt.

Tabelle 3.1: Übersicht der ausgewählten Ansätze und die zur Auswertung betrachteten Literaturquellen.

Einordnung	CPS	CPS	CPS	CPS	Automotive	Automotive	Automotive
Methode	STPA-SafeSec	CHASSIS	Six-Step Model	FMVEA	EVITA	HEAVENS	SAHARA
Analysierte Literatur	[22, 42, 70, 80, 89, 122, 123, 124, 152, 155, 198, 217]	[19, 41, 67, 108, 109, 116, 124, 126, 157, 158, 159, 160, 180, 192]	[14, 16, 34, 124, 138, 169, 170, 171, 172, 175]	[22, 41, 63, 91, 116, 124, 126, 179, 180, 196, 198, 206, 212]	[50, 56, 83, 84, 126, 139, 153, 167]	[21, 40, 50, 92, 141, 146, 153, 154]	[41, 50, 56, 124, 127, 128, 129, 139, 184]

Neben der Analyse der jeweiligen Ansätze werden außerdem die Unterschiede zu dem in dieser Arbeit präsentierten Ansatz (SGM) erläutert. Das Kapitel schließt mit einer Zusammenfassung der Methoden ab und diskutiert in Abschnitt 3.4.1 jene Bereiche die nicht abgedeckt werden.

3.2 Kombinierte Bedrohungsanalysen für cyber-physikalische Systeme

In den nachfolgenden Abschnitten werden vier Methoden zur kombinierten Analyse hinsichtlich Safety und Security aus dem Umfeld der CPS vorgestellt. Wie bereits erläutert ist es sinnvoll, ebenfalls Ansätze für vernetzte CPS-Systeme zu betrachten, da sich die grundsätzlichen Systemkomponenten von denen im Fahrzeug nur gering unterscheiden. Außerdem sind die eingesetzten Analysetechniken für das Studienfeld Safety in beiden Domänen oftmals identisch. Dies lässt sich mit dem gemeinsamen Safety-Standard IEC 61508 [47] erklären, der als Vorbild für die davon abgeleiteten Standards in der CPS- und Automotive-Domäne dient. Somit unterscheiden sich diese nur in Begrifflichkeiten und Metriken, welche zur Bewertung der Safety-Vorfälle eingesetzt werden. Hinsichtlich der Security und der Bedrohungsanalyse zeigen beide

Studienfelder keine Unterschiede. So werden oftmals gleiche Analysetechniken wie Microsoft's STRIDE [137] oder Angriffsbäume für die Analyse von CPS- und Fahrzeugsystemen eingesetzt. Entsprechend lässt sich argumentieren, dass eine Betrachtung von Methoden aus dem CPS-Umfeld für Fahrzeuge ebenfalls sinnvoll ist, da grundlegende Analyseschritte identisch sind.

3.2.1 STPA-SafeSec

Als erste Methode soll STPA-SafeSec vorgestellt werden, die einen Top-Down Ansatz verfolgt und sich das Ziel setzt, Safety und Security umfassend in die Analyse zu integrieren, um beide Felder gemeinsam betrachten zu können [70, 80, 124, 198]. Hierzu sieht der Ansatz ein System als Kontrollstrukturen mit Regelkreisen vor, die aus Sensoren, Kontrollern und Aktoren bestehen, welche physikalische Prozesse beeinflussen können. STPA-SafeSec betrachtet diese Kontrollstrukturen in Kombination mit unsicheren Kontrollaktionen, um potenzielle Abweichungen vom Systemverhalten festzustellen. Die Autoren argumentieren ihre Fokussierung auf Kontrollstrukturen mit dem Umstand, dass Security-Schwachstellen oft als nicht kritisch bewertet werden, weil deren Effekte auf physikalische Prozesse nicht vollständig verstanden sind.

STPA-SafeSec baut auf mehreren zuvor entwickelten Analyse-Methoden auf, die sukzessiv erweitert wurden. Die grundlegende Idee stammt vom Systems-Theoretic Accident Model and Processes (STAMP)-Ansatz ab, welcher für die Safety-Analyse von komplex vernetzten Systemen entwickelt wurde [123]. Es handelt sich dabei um einen auf Systemtheorie basierenden Ansatz, der ein Kausalitätsmodell zur Analyse von Unfällen verwendet. Hiermit können neben Komponentenausfällen und Ausfallketten auch komplexere Prozesse und unsichere Interaktionen zwischen Systemkomponenten betrachtet werden. Gegenüber traditionellen Methoden, die Safety als Fehlervermeidung im System ansehen, sieht STAMP die Safety als ein dynamisches Regelungsproblem. Basierend darauf wurde die Systems Theoretic Process Analysis (STPA) [122] entwickelt, um als neue Hazard-Analyse-Methodik zu dienen. Damit auch Cyber-Angriffe von einer Safety-Perspektive analysiert und die potenziellen Auswirkungen auf die Safety festgestellt werden konnten, mussten die Beziehungen zwischen Cyber-Angriffen und physikalischen Prozessen vollständig verstanden werden. Dies erforderte eine neue dedizierte Analyse-Technik, die mit der Systems- Theoretic Process Analysis for Security (STPA-Sec) [217] vorgestellt wurde. Sie folgt auf STPA und bezieht die grundlegenden Prinzipien des Studienfeldes Security in die Safety-Analyse mit ein.

Nach Ansicht der Autoren von STPA-SafeSec war STPA-Sec jedoch nur ein Versuch von Leveson et al. [123], um zu zeigen, dass STPA grundlegend fähig ist, die Security eines Systems analysieren zu können, indem Bedrohungen zum Ableiten von Security-Anforderungen verwendet werden [70]. Da STPA-Sec nur den Einfluss von Security auf Safety betrachtet und nicht vice versa, sahen die Autoren die Notwendigkeit STPA-Sec weiter zu verbessern [70, 89]. Hierzu kombinierten sie die Safety-Analysetechnik STPA

und die Security-Analysetechnik STPA-Sec zu einem einheitlichen Framework (STPA-SafeSec). Dieses ermöglicht das Aufdecken von Wechselwirkungen zwischen Safety und Security in vernetzten Systemen und ein Ableiten von Abschwächungsstrategien für Angriffe [155]. Die Ergebnisse der Methodik zeigen die potenziellen Bedrohungen, die durch spezifische Security oder Safety-Schwächen im System ausgelöst werden können. Damit sollen Strategien zur Abschwächung leichter entworfen und ihre Wirksamkeit bewertet werden können. Darüber hinaus stellt STPA-SafeSec die potenziellen Systemgefährdungen und Systemverletzungen heraus, welche zur Fehlfunktion einer spezifischen Komponente führen können. Nach Ansicht der Autoren bietet der Ansatz damit die Möglichkeit, den Zusammenhang und die Auswirkungen zwischen bestimmten Komponenten und deren Kommunikationsverbindungen zu identifizieren [70]. Die notwendigen Schritte die zu Analyse durchgeführt werden müssen, sind in Anhang A.1 ausgeführt.

Diskussion STPA-SafeSec

Bei STPA-SafeSec handelt es sich um einen neuen Ansatz, sodass umfangreiche Fallstudien und Werkzeuge zur Unterstützung der Methodik fehlen [152]. Darüber hinaus wird nicht diskutiert, wie identifizierte Bedrohungen ausgearbeitet werden oder welche Annahmen ihnen zugrunde liegen [155]. Die Autoren von STPA-SafeSec argumentieren mit der Notwendigkeit, dass die Security-Analyse auf der Komponenten- anstatt auf der Kontrollebene durchgeführt werden muss. Damit sei es möglich, identifizierte Security-Verletzungen auf Schwachstellen im System abzubilden und die Angriffsfläche effektiver modellieren zu können. Es werden allerdings keine Schwachstellenabhängigkeiten aufgezeigt, um Kombinationen oder Ketten von Schwachstellen zu identifizieren, die zu einer Systemkompromittierung führen können [22]. Die Risikoanalyse der SGM hingegen verwendet Schwachstellenketten, die unterschiedlichste Schwachstellenkombinationen aufzeigen. Der Ansatz zeigt außerdem keine Form der Priorisierung und somit kann er nicht für eine Risikobewertung und Risikoanalyse verwendet werden. Die SGM hingegen ermöglicht durch die Erzeugung von Angriffspfaden und der Einbindung von Risikowerten eine Risikobewertung.

3.2.2 CHASSIS

CHASSIS beschreibt einen Ansatz zur Kombination von Safety und Security, der für frühe Entwicklungsphasen geeignet ist [124]. Der Ansatz baut auf den Arbeiten von Raspoting [158] und Firesmith et al. [67] auf und verfolgt die Ideen der Forscher Srivatanakul et al. [192]. Die Methode unterstützt nach Ansicht der Autoren die Erhebung von Safety- und Security-Anforderungen hinsichtlich Gefährdungen und Ausfällen (Safety) sowie Bedrohungen und Schwachstellen (Security) [157]. Der Ansatz verwendet Unified Modeling Language (UML)-basierte Diagramme, um eine visuelle Modellierung und einen strukturierten Prozess für die Schadensbewertung zu ermöglichen [41, 116,

158]. Hierzu wird die Misuse-Case Technik mit textuellen Vorlagen und mit UML-Diagrammen kombiniert [159]. Für Ersteres werden die spezifizierten Anwendungsfälle (engl. use cases) invertiert und damit Missbrauchsfälle (engl. misuse cases) gewonnen, die einem Abweichen vom spezifizierten Verhalten entsprechen [103, 188]. Um die Kreativität während der Analyse zu steigern und einen strukturierten Prozess zu erhalten wird zudem HAZOP eingesetzt. Allerdings werden keine spezifischen Leitwörter für die HAZOP-Durchführung spezifiziert, sondern Standardleitwörter [48, S.16] wie *kein* oder *wenig* mit Confidentiality, Integrity, and Availability (CIA)-Attributen gekoppelt. Aufgrund dessen weist der Ansatz Ähnlichkeiten zur Arbeit von Winter et al. [209] auf. Die Verwendung von Misuse-Cases (MUCs) erlaubt, nach Ansicht der Autoren, einen frühen Fokus auf Security-Probleme im Entwicklungsprozess [158]. Zur Unterscheidung zwischen Gefährdungen und Bedrohungen im gleichen Misuse-Case-Diagram (MUCD) werden Gefährdungs- und Bedrohungs-Knoten farblich unterschieden (siehe Anhang). Das in der Methode verwendete Misuse-Case-Sequenz-Diagram (MUSD) erweitert das UML-Sequence-Diagram (SD) mit Sequenzen für die Visualisierung von Angriffen. Dies wird erreicht, indem beschrieben wird, wie ein Angreifer Schwachstellen ausnutzen kann [108]. Hinsichtlich der Safety und einer Analyse von Ausfällen erweitern das Failure-Sequence-Diagram (FSD) die Misuse-Cases mit Safety-Aspekten [160]. Die Methodik besteht aus mehreren Analyseschritten, die im Anhang A.1.1 erläutert sind.

Diskussion CHASSIS

Nach den Autoren von CHASSIS ermöglicht die Vereinheitlichung der Safety- und Security-Bewertung Vorteile. So seien durch gemeinsame Techniken und Artefakte die Zeit und der Lernaufwand gegenüber getrennten Bewertungen reduziert. Außerdem müssen Grundwissen und das gemeinsame Verständnis nur einmalig aufgebaut und erreicht werden [158]. Die gemeinsamen Artefakte wirken sich ebenso positiv auf spätere Phasen aus, weil sie wiederverwendet werden können. Die Komplexität einer gemeinsamen Betrachtung ist nach Ansicht der Autoren hingegen höher [158] und nach Macher et al. benötigt CHASSIS einen höheren Detailgrad in der Bedrohungsanalyse-Phase, der erst am Ende der Konzepthase zur Verfügung steht [126]. So sehen auch Schmittner et al. [126] CHASSIS als nicht geeignet für die Bewertung eines frühen Konzepts. Der Ansatz ist nach ihrer Meinung nur sinnvoll anzuwenden, sollte eine detaillierte Modellierung der Anwendungsfälle und Sequenzen möglich sein [126].

Obwohl CHASSIS Safety und Security mit der gleichen Methodik analysiert, werden die beiden Disziplinen in separaten Sitzungen analysiert und die Ergebnisse nachträglich kombiniert [180]. Außerdem zeigt CHASSIS keine Risikoanalyse und Bewertung für beide Domänen [41]. Daher ist die Eintrittswahrscheinlichkeit eines der Themen, die eine gemeinsame und vergleichbare Safety- und Security-Analyse erschweren [180]. Auch Kriaa et al. [116] sehen das Fehlen einer konventionellen Safety- und Security-Risikobewertung als kritisch an. Ein weiterer Nachteil besteht darin,

dass keine spezifische Liste von Leitwörtern für die Analyse mit HAZOP präsentiert wird, sodass die Standardmenge aus dem Safety-Umfeld verwendet werden muss [180]. Nach Lisova et al. [124] sind für die Anwendung von CHASSIS außerdem ein hoher Detaillierungsgrad und ein hohes Expertenwissen erforderlich. Dies erschwert nach Ansicht von Schmittner et al. [180] die Wiederverwendbarkeit von Analyse-Artefakten, da der Wissensstand verschiedener Analyse-Teams unterschiedlich sein kann, was die Analyse-Ergebnisse beeinflusst.

Im Gegensatz zu CHASSIS analysiert und bewertet die SGM Safety- und Security-Aspekte gemeinsam. Dies kann zu einer erhöhten Komplexität der Analyse führen, ermöglicht allerdings das Expertenwissen von einer Domäne in die andere zu übertragen. Hiermit sind prinzipiell bessere Ergebnisse zu erwarten. Weiterhin verwendet die SGM, Artefakte der Safety-Analyse – wie den Schadenswert (engl. Severity) – für die Security-Bewertung wieder. Durch die Koppelung der konventionellen Gefährdungsanalyse aus der ISO 26262 mit einer Security-Methodik, bleibt die SGM kompatibel zu etablierten Standards und nutzt wie CHASSIS den HAZOP-Ansatz mit Leitwörtern, um eine strukturierte Analyse zu etablieren. Gegenüber CHASSIS stellt SGM eine spezielle Menge von Leitwörtern zur Verfügung, welche auf die Security-Analyse fokussiert sind. Mittels der Erzeugung von Angriffspfaden aus einem formalen Modell und der Ableitung von Eintrittswahrscheinlichkeiten ist die SGM außerdem in der Lage quantitative Risiko-Bewertungen durchzuführen. Im Gegensatz dazu beinhaltet CHASSIS weder eine Risikobewertungs-Methodik noch eine Einbindung von Risikowerten aus der Safety-Analyse. Da sich die Analyse-Ergebnisse bei der SGM speichern lassen und die Risikoanalyse anhand von Angriffspfaden durchgeführt wird, können im Nachgang Analyse-Ergebnisse verglichen werden. Weiterhin ist es durch die Transformation der Eingangsinformationen in einen Zustandsautomaten möglich, den Einfluss neuer Komponenten auf ein bereits analysiertes System bestimmen zu können.

3.2.3 Six-Step Model

Das Six-Step Model (SSM) wurde zur Modellierung und Analyse von Safety und Security von CPS entwickelt. Der Ansatz soll nach Ansicht der Autoren eine umfassende Analyse hinsichtlich Safety und Security ermöglichen [14, 170]. Hierzu verwendet das Modell die folgenden sechs Dimensionen von CPS: Funktionen, Strukturen, Ausfälle, Safety-Gegenmaßnahmen, Cyber-Angriffe und Security-Gegenmaßnahmen. Beziehungen zwischen den einzelnen Dimensionen sind durch Beziehungs-Matrizen dargestellt (siehe Abbildung A.5). Funktionen und Strukturen dienen dabei als Wissensbasis, die aufzeigen, welche Effekte oder Konsequenzen Ausfälle und Cyber-Angriffe haben können. SSM basiert auf dem 3-Step Model [34], das eine Erweiterung des GTST-MLD-Ansatzes [138] mit Fehlern und Ausfällen darstellt.

GTST-MLD steht für Goal Tree Success Tree (GTST) und Master Logic Diagram (MLD), was in Abbildung 3.2 näher erläutert wird.

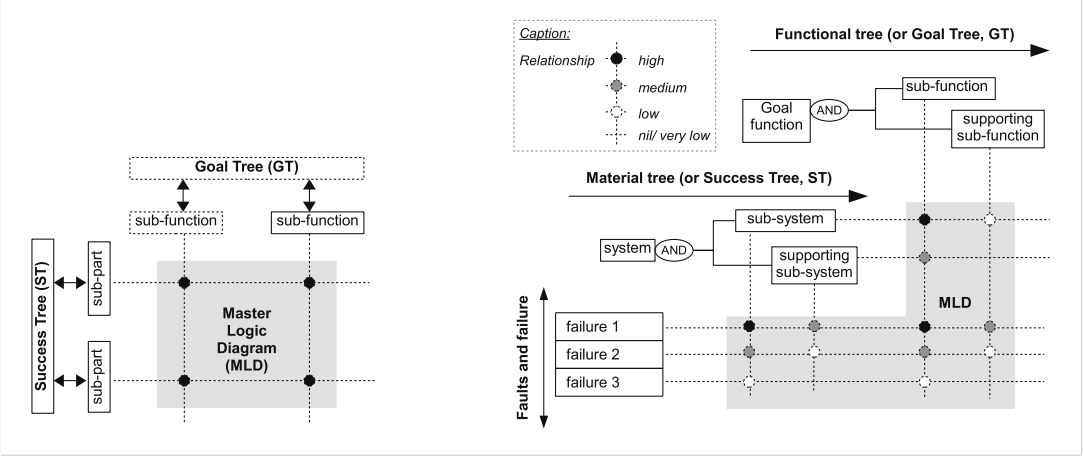


Abbildung 3.2: Links das GTST-MLD Framework [138] mit dem Goal Tree (GT) und dem Success Tree (ST), die aufzeigen, welches Teilsystem, welche Teilfunktionen realisiert, indem die Abhängigkeiten im MLD festgehalten werden. Rechts das 3-Step Model [34], das ebenfalls die GT, ST und das MLD aufzeigt, jedoch zusätzlich die Beziehungen von Fehlern und Ausfällen über das MLD einbindet. Die Farben der Kreise spiegeln die Stärke des Zusammenhangs [170].

Die (GTST) wurden 1980 eingeführt, um eine funktionale Zerlegung von komplexen CPS zu ermöglichen. Hierbei startet der GT mit einem funktionalen Ziel, das in Teilfunktionen zerlegt wird, die weiter in Haupt- und Nebenfunktionen unterschieden werden. Hauptfunktionen können direkt von der Zielfunktion abgeleitet werden und Nebenfunktionen die Hauptfunktionen unterstützen. Ein ST hingegen beschreibt die Struktur eines Systems, um die im GT identifizierte Funktion umzusetzen. Schlussendlich zeigt ein MLD die Beziehungen zwischen dem GT und dem ST auf. MLDs werden daher im 3-Step Model verwendet, um Zusammenhänge zwischen Fehlern oder Ausfällen von Elementen und Funktionen darzustellen.

Aufgrund des Umstands, dass das 3-Step Model nicht in der Lage ist, neben Fehlern und Ausfällen auch Schwachstellen analysieren zu können, erweitert SSM das 3-Step Model um die Betrachtung von: Cyber-Angriffen, Safety- und Security-Gegenmaßnahmen sowie die Modellierung von Beziehungen zwischen den beiden Studienfeldern [14]. Hierbei dienen die Systemfunktionen und -strukturen als Wissensbasis für die Safety- und Security-Analyse sowie zur Bestimmung von Effekten durch Ausfälle oder Angriffe auf das System. In Abbildung A.5 sind diese zusammengeführt und ergeben die SSM mit Matrizen zur Darstellung von Beziehungen.

Bei der Anwendung des Six-Step-Models werden sechs Schritte durchlaufen die in Anhang A.2 ausgeführt sind. Diese Schritte wurde durch die Autoren beispielhaft an einem Wasseraufbereitungssystem (SWaT [132]) angewendet, um die Anwendbarkeit des Ansatzes zu zeigen. Aufbauend darauf wurde der SSM-Ansatz durch Integration von Informationsfluss-Diagrammen (IFD) erweitert, um in Kombination mit Angriffsbäumen Cyber-Angriffe zu identifizieren [169, 171]. Diese stellen sich als Verhaltensdiagramme dar und zeigen den Informationsfluss zwischen Systemelementen

auf, was hilfreich bei der Identifizierung von Schwachstellen und Angriffen ist [169]. Die Autoren begründen ihre Entscheidung mit der Aussage, das Information Flow Diagrams (IFDs) bereits für Security-Analysen von CPS eingesetzt wurden und daher sinnvoll in SSM integrierbar sind, um einen strukturierten Ablauf bei der Identifizierung von Ausfällen und Cyber-Angriffen zu ermöglichen [16, 175].

Dazu zeigen die IFDs, welche Informationen, in welcher Richtung und mit welcher Frequenz ausgetauscht werden. Sie liefern damit wertvolle Informationen über Schwachstellen in Kommunikationskanälen [169]. Bei dem erweiterten SSM-Ansatz werden die IFDs mit CPS relevanten Elementen und Attributen angereichert und integriert (Anhang A.2.1). Dies geschieht in zwei Phasen: i) Konstruktion der IFDs anhand der Systeme und Verbindungen (Struktur-Matrizen), ii) die Verfeinerung der IFDs, indem die Verbindungskanäle näher beschrieben und in Informationsflüsse (Flussrichtung, Kontexte und Frequenz) konvertiert werden. Zur Begründung der sinnvollen Anwendung des erweiterten Six-Step Models wurde die Methode an einem autonomen Fahrzeug evaluiert [171, 172].

Diskussion Six-Step Model

Zusammenfassend kann gesagt werden, dass das SSM zu den kombinierten Safety- und Security-Ansätzen gehört, die Gefährdungen und Bedrohungen separat analysieren. Der Ansatz benötigt daher einen Integrationsschritt, um beide Aspekte zu vereinigen [124]. Dies geschieht durch die Matrix (A-B), welche die Beziehungen zwischen Angriffen und Ausfällen zeigt und nach der Hazard and Risk Analysis (HARA) und Threat and Risk Analysis (TARA) erzeugt wird (siehe Schritt 3 und 4 in Abbildung A.5). Dies kann zu einer hohen Anzahl von Iterationen führen, um Konflikte in Safety- und Security-Anforderungen aufzulösen [124]. Die SGM baut hingegen auf den Ergebnissen der Gefährdungsanalyse auf und betrachtet Security und Safety in sequentieller Weise. Dies ermöglicht ein einfacheres Auflösen von Konflikten, da bei der Bestimmung von Security-Maßnahmen die dazugehörigen Safety-Maßnahmen nachverfolgt werden können. Weiterhin ist durch die vielschichtigen Verknüpfungen der 15 Matrizen, die Skalierbarkeit der Methodik erschwert. So wächst der Umfang der Analyse schnell an, was eine manuelle Anwendung infrage stellt. Außerdem zeigt der Ansatz kein Vorgehen für eine Risikobewertung, sodass die Priorisierung von Angriffen oder Bedrohungen unklar ist. Bezogen auf die Safety könnte dies bei Kraftfahrzeugen durch die Bestimmung des ASIL aus der ISO 26262 ermöglicht werden. Bei Security hingegen ist nur der Verweis auf Security-Risiken mit *Security Risks* in Abbildung A.6 gegeben. Hier wäre es denkbar, Angriffsbäume zur Risiko-Analyse und -bewertung einzusetzen. Die SGM bietet hingegen eine Risikobewertung auf Basis von Angriffspfaden, die auch als Angriffsbäume dargestellt werden können.

3.2.4 FMVEA

Die FMVEA ist eine Methode zur parallelen Bewertung von Risiken hinsichtlich Safety und Security [41]. Dies bedeutet, dass die Safety- und Security-Analyse in paralleler Weise durchgeführt wird. Der Ansatz erweitert die existierende Safety-Analysetechnik aus der ISO 26262 und der IEC 60812 mit einer Security-Analyse [198]. Letztere setzt auf der Microsoft STRIDE Klassifizierung auf und ist komponenten-fokussiert [22, 126]. Initial wurde der Ansatz für Informationssysteme aus der Industrie entwickelt und zeigt ein Modell für die Analyse von Ursachen und Wirkungen hinsichtlich Safety und Security. Die Methode basiert auf der FMEA [91], die eine strukturierte Technik zur Untersuchung von Systemausfällen und deren Ursachen darstellt. Letztere wurde 1950 vom US Militär entwickelt, um die Zuverlässigkeit der militärischen Ausrüstung zu verbessern. FMVEA erweitert dieses Konzept durch Einbindung von Schwachstellen und Angriffen bezogen auf die Security eines Systems und erweitert den Gefährdungskatalog um relevante Security-Bedrohungen [196]. Das Ursache-Wirkungs-Modell ermöglicht nach Ansicht der Autoren die Untersuchung kombinierter Risiken für ein System [179]. Das Modell orientiert sich an Konzepten aus der Safety-Domäne und ist in Anhang A.3 aufgezeigt und näher erläutert.

Bei der FMEA setzt sich die Angriffswahrscheinlichkeit aus den Bedrohungseigenschaften (engl. Threat Properties) und der Systemanfälligkeit (engl. Suspectibility) zusammen. Dabei ergeben die Summe aus der Angreifermotivation und dessen Fähigkeiten (engl. Capabilities) die Bedrohungseigenschaften (engl. Threat Properties) (Tabelle A.1). Da nach Ansicht der Autoren auch Systemeigenschaften die Eintrittswahrscheinlichkeit beeinflussen, wird die Systemanfälligkeit einbezogen. Diese besteht aus der Ungewöhnlichkeit und Erreichbarkeit des Systems. Die Autoren verstehen diese als den benötigten Aufwand, um Zugriff auf das System zu erlangen. Die Ungewöhnlichkeit des Systems wird interpretiert als das Wissen, das ein Angreifer über ein System besitzt. Es wird davon ausgegangen, dass ein Angreifer über gewöhnliche Systeme mehr Wissen besitzt und damit der Aufwand zum Auffinden und Ausnutzen von Schwachstellen geringer ist. Dieses Vorgehen orientiert sich an dem Vorgehen von Damage, Reproducibility, Exploitability, Affected users, Discoverability (DREAD) [186] und dem Open Web Application Security Project (OWASP) [148], was zu einer semi-quantitativen Bewertung führt.

Diskussion FMVEA

FMVEA basiert auf der FMEA-Methodik [134] und hat somit das Ziel, Fehlerursachen aufzufinden. Dies bedeutet jedoch, dass bereits Fehler identifiziert sein müssen, bevor die Analyse durchgeführt werden kann. Aufgrund dessen ist der Ansatz nicht für frühe Entwicklungsphasen geeignet, sondern für die Designphase, wo bereits durch andere Analyse-Techniken – wie HAZOP – Fehler identifiziert sind. Nach Ansicht von Macher et al. [126] und Chockalingam et al. [41] benötigt der Ansatz Details über das Systemde-

sign, sodass auch diese Forscher den Ansatz für die Analyse in der Entwicklungsphase als ungeeignet ansehen. Bei der Analyse der Ausfälle und Bedrohungen besteht außerdem kein Austausch zwischen beiden Analysepfaden (siehe Schritt 4 in Abbildung A.8). Dies kann zur Situation führen, dass Safety- und Security-Wechselwirkungen übersehen werden [63]. Aufgrund dessen führt die Methode die Safety- und Security-Analyse unabhängig voneinander durch [116, 126]. Die Verknüpfung von Safety und Security findet damit erst in der gemeinsamen Bewertung des Schadens für die Ausfall- und Bedrohungsmodi statt (Schritt 5 Abbildung A.8). Die hier präsentierte SGM analysiert dagegen die Safety und Security gemeinsam und konzentriert sich auf Synergieeffekte bei der Analyse von Safety und Security. FMVEA besitzt außerdem die Einschränkungen, dass nur einzelne Ursachen eines Effektes betrachtet werden, mehrstufige Angriffe können nicht betrachtet werden [206]. Die SGM ist andererseits durch die Generierung von Angriffspfaden in der Lage, mehrstufige Angriffe zu identifizieren. Ebenfalls als kritisch zu sehen ist es, dass kein Vorgehen gezeigt wird, bei dem Ausfälle und Bedrohungen identifiziert werden [126], was die Erstanwendung der Methodik erschwert. SGM bietet durch die Anwendung von Leitwörtern einen vereinfachten Einstieg in die Analyse-Methodik, wie zwei Studien aus den Bereichen Automobil und SCADA zeigen [212]. Nach Lisova et al. deckt FMVEA außerdem die Anforderungserfassungsphase nicht ab, SGM hingegen kann nach Ansicht von Lisova et al. [124] dafür eingesetzt werden.

3.3 Kombinierte Bedrohungsanalysen für automotive Systeme

Nach Auflistung der Analyse-Methoden für CPS-Systeme werden im Folgenden Ansätze betrachtet, die explizit für Fahrzeugsysteme entwickelt wurden. Als Pionier muss hier das EVITA-Projekt genannt werden, das als Erstes einen Vorschlag für die Kombination von Safety und Security im Fahrzeug zeigte und für einige nachfolgenden Methoden als Vorbild diente. Insbesondere die von EVITA aus der ISO 15408 [100] entnommenen und für Fahrzeuge angepassten Risikometriken sind vielfach von anderen Methoden übernommen und modifiziert worden.

3.3.1 EVITA

Das E-safety Vehicle Intrusion Protected Applications (EVITA)-Projekt setzt sich zur Aufgabe, die Fahrzeug-zu-Fahrzeug (Vehicle-to-Vehicle (V2V)) und die Fahrzeug-zu-Infrastruktur-Kommunikation (Vehicle-to-Infrastructure (V2I)) sicher und vertrauenswürdig zu entwickeln. Erreicht werden soll dies durch den Entwurf geeigneter Architekturen für Fahrzeugnetzwerke, welche die notwendigen Primitive für Security-Maßnahmen bereitstellen und dabei kosteneffektiv sind [84]. Das Projekt startet mit einer Analyse von Cyber-Sicherheitsanforderungen bis hin zum Design von Security-Mechanismen in Hard- und Software, die an einem Prototyp gezeigt wurden. Im Fokus stehen dabei schnurlose Verbindungen, die nach außen gerichtet sind und Funktio-

nalitäten, wie vorausschauende Verkehrsampelschaltungen zur Benachrichtigung von Notfallbremsungen. Diese teils safety-kritischen Funktionen erfordern nach EVITA besondere Security-Maßnahmen, die zum Zeitpunkt des Projekts im Fahrzeug noch nicht verfügbar waren [84]. Als Fundament der neuen Maßnahmen sieht das Projekt die Notwendigkeit einer geeigneten Hardware-Sicherheits-Architektur, die es zu bestimmen gilt. Zur Ableitung der notwendigen Security-Maßnahmen und der damit einhergehenden Bedrohungsanalyse definiert das Projekt eine beispielhafte E/E-Architektur die in Anhang A.4 gezeigt ist.

Zur Dokumentation der gefundenen Bedrohungen setzt das Vorhaben Angriffsbäume ein, die auch die Risikoanalyse unterstützen. Außerdem dienen die Bäume zur Identifikation gemeinsamer Angriffsmuster. Hierzu definiert das Projekt einen einheitlichen Aufbau des Baums, der in Abbildung A.10 gezeigt ist. Die Autoren von EVITA weisen darauf hin, dass ein Gruppieren der Risikoanalyse-Ergebnisse anhand des Vorkommens der Bedrohungen für eine der festgelegten Risikostufen sinnvoll sein kann. Nach den Autoren ist es möglich, eine Priorisierung auf bestimmte Angriffe zu erreichen. So sei jedoch zu beachten, dass Angriffe, die einen geringen Risikowert haben, aber häufiger auftreten, wichtiger sein können als höhere Risikowerte, die seltener auftreten. Hinsichtlich des Auffindens von Cyber-Bedrohungen sehen die Autoren die Problematik, dass der Vorgang stark erfahrungsorientiert ist und keine Vollständigkeit gezeigt werden kann. Bezogen auf die Angriffsbäume und deren Knoten trifft EVITA allerdings die Aussage, dass eine vollständige Abdeckung der Security-Anforderungen gewährleistet ist [83].

Diskussion EVITA

Der Ansatz betrachtet Cyber-Bedrohungen aus verschiedenen Blickwinkeln (operational, safety, privacy und financial), die sich in nur geringen Unterschieden im Security-Level widerspiegeln [50]. Dabei benötigt das Vorgehen im Allgemeinen viele Details für die Klassifikation der Bedrohungen. Dies kann grundsätzlich zu einer präziseren Klassifikation führen, die Kategorien weisen jedoch ein hohes Diskussionspotenzial auf [126], was die Bestimmung des Angriffspotenzials erschwert [83, 153]. Weiterhin muss angenommen werden, dass kein Safety-Analyst zur Analyse hinzugezogen wird, sodass der Security-Analyst festlegen muss, ob eine Cyber-Bedrohung eine Safety-Relevanz besitzt und ob diese analysiert werden muss. Dies setzt fundiertes Wissen über das Studienfeld Safety beim Security-Analysten voraus, was im Allgemeinen nicht angenommen werden kann, da beide Studienfelder unterschiedlich sind. Weiterhin werden keine Ergebnisse aus der Safety-Analyse wiederverwendet, was zu einem Übersehen von relevanten Safety-Fällen führen kann. Der SGM-Ansatz hingegen übernimmt Safety-Ergebnisse aus der HARA, sodass die Situation einer unterschiedlichen Klassifizierung von Bedrohungen nicht eintritt. Darüber hinaus sind die mit der SGM erzeugten Angriffspfade eindeutig definiert. Hier repräsentiert die Wurzel den Eintrittspunkt des Angreifers,

die Teilknoten das Ausnutzen von Schwachstellen und der Endknoten entspricht dem Erreichen des Angreiferziels. Durch die Verwendung der CVSS-Werte ergeben sich außerdem keine Unterschiede bei Bewertungen gleicher Bedrohungen, da die Werte einmalig bestimmt und wiederverwendet werden. Bei EVITA hingegen muss der Analyst bei jeder Risikobewertung den Wert anhand der oben genannten Metriken von Neuem bestimmen, was zu unterschiedlichen Risikowerten führen kann.

3.3.2 HEAVENS

Das HEAVENS-Projekt setzt sich zum Ziel, einen systematischen Ansatz für das Ableiten von Security-Anforderungen für Kraftfahrzeuge bereitzustellen. Es betrachtet dabei Safety und Security in E/E-Architekturen und bezieht den Safety-Standard ISO 26262 sowie die AUTomotive Open System ARchitecture (AUTOSAR) in die Betrachtung mit ein [146, Seite 20]. HEAVENS fokussiert sich auf die Identifikation von Security-Schwachstellen unter der Betrachtung von Schutzzielen (engl. security objectives), die von EVITA abgeleitet sind (siehe Abschnitt 3.3.1). Die primären Security-Attribute entsprechen einer erweiterten Version der CIA-Triad und wurden vom OCTAVE-Framework übernommen [40]. Konkret werden die Attribute: Vertraulichkeit, Integrität, Verfügbarkeit, Authentizität, Autorisierung, Nichtleugbarkeit, Datenschutz und Aktualität verwendet [146, Seite 24]. Das Vorgehen zur Modellierung der Cyber-Bedrohungen und die jeweiligen Schritte basieren auf den Arbeiten von Myagmar et. al. [141] und dem Microsoft Security Development Lifecycle (SDL) [197]. Das Ergebnis dieses Vorgehens ist ein Bedrohungsprofil des zu betrachtenden Systems [197]. Hierzu setzt HEAVENS ein Modell ein, das auf die Identifikation von Besitzern, Vermögenswerten, Schwachstellen, Gegenmaßnahmen, Bedrohungen und Angreifern abzielt und in Abbildung A.11 dargestellt ist. Auffällig hierbei ist, dass das HEAVENS-Modell große Ähnlichkeiten mit den Security-Konzepten und -beziehungen, aus der ISO 15408 (Common Criteria (CC)) aufweist [100, Seite 24] und in Abbildung A.11 gezeigt ist. Basierend auf einem abgeleiteten Bedrohungsprofil legt HEAVENS Security-Anforderungen fest, die der Notation der funktionalen Sicherheit entsprechen. Hierzu führt das Projekt das Security-Level (SL) für identifizierte Bedrohungen ein, was sich auf einen Vermögenswert des Analyseobjekts (Target of Evaluation (TOE)) bezieht. Der Ablauf bestehend aus drei Schritten und ist in Anhang A.5 erläutert.

Nach der Risikobewertung geht die Methodik in das Ableiten der Security-Anforderungen über. Das Projekt sieht eine Security-Anforderung als eine Funktion von Assets, Bedrohungen, dem SL und den Security-Attributen (Abbildung A.12). Die abgeleiteten Anforderungen sind auf der Ebene funktionaler Anforderungen einzuordnen und beziehen sich somit auf die Konzeptphase. Damit sind sie als abstrakte Security-Anforderungen zu verstehen, die in den nachfolgenden Entwicklungsphasen sukzessive konkretisiert werden müssen [146, Seite 51]. Basierend auf den Security-Anforderungen legt HEAVENS schlussendlich die Security-Mechanismen fest. Hierzu wird die Annah-

me getroffen, dass ausgewählte Mechanismen im späteren Verlauf der Entwicklung korrekt implementiert werden. Aufgrund dessen betrachtet das Projekt nicht die Wirksamkeit der ausgewählten Mechanismen, sondern wählt aus einer Menge von Mechanismen aus, die zu den identifizierten Bedrohungen passen.

Diskussion HEAVENS

Das HEAVENS-Projekt verwendet ausschließlich Begrifflichkeiten aus der Security-Domäne, sodass das Fehlen von Safety-Begrifflichkeiten den Austausch zwischen beiden Domänen erschwert [146, Seite 48]. Darüber hinaus wird erst in der Risikobewertung die Safety-Relevanz von Cyber-Bedrohungen bestimmt und die beiden Studienfelder verknüpft werden, was einer getrennten Analyse von Safety und Security entspricht [92]. SGM zeigt hingegen die Beziehungen zwischen Ausfällen und Bedrohungen auf und verknüpft damit beide Studienfelder.

Bei der Risikoanalyse weisen die Autoren von HEAVENS daraufhin, dass die Bestimmung von Bedrohungs- und Schadenslevel ein entscheidender Aspekt für die Konsistenz in der Bewertung ist. Hierzu sollten gleiche Prinzipien für die Bestimmung von Parametern angewandt werden. Die Aussage steht jedoch im Kontrast zur Verwendung der Parameter: *Expertise*, *Knowledge about TOE*, *Window of opportunity* und *Equipment*, die – wie bei EVITA – eine hohe Subjektivität aufweisen [21, Seite 59]. Weiterhin benötigt jeder Risikobewertungsfaktor eine umfassende Diskussion bei der Bewertung [153]. Die Forscher Cui und Sabaliauskaite merken außerdem den hohen Aufwand bei der Bestimmung der individuellen Bedrohungen an [50]. Auch die Werkzeugunterstützung für die Bedrohungsanalyse mit dem Threat Modeling Tool (TMT) zeigt ein Defizit. So stammen die verwendeten TMT-Vorlagen aus der klassischen IT und sind nicht für das Automotive-Umfeld und dessen Komponenten entwickelt worden [146, Seite 77]. Dies führt zu einer Problematik bei der Modellierung von Informationsflüssen von Bussystemen mit Broadcast-Funktionalität, wie dem CAN-Bus. So ist es mit dem TMT nicht möglich, bei einem Angriff auf einen der CAN-Teilnehmer den Einfluss auf andere CAN-Teilnehmer zu bestimmen. Weiter kritisch ist am TMT zu sehen, dass mit diesem keine mehrstufigen Angriffe identifiziert werden können. Somit obliegt es dem Analysten die Zusammenhänge zwischen den Bedrohungen selbst aufzudecken, um mehrstufige Angriffe zu erkennen. SGM hingegen ist in der Lage neben den Verletzungen von Vermögenswerten auch mehrstufige Angriffe aufzudecken. Im letzten Schritt von HEAVENS und der Auswahl von Security-Mechanismen bezieht der Ansatz die Angreiferklasse mit ein, sodass bei einer falschen Abschätzung des Angreifers das Risiko besteht, unzureichende Mechanismen zu implementieren [21, Seite 72]. An dieser Stele sei allerdings angemerkt, dass diese Problematik ebenfalls für die anderen Methoden in diesem Kapitel zutrifft, wobei ausschließlich EVITA und HEAVENS konkret auf die Implementierung von Gegenmaßnahmen eingehen.

3.3.3 SAHARA

Die SAHARA Methode ist ein Ergebnis des SEcurity and SAfety MOdelling (SeSaMo)-Projektes [128, 184] und setzt sich als Ziel, Safety- und Security-Analysen zu vereinen. Nach Ansicht der Autoren wird dies durch die Kombinierung der Gefährdungsanalyse (HARA) mit einer Bedrohungsanalyse-Methodik erreicht [129]. Darüber hinaus soll die Methode die Auswahl einer geeigneten Anzahl von Gegenmaßnahmen für die identifizierten Probleme unterstützen. Hierzu soll SAHARA in der Konzeptphase der Fahrzeugentwicklung eingesetzt werden [129]. Das Vorgehen ist zweistufig und die Ergebnisse der Security-Analyse werden im Einklang mit dem ASIL aus der HARA bewertet. Die Autoren sehen dieses Merkmal als einen wichtigen Vorteil der Methode an [128]. SAHARA ist somit eine sequenziell integrierte Methode zur Bewertung von Sicherheitsrisiken [41], da die Analyse und Bewertung von Security und Safety in einer definierten Reihenfolge stattfindet. Genauer gesagt, werden Bedrohungen zuerst auf ihr Security-Risiko und danach auf ihr Safety-Risiko bewertet. Das hierfür notwendige Vorgehen ist in Anhang A.6 detailliert erläutert.

Bei der Risikoanalyse verwendet der Ansatz eine Quantifizierung des erforderlichen Know-hows und der notwendigen Werkzeuge, gegenüber einer direkten Abschätzung der Eintrittswahrscheinlichkeit für die Bedrohungen (Gleichung (A.1)). Die Autoren begründen dieses Vorgehen mit der Aussage, dass dies stärker im Einklang mit der HARA-Klassifizierung steht und die vorgeschlagenen Metriken über die gesamte Lebensdauer identisch bleiben [127, 128]. Eine Evaluierung mit einem Batterie Managment System (BMS) zeigte eine um 34 % gesteigerte Anzahl von Gefährdungssituationen (hazardous situations) gegenüber dem klassischen HARA Ansatz [128], womit die Autoren die sinnvolle Anwendbarkeit des Ansatzes begründen.

Diskussion SAHARA

SAHARA verfolgt den Ansatz, identifizierte Security-Bedrohungen auf ihre Safety-Relevanz zu überprüfen und diese gegebenenfalls an die Gefährdungsanalyse weiterzureichen. Die Safety-Relevanz einer Bedrohung ist somit von der korrekten Abschätzung der Auswirkungen durch den Security-Analysten abhängig [56, S.17]. Außerdem kann durch den fehlenden Austausch zwischen der HARA und der Security-Analyse nicht geprüft werden, ob Gefährdungen durch Security-Verletzungen ausgelöst werden können. Im Unterschied dazu übernimmt die SGM die Ergebnisse aus der HARA und überprüft, ob Gefährdungen durch Verletzungen der Security entstehen können. Ferner ist ein hoher Aufwand notwendig, um das Security-Level für jede Bedrohung zu bestimmen und es besteht ein hohes Diskussionspotenzial bei der Festlegung des Einfluss- (siehe Tabelle A.8) und Eintritts-Levels (siehe Tabellen A.9 und A.10) [126]. SAHARA fokussiert sich bei den Angriffszielen alleinig auf Safety-Anwendungsfälle und ist damit nicht in der Lage, Angriffe zu betrachten, die mehr als ein Security-Ziel betreffen [139]. Weiterhin werden die safety-relevanten Bedrohungen anhand der ISO 26262 Metriken

bewertet, was in diesem Fall zu einer Bewertung mit dem Controllability-Wert aus der ISO 26262 führt. Dieses Vorgehen ist fraglich, da der Angreifer bei der Planung seines Angriffs die Möglichkeit hat, Maßnahmen zu ergreifen die dem Eingreifen des Fahrers und dem Abwenden der Gefahrensituation (engl. hazardous situation) entgegenwirken. Basierend auf dieser Annahme kann der Controllability-Wert nicht bestimmt werden. SGM hingegen kann durch die Zuordnung der Leitwörter auf Informationsgüter – neben den Safety-Anwendungsfällen – weitere Security-Ziele betrachten. Außerdem wird der kritische Controllability-Wert bei SGM nicht verwendet. Nach Lisova et. al [124] deckt der Ansatz zudem nicht die Phase der Anforderungserfassung (Requirements Engineering) ab. SGM ist andererseits fähig, anhand erzeugter Artefakte, Security-Anforderungen in der Konzeptphase abzuleiten [2, 124]. Darüber hinaus können die Artefakte der SGM bei der Durchführung von Penetrationstests unterstützen [3], SAHARA gibt hierzu keine Empfehlungen. Bezogen auf Remote-Angriffe und Angriffe gegen eine Fahrzeugflotte fehlt SAHARA ein Klassifizierungsschema hinsichtlich des Schadenspotenzials und der betroffenen Anwender [127]. Das für die Risikobewertung der SGM aufgestellte Angreifer-Modell hingegen betrachte unterschiedliche Eintrittspunkte, die ein Angreifer nehmen kann, bei dem zwischen lokalen und remote-basierten Angriffen unterschieden wird.

3.4 Zusammenfassung und Diskussion bestehender Analysemethoden

In diesem Abschnitt werden die oben genannten (Abschnitt 3.3 und Abschnitt 3.2) Bedrohungsanalyse-Methoden zusammengefasst und anhand von Kriterien eine Bewertung vorgenommen.

- **Entwicklungsphase** beschreibt, in welchem Teil der Konzeptphase die Methodik eingesetzt werden sollte. Die Aufteilung dieser Phase in einen frühen und späten Abschnitt lässt sich anhand der jeweiligen benötigten Artefakte erklären. In der frühen Konzeptphase stehen die funktionale Systembeschreibung, die vorläufigen Architekturelemente und deren grundsätzliche Interaktionen zur Verfügung. Nach der ISO 26262 [99] entspricht dieser Abschnitt der Durchführung der Gefährdungsanalyse. In der späten Konzeptphase hingegen wird die funktionale Systembeschreibung durch technische Details angereichert und Systemkomponenten weiter konkretisiert. Hier wird beispielsweise das funktionale Safety-Konzept mittels FMEA oder Fehlerbäumen bestimmt, sodass Fehler und Fehlerketten zur Verfügung stehen.

- **Benötigte Informationen** beschreiben, welches Wissen über das System zur Analyse vorliegen muss, damit die Methodik ausgeführt werden kann. Grundsätzlich benötigen die Methoden die zu realisierenden Funktionen, die Struktur des Systems, dessen Komponenten sowie deren Kommunikation untereinander.

- **Koppelung der Safety- und Security-Analyse** beschreibt in welcher Reihenfolge die Safety- und Security-Analysen durchgeführt werden und wie beide Studienfelder gekoppelt sind. Letztere können getrennt voneinander analysiert oder Artefakte und Beziehungen eines Studienfeldes für die Analyse des anderen Studienfeldes weiter verwendet werden. Als Referenz der Kopplung ist zusätzlich die jeweilige Kombinationsmöglichkeit aus Abbildung 3.1 beschrieben.

- **Bedrohungsmodellierung** beschreibt die Technik, die zur Identifikation der Cyber-Bedrohungen eingesetzt wird.

- **Identifikation mehrstufiger Bedrohungen** zeigt auf, ob es möglich ist, mehrstufige Bedrohungen identifizieren zu können. Die ist insbesondere für die hoch vernetzten Fahrzeugsysteme von Relevanz. So hat sich bei einer Analyse bisheriger Angriffe auf Fahrzeuge gezeigt, dass für ~ 78 % der bekannten Angriffe mehrere Angriffsschritte notwendig waren [5, 13]. Das lässt sich mit der Situation erklären, dass ein Angreifer zuerst einen Eintrittsvektor in das Fahrzeug identifizieren muss, bevor er mit dem eigentlichen Angriffsziel (intern verbaute Electronic Control Units (ECUs)) kommunizieren kann.

- **Risikoanalyse und Bewertung** zeigt, ob in der betrachteten Methodik Risiken anhand von Risikoanalysen identifiziert und bewertet werden. Die Aufnahme dieser Kategorie lässt sich damit erklären, dass einige Methoden keine Vorschläge zum Umgang mit Risiken machen und andere nur einen Teil betrachten.

- **Werkzeugunterstützung** legt dar, ob Werkzeuge oder Vorlagen zur Unterstützung bereitgestellt werden. Dies umfasst software- oder textbasierte Unterstützung bei der Durchführung der Methodik.

- **Aufwand** kategorisiert die Aufwendungen, die für die Durchführung der jeweiligen Methodik notwendig sind. Das Intervall ist dreistufig gegliedert: gering, mittel und hoch.

- **Nachvollziehbarkeit** beschreibt, inwieweit eine nicht an der Analyse beteiligte Person die Ergebnisse der Methode nachvollziehen kann. Konkret stellt sich die Frage, ob ein Auditor die identifizierten Bedrohungen nachverfolgen und die festgelegten Risikowerte verstehen kann. Das Intervall ist dreistufig: gering, mittel und hoch.

- **Ableitung von Safety- und Security-Anforderungen** gibt an, ob neben der Identifikation und Bewertung von Bedrohungen auch Anforderungen für das Security-Konzept abgeleitet werden. Es wird unterschieden zwischen keiner Anforderungsanalyse und einer Anforderungsanalyse hinsichtlich einem oder beider Studienfelder.

Tabelle 3.2: Vergleich kombinierte Bedrohungsanalyse-Methoden aus der CPS Domäne.

Methode/Kriterium	STPA-SafeSec	CHASSIS	Six-Step Model	FMVEA
Entwicklungsphase	Späte Konzeptphase	Frühe Konzeptphase	Späte Konzeptphase	Späte Konzeptphase
Notwendiges Wissen	Systemfunktionen, Systembeschreibung (detalliertes Systemwissen)	Systemfunktionen, Systembeschreibung, detaillierte Anwendungsfälle, Sequenzen	Systemfunktionen, Systembeschreibung, Systemstruktur	Systemfunktionen
Koppelung der Safety- und Security-Analyse	Zuerst wird Safety, anschließend Security analysiert (entspricht Pfad *c)* in Abbildung 3.1)	Getrennte Analysen, wobei Beziehungen zwischen den Analyseergebnissen hergestellt werden	Getrennte Analysen, wobei Beziehungen zwischen Ausfällen und Angriffen hergestellt werden	Getrennte Analysen, mit gemeinsamer Risikobewertung
Bedrohungsmodellierung	Systemtheoretischer Ansatz mit Kausalfaktoren hinsichtlich Integrität und Verfügbarkeit	UML-Diagramme, Misuse-Case-Technik und HAZOP	Ziel- und Erfolgsbäume sowie Beziehungsmatrizen	FMEA mit STRIDE und Datenflussdiagrammen
Identifikation mehrstufiger Bedrohungen	Ja, wenn Angriffsbäume eingesetzt werden	Ja, jedoch auf UML-Sequenz-Diagramme beschränkt	Nicht gezeigt, jedoch durch Integration von Angriffsbäumen grundsätzlich möglich	Nicht möglich
Risiko-Analyse und Bewertung	Nicht beschrieben	Nicht beschrieben	Nicht beschrieben	Metriken für Schadenswert und Eintrittswahrscheinlichkeiten
Werkzeugunterstützung	Nicht gezeigt	Misuse-Case und UML Sequenz-Diagramme	Nicht gezeigt	Microsoft TMT
Aufwand	Hoch	Hoch	Hoch	Mittel
Nachvollziehbarkeit	Mittel	Gering bis Mittel	Hoch	Mittel
Ableitung von Safety- und Security-Anforderungen	Safety- und Security-Anforderungen	Safety- und Security-Anforderungen	Safety- und Security-Anforderungen	Nicht gezeigt

Tabelle 3.3: Vergleich kombinierte Bedrohungsanalyse-Methoden, welche explizit für die automotive Domäne entwickelt wurden.

Methode/Kriterium	SAHARA	EVITA	HEAVENS	SGM
Entwicklungsphase	Frühe Konzeptphase	Frühe Konzeptphase	Frühe Konzeptphase	Frühe Konzeptphase
Notwendiges Wissen	Systemfunktionen, Systembeschreibung, Systemstruktur (E/E-Architektur), Informationsflüsse	Systemfunktionen, Systembeschreibung, Systemstruktur (E/E-Architektur), Informationsflüsse	Systemfunktionen, Systembeschreibung, Systemstruktur (E/E-Architektur), Informationsflüsse	Systemfunktionen, Systembeschreibung, Systemstruktur (E/E-Architektur), Informationsflüsse
Koppelung der Safety- und Security-Analyse	Zuerst Security, anschließend Safety (entspricht Pfad *a)* in Abbildung 3.1)	Getrennte Analysen, mit gemeinsame Risikobewertung	Getrennte Analysen, mit gemeinsame Risikobewertung	Zuerst Safety, anschließend Security
Bedrohungsmodellierung	STRIDE	Use-Case-Technik, funktionale Pfade und Angriffsbäume	STRIDE	Leitwörter, IFD und Angriffspfade/Bäume
Identifikation mehrstufiger Bedrohungen	Nein, da ausschließlich STRIDE verwendet wird	Ja, mittels Angriffsbäumen	Nein, da ausschließlich STRIDE verwendet wird	Ja, mittels automatisiert generierten Angriffspfaden
Risikoanalyse und Bewertung	Metriken für Schadenswert und Eintrittswahrscheinlichkeiten	Metriken für Schadenswert und Eintrittswahrscheinlichkeiten, die auf der ISO 15408 [100] basieren	Metriken für Schadenswert und Eintrittswahrscheinlichkeiten, die aus dem EVITA-Projekt übernommen wurden	CVSS, Common Weakness Enumeration (CWE) und Angreifermodel
Werkzeug-Unterstützung	Nicht gezeigt	Werkzeug zum Zeichnen von Angriffsbäumen	Microsoft TMT	Framework für die Bedrohungsmodelierung und die Risikobewertung
Aufwand	Mittel	Hoch	Hoch	Gering bis Mittel
Nachvollziehbarkeit	Mittel	Mittel	Hoch	Mittel
Ableitung von Safety- und Security-Anforderungen	Nicht gezeigt	Security-Anforderungen	Security-Anforderungen	Security-Anforderungen

3.4.1 Abschließende Diskussion und Bewertung

Bei der Betrachtung der Methoden fällt auf, dass sich diese im Detailgrad der Analyse unterscheiden. Methoden aus dem CPS-Umfeld binden im Allgemeinen die Safety-Analyse und deren Artefakte stärker in die Betrachtung von Cyber-Bedrohungen mit ein als die fahrzeugspezifischen Methoden. Sie zeigen außerdem eine tiefer gehende Betrachtung der Systemkomponenten als jene Methoden aus dem Automotive-Umfeld. Besonders detailreich analysiert STPA-SafeSec das zu betrachtende System. So analysiert kein anderer Ansatz in dem Detailgrad die Regelschleifen eines Systems, wie es hier der Fall ist. Obwohl solch eine detaillierte Analyse eine höhere Anzahl von Cyber-Bedrohungen aufdecken kann, werden mehr Systemdetails benötigt. Dies führt zur Situation, dass STPA-SafeSec erst zum Ende der Konzeptphase eingesetzt werden kann.

Bis auf CHASSIS kann für die CPS-bezogenen Methoden in Tabelle 3.3 grundsätzlich gesagt werden, dass sie für eine Durchführung in einer späten Konzeptphase konzipiert sind und einen höheren Detailgrad der Eingangsinformationen benötigen. Ansätze, die im Fahrzeug Anwendung finden, konzentrieren sich hingegen auf frühe Konzeptphasen und agieren auf einem höheren Abstraktionslevel. Das Ziel hierbei ist, möglichst frühzeitig Bedrohungen aufzufinden, um Maßnahmen direkt und nicht nachgelagert ergreifen zu müssen, was mit dem Security-By-Design Paradigma im Einklang steht. Dies ist von besonderer Bedeutung, da Entwicklungszyklen von circa drei Jahren [177, Seite 21] nur wenig Spielraum für Revisionen lassen. Hieraus erklärt sich auch der Wunsch nach einem geringen Aufwand für die Analyse, da nur eine geringe Zeitspanne zur Verfügung steht. Im Kontrast zu diesem Wunsch steht jedoch die geringe Werkzeugunterstützung, die von den etablierten Analysemethodiken bereitgestellt wird. So zeigt sich, dass nur geringfügig erzeugtes Wissen wiederverwendet wird, was die Effizienz der Analysen beeinträchtigt. Insbesondere für automotive Systeme, die häufig nur geringe Unterschiede bei Folgegenerationen aufweisen, ist das Fehlen von generischem Wissen (wie Angriffe) ungünstig.

Die automotiven Ansätze zeigen außerdem nur eine geringe Koppelung beider Studienfelder, was in einem niedrigeren Grad der Wiederverwendung von Safety-Artefakten für die Bedrohungsanalyse resultiert. Es zeigen außerdem nur drei der acht Ansätze in den Tabellen 3.2 und 3.3 (ohne die SGM) die Fähigkeit, mehrstufige Angriffe/Bedrohungen identifizieren zu können, wobei keiner der drei Ansätze computergestützt mehrstufige Angriffe aufdeckt. So werden ausschließlich Angriffsbäume und UML-Diagramme verwendet, die anhand des Analysten-Wissens erzeugt werden müssen. Computergestützte Ansätze fehlen an dieser Stelle.

Weiterhin ist auffallend, dass drei der vier Methoden aus dem CPS-Umfeld keine Risikoanalyse und Bewertung vorschlagen (siehe Zeile 6 in Tabelle 3.2). Hinsichtlich des Kriteriums der Nachvollziehbarkeit von Analyseergebnissen muss zwischen der

Beschreibung von Bedrohungen und der Risikobewertung unterschieden werden. Bei Ersterem zeigt das Six-Step Model durch seine Verknüpfungsmatrizen eine hohe Nachvollziehbarkeit, ebenfalls STPA-SafeSec und CHASSIS sind durch ihre gewählten Modellierungsansätze gut nachzuvollziehen. Die automotiven Ansätze zeigen hingegen eine schlechtere Nachvollziehbarkeit und sind in dieser Kategorie den CPS-Ansätzen unterlegen. Bezogen auf die Risikobewertung sind im Besonderen die Ergebnisse des EVITA und HEAVENS-Projektes schwer nachzuvollziehen. Dies lässt sich mit den subjektiven Risikometriken erklären, die bei diesen Ansätzen eingesetzt werden. Auf Seite der CPS-Ansätze kann hier keine Bewertung abgegeben werden, da ausschließlich die FMVEA eine Risiko-Metrik vorschlägt.

Zusammenfassend kann aus der Analyse der etablierten Bedrohungsanalysemethoden gesagt werden, dass es Lücken bei der Wiederverwendung von Safety-Artefakten für die Security-Analyse gibt. Dies führt zum einen zur Situation, dass Bedrohungen bei der Analyse übersehen werden können die eine Safety-Relevanz besitzen. Zum anderen führt die fehlende Wiederverwendung von erzeugten Analyse-Artefakten dazu, das Bedrohungen teilweise doppelt identifiziert werden, da kein Austausch zwischen den Analyse-Typen besteht. Damit reduziert sich die Effizienz der Analysen, was im Konflikt zu dem verfügbaren Zeitrahmen für Bedrohungsanalysen in der Fahrzeugentwicklung steht. Die SGM verwendet hingegen – die in der Safety-Analyse identifizierten – Gefährdungen für die Bedrohungsanalyse wieder. Damit reduziert sich die Gefahr, Bedrohungen zu übersehen, die eine Verletzung der Safety implizieren können. Außerdem steigt die Effizienz der kombinierten Analyse, da durch den Austausch von Analyse-Artefakten, Bedrohungen nicht mehrmals analysiert werden. Zusätzlich beschleunigt die Übernahme des Schweregrades aus der Safety-Analyse, die Risikobewertung von safety-relevanten Bedrohungen.

Bezogen auf die Bedrohungsmodellierungen zeigt sich, dass nur eine geringe Anzahl der existierenden Methoden in der Lage ist, mehrstufige Bedrohungen zu identifizieren können. Im Automotive-Umfeld ist dazu ausschließlich der EVITA-Ansatz in der Lage. Dies ist allerdings als nachteilig anzusehen, da die in dieser Arbeit präsentierte Angriffssammlung [5] mehrheitlich mehrstufige Angriffe aufweist. Obwohl der EVITA-Ansatz mittels Angriffsbäumen Angriffsketten aufdecken kann, wird deren Modellierung manuell durchgeführt. Die fehlende Werkzeugunterstützung erhöht dabei den Aufwand für die Durchführung der Analyse. Das Software-Framework der SGM ermöglicht hingegen eine automatisierte Erzeugung von Angriffspfaden und reduziert damit den Aufwand für die Analyse. Außerdem ermöglicht die Einbindung von historischem Wissen – das in Form von bekannten Angriffen in die Analyse eingebunden wird – eine Wiederverwendung von Security-Artefakten.

Hinsichtlich der Priorisierung von Bedrohungen präsentieren nur vier der betrachteten Ansätze eine Risikoanalyse. Die hierzu vorgeschlagenen Metriken setzen allerdings auf Kategorien, die ein hohes Diskussionspotenzial aufweisen und damit die Ver-

gleichbarkeit von Risikoanalysen erschweren. Weiterhin führt die Subjektivität der Kategorien dazu, dass ein Analyst bei der Festlegung von konkreten Werten, einen hohen Arbeitsaufwand hat. Diesem steht die automatisierte Risikobewertung mit dem SGM-Framework gegenüber. Hierzu sind Schwachstellen nach dem CVSS-Konzept vorab bewertetet und in einer Datenbasis abgelegt. Diese werden den Angriffsschritten zugeordnet, sodass für einen Angriffspfad – unter Einbezug des Schweregrades aus der Safety-Analyse – der Risikowert computergestützt bestimmt werden kann.

Abschließend lässt sich sagen, dass die Literatur keine kombinierte Bedrohungsanalysemethodik aufweist, die in der Konzeptphase eingesetzt werden kann und einen hohen Grad an Wiederverwendung von Safety-Artefakten für Security-Analysen ermöglicht. Darüber hinaus zeigt keine der Methoden eine Werkzeugunterstürzung für die automatisierte Modellierung von mehrstufigen Angriffen in der Bedrohungsanalyse. Ebenfalls stellt die Literatur keine Methodik zur Verfügung, die den hohen Aufwand bei der Risikobewertung durch computergestützte Ansätze reduzieren kann.

4

Herleitung der Security Guideword Methode

Anknüpfend an den Stand der Technik sollen im Folgenden die Anforderungen an die kombinierte Bedrohungsanalyse-Methodik aufgelistet und die jeweiligen Lösungsbausteine vorgestellt werden. Die Anforderungen an die Methodik ergeben sich aus den Forschungsfragen in Kapitel 1 und den identifizierten Defiziten bestehender Ansätze in Abschnitt 3.4 und Abschnitt 3.4.1.

Die Anforderungen an die neue Bedrohungsanalysemethodik lassen sich folgendermaßen zusammenfassen:

1. Die Methode muss in der Lage sein Cyber-Bedrohungen zu identifizieren, welche die Safety beeinflussen.
2. Die Methode muss in einer frühen Entwicklungsphase mit wenigen verfügbaren Informationen anwendbar sein.
3. Die Methode muss einfach anwendbar sein und Analysten strukturiert durch die Analyse führen.
4. Die Methode muss eine hohe Anzahl von Artefakten aus der Safety-Analyse wiederverwenden.
5. Die Risikobewertung soll im Einklang mit dem Vorgehen und den Begrifflichkeiten der Safety-Analyse sein.
6. Die Methode soll in der Weise konzipiert sein, dass diese von Safety- und Security-Ingenieuren verstanden werden kann.

Hierbei ist das übergeordnete Ziel die Identifikation von Cyber-Bedrohungen, welche die Safety beeinflussen können. Damit dies erreicht werden kann, wird im Folgenden

ein kausales Modell hergeleitet, welches die Beziehungen zwischen Cyber-Bedrohungen und Safety-Gefährdungen aufzeigt (Abschnitt 4.1.1). Darauf aufbauend und unter Einbezug von CPS-Eigenschaften wird anschließend ein Metamodell vorgestellt, das aufzeigt, welche Zusammenhänge zwischen Safety und Security bestehen und wie diese für die Identifikation von safety-relevanten Bedrohungen verwendet werden können (Abschnitt 4.1.2). Das Metamodell dient außerdem zur Erfassung der relevanten Aspekte der Studienfelder Safety und Security und dient im Laufe der Arbeit als Fundament für die Algorithmen der entwickelten Software-Applikation in Kapitel 7 und Anhang E.

4.1 Modell für die kombinierte Bedrohungsanalyse

Um ein übergeordnetes Bild von Safety und Security zu erhalten, wird im Folgenden ein kausales Modell aufgezeigt, das die Beziehungen zwischen Safety und Security darlegt (Abbildung 4.1). Auf diesem Fundament wird anschließend in Abschnitt 4.1.2 ein Metamodell abgeleitet, welches die identifizierten Beziehungen und Abhängigkeiten auf Cyber-physischen Systeme (CPS) überträgt (Abbildung 4.4).

4.1.1 Kausales Modell

Bezogen auf das Studienfeld Safety zeigt Abbildung 4.1, dass die grundlegenden Ursachen für einen Schaden an der Umwelt durch Fehler (engl. fault) in den E/E-Systemen entstehen können. Die Wirkkette zeigt dabei, dass Fehler zu einem Ausfall (engl. failure) des Systems führen und gegebenenfalls eine Gefährdung auslösen (④). Gekoppelt mit den Betriebsszenarien des Fahrzeugs, wie einer Autobahnfahrt, ergeben sich die Gefahrensituationen, die zu einem Unfall führen und einen Schaden für die Personen bedeuten können. Ob ein Unfall eintritt, hängt primär von der Art der Gefährdung und den Gegebenheiten in der Umwelt ab. So kann beispielsweise das unerwünschte Verlassen der Fahrbahn in ein weitläufiges Feld weniger kritisch sein als beispielsweise in einem Wald. Letztere Situation zeigt dabei eine deutlich höhere Wahrscheinlichkeit für einen kritischen Schaden. Der blaue Bereich in Abbildung 4.1 zeigt hingegen die Zusammenhänge der Security-Artefakte. So lässt sich die Cyber-Schwäche (engl. weakness) im E/E-System als die grundlegende Ursache für einen Cyber-Angriff identifizieren (⑤). Das Beifügen des Terms *Cyber-*, soll an dieser Stelle verdeutlichen, dass es sich um Schwächen handelt, welche die Kompromittierung eines IT-basierten Systems erlauben. Das können zum einen Schwächen in der Software wie fehlende Längenüberprüfungen oder das Ablegen von Geheimnissen im Quellcode sein, zum anderen auch offene Hardware-Schnittstellen oder ungeschützte Hardware-Bausteine, die ein Auslesen oder Modifizieren von Daten ermöglichen. Allgemein kann in dieser Arbeit eine Cyber-Schwäche als eine nicht erwünschte Systemeigenschaft gesehen werden, die es erlaubt, einen informationstechnischen Vermögenswert zu verletzen. Sollte die Cyber-Schwäche durch eine bestimmte Technik (engl. exploit) ausgenutzt

werden können, um die informationstechnischen Vermögensgüter des Systems zu verletzen, so wird von einer Schwachstelle (engl. vulnerability) gesprochen. Diese ermöglicht es dem Angreifer, seinen Cyber-Angriff auszuführen.

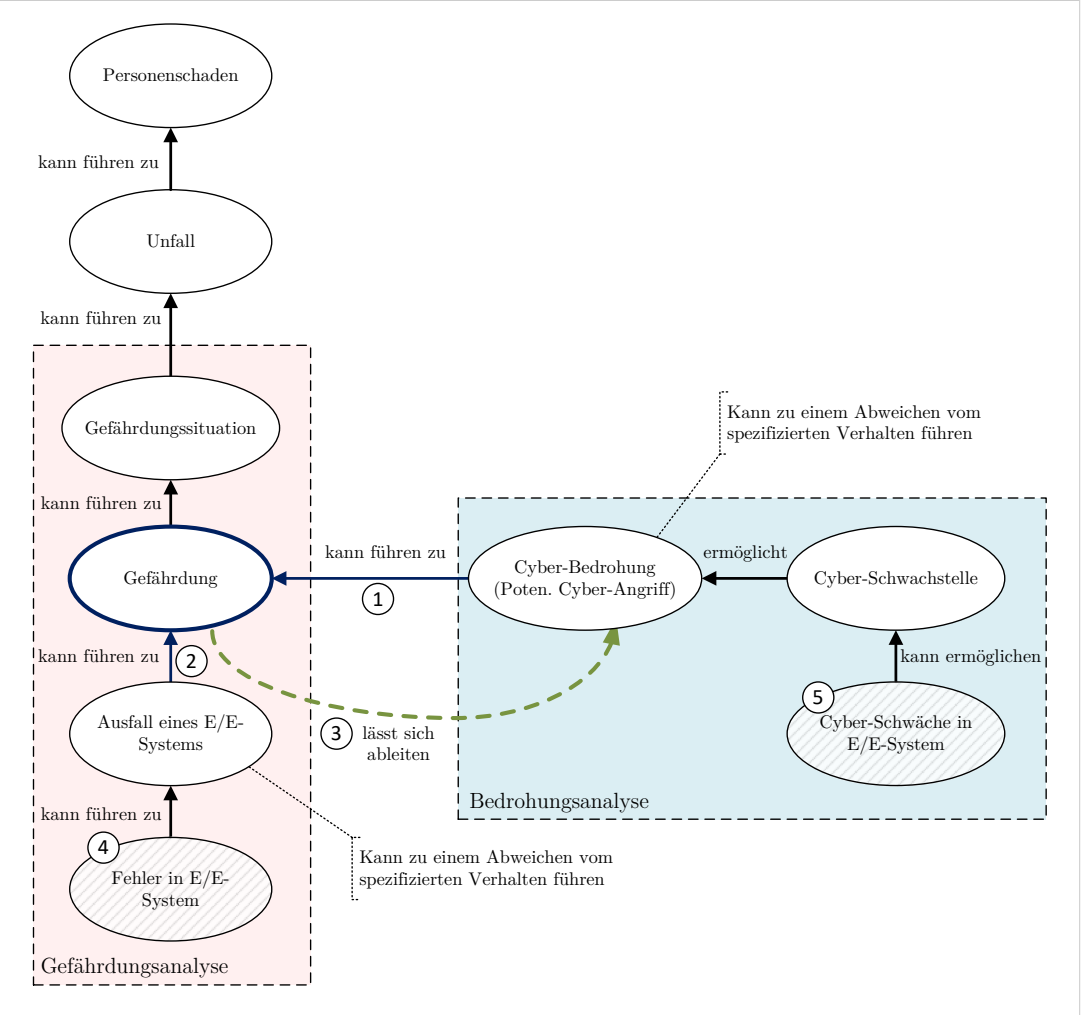


Abbildung 4.1: Kausales Modell, das Abbildung 1.1 um die Zusammenhänge von Safety und Security erweitert. Die Ellipsen im rot-gefärbten Bereich zeigen die Artefakte der Gefährdungsanalyse und die im hellblauen die der Bedrohungsanalyse. Ellipsen, die grau schraffiert sind, repräsentieren die jeweiligen grundlegenden Ursachen (engl. root causes). Weiterhin zeigen die Markierungen (1) und (2), das sowohl Safety- als auch Security-Ursachen zu einer Gefährdung führen können.

Darüber hinaus besteht ein weiterer Zusammenhang zwischen dem Begriff Cyber-Angriff und Bedrohung. Letzteres wird in dieser Arbeit als potenzieller Cyber-Angriff gesehen, bei dem noch geklärt werden kann, ob dieser jemals von einer Entität durchgeführt wird [68, Seite 34]. Die Bedrohungsanalyse hat damit die Aufgabe, denkbare Cyber-Angriffe aufzufinden, die theoretisch am vorliegenden Analysegegenstand durchgeführt werden könnten. Ähnlich verhält es sich, wenn die Safety-Analyse (Gefährdungsanalyse) betrachtet wird, denn auch hier haften Wahrscheinlichkeiten den Safety-Artefakten an. Konkret haften den Ausfällen in Abbildung 4.1, Wahrschein-

lichkeiten an. Diese beschreiben, wie wahrscheinlich es ist, dass der Ausfall in einem E/E-System stattfinden wird. Diese Ausfallwahrscheinlichkeiten lassen sich – im Falle von Hardwarekomponenten – bei der Safety-Analyse statistisch bestimmen. Für Cyber-Bedrohungen ist das grundsätzlich nicht möglich, was einen Unterschied beider Studienfelder aufzeigt [218]. Eine Gemeinsamkeit hingegen besteht bei den Auswirkungen, die Ausfälle (Safety) und Cyber-Bedrohungen (Security) auslösen können. Beides kann zu Abweichungen vom spezifizierten Verhalten führen, was in einer Gefährdung resultieren kann (① und ② in Abbildung 4.1). So kann ein *Ausfall* eines E/E-Systems oder eine *Cyber-Bedrohung* eine Gefährdung auslösen, was als kausale Verknüpfung zwischen Safety und Security angesehen werden kann. Bezogen auf die in Abschnitt 3.4.1 identifizierte Lücke zur unzureichenden Koppelung und Wiederverwendung von Artefakten aus der Safety-Analyse für die Security-Analyse lässt sich hiermit Ersteres folgern. Da eine Gefährdung als Ursache einen *Ausfall* oder eine *Cyber-Bedrohung* besitzen kann, lassen sich aus identifizierten Gefährdungen *Ausfälle* und *Cyber-Bedrohungen* ableiten. Dieses deduktive Vorgehen ist in Abbildung 4.1 mit ③ hervorgehoben und beschreibt die Möglichkeit zum Ableiten von safety-relevanten Cyber-Bedrohungen aus bestehenden Safety-Gefährdungen. Aus Sicht einer Wiederverwendbarkeit von Safety-Artefakten erscheint es somit als sinnvoll, identifizierte Gefährdungen auf mögliche Cyber-Bedrohungen zu untersuchen. Darüber hinaus kann der Schadenswert (Severity [99]), die für Gefährdungssituationen vergeben wird, ebenfalls für die Risikobewertung von safety-revelevanten Cyber-Bedrohungen verwendetet werden. So hängt der Schaden für die Umwelt ausschließlich von der Art der Gefährdung ab und nicht von den Ursachen der Gefährdung. Sollte demnach ein *Ausfall* (Safety) oder eine *Cyber-Bedrohung* (Security) die gleiche Gefährdung auslösen, so kann der Schadenswert der dazugehörigen Gefährdungssituation für die Risikobewertung der Cyber-Bedrohung übernommen werden.

4.1.2 Metamodell

Basierend auf den zuvor erläuterten Zusammenhängen wird in diesem Abschnitt ein Metamodell vorgestellt, das als Fundament für die Entwicklung der kombinierten Bedrohungsanalyse (SGM) dient. Außerdem werden die im Modell gezeigten Zusammenhänge für den im Abschnitt 7.6 entwickelten Algorithmus 1 verwendet. Hierbei wird dem Paradigma gefolgt, eine Abstraktion zu wählen, die nur die notwendigsten Elemente in den Modellen aufzeigt und nur Pfeile aufzuzeigen, die einen expliziten Zusammenhang darstellen. Als notwendig werden hierbei jene verstanden, die für die kombinierte Bedrohungsanalyse und die entwickelten Algorithmen erforderlich sind.

Der Aspekt, dass ein Schaden für die Umwelt entsteht, ist in Abbildung 4.2 mit der Klasse *Vermögenswert* repräsentiert. Diese beschreibt mit den Klassenelementen *Safety* und *Security*, Qualitätsfaktoren, die für eine Entität von Wert sind. Eine mögliche Entität ist hier der Original Equipment Manufacturer (OEM), der aufgrund des Pro-

dukthaftungsgesetzes verpflichtet ist, die Safety eines Fahrzeugs zu gewährleisten. Ebenfalls kann ein Schaden durch eine Verhaltensabweichung des Fahrzeugs in seiner Umgebung ausgelöst werden, was in Abbildung 4.2 mit der Klasse *Unfall* repräsentiert wird (Abschnitt 4.1.1). Dabei kann ein Unfall aus einer *Gefährdungssituation* resultieren, die eine Eintrittswahrscheinlichkeit besitzt und nach der ISO 26262 [99] eine Kontrollierbarkeit aufweist. Letztere beschreibt, inwieweit ein Fahrzeugführer die Konsequenzen der auftretenden Gefährdungssituation durch sein Handeln abschwächen oder verhindern kann.

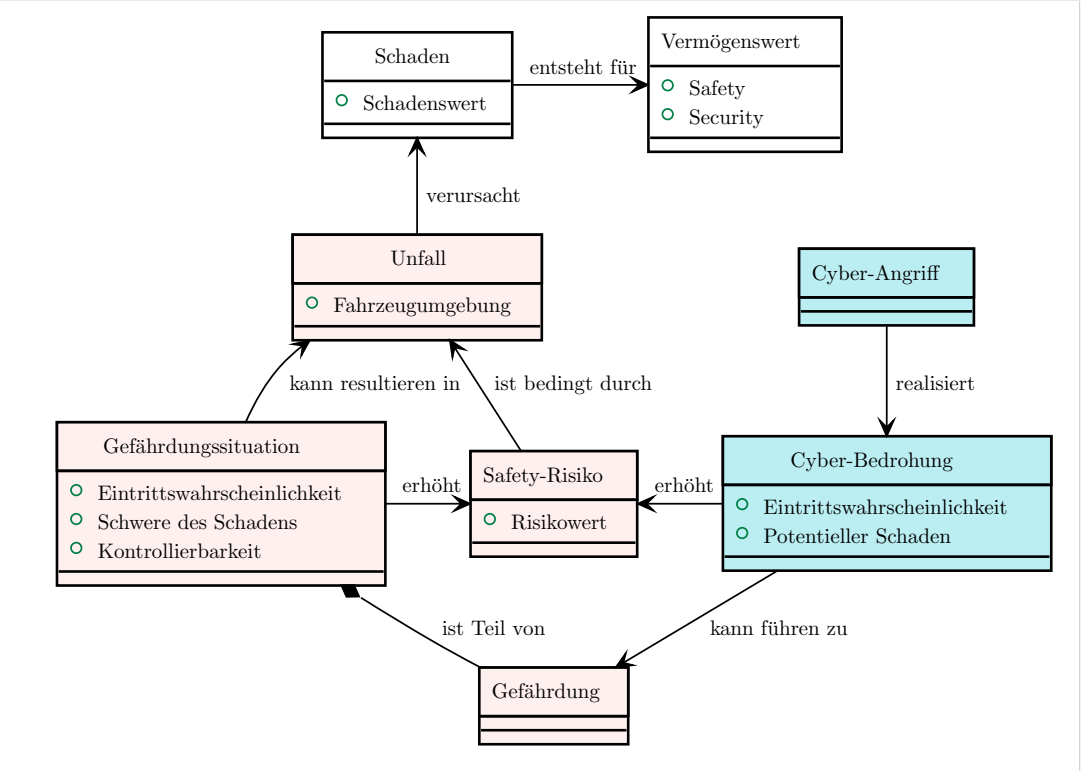


Abbildung 4.2: Metamodell der ersten Detailstufe. Die hellroten Klassen repräsentieren Safety-Artefakte und die hellblauen die Security-Artefakte. Als Notation wird UML 2 verwendet.

Neben der Übernahme des Zusammenhangs zwischen Unfall und Gefährdungssituation aus dem kausalen Modell kann ebenfalls die Kausalkette zwischen einem Unfall und Cyber-Bedrohungen für das Metamodell übernommen werden. Außerdem repräsentiert in Abbildung 4.2 die gleiche Anordnung der Begrifflichkeiten *Gefährdungssituation*, *Safety Risiko* und *Cyber Angriff*, ähnliche Eigenschaften. So besitzen die drei Klassen die Gemeinsamkeit eines probabilistischen Charakters, genauer gesagt die Anheftung des *Risikowertes*. Als Letztes repräsentiert die Klasse *Gefährdung* in Abbildung 4.2 den kausalen Zusammenhang zwischen *Gefährdungssituation* und *Cyber-Bedrohung*. So gehört zur *Gefährdungssituation* stets eine *Gefährdung*, die durch eine *Cyber-Bedrohung* ausgelöst werden kann. Ebenso besteht eine Beziehung zwischen der *Safety-Maßnahme*, um eine Gefährdung abzuschwächen und der *Security-Maßnahme*, die zur Abschwä-

chung von Cyber-Schwachstellen dient. Beide Maßnahmenarten können sich gegenseitig beeinflussen, indem sie im Konflikt zueinander stehen oder sich gegenseitig ergänzen. Eine ergänzende Situation ist beispielsweise eine Plausibilisierung eines Sensorwertes, welche die böswillige Manipulation des Wertes aufdecken und abschwächen kann. Im Konflikt können unter anderem eine Verschlüsselung und eine Echtzeitanforderung stehen, die durch den Aufwand für die Ver- und Entschlüsselung nicht mehr eingehalten werden können.

Wird nun die erste Abstraktionsebene des Metamodells mit weiteren Artefakten und Beziehungen angereichert, ergibt sich mit Abbildung 4.3 die zweite Abstraktionsebene des Metamodells. Hierbei ist das Metamodell aufseiten der Safety mit der *Betriebssituation* und dem *Ausfall* erweitert. Die Beziehungen beider Erweiterungen entsprechen der Komposition aus der UML-Notation, was für die *Gefährdungssituation* bedeutet, dass sie nicht ohne eine *Gefährdung* und *Betriebssituation* existieren kann [168]. Ähnlich gilt dies für die Beziehung zwischen der *Gefährdung* und dem *Ausfall*. Letzteres entspricht der Ausführung eines Defekts, der zu einem Ausfall führt und Inkonsistenzen im spezifizierten Verhalten auslösen kann [68]. Ein *Defekt* entspricht somit einer Safety-Schwäche im E/E-System, die zu einem Ausfall führen kann. Die Klasse *Betriebssituation* beschreibt im Kontext der ISO 26262 [99], welche Fahrsituationen das Fahrzeug erleben wird. Ein Beispiel hierfür ist die Autobahnfahrt.

Bezogen auf das Studienfeld Security entspricht die Klasse *Angreifer* der ersten Erweiterung des zuvor beschriebenen Metamodells in Abbildung 4.2. So verfolgt ein Angreifer bei einem Cyber-Angriff eine bestimmte Motivation und benötigt neben Fähigkeiten und Ressourcen auch Zugriff auf das Zielsystem. Der Angreifer zielt durch seinen Cyber-Angriff stets auf die IT-Vermögenswerte ab. So kann in einem Fall das Ausschleusen von Geheimnissen (Vertraulichkeit) im Fokus stehen und in einem anderen Fall die Nicht-Verfügbarkeit eines Systems. Der *IT-Vermögenswert* ist außerdem eine Spezialisierung der Oberklasse *Vermögenswert* und beschreibt ausschließlich die informationstechnischen Vermögenswerte wie Integrität, Vertraulichkeit, Verfügbarkeit und Authentizität. Um diese zu verletzen, verwendet der Angreifer bei seinem Angriff einen *Exploit*. Dieser nutzt eine Cyber-Schwäche im System aus. Im Falle, das zu einer Cyber-Schwäche ein *Exploit* existiert, wird von einer Schwachstelle anstelle einer Cyber-Schwäche gesprochen.

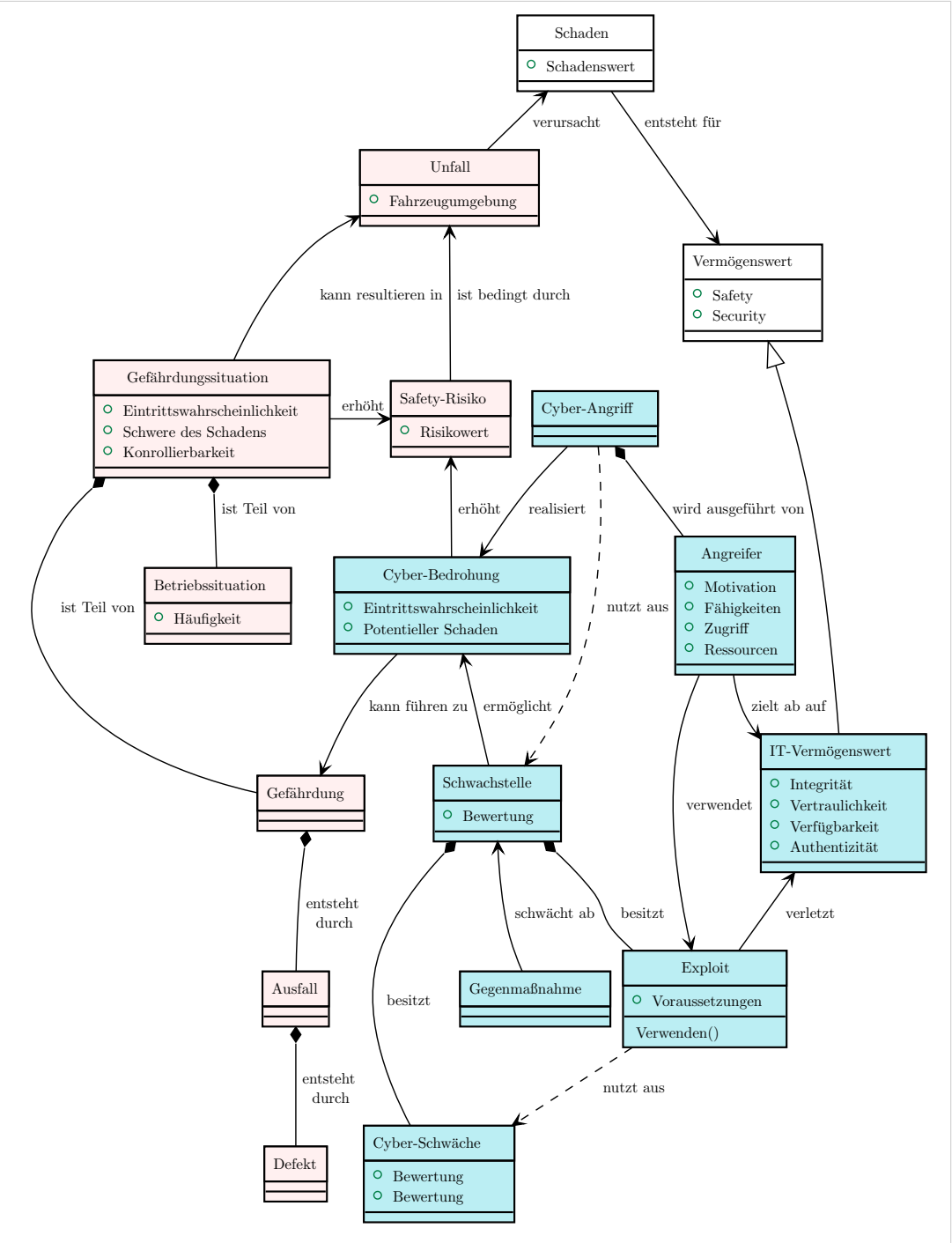


Abbildung 4.3: Metamodell der zweiten Detailstufe, das auf Abbildung 4.2 aufbaut. Die hellroten Elemente repräsentieren Safety-Artefakte, die hellblauen die Security-Artefakte. Als Notation wird auch hier UML 2 verwendet.

Unter Einbezug fahrzeugbezogener Artefakte wie der E/E-Architektur, mit den darin enthaltenen CPS, ergibt sich mit Abbildung 4.4 die letzte und höchste Detailstufe des Metamodells. Der obere Teil zeigt dabei das Metamodell der zweiten Stufe, die untere Hälfte hingegen die Erweiterung durch fahrzeugbezogene Elemente und deren Beziehungen zu den bekannten Safety- und Security-Artefakten. Beginnend auf der

Vernetzungsebene ist dies die E/E-Architektur, die aus den CPS und den Kommunikationsverbindungen zwischen diesen besteht. Mit der Annahme, dass ein CPS Sensoren und Aktoren zum Erfassen und Manipulieren der Umwelt einsetzt, ergeben sich die in Abbildung 4.4 gezeigten Assoziationen zwischen *CPS*, *Sensor* und *Aktuator*. Die E/E-Architektur kann außerdem Mensch-Maschine-Schnittstellen (MMS) aufweisen. Ein Beispiel hierfür ist der OBD-Port, der eine Kommunikation von außen mit dem fahrzeug-internen Netzwerk ermöglicht. Basierend auf dem Konzept, dass MMS die Kommunikation von außen mit dem fahrzeug-internen Netzwerk ermöglichen, fallen ebenfalls die Schnittstellen Compact Disc (CD), Digital Versatile Disc (DVD) USB, etc. in diese Menge. Eine MMS kann dabei unterschiedlichste Verbindungstypen nutzen, um mit dem fahrzeug-internen Netzwerk zu kommunizieren. Hierbei kann grundsätzlich zwischen kabelgebundenen und kabellosen Verbindungen unterschieden werden [1, Seite 12]. Diese können ebenfalls von den CPS genutzt werden, um Informationen mit anderen Elementen in der E/E-Architektur auszutauschen. CPS sind neben Softwarefunktionen zum Steuern von Aktoren und Lesen von Sensoren zusätzlich mit Funktionen ausgestattet, die eine Diagnose der Aktoren, Sensoren, sowie der Systeme selbst (ECUs) ermöglichen. Hinsichtlich Cyber-Security ist dies von besonderer Relevanz, da Diagnosefunktionen in Aktuatoren das gezielte Ansteuern dieser ermöglichen. Hiermit kann bei böswilliger Ausführung der Diagnosefunktion eine Gefährdungssituation entstehen. Voraussetzung hierfür ist, dass die Diagnosefunktion eine Cyber-Schwäche aufweist, die durch einen Exploit ausgenutzt werden kann. Ein Beispiel hierfür ist die im Rahmen dieser Arbeit aufgedeckte Schwachstelle in einem Airbag-System [7].

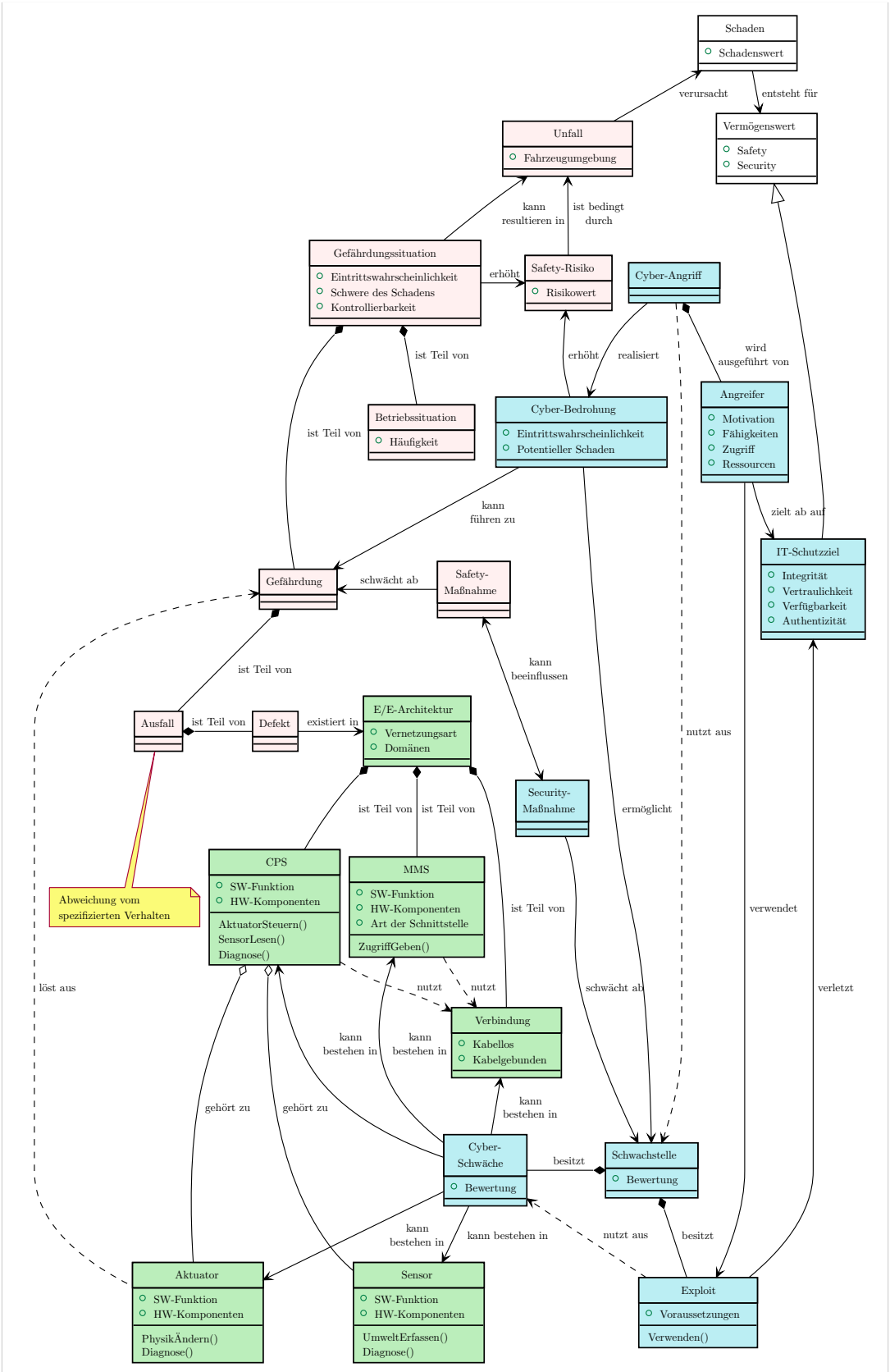


Abbildung 4.4: Metamodell der dritten Detailstufe. Die hellroten Elemente repräsentieren Safety-Artefakte, hellblaue Security-Artefakte und die hellgrünen Artefakte die fahrzeugbezogenen Elemente. Als Notation wird UML 2 verwendet.

Neben dem Ausnutzen einer Cyber-Schwäche in Diagnosefunktionen von Sensoren und Aktoren kann ein Angreifer auch Schwächen in einem Kommunikationskanal ausnutzen (Klasse *Verbindung* in Abbildung 4.4). Für die Kommunikationsverbindung CAN wäre das beispielsweise der grundsätzlich fehlende Schutz vor Denial-of-Service Angriffen, was ein Verletzen der Verfügbarkeit bedeuten kann. Dies gilt ebenso für kabellose Verbindungen, wie einem Wireless Local Area Network (WLAN), das von einem Fahrzeug ausgestrahlt werden und den Zugriff auf Daten gewähren kann.

Als besonders kritisch ist das erfolgreiche Ausnutzen einer Schwachstelle in einem Aktuator zu sehen. So kann ausschließlich ein Aktuator die Physik des Fahrzeugs explizit ändern und ist damit das einzige Element in einer E/E-Architektur, das direkt die Safety beeinflussen kann. Alle anderen Elemente können dies nur indirekt. Damit besteht ein direkter Zusammenhang zwischen einer Gefährdung und dem Fehlverhalten eines Aktuators, was in Abbildung 4.4 mit der gerichteten Abhängigkeit zwischen den Klassen *Gefährdung* und *Aktuator* dargestellt ist.

Im Umkehrschluss kann gefolgert werden, dass eine direkte oder indirekte Manipulation eines Aktuators zu einer Gefährdung führen kann. Die direkte Manipulation beschreibt das physische Manipulieren des Aktuators oder das Verändern seiner Softwarekomponenten. Die indirekte Manipulation hingegen bezieht sich auf das Verändern der Eingangsinformationen des Aktuators. Ermöglicht wird dies durch Cyber-Schwächen in den Aktuatoren selbst, in Kommunikationskanälen oder in CPS, welche die Aktuatoren ansteuern. Hinsichtlich einer kombinierten Bedrohungsanalyse-Methode sollten, daher – neben den Gefährdungen – die Aktuatoren und deren Informationsflüsse erfasst werden. Anschließend müssen diese auf mögliche Cyber-Schwächen überprüft werden. Wie dies anhand eines strukturierten Vorgehens ermöglicht werden kann, zeigt der nächste Abschnitt, der die entwickelte SGM vorstellt.

4.2 Konzeption der Methodik

Bereits in Kapitel 1 wurde dargelegt, dass der Fokus dieser Arbeit auf safety-kritische Bedrohungen abzielt. Aufgrund dessen, zeigen die nachfolgenden Ausführungen den Teil einer Bedrohungsanalyse, der sich auf den Vermögenswert Safety bezieht. Cyber-Bedrohungen die sich auf Vermögenswerte wie Reputation, Finanzen, geistiges Eigentum, etc. beziehen, müssen durch ein weiteres Vorgehen identifiziert werden (Abbildung 4.5).

Die vorgestellte SGM soll als ein Teil einer übergeordneten Bedrohungsanalyse eingesetzt werden. So zeigt Abbildung 4.5, das die SGM an die Gefährdungsanalyse angeknüpft und von Safety-Ingenieuren durchgeführt werden soll. Diese übergeben die identifizierten – safety-kritischen – Cyber-Bedrohungen an die Security-Ingenieure, welche die safety-kritischen Bedrohungen – beispielsweise mit Angriffsbäumen – tiefer

gehend analysieren und die Bedrohungsanalyse für weitere Vermögenswerte durchführen.

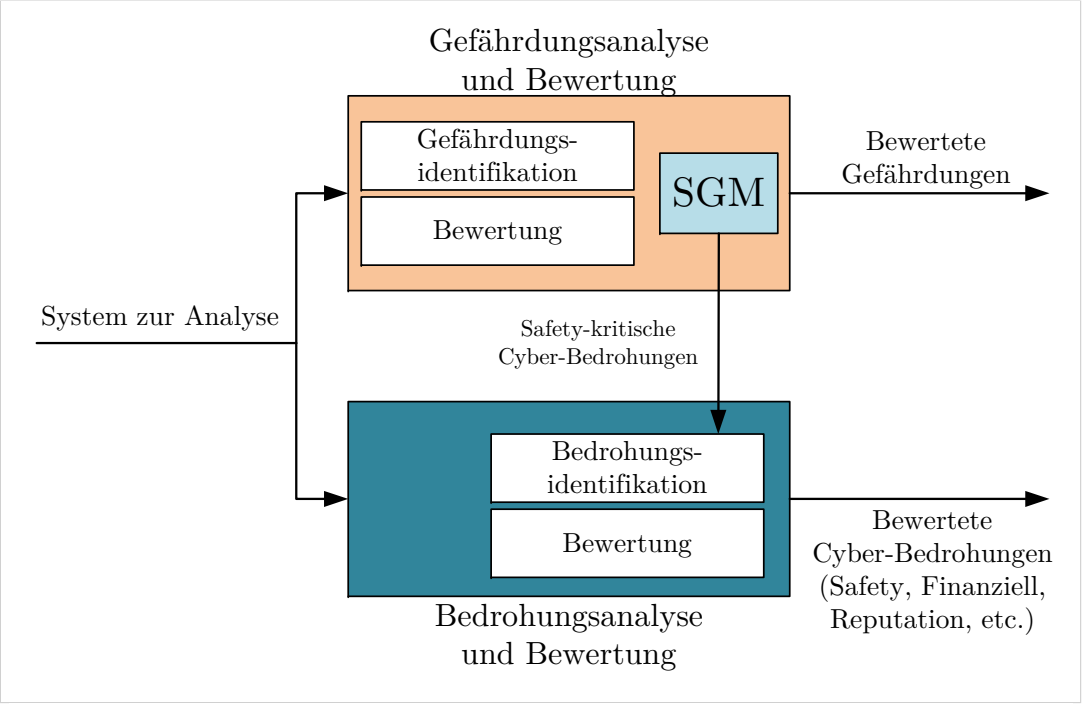


Abbildung 4.5: Einordnung der SGM in den Prozess zur Gefährdungs- und Bedrohungsanalyse. Die SGM ist ein nachgelagerter Teil der Gefährdungsanalyse und übergibt safety-kritische Cyber-Bedrohungen an die Bedrohungsanalyse, bei der Security-Ingenieure Bedrohungen für weitere Vermögenswerte (Finanzen, Reputation, etc.) analysieren.

Hiermit werden Security-Ingenieure – für den Vermögenswert Safety – durch Safety-Ingenieure unterstützt. Den Safety-Ingenieure kommt damit eine unterstützender Faktor zu, sie ersetzen die Security-Ingenieure bei der Bedrohungsanalyse allerdings nicht. Für die Bedrohungsanalyse kann grundsätzlich eine Methodik frei ausgewählt werden.

Für die Konzeption der Methodik werden die Beziehungen im kausalen Modell sowie im Metamodell und dem darin identifizierten Artefakten verwendet. Konkret lässt sich aus dem kausalen Modell ableiten, dass eine Safety-Analyse vorangegangen sein sollte. Sind Gefährdungen und Gefährdungssituationen anhand einer Safety-Analyse identifiziert, so können jene Cyber-Bedrohungen betrachtet werden, welche die Safety verletzen, was Anforderung 1 erfüllt und der Markierung ③ im kausalen Modell entspricht (Abbildung 4.1). Aufgrund der Situation, dass in der Gefährdungsanalyse die zu den Gefährdungen gehörenden Ausfälle identifiziert werden, ist es ebenfalls sinnvoll diese für die Bedrohungsanalyse wiederzuverwenden. Letzteres ermöglicht es, die Cyber-Bedrohungen detaillierter zu formulieren und erleichtert das Identifizieren von potenziellen Schwachstellen in einem CPS. Somit werden alle Safety-Probleme an die Security-Analyse übergeben, was ein großer Vorteil der Methode ist. Gesamtheit-

lich ergibt sich als Reihenfolge für die Gefährdungs- und Bedrohungsanalyse der in Abbildung 4.6 dargestellte Verlauf.

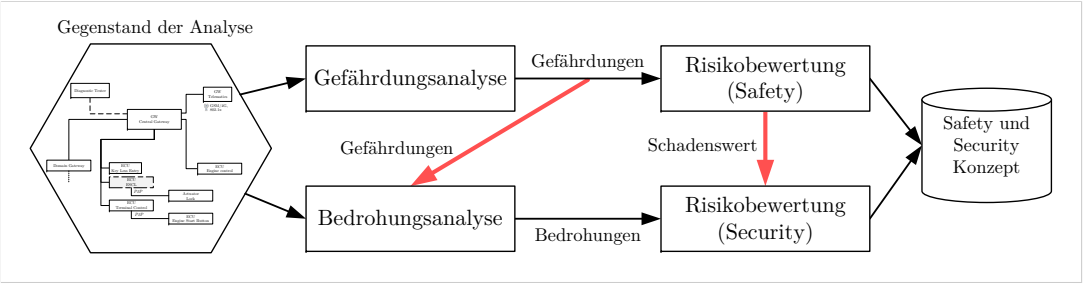


Abbildung 4.6: Kombination der Gefährdungs- und Bedrohungsanalyse um Safety-Artefakte in der Bedrohungsanalyse weiterverwenden zu können. Die Darstellung entspricht der präsentierten Methodenreihenfolge c) aus Abbildung 3.1 mit der Erweiterung zum Austausch des Artefaktes *Schadenswert* aus der Safety-Analyse für die Security-Risikobewertung und Gefährdungen für die Bedrohungsanalyse. Der Austausch von Safety-Artefakten ist mit den roten Pfeilen hervorgehoben.

Die gezeigte Reihenfolge entspricht der Kombinationsmöglichkeit c) aus Abbildung 3.1 und beschreibt das Vorangehen der Gefährdungsanalyse mit einem Weiterleiten der Gefährdungen an die Bedrohungsanalyse. Aus dem Metamodell lässt sich außerdem entnehmen, dass Gefährdungen durch Ausfälle entstehen können, die durch Defekte in E/E-Architekturelementen ausgelöst werden. Neben den Defekten können allerdings auch Cyber-Schwächen zum Ausfall eines Elements und damit zu einer Gefährdung führen. Ein Beispiel für solch eine Safety- und Security-Wirkkette zeigt Abbildung 4.7 anhand einer ungewollten Airbag-Zündung durch einen Angreifer.

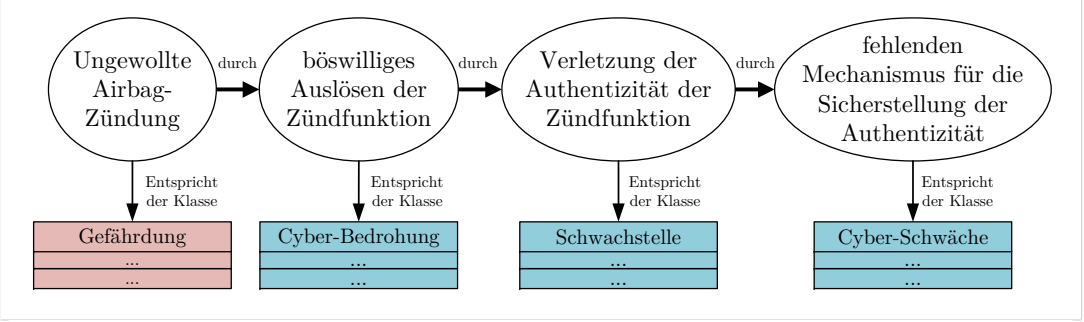


Abbildung 4.7: Safety- und Security-Wirkkette für ein ungewolltes Auslösen einer Airbag-Zündung durch einen Cyber-Angriff. Im oberen Teil des Bildes ist die Wirkkette für die Airbag-Zündung gezeigt und darunter die Zuordnung der Klassen aus dem Metamodell (Abbildung 4.4). Rote Klassen sind der Safety- und blaue Klassen der Security-Domäne zugeordnet. Hiermit ist ersichtlich, dass ausgehend von der Safety-Gefährdung *Ungewollte Airbag-Zündung*, sukzessive die relevanten Security-Artefakte abgeleitet werden können.

Die Wirkkette in Abbildung 4.7 zeigt, dass die Safety-Gefährdung *Ungewollte Airbag-Zündung* durch die Cyber-Bedrohung *böswilliges Auslösen der Zündfunktion* erreicht werden kann. Letzteres ist möglich, da eine Schwachstelle in einem E/E-Architekturelement besteht, welche die *Verletzung der Authentizität der*

Zündfunktion erlaubt. Diese Schwachstelle existiert, da das entsprechende E/E-Architekturelement einen *fehlenden Mechanismus für die Sicherstellung der Authentizität* aufweist. Letzteres ist durch eine Cyber-Schwäche repräsentiert, die in den E/E-Architekturelementen des Analysegegenstandes existieren können. Daher ist es sinnvoll jene E/E-Architekturelemente aus der Gefährdungsanalyse zu übernehmen, denen ein Ausfall zugeordnet wurde und diesen in der Bedrohungsanalyse zu betrachten. Sie können als das Ziel des Angreifers interpretiert werden und dienen als weiterer Ausgangspunkt für die Schwachstellenanalyse (Abbildung 4.7). An dieser Stelle kann mittels der Beziehung zwischen Gefährdung und Aktuator die Bedrohungsanalyse weiter fokussiert werden, was der zweiten Anforderung gerecht wird.

Um Anforderung 3 gerecht zu werden, ist ein Blick auf etablierte Safety-Analysetechniken sinnvoll. Da in dieser Arbeit aus bestehenden Safety-Gefährdungen Cyber-Bedrohungen abgeleitet werden sollen, handelt es sich nach den Ausführungen von Ercison [62, Seiten 48–50] um ein induktives Vorgehen (siehe Abschnitt 2.4). Um die möglichen Ansätze weiter einzugrenzen, kann die Anforderung herangezogen werden, dass die Methode in einer frühen Entwicklungsphase Anwendung finden kann. Da zu diesem Zeitpunkt wenige Informationen zur Verfügung stehen, sind die erfahrungsorientierten Ansätze im Fokus, die überwiegend auf Brainstorming-Techniken aufbauen. Eine etablierte Technik ist hierbei HAZOP [111, 119], die

- in einer frühen Entwicklungsphase angewendet werden kann,
- sich bereits als geeignet für Safety- und Security-Analysen gezeigt hat,
- den Analysten strukturiert durch die Brainstorming-Phase leitet.

Neben dem umfangreichen Einsatz in Safety-Analysen, wurde HAZOP bereits in verschiedenen Security-Anwendungen erprobt. So wurde die Methode beispielsweise von der Firma Intel eingesetzt, um Sicherheitsziele, Bedrohungen und Testfälle für Penetrationstests zu identifizieren [54]. HAZOP ist nicht auf ein bestimmtes Produkt limitiert und kann auf diverse Architekturen angewendet werden. So stellten die Forscher Burzin et al. [54] fest, dass sich HAZOP sehr gut für die Bedrohungsmodellierung eignet. HAZOP basiert auf der Kombination von Anwendungsfällen und Leitwörtern, die den Analysten durch die Analysephase leiten [62, Seite 365].

Hinsichtlich HAZOP-Leitwörtern präsentierten Winther et al. [209] eine Menge von Wörtern, die zur Security-Analyse von Informations- und Kommunikationssystemen dienen. Hierzu schlugen die Forscher Leitwörter und Attribute für die Analyse vor, die der Negation der CIA-Triad [23, Seite 240] entsprechen. Mit diesen waren die Forscher in der Lage, Cyber-Bedrohungen in unterschiedlichen Projekten aufzudecken [209]. Ihre vorgeschlagenen Leitwörter konzentrierten sich allerdings nicht auf das Auffinden safety-relevanter Cyber-Bedrohungen, sondern alleinig auf Security-Bedrohungen.

4.3 Die Security Guide-word Methode (SGM)

Basierend auf dem Fundament, das in den vorherigen Abschnitten gelegt wurde, wird in diesem Paragrafen die entwickelte Security Guidewords Method (SGM) vorgestellt. Die kombinierte Cyber-Bedrohungsanalyse ermöglicht das Auffinden von Cyber-Bedrohungen, die Gefährdungen auslösen können (Abbildung 4.8). Hierzu werden die in der Safety-Analyse identifizierten Gefährdungen an die Bedrohungsanalyse übergeben. Basierend auf diesen Gefährdungen, einer textuellen Vorlage und einem security-spezifischen Set an Leitwörtern, werden Safety-Ingenieure in die Lage versetzt, safety-kritische Cyber-Bedrohungen aufzudecken. Abbildung 4.8 zeigt die sieben Schritte der Methodik sowie deren Ein- und Ausgangsartefakte, die auf den Forschungsergebnissen [1, 2] basieren.

Methodenschritt 1 stellt mit der Definition des Analyseumfangs den initialen Schritt der Methode dar und wird ebenfalls von der ISO 26262 [99] gefordert. Da die Analyse in einer frühen Entwicklungsphase durchgeführt wird, bietet sich ein Kontextdiagramm als Notation für den Analyseumfang an. Dieses entspricht einem Informationsflussdiagramm und zeigt die relevanten Entitäten sowie deren Informationsflüsse. Bezogen auf den hier betreffenden Kontext enthält das Diagramm die Komponenten der E/E-Architektur, was den hellgrünen Elementen im Metamodell entspricht (Abbildung 4.4). Darüber hinaus kann das Diagramm, abhängig von der Festlegung des Analyseumfangs, umliegende Netzwerkdomänen oder externe Entitäten wie beispielsweise den Fahrzeugführer enthalten, die zum Zeitpunkt der Analyse bekannt sind (siehe Beispiel in Abschnitt 4.3.1).

Methodenschritt 2 entspricht der Auswahl und Instanziierung der Leitwörter für die Safety-Analyse. Hierzu kann das Standard HAZOP-Set verwendet werden [46]. Dieses enthält Leitwörter wie *kein, ungewollt, zu früh, zu spät, mehr, wenig, invertiert, zeitweise auftretend oder später*. Das Set kann außerdem durch domänenspezifische Leitwörter erweitert werden, um die Analysten auf typische Ausfälle und Probleme zu lenken. Unter Anwendung des Kontextdiagramms, der funktionalen Beschreibung der Elemente im Analyseumfang und der Leitwörter lassen sich Ausfälle und die darauf bezogenen Gefährdungen identifizieren [62, 143]. Für ein HAZOP-Arbeitsblatt sei auf Abschnitt 2.4 verwiesen.

Methodenschritt 3 beschreibt die Auflistung von Betriebssituationen, die der Analysegegenstand erfahren wird. Da die Safety-Analyse im Einklang mit der ISO 26262 [99] steht, werden die Betriebssituationen auf Fahrzeugebene festgelegt. Dies ermöglicht – im späteren Analyseverlauf – das Bestimmen des Schadenwertes. Die Betriebssituationen entsprechen somit den Fahrsituationen des Fahrzeugs und können aus vorangegangenen Analysen wiederverwendet werden. Letzteres ist gegeben, da sich die Fahrsituationen in der Regel nicht ändern. Zur Auswahl der relevanten Betriebssituationen können die Anforderungen des Analysegegenstands und das Kontextdiagramm verwendet werden.

Beide Informationsquellen zeigen den operativen Kontext des Gegenstands, was die gezielte Auswahl der adäquaten Betriebssituationen ermöglicht.

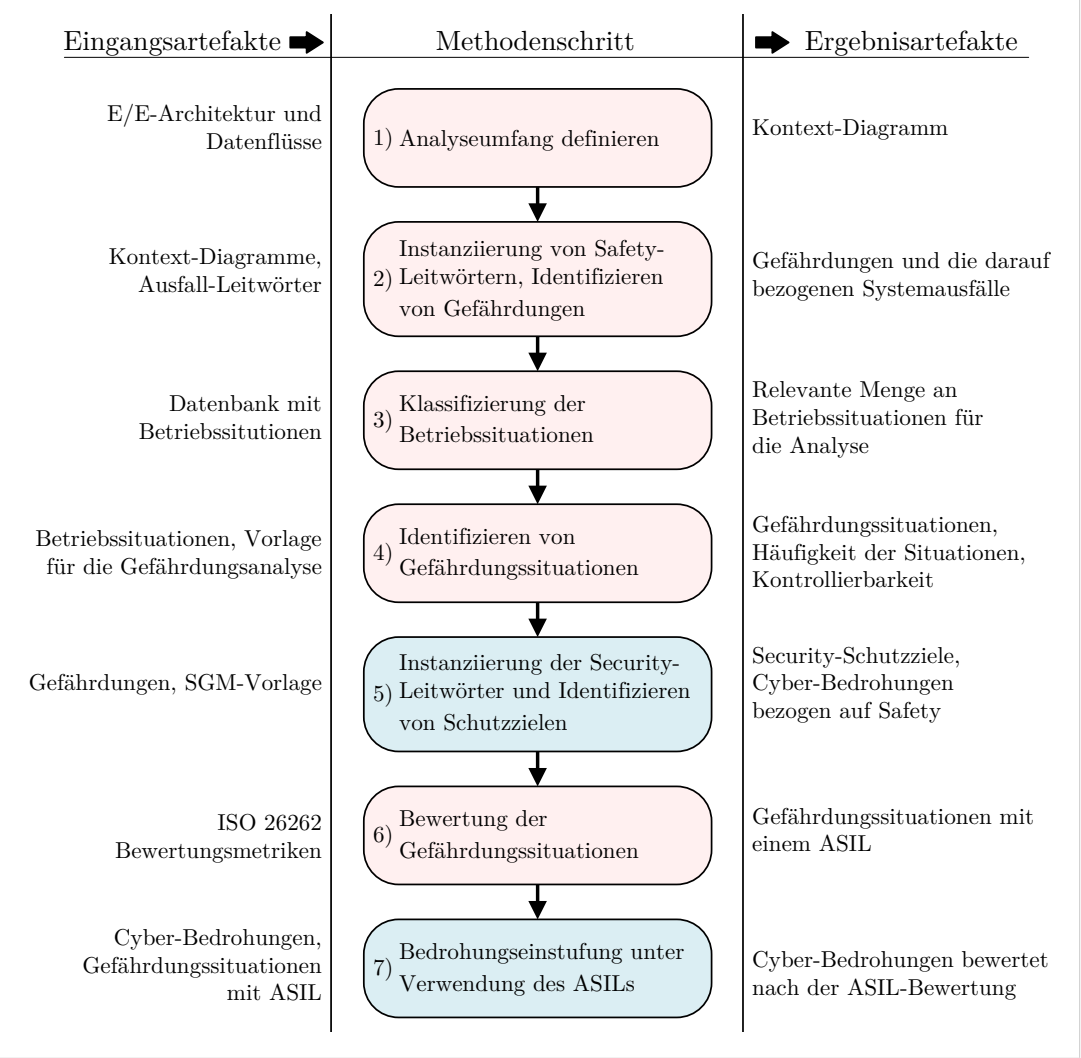


Abbildung 4.8: Die sieben Methodenschritte zur kombinierten Safety- und Security-Analyse. Der Ablauf zeigt wie die Safety-Analyse (hellrote Elemente) mit den SGM-Erweiterungen (hellblaue Elemente) erweitert ist. Bei dieser Abbildung ist die Annahme getroffen, dass für die Safety-Analyse HAZOP als Analyse-Technik eingesetzt wird. Links sind jeweils die Eingangsartefakte gezeigt die in den jeweiligen Methodenschritt einfließen. Rechts, jene Ergebnisartefakte, die in einem Methodenschritt erzeugt wurden.

Methodenschritt 4 repräsentiert die Identifizierung der Gefährdungssituationen. Jede Gefährdung wird mit denen aus dem vorherigen Schritt bestimmten Betriebssituationen kombiniert und damit potenzielle Gefährdungssituationen aufgedeckt. Diese werden auf Fahrzeugebene beschrieben und die resultierenden Konsequenzen aufgelistet. An dieser Stelle soll hervorgehoben werden, dass alleinig anhand der Gefährdungssituationen der Schadenswert, die Häufigkeit der Betriebssituation und die Kontrollierbarkeit abgeleitet werden können.

Methodenschritt 5 entspricht dem ersten Einschub der SGM in die Safety-Analyse. Hierzu wird dem Analysten die textuelle SGM-Vorlage zur Identifikation von Schutzzielen und Bedrohungen bereitgestellt (Tabelle 4.2). Dazu werden die Gefährdungen – aus der Safety-Analyse – übernommen und mit den automotive-spezifischen Security-Leitwörtern kombiniert. Diese basieren auf den Arbeiten von Winther et al. [209] und sind anhand bekannter Analysen [5, 10] von Security-Angriffen auf Fahrzeuge abgeleitet. Sie sind in Tabelle 4.1 dargestellt und repräsentieren die böswilligen Absichten eines Angreifers.

Tabelle 4.1: Security-Leitwörter für die Verwendung der SGM-Vorlage in Tabelle 4.2 sowie die Zuordnung zur STRIDE-Nomenklatur durch die Zwischenallokation auf Confidentiality, Integrity, Availability, Authenticity (CIAA) [121, 150], die sich als Schutzziele darstellen [2, 3].

SGM	CIAA	STRIDE
Auslösen	Authentizität	Spoofing
Einschleusen	Integrität	Tampering
Manipulieren	Integrität	Tampering
Unterbrechen	Verfügbarkeit	Denial-of-service
Verzögern	Verfügbarkeit	Denial-of-service
Löschen	Verfügbarkeit	Denial-of-service
Stoppen	Verfügbarkeit	Denial-of-service
Zurücksetzen	Verfügbarkeit	Denial-of-service
Auslesen	Vertraulichkeit	Information disclosure

Neben der Auflistung der Leitwörter in Tabelle 4.1 entspricht die zweite Spalte der Zuordnung von IT-Vermögenswerten aus dem Metamodell in Abbildung 4.4. Diese repräsentieren die Security-Schutzziele, welche für Security-Analysten verständlich sind und das anschließende Ableiten von High-level Security-Maßnahmen erlauben. Außerdem werden hiermit die Security-Ingenieure befähigt, Safety-Probleme aus Sicht der Security verstehen zu können, was eine Schnittstelle zwischen Safety und Security etabliert. Die weiteren Spalten in Tabelle 4.2 ermöglichen anschließend die strukturierte Sammlung der Bedrohungen.

Die erste Spalte (B-ID) dient zur eindeutigen Erfassung der Bedrohungen durch einen numerischen Bezeichner. Eine Bedrohung bildet sich dabei durch die Inhalte der Felder (1) bis (5) in Tabelle 4.2, wobei die G-ID Spalte den in der Safety-Analyse identifizierten Gefährdungen entspricht und direkt aus dieser übernommen werden kann. Anschließend wird in Feld (2) eines der Security-Leitwörter aus Tabelle 4.1 eingetragen. Darauffolgend wird das Kontextdiagramm aus Methodenschritt 1 herangezogen, um Signale oder Funktionsaufrufe übernehmen zu können, die in Feld (3) eingefügt werden.

Tabelle 4.2: SGM-Vorlage, die der Analyst verwendet, um die Bedrohungen zu identifizieren. Die erste Spalte entspricht einem eindeutigen Bezeichner (ID) für jede Bedrohung. Die zweite verweist auf die jeweilige Gefährdung aus der Safety-Analyse. Alle restlichen Spalten fokussieren den Analysten auf die relevanten Artefakte während der Analyse.

B-ID	G-ID	kann ausgelöst werden durch	Signal oder (Diagnose-) Funktion	für Komponente oder Teilsystem	Eintrittspunkt
1	(1)	(2)	(3)	(4)	(5)
2	...	...	...	...	...

Funktionsaufrufe entsprechen hier dem Aufruf einer Funktion, die auf dem Zielsystem (4) implementiert ist. In Feld (4) wird anschließend diejenige Komponente oder das Teilsystem eingetragen, die das Signal oder den Funktionsaufruf aus Feld (3) verarbeitet. Somit repräsentiert die Spalte, welches Signal kompromittiert werden kann oder welche Funktionalität der Angreifer auslösen könnte. Anschließend wird in Feld (5) der mögliche Ort der Manipulation festgehalten. Wurde in Feld (3) ein Signal ausgewählt, so sollte in Feld (5) die Datenverbindung genannt werden, auf der das Signal übertragen wird. Diese Information lässt sich aus dem Kontextdiagramm ableiten, das nach der obigen Definition den Informationsfluss des Analysegegenstandes beschreibt.

Analog zu HAZOP identifiziert die SGM-Vorlage ausschließlich Bedrohungen, die auf Einzelausfällen beruhen. Bedrohungen die mehrere Ausfälle erfordern, können nicht direkt mit der Vorlage erfasst werden. Die SGM-Vorlage kann allerdings mit Fehlerbäumen [62, Seiten 183–186] erweitert werden, um ebenso Bedrohungen mit mehrfachen Ausfälle zu adressieren. Hierzu wird in der Wurzel des Fehlerbaums die Gefährdung aus der SGM-Vorlage übernommen und in den Knoten, die Ausfälle und deren Abhängigkeiten erfasst (Abbildung 4.9). Für den beispielhaften Baum in Abbildung 4.9 bedeutet dies, dass die Ausfälle A und B zur gleichen Zeit auftreten müssen, um die Gefährdung in der Wurzel auszulösen.

Neben der Erfassung der Abhängigkeiten durch *UND* und *ODER* Bedingungen, können die Fehlerbäume an die Security-Ingeniuere übergeben und in Angriffsbäume transformiert werden. Der Aufwand hierzu ist gering, da beide Baumarten eine gleiche Grundstruktur aufweisen, was eine computergestützte Transformation erlaubt.

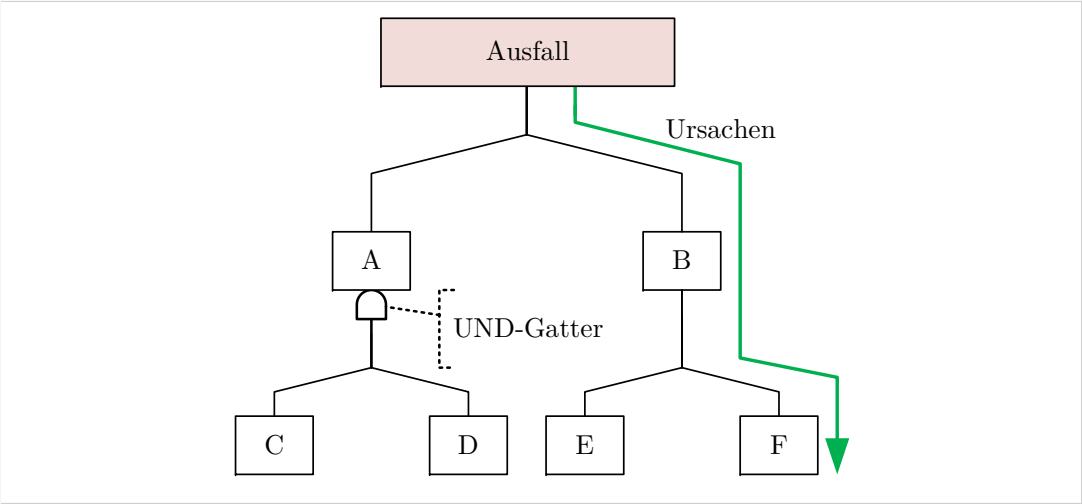


Abbildung 4.9: Optionale Erweiterung der SGM mit Fehlerbäumen zur Analyse von Bedrohungen mit mehrfachen Ausfällen.

Für jeden Knoten (A, B, C und D) in Abbildung 4.9 kann wiederum die SGM-Vorlage mit den Leitwörtern angewendet werden, um die security-bezogenen Kausalfaktoren zu identifizieren, was schlussendlich zu einem Angriffsbaum führen kann. Für eine Erweiterung der SGM mit Angriffsbäumen ist auf Abschnitt 6.1.1 verwiesen.

Methodenschritt 6 wechselt zurück zur Safety-Analyse und führt die Risikobewertung für jede Gefährdungssituation – nach der ISO 26262 – durch. Hierzu müssen der Schadenswert (engl. severity), die Häufigkeit der Betriebssituation (engl. exposure of operational situation) und die Kontrollierbarkeit (engl. controllability) festgelegt werden. Anschließend kann mit diesen und der Zuordnungsmatrix aus dem dritten Teil der ISO 26262 [99] das ASIL bestimmt werden. Die Einordnung beginnt in diesem Fall mit der Klasse QM, was dem geringsten Risiko entspricht. Mit steigendem Risiko folgen die Klassen A,B,C und D. Letztere entspricht dabei der höchsten Risikoklasse. Anschließend müssen für jede Klasse – außer QM – Safety-Ziele abgeleitet werden, die das funktionale Safety-Konzept formen.

Methodenschritt 7 bewertet die Bedrohungen anhand der im vorherigen Schritt festgelegten Schadenswerte. Hierzu werden zu jeder Bedrohungs-ID aus Tabelle 4.2 die jeweilige Gefährdung (G-ID) ausgewählt und alle Gefährdungssituationen ermittelt, die ausgelöst werden können. Aus dieser Menge von Gefährdungssituationen wird jene ausgewählt, die den höchsten ASIL aufweist. Dieser Wert wird anschließend der Bedrohung angeheftet, was eine schnelle Risikowertung anhand des ASIL-Wertes erlaubt. Da bei der Übernahme des ASIL-Wertes allerdings Diskussionspotenzial hinsichtlich der Eintrittswahrscheinlichkeit der Betriebssituation und der Festlegung der Kontrollierbarkeit durch den Fahrer besteht, soll an dieser Stelle auf Kapitel 7 verwiesen werden. Hier wird die erweiterte SGM vorgestellt, die ausschließlich den Schadenswert (Severity) aus der Safety-Analyse übernimmt und diesen mit einer Security-Eintrittswahrscheinlichkeit der jeweiligen Bedrohung kombiniert.

Nachdem die Bewertung der Bedrohungen abgeschlossen ist, endet die SGM-Methode. Letztere ermöglicht allerdings – analog zum Safety-Vorgehen – auch das Ableiten von Security-Anforderungen. Konkret geht es um das funktionale Safety- und Security-Konzept, das an die Gefährdungs- und Bedrohungsanalyse anknüpft. Hinsichtlich des Safety-Konzeptes fordert die ISO 26262, dass für jedes identifizierte Safety-Ziel Anforderungen abgeleitet werden, die in der Gesamtheit das Safety-Konzept formen. Hierzu liefert der Standard ein strukturiertes Vorgehen. Bezogen auf die Erhebung von Security-Anforderungen können Schritt 5 der Methode und die dort identifizierten Artefakte (Tabelle 4.2) einen Beitrag leisten. So können die Schutzziele verwendet werden, um Security-Anforderungen abzuleiten. In Tabelle 4.9 ist eine beispielhafte Ableitung von Security-Anforderungen mit Hilfe von Security-Schutzzielen gegeben. Diesbezüglich verwendet der Security-Ingenieur die beschriebene Bedrohung und transformiert diese in eine Security-Anforderung, indem er beschreibt, wie das Security-Schutzziel eingehalten werden kann. Letzteres unterstützt die Entwicklung von passenden Gegenmaßnahmen.

Nachdem der Ablauf der Bedrohungsanalyse beschrieben ist, soll an dieser Stelle noch einmal die Abgrenzung der SGM-Methodik hervorgehoben werden. So zielt das hier gezeigte Vorgehen ausschließlich auf Bedrohungen ab, welche die Safety verletzen können (Abbildung 4.5). Andere Vermögenswerte wie der Datenschutz werden zum Zeitpunkt dieser Arbeit nicht adressiert. Eine Erweiterung der Leitwörter sowie der Vorlage zur Erfassung von Bedrohungen auf Basis des Datenschutzes ist jedoch gegeben. So bietet beispielsweise das Leitwort *Auslesen* die Charakteristik des ungewollten Enthüllens von Informationen, das wiederum dem Schutzziel Datenschutz zugeordnet werden könnte. Die SGM ersetzt somit keine umfassende Bedrohungsanalyse, die weitere Vermögenswerte betrachtet. Die SGM kann allerdings in eine andere Bedrohungsanalysemethodik integriert werden, um diese zu ergänzen, falls der Vermögenswert *Safety* betrachtet werden soll. Die Voraussetzung ist jedoch, das eine Safety-Analyse durchgeführt wurde und die identifizierten Gefährdungen an die SGM weitergeleitet werden. Dabei zeigt Abbildung 4.10 das grundsätzliche Vorgehen zur Ergänzung einer bestehenden Bedrohungsanalysemethodik durch die SGM. Hierzu werden die Methodenschritte 5+7 der SGM durchgeführt und die safety-relevanten Bedrohungen weitergeleitet. Die Koppelung beider Analysen ist in Abbildung 4.10 beim Pluszeichen dargestellt. An dieser Stelle werden die safety-relevanten Bedrohungen der SGM und die Cyber-Bedrohungen der allgemeinen Bedrohungsanalyse zusammengeführt und anschließend an eine gemeinsame Risikobewertung übergeben. Um Duplikate beim Asset *Safety* zu vermeiden, ist es sinnvoll die SGM vor der allgemeinen Bedrohungsanalyse durchzuführen. Hiermit können den Security-Ingenieuren die safety-kritischen Bedrohungen übergeben werden, sodass diese Bedrohungen nicht ein zweites Mal – in der allgemeinen Analyse – identifiziert werden. Sollten während der allgemeinen Bedrohungsanalyse, weitere safety-kritische Bedrohungen identifiziert werden, können

diese an das Safety-Team übergeben werden. Das Safety-Team bewertet dann den Schaden und die Eintrittswahrscheinlichkeit der Bedrohung, was mit dem gestrichelten Pfeil in Abbildung 4.10 dargestellt ist.

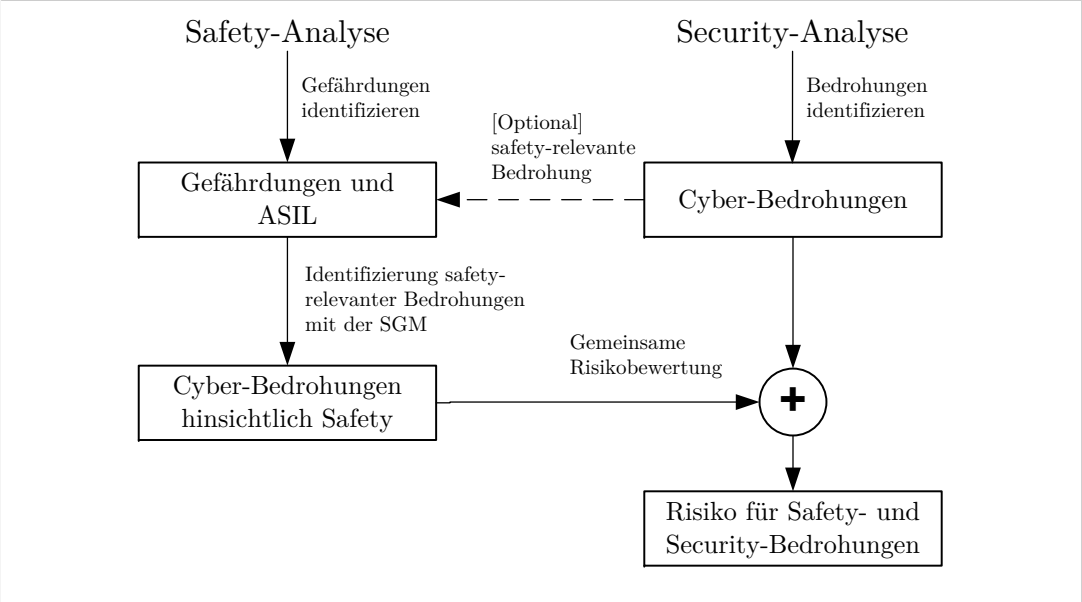


Abbildung 4.10: Konzeptionelles Vorgehen zur Ergänzung einer Bedrohungsanalysemethodik mit der SGM, um safety-relevante Cyber-Bedrohungen betrachten zu können. Die Rechtecke entsprechen den von den Methoden erzeugten Ergebnisartefakten und die Pfeile erklären, wie die Artefakte weitergereicht werden.

Aufgrund der Tatsache, dass eine Risikobewertung den Schaden und die Eintrittswahrscheinlichkeiten jeder Bedrohung bestimmen muss, kann der in der SGM erfasste Schadenswert die Risikoanalyse beschleunigen, wenn dieser übergeben wird. Hiermit kann angenommen werden, dass mit einer Ergänzung durch die SGM eine Unterstützung von bestehenden Bedrohungsanalysemethoden erreicht werden kann.

4.3.1 Beispielhafte Anwendung der SGM

In diesem Abschnitt soll anhand eines realen Beispiels die SGM angewendet werden, um die zuvor beschriebenen Schritte zu verdeutlichen. Hierzu dient das elektronische Lenkradschloss (Electronic Steering Column Lock (ESCL)), das in eine exemplarische E/E-Architektur (siehe Abbildung 4.11) eingebettet ist und die realitätsnahen Kommunikationspfade mit anderen Steuergeräten im Systemverbund aufzeigt. Die in Abbildung 4.11 präsentierte E/E-Architektur ist von unterschiedlichen E/E-Architekturen verschiedener Hersteller abgeleitet und stellt den aktuellen Trend von E/E-Architekturen dar. Darüber hinaus verfolgt diese den Ansatz getrennter Domänen. Dies bedeutet, das die zu implementierenden Steuergeräte – bezogen auf ihre Funktionalität – in eine der vier dargestellten Domänen eingeordnet werden. Steuergeräte, deren Funktionalitäten der Unterhaltung und dem Komfort des Fahrers dienen, sind in der Infotainment-Domäne angesiedelt. Diese Domäne zeigt gemäß Abbildung 4.11 die

Auffälligkeit eines Domänen-Gateways. Dieses koppelt die Infotainment-Domäne von anderen Steuergeräten ab und erlaubt die kontrollierte Weiterleitung von Informationen an andere Steuergeräte außerhalb der eigenen Domäne. Dies bietet beispielsweise Vorteile, falls Verbindungstechniken mit unterschiedlichen Bandbreiten eingesetzt werden und eine der Verbindungstechniken eine geringere Bandbreite als die anderen aufweist. Durch die Trennung der Domänen wird ermöglicht, Verbindungstechniken mit hohen und niedrigen Bandbreiten zu koppeln, indem nur ein geringer Teil der Kommunikation von einer Domäne in die andere weitergeleitet wird. Prinzipiell kann jede Domäne solch ein Gateway besitzen. Zur besseren Veranschaulichung zeigt die E/E-Architektur in Abbildung 4.11 jedoch nur ein Gateway in der Infotainment-Domäne.

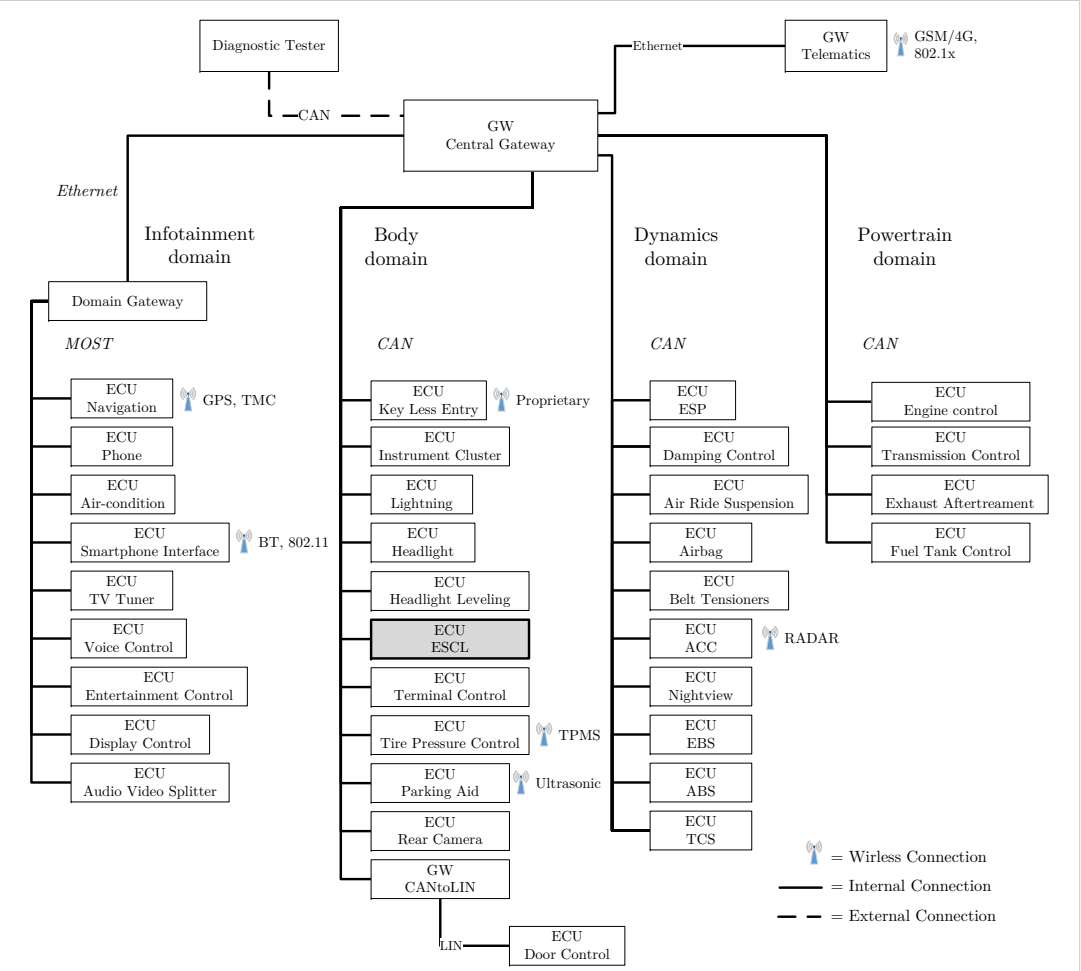


Abbildung 4.11: Beispielhafte E/E-Architektur eines Fahrzeuges. Gezeigt sind vier Domänen-Controller, ein zentrales Gateway (Central Gateway) sowie ein Telematik-Steuergerät (Telematics Gateway) zur Kommunikation mit dem Backend des Herstellers.

Steuergeräte, deren Aktuatoren/Sensoren (wie Abblendlicht, Parksensoren, etc.) sich an der Karosserie des Fahrzeuges befinden, werden in die Body-Domäne eingeordnet. Funktionalitäten, welche die Dynamik eines Fahrzeugs kontrollieren, sind in der Dynamics-Domäne eingebettet. Typisch in dieser Domäne sind die Steuergeräte für

das Electronic Stability Control (ESP), das Anti-lock Braking System (ABS) oder den Airbag. In der Powertrain-Domäne sind ECUs zu finden, die den Motor, das Getriebe und, falls vorhanden, die Abgasanlage kontrollieren. Alle vier Domänen sind über ein gemeinsames Gateway verbunden, das für das Routing von Informationen zwischen den Domänen verantwortlich ist. Neben dem Routing von Inter-Domänen-Kommunikation bindet das Zentrale Gateway in Abbildung 4.11 eine Telematik-Einheit an. Diese stellt die Schnittstelle zur informationstechnischen Infrastruktur des Herstellers dar und kommuniziert mittels Mobilfunktechnik mit dem Backend des Herstellers. Über diese Verbindung werden Mehrwertdienste, wie die Anbindung an das Internet, ermöglicht. Alle Datenströme, die von der Telematik-Einheit nach außen verlaufen, werden über das Backend geleitet, auf das der Fahrzeughersteller grundsätzlich zugreifen kann. Neben der Telematik-Einheit bindet das zentrale Gateway in Abbildung 4.11 mit dem OBD-Port eine Schnittstelle zur Fahrzeug-Diagnose an. Die OBD-Schnittstelle wird in Werkstätten verwendet, um einen Diagnosetester mit dem Fahrzeug zu verbinden und die Fehlerspeicher der Steuergeräte auslesen zu können.

Anwendung der SGM

In **Methodenschritt 1** werden das Kontext-Diagramm und der Analyseumfang definiert. Hierzu werden die zu diesem Zeitpunkt bekannten Informationen des ESCL-Systems gesammelt. Diese sind neben den Funktionalitäten die High-Level E/E-Architektur sowie die bekannten Informationsflüsse zwischen den Komponenten. Letztere sind in Abbildung 4.12 durch ein Brief-Symbol gekennzeichnet und aus den wissenschaftlichen Arbeiten [1, 2] entnommen. Das elektronische Lenkradschloss besteht aus dem *ESCL* Steuergerät und einem Schließzylinder (*Actuator Lock*), der in Abbildung 4.12 mit einer direkten Verbindung (*P2P*) an die ESCL ECU angebunden ist. Durch Ausfahren des Schließzylinders lässt sich die Rotation des Lenkrads blockieren und durch Einfahren des Zylinders freigeben. Die Funktionalitäten zum Sperren und Freigeben des Lenkrades sind alleinig auf der ESCL-ECU verortet und nicht über den Steuergeräteverbund hinweg verteilt. Anhand der indirekten Informationen über den Zustand des Startknopfes (*MotorAN*), die über das Zündsystem (*Terminal Control*) auf dem CAN-Bus zur Verfügung gestellt werden, und der Informationen vom schlüssellosen Schließsystem (*Key Less Entry*) entscheidet die ESCL-ECU, ob der Schließzylinder ein- oder ausgefahren werden muss. Darüber hinaus kommuniziert die ESCL-ECU über das zentrale Gateway (Central Gateway) mit dem Motor-Steuergerät, das in der Antriebsstrang-Domäne eingebunden ist. Dies ist notwendig, um das Starten des Motors zu unterbinden, falls das Lenkrad verriegelt ist. Wird die Verriegelung hingegen – durch einen Cyber-Angriff – während der Fahrt aktiviert, so kann der Fahrer keine Lenkbewegungen mehr ausführen und das Fahrzeug nicht mehr führen.

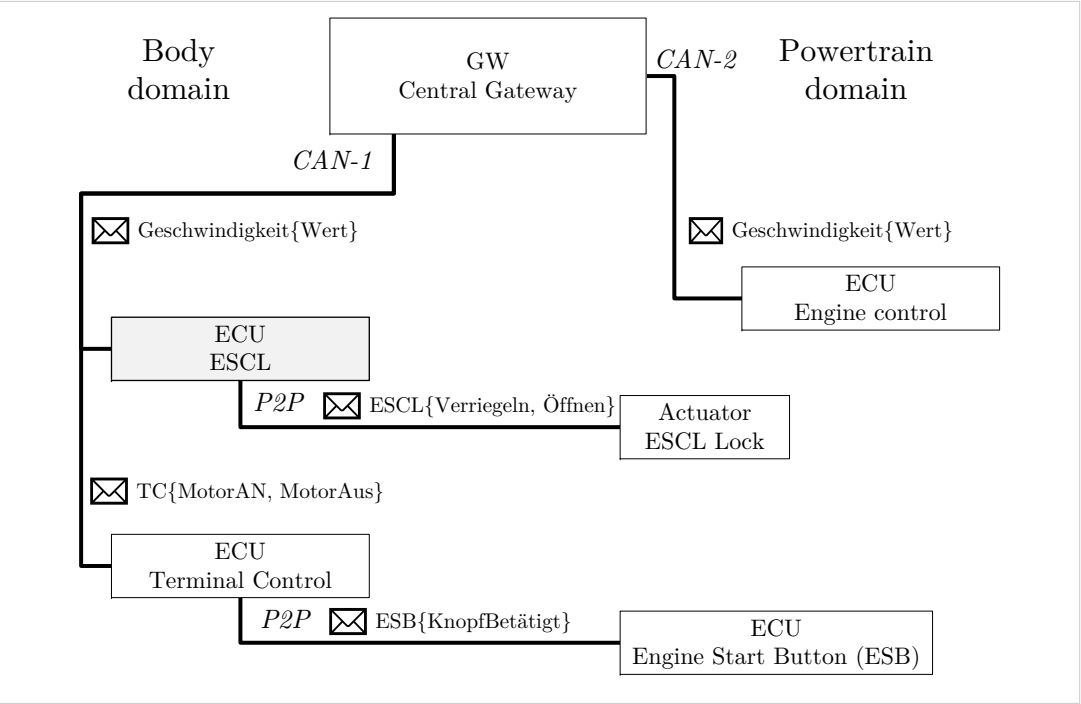


Abbildung 4.12: Kontext-Diagramm des ESCL, das einem Auszug von Abbildung 4.11 entspricht. Es zeigt die relevanten Komponenten des elektronischen Lenkradschlosses (ESCL) sowie den Informationsaustausch zwischen diesen. Informationen sind durch das Brief-Symbol gekennzeichnet und beschreiben den Namen sowie den Wertebereich der Information in den geschweiften Klammern. Für die Information *Geschwindigkeit* ist das beispielsweise der Geschwindigkeitswert, für die Information *ESB* (*Engine Start Button*) ob der Startknopf durch den Fahrer gedrückt wurde, was einer booleschen Variable entspricht.

Neben den beiden CAN-Bussen, zeigt Abbildung 4.12 zwei direkt verdrahtete Verbindungen Peer-to-Peer (P2P). Diese werden für die Kommunikation zwischen der ESCL-ECU und dem Schließzylinder (ESCL Lock) eingesetzt und übermitteln die Befehle zum Verriegeln oder Öffnen des Lenkradschlosses. Die generelle Funktionalität des Systems kann aus den Anforderungen abgeleitet werden, die in Tabelle 4.3 aufgelistet sind. Aus Gründen der Darstellung sind nur die zwei wichtigsten Anforderungen aufgelistet.

Tabelle 4.3: Funktionale Anforderungen an das ESCL-System.

R-ID	Beschreibung der Anforderung
1	Die Lenksäule muss verriegelt sein, wenn der Fahrzeugführer das Fahrzeug abstellt.
2	Die Lenksäule muss entriegelt sein, wenn der Fahrer das Fahrzeug bewegen möchte.
...	...

Mittels der Ableitung von Funktionalitäten aus den Anforderungen in Tabelle 4.3 können an dieser Stelle können bereits erste Gefährdungen identifiziert werden. So

stellt eine verriegelte Lenksäule während der Fahrt eine Gefährdung dar, da in diesem Falle der Fahrer das Lenkrad nicht mehr rotieren kann.

Das strukturierte Vorgehen zur Identifikation der Gefährdung erfordert im ersten Schritt das Instanziieren der Leitwörter, die als generische Ausfälle gesehen werden können (**Methodenschritt 2** der SGM). Anschließend werden in **Methodenschritt 3** mit den ausgewählten Leitwörtern und den Funktionalitäten des ESCL die Gefährdungen identifiziert. Aus Gründen der Darstellung werden an dieser Stelle nur zwei primäre Funktionalitäten des ESCL betrachtet: das Verriegeln und Entriegeln der Lenksäule, die in Kombination mit den HAZOP-Leitwörtern die Gefährdungen (G-ID) bilden (Tabelle 4.4).

Tabelle 4.4: Auszug aus der mit HAZOP aufgedeckten Gefährdung für das ESCL.

G-ID	Beschreibung des Ausfalles
1	Kein Verriegeln der Lenksäule
2	Kein Entriegeln der Lenksäule
3	Ungewolltes Verriegeln der Lenksäule
4	Ungewolltes Entriegeln der Lenksäule
5	Zu frühes Verriegeln der Lenksäule
6	Zu frühes Entriegeln der Lenksäule
7	Zu spätes Verriegeln der Lenksäule
8	Zu spätes Entriegeln der Lenksäule
...	...

Durch die Anwendung der Leitwörter *kein* und *ungewollt*, ergeben sich somit die Gefährdungen (G-ID) 1 bis 4 und mit den Leitwörtern *zu früh* und *zu spät* die Gefährdungen 5 bis 8. Die Leitwörter *mehr* und *weniger* werden nicht betrachtet, da der Wertebereich der Funktion Verriegeln nur zwei Zustände aufweist. Die übrigen Leitwörter wurden aus Darstellungsgründen ausgelassen, da das grundsätzliche Vorgehen zur Identifizierung der Ausfälle mit den Zeilen 1 bis 8 nachvollziehbar ist. Anschließend werden in **Methodenschritt 4** die aufgedeckten Gefährdungen mit den Betriebssituationen des Fahrzeugs kombiniert und mögliche Gefährdungssituationen bestimmt. Relevante Betriebssituationen können dabei aus historischen Analysen oder Wissen entnommen werden, da sich diese nur geringfügig ändern. Für dieses Beispiel sollen die folgenden vier Betriebssituationen dienen.

Tabelle 4.5: Vier ausgewählte Betriebssituationen, die während der Lebensdauer eines Fahrzeuges auftreten können.

O-ID	Beschreibung der Betriebssituation
1	Fahren auf der Autobahn
2	Schnelles Abbiegen
3	Fahrzeug ist geparkt
4	Fahrzeug ist in der Werkstatt
...	...

Die erste Betriebssituation beschreibt das Fahren auf einer Autobahn. Da hier höhere Geschwindigkeiten vorherrschen, können Gefährdungen besonders safety-kritisch sein. Ebenfalls stellt das schnelle Abbiegen eine Betriebssituation dar, die ein hohes Schadenspotenzial aufweist. Im Kontrast dazu stellt sich der Betriebszustand *Fahrzeug parken* als weniger kritisch dar, weil sich das Fahrzeug selbst nicht fortbewegt und somit kein Schaden für die Umwelt resultieren kann. Der letzte Betriebszustand (*Fahrzeug ist in der Werkstatt*) entspricht dabei keiner klassischen Betriebssituation, die bei der Safety-Analyse betrachtet wird. Diese untypischen Situationen sind dennoch relevant, da sie eine wichtige Rolle bei der Bedrohungsanalyse spielen können. Insbesondere die Betrachtung von Wartungs- und Werkstattsituationen kann sinnvoll sein. Hier entsteht die Gegebenheit, dass mit dem Fahrzeug-internen Netzwerk – durch Diagnosesysteme – kommuniziert wird und besondere Softwarefunktionen (Diagnosefunktionen) aktiviert werden.

Schließlich lassen sich durch die Kombinierung der Betriebssituationen und der Gefährdungen aus den Tabellen 4.4 und 4.5 die Gefährdungssituationen bestimmen. Ein Auszug dieser ist in Tabelle 4.6 aufgelistet.

Tabelle 4.6: Identifizierte Gefährdungssituationen durch Kombinationen von Tabelle 4.4 und Tabelle 4.5.

GS-ID	Beschreibung der Gefährdungssituation	G-ID	O-ID
1	Verlust der Lenkung durch ungewolltes Verriegeln der Lenksäule während der Autobahnfahrt.	3	1
2	Verlust der Lenkung durch ungewolltes Verriegeln der Lenksäule bei schneller Kurvenfahrt.	3	2
3	Lenksäule ist nicht verriegelt, während das Fahrzeug parkt.	1	3
...	...		

Als hoch-kritisch sind dabei die ersten beiden Situationen (in Tabelle 4.6) einzustufen. Zum einen können beide zu Verlust der Lenkkontrolle des Fahrzeuges führen, was mit hoher Wahrscheinlichkeit in einem schweren Unfall resultieren kann. Zum anderen sind

es Situationen, die sehr häufig im Fahrzeugleben vorkommen, sodass eine hohe Wahrscheinlichkeit für einen Unfall gegeben ist. Weniger kritisch für den Vermögenswert *Safety* ist die Gefährdungssituation 3, bei der die Lenksäule des Fahrzeugs beim Parken nicht verriegelt wird. Das Fahrzeug bewegt sich in diesem Betriebszustand nicht, sodass keine Gefährdung für die Umwelt ausgeht. Aus dem Blickwinkel möglicher anderer Vermögenswerte ist die Betriebssituation 3 in Tabelle 4.7 im finanziellen Kontext interessant. So beschreibt diese eine nicht ordnungsgemäß funktionierende Lenkradverriegelung, was den Diebstahl des Fahrzeuges erleichtert und einen finanziellen Schaden bedeuten kann.

An dieser Stelle sind die notwendigen Artefakte erzeugt, um mit der Bedrohungsanalyse (**Methodenschritt 5**) beginnen zu können. Wie in Abschnitt 4.3 erläutert werden die Gefährdungen aus der Safety-Analyse in die SGM-Vorlage übernommen, was der zweiten Spalte in Tabelle 4.7 entspricht. Für das aufgezeigte Beispiel ist hierzu die Gefährdung *ungewolltes Verriegeln* ausgewählt, was durch den Bezeichner G-ID mit der Nummer 3 repräsentiert ist (G-ID=3). Als Security-Leitwörter sind *auslösen*, *zurücksetzen* und *manipulieren* in der dritten Spalte eingesetzt, um die möglichen Bedrohungen zu identifizieren.

Tabelle 4.7: Beispielhafte Bedrohungsanalyse des ESCL-Systems mit der SGM-Vorlage sowie den Ausfällen aus Tabelle 4.4. G-ID=3 entspricht der Gefährdung *ungewolltes Verriegeln* aus Tabelle 4.4.

B-ID	G-ID	kann ausgelöst werden durch	Signal oder (Diagnose-) Funktion	für Komponente oder Teilsystem	Eintrittspunkt
1	3	auslösen	Verriegelung	ESCL-ECU	CAN-1
2	3	zurücksetzen	-	ESCL-ECU	P2P, CAN-1
3	3	manipulieren	ESCL{Verriegeln, Öffnen}	ESCL	P2P
4	3	manipulieren	Geschwindigkeit {Wert}	ESCL-ECU	CAN-1, CAN-2
...	...	...	...	...	...

So ergibt sich in der ersten Zeile (B-ID=1) die Bedrohung: *Ungewolltes Verriegeln der Lenksäule kann ausgelöst werden durch Auslösen der Funktion Verriegeln der Komponente ESCL-ECU über CAN-1.* Die aufgedeckte Bedrohung beschreibt dabei das nicht autorisierte Ausführen der Verriegelungsfunktion, die in der ESCL-ECU implementiert ist. Zusätzlich unterstützt das Präfix *Diagnose-* – in der Beschriftung der vierten Spalte – das Aufdecken von Bedrohungen, die auf einen Missbrauch einer Diagnosefunktion abzielen. Das ist damit zu begründen, dass Steuergeräte, die Aktuatoren kontrollieren, mit hoher Wahrscheinlichkeit Diagnosefunktionen aufweisen,

um diese überprüfen zu können (siehe Metamodell in Abbildung 4.4). Die zweite Bedrohung beschreibt das Zurücksetzen der ESCL-ECU, was beispielsweise durch Reset-Signale oder durch eine Überlastung der ECU ausgelöst werden kann. Letzteres kann durch einen DoS-Angriff auf den Kommunikationsverbindungen *CAN-1* und *P2P* erreicht werden. Eine weitere Bedrohungsart stellen die Bedrohungen 3 und 4 dar. Diese spiegeln das Manipulieren von Werten auf den Kommunikationsverbindungen wieder. So wird im Falle von Bedrohung 3 der Zustand des Signals *ESCL* auf den Zustand Verriegeln gesetzt, um den Schließzylinder in die Lenksäule einzufahren und das Lenkrad zu sperren. Bei Bedrohung 4 wird hingegen durch eine Manipulation des Geschwindigkeitswertes ein falscher Betriebszustand vom Steuergerät ESCL-ECU angenommen. Konkret könnte dies durch das Setzen eines Geschwindigkeitswertes auf den Wert 0 erreicht werden, was dem Steuergerät den Betriebszustand Parken vortäuscht und damit die Lenksäule verriegelt. An dieser Stelle muss allerdings erwähnt werden, dass die konkreten Ausprägungen der Bedrohungen – wie es ein DoS-Angriff ist – vom Analysten bestimmt werden müssen. Da sich allerdings zu jedem Leitwort eine Liste von Angriffen zuordnen lässt, kann dem Analysten eine « Angriffsliste » bereitgestellt werden.

Die in Tabelle 4.7 präsentierten Beispielbedrohungen zeigen die einfache Anwendung der SGM-Vorlage, um Bedrohungen anhand von Safety-Artefakten abzuleiten. Die Vorlage leitet den Analysten anhand der jeweiligen Tabellenspalte und der Leitwörter strukturiert durch die Bedrohungsanalyse. Mittels dem Hinweis auf Diagnosefunktionen in Spalte 4 wird außerdem ein zusätzlicher Fokus auf Diagnosefunktionalitäten gesetzt, die eventuell vergessen werden könnten. Im nächsten Schritt wechselt der Ablauf zurück zur Safety-Analyse, um die Gefährdungssituationen zu bewerten, was dem **Methodenschritt 6** entspricht. Hierzu werden die Situationen anhand der drei Bewertungsklassen der ISO 26262 [99, Seiten 9–10] klassifiziert. Zu bestimmen ist der Schaden (engl. severity), die Häufigkeit der Betriebssituation (engl. exposure of operational situations) und die Kontrollierbarkeit (engl. controllability). Bezogen auf die Gefährdungssituationen in Tabelle 4.6 ergibt sich als Ergebnis die Tabelle 4.8 mit der ASIL-Einstufung in der fünften Spalte.

Tabelle 4.8: Bewertung der Gefährdungssituationen aus Tabelle 4.6 mittels der ISO 26262 [99, Seiten 9–10]. Die Metriken hierzu sind die Schwere des Schadens (engl. severity), die Häufigkeit der Betriebssituation (engl. exposure of operational situations) und die Kontrollierbarkeit (engl. controllability). Die letzte Spalte zeigt den sich ergebenen ASIL-Wert.

GS-ID	Schaden (S)	Häufigkeit (E)	Kontrollierbarkeit (C)	ASIL
1	S3	E4	C3	D
2	S3	E4	C3	D
3	S0	E4	C0	QM

Für die ersten beiden Gefährdungssituationen ergibt sich ein ASIL der Stufe D, was der höchsten Einstufung entspricht. Für die erste Gefährdungssituation ergibt sich dies durch den zu erwarteten schweren Schaden (S3), der eintritt, wenn bei einer Autobahnfahrt die Lenksäule verriegelt wird und der Fahrer das Fahrzeug nicht mehr kontrollieren kann. Darüber hinaus entspricht eine Autobahnfahrt einer Betriebssituation, die häufig auftritt und daher mit E4 bewertet wird. Ein Fahrer wird außerdem aufgrund der hohen Fahrzeuggeschwindigkeiten, nicht in der Lage sein eine Gefahrensituation abzuwenden, die durch ein blockierendes Lenkrad ausgelöst wird. Dies führt in der Klasse Kontrollierbarkeit zum höchsten Wert C3. Die gleiche Einordnung erhält die zweite Gefährdungssituation in Tabelle 4.8. Die dritte hingegen erhält die ASIL-Einstufung QM. Dies ist damit zu begründen, dass ein nicht verriegeltes Lenkrad während des Parkens keinen Schaden für die Umwelt auslösen kann. Für die Häufigkeit ergibt sich die Einstufung E4, für die Kontrollierbarkeit der Situation ergibt sich eine Klassifizierung von C0, da die Kontrollierbarkeit vollends gegeben ist.

Unter Einbezug der ASIL-Klassifizierung können die identifizierten Bedrohungen bewertet werden. Hierzu wird wie in **Methodenschritt 7** beschrieben, für jede Bedrohung, die zugehörige Gefährdung ausgewählt. Die entspricht der ersten Mengenabbildung (Bedrohung → Gefährdung) in Abbildung 4.13. Die zweite Mengenabbildung entspricht der anschließenden Zuordnung von Gefährdungen zu Gefährdungssituationen. Hiermit werden jene Situationen ausgewählt, die sich durch die Gefährdung ergeben können. Aus dieser Menge von Gefährdungssituationen (graue Fläche in Abbildung 4.13) wird anschließend jenes ASIL für die Bewertung ausgewählt, welches das höchste Risiko besitzt. Für die Bedrohung B-ID=1 aus Tabelle 4.7 bedeutet dies beispielsweise, dass die zugeordnete Gefährdung G-ID=3 zu den Gefährdungssituationen GS-ID=1 und GS-ID=2 führt. Diese sind nach Tabelle 4.8 jeweils mit einem ASIL-D bewertet. Mit dem Maximum aus diesen beiden Bewertungen ergibt sich eine Risikoeinstufung von ASIL-D. Zusätzlich kann diese Einstufung einem qualitativen Risikowert wie *Critical* zugeordnet werden, wenn beispielsweise die Zuordnung von HEAVENS verwendet wird (Tabelle A.7).

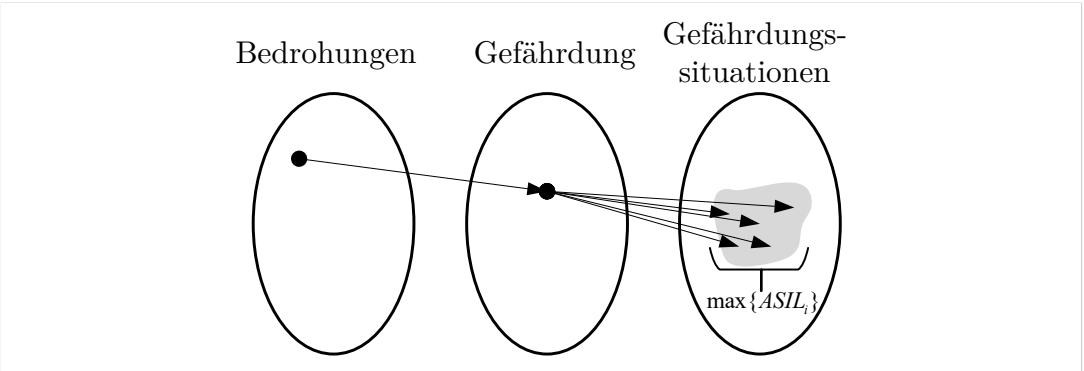


Abbildung 4.13: Darstellung der Beziehungen zwischen den Mengen Bedrohungen, Gefährdung und Gefährdungssituationen, die zur Klassifizierung von Bedrohungen verwendet werden.

An dieser Stelle ist die Bedrohungsanalyse mit der SGM abgeschlossen. Nachfolgend soll jedoch kurz auf das Ableiten von Security-Anforderungen anhand der SGM-Ergebnisse eingegangen werden. Als Beispiel ist hierzu Bedrohung 1 ausgewählt und in die Spalte *Bedrohung* von Tabelle 4.9 übertragen.

Tabelle 4.9: Gegenüberstellung von Bedrohungen 1 aus Tabelle 4.7 und der daraus abgeleiteten Security-Anforderung in der zweiten Spalte.

Bedrohung	Security-Anforderung
Ein Angreifer kann eine ungewollte Verriegelung des Lenkrads verursachen, indem er die Verriegelungsfunktion des Aktuators über die Komponente ESCL-ECU am Bussystem CAN-1 auslöst.	Die Verriegelungsfunktion der Komponente ESCL-ECU am Bussystem CAN-1 muss vor nicht autorisierten Zugriff geschützt werden.

Aus Gründen der Lesbarkeit ist die Beschreibung der ursprünglichen Bedrohung mit dem Begriff des Angreifers erweitert worden, um die auszuführende Entität zu nennen. Mittels der Zuordnung des verwendeten Leitwortes *Auslösen* auf den IT-Vermögenswerten *Authentizität* und *Integrität* in Tabelle 4.1, lässt sich eine Security-Anforderung ableiten [2, 124]. Konkret kann anhand der Verletzung der IT-Vermögenswerte die Notwendigkeit für den Schutz von unbefugten Zugriff hergeleitet werden. Damit ergibt sich die Security-Anforderung in der rechten Spalte von Tabelle 4.9, die im weiteren Entwicklungsverlauf kontinuierlich konkretisiert wird und schlussendlich in der Implementierung einer Maßnahme endet.

4.4 Analytische Bewertung und kritische Auseinandersetzung

Die vorgestellte SGM ermöglicht es, in einer frühen Konzeptphase eine Bedrohungsanalyse bezogen auf den Vermögenswert Safety durchzuführen. Basierend auf dem Konzept von HAZOP sind eine High-Level Architektur und abstrakte Informationsflüsse ausreichend, um Bedrohungen identifizieren zu können. Hierzu dienen insbesondere die SGM-Leitwörter, die als generalisierte Angriffstechniken interpretiert werden können und den Analysten bei der Identifizierung der Bedrohungen leiten.

Anhand der entwickelten Vorlage wird außerdem der direkte Zusammenhang zwischen Gefährdungen und den Verletzungen von Security-Schutzzielen hergestellt. Dies erlaubt ein besseres Verständnis der Wechselwirkungen der beiden Studienfelder, was die Kommunikation zwischen beiden Welten verbessert. Zusätzlich lenkt die Vorlage – durch Spalte *Signal oder (Diagnose-)Funktion* – den Fokus auf Diagnosefunktionalitäten, die einen relevanten Anteil der bisher gezeigten Angriffe darstellen (Tabelle 4.7). Weiterhin sammelt die SGM-Vorlage neben den Safety-Artefakten ebenfalls mögliche Eintrittspunkte, die zur Realisierung der Bedrohung notwendig sind. Dies ermöglicht es frühzeitig abzuschätzen, welchen Zugriff (physisch, kabelgebunden, kabellos) ein Angreifer benötigen wird. Zur Risikobewertung der Bedrohungen verwendet der Ansatz ebenfalls Safety-Artefakte, um den Aufwand der Bewertung von Bedrohungen zu verringern. Ein Teil dieser Safety-Artefakte ist allerdings nicht vollkommen passend für die Bewertung von Cyber-Bedrohungen. So ist es sinnvoll den Schadenswert (S) aus der Safety-Analyse zu verwenden, die Kontrollierbarkeit (C) und Häufigkeit der Betriebssituation (E) bergen hingegen Diskussionspotenzial bzgl. Security. So kann ein Angreifer Maßnahmen ergreifen, welche es dem Fahrer erschwert, einen Safety-Vorfall abzuwenden, was die Bestimmung der Kontrollierbarkeit infrage stellt. Ebenfalls kann angenommen werden, dass ein Angreifer seinen Angriff in der Weise plant, dass er in der gefährlichsten Betriebssituation stattfindet. Somit müsste stets der höchste Wert für die Häufigkeit der Betriebssituation gewählt werden, was die Verteilung der Risikobewertung negativ beeinflusst. Aufgrund dessen wird in Kapitel 7 ein Vorgehen präsentiert, das weiterhin den Schadenswert der Safety übernimmt, die Eintrittswahrscheinlichkeit allerdings anhand von Angriffspfaden und Security-Metriken bestimmt.

5

Empirische Evaluierung der SGM

Im letzten Kapitel wurde die SGM analytisch betrachtet und auf eine sinnvolle Anwendung geschlossen. Damit diese Behauptung weiter untermauert werden kann, wird in diesem Kapitel eine experimentelle Evaluierung der SGM dargelegt. Das Experiment misst mittels vier Klassen, in welcher Qualität die Teilnehmer des Experiments Bedrohungen identifizieren konnten. Das genaue Design des Experiments ist in Abschnitt 5.2 erläutert.

Bezogen auf die Analysemethoden aus Kapitel 3 wurde für die Evaluierung der SGM ein empirischer Ansatz ausgewählt. Diese Entscheidung ist mit der Situation zu begründen, dass die bestehenden Analysemethoden anhand von Beispielanwendungen ihren Nutzen argumentieren, die für Außenstehende allerdings schwer nachzuvollziehen sind. Aufgrund dessen ist ein empirischer Ansatz mit Hypothesentests für die Evaluierung ausgewählt, der sich an den Arbeiten der Forscher Scandariato et al. [176] orientiert. Diese evaluierten mit diesem Vorgehen den Microsoft STRIDE-Ansatz. Darüber hinaus sind empirische Evaluierungen ein gängiger Ansatz, um Methoden in den Domänen Software Engineering [187, Seite 1] und Safety [105, 193, 194, 195] zu evaluieren. Der Aufbau der Evaluation basiert auf Seltman et al. [183, Seite 2] und Beckers et al. [29]. Hiermit grenzt sich die SGM weiter vom Stand der Technik ab, da die Ansätze in Kapitel 3 keine empirische Evaluierung aufzeigen.

Der übergeordnete Ablauf der Evaluierung ist mit Abbildung 5.1 gezeigt und beschreibt die Inhalte der nachfolgenden Abschnitte. So wird, ausgehend vom aktuellen Stand des Wissens, eine primäre Hypothese aufgestellt, welche in vier Testhypothesen aufgeteilt wird. Diese überprüfen, in welcher Qualität die SGM Bedrohungen identifizieren kann. Hierzu wird der Entwurf des Experiments erläutert, der die notwendigen Daten für die Hypothesentests erhebt und mit mehreren Teilnehmergruppen durchgeführt wurde. Basierend auf diesen Ergebnissen zeigt die statistische Auswertung, ob die definierten Hypothesen akzeptiert oder abgelehnt werden können. Hierzu wird ein rechtsseitiger t-Test [32, Seite 155] eingesetzt, der auf die normalverteilten Stichproben angewendet

wird (Abschnitt 5.3). Zur Bestimmung der Normalverteilung der Zufallsvariable wurde jeweils ein Shapiro-Wilk-Test [82, Seite 466] angewendet (Tabelle 5.2). An dieser Stelle muss darauf hingewiesen werden, das die Statistik auf kleinen Zahlen beruht und eine Konfidenz von 90% aufweist (Abschnitt 5.4). Abschließend wird der Ausgang des Experiments interpretiert und die Ergebnisse kritisch diskutiert. Der Aufbau des Experiments und die ausgewählten Methoden – wie die Hypothesentests – entsprechen dabei dem anerkannten Vorgehen für die experimentelle Evaluierung einer Methodik [78, 183, 204, 210].

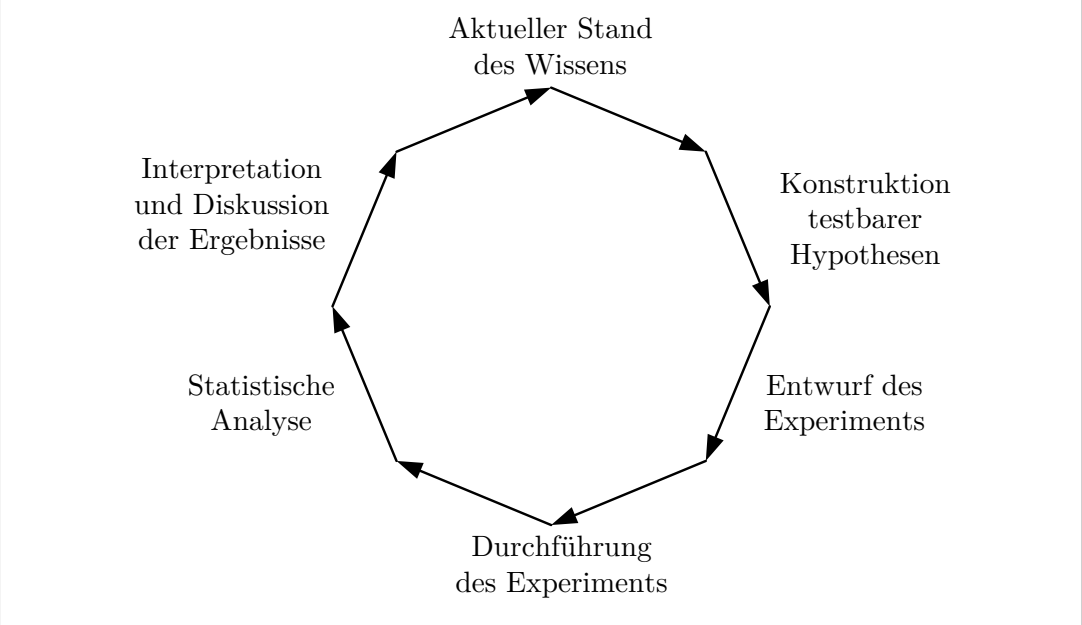


Abbildung 5.1: Phasen der empirischen Evaluation der SGM basierend auf [183, Seite 2].

Für die empirische Evaluation wurden zwei Gruppen gebildet. Die erste Gruppe bestand aus Security-Ingenieuren eines Automobilzulieferers und führte die Bedrohungsanalyse ohne die SGM durch. Diese waren frei bei der Auswahl ihrer Analysemethode, wobei sich die Mehrheit der Teilnehmer für den STRIDE-Ansatz [137] entschied. Für die zweite Gruppen, welche die Bedrohungsanalyse ohne die SGM durchführte, wurden Safety-Ingenieure des gleichen Automobilzulieferers sowie festangestellte wissenschaftliche Mitarbeiter der Hochschule Karlsruhe – Technik und Wirtschaft gewonnen. Mittels einer Teilnehmerbefragung, die dem Experiment voran ging, konnte aufgezeigt werden, dass die Safety-Ingenieure und wissenschaftlichen Mitarbeiter ein geringes Wissen im Bereich Security aufwiesen und deshalb als die Menge der Nicht-Security-Ingenieure angesehen werden (näheres in Anhang B.1). Die Auswahl von Safety-Ingenieuren ist mit der Annahme verknüpft, dass die SGM eine Verknüpfung zwischen Safety und Security etabliert und Safety-Ingenieure damit in die Lage versetzt werden, Schutzziele und Bedrohungen aufdecken zu können. Die ausgewählten wissenschaftlichen Mitarbeiter stellen sich als Kontrollgruppe dar, die wenig Erfahrung in beiden Domänen besitzt (Safety- und Security). Aufgrund dessen wurde diese Gruppe nicht in die

Hypothesentests in Abschnitt 5.1 einbezogen. Nichtsdestotrotz sind die Ergebnisse dieser Gruppe interessant, da sie aufzeigen wie unerfahrene Personen mit der SGM in der Lage sind, Bedrohungen zu identifizieren. Es wird außerdem angenommen, dass die Teilnehmer der drei Gruppen als Ingenieure angesehen werden können und die Grundmenge für das Experiment bilden. Die genaue Zusammensetzung der Teilnehmer ist in Abbildung 5.2 gezeigt.

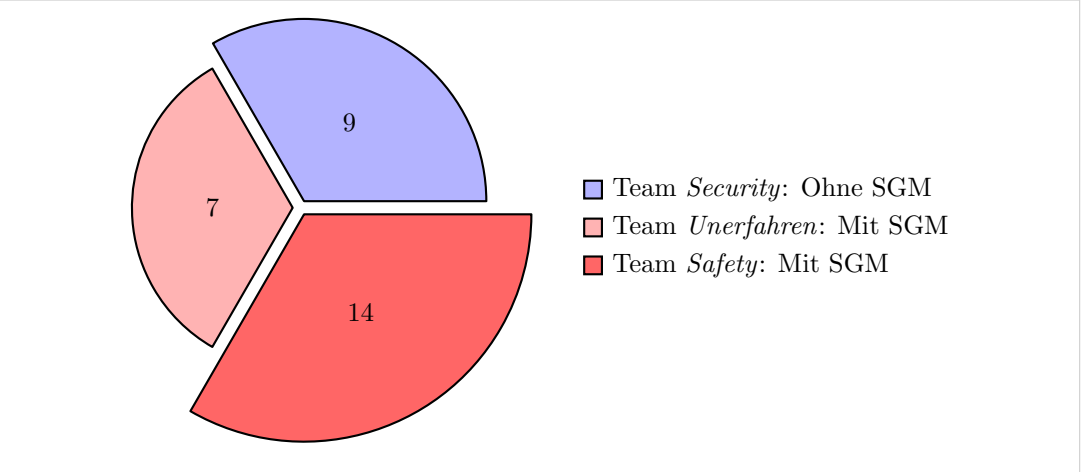


Abbildung 5.2: Zusammensetzung der Teilnehmer für die Evaluierung der SGM.

Insgesamt nahmen 30 Teilnehmer bei der Evaluierung teil, die aus dem automotiven Umfeld entstammten. Team *Unerfahren* in Abbildung 5.2 entspricht dabei den oben genannten wissenschaftlichen Mitarbeitern und Team *Safety* jenen, die zur Auswertung der Hypothesentests herangezogen wurden. Obwohl die Stichprobe mit $N = 14$ für dieses Team als klein zu bewerten ist, ist der ausgewählte Test dennoch in der Lage, Aussagen über die getroffenen Hypothesen abzuleiten (Abschnitt 5.4). Der Aufbau des Experiments und dessen Durchführung ist in den nächsten Abschnitten erläutert.

5.1 Aufstellen der Hypothesen

Bevor das Studiendesign und die Ergebnisse des Experiments aufgezeigt werden, werden im Folgenden die primäre Hypothese sowie die dazu gehörigen Testhypothesen für die empirische Evaluierung erläutert. Hierbei ist die Grundannahme getroffen, dass Safety-Ingenieure grundsätzlich weniger Wissen im Security-Bereich aufweisen und ohne die SGM schlechtere Ergebnisse bei einer (safety-fokussierten) Bedrohungsanalyse erzielen als Security-Ingenieure. Daraus folgt die Annahme, dass Safety-Ingenieure mit der SGM in der Lage sind, Schutzziele in ähnlicher Qualität zu identifizieren wie die Security-Ingenieure ohne die SGM, wenn der Vermögenswert *Safety* betrachtet wird. Hiermit lässt sich die primäre Hypothese formulieren:

Hypothese (H5.1)**.** *Mit der SGM sind Safety-Ingenieure in der Lage, safety-relevante Bedrohungen und Schutzziele in mindestens der gleichen Weise aufzudecken wie Security-Ingenieure ohne die SGM.*

Damit die primäre Hypothese H5.1 getestet werden konnte, wurden vier Testhypothesen definiert, die sich auf die Zufallsvariablen True Positives (TP) und False Positives (FP) beziehen [66, 112, 176]. Damit eine Testhypothese (H_1) bestätigt werden kann, muss die dazugehörige Nullhypothese (H_0) abgelehnt werden. Letztere sind durch die Nullhypothesen 1 bis 4 repräsentiert. Zur Auswertung dieser werden die Erwartungswerte $\mu\{TP_{Saf}\}$, $\mu\{FP_{Saf}\}$, $\mu\{PPV_{Saf}\}$ und $\mu\{PRO_{Saf}\}$ betrachtet. Hierbei entsprechen die TPs der Anzahl der korrekt aufgedeckten und FPs den falsch aufgedeckten Bedrohungen[1]. Zur Bestimmung, ob eine Bedrohung als korrekt gewertet werden kann, überprüften zwei Experten aus dem Safety und Security-Umfeld die von den Teilnehmern identifizierten Bedrohungen im Arbeitsblatt. Beide Experten nahmen nicht an dem Experiment teil, sodass sie als unabhängig zu werten sind. Wurde durch die Experten eine Bedrohung hingegen als falsch gewertet, so entspricht diese einer FP. Die Anzahl der Zeilen (Bedrohungen), die ein Teilnehmer ausfüllte, wurde mit dem Wert Fields Initiated (FI) festgehalten. Abgesehen davon wurde den Teilnehmern eine feste Anzahl von Zeilen vorgegeben, die sie maximal ausfüllen konnten. Diese sind durch die Number of Fields (NF) repräsentiert. Das Maximum wurde auf 15 Zeilen festgelegt und entspricht bei einer Analysedauer von 15 Minuten, einer Zeile beziehungsweise Bedrohung pro Minute. Dieser Umfang zeigte sich nach Auswertung eines Pilotversuchs – mit Security-Ingenieuren – als sinnvoll.

Die erste Hypothese entspricht der Annahme, dass das Safety-Team im Durchschnitt mindestens die gleiche Anzahl an korrekten Bedrohungen (TPs) wie das Security-Team aufdeckt ($H^{TP}_{1,Saf} : \mu\{TP_{Saf}\} \geq \mu\{TP_{Sec}\}$). Die dazugehörige Nullhypothese stellt in Folge die Annahme, dass die Safety-Ingenieure eine geringere Anzahl von korrekten Bedrohungen auffinden, was zu Nullhypothese 1 führt.

Nullhypothese 1 ($H_0 1$)**.** $H^{TP}_{0,Saf} : \mu\{TP_{Saf}\} < \mu\{TP_{Sec}\}$

Im ähnlichen Sinne überprüft die zweite Hypothese die Annahme, dass das Safety-Team im Durchschnitt keine höhere Anzahl an falschen Bedrohungen FP identifiziert als die Security-Ingieure ($H^{FP}_{1,Saf} : \mu\{FP_{Saf}\} \leq \mu\{FP_{Sec}\}$). Die Nullhypothese lässt sich dann folgendermaßen bilden.

Nullhypothese 2 ($H_0 2$)**.** $H^{FP}_{0,Saf} : \mu\{FP_{Saf}\} > \mu\{FP_{Sec}\}$

Die dritte Testhypothese beschreibt hingegen die von den Teilnehmern erreichte Präzision (PPV). Diese kann auch als Wert der positiven Vorhersagen verstanden

[1] Die Teilnehmer des Experiments hatten die Aufgabe ausschließlich safety-relevante Bedrohungen zu identifizieren.

werden und sagt aus, wie wahrscheinlich es ist, dass die vom Teilnehmer als korrekt markierten Bedrohungen tatsächlich zutreffend (TP) sind. Zur Bestimmung dieser Metrik und bezogen auf die Zufallsvariablen TP und FP kann Gleichung (5.1) verwendet werden.

$$PPV = \frac{TP}{TP + FP} \tag{5.1}$$

Für die daraus folgende Testhypothese wird die Annahme getroffen, dass Team *Safety* im Durchschnitt mindestens den gleichen Wert für die Vorhersage korrekter Bedrohungen erreicht, als die Security-Gruppe ($H^{PPV}_{1,Saf} : \mu\{PPV_{Saf}\} \geq \mu\{PPV_{Sec}\}$). Dies wurde in ähnlicher Weise bereits von Scandariato et al. [176] zur Evaluierung des STRIDE-Ansatzes eingesetzt. Die Forscher trafen allerdings die Annahme, dass die Testgruppe im Mittel der Ergebnisse einen Schwellwert von 80 % überschreitet. Da in dieser Arbeit die erzielten Ergebnisse allerdings relativ zwischen den Gruppen verglichen werden konnten, ergibt sich für PPV die folgende Nullhypothese:

Nullhypothese 3 ($H_0 3$)**.** $H^{PPV}_{0,Saf} : \mu\{PPV_{Saf}\} < \mu\{PPV_{Sec}\}$

Somit beschreibt Nullhypothese 3, dass im Mittel die Safety-Ingenieure eine geringere Präzision erreichen als Team *Security*, was der negierten Testhypothese entspricht. Im gleichen Sinne ist die Nullhypothese für die Produktivität (PRO) definiert. Letztere stellt sich als Verhältnis zwischen der Anzahl der gefundenen (NI) und zu der maximal ausfüllbaren Anzahl von Zeilen ($NF := 15$) dar und ist mit Gleichung (5.2) gezeigt.

$$PRO = \frac{NI}{NF} \tag{5.2}$$

Damit beschreibt PRO, wie viele Bedrohungen jeder Teilnehmer gesamtheitlich identifizierte. Die Testhypothese ist hierbei, dass das Safety-Team im Durchschnitt eine höhere Anzahl von Bedrohungen aufdeckt als Team-Security. Die dazugehörige Nullhypothese lässt sich folgendermaßen beschreiben:

Nullhypothese 4 ($H_0 4$)**.** $H^{PRO}_{0,Saf} : \mu\{PRO_{Saf}\} \leq \mu\{PRO_{Sec}\}$

Nullhypothese 4 zeigt somit, dass die Safety-Ingenieure eine geringere Produktivität aufweisen als die Security-Kollegen.

5.2 Design des Experiments

Das zugrundeliegende Studiendesign orientiert sich an den Arbeiten von Scandariato et. al [176] sowie Beckers et al. [29]. Das Vorgehen ist in drei Phasen eingeteilt und in

Abbildung 5.3 aufgezeigt. Die erste Phase beinhaltete eine kurze Einführung in die funktionale Safety und die automotive Security. Außerdem wurden den Teilnehmern der Zusammenhang zwischen Safety- und Security-Vorfällen erläutert und wie sich die gegenseitige Beeinflussung darstellt. Ebenfalls wurden die Begriffe Safety und Security anhand ihre grundlegenden Unterschiede erläutert. Dies war notwendig, da die Teams *Unerfahren* und *Safety* bisher nicht an Security-Themen gearbeitet hatten.

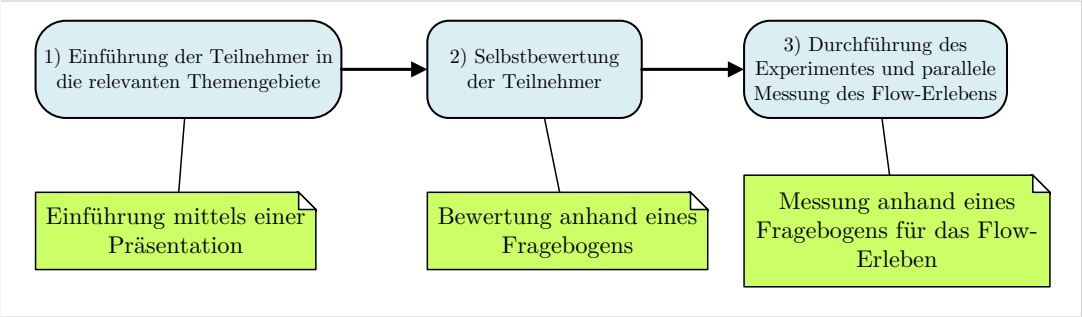


Abbildung 5.3: Darstellung der drei Phasen des Experiments und welche Arbeitsmittel in der jeweiligen Phase eingesetzt wurden.

In der zweiten Phase wurde ein Fragebogen an die Teilnehmer verteilt, der das Wissen und die Erfahrung jedes einzelnen Teilnehmers bewertete. Der Fragebogen ist in zwei Teile gegliedert und in Anhang B.2.4 gezeigt. Der erste Teil beschreibt eine Selbstbewertung bezogen auf die Themen: Automotive Software-Engineering, funktionale Sicherheit, automotive Bussysteme, Echtzeitsysteme, Security sowie Gefährdungs- und Bedrohungsanalysen. Die Teilnehmer wählten aus fünf Erfahrungsstufen, die von unerfahren bis Experte reichten. In der zweiten Phase wurden zu allen zuvor genannten Themen Fragen gestellt, um die von den Teilnehmern ausgefüllte Selbstbewertung validieren zu können. Außerdem wurde das Flow-Erleben der Teilnehmer während des Experiments anhand der Skala von Rheinberg et al. [164] bewertet. Diese zeigt drei Kriterien *Flow*, *Bedenken* und *Herausforderung*, wobei sich die Werte jeweils im Intervall $[1, 7]$ bewegen. Die Bewertung muss allerdings unterschiedlich interpretiert werden. So beschreibt ein höherer Wert für das Kriterium *Flow*, dass sich die Teilnehmer stärker im Arbeitsablauf befand, was als positiv zu werten ist. Für das Kriterium *Bedenken* hingegen ist ein steigender Wert als negativ zu bewerten, da die Bedenken des Teilnehmers vor der Aufgabe steigen. Ähnlich verhält es sich mit dem Kriterium *Herausforderung*, bei dem ein kleinerer Wert als besser anzusehen ist. So beschreibt letzterer, dass sich der Teilnehmer weniger von der Aufgabe herausgefordert fühlten. Anhand dieser Kriterien ist es möglich zu erfassen, wie gefordert sich die Teilnehmer bei der Durchführung der Bedrohungsanalyse fühlten. Hiermit lässt sich anschließend argumentieren wie einfach sich die Anwendung der SGM gestaltet.

Der praktische Teil des Experiments startete mit einer Einführung der SGM. Hierbei wurden die Methodenschritte aus Abschnitt 4.3 detailliert erklärt und anhand eines Beispieles eines Bremssystems die Anwendung aufgezeigt. Außerdem wurden den

Teilnehmern die Arbeitsblätter und der Kontext des ESCL-Systems erklärt, das als Analysegegenstand diente. Anschließend wurden die Teilnehmer gebeten, die Methode am ESCL-System anzuwenden. Das Beispiel wurde sorgfältig ausgewählt, um eine Komplexität zu gewährleisten, die innerhalb eines Zeitrahmens von bis zu 15 Minuten bewältigt werden konnte. Die Entscheidung für den ausgewählten Zeitrahmen war damit begründet, dass das Experiment insgesamt nicht mehr als eine Zeitstunde benötigen sollte. Hierzu wurde das ESCL-System in der Abbildung 5.4 verwendet. Dieses entspricht grundsätzlich der Topologie von Abbildung 4.12, setzt allerdings auf eine kompaktere Notation, die der UML-Erweiterung UML4PF [81] entspricht. Diese verwendet Stereotypen, um Elementen eine bestimmte Bedeutung zu geben. So beschreibt $\ll domain \gg$ beispielsweise bereits vorhandene Elemente in der Umgebung des Analysegegenstandes (ESCL). Die Klasse mit der Erweiterung $\ll GW \gg$, hingegen bedeudet, dass es sich um ein Gateway handelt und der Zusatz $\ll item \gg$, dass es sich um den Gegenstand der Analyse handelt. Die Kommunikation zwischen Elementen wird dabei durch UML-Assoziationen repräsentiert, die mittels Stereotypen die Art der Verbindung erklären. So beschreibt $\ll can \gg$ eine CAN-Bus-Verbindung und $\ll physical \gg$ eine Ende-zu Ende Verbindung, die analoge Signale zur Kommunikation verwendet.

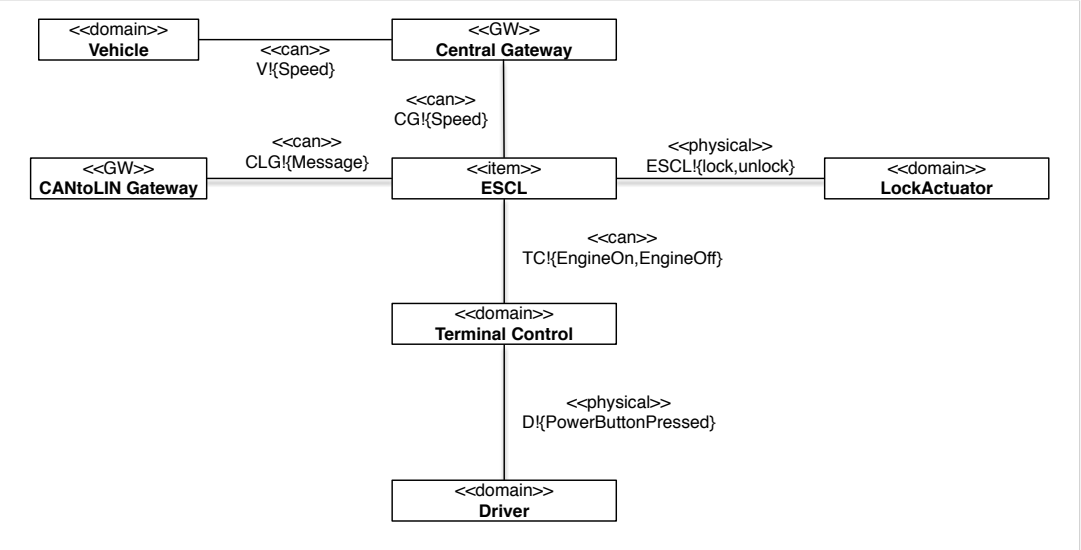


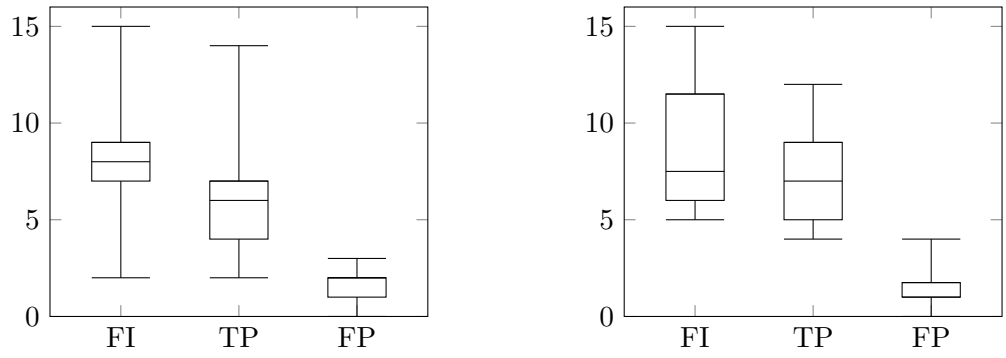
Abbildung 5.4: Kontext-Diagramm des ESCL-Systems entnommen aus [1]. UML-Klassen zeigen die relevanten Elemente des Analyseumfangs sowie jene Elemente, die mit dem Analysegegenstand (ESCL) kommunizieren. Mittels der UML4PF [81] Erweiterung ist durch Stereotypen ($\ll$*Stereotyp*$\gg$) den Elementen eine bestimmte Bedeutung zugewiesen. Gezeigt sind zwei Gateways ($\ll$*gw*$\gg$), vier Domänen ($\ll$*domain*$\gg$), der Analysegegenstand ($\ll$*item*$\gg$) sowie fünf Kommunikationsverbindungen ($\ll$*physical*$\gg$,$\ll$*can*$\gg$).

Neben dem Kontext-Diagramm wurden den Teilnehmern ebenfalls die Funktionalitäten des Systems genannt und zwei Ausfälle erläutert. Konkret sind das die Gefährdungen *ungewolltes Verriegeln* und *kein Verriegeln*. Das Experiment geht somit von einer bereits durchgeführten Safety-Analyse aus (Methodenschritte 1-4 in Abbildung 4.8).

Das Experiment wurde zweimal und mit verschiedenen Teilnehmern durchgeführt, um Voreingenommenheit der Teilnehmer auszuschließen. Der erste Durchlauf entsprach dabei dem Pilotversuch, welcher der Überprüfung auf Fehler in den Arbeitsunterlagen diente und ob die in der Selbstbewertung gestellten Fragen verständlich war. Nach dem Einpflegen aufgedeckter Verbesserungen wurde anschließend das Experiment mit den Teilnehmern durchgeführt. Die dabei gemessenen Ergebnisse sind im nächsten Abschnitt aufgeführt.

5.2.1 Zusammenfassung der Ergebnisse

Abschließend sollen an dieser Stelle die Whisker-Plots aus Anhang B.2 nochmals komprimiert aufgezeigt und jeweils gegenüber gestellt werden. Außerdem sind mit Tabelle 5.1 die Messergebnisse jener Klassen aufbereitet, die zum Hypothesen-Test in Abschnitt 5.3 verwendet werden.



(a) Team *Security* ohne SGM **(b) Team *Safety* mit SGM**

Abbildung 5.5: Ergebnisse für FI, TP und FP des Teams *Security* und *Safety*.

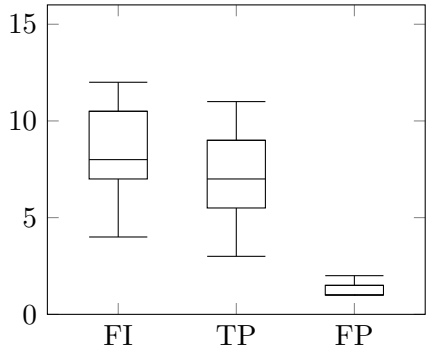


Abbildung 5.6: Ergebnisse für FI, TP und FP des Teams *Unerfahren* mit der SGM.

Tabelle 5.1: Gemessene Mittelwerte für das Team *Security*, *Safety* und *Unerfahren*.

Klasse	Security	Safety	Unerfahren
TP	6,11	7,21	7,14
FP	1,67	1,36	1,29
FI	7,778	8,57	8,43
PPV	78,29%	84,75%	83,33%
PRO	51,85%	57,14%	56,19%
Effizienz	31,111	34,29	33,71

Bei Vergleich der Ergebnisse von Team *Security* und *Safety* in Abbildungen 5.5a und 5.5b zeigt sich, dass beide Teams ähnliche Maxi- und Minima für die Anzahl der insgesamt identifizierten Bedrohungen (FI), der korrekt identifizierten Bedrohungen

(TP) und der falsch identifizierten Bedrohungen (FP) aufweisen. Ebenfalls zeigen die Mittelwerte der drei Klassen ähnliche Werte, was darauf hindeutet, dass die Safety-Ingenieure mit der SGM in der Lage sind, eine ähnliche Qualität bei der Bedrohungsanalyse zu erreichen wie das Security-Team. Bei genauer Betrachtung der beiden Whisker-Plots zeigt sich außerdem, das die Ergebnisse des Safety-Teams weniger gestreut sind als die des Security-Teams. Diese erhöhte Homogenität in den Ergebnissen kann mit der Annahme erklärt werden, das der strukturierte Analyseablauf – mit der SGM – eine gezieltere Identifikation der Bedrohungen erlaubt. So zeigt auch Team *Unerfahren* in Abbildung 5.6 eine geringere Streuung der Ergebnisse, wobei die Maxima unter der des Security-Teams liegen. Erstaunlich ist hingegen, dass eine unerfahrene Teilnehmergruppe – mit der SGM – fähig war, Messergebnisse bei der Analyse zu erreichen, die nur geringfügig schlechter waren, als die der Security-Gruppe (Abbildungen 5.5a und 5.6). Eine Erklärung hierfür kann sein, dass durch die textuelle SGM-Vorlage eine Fokussierung geschaffen wird, die es unerfahrenen Personengruppen ermöglicht, deduktiv Bedrohungen abzuleiten, wenn die Leitwörter generisch gehalten sind.

Ein weiterer Anhaltspunkt für die sinnvolle Anwendung der SGM kann aus Tabelle 5.1 abgeleitet werden. Diese zeigt die Durchschnittswerte für die True Positives (TP), FP, Präzision (PPV) und Produktivität (PRO) sowie die erreichten Mittelwerte für die Effizienz und die aufgedeckte Anzahl von Bedrohungen FI für die drei Teilnehmergruppen. Hierbei stellt sich heraus, dass Team *Safety* in allen fünf Kategorien die besten Durchschnittswerte erreichte. Das gleiche Bild zeichnet sich für Team *Unerfahren* ab, die trotz geringer Erfahrungen bei Security-Analysen, hohe Durchschnittswerte erreichten und mit 1,29 FPs den kleinsten Mittelwert der drei Gruppen erzielten.

Zusammenfassend kann an dieser Stelle gesagt werden, dass jene Teilnehmer, welche die SGM anwendeten, ähnlich gute Ergebnisse erreichten wie das Security-Team. Neben der höheren Homogenität der Ergebnisse in den Abbildungen 5.5b und 5.6, weisen die SGM-Gruppen einen geringen Median für inkorrekt identifizierte Bedrohungen (FP) auf. Dies ist als positiv zu bewerten, da ausschließlich korrekt aufgedeckte Bedrohungen (TP) weiter verwendet werden können. Diese Ergebnisse legen nahe, dass die SGM, Nicht-Security-Ingenieure in die Lage versetzt, safety-relevante Cyber-Bedrohungen aufzudecken. Darüber hinaus zeigt die Auswertung des Workflows einer dieser Teilnehmergruppen (Team *Safety*), dass sich die Teilnehmer – mit der SGM – mehr im Flow befanden, weniger Bedenken vor der Aufgabe hatten und sich weniger herausgefordert fühlten (Tabelle B.6). Dies unterstützt die Annahme der gezielten Unterstützung des Analysten durch die SGM.

5.3 Test der Hypothesen

Nach Auswertung der Merkmale FI, TP und FP wird in diesem Abschnitt die Gültigkeit der in Abschnitt 5.1 aufgestellten Hypothesen überprüft. Für die Auswahl einer Testart kann aus zwei Klassen gewählt werden. Die Klasse der parametrischen Tests erfordert eine zugrundeliegende Verteilung wie beispielsweise eine Normalverteilung, die nicht parametrischen Tests können hingegen angewendet werden, wenn keine bestimmte Verteilung vorliegt. Aufgrund der definierten Nullhypothesen ist es außerdem erforderlich, eine Testart zu wählen, die einen links- und rechtsseitigen Test erlaubt. Zwei Kandidaten, die dies erfüllen, sind der t-Test [32, Seite 155] im Falle einer normalverteilten Stichprobe und der Wilcoxon-Test [82, Seite 527], der keine Verteilung erfordert (nicht parametrischer Test). Darüber hinaus sind beide Tests für kleine Stichproben geeignet. So kann nach Winter [55] der t-Test für Stichprobengrößen $N \geq 5$ und der Shapiro-Wilk-Test für Stichprobengrößen von $N \geq 3 \leq 50$ eingesetzt werden [182, 185], was für die vorliegende Stichprobe von $N = 14$ zu trifft.

Für den Nachweis auf Normalverteilung für die Stichproben TP, FP, PPV, und PRO wurde der Shapiro-Wilk-Test [82, Seite 466] mit einem Signifikanzniveau von $\alpha = 5\ \%$ verwendet. Hierbei wird für jede Stichprobe überprüft, ob die zugrundeliegende Verteilung einer Normalverteilung entspricht (H_1^{Normal}) [82, Seite 466]. Die Stichprobe kann als normalverteilt angesehen werden, wenn der p-Wert des Tests größer als das gewählte Signifikanzniveau α ist, was in diesem Falle $p > 0,05$ bedeutet. Die Ergebnisse des Tests sind in Tabelle 5.2 zusammengefasst.

Tabelle 5.2: Ergebnisse für den Shapiro-Wilk-Test [82, Seite 466] mit einem Signifikanzniveau von $\alpha = 5\ \%$. Eine Stichprobe wird als normalverteilt angesehen, wenn der p-Wert des Tests größer als das Signifikanzniveau α ist (p-Wert $> 0,05$).

Stichprobe	H_1^{Normal}	p-Wert des Tests	Teststatistik W
TP_{Saf}	akzeptiert	0,17	0,89 %
FP_{Saf}	akzeptiert	0,09	0,89 %
PPV_{Saf}	akzeptiert	0,91	0,94 %
PRO_{Saf}	akzeptiert	0,12	0,88 %

Die zweite Spalte von Tabelle 5.2 zeigt auf, dass für die Stichproben TP_{Saf}, FP_{Saf}, TPR_{Saf}, PPV_{Saf} und PRO_{Saf} die Hypothese einer Normalverteilung angenommen werden kann. Für TP_{Saf} ist dies mit einem p-Wert von $p = 0,17$, für FP_{Saf} mit $p = 0,09$, für PPV_{Saf} mit $p = 0,91$ und für PRO_{Saf} mit $p = 0,12$ gegeben. Mit diesem Ergebnis kann der t-Test[2] für den Hypothesentest eingesetzt werden. Hierzu werden die Ergebnisse aus Tabelle 5.1 verwendet, um einen links- beziehungsweise rechtsseitigen

[2]Neben dem t-Test wurde ebenso ein Hypothesentest mit dem Wilcoxon-Test durchgeführt, der unabhängig von der zugrunde liegenden Verteilung ist. Dieser Test kommt dabei zum gleichen Ergebnis wie der t-Test (Anhang B.2.4).

Test der Nullhypothesen durchführen zu können [32, Seite 155]. In der Variante des Einstichproben-Tests ermöglicht dieser Test die Überprüfung, ob der Mittelwert ($\overline{x}$) einer Stichprobe signifikant von dem festgelegten Erwartungswert μ abweicht [32, Seiten 157–159],[82, Seiten 478–479]. Letzterer entspricht den Schwellwerten der Nullhypothesen, die in der zweiten Spalte von Tabelle 5.3 nochmals aufgeführt sind. Es wird somit überprüft, inwieweit eine Stichprobe die Nullhypothese erfüllt. Ist Letzteres nicht gegeben, so gilt die dazu gehörige Testhypothese (H_1) als erfüllt. Dieser Fall tritt ein, wenn das Ergebnis des t-Tests (p-Wert) kleiner als das Signifikanzniveau α ist. Im umgekehrten Falle lässt sich keine Aussage treffen. Als Signifikanzniveau wurde $\alpha = 10$ % gewählt, die Ergebnisse des Tests in Form des t- und p-Wertes sind in den Spalten 4 und 5 von Tabelle 5.3 aufgeführt.

Tabelle 5.3: Ergebnisse für den t-Test [32, Seite 155] mit einem Signifikanzniveau von $\alpha = 10$ %. Die Nullhypothesen 2 bis 4 sowie die Schwellwerte (μ) der Testhypothesen (H_1) sind aus Abschnitt 5.1 abgeleitet. Die letzte Spalte beschreibt, ob die Testhypothese akzeptiert oder abgelehnt wurde.

Nullhypothese	**Akzeptiertes μ für H_1**	**Status H_0**	**p-Wert des t-Tests für H_0**	**Status H_1**
$H_{0,Saf}^{TP}$	$> 6{,}11$	abgelehnt	0,083	akzeptiert
$H_{0,Saf}^{FP}$	$> 1{,}36$	akzeptiert	0,18	abgelehnt
$H_{0,Saf}^{PPV}$	$> 78{,}29$ %	abgelehnt	0,06	akzeptiert
$H_{0,Saf}^{PRO}$	$> 51{,}85$ %	akzeptiert	0,20	abgelehnt

Die erste Zeile von Tabelle 5.3 zeigt auf, dass die Hypothese $H_{1,Saf}^{TP}$ zur Identifikation korrekter Bedrohungen (TP) angenommen werden kann, da die entsprechende Nullhypothese durch den t-Test abgelehnt wurde. Somit lässt sich folgern, dass das Safety-Team – mit der SGM – im Schnitt eine höhere Anzahl an korrekten Bedrohungen aufdeckte als das Security-Team. Für die FP-Hypothese $H_{1,Saf}^{FP}$ ergibt sich allerdings ein anderes Bild. So muss mit einem p-Wert von 0,18 die Nullhypothese akzeptiert werden und die Hypothese abgelehnt werden, dass Nicht-Security-Ingenieure mit der SGM in der Lage sind, weniger beziehungsweise gleich viele inkorrekte Bedrohungen zu identifizieren wie Security-Ingenieure, die nicht die SGM einsetzen. Hingegen zeigen die Ergebnisse des t-Tests mit einem p-Wert von 0,06 für $H_{1,Saf}^{PPV}$, dass die dritte Hypothese angenommen werden kann. Diese sagt aus, dass Safety-Ingenieure mit der SGM in der Lage sind, mindestens die gleiche Präzision – bei der Bedrohungsidentifikation – zu erreichen, wie Security-Ingenieure. Konkret erreichte die Safety-Gruppe im Durchschnitt eine Präzision von 84,74 %, die über dem definierten Schwellwert von 78,29 % des Security-Teams liegt. Zuletzt muss die Hypothese $H_{1,Saf}^{PRO}$ abgelehnt werden. Diese nahm an, dass Safety-Ingenieure die gleiche Produktivität erreichen können wie das

Security-Team. Der durchgeführte t-Test zeigt mit einem p-Wert von 0,20, dass die Nullhypothese akzeptiert werden muss, obwohl sich der erreichte Durchschnittswert des Safety-Teams mit 57,14 % größer darstellt als der des Security-Teams.

5.4 Diskussion und kritische Auseinandersetzung

Bezogen auf die Ergebnisse des Hypothesentests in Tabelle 5.3 muss die primäre Hypothese H5.1 abgelehnt werden. So zeigt sich, dass ausschließlich die Testhypothesen 1 und 3 akzeptiert werden können. So müssen die Testhypothese 2 und 4 abgelehnt werden, da die dazugehörigen Nullhypothesen akzeptiert wurden. Ein Grund hierfür kann in der Erfahrung des Security-Teams mit Security-Analysen gesehen werden. So kann anhand der Teilnehmerbefragung in B.1.1 angenommen werden, dass die Security-Ingenieure mit Bedrohungsanalysen vertrauter waren, wodurch sie Bedrohungen schneller identifizieren und eine höhere Produktivität als das Safety-Team erreichten konnten. Ungeachtet dessen kann jedoch gesagt werden, dass Team *Safety* – mit der SGM – in der Lage war, eine höhere Anzahl an korrekten Bedrohungen aufzudecken und eine gesteigerte Präzision im Experiment zu erreichen, was als positiv zu werten ist.

Neben den positiven Tendenzen die für eine sinnvolle Anwendung der SGM sprechen, sollten die Ergebnisse allerdings kritisch diskutiert werden. So ist für den Hypothesentest mit dem t-Test ein von Signifikanzniveau $\alpha = 10$ % ausgewählt. Somit kann nur mit einer Sicherheit von 90 % gesagt werden, dass die Hypothesen zu TP, FP, PPV und PRO aus Abschnitt 5.1 bestätigt sind. Außerdem besteht eine höhere Wahrscheinlichkeit für Fehler zweiter Ordnung gegenüber einem Signifikanzniveau von $\alpha = 5$ % [183, S.158-159]. Würde Letzteres ausgewählt, um eine Sicherheit von 95 % ($\alpha = 5$ %) zu erreichen, so müssten alle Testhypothesen abgelehnt werden. So müsste Nullhypothese 1 mit einem p-Wert von 0,083 akzeptiert werden, was eine Ablehnung der dazugehörigen Testhypothese ($H^{TP}_{1,Saf}$) zur Folge hätte. In gleicher Weise müsste Nullhypothese 2 mit einem p-Wert von 0,18 akzeptiert werden und Hypothese $H^{TPR}_{1,Saf}$ abgelehnt werden. Das Gleiche gilt für Nullhypothese 3 mit einem p-Wert von 0,06 und Nullhypothese 4 mit einem p-Wert von 0,20. Zusammenfassend kann damit gesagt werden, dass bei einem Signifikanzniveau von $\alpha = 5$ % keine Aussage über die sinnvolle Anwendbarkeit der SGM getroffen werden kann. In diesem Fall sprechen alleinig die analytischen Ergebnisse und die des Flow-Erfahrens für die SGM. Aufgrund dessen sollten die Ergebnisse des Experiments als qualitativ angesehen werden.

Neben der kritischen Betrachtung der Auswahl des Signifikanzniveaus muss ebenfalls das Design des Experiments hinterfragt werden. Im Folgenden sollen dazu vier Bedrohungsklassen für die Gültigkeit des Experiments analysiert werden. Als Klassen sind die Gültigkeit des Aufbaus, der Ergebnisse sowie die interne und externe Gültigkeit

des Experiments betrachtet, die aus den Arbeiten von Wohlin et al. [210] entnommen sind.

Hinsichtlich der **Gültigkeit des Aufbaus** des Experiments besteht die Gefahr, dass nicht repräsentative Kriterien – TP, FP, PPV und PRO – zur Messung, der Anwendbarkeit der SGM eingesetzt wurden. Bei der Betrachtung anderer Studien im Umfeld hingegen konnten keine weiteren Kriterien identifiziert werden als die in der Arbeit von Scandariato et al. [176] vorgestellten. Dieser Umstand lässt sich weiter begründen, dass kein anderer betrachteter Ansatz eine empirische Studie vorstellt, was die Auswahl und Vergleichbarkeit von Metriken erschwert. So zeigt auch die umfassende Untersuchung von Bedrohungsanalysemethodiken durch die Forscher Lisova et al. [124], dass bis auf die SGM kein Ansatz empirisch evaluiert wurde.

Bezogen auf die **Gültigkeit der Ergebnisse** ist die kleine Teilnehmeranzahl ein klarer Kritikpunkt. So nahmen nur 14 Safety-Ingenieure an dem Experiment teil, was die Aussage der Signifikanz der statistischen Auswertung gefährdet. Dadurch sind die Ergebnisse des Hypothesentests gefährdet, da bei kleinen Stichprobengrößen die Teststärke des t-Tests abnimmt. Allerdings zeigte eine Analyse des t-Tests für kleine Stichproben ($N \leq 5$) durch Winter [55], dass für Stichproben mit $N = 5$ ebenso eine hohe statistische Aussagekraft erreicht werden kann. Die Forscher kamen zum Schluss, dass es keine grundsätzlichen Einwände gegen die Verwendung eines t-Tests bei kleinen Stichproben gibt [55]. Aufgrund dessen kann der durchgeführte Hypothesentest mit $N = 14$ für die Safety-Gruppe als akzeptabel angesehen werden. Winter deckte allerdings auf, dass bei einem einseitigen t-Test auf nicht normalverteilten Stichproben, eine hohe Quote von Typ 1 Fehlern [187, Seite 168] beobachtet werden kann. Diese Situation ist als relevant anzusehen, da der durchgeführte Shaprio-Wilk-Test ebenfalls von der kleinen Stichprobe beeinflusst wird und daher die Aussage auf Normalverteilung der Stichproben gefährdet ist. In folge dessen wurde eine weitere Überprüfung der Hypothesen mit dem verteilungsunabhängigen Wilcoxon-Test [82, Seite 527] vollzogen. Die dabei erreichten Ergebnisse bestätigen allerdings die des t-Tests (Anhang B.2.4). Außerdem werden in anderen Bereichen, wie beispielsweise der Pharmazie, Teilnehmerzahlen von 12 Personen pro Gruppe als ausreichend angesehen [104], was mit einer Gruppengröße von 14 Teilnehmern für Team *Safety* gegeben ist. Grundsätzlich stellt sich allerdings die Erreichung einer großen Anzahl von Teilnehmern, die täglich in diesen Themenfeldern arbeiten, als Herausforderung dar. So müssen Safety und Security-Ingenieure für die Zeit der Studie von ihren Projekten abgezogen werden, was in der Gesamtheit einen bedeutenden Kostenfaktor darstellt. Somit ist es als positiv zu bewerten, dass insgesamt 23 Ingenieure eines Automobilzulieferers gewonnen werden konnten. Außerdem zeigt die durchgeführte Hintergrundbewertung der Teilnehmer, dass alle Personen in einer Gruppe einen ähnlichen Wissensstand aufweisen und somit als gleich verteilt angesehen werden können (Anhang B.1).

Eine Gefahr, welche die **interne Gültigkeit** des Experiments betrifft, liegt in der Annahme, dass Nicht-Security-Ingenieure nur ein geringes Wissen im Studienfeld Security aufweisen und vom Experiment überfordert sind. Dies würde die sinnvolle Anwendung der SGM für Nicht-Security-Ingenieure infrage stellen. Aufgrund dieser Situation wurde – parallel zu dem Experiment – das Flow-Erleben nach Rheinberg et al. [164] gemessen. Die Auswertung legt nahe, dass sich die Safety-Ingeniuere mit der gestellten Aufgabe weniger gefordert fühlten und sich stärker im Arbeitsablauf befanden (Tabellen B.2, B.5 und B.6). Dies erlaubt die Schlussfolgerung, dass die SGM Nicht-Security-Ingenieure strukturiert durch die Analyse leitet und nicht überfordert.

Als letzte Gefahr muss die **externe Gültigkeit** des Experiments hinterfragt werden. So nahm mit Team *Unerfahren* eine Gruppe am Experiment teil, die hauptberuflich keine Analysen durchführt und mit den Konzepten und Begrifflichkeiten im Experiment überfordert sein könnte, was die Ergebnisse verfälschen würde. Insbesondere die korrekte Beschreibung von Bedrohungen könnte darunter leiden, was die Zahl der FP erhöhen würde. Allerdings zeigen die in Tabelle 5.1 erreichten Ergebnisse, dass die Gruppe in der Lage war eine hohe Anzahl korrekter Bedrohungen aufzudecken. Darüber hinaus zeigte Team *Unerfahren* bei der Hintergrundbefragung, dass die Mitglieder der Gruppe Wissen in der Fahrzeug-Domäne aufweisen und mit den relevanten Konzepten vertraut sind (Anhang B.1). Damit können die von der Gruppe erzielten Ergebnisse als repräsentativ angesehen werden. Eine weitere Bedrohung für die externe Gültigkeit ist die Möglichkeit, dass Safety-Ingenieure Security-Wissen besitzen und damit die zugrundeliegenden Annahmen des Experiments nicht erfüllen. Die Hintergrundbefragung zeigt hingegen, dass weder Team *Safety* noch Team *Unerfahren* als Security-Experten zu bewerten sind. Ebenso könnte die Komplexität des ausgewählten Beispiels einen Einfluss auf die gemessenen Ergebnisse haben. So würde ein zu einfaches Beispiel durchgehend bessere Ergebnisse erzeugen. Insofern Team *Security* und *Safety* jeweils das gleiche Beispiel erhielten, wären beide Gruppen gleichermaßen beeinflusst und ein relativer Vergleich weiterhin gültig. Ebenso zeigten Gespräche mit den Teilnehmern des Pilots, dass der Umfang des Beispiels passend war. Darüber hinaus zeigt die Auswertung des Flow-Erlebens in Anhang B.2, dass sich keine Teilnehmergruppe unterfordert fühlte. Hiermit kann angenommen werden, dass das ausgewählte Beispiel eine sinnvolle Komplexität aufwies. Als letzter Punkt für die externe Gültigkeit des Experiments muss der fehlende Vergleich mit einer Gruppe, die weder Security noch Safety-Wissen aufweist, diskutiert werden. Diese Gruppe wurde im Experiment nicht betrachtet, da Teilnehmern ohne jede Erfahrung in Safety&Security die grundlegenden Zusammenhänge nur schwer vermittelt werden können, auf denen die SGM aufbaut. Das ist unter anderem damit zu erklären, dass beide Fachbereiche komplex sind und teilweise konträre Sichtweisen aufweisen. Aufgrund dessen ist angenommen, dass Ergebnisse solch einer Gruppe schwierig zu bewerten sind. Beispielsweise könnte nicht nachvollzogen werden, ob ein schlechteres

Ergebnis der Gruppe auf fehlendes Verständnis in einem der beiden Studienfelder oder auf eine ungenügende Methodik zurückzuführen ist.

Abschließend lässt sich aus der Evaluierung der SGM annehmen, dass eine Unterstützung für Personen gegeben ist, die nur geringes Wissen im Bereich Security haben. Eine Anwendung der Methodik durch Security-Ingenieure erscheint allerdings ebenfalls sinnvoll, da sie den unterstützenden Charakter der SGM nutzen können. Als interessant zu werten ist, dass die Safety-Ingenieure im Schnitt vergleichbare Ergebnisse wie Team *Security* erbracht haben. Das ist damit zu begründen, dass Safety-Ingenieure ein hohes Systemwissen besitzen, das sie mit den SGM-Leitwörten kombinieren, um Bedrohungen aufzudecken. Diese Aussage wird von der Tatsache gestützt, dass nahezu alle Safety-Ingenieure mit der Identifikation von Bedrohungen begannen, die einen hohen Safety-Schaden auslösen können. So konnte – bei Team *Safety* – eine höhere Anzahl von Bedrohungen bezogen auf den Ausfall *ungewolltes Verriegeln* festgestellt werden. Hieraus lässt sich die Empfehlung geben, dass Safety-Ingenieure Methodenschritt 5 der SGM durchführen können, um das in der Safety-Analyse erlangte Systemwissen für die Security-Analyse wiederzuverwenden. Anschließend können die Ergebnisse dieses Schrittes an die Security-Ingenieure übergeben werden, welche dann Schritt 7 der SGM durchführen. Ein Vorteil, der sich hierbei ergibt, ist, dass das Safety-Team viele Security-Bedrohungen durch die geführte Methodik identifizieren und somit die Arbeiten des Security-Teams ergänzen kann. Damit kann das Security-Team weiterhin kreativ sein, um Bedrohungen aufzudecken.

6

SGM-Fallstudie mit einem Penetrationstest

In den bisherigen Kapiteln wurde die Anwendung der SGM für die Designphase aufgezeigt und evaluiert. Die Bedrohungsmodellierung mit der SGM lässt sich allerdings auch für eine spätere Entwicklungsphase einsetzen. Konkret kann die SGM in einen Penetrationstest integriert werden, um safety-relevante Testfälle zu identifizieren. Hierbei ist Einbettung der SGM insbesondere dann sinnvoll, wenn die SGM in der Designphase nicht angewendet wurde.

Ein Penetrationstest dient zur Feststellung, inwieweit das Zielsystem einem Angreifer widerstehen kann und soll bisher nicht aufgedeckte Schwachstellen identifizieren. Der Tester agiert aus Sicht des Angreifers, d. h. Penetrationstests werden häufig als Black-Box [85, Seite 37] Tests durchgeführt. Dies bedeutet, dass der Tester typischerweise kein Wissen über das Zielsystem besitzt und der Test damit erfahrungsbasiert ist. Aufgrund dessen empfehlen etablierte Penetrationtesting-Standards [85, 145, 147, 161], das Zielsystem auf Schwachstellen zu analysieren und Angriffe zu modellieren, was einer impliziten Bedrohungsmodellierung gleich kommt. Insbesondere fordert der Penetration Testing Execution Standard (PTES) [200] diese Modellierungsphase explizit. Aufgrund dessen wird im Folgenden ein Vorgehen zur Einbettung der SGM in den PTES vorgestellt. Darüber hinaus wird die SGM mit Angriffsbäumen erweitert, um mehrstufige Bedrohungen abbilden zu können, was einer Erweiterung der SGM entspricht. Als Ergebnis dieses Vorgehens konnte eine safety-kritische Schwachstelle in einem Airbag-System identifiziert werden, die eine Auslösung der Airbags ermöglicht.

6.1 Methodik zur Einbettung der SGM in einen Penetrationstest

Die Integration der SGM in einen Penetrationstest ist grundsätzlich in jenen Phasen möglich, bei denen Bedrohungen beziehungsweise Angriffe modelliert werden. Hierzu wurden etablierte Standards und Leitfäden für Penetrationstests mit in der Arbeit [3]

analysiert und festgestellt, dass der PTES die meisten Vorteile für eine Anwendung im Automotive Bereich aufweist. In Abbildung 6.1 sind die sieben Phasen des PTES dargestellt. Um den Umfang der Erläuterung zu reduzieren werden im Folgenden, ausschließlich jene Phasen erläutert, die für die Einbindung der SGM relevant sind.

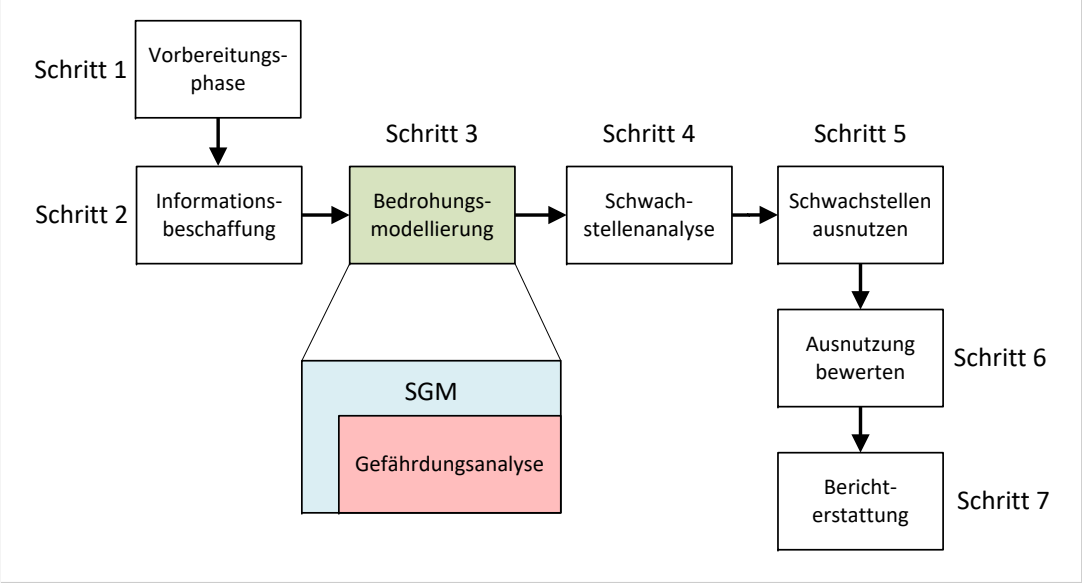


Abbildung 6.1: Dargestellt sind die sieben Phasen des PTES [200]:Vorbereitungsphase, Informationsbeschaffung, Bedrohungsmodellierung, Sachwachstellenanalyse, Schwachstellen ausnutzen, Ausnutzung bewerten und Berichterstattung. Schritt 3 zeigt die Einbindung der SGM (blaues Rechteck) mit der notwendigen Gefährdungsanalyse (rotes Rechteck).

In der zweiten Phase des PTES werden alle verfügbaren Informationen gesammelt, die auf den Analysegegenstand bezogen sind. Dieser Schritt ist einer der wichtigsten im gesamten Vorgehen, da die nachfolgenden Schritte auf den gesammelten Informationen aufbauen. In Schritt 3 werden anschließend die Bedrohungen modelliert, indem die gesammelten Informationen verwendet werden. Der PTES [200, Seiten 37–44] zeigt hierzu allerdings keine konkrete Methodik auf, weshalb an dieser Stelle die SGM eingebettet wird. Ziel der Integration ist es, den Prozess der Bedrohungsmodellierung zu strukturieren und Artefakte aus der linken Seite des V-Modells wiederzuverwenden. Dies betrifft die Ergebnisse der Safety-Analyse sowie relevante Informationen über das Testobjekt, wie dessen Funktionalität und die E/E-Architektur. In Abbildung 6.1 ist diese Erweiterung durch die blauen und roten Rechtecke dargestellt, wobei die SGM von einer existenten Gefährdungsanalyse ausgeht (Gefährdungsanalyse → SGM).

Das detaillierte Vorgehen zur Modellierung der Bedrohungen und der Ableitung von Testfällen zeigt Abbildung 6.2. So muss zu Beginn geprüft werden, ob eine Gefährdungsanalyse zur Verfügung steht. Ist dies nicht der Fall, so muss diese durchgeführt werden, um die notwendigen Safety-Artefakte für die SGM bereitstellen zu können, was mit Schritt 3.1 in Abbildung 6.2 beschrieben ist. Schritt 3.2 zeigt hingegen die

Bedrohungsmodellierung mit der SGM, die identisch zum Vorgehen in Abschnitt 4.3 ist.

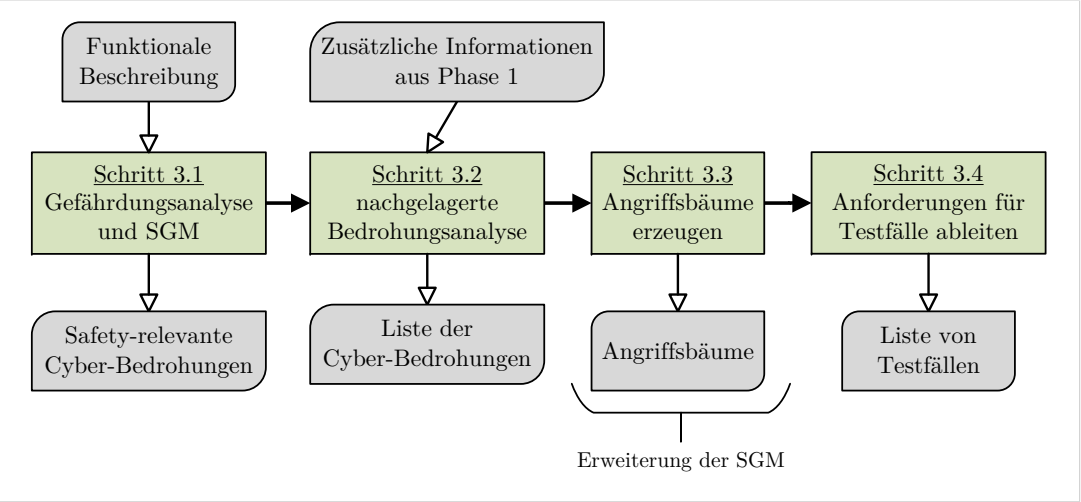


Abbildung 6.2: Detailliertes Vorgehen zur Bedrohungsmodellierung mit der SGM für Schritt 3 des PTES. Schritt 3.3 repräsentiert die Erweiterung der SGM mit Angriffsbäumen, um Anforderungen für Testfälle ableiten zu können.

Nach Abschluss der Bedrohungsidentifikation werden Angriffsbäume erzeugt, um mögliche Angriffspfade aufzudecken, welche die Bedrohungen auslösen können (Schritt 3.3). Diese Bäume dienen anschließend zum Ableiten von Anforderungen für die Testfälle im letzten Schritt (Schritt 3.4). Wie solch ein Angriffsbaum aufgebaut ist und in die SGM integriert wird, zeigt der nachfolgende Abschnitt.

6.1.1 Erweiterung der SGM mit Angriffsbäumen

Durch die Integration von Angriffsbäumen in die ursprüngliche Version der SGM, ist es mit der Methodik möglich, mehrstufige Angriffe zu identifizieren. Hierzu wird die Baumstruktur in Abbildung 6.3 verwendet. Die Wurzel entspricht dabei einer – mit der SGM – identifizierten Bedrohung. Die unter der Wurzel liegenden Zwischenknoten werden als Angriffsschritte interpretiert und beschreiben jene Teilschritte, die ein Angreifer tätigen muss, um die Bedrohung realisieren zu können. In den Blättern wird hingegen eine Verletzung der zugehörigen Schutzziele (CIAA) aufgeführt. Ein Beispiel hierfür ist die Verletzung der Integrität einer Diagnosebotschaft. Die Blätter können somit als die atomaren Angriffsschritte gesehen werden, die sich nicht weiter auftrennen lassen. Der Bogen zwischen den Knoten unter der Wurzel repräsentiert eine *UND* und die restlichen Kanten eine *ODER*-Verknüpfung (Abschnitt 2.5.2). Diese dienen im späteren Verlauf dazu, sinnvoll Risikowerte von den Blättern zur Wurzel propagieren zu können.

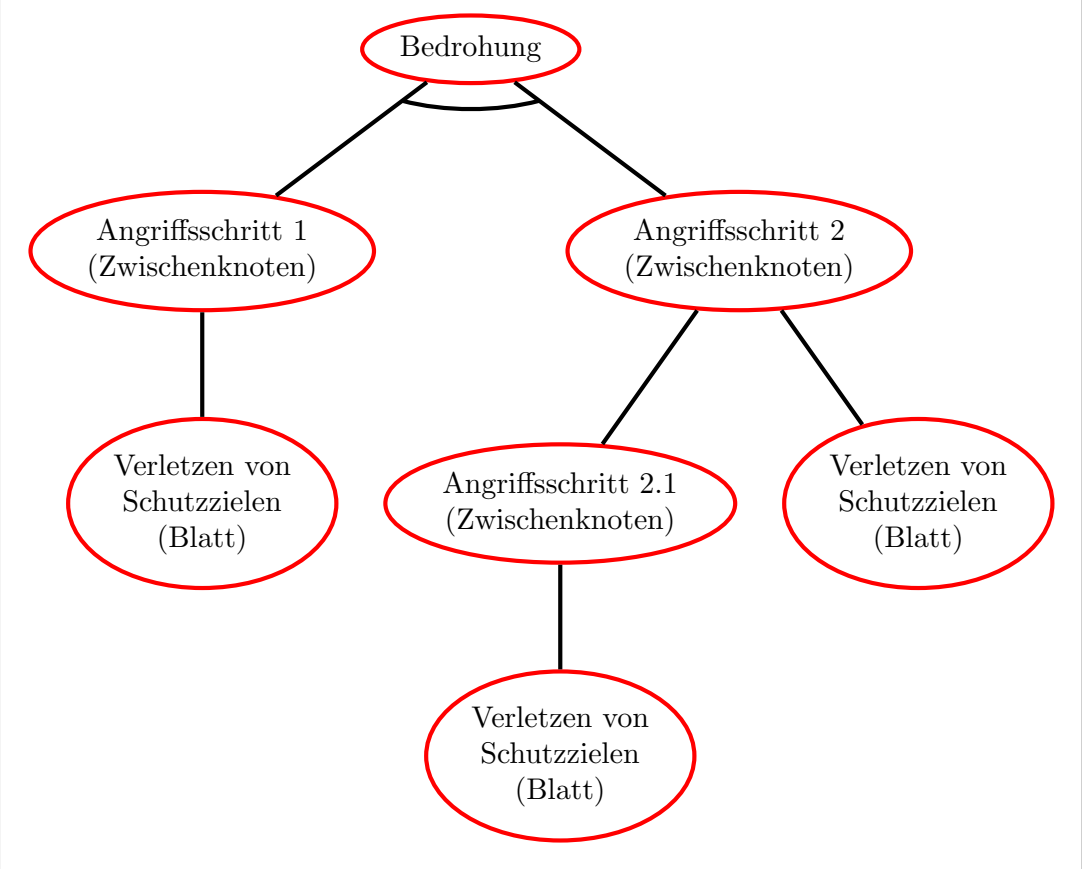


Abbildung 6.3: Generischer Angriffsbaum, bei dem die Baumwurzel der Bedrohung entspricht, die Zwischenknoten den Angriffsschritten und die Blätter den verletzten IT-Vermögenswerten bezogen auf CIAA. Der Bogen zwischen den beiden ausgehenden Linien der Wurzel entspricht einer *UND*-Verknüpfung und eine fehlende Linie einer *ODER*-Verknüpfung.

Basierend auf den verletzten Schutzzielen (CIAA) werden in Schritt 3.4 Anforderungen für mögliche Testfälle abgeleitet. Würde beispielsweise Angriffsschritt 1 dem Auslösen der ESCL-Verriegelung durch eine Diagnosebotschaft entsprechen und das dazu gehörige Blatt aufzeigen, dass die Authentizität der Diagnosebotschaft *Verriegelung* verletzt werden muss, ergäbe sich der Testfall zu: *Die Authentizität der Diagnosebotschaft « Verriegelung » sollte verletzt werden, um das Sperren des Lenkrades auszulösen.* Wie genau dies erreicht werden kann, hängt vom Zielsystem und dessen Komponenten ab. Aufgrund dessen muss ein Tester die passenden Angriffstechniken, basierend auf den vorliegenden Informationen, auswählen. Bei dem hier genannten Beispiel wäre das beispielsweise ein Wiedereinspielen der Diagnosebotschaft « Airbag-Detonation » auf dem CAN-Bus.

6.2 Exemplarische Anwendung auf ein Airbag-System

Nach Erläuterung der Methodik zur Einbettung der SGM in den PTES, beschreibt dieser Abschnitt, wie der Test an einem realen Fahrzeug durchgeführt wird. Als Analy-

segegenstand ist ein Airbag-Steuergerät eines deutschen Premium-Fahrzeugherstellers ausgewählt.

Für den Penetrationstest standen zwei identische Steuergeräte zur Verfügung. Das erste befand sich in einem Serienfahrzeug und das zweite in einem Prüfstand, der hierfür entwickelt wurde und in Abschnitt 6.2.4 näher erläutert wird. Um keine Beeinflussung der Testfälle und eine Reproduzierbarkeit zu erreichen, wurde vor jeder Durchführung eines Testfalls das Steuergerät zurückgesetzt und dessen Fehlerspeicher gelöscht.

Der Ablauf des Penetrationstests folgt den Methodenschritten in Abbildung 6.1 und der detaillierten Sequenz für die Bedrohungsmodellierung mit der SGM in Abbildung 6.2. Alle identifizierten Testfälle wurden in erster Instanz am Prüfstand ausgeführt und bei erfolgreichem Ausgang am realen Fahrzeug validiert.

6.2.1 Informationen sammeln

Nach Abbildung 6.1 werden im ersten Schritt der PTES-Methodik verwertbare Informationen aus den verfügbaren Quellen gesammelt. Hierzu wurden die grundlegenden Funktionalitäten des Airbag-Systems in Erfahrung gebracht. Neben technischen Schriftstücken und wissenschaftlichen Publikationen wie [20] wurden ebenfalls Webauftritte von Unternehmen analysiert, die mit Airbag-Systemen für OEMs werben. Dies ermöglichte es, die grundsätzlich eingesetzten Verbindungstechnologien aufzudecken. Mittels Reverse-Engineering und dem im Fahrzeug verbauten Steuergerät war es anschließend möglich, die Bussysteme, angebundene Sensoren und Aktoren sowie die Botschaften der Kommunikation zu identifizieren. Dies ermöglichte es, die E/E-Architektur in Abbildung 6.4 zu rekonstruieren.

Die Airbag-ECU ist in diesem Fall über CAN-1 mit dem zentralen Gateway verbunden. Außerdem befinden sich die ESP-ECU und das Steuergerät für die Gurtstraffer (*ECU Belt Tensioners*) am gleichen CAN-Bus, was eine Kommunikation zwischen diesen drei Steuergeräten ermöglicht. Das Airbag-Steuergerät bezieht des Weiteren Informationen über einen potenziellen Unfall des Fahrzeugs durch die Einschlagsensoren (*Sensor Impact*). Diese messen die Stärke des Einschlags des Fahrzeugs und liefern damit die Messwerte, welche die Airbag-ECU auswertet. Basierend darauf entscheidet das Steuergerät, ob eine Zündung der Airbags notwendig ist. Sowohl die Zündkapseln (*Actuator Pyrotechnic Charge*) als auch die Einschlagsensoren sind über eine direkte Verbindung (*P2P*) mit dem Steuergerät verbunden.

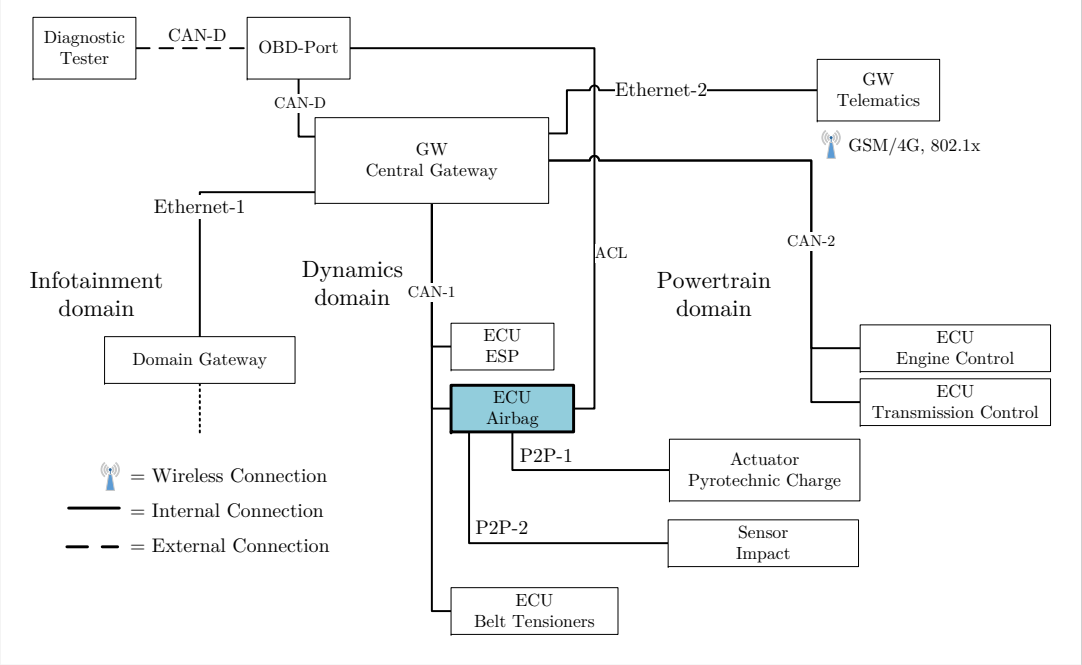


Abbildung 6.4: E/E-Architektur des Airbag-Systems, die mit einer zusätzlichen Kommunikationsverbindung (ACL) zwischen dem OBD-Port und dem Airbag-Steuergerät erweitert ist. Die Airbag-ECU erhält vom Einschlagsensor (*Sensor Impact*), über eine direkte Verbindung (*P2P*) Sensorwerte über die Stärke des Einschlags bei einem Unfall. Anhand derer entscheidet die Airbag-ECU, ob die Sprengkapseln der Airbag-Säcke (*Actuator Pyrotechnic Charge*) gezündet werden sollen.

Neben diesen Informationen wurden zusätzlich bekannte Standards analysiert, die im Kontext von Airbag-Systemen stehen. Zu nennen sind hier der ISO Standard 14229 [98] für Diagnose nach Unified Diagnostic Services (UDS) über CAN sowie die ISO 26021 [95]. Letzt genannter beschreibt den Ablauf zum Auslösen der Airbags, sollte das Fahrzeug das Ende seines Lebenszyklus erreicht haben und ausgemustert werden. Dieser Vorgang ist relevant, da die Sprengkapseln während der Ausmusterung detonieren können. Aufgrund dessen müssen diese zur Explosion gebracht werden, bevor die Ausmusterung durchgeführt werden kann. Der Vorgang hierzu wird in der ISO 26021 [95] detailliert beschrieben. Auf Basis dieses Standards konnte eine zusätzliche Kommunikationsverbindung (Additional Communication Line (ACL)) zwischen dem Steuergerät und dem OBD-Port identifiziert werden (Abbildung 6.4). Diese soll Herstellern dazu dienen, die Sequenz für die Sprengung an eigene Bedürfnisse anzupassen [95, Seiten 5, 7]. Bei den in dieser Arbeit analysierten Airbag-Systemen wurde allerdings die ACL von den Herstellern nicht verwendet.

6.2.2 Bedrohungsmodellierung

Nach der Analyse der erhobenen Informationen wurde die Bedrohungsmodellierung mit der SGM anhand der Schritte in Abbildung 6.2 durchgeführt. Aufgrund der Tatsache, dass für das Airbag-System keine Gefährdungsanalyse zur Verfügung stand, musste diese durchgeführt werden (Schritt 3.1 in Abbildung 6.2). Hierzu wurde die primäre

Funktionalität des Airbag-Systems bestimmt, was in Tabelle 6.1 dargestellt ist. Diese entspricht dem Auslösen der Airbags, sollte ein Unfall erkannt werden. Dies wird durch Einschlagsensoren im Fahrzeug-Chassis bestimmt, welche die Stärke des Einschlags des Fahrzeugs messen.

Tabelle 6.1: Auszug der funktionalen Anforderungen an das Airbag-System.

R-ID	Beschreibung der Anforderung
1	Das Airbag-System muss die Sprengkapseln zünden, wenn ein Unfall festgestellt wurde.
...	...

Basierend auf den funktionalen Anforderungen in Tabelle 6.1 wurde die Gefährdungsanalyse mit HAZOP durchgeführt. Hierzu wurden die Leitwörter *ungewollt*, *kein*, *zu spät* und *zu früh* verwendet, um die Gefährdungen in Tabelle 6.2 bestimmen zu können.

Tabelle 6.2: Auszug der Ergebnisse der Gefährdungsanalyse mit HAZOP für das Airbagsystem.

G-ID	Gefährdung
1	Ungewolltes Zünden der Airbags
2	Kein Zünden der Airbags
3	Zu spätes Zünden der Airbags
4	Zu frühes Zünden der Airbags
...	...

Obwohl die Gefährdungen 1 bis 4 grundsätzlich als safety-relevant einzustufen sind, wird nachfolgend ausschließlich Gefährdung 1 betrachtet. Diese Entscheidung ist mit dem zeitlichen Aufwand zu begründen, der notwendig gewesen wäre, alle Testfälle durchzuführen. Die ausgewählte Gefährdung eröffnet außerdem die Möglichkeit, Airbags zu zünden, ohne dass ein Unfall vorliegt. Gefährdung 2 entspricht dem Ausfall des Airbagsystems bei einem Unfall. Gleichermaßen sind auch die Gefährdungen 3 und 4 auf eine Unfallsituation bezogen. Hier besteht die Gefährdung, dass die Airbags zu früh oder zu spät ausgelöst werden. Damit zeigt sich, dass die Gefährdungen 2 bis 4 nur im Falle eines Unfalls relevant sind, Gefährdung 1 kann hingegen jederzeit auftreten und zu einem Unfall führen. Aufgrund dessen lässt sich Gefährdung 1 bereits mit dem Risikowert von ASIL-D bewerten, da ein hoher Schaden (S=3) zu erwarten ist, die Fahrsituation (C=3) für den Fahrer nicht kontrollierbar ist und die Gefährdung in jeder Betriebssituation (E=4) auftreten kann.

Nachdem die Gefährdungen identifiziert sind, kann die Bedrohungsanalyse mit der SGM erfolgen. Die in der SGM identifizierten Bedrohungen sind in Tabelle 6.3 aufgeführt.

Tabelle 6.3: Ergebnisse der Bedrohungsanalyse des Airbagsystems mit der SGM-Vorlage sowie den Ausfällen aus Tabelle 6.2. G-ID=1 entspricht der Gefährdung *ungewolltes Zünden der Airbags* aus Tabelle 6.2.

B-ID	G-ID	kann ausgelöst werden durch	Signal oder (Diagnose-) Funktion	für Komponente oder Teilsystem	Eintrittspunkt
1	1	auslösen	Zündfunktion	Airbag-ECU	CAN-1, ACL
2	1	manipulieren	Diagnosebotschaft *Airbag zünden*	Airbag-ECU	CAN-D
3	1	manipulieren	Unfall-Botschaft	Airbag-ECU	P2P-2, Sensor Impact
...	...	...	...	...	...

Anschließend wird in Schritt 3.3 ein Angriffsbaum aufgestellt, der die Bedrohungen weiter auflöst. Für die übergeordnete Gefährdung « des ungewollten Zündens der Airbags » (G-ID=1) zeigt Abbildung 6.5 den erzeugten Baum. Dieser fasst die Bedrohungen B-ID=1 bis 3 zusammen, welche die identische Gefährdung auslösen können (G-ID= 1). Außerdem bilden die Einträge der Spalten 3 bis 6 von Tabelle 6.3 die erste Knotenebene des Baumes.

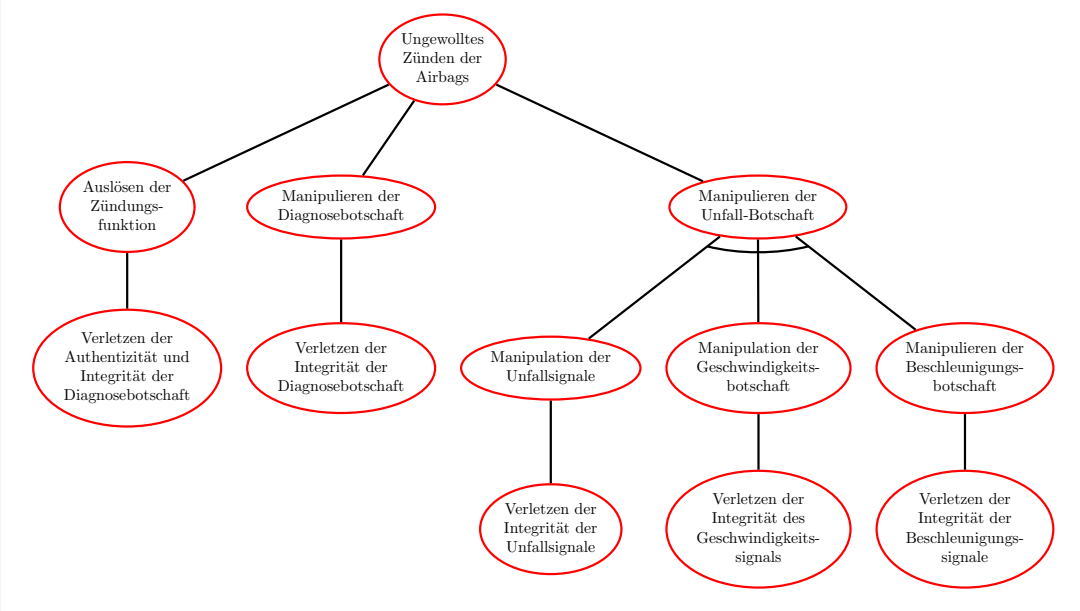


Abbildung 6.5: Modellierter Angriffsbaum für das Airbag-System, der auf den identifizierten Bedrohungen von Tabelle 6.3 basiert.

Die darunter folgende Knotenebene kann nur teilweise aus Tabelle 6.3 abgeleitet werden. Für den Fall, dass nach der ersten Knotenebene keine weitere Auftrennung des Angriffsschrittes möglich ist, kann das dazugehörige Blatt aus dem verwendeten Leitwort bestimmt werden. Für den linken Pfad in Abbildung 6.5 und den Knoten

Auslösen der Zündungsfunktion lässt sich das Blatt *Verletzen der Authentizität und Integrität der Diagnosebotschaft* anhand des Leitworts *Auslösen* und der Zuordnung in Tabelle 4.1 ableiten.

Für die Knotenebenen des Teilbaums *Manipulation der Unfall-Botschaft*, konnte hingegen eine weitere Auftrennung identifiziert werden, die allerdings nicht direkt aus der SGM-Tabelle abgeleitet ist. Diese ergab sich durch eine Analyse der bekannten Funktionalitäten des Airbag-Systems. So fordert die gezeigte *UND*-Verknüpfung, dass sowohl die Unfall- als auch die Geschwindigkeits- und Beschleunigungsbotschaft manipuliert werden müssen. Letzteres entspricht den Botschaften des Beschleunigungssensors, der im Airbag-Steuergerät verbaut ist und anhand einer negativen Beschleunigung einen möglichen Unfall mit den Signalen der Einschlagsensoren plausibilisiert.

Somit lässt sich an dieser Stelle bereits abschätzen, wie komplex ein Testfall (Angriff) ist. Das Manipulieren der Signale des Beschleunigungssensors ist deutlich aufwendiger, da der Sensor nicht physisch zugänglich ist. Die beiden linken Pfade in Abbildung 6.5 basieren hingegen auf der Manipulation von Bus-Nachrichten, die zugänglich sind. Basierend auf diesem Vorgehen können Testfälle bei der Durchführung priorisiert werden. Zur Ableitung dieser wird jeder Pfad im Angriffsbaum durchlaufen und ein einzelner Testfall bestimmt (Schritt 3.4 in Abbildung 6.2). Ein Auszug des Vorgehens ist mit Tabelle 6.4 beschrieben.

Tabelle 6.4: Auszug der abgeleiteten Anforderungen von Testfällen für das Airbagsystem.

TF-ID	Anforderung an den Testfall
1	Die Authentizität und Integrität der für die Auslösung verwendeten Diagnose-Botschaft muss verletzt werden.
2	Die Nutzdaten von Botschaften, die für Diagnosesitzungen eingesetzt werden, sind zu manipulieren.
3	Die Unfall-Botschaften oder Signale müssen manipuliert werden, um eine Unfall-Situation vorzutäuschen.
...	...

Der erste Testfall (TF-ID=1) steht für das Wiedereinspielen einer aufgezeichneten Diagnose-Botschaft und fällt damit in die Kategorie eines Replay-Angriffs [88]. Der zweite beschreibt hingegen die Manipulation der Nutzdaten der Diagnose-Botschaft, um die Airbag-Zündung auszulösen, was einem Man-in-the-Middle (MITM) Angriff entspricht. Ähnlich beschreibt Testfall 3 die Manipulation von Signalen und Botschaften, um den Zustand eines Unfalls vorzutäuschen, der ebenfalls zu einer Zündung der Airbags führt.

6.2.3 Schwachstellenanalyse

Mittels der im vorherigen Abschnitt identifizierten Testfälle und den gesammelten Informationen konnten die Testfälle konkretisiert werden. Hierbei zeigt die Analyse der oben genannten Standards, dass durch Diagnose-Botschaften – über den OBD-Port – eine Auslösung der Airbags möglich ist. Hierzu muss allerdings eine Autorisierung am Steuergerät erfolgen, um die Zündfunktion freischalten zu können. Der dazu notwendige Ablauf folgt dem ISO 14229:2013 [98] und ist in Anhang C.1 detailliert erläutert. Bei der weiteren Analyse dieses Autorisierungsmechanismus konnte festgestellt werden, dass es sich um ein Seed and Key-Verfahren handelt (vgl. Anhang C.1). Wird der Key richtig berechnet, ist die anfragende Entität autorisiert und kann die Airbag-Zündung ausführen [95].

Bei der Analyse für den Wertebereich der Seeds fiel zudem auf, dass ausschließlich 2 Bytes für die Bildung der Zahlen (Seed/Key) vorgesehen sind. Außerdem zeigte sich, dass sich die Länge des Keys äquivalent zur Länge des Seeds verhält. Das bedeutet, dass auf den empfangen und 2 Byte langen Seed mit einem 2 Byte langen Key geantwortet werden muss. Somit ergibt sich für 2^{16} unterschiedliche Seeds mit jeweils 2^{16} verschiedenen Keys ein Suchraum von $2^{16}\ 2^{16} = 4294967296$ Kombinationen. Mit der Möglichkeit, alle 10 ms einen Seed zu erhalten und mit einem Key zu antworten, können alle Kombinationen in ca. 12 h getestet werden. Unglücklicherweise sieht der Standard jedoch vor, dass das erste Byte des Seeds die Versionsnummer des Standards beinhalten soll, was Abbildung 6.6 aufzeigt.

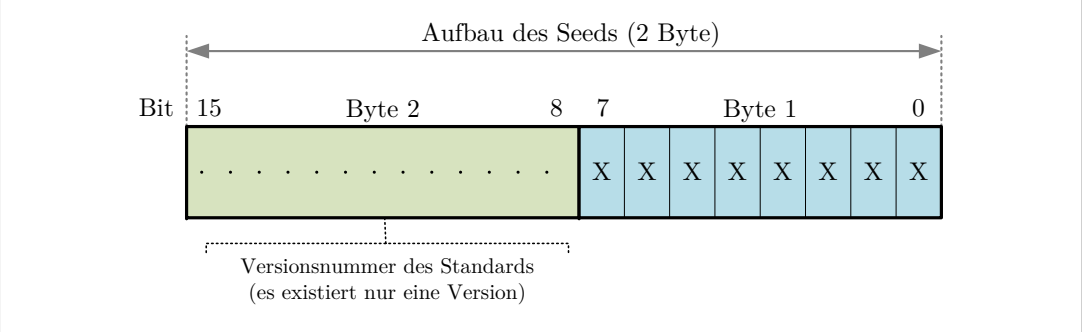


Abbildung 6.6: Aufbau des 2 Byte langen Seeds, das in Byte 2 die Versionsnummer beinhaltet. Da nur eine Version des Standards existiert, variiert Byte 2 nicht und nur Byte 1 ist für die Bildung des Seeds relevant

Es existiert allerdings nur eine Version des Standards, sodass Byte 2 vorbestimmt ist und nicht variiert (Abbildung 6.6). Damit reduziert sich die Anzahl der möglichen Seeds von 2^{16} auf $2^8 = 256$ und damit der Suchraum auf $2^8\ 2^{16} = 16777216$ Kombinationen, was den Brute-Force-Angriff vereinfacht. Durch das fehlende Sperren des Airbag-Systems bei mehreren Fehlversuchen, kann der Suchraum außerdem ungehindert durchlaufen werden, was den Brute-Force-Angriff praktikabel macht.

Eine Analyse zahlreicher Seed-und-Key Paare unterschiedlicher Hersteller zeigte außerdem, dass die Berechnungsvorschrift für die Key-Berechnung stets identisch war. Konkret ist das Einerkomplement als Vorschrift gewählt, sodass eine bitweise Negation des Seeds zur Bildung des Keys ausreichend ist. Diese einfache Berechnung bietet wenig Sicherheit und da in der Geheimhaltung dieser Vorschrift die Sicherheit des Verfahrens liegt, ist dies umso kritischer. Damit verletzt die implementierte Seed und Key Berechnung das Kerckhoffs'sche Prinzip [110]. So fordert das Prinzip, dass ein Mechanismus auch dann als sicher gelten muss, wenn dem Angreifer der Berechnungsalgorithmus bekannt ist. Die Sicherheit des Mechanismus beruht dabei ausschließlich auf der Geheimhaltung der Schlüssel. Dieses Prinzip ist an dieser Stelle verletzt, da die Berechnungsvorschrift dem Geheimnis entspricht.

6.2.4 Schwachstellen ausnutzen

Für den Nachweis der modellierten Bedrohungen müssen die dazugehörigen Schwachstellen ausgenutzt werden können. Hierzu wurde ein Prüfstand entwickelt, der eine automatisierte Testfallausführung erlaubt. Hierzu wurde auf dem in Abbildung 6.7 schematisch dargestellten Personal Computer (PC) eine Linux-Distribution eingesetzt, die über einen USB-zu-CAN Adapter mit dem Steuergerät kommuniziert und die Testfälle ausführt. Das Airbag-Steuergerät wurde ohne die dazugehörigen Sprengkapseln in den Prüfstand integriert, um möglichen Verletzungen vorzubeugen. Zur Feststellung der Airbag-Auslösung wurde stattdessen der Zündimpuls für die Sprengkapseln detektiert. Hierzu wurde eine elektrische Schaltung entworfen, die mittels eines Mikroprozessors die Spannungsspitze des Zündimpulses für jede Ladung erkennen kann. Die *Zünddetektion* kommuniziert außerdem mit dem PC, um die erfolgreiche Zündung der Airbag-Ladung aufzeichnen zu können.

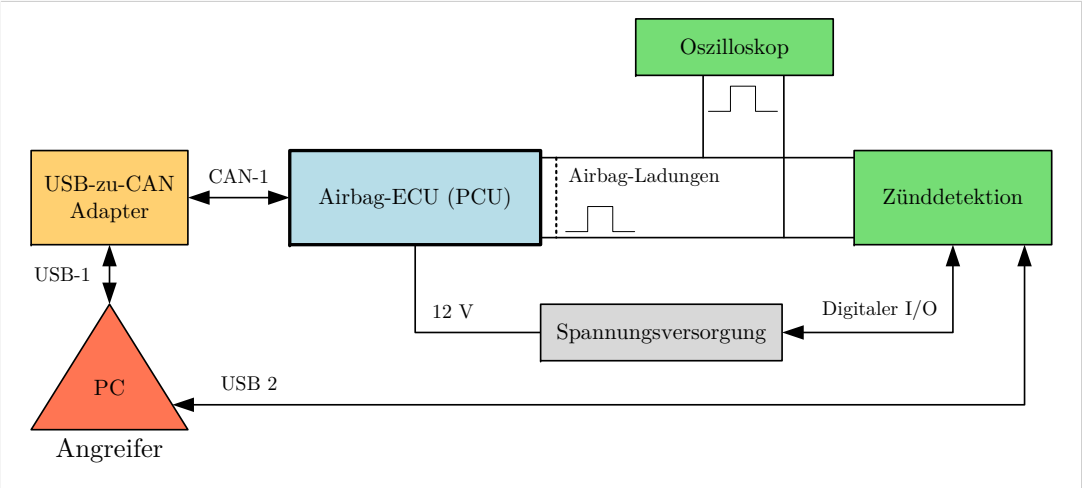


Abbildung 6.7: Prüfstand für den Penetrationstest des Airbag-Steuergerätes (PCU) aus Abbildung 6.4. Gezeigt ist der Angreifer mit einem PC und Adapter für CAN zur Kommunikation mit dem Steuergerät. Die grünen Rechtecke zeigen jene Elemente die zur Detektion der Zündimpulse für die Airbag-Ladungen der Pyrotechnical Control Unit (PCU) dienen. Das graue Rechteck repräsentiert, die digital-steuerbare Stromversorgung, um den Betriebszustand der ECU kontrollieren zu können.

Das Airbag-Steuergerät selbst – auch Pyrotechnical Control Unit (PCU) genannt – wird über ein 12V-Netzteil versorgt, das sich über den PC und die Zünddetektion steuern lässt. Letzteres ist sinnvoll, um das Steuergerät kontrolliert stromlos machen zu können. Dies ermöglicht das Rücksetzen des Steuergerätes, bevor ein neuer Testfall durchgeführt wird, indem die Stromversorgung unterbrochen wir. Diese Fähigkeit ist außerdem von Bedeutung, sollten Passwörter gebrochen werden müssen. So kann nach einer bestimmen Anzahl falscher Passwortanfragen das Steuergerät zurückgesetzt werden. Dies verhindert ein mögliches Sperren der ECU, was das Steuergerät unbrauchbar machen würde. Im konkreten Fall war dies allerdings nicht notwendig, da das Airbag-Steuergerät keine Sperr-Richtlinie einsetzte.

Mit dem Prüfstand als Fundament wurden die oben genannten Testfälle realisiert. Hierbei wurde bei Testfall 1 der Autorisierungsmechanismus durch einen Brute-Force-Angriff gebrochen und das Auslösen der Sprengkapseln ermöglicht. Für den detaillierten Ablauf des Zündens der Sprengkapsel ist auf Anhang C.2 sowie auf die Publikation [3] verwiesen. Testfall 2 wurde ebenfalls aufgeführt, indem alle möglichen binären Kombinationen des Nutzdatenfeldes durchprobiert wurden. Der Testfall führte allerdings zu keinem Auslösen der Airbags, da für eine erfolgreiche Auslösung eine gewisse Sequenz an hintereinander folgenden Botschaften notwendig war (Anhang C.2). Testfall 3 wurde hingegen nicht umgesetzt, da dieser physische Manipulationen am Airbag-Steuergerät voraussetzt, konkret das Manipulieren der Beschleunigungssignale auf den Leiterbahnen des Airbag-Steuergerätes. Aufgrund dessen wurde nach erfolgreichem Ausnutzen von Testfall 1 entschieden, das Risiko für ein eventuelles Zerstören der PCU durch Testfall 3 nicht zu akzeptieren.

An dieser Stelle sei außerdem angemerkt, dass im realen Fahrzeug niemals die Auslösung der Airbags selbst durchgeführt wurde, sondern ausschließlich die Teilangriffe, die diese Möglichkeit eröffnen können. Das Auslösen der Airbag-Ladung wurde daher ausschließlich am Prüfstand durchgeführt.

6.2.5 Berichterstattung

Die aufgedeckte Schwachstelle ermöglicht es – ohne, dass ein Unfall vorliegt – die Airbags gezielt zu zünden. Hierzu muss lediglich ein Zugriff auf jenem CAN-Bus bestehen, an dem das Airbag-Steuergerät angeschlossen ist. Dies kann neben dem Kompromittieren eines Steuergerätes am gleichen Bus ebenfalls über den OBD-Port erfolgen, da dies vom ISO Standard 26021 [95] gefordert ist. Aufgrund des häufigen Vorkommens der Schwachstelle in unterschiedlichen Fahrzeugen, wurde ein Zufall ausgeschlossen. Die Lösung dieses Phänomens ergab sich bei einer erneuten Betrachtung des ISO Standards 26021 [95]. In diesem wird für den Seed-und-Key Algorithmus das Einerkomplement als Beispielimplementierung vorgeschlagen. Dieser Vorschlag wurde von den Herstellern übernommen, sodass die Schwachstelle in zahlreichen Fahrzeugen aufgefunden werden konnte. Dementsprechend kann von einer skalierenden Schwachstelle gesprochen werden, die über mehrere Fahrzeuge hinweg existiert. Die Schwachstelle wurde am 10.04.2017 dem Bundesamt für Sicherheit in der Informationstechnik (BSI), am 13.04.2017 der zuständigen Abteilung der ISO und am 08.06.2017 dem Verband der Automobilindustrie (VDA) gemeldet. Am 25.10.2017 wurde die Schwachstelle in die Common Vulnerabilities and Exposures Enumeration (CVE)-Datenbank, mit dem Eintrag CVE-2017-14937 [52], aufgenommen.

6.3 Diskussion und kritische Auseinandersetzung

Die beispielhafte Anwendung hat gezeigt, dass die SGM ein strukturiertes Penetration Testing unterstützen kann. Sollten die Ergebnisse einer Gefährdungsanalyse bereitgestellt werden, so können durch deren Wiederverwendung Synergieeffekte ausgenutzt werden, was den Aufwand beim Testen von safety-relevanten Funktionen verringert. Mit Hinblick auf das V-Modell, bei dem die Gefährdungsanalyse vor dem Abnahmetest durchgeführt wird, kann von existierenden Safety-Artefakten ausgegangen werden [7]. Dies kann im Allgemeinen für einen OEM angenommen werden. Bestehen allerdings keine Safety-Artefakte, kann eine HAZOP basierend auf den gesammelten Informationen durchgeführt werden. Durch die Wiederwendung der Gefährdungen und Ausfälle aus der Gefährdungsanalyse, verlagert sich der Penetrationstest vom Black-Box zum Gray-Box Szenario. Dies ist grundsätzlich wünschenswert, da dem Tester umfangreichere Informationen zur Verfügung stehen und der Test damit gezielter durchgeführt werden kann.

Hinsichtlich der sinnvollen Anwendung der SGM als Bedrohungsanalysetechnik kann mit den abgeleiteten Testfällen und der aufgedeckten Schwachstelle argumentiert

werden. So zeigt die Einbettung von Angriffsbäumen die Möglichkeit der gezielten Ableitung von Testfällen, mit der Fähigkeit diese zu priorisieren. Außerdem zeigt die Übertragung des SGM-Ansatzes in eine weitere Phase des Entwicklungsprozesses, dass sich die SGM in bestehende Methoden einbetten lässt, die eine Bedrohungsmodellierung erfordern.

Obwohl auf den ersten Blick der Aufwand durch die Einbettung der SGM steigt, sinkt der Aufwand gesamtheitlich für den Penetrationstest. Unabhängig von der gewählten Methodik für einen Penetrationstest wird eine explizite oder implizite Bedrohungsmodellierung benötigt, um Testfälle ableiten zu können. Durch das strukturierte Vorgehen mit der SGM wird somit die Bedrohungsmodellierung beschleunigt, und die erzielten Ergebnisse sind vollständiger (vgl. Abschnitt 5.2.1). Das manuelle Auffinden der Angriffspfade in den Angriffsbäumen ist allerdings sehr zeitintensiv. So ist das Auffinden und Verknüpfen von Schwachstellen hochkomplex und erfordert viel Erfahrung bei der Festlegung von Angriffspfaden. Eine automatisierte Erzeugung dieser Pfade könnte den Aufwand deutlich verringern. Außerdem könnten durch Einbezug von bekannten Schwachstellen mehr Angriffspfade identifiziert werden, was genutzt werden kann, um ein System resistenter gegen Cyber-Angriffe zu gestalten. Aufgrund dessen und der im Abschnitt 4.4 genannten Möglichkeit, über Angriffspfade Eintrittswahrscheinlichkeiten für die Risikobewertung der SGM zu berechnen, zeigt das nächste Kapitel einen Ansatz zur automatisierten Erzeugung von Angriffspfaden.

7

Erweiterungsansatz für die Risikobestimmung mit der SGM

Die Ausführungen im vorherigen Kapitel zeigen auf, wie die SGM – mittels Angriffsbäumen – genutzt werden kann, um mehrstufige Angriffe zu modellieren. Wie dort gezeigt, führt die Modellierung der Angriffe zu genaueren Ergebnissen und sichereren Systemen, allerdings auch zu einem erhöhten Arbeitsaufwand. In diesem Kapitel wird ein Ansatz präsentiert, mit dem die Modellierung und Risikobewertung formalisiert und automatisiert werden kann, was den Arbeitsaufwand maßgeblich reduziert. Konkret wird aufgezeigt, wie mehrstufige Angriffe durch das entwickelte Software-Framework (ASTMT) automatisiert modelliert und bewertet werden können.

7.1 Ansatz

Die Basis zur automatisierten Modellierung mehrstufiger Angriffe mit dem ASTMT, bilden die Inhalte der SGM-Tabelle aus Abschnitt 4.3 und die E/E-Architektur des zu untersuchenden Fahrzeugs (Abbildung 7.1). Das erste Eingangsartefakt ist die SGM-Tabelle, aus der die identifizierten Bedrohungen, Leitwörter, Zielkomponenten und Eintrittspunkte des Angreifers übernommen werden. Die Bedrohungen werden für die Erstellung des Berichts benötigt. Die Leitwörter, Zielkomponenten und Eintrittspunkte werden für das formale Systemmodell verwendet, das der Model-Generator erzeugt. Ebenso benötigt der Model-Generator die E/E-Architektur und die darauf anwendbaren Schwachstellen. Hierbei wird die E/E-Architektur – mit dem grafischen Editor des ASTMTs – erzeugt und in Form eines bidirektionalen Grafen an den Modellgenerator übergeben. Dieser appliziert ebenso die Schwachstellen, die für das jeweilige E/E-Architekturelement passend sind. Als letztes Eingangsartefakt benötigt das ASTMT ein Rechtemodell, um das sukzessive Ausnutzen von Schwachstellen durch den Angreifer modellieren zu können.

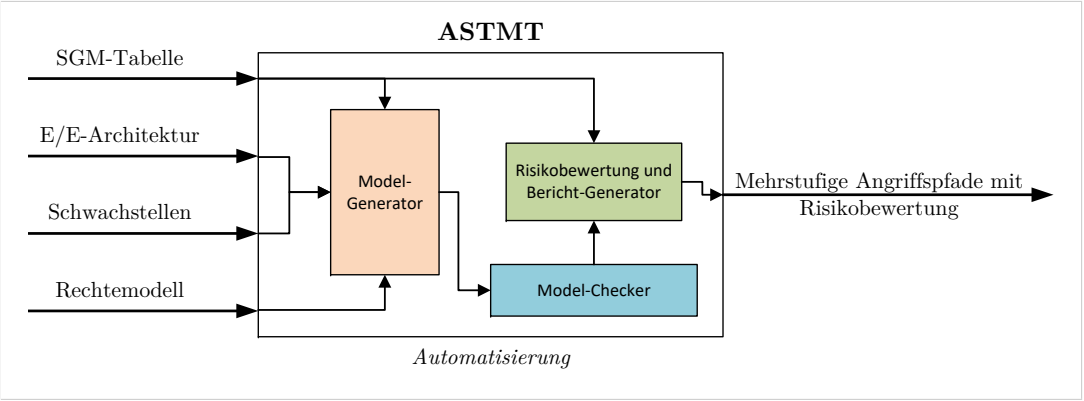


Abbildung 7.1: Darstellung der notwendigen Ein- und Ausgangsartefakte für den präsentierten Ansatz, der an dieser Stelle als ASTMT bezeichnet wird. Als Eingangsinformationen werden die SGM-Tabelle, E/E-Architektur, Schwachstellen sowie das Rechtemodell benötigt. Als Ergebnis wird eine priorisierte Liste von Bedrohungen erzeugt, die mehrstufige Angriffsschritte aufzeigt (Angriffspfade).

Das erzeugte Systemmodell wird anschließend vom Model-Checker überprüft und identifizierte Angriffspfade werden an die Risikobewertung beziehungsweise an den Bericht-Generator übergeben. Letzterer erzeugt eine Liste der aufgedeckten Angriffspfade, die anhand ihrer Risikowerte absteigend sortiert sind und an den Anwender übergeben werden. Für eine detaillierte Beschreibung der Softwarekomponenten und Softwarearchitektur des ASTMTs ist auf Anhang E verwiesen.

Da in den bisherigen Kapiteln bereits auf die SGM-Tabelle und E/E-Architekturen eingegangen wurde, werden im Folgenden die Schwachstellensammlung und das Rechtemodell vorgestellt.

7.2 Schwachstellen

Für die Erzeugung des Systemmodells M werden Schwachstellen benötigt, die in Transitionen transformiert werden können. Hierzu wird eine Sammlung von bereits bekannten Cyber-Angriffen auf Fahrzeuge eingesetzt, die in Form einer Schwachstellendatenbank [5] zur Verfügung gestellt wird. Diesbezüglich wurden 162 Cyber-Angriffe auf Fahrzeuge analysiert sowie eine einheitliche Klassifikation zur Beschreibung der Schwachstellen entwickelt [13]. Da die betrachteten Angriffe hauptsächlich aus mehreren Angriffsschritten bestanden, wurden die Angriffe in ihre elementaren Schritte aufgetrennt. Dies führte zu 413 Angriffsschritten, wobei jedem Schritt eine konkrete Schwachstelle zugeordnet ist.

Die Analyse der Angriffsschritte führte zu einer Klassifikation, die 23 Kategorien aufweist und mit Abbildung 7.2 erläutert wird. Aufgrund der umfangreichen Taxonomie werden nur kurz die Kategorien beschrieben, die für das Erzeugen des Systemmodells von Relevanz sind. Für die umfassende Beschreibung der Taxonomie ist auf die Veröffentlichung [13] verwiesen.

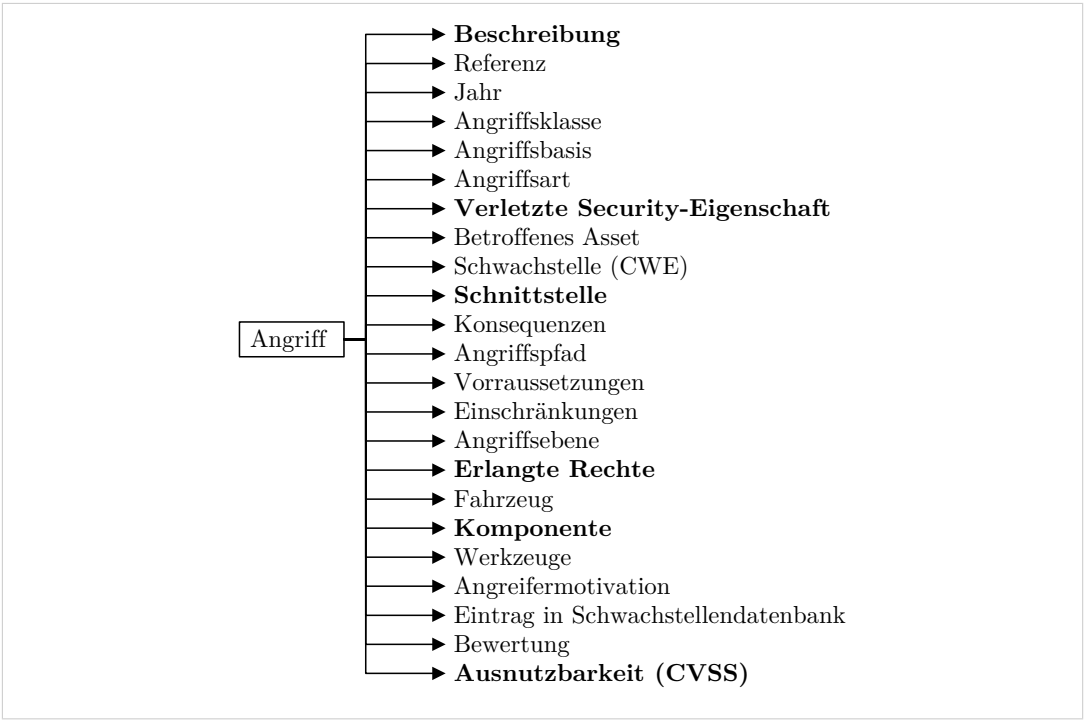


Abbildung 7.2: Hierarchie der vorgestellten Taxonomie, die zur Klassifizierung von Cyber-Angriffen auf Fahrzeuge entwickelt wurde und 23 Kategorien aufweist [13].

Die erste Kategorie, die für das ASTMT relevant ist, ist die *Beschreibung*, die einer textuellen Umschreibung des Angriffs entspricht und den Effekt, das Ziel sowie die Art des Angriffes zusammenfasst. Diese Informationen werden insbesondere für die Berichterstattung und die textuellen Angriffspfade benötigt. Die nächste hervorgehobene Kategorie ist die *Verletzte Security-Eigenschaft*, welche die Security-Eigenschaften beschreibt, die beim Ausnutzen einer Schwachstelle in einer Komponente oder Kommunikationsverbindung verletzt werden. Zu den Security-Eigenschaften zählen unter anderem die Vertraulichkeit, Integrität, Verfügbarkeit, Authentizität, Nichtabstreitbarkeit und Autorisierung. So kann beispielsweise die Vertraulichkeit der Daten (Firmware, kryptographische Schlüssel etc.) verletzt sein, die in einem Steuergerät gespeichert sind.

So lassen sich für jede Schwachstelle eines Angriffs die Schutzziele ableiten, die einen Teil des Systemmodells M darstellen. Die *Schnittstelle* über die ein Kommunikationspfad zum System aufbaut werden kann ist die nächste wichtige Kategorie. Schnittstellen können hierbei drahtlos wie WLAN, Global Positioning System (GPS) oder Radio Aircraft Detection and Ranging (Radar) sein, oder kabelgebunden wie beispielsweise CAN oder Ethernet. Diese Kategorie ist demzufolge wichtig für die Übertragung von Schwachstellen auf neu zu betrachtende E/E-Architekturelemente. So wird für den Aufbau des Systemmodells überprüft, welche Schnittstellen die jeweiligen Elemente der E/E-Architektur aufweisen und anhand dieser werden die passenden Schwachstellen zugewiesen.

Wurde eine Schwachstelle auf einer der vier Angriffsebenen ausgenutzt, so ist es dem Angreifer unter Umständen möglich privilegierte Rechte zu erhalten. Diese sind in der Kategorie *Erlangte Rechte* abgebildet und entsprechen den Rechteleveln, die mit Tabelle 7.1 gezeigt sind. Diese Kategorie fließt folglich direkt in den Aufbau des Systemmodells ein und bildet die Grundlage für das Verknüpfen von Schwachstellen durch ihr Rechtelevel. Im Gegensatz dazu fließt die Kategorie *Komponente* aus Abbildung 7.2 direkt in die Generierung des Systemmodells ein. Diese ordnet jede Komponente in eine der drei Klassen ECU, Sensor, Aktuator ein. Dies ermöglicht, die relevanten Schwachstellen aus der Datenbasis auf neue Komponenten zu übertragen. Die Kategorien zeigen außerdem auf einer nächsten tieferen Ebene mehr Details, sodass eine genauere Zuordnung erfolgen kann. Für die Klasse ECU kann dies beispielsweise ein Motor- oder Airbag-Steuergerät sein.

Die Kategorie *Schwachstellendatenbank* zeigt auf, ob eine Schwachstelle bereits in einer öffentlichen Datenbank gelistet ist. Etablierte Datenbanken sind beispielsweise die National Vulnerability Database (NVD) [144], die CVE [51] sowie Rapid7 [156] und Paket Storm [149]. Von besonderer Bedeutung ist hierbei die Schwachstellensammlung der CVE, welche die höchste Anzahl an fahrzeugbezogenen Schwachstellen aufzeigt [214]. Aufgrund dessen sind die in der erstellten Datenbank gelisteten Schwachstellen nach dem Schema der CVE [51] bewertet. Vertreten wird dies durch die Kategorie *Bewertung*, welche die CVSS-Bewertung [53] jeder Schwachstelle aufzeigt. Dieses Bewertungsschema erlaubt es, Schwachstellen einheitlich zu bewerten und hat sich als sinnvoll erwiesen [102]. Der sich daraus ergebene CVSS-Wert wird aus mehreren Teilwerten berechnet, die in Abschnitt 2.5.4 näher erläutert werden. Aus einer Menge dieser Teilwerte lässt sich außerdem die Ausnutzbarkeit einer Schwachstelle berechnen, die in der Kategorie *Ausnutzbarkeit* festgehalten ist. Sie dient in dieser Arbeit zur Brechung der Eintrittswahrscheinlichkeit von Angriffsschritten und löst damit die Gegebenheit der fehlenden Wahrscheinlichkeitswerte für Cyber-Bedrohungen.

Die Sammlung der Schwachstellen, welche nach der aufgezeigten Taxonomie beschrieben sind, ist unter [5] bereitgestellt.

7.3 Rechtemodell

Das Rechtemodell dient zur Beschreibung der notwendigen Rechtelevels, die benötigt werden, um eine Aktion auszuführen sowie nachfolgend ein höheres Rechtelevel zu erlangen. Hierzu sind fünf Klassen modelliert, die in der Arbeit [13] vorgestellt werden und sich bei einer Analyse von Cyber-Angriffen auf Fahrzeuge ergaben (Tabelle 7.1). Die gezeigten Klassen sind aus Sicht des Angreifers modelliert und stellen einen Vorschlag dar, bei dem zukünftige Erweiterungen sinnvoll erscheinen.

Tabelle 7.1: Die fünf Rechteklassen, die auf der durchgeführten Analyse [13] von Cyber-Angriffen auf Fahrzeuge basieren. Hierbei bezieht sich Stufe 1 auf Kommunikationsverbindungen und die Stufen 2 bis 5 auf Komponenten einer E/E-Architektur (ECU, Gateway, Sensor, etc.).

Stufe	Klasse	Typ
RL1	Lesen/Schreiben	Kommunikationskanal
RL2	Ausführen	Komponente
RL3	Lesen	Komponente
RL4	Schreiben	Komponente
RL5	Vollzugriff	Komponente

RL1 beschreibt die Privilegien, auf ein Kommunikationsmedium zu schreiben oder von diesem zu lesen. Dies entspricht beispielsweise dem Auslesen oder Senden von Botschaften auf einer CAN-Verbindung. Der Grund, weshalb das Recht für *Lesen/Schreiben* gleich behandelt wird, ist mit dem Umstand erklärt, dass Kommunikationskanäle mehrheitlich keine Maßnahmen besitzen, die ein Senden oder Auslesen von Daten unterbinden können, sollte ein Angreifer Zugriff auf das Übertragungsmedium haben. Ein Angreifer hat somit stets die Möglichkeit, Daten einer Kommunikationsverbindung lesen und schreiben zu können. Ob die ausgelesenen Daten hingegen interpretiert werden können, hängt von den Sicherheitsmechanismen wie beispielsweise einer Verschlüsselung ab.

RL2 entspricht dem *Ausführen* von Funktionen, die auf einer Komponente implementiert sind. Ein Beispiel hierfür ist die oben genannte Diagnosebotschaft, welche die Funktion der Airbag-Auslösung aktiviert. Für das (Aus-)Lesen auf einer funktionalen Komponente (RL3) ist hingegen ein höheres Rechtelevel erforderlich. Besitzt der Angreifer dieses Recht, ist es ihm möglich, Daten aus der Komponente zu extrahieren. Ein typisches Beispiel hierfür ist das Ausschleusen von kryptografischem Schlüsselmaterial einer ECU durch eine Schwachstelle. Hieran anknüpfend kann ebenfalls das Rechtelevel *Schreiben* (RL4) erläutert werden. Dieses entspricht der Berechtigung, Daten auf eine Komponente zu schreiben, als Beispiel das Ändern von Daten auf einer ECU. Das letzte und höchste Rechtelevel (RL) stellt den *Vollzugriff* auf der funktionalen Komponente (RL5) dar. Diese Berechtigung erlaubt die volle Kontrolle der Komponente und ist mit dem sogenannten « Root-Anwender » aus der IT vergleichbar (q_0 in Abbildung 7.1). Grundsätzlich ist es möglich, beliebig zwischen den Privilegien zu wechseln, d.h. ein Angreifer kann auch versuchen, von RL4 zu RL2 zu gelangen. Es gibt somit nicht unbedingt eine Hierarchie zwischen den RLs, obwohl man argumentieren kann, dass das RL der vollen Kontrolle die anderen RLs einschließt.

Das Rechtemodell wurde in der Arbeit [4] vorgestellt und ist grafisch mit Abbildung 7.3 dargestellt. Es zeigt die Zuordnung der Rechtelevel zu Komponenten und Kommunikationsverbindungen in einer E/E-Architektur.

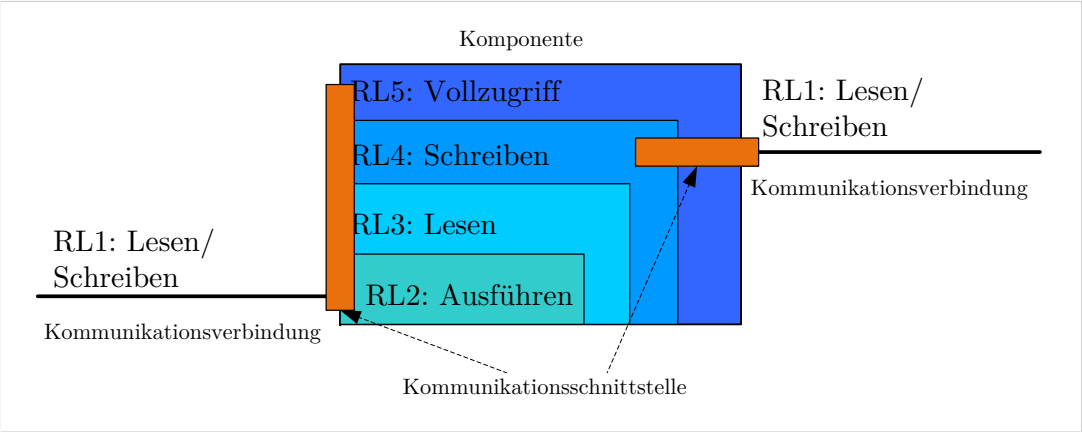


Abbildung 7.3: Rechtemodell, welches die Zuordnung der RL aus Tabelle 7.1 den Komponenten und Kommunikationsverbindungen zuweist. Den Komponenten sind hierbei die RL von 2 bis 5 und den Kommunikationsverbindungen das RL 1 zugewiesen.

Der Angreifer kann auf einem E/E-Architekturelement sein Rechtelevel erweitern, indem er bestimmte Aktionen ausführt. Konkret entsprechen diese Aktionen dem Ausnutzen von Schwachstellen, die eine Komponente besitzen kann. Sind mehrere Schwachstellen existent, kann ein Angreifer diese – abhängig von den Schwachstellenbedingungen – aufeinander aufbauend ausnutzen. Mit diesem Ansatz ist es möglich, eine Rechteausweitung über mehrere Schwachstellen hinweg zu modellieren. Erreicht ein Angreifer das Rechtelevel 4 oder 5, so wird ihm die Möglichkeit zugeschrieben, Kommunikationsschnittstellen einer Komponente kontrollieren zu können. Hiermit wird es ihm ermöglicht, Kommunikationsverbindungen zu nutzen, um mit benachbarten Komponenten in der E/E-Architektur kommunizieren zu können.

7.4 Automotive Security-Threat Modelling Tool (ASTMT)

Nachdem die notwendigen Eingangsartefakte für das ASTMT erläutert wurden, zeigt dieser Abschnitt, wie das Systemmodell M und die Spezifikation φ automatisiert erzeugt und überprüft werden. Für die Erzeugung des Zustandsraums und dessen Überprüfung wird eine Model-Checking Technik eingesetzt. Das ist insbesondere damit zu begründen, dass je nach Anzahl der E/E-Architekturelemente und Schwachstellen ein großer Zustandsraum erzeugt und überprüft werden muss. So sind Zustandsräume mit einer Größe von $\geq 10^5$ Zuständen möglich, sodass ein manueller Aufbau nicht zielführend ist. Model-Checking Techniken ermöglichen, mittels Beschreibungssprachen, ein Systemmodell computergestützt zu beschreiben und den Zustandsraum automatisiert zu erzeugen. Darüber hinaus sind in den letzten 30 Jahren Ansätze entwickelt worden, um mit dem Problem der Zustandsraumexplosion [44, 213] umgehen zu können, das bei dem hier verfolgten Ansatz eintreten kann.

Model-Checking Techniken können implizite Regelmäßigkeiten der Modellstruktur ausnutzen, um große Zustandsräume effizient zu erzeugen [26, Seite 16],[26, Seiten 13–14].

Hierzu gehören insbesondere symbolische Techniken [27] wie binäre Entscheidungsbäume [17, 35, 120] oder partielle Ordnungsreduktion [74]. Mit diesen Techniken ist der in dieser Arbeit verwendete Model-Checker [202] in der Lage, Zustandsräume mit $3,03 \cdot 10^{313}$ Zuständen im Millisekundenbereich zu erzeugen [43]. Zur Verifikation der Spezifikation durchläuft der Model-Checker alle möglichen Zustände des Systems und überprüft, ob bestimmte Eigenschaften (Spezifikation) eingehalten werden. Ist dies nicht der Fall, wird ein Gegenbeispiel erzeugt, was – in dieser Arbeit – einem Angriffspfad entspricht. Das hierzu notwendige Vorgehen ist mit Abbildung 7.4 zusammengefasst.

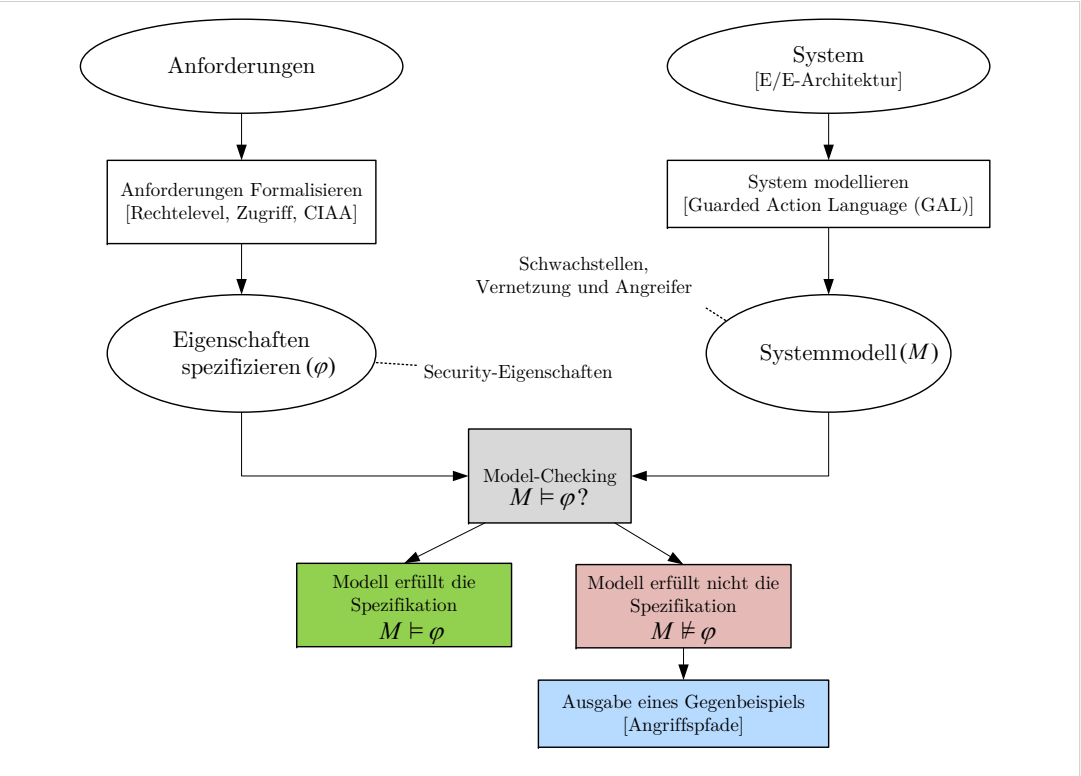


Abbildung 7.4: Systematischer Überblick über den Model-Checking-Ansatz, basierend auf [26, Seite 8]. Hierbei entspricht M dem Systemmodell, φ der Spezifikation und $\models$ der Erfüllungsrelation [26, Seite 320]. Die in den eckigen Klammern dargestellten Begriffe beschreiben außerdem die für den Ansatz geltenden Ausprägungen.

Als Eingangsartefakte für das Model-Checking werden das zu überprüfende System und die Anforderungen an dieses benötigt. Bezogen auf das hier betrachtete Problem entspricht dies einer E/E-Architektur mit ihren Komponenten, Kommunikationsverbindungen, Schwachstellen sowie der Anforderung, dass keine Bedrohung realisiert werden kann. Der Model-Checker überprüft somit, ob sich die mit der SGM identifizierten Bedrohungen realisieren lassen. Dies ist grundsätzlich gegeben, wenn ein Pfad von einem Startzustand zu einem Zustand führt, der eine Bedrohung auslösen kann. Die Informationen hierzu sind aus der SGM-Tabelle entnommen.

7.4.1 Modellierungsansatz

Ziel des Modellierungsansatzes ist es, dem Anwender, die vollständige Modellierung der Angriffe und deren Risikobewertung abzunehmen. Die Modellierung setzt auf einen Zustandsautomaten, bei dem die Zustände die möglichen Angriffsschritte eines Angreifers in einer E/E-Architektur darstellen. Hiermit wird modelliert, mit welchen Teilangriffen ein Angreifer eine Bedrohung aus der SGM-Tabelle realisieren kann (Tabelle 4.2). Ein Zustand spiegelt den Security-Status eines Architekturelementes und das Rechtelevel des Angreifers auf dem Architekturelement wider. Die Transitionen des Zustandsautomaten entsprechen dem Ausnutzen einer Schwachstelle, die ein bestimmtes Rechtelevel erfordert und zu einem neuen Rechtelevel führen kann.

Der Security-Status entspricht Schutzzielen die sich auf Daten beziehen, welche auf einem Architekturelement gespeichert oder über ein Element übertragen werden. Hierbei ist die Annahme getroffen, das ein Verletzten eines Schutzziels stets für das gesamte Element zutrifft. So bedeutet beispielsweise das Verletzten der Integrität des Airbag-Steuergerätes, dass grundsätzlich alle auf dem Airbag-Steuergerät vorhanden Daten manipuliert sein können. Eine Auftrennung in unterschiedliche Assets wie beispielsweise Firmware oder kryptographisches Schlüsselmaterial ist zum aktuellen Zeitpunkt nicht modelliert. Der Modellierungsansatz ist allerdings in der Lage, mehrere feingranulare Assets in den E/E-Architekturelementen abbilden zu können. Diese Abstraktion ist damit zu erklären, dass zum einen die Komplexität reduziert wird und zum anderen, dass in frühen Entwicklungsphasen nur wenig Details über mögliche Assets in den Architekturelementen bekannt sind.

Damit Eigenschaften – wie die Schutzziele eines Assets – überprüft werden können, müssen diese zuvor formalisiert werden. Hierzu werden Formeln der temporalen Logik verwendet, um die Verletzung von Eigenschaften (φ) zu beschreiben. Konkret wird die Computation Tree Logic (CTL) [26, Seite 317] verwendet, die eine Erweiterung der Modallogik um temporale Operatoren ist und zeitliche Einschränkungen formulieren kann [72, Seite 106]. Für das vorliegende Problem bedeutet dies, dass Anforderungen in folgender Form beschrieben werden können: *Ein Angreifer darf niemals das Rechtelevel 5 auf der Airbag-ECU erlangen und dessen Integrität verletzten.* Zu dessen Überprüfung werden alle denkbaren Ausführungen des Systems als alternierende Sequenz von Zuständen und Aktionen verstanden [26, Seite 24]. Hierzu durchläuft ein Model-Checker alle Ausführungen und prüft, ob jemals ein Zustand die spezifizierte Eigenschaft verletzt ($M \not\models \varphi$).

Systemmodell M

Das Systemmodell bildet die E/E-Architektur mit den darin enthaltenen Schwachstellen ab, die ein Angreifer ausnutzen kann, um spezifische Assets anzugreifen und damit Bedrohungen auslösen zu können. Hierzu wird für jede Komponente und Kommunika-

tionsverbindung in der E/E-Architektur geprüft, welche Schwachstellen applizierbar sind und bei einer Übereinstimmung werden diese in das Modell aufgenommen[1]. Da die Schwachstellen aus der Datensammlung die benötigten und zu erreichenden Rechtelevel sowie die Security-Eigenschaften aufweisen, kann mit diesen Informationen der Zustandsraum mit Zustandsübergängen erzeugt werden.

Zur formalen Beschreibung des Systemmodells wird eine Modellierungssprache verwendet, die der Model-Checker in ein Transitionssystem (TS) übersetzt. Ein TS stellt sich als eine besondere Form des Akzeptors dar und lässt sich nach Baier et al. [26, Seite 20] als ein 6-Tupel $TS = (S, Act, \rightarrow, S_0, AP, L)$ darstellen.

- S eine Menge von Zustände,
- Act die Menge der Aktionen,
- $\rightarrow \subseteq S \times Act \times S$ entspricht der Transitionsrelation,
- $S_0 \subseteq S$ beschreibt die Menge der Startzustände,
- AP ist eine Menge von atomaren Aussagen und
- $L : S \rightarrow 2^{AP}$ eine Beschriftungsfunktion (engl. labeling function)

Angewendet auf ein Beispiel lässt sich das TS wie in Abbildung 6.4 beschreiben. Hierbei handelt es sich um ein Airbag-Steuergerät, das direkt mit dem OBD-Port verbunden ist und bei dem der Angreifer Zugang zum OBD-Port erlangt hat. Hierbei entsprechen $s_0, s_1 \in S$ den Zuständen des TS und s_0 dem Startzustand. Die Transition $\xrightarrow{\alpha_0}$ mit $\alpha_0 \in Act$ wird als das Ausnutzen einer Schwachstelle verstanden, welche ausgenutzt werden muss, um in den Zustand s_1 wechseln zu können. Die konkrete Schwachstelle υ_0 ist in diesem Fall, dass der OBD-Port keine Maßnahme zur Autorisierung für das Senden von CAN-Botschaften fordert und der Angreifer in der Lage ist, eine « Zündungs-Botschaft » zu versenden, welche die Airbags auslöst.

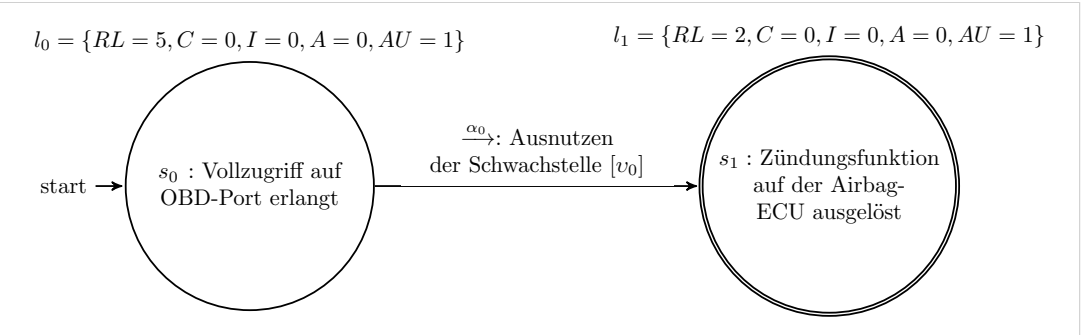


Abbildung 7.5: Exemplarisches TS mit der Transition $\xrightarrow{\alpha_0}$ und der dazugehörigen Schwachstelle υ_0. Außerdem sind die beiden Beschriftungen l_0 und l_1 aufgezeigt (vgl. Baier et al. [26, Seite 20]). Diese zeigen die Eigenschaften Rechtelevel (RL) und die Schutzziele (C, I, A, AU) eines Zustands (s_i) auf.

[1]Für einen graphischen Ablauf der Erzeugung des Systemmodells ist auf Anhang D.1 verwiesen.

Für das Ausnutzen der Schwachstelle v_0 muss der Angreifer ein bestimmtes Rechtelevel erreicht haben, das sich nach einem erfolgreichen Ausnutzen von v_0 erhöhen kann. Um dies abzubilden, wird das Rechtemodell aus Abschnitt 7.3 in die Modellierung mit einbezogen, was in der Arbeit [4] aufgezeigt wurde. Für s_0 in Abbildung 7.5 ist dies der Vollzugriff ($RL_{s_0} = 5$) auf dem OBD-Port mit dem verletzten Schutzziel *Authentizität*[2] sowie s_1 mit dem $RL_{s_1} = 2$ auf dem Airbagsystem. Somit ergibt sich

- die Menge der Zustände S aus dem kartesischen Produkt aus den E/E-Architekturelementen (z.B. OBD) $\times$ Rechteleveln (RL_{s_i}) $\times$ Schutzzielen (C, I, A, AU),
- die Menge der Aktionen *Act* die den applizierbaren Schwachstellen entsprechen,
- der Startpunkt s_0 aus der Spalte *Eintrittspunkt* der SGM-Tabelle,
- das Ziel des Angreifers aus der Spalte *Komponente/Teilsystem* der SGM-Tabelle.

Die Eigenschaften jedes Zustands werden durch die Beschriftungen l_0 und l_1 festgehalten und dienen im späteren Verlauf dazu, auf bestimmte Eigenschaften – wie die Schutzziele – zu prüfen. Bezogen auf die Eingangsartefakte aus Abbildung 7.1 ist s_1 aus der SGM-Tabelle entnommen und entspricht dem Ziel des Angreifers. Für den Zustand s_0 ist die eingehende E/E-Architektur analysiert und eine Kommunikationsverbindung zwischen dem OBD-Port und der Airbag-ECU identifiziert, was das Übertragen der Diagnosebotschaft ermöglicht.

Ein TS mit mehreren Zuständen, das dem gleichen Modellierungsansatz wie in Abbildung 7.5 folgt, ist in Abbildung 7.6 gezeigt. Hier ergeben sich l_1, l_2, l_n, l_m durch die Beschriftungsfunktion $L(s)$, welche die erreichten Rechte des Angreifers (RL) und die Schutzziele (C,I,A,AU) abbildet. Die Transitionen $\xrightarrow{\alpha_i}$, können – wie bereits erläutert – als Ausnutzen einer Schwachstelle durch den Angreifer verstanden werden. Hier sind die $v_i \in V$ jene Schwachstellen, die der Menge der applizierbaren[3] Schwachstellen entsprechen.

[2] Vertraulichkeit: $C = 0$, Integrität: $I = 0$, Verfügbarkeit: $A = 0$ und ¬Authentizität: $AU = 1$

[3] Die applizierbaren Schwachstellen entsprechen Schwachstellen, die aus einer Datensammlung [5] entnommen sind und die sich einem E/E-Architekturelement nach Anhang D.1 zuordnen lassen.

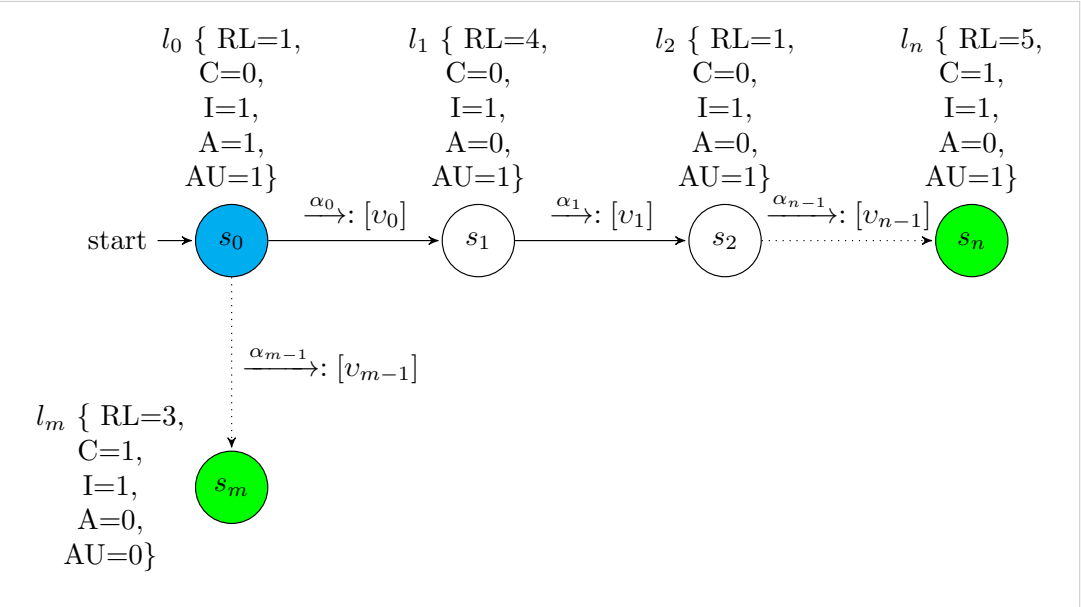


Abbildung 7.6: Grafische Darstellung des Transitionssystem (TS) mit dem blau markierten Startzustand (s_0), den grünen Endzuständen s_n, s_m sowie den Transitionen $\xrightarrow{\alpha_i}$ und den Beschriftungen der Zustände durch l_i. Letztere zeigen das Rechtelevel (RL), den Zugriff (Z) sowie die Schutzziele: Integrität (I), Vertraulichkeit (C), Verfügbarkeit (A) und Authentizität (AU).

Die Beschriftung l_1 in s_1 besitzt die Eigenschaften RL=4, I=1 und AU=1. Die beiden letzten Eigenschaften bedeuten, dass in diesem Zustand die Integrität und Authentizität des Systems verletzt ist. Grund hierfür ist das Ausnutzen der Schwachstelle v_0, die dem Angreifer Zugriff auf die Komponente und ein Rechtelevel (RL) von 4 ermöglichte. Übertragen auf die konkrete Komponente OBD-Port kann die Transition $s_0 \xrightarrow{\alpha_0} s_1$ als die Möglichkeit zum Einstecken eines OBD-Adapters verstanden werden. Die Schwachstelle[4] v_1 ist hierbei ein fehlender Mechanismus für die Autorisierung der Verwendung des OBD-Ports. Hiermit kann der Angreifer Botschaften an das Airbag-Steuergerät senden. Durch Ausnutzen weiterer Schwachstellen, die in der Airbag-ECU zu finden sind, erlangt der Angreifer abschließend (s_n) das RL=5 und kann damit die Vertraulichkeit (C) und Integrität (I) der ECU verletzen. An dieser Stelle hat der Angreifer das RL von 5 erreicht. Dies widerspricht der Spezifikation, das ein Angreifer niemals RL=5 auf der Airbag-ECU erlangen darf, was zur Ausgabe eines Gegenbeispiels führt.

Die konzeptionelle Modellierung mit einem TS erfordert eine Transformation der E/E-Architektur und der dazu passenden Schwachstellen. So werden in einem ersten Schritt die bekannten Schwachstellen den Elementen der E/E-Architektur zugeordnet. Für jede Komponente oder Kommunikationsverbindung wird dazu überprüft, ob ein Element aus der Menge der Schwachstellen applizierbar ist. Um dies entscheiden zu können, werden die Eigenschaften der Schwachstelle mit denen des Elementes vergli-

[4]Die Integration der Schwachstellen aus der Datenbank in das Modell ist automatisiert (Anhang E).

chen. Das ist insbesondere die Art des Elements, die zu einer Unterscheidung zwischen CPS-Elementen (ECU, Gateway, Aktuator, Sensor, MMS) oder einer Kommunikationsverbindung führt. Dies wird weiter spezialisiert, indem die Art des Elementes konkretisiert wird. Ist beispielsweise als Komponente eine ECU ausgewählt und das Element entspricht einer Airbag-ECU, so werden nur jene Schwachstellen zugewiesen, die für eine Airbag-ECU bekannt sind. Ist die Schwachstelle applizierbar, so wird diese dem E/E-Architekturelement zugewiesen und abgespeichert. Der dazugehörige Ablauf ist in Anhang D.1 grafisch aufbereitet.

Sind alle E/E-Architekturelemente betrachtet und die relevanten Schwachstellen appliziert, werden die gespeicherten Schwachstellen automatisierst in Transitionen überführt. Hierzu verwendet das entwickelte Softwarewerkzeug die Beschreibungssprache Guarded Action Language (GAL) des Model-Checkers [201] und erzeugt eine textuelle Beschreibung des Systemmodells. Hierbei wird die Vernetzung der Komponenten, die durch die Kommunikationsverbindungen beschrieben sind, beim Systemmodell miteinbezogen. Dazu leitet das Softwarewerkzeug eine Adjazenzmatrix von der – vom Anwender bereitgestellten – E/E-Architektur ab, bei der die Komponenten und Kommunikationsverbindungen als eindeutiger Bezeichner erfasst werden. Diese dienen im Systemmodell zur Speicherung der aktuellen Position des Angreifers in der E/E-Architektur und ermöglichen, dass der Angreifer zwischen E/E-Architekturelementen wechseln kann. Hiermit wird es ermöglicht, auch Bussysteme modellieren zu können, womit sich das ASTMT beispielsweise vom Microsoft TMT [154] abgrenzt. Ein konkretes Modellierungsbeispiel ist in Anhang D.2 gegeben.

7.4.2 Spezifikation φ

Ist der Zustandsraum erzeugt, überprüft der Model-Checker anschließend, ob die gestellte Spezifikation $(M, s \models \varphi)$ erfüllt ist. Hierzu muss allerdings eine Spezifikation definiert sein, die sich aus der SGM-Tabelle ableiten lässt. Diesbezüglich sind es die Leitwörter und deren Zuordnung zu den Schutzzielen, die sich als Vertraulichkeit (C), Integrität (I), Verfügbarkeit (A) darstellen. Die Authentizität (AU) wird als Bestandteil der Integrität gesehen, um den Zustandsraum möglichst kleinzuhalten. So kann aus jedem Leitwort eine CTL-Formel bestimmt werden, die in Tabelle 7.2 aufgeführt sind.

Der Allquantor AG beschreibt dabei, dass auf allen Pfaden beginnend in s die spezifizierten Eigenschaften eingehalten werden müssen [26]. Damit allerdings gezielt auf Verletzungen der Schutzziele geprüft werden kann, werden die CTL-Formeln negiert. Die Formel fordert somit, dass zu keinem Zeitpunkt die Schutzziele als verletzt markiert sind. Die Variable AE in Tabelle 7.2 entspricht einem E/E-Architekturelement, das vom Angreifer kompromittiert werden muss, um die Bedrohung auszulösen. Die Information hierzu kann aus der 5 Spalte der SGM-Tabelle entnommen werden (Tabel-

le 4.2). So kann für das Architekturelement *Airbag_ECU* die Spezifikation in folgender Form beschrieben werden: $AG\ !(Airbag\_ECU.RL >= 2 \wedge Airbag\_ECU.I = 1)$.

Tabelle 7.2: Ableitung der CTL-Formeln aus den SGM-Leitwörtern. Hierbei ist zwischen Komponenten und Kommunikationsverbindungen unterschieden. Die Formeln zeigen die Schutzziele Vertraulichkeit (C), Integrität (I), Verfügbarkeit (A) sowie das Rechtelevel (RL).

Leitwort	Komponente	Verbindung
Auslösen	$AG\ !(AE.RL >= 2 \wedge AE.I = 1)$	-
Einschleusen	$AG\ !(AE.RL >= 4 \wedge AE.I = 1)$	$AG\ !(AE.RL = 1 \wedge AE.I = 1)$
Manipulieren	$AG\ !(AE.RL >= 4 \wedge AE.I = 1)$	$AG\ !(AE.RL = 1 \wedge AE.I = 1)$
Unterbrechen	$AG\ !((AE.RL = 2 \vee AE.RL >= 4) \wedge AE.A = 1)$	$AG\ !(AE.RL = 1 \wedge AE.A = 1)$
Verzögern	$AG\ !((AE.RL = 2 \vee AE.RL >= 4) \wedge AE.A = 1)$	$AG\ !(AE.RL = 1 \wedge AE.A = 1)$
Löschen	$AG\ !(AE.RL >= 4 \wedge AE.A = 1 \wedge AE.I = 1)$	$AG\ !(AE.RL = 1 \wedge AE.A = 1 \wedge AE.I = 1)$
Stoppen	$AG\ !((AE.RL = 2 \vee AE.RL >= 4) \wedge AE.A = 1)$	-
Zurücksetzen	$AG\ !((AE.RL = 2 \vee AE.RL >= 4) \wedge AE.A = 1)$	-
Auslesen	$AG\ !(AE.RL >= 3 \wedge AE.C = 1)$	$AG\ !(AE.RL = 1 \wedge AE.C = 1)$

Die gezeigten Leitwörter müssen im Hinblick auf die SGM und deren Safety-Fokus interpretiert werden. So verletzt beispielsweise das *Einschleusen* oder das *Manipulieren* von Daten die Integrität einer Komponente oder einer Kommunikationsverbindung. Hierbei werden entweder Daten eingebracht oder bestehende Daten verändert, was zu einem Abweichen des Systemverhaltens führen kann. In gleicher Weise verhält es sich mit den Leitwörtern *Unterbrechen* und *Verzögern*, welche die Nicht-Verfügbarkeit von Daten repräsentieren. So kann das Unterbrechen einer Funktionalität oder das Verzögern von Daten zu einer Abweichung des Systemverhaltens führen. Ebenso kann das *Stoppen* oder *Zurücksetzten* einer Komponente dazuführen, dass für das spezifizierte Systemverhalten entscheidende Funktionalitäten nicht zur Verfügung stehen.

Für das SGM-Leitwort *Einschleusen* kann die Spezifikation interpretiert werden, dass niemals ein RL=4 oder höher erreicht werden darf und dass die Integrität nicht verletzt ist (I=1). Die Festlegung dieser Rechtelevel ist damit zu begründen, dass mit RL=4 der Angreifer auf einer Komponente Schreibrechte erlangt und die Integrität der Komponente verletzt. Außerdem besitzt der Angreifer mit RL=5 den Vollzugriff auf der Komponente, was ebenfalls eine Integritätsverletzung ermöglicht. Für Bedrohungen, die auf die Verfügbarkeit einer Komponente abzielen, sind die RL=1, 2, 4 und 5 von Relevanz. So kann ein Angreifer mit RL=1 auf einer Kommunikationsverbindung eine hohe Anzahl von Botschaften senden und diese überlasten. In der gleichen Weise kann er mit einem RL=2 für eine Komponente eine hohe Anzahl von Anfragen zur Ausführung einer Funktionalität stellen und damit die Komponente überlasten. Mit RL=4 und 5 ist es hingegen möglich, Schadcode einzuschleusen oder bestehende Funktionalitäten zu überschreiben, sodass bestimmte Dienste möglicherweise nicht mehr zur Verfügung stehen.

7.5 Generierung der Angriffspfade

Erfüllt das Systemmodell M die Spezifikation φ, ist dies gleichbedeutend mit der Aussage, dass die betrachtete Bedrohung nicht realisiert werden kann, da es keine Pfade vom Startzustand zu jenen Zuständen gibt, welche eine Bedrohung auslösen können. Ist die Spezifikation hingegen verletzt ($M, s \not\models \varphi$), so wird ein Gegenbeispiel in Form eines Angriffsgrafen ausgegeben, wie beispielhaft mit Abbildung 7.7 gezeigt ist.

Der Startzustands (s_0) wird aus der Spalte *Eintrittspunkt* der SGM-Tabelle abgeleitet und entspricht einem E/E-Architekturelement, auf das der Angreifer mit einem bestimmten Rechtelevel Zugriff hat. Ein Beispiel hierfür ist der OBD-Port. Zustände, die eine Bedrohung realisieren, werden ebenfalls aus der SGM-Tabelle abgeleitet. So wird aus Spalte 5 das E/E-Architekturelement übernommen und anhand des Leitwortes aus der dritten Spalte das Schutzziel identifiziert (vgl. Tabelle 6.3). Für beispielsweise das Leitwort *manipulieren* und das E/E-Architekturelement *Airbag-ECU* würden alle Zustände, welche eine Verletzung der Integrität auf der *Airbag-ECU* aufzeigen, somit die Bedrohung – das ungewollte Auslösen der Airbags – realisieren. Dies ist in Abbildung 7.7 mit den Zuständen s_4, s_6, s_7, s_8, s_9 hervorgehoben.

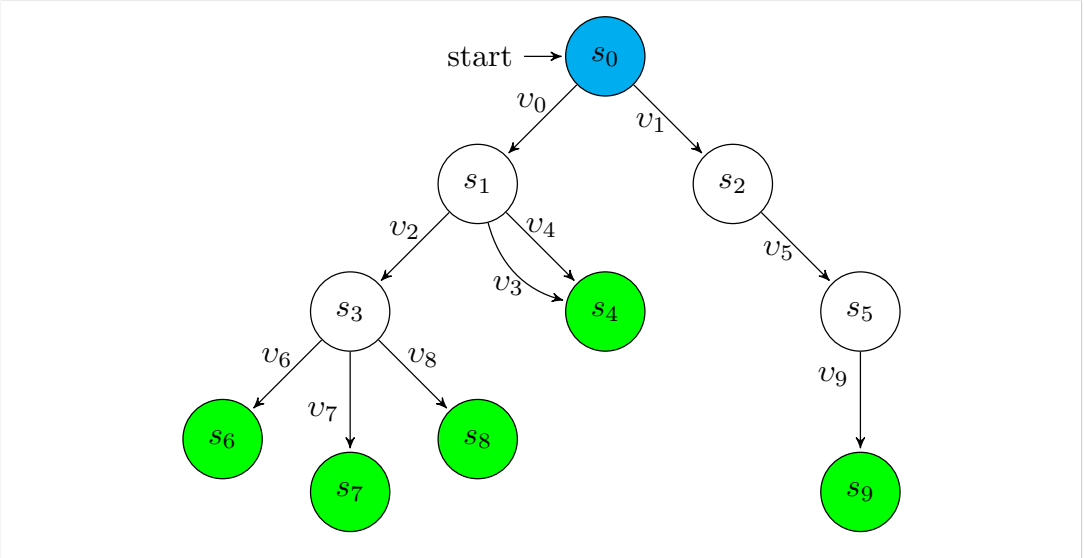


Abbildung 7.7: Darstellung des Gegenbeispiels, das bei Verletzen der Spezifikation (φ) vom Model-Checker ITS-tools [202] ausgegeben wird. Der Startzustand s_0 entspricht dabei dem Eintrittspunkt des Angreifers und s_6, s_7, s_8, s_4, s_9 jenen Zuständen, welche φ durch ihre Beschriftungsfunktion $L(s_i)$ verletzen.

Das Gegenbeispiel weist Elemente auf, die in einer Beziehung mit der SGM-Tabelle stehen, sodass eine Verknüpfung zwischen der SGM-Tabelle und dem Gegenbeispiel besteht. So entspricht der blau-markierte Startzustand s_0 dem Eintrittspunkt in der sechsten Spalte der SGM-Tabelle. Die grün markierten Zustände (s_4, s_6, s_7, s_8, s_9) beziehen sich hingegen auf die – in der SGM-Tabelle – identifizierten Bedrohungen. Die

Bedrohungen können als realisiert betrachtet werden, wenn ein grün-markierter Zustand erreicht wird, was aus Sicht des Model-Checking eine Verletzung der Spezifikation (φ) darstellt.

Die Begründung für mehrere Endzustände (s_4, s_6, s_7, s_8, s_9) in Abbildung 7.7 kann mit der Beschreibungsfunktion $L : S \rightarrow 2^{AP}$ gegeben werden, die jedem Zustand eine Menge von Eigenschaften (atomare Aussage) zuordnet. Entspricht die definierte Spezifikation allerdings nur einer Teilmenge der möglichen atomaren Aussagen, so können mehrere Beschriftungen ($l_i \in L$) die Spezifikation verletzen. Zur Verdeutlichung seien zwei Schwachstellen herangezogen, die ein Erlangen des RL=5 ermöglichen und zusätzlich die Integrität (I) beziehungsweise die Vertraulichkeit (C) verletzen. Wird nun überprüft, ob jemals das RL=5 durch den Angreifer erlangt werden kann, so beinhaltet der Zustandsraum unter anderem die Zustände mit den Beschriftungen $l_1\{RL = 5, I = 1\}$ und $l_2\{RL = 5, C = 1\}$. Beide verletzen die Forderung, dass das RL niemals 5 sein darf, unterscheiden sich allerdings in ihren Schutzzielen und ergeben so zwei unterschiedliche Zustände, welche die gleiche Bedrohung auslösen können.

Die in Abbildung 7.7 gezeigten Transitionen $\upsilon_0, ..., \upsilon_9$ entsprechen den jeweilig ausgenutzten Schwachstellen. Aufgrund der Tatsache, dass die Schwachstellentransitionen eine Veränderung der RL repräsentieren und unterschiedliche Schwachstellen zum gleichen RL führen können, sind multiple Übergänge im Gegenbeispiel möglich (υ_3, υ_4). Dies führt dazu, dass die Anzahl der Pfade von einem Startzustand (s_0) zu einem terminierenden Zustand (s_4) im gleichen Maße mit den parallelen Kanten anwächst. So ergeben sich beispielsweise mit υ_3 und υ_4 zwei Pfade, die zur Realisierung der identischen Bedrohung führen ($s_0 \xrightarrow{\upsilon_0} s_1 \xrightarrow{\upsilon_3} s_4$, $s_0 \xrightarrow{\upsilon_0} s_1 \xrightarrow{\upsilon_4} s_4$).

7.5.1 Vergleichbare Modellierungsansätze

Ein Ansatz mit einer ähnlichen Vorgehensweise ist von den Forschern Richtey und Ammann in einem Papier [166] vorgestellt, um Schwachstellen in Computernetzen zu analysieren. Allerdings zeigten die Forscher kein Konzept zur automatisierten Erzeugung des Systemmodells, sodass die Erstellung des Systemmodells manuell durchgeführt werden muss. Darüber hinaus führten die Forscher kein Rechtemodell für die Modelierung der Transitionen ein, sondern setzten das Vorhandensein von Exploits voraus, um Schwachstellen ausnutzen zu können. Hiermit hängt allerdings die Anzahl der Angriffspfade von Schwachstellen- und Exploitmenge ab, die manuell befüllt werden muss. Außerdem konnten die Forscher bei der Verifikation der Spezifikation ausschließlich einen Angriffspfad erzeugen, auch wenn im Systemmodel etliche weitere Angriffspfade bestanden.

Eine weitere und verwandte Arbeit wurde von Salfer et al. [130] vorgestellt, die eine modifizierte Tiefensuche zum Aufbau des Zustandsraums (Angriffsgrafen) und zur Bestimmung der Eintrittswahrscheinlichkeiten einsetzt. Das gezeigte Vorgehen bezieht

allerdings keine bekannten Security-Probleme wie Schwachstellen in die Pfadsuche mit ein, sondern orientiert sich an der Angriffsfläche der betrachteten Elemente. Die Forscher reduzieren außerdem den Aufwand für die Pfadsuche, indem unwahrscheinliche Pfade verworfen werden [130]. Hierzu ist eine Schranke für die Eintrittswahrscheinlichkeit definiert, die einen Pfad verwirft, wenn sie unterlaufen wird. Hiermit wird die Analyse beschleunigt, die Vollständigkeit der Angriffspfade – bezogen auf das Systemmodell – ist allerdings nicht mehr gegeben.

7.5.2 Diskussion und kritische Auseinandersetzung

Der hier vorgestellte Ansatz setzt einen Fokus auf die computergestützte Modellierung von E/E-Architekturen und deren Schwachstellen, die den Anwender bei der Bedrohungsmodellierung in allen Analyseschritten unterstützt. Konkret muss der Anwender ausschließlich die E/E-Architektur und die SGM-Tabelle übergeben. Das Einbeziehen und Zuordnen von Schwachstellen aus einer aufgebauten Datenbank sowie das Erzeugen und Prüfen des dazugehörigen Systemmodells geschehen vollständig automatisiert. Für die Erreichung dieses Ziels wurde die Model-Checking Technik ausgewählt, um den Zustandsraum und die Transitionen automatisiert aus einem formalen Systemmodell erzeugen zu können. Hierbei wird ebenso das Systemmodell automatisiert erzeugt und dem Model-Checker übergeben, der das aufgebaute Modell auf die Verletzung von Security-Eigenschaften prüft. Wird die Spezifikation verletzt, entspricht das ausgegebene Gegenbeispiel allen möglichen Angriffspfaden, die ein Angreifer durch Ausnutzen von Schwachstellen erreichen kann.

Eine Aussage über die Vollständigkeit der erzeugten Angriffspfade kann nicht gegeben werden, da durch die Modellierung die realen Fahrzeugsysteme abstrahiert werden. Die Modellierung ist allerdings erforderlich, um die Komplexität der Problemstellung zu verringern und praktikabel lösen zu können. So zeigen auch verwandte Arbeiten [30, 38, 71, 90, 130] die Notwendigkeit zur Abstraktion, um Angriffspfade praxistauglich erzeugen zu können. Im Gegensatz dazu ermöglicht der hier vorgestellte Ansatz die vollständige Ausgabe aller Angriffspfade, die sich aus dem Systemmodell ableiten lassen.

7.6 Risikobewertung

Nachdem der automatisierte Ansatz zur Erzeugung der Angriffspfade in Abschnitt 7.5 gezeigt wurde, wird in diesem Abschnitt das Vorgehen für eine Risikobewertung der aufgedeckten Angriffspfade präsentiert. Für die sinnvolle Priorisierung der auf safety-bezogenen Angriffspfade kann der Schweregrad S_S aus der Safety-Analyse übernommen werden (vgl. Abschnitt 2.3.1). Die Kontrollierbarkeit C kann hingegen nicht verwendet werden, da bei diesem Kennwert die Annahme getroffen wird, dass bei einer Verhaltensabweichung der Fahrer auf die Abweichung adäquat reagieren kann. Dies beinhaltet die Voraussetzung, dass keine weiteren Situationen eintreten, die es dem Fahrer erschweren,

auf die Verhaltensabweichung zu reagieren. Ein Angreifer kann eine solche Situation herbeiführen, indem er beispielsweise den Fahrersitz in die Endstellung fährt oder eine Verriegelung des Lenkrades herbeiführt. Hiermit ist der Fahrer nur eingeschränkt in der Lage, auf die Verhaltensabweichung zu reagieren. Ähnlich verhält es sich mit der Bewertung für die Häufigkeit der Fahrsituation E, die im Safety-Kontext abgeschätzt wird. So muss im Security-Kontext davon ausgegangen werden, dass sich der Angreifer stets jene Fahrsituation auswählt, die zum höchsten Schaden führt. Hieraus entsteht die Gegebenheit, dass Eintrittswahrscheinlichkeiten für safety-relevante Cyber-Bedrohungen auf eine andere Weise bereitgestellt werden müssen. Aufgrund dessen zeigen die nachfolgenden Abschnitte, wie die Bewertungsmetrik Common Vulnerability Scoring System (CVSS) mit den Ergebnissen des Model-Checkers kombiniert werden kann, um Aussagen über die Eintrittswahrscheinlichkeit von Angriffspfaden treffen zu können. Hierauf aufbauend und unter Einbezug des Schadenswerts aus der Safety wird anschließend für jeden Angriffspfad automatisiert ein Risikowert bestimmt.

7.6.1 Bestimmung der Eintrittswahrscheinlichkeit mit dem CVSS

Das in Abschnitt 2.5.4 aufgezeigte Common Vulnerability Scoring System (CVSS) ermöglicht es, die Ausnutzbarkeit von Schwachstellen zu beschreiben. Hierzu wird die *Exploitability Metric* der *Base Metric Group* verwendet. Sie lässt sich als Eintrittswahrscheinlichkeit für eine Schwachstelle interpretieren, da sie beschreibt, wie aufwendig es für den Angreifer ist, diese auszunutzen. Basierend auf diesem Ansatz kann jeder Kante des Gegenbeispiels eine Eintrittswahrscheinlichkeit zugeordnet werden. Hierbei entspricht ϵ_i der Ausnutzbarkeit für die Schwachstelle v_i, sodass das Gegenbeispiel des Model-Checkers in den nachfolgenden Grafen transformiert wird (Abbildung 7.8).

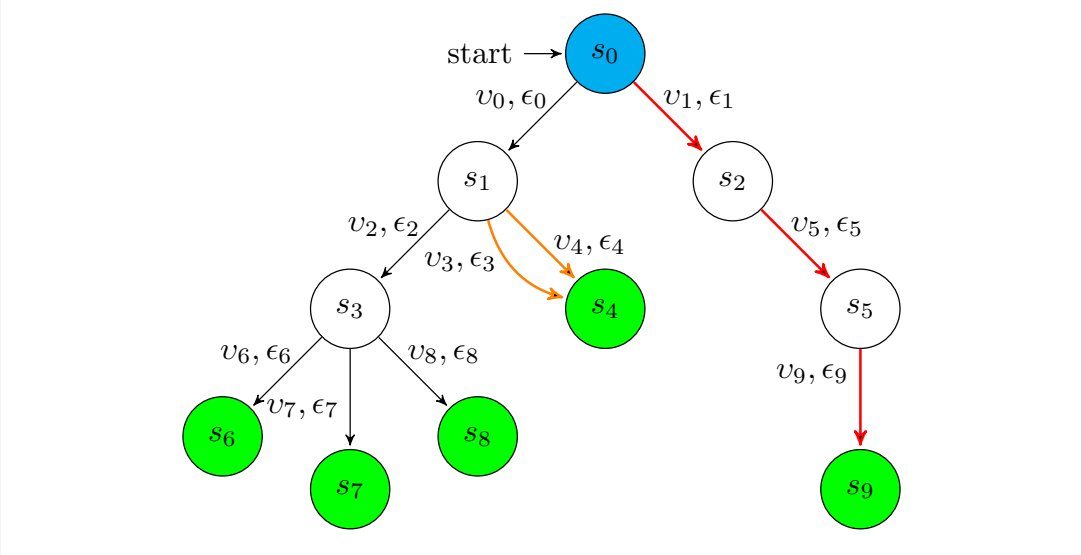


Abbildung 7.8: Darstellung eines Gegenbeispiels, das durch die Verletzung der Spezifikation (φ) vom Model-Checker ITS-tools [202] ausgegeben wird. Der Startzustand s_0 entspricht dabei dem Eintrittspunkt des Angreifers und s_6, s_7, s_8, s_4, s_9 jenen Zuständen, die φ durch ihre Beschriftungsfunktion $L(s_i)$ verletzen. Außerdem entsprechen die ϵ_i den Eintrittswahrscheinlichkeiten der jeweiligen Schwachstelle υ_i.

Wird ein Pfad (a) als Abfolge von Zuständen und Transitionen verstanden $a = \{s_0, \upsilon_0, s_1, \upsilon_1, ..., \}$, sowie die numerischen Werte für die Ausnutzbarkeit ϵ_i einer Schwachstelle auf einen Wertebereich von $[0, 1]$ normiert ($\hat{\epsilon_i} = \frac{\epsilon_i - \epsilon_{min}}{\epsilon_{max} - \epsilon_{min}}$) und mit $\hat{\epsilon_i}$ bezeichnet, kann mit Gleichung (7.1)

$$P(a) = \prod_{i=0}^{n} \hat{\epsilon_i} \ , \tag{7.1}$$

die Ausnutzbarkeit eines Angriffspfades bestimmt werden. Hierbei entspricht $\epsilon_{min} = 0$ der unteren und $\epsilon_{max} = 3,9$ der oberen Intervallgrenze der *Exploitability Metric* aus Abschnitt 2.5.4. Bezogen auf den rechten und rot markierten Pfad in Abbildung 7.8 entspricht das Anwenden von Gleichung (7.1) der Multiplikation der normierten Wahrscheinlichkeitswerte $P(a_1) = \hat{\epsilon}_0 \hat{\epsilon}_4 \hat{\epsilon}_8$ entlang des Pfades a_1. Hierbei ist die Annahme getroffen, dass die $\hat{\epsilon_i}$ unabhängigen Wahrscheinlichkeiten entsprechen. Die Annahme ist damit zu begründen, dass zum Zeitpunkt der Arbeit keine Daten zur Verfügung standen, die eine statistische Abhängigkeit zwischen den Eintrittswahrscheinlichkeiten von Schwachstellen aufzeigen. Die Annahme bezieht sich außerdem auf die statistische Unabhängigkeit bei einer sequentiellen Abfolge von Schwachstellen und nicht auf die statistische Unabhängigkeit der Pfade. Durch operativen Einsatz des hier vorgestellten Ansatzes wird es zukünftig möglich sein, eine ausreichend große Datensammlung zu erhalten, welche die Annahme der statistischen Unabhängigkeit validieren kann.

Das Vorgehen zur Normierung der Eintrittswahrscheinlichkeiten und der anschließenden Multiplikation lässt sich mit der Länge eines Pfades erklären. So kann grundsätzlich davon ausgegangen werden, dass mit zunehmender Länge des Angriffspfades die Eintrittswahrscheinlichkeit sinkt, da der Aufwand ansteigt. So muss der Angreifer entsprechend mehrere Angriffsschritte durchführen, was den Angriff unwahrscheinlicher macht. Die Multiplikation der Eintrittswahrscheinlichkeiten ermöglicht genau dies zu modellieren. So konvergiert die Eintrittswahrscheinlichkeit $P(a)$ gegen null, wenn die miteinander multiplizierten Eintrittswahrscheinlichkeiten $\hat{\epsilon_i} < 1$ sind. Sind hingegen alle Wahrscheinlichkeiten eines Pfades gleich $P(\epsilon_i) = 1$, so hat der Angriffspfad eine Eintrittswahrscheinlichkeit von 100%.

Die Normierung zeigt einen weiteren Vorteil, es ergibt sich für das Ergebnis der Multiplikation ebenfalls ein Wert, der im Intervall $[0, 1]$ liegt, was die Transformation in qualitative Werte vereinfacht. Konkret können damit Teilintervalle gebildet werden, die einem qualitativen Wert entsprechen. So könnte beispielsweise für die Eintrittswahrscheinlichkeit *Gering* das Intervall $[0,\ 0.3)$, für *Mittel* $[0.3,\ 0.6)$ und für *Hoch* $[0.6,\ 1]$ festgelegt werden.

7.6.2 Automatisierte Risikobewertung der Angriffspfade

Das bisher erläuterte Vorgehen impliziert, dass einzelne Angriffspfade vorliegen und bewertet werden können. Diese Voraussetzung trifft für das ausgegebene Gegenbeispiel des Model-Checkers allerdings nicht zu. Dieser erzeugt einen Angriffsgrafen und separiert die Angriffspfade nicht. Aufgrund dessen müssen die Angriffspfade nach Ausgabe des Gegenbeispiels extrahiert werden. Dies geschieht durch eine angepasste Tiefensuche (engl. depth first search) [106, Seiten 254–258], die von einer Quelle (s_0) startet, die besuchten Kanten speichert und bei Erreichen eines Ziels (s_n) den Pfad ausgibt (Algorithmus 1). Das Ziel entspricht dabei einem Blatt des Angriffsgrafen, was durch die grün-markierten Knoten in Abbildung 7.8 dargestellt ist. Um Schleifendurchläufe bei der Tiefensuche zu unterbinden, werden außerdem die Kanten markiert und nur einmalig besucht. Ein weiterer Aspekt, der für dieses Vorgehen spricht sind die erzeugten Angriffspfade. Diese werden explizit als eine Modellierungsmethode in der ISO 21434 [97] genannt, sodass die erzeugten Ergebnisse im Einklang mit der ISO 21434 sind.

Im Folgenden repräsentiert Algorithmus 1 die Tiefensuche und zeigt mit der Funktion *BewertePfadRekursiv*, dass ein rekursiver Ansatz verfolgt wird. Hierzu wird die Funktion rekursiv in der Hauptfunktion *BewerteAngriffspfade* aufgerufen. Letztere benötigt als Eingangsparameter das Gegenbeispiel des Model-Checkers in Form des gerichteten Grafen G, sowie die Eintrittswahrscheinlichkeiten $\hat{\epsilon}$ für jede Kante im Grafen G (Zeile 19 in Algorithmus 1). Außerdem müssen der Startknoten u_0 sowie der Zielknoten u_n übergeben werden. Anschließend wird der Suchpfad (*AktuellerPfad*) initialisiert und alle Knoten u als nicht besucht markiert, was die Zeilen 21 bis 23

aufzeigen. Anknüpfend wird die Funktion *BewertePfadRekursiv* aufgerufen. Diese markiert mit Zeile 2 den aktuell besuchten Knoten u, was beim ersten Aufruf dem Startknoten u_n entspricht. Außerdem wird der Knoten dem Suchpfad angehängt. Hierbei wird das Kellerprinzip (engl. stack) angewendet. Damit wächst das Speicherfeld *AktuellerPfad* sukzessive, wobei das aktuelle Element stets dem zuletzt eingefügten Knoten entspricht. In Zeile 4 wird anschließend überprüft, ob der aktuell betrachtete Knoten u_i einem Zielknoten u_n entspricht. Ist dies der Fall, werden alle bisher in *AktuellerPfad* gespeicherten Knoten in das Speicherfeld *AngriffsPfade* kopiert, da sich diese als ein Pfad zwischen u_0 und u_n darstellen. Darüber hinaus wird zu diesem Pfad ebenso seine Eintrittswahrscheinlichkeit gespeichert. Diese ergibt sich durch die sukzessive Multiplikation der Wahrscheinlichkeiten ϵ der Kanten im Pfad (Zeile 14).

Entspricht der Knoten in Zeile 4 allerdings nicht dem Ziel, muss die Suche fortgesetzt werden. Hierbei muss eine Besonderheit des Grafen in Abbildung 7.8 beachtet werden. Konkret sind es die orangefarbenen Mehrfachkanten υ_3 und υ_4, die den Aufwand der Pfadsuche beeinflussen. So ergibt sich alleinig durch die zweite Kante υ_4 ein weiterer Pfad, der ebenfalls die Zustände s_1 und s_4 betrachtet. Durch jede weitere Kante zwischen s_1 und s_4 steigt somit die Anzahl der Suchpfade und damit der Aufwand. Dies verschärft sich weiter, wenn in einem Pfad – zwischen mehreren Knoten – Mehrfachkanten durchlaufen werden müssen. Hierbei ergibt sich die Anzahl aller möglichen Pfade durch die Multiplikation aller Mehrfachkanten im Pfad, was zu einer Explosion der Pfade führen kann. Diese Problematik wird in den Zeilen 8 bis 17 von Algorithmus 1 entschärft. Hierzu werden in einem ersten Schritt alle Kanten gesucht, die zwischen dem aktuellen Knoten u_i und einem seiner Nachbarknoten u_{i+1} verlaufen. Für jede dieser Kanten wird die Eintrittswahrscheinlichkeit extrahiert ($\hat{\epsilon}_i$) und jene Kante selektiert, die den höchsten Wert aufweist. Das Vorgehen entspricht damit der Annahme, dass ein Analyst die bedrohlichste Schwachstelle aufdecken möchte, die einen Zugangsübergang $s_i \xrightarrow{\alpha} s_{i+1}$ (Ausnutzen einer Schwachstelle) ermöglicht. Der Wahrscheinlichkeitswert P_{neu} der ausgewählten Kante wird anschließend mit dem bisherigen Wert der Eintrittswahrscheinlichkeit des aktuellen Suchpfades multipliziert und ergibt den aktualisierten Wert für die Eintrittswahrscheinlichkeit P_{akt} (Zeile 15). An dieser Stelle soll der Hinweis gegeben werden, dass der Algorithmus mit diesem Ansatz stets die bedrohlichste Kante zwischen zwei Knoten für die weitere Suche auswählt[5]. Dies entspricht grundsätzlich dem Vorgehen des Dijkstra Algorithmus, allerdings kann in der gegebenen Variante flexibel zwischen 1 und n Kanten gewählt werden.

[5]Bei dem entwickelten Softwarewerkzeug kann der Anwender entweder diese Art der Optimierung wählen oder sich n bzw. alle Pfade ausgeben lassen.

Algorithmus 1 : BewertePfade (G, Startknoten u_s, Zielknoten u_z)

Data : graf $G = (U, E)$

Result : AngriffsPfade

Function *BewertePfadeRekursiv (G,u_i,AktuellerPfad,AngriffsPfade,$P_{textakt}$,u_n)*
- $u.besucht \leftarrow$ **true**
- AktuellerPfad = AktuellerPfad $\cup \{u_i\}$
- **if** $u_i = u_z$ **then**
  - $P_{Pfad} \leftarrow P_{akt}$
  - AngriffsPfade[P_{Pfad}] $\leftarrow$ AktuellerPfad
- **else**
  - **foreach** *Nachbarknoten u_{i+1} von u_i* **do**
    - **for** $e_j \in \{(u_i, u_{i+1}) \in E\}$ **do**
      - $\hat{\epsilon_j} \leftarrow P(e_j)$
      - **if** $\hat{\epsilon_j} \geq \hat{\epsilon}_{j-1}$ **then**
        - $P_{\text{neu}} \leftarrow \hat{\epsilon_j}$
      - **end**
    - **end**
    - $P_{\text{akt}} \leftarrow P_{\text{akt}} \cdot P_{\text{neu}}$
    - *BewertePfadRekursiv (G,u_i,AktuellerPfad,AngriffsPfade,P_{akt},u_z)*
  - **end**
- **end**
- AktuellerPfad = AktuellerPfad$\backslash\{u_i\}$

Function *BewertePfade (G, Startknoten u_s, Zielknoten u_z)*
- AktuellerPfad $\leftarrow \emptyset$
- AngriffsPfade $\leftarrow \{\emptyset\}$
- **foreach** $u \in U$ **do**
  - $u.besucht \leftarrow$ **false**
- **end**
- *BewertePfadeRekursiv (G,u_s,AktuellerPfad,AngriffsPfade,P_{akt},u_z)*

Der Aufwand für das Durchlaufen der Pfade und das Bestimmen der Eintrittswahrscheinlichkeiten kann hoch sein. Dies wird insbesondere von Bedeutung, sollte eine umfangreiche E/E-Architektur mit zahlreichen Elementen und einer hohen Zahl von Schwachstellen analysiert werden. Hierbei wird ebenso das ausgegebene Gegenbeispiel komplexer, und die Laufzeit der Pfadsuche kann unpraktikabel werden. Um dieser Situation entgegenzutreten, kann durch die Transponierung des Eingangsgrafen G zu $G^\intercal$ die Pfadsuche effizient parallelisiert werden. Durch die Transponierung kann die Pfadsuche – anstatt von einem Startpunkt aus – von mehreren Startpunkten zugleich begonnen werden. Hierbei besitzt der transponierte Graf $G^\intercal$ die gleiche Menge an Knoten und die gleiche Anzahl an Kanten, die allerdings in ihrer Ausrichtung

invertiert sind, sodass sich für $G^{\mathsf{T}} = (U, E^{\mathsf{T}})$ ergibt [64]. Die Menge U entspricht dabei den Knoten und die Menge E den Kanten von G beziehungsweise G^{T}. Der Aufwand für die Transponierung liegt in $\mathcal{O}(U + E)$. Hiermit gehen die Endzustände s_4, s_6, s_7, s_8, s_9 in Startzustände über, sodass der Startzustand s_0 zum neuen Endzustand wird. Das Ergebnis dieser Transposition ist mit Abbildung 7.9 gezeigt.

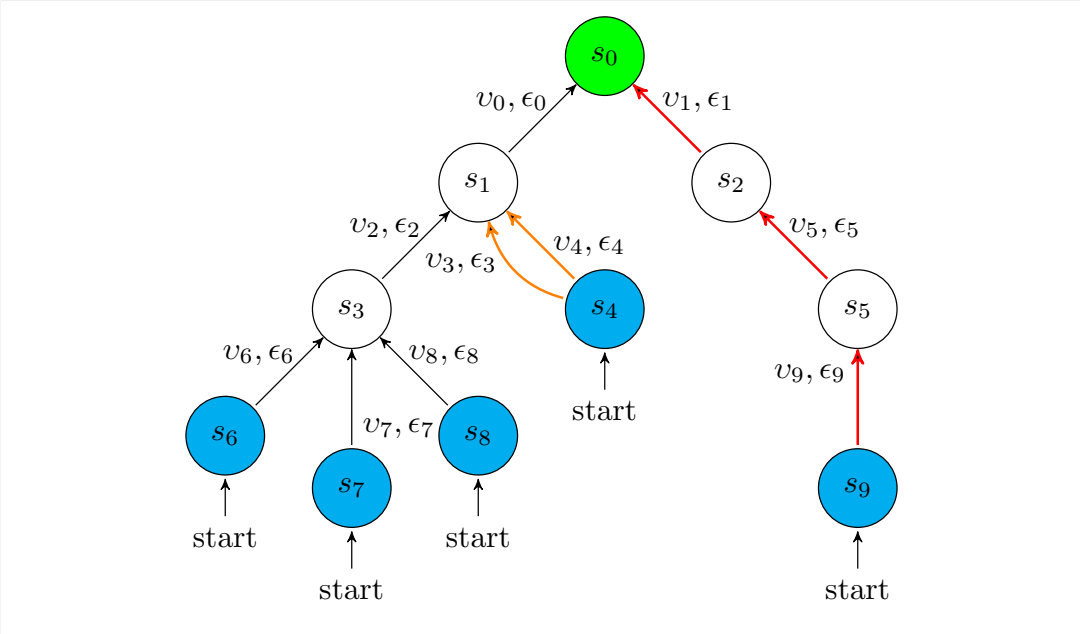


Abbildung 7.9: Eingangsgraf G^{T} der sich durch Transponierung von G aus Abbildung 7.8 ergibt. Hierbei sind die Richtungen der Kanten getauscht und die bisherigen Startzustände (s_0) werden zu Endzuständen. In gleicher Weise werden die vorherigen Endzustände (s_6, s_7, s_8, s_4, s_9) zu den neuen Startzuständen für Risikobewertung.

Hiermit ist es möglich, Algorithmus 1 in mehreren Instanzen parallel auszuführen, bei dem jeder Pfad zwischen den Startzuständen s_4, s_6, s_7, s_8, s_9 und dem Endzustand s_0 identifiziert und die Eintrittswahrscheinlichkeit P automatisiert bestimmt wird. Das Vorgehen weist dabei einen weiteren Vorteil auf. Jede Suchinstanz läuft von einem Startpunkt s_4, s_6, s_7, s_8, s_9 aus, genau auf einen Endpunkt s_0 zu. Hiermit sinkt die Wahrscheinlichkeit, dass die Suche eine Richtung einschlägt, die nicht zu einem gesuchten Endpunkt führt und irrelevante Pfade durchläuft. Dieses Vorgehen reduziert die zugrundeliegende Komplexität erwartungsgemäß nicht, verringert allerdings den Aufwand für dieses Problem, da der Ansatz mit der Problemstellung skaliert. So kann für jeden Startknoten eine Pfadsuche unabhängig von den jeweils anderen Startknoten durchgeführt werden. Dieses Vorgehen setzt das in Anhang E vorgestellte Software-Werkzeug ein, um mit Multithreading-Techniken die automatisierte Pfadsuche und Bestimmung der Risikowerte in einer praktikablen Laufzeit durchführen zu können.

7.6.3 Skalierung der Eintrittswahrscheinlichkeiten

Die bis zu diesem Zeitpunkt verwendeten Wahrscheinlichkeiten sind aus Teilmetriken der CVSS abgeleitet und können als der Grad der Schwierigkeit interpretiert werden, den ein Angreifer bewältigen muss, um die jeweilige Schwachstelle bewältigen und ausnutzen zu können. Hiermit werden dem Angreifer benötigte Fähigkeiten zur Ausnutzung der Schwachstellen zugeschrieben, was als Angreifer-Modell interpretiert werden kann. Ein weiterer Teil, der hierbei betrachtet werden sollte, ist der Eintrittspunkt des Angreifers. Konkret entspricht dies dem ersten Knoten im Angriffspfad und der Ausnutzung der ersten Schwachstelle (Wurzelknoten in Abbildung 7.8). Durch diesen wird es dem Angreifer ermöglicht, die nachfolgenden Schritte durchführen zu können, sodass dieser Knoten einen besonderen Stellenwert einnimmt. Hierbei sollte unterschieden werden, wie hoch der Aufwand für den Zugriff auf den Eintrittspunkt ist.

Für einen Angreifer ist es beispielsweise leichter eine Schwachstelle auszunutzen, die aus dem Internet erreichbar ist, als eine, die einen physikalischen Zugriff erfordert. Aufgrund dessen wird die Eintrittswahrscheinlichkeit jedes Pfades mit dem Schwierigkeitsgrad für den Erstzugriff skaliert. Hierzu zeigt Abbildung 7.10 sechs Kategorien für die Skalierung der Angriffspfade. Kann der Zugriff auf die erste Schwachstelle über eine kabellose Verbindung gelingen, wird die Eintrittswahrscheinlichkeit mit dem Faktor 1 multipliziert. Hierzu zählen neben offensichtlichen Funkverbindungen wie WLAN oder Bluetooth auch Sensoren, die andere elektromagnetische Wellen aus der Umwelt aufnehmen. Das sind beispielsweise Kameras, Light Detection and Ranging (Lidar)- und Radar-Sensoren. Ist hingegen ein physikalischer Zugriff auf eine fahrzeuginterne MMS erforderlich, wird die Wahrscheinlichkeit des Angriffspfades mit dem Faktor 0,7 multipliziert. Dies lässt sich damit begründen, dass der Aufwand für den physikalischen Zugriff die Eintrittswahrscheinlichkeit reduziert.

Muss der Angreifer tiefer in das Fahrzeug eindringen, um beispielsweise Zugriff auf interne Komponenten wie Steuergeräte, Gateways, Sensoren oder Aktuatoren zu erhalten, wird der Angriffspfad mit dem Faktor 0,5 gewichtet. Weit schwieriger als dies stellt sich ein Zugriff auf interne Netzwerke wie das CAN oder Ethernet dar. Hierbei muss der Angreifer physikalischen Zugriff auf die Verdrahtung der Netzwerke erhalten, was mit einem Entfernen von Isolationsmaterial oder Chassis-Abdeckungen einhergeht. Aufgrund dieser Situation wird nach Abbildung 7.10 die Eintrittswahrscheinlichkeit um den Faktor 0,4 multipliziert.

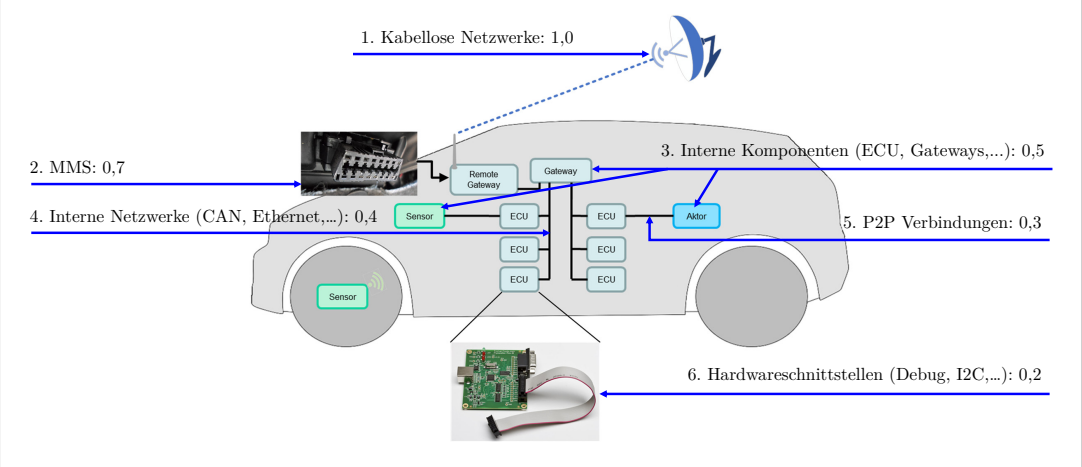


Abbildung 7.10: Schichtenmodell für die Bestimmung des Schwierigkeitsgrades beim Erstzugriff. Gezeigt sind sechs Abstufungen und deren Skalierungsfaktor: Kabellose Netzwerke 1, Mensch-Maschine-Schnittstelle (MMS) 0,7, interne Komponenten 0,5, interne Netzwerke 0,4, P2P-Verbindungen 0,3, Hardwareschnittstellen 0,2.

Neben diesen internen Netzwerken kann es für den Angreifer ebenso erforderlich sein, Zugriff auf direkte Verbindungen (P2P) zu erlangen. Dies sind beispielsweise Verbindungen, die zwischen einem Aktuator und einem Steuergerät bestehen. Ist dies der Fall, wird der Angriffspfad mit dem Faktor 0,3 skaliert. Die letzte und schwierigste Stufe des Zugriffs ist die Notwendigkeit auf Hardwareschnittstellen wie Debug, Inter-Integrated Circuit (I2C), Serial Peripheral Interface (SPI) etc. Hierbei wird die Wahrscheinlichkeit für den Angriff um den Faktor 0,2 multipliziert. Dieses Vorgehen ist mit Gleichung (7.2) und dem Skalierungsfaktor α_{v_E} zusammengefasst.

$$P(a) = \alpha_{v_E} \prod_{i=0}^{n} \hat{\epsilon}_i \ . \tag{7.2}$$

Hierbei entspricht v_E der ersten ausgenutzten Schwachstelle im Pfad und somit jener, die aus Sicht des physikalischen Zugriffs betrachtet werden muss.

Bestimmung des Risikowertes

Abschließend können die erzeugten Teilergebnisse zusammengeführt werden, um den Risikowert $R_{Bedrohung}$ eines Pfades zu bestimmen. Die dafür notwendigen Schritte sind automatisiert und werden zusammen mit der Extraktion der einzelnen Pfade aus dem Angriffsgrafen nach Algorithmus 1 durchgeführt. Um Safety-Aspekte bei der Bestimmung des Risikowertes miteinzubeziehen, wird der Schadenswert S_S anhand der Safety-Analyse abgeleitet und mit der Eintrittswahrscheinlichkeit des Pfades multipliziert. Die Schadensklassifizierung S_S setzt auf die Einteilung aus der ISO 26262 [99] (S1, S2 und S3) und wird vom Anwender über die SGM-Tabelle an das entwickelte

Softwarewerkzeug übergeben (Anhang E). Als Skalierung ist $S_0 := 10^0, S_1 := 10^1$, $S_2 := 10^2$ und $S_3 = 10^3$ gewählt, wobei die Abstufung an EVITA [167, Seite 96] angelehnt ist.

$$R_{Bedrohung} = \underbrace{S_S}_{Schaden} \underbrace{\left(\alpha_{v_E} \prod_{i=0}^{n} \hat{\epsilon_i}\right)}_{P(a)} . \tag{7.3}$$

Der Safety-Schadenswert S_S kann von der Gefährdungsanalyse (HARA) übernommen und für jede Gefährdung in der SGM-Tabelle zugeordnet werden (Tabelle 4.2). Dies ist möglich, da eine Safety-Bedrohung stets den gleichen Schaden auslöst wie die dazugehörige Gefährdung (vgl. Abbildung 4.1). Hierbei entspricht v_E der ersten ausgenutzten Schwachstelle im Angriffspfad und damit dem Eintrittspunkt für den Angreifer.

Für das am Anfang des Kapitels gezeigte Beispiel (Abbildung 7.5) mit dem Auslösen der Airbag-Ladungen ($S_S = S_3$) durch die Schwachstelle v_0 ($\hat{\epsilon_0} = 0,7$) über den OBD-Port (α_{v_E} =MMS) ergibt sich der folgende Risikowert:

$$R_{Bedrohung} = \underbrace{10^3}_{S_S} \cdot \underbrace{\underbrace{0,7}_{\alpha_{v_E}} \cdot \underbrace{0,7}_{\alpha_{v_0}}}_{P(a)} . \tag{7.4}$$

Da mehrere Pfade die gleiche Bedrohung realisieren können, ergeben sich für eine Bedrohung unterschiedliche Risikowerte. Diese werden durch das Softwarewerkzeug absteigend sortiert, um dem Security-Analysten eine priorisierte Liste von Bedrohungen zur Verfügung zu stellen.

7.6.4 Diskussion und kritische Auseinandersetzung

Für die Bestimmung der Eintrittswahrscheinlichkeiten wird die CVSS Metrik verwendet, die bereits für Schwachstellen von Fahrzeugen eingesetzt wird [214],[205, Seiten 24–26]. Diese Metrik hat sich im IT-Umfeld bereits als sinnvoll erwiesen [102] und steht im Einklang mit der ISO 21434 [97]. Mit dieser und der aufgezeigten Taxonomie [13] ist es möglich, Schwachstellen in Fahrzeugen einheitlich zu beschreiben, diese in einer Datensammlung zu speichern und sie für eine automatisierte Bedrohungsanalyse und Risikobestimmung wiederverwenden zu können. Im Gegensatz zu den in Abschnitt 3.4 betrachteten Methoden zeigt die erweiterte SGM damit eine Werkzeugunterstützung für die computergestützte Identifikation und Bewertung von mehrstufigen Bedrohungen, durch die Wiederverwendung bekannter Security-Probleme.

Kritisch zu betrachten ist, dass die Berechnung der Eintrittwahrscheinlichkeiten mit Gleichung (7.1) zum aktuellen Zeitpunkt eine Ungenauigkeit aufweist. So ist die Annahme getroffen, dass es sich um unabhängige Wahrscheinlichkeiten handelt, was nicht der Realität entsprechen muss. Allerdings wird dieses Vorgehen ebenso von Ansätzen wie beispielsweise EVITA [84] zur Risikoberechnung eingesetzt. Nichtsdestotrotz sollte diese Annahme in weiteren Forschungsarbeiten überprüft werden.

Mit Gleichung (7.3) wird der Aufwand des Angreifers für den Zugriff auf die erste Schwachstelle (Eintrittspunkt) in die Berechnung einbezogen. Die Gleichung bildet allerdings nur einen Eintrittspunkt mit dem Skalierungsfaktor α_{v_E} ab. Das bedeutet: Sollten zwei oder mehrere Eintrittspunkte zugleich ausgenutzt werden müssen, kann dies zum aktuellen Zeitpunkt nicht modelliert werden. Eine mögliche Lösung hierfür ist allerdings, dass der kleinste Skalierungsfaktor α_{v_E} jener Eintrittspunkte ausgewählt wird, die gleichzeitig ausgenutzt werden müssen. Damit entspricht das Vorgehen der Berechnung von Wahrscheinlichkeiten für einen *UND*-Knoten in einem Angriffsbaum, wie es bei EVITA [84] umgesetzt ist. Eine weitere Lösung ist das Prüfen der gleichen Bedrohung beziehungsweise Spezifikation auf der identischen E/E-Architektur, mit unterschiedlichen Eintrittspunkten. Die hierbei erzeugten Ergebnisse können im Nachgang zusammengeführt werden, um alle denkbaren Eintrittspunkte für eine bestimmte Bedrohung abzudecken. Ein weiterer Ansatz zur Lösung dieses Problems ist die Analyse der Architektur mittels des PageRank [33] Algorithmus von Google. Hiermit könnten E/E-Architekturelemente in einer Weise priorisiert werden, dass jenen Elementen das größte Risiko zugewiesen wird, durch welche die meisten Angriffspfade führen. So würde ein Gateway beispielsweise als kritischste Komponente identifiziert werden, wenn eine hohe Anzahl von Angriffspfaden die Kompromittierung dieses Elements erfordert, um schlussendlich andere Komponenten angreifen zu können. Bei diesem Ansatz ist zum Zeitpunkt der Arbeit allerdings nicht geklärt, ob dieser mit der ISO 21434 im Einklang steht.

Ebenso müssen die Werte zur Skalierung der Eintrittswahrscheinlichkeiten hinterfragt werden. Diese sind von existenten Angriffen abgleitet, aber als willkürlich anzusehen. Sie sind in jeder Risikoanalyse identisch und haben das Ziel die Ergebnisse vergleichbarer zu machen. Ungeachtet dessen ist jedoch noch zu zeigen, ob die gewählten Werte sinnvoll sind, was eine grundsätzliche Notwendigkeit bei jeder Metrik zur Risikobewertung darstellt.

Das vorgestellte Angreifermodell betrachtet den notwendigen physikalischen Zugriff auf jene Komponenten oder Verbindungen, welche die erste Schwachstelle aufweisen. Die Entscheidung für diese Modellierung kann mit mehreren Argumenten begründet werden. Ein erster Grund ist die geringere Subjektivität des benötigten Zugriffs. So kann eindeutig festgelegt werden, ob und welche Art von Zugriff benötigt wird. Die EVITA Kategorien zeigen hingegen einen größeren Spielraum für Entscheidungen. So muss beispielsweise für die Abschätzung des Zugriffs (window of opportunity) bestimmt

werden, welches Zeitfenster ein Angriff benötigt. Dies ist schwierig abzuschätzen, insbesondere, da die benötigte Zeitdauer vom Wissen des Angreifers und dessen Ausrüstung abhängt. So ist grundsätzlich anzunehmen, dass ein Angreifer mit hohem Wissensstand und hoch entwickelter Ausrüstung den Angriff schneller durchführen kann und damit eine kürzere Zeitspanne benötigt. So kann hinterfragt werden, ob die Abschätzungen für den Wissensstand und der Grad der Ausrüstung nicht in die Bestimmung für die Zeitdauer des Zugriffes einfließt, statt separat abgeschätzt zu werden. Die Bestimmung des notwendigen physikalischen Zugriffes kann im Kontrast dazu mit dem Schichtenmodell in Abbildung 7.10 für jede Schwachstelle automatisiert abgeleitet werden. Hierzu werden die bekannten Komponenten- und Verbindungsarten, die eine Schwachstelle aufweisen kann, einer der sechs Ebenen in Abbildung 7.10 zugewiesen. Damit ist es möglich, die Angriffspfade automatisiert anhand ihres benötigten Zugriffes zu skalieren. Zur einheitlichen Einordnung der Komponenten- und Verbindungsarten, werden die Klassen der Schwachstellentaxonomie aus Abschnitt 7.2 verwendet. Dies ermöglicht auch zukünftige Schwachstellen einzubeziehen, da eine einheitliche Klassifizierung existiert.

7.7 Evaluierung des Werkzeugprototyps

Zur Evaluierung der zuvor präsentierten Konzepte wurde ein softwarebasierter Prototyp mit dem Namen Automotive Security-Threat Modelling Tool (ASTMT) entwickelt, der in Anhang E detailliert beschrieben ist. Dieser unterstützt den Analysten bei der grafischen Modellierung der E/E-Architektur und Übertragung der Inhalte der SGM-Tabelle in das Werkzeug. Darüber hinaus übernimmt das Softwarewerkzeug die automatisierte Erzeugung und Analyse des formalen Modells sowie die automatisierte Durchführung der Risikowertung. Als Ergebnis erhält der Anwender einen Analysebericht, der die jeweiligen Angriffspfade mit deren Risikowerten aufzeigt. Die Evaluierung teilt sich in einen Nachweis für die Angriffspfadgenerierung in Abschnitt 7.7.1 und einer Performance Evaluierung in den Abschnitten 7.7.2 und 7.7.3 auf.

Als Fallbeispiel dient die beispielhafte E/E-Architektur mit dem Airbag-System, das in Abbildung 7.11 gezeigt ist. Insgesamt sind vier Eintrittspunkte für den Angreifer definiert, die mit rot hervorgehoben sind. Das Angriffsziel, welches durch die Airbag-Ladungen, repräsentiert ist, ist mit blau hervorgehoben.

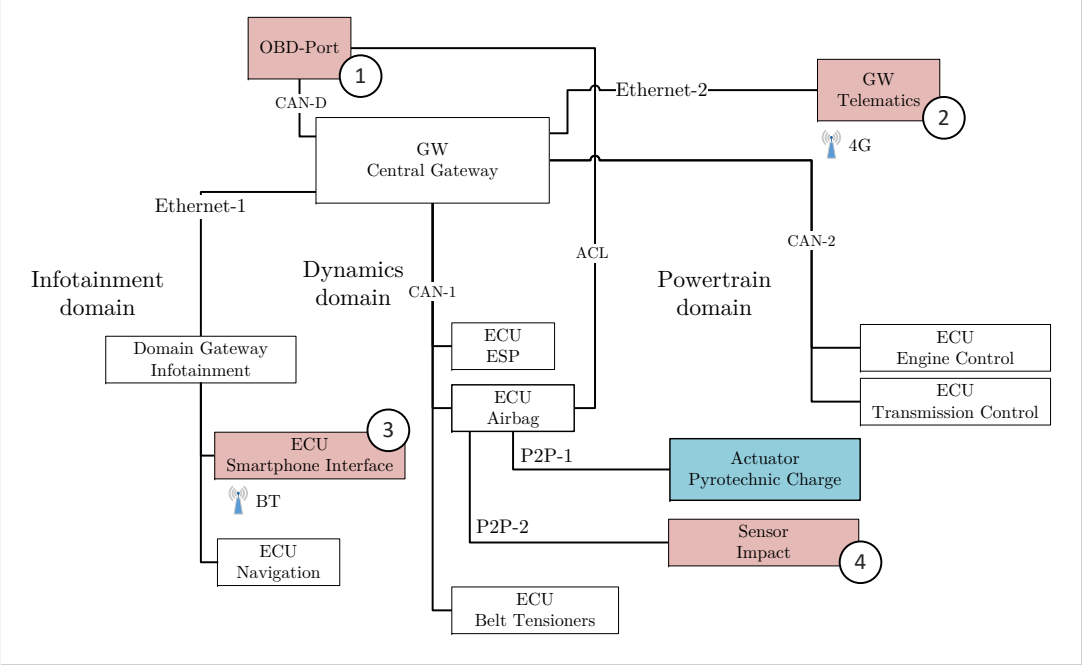


Abbildung 7.11: E/E-Architektur des Airbag-Systems, mit der zusätzlichen Kommunikationsverbindung (ACL) zwischen dem OBD-Port und dem Airbag-Steuergerät. Die Airbag-ECU erhält von den Einschlagsensoren (*Sensor Impact*) über eine direkte Verbindung (*P2P*) Sensorwerte, welche die Stärke des Einschlags bei einem Unfall messen. Anhand dessen entscheidet die Airbag-ECU, ob die Sprengkapseln der Airbag-Säcke (*Actuator Pyrotechnic Charge*) gezündet werden sollen. Die rot gefärbten Elemente entsprechen dabei den gewählten Eintrittspunkten des Angreifers und das blaue dem Ziel des Angreifers.

Die ausgewählten Eintrittspunkte zeigen dabei ein breites Spektrum von unterschiedlichen Zugriffsarten. So weist die Architektur neben schnurlosen Schnittstellen (4G und Bluetooth (BT)) mit dem OBD-Port ebenso einen Vertreter der MMS auf. Außerdem vertritt der Einschlagsensor (*Sensor Impact*) ein Architekturelement, das auf der fünften Ebene des Schichtenmodells angeordnet ist (Abbildung 7.10).

7.7.1 Bewertung der Angriffspfadgenerierung

Für die Beurteilung der erzeugten Angriffspfade stehen zwei Aspekte im Fokus. Zum einen, ob der Ansatz zur Angriffspfadgenerierung das erwartete Ergebnis liefert und zum anderen, ob der Ansatz die gewünschten Nutzungsziele erfüllt. Für Ersteres ist die Vollständigkeit der Ergebnisse ein wichtiger Aspekt, der über die eingesetzte Model Checking Technik argumentiert werden kann. So sind die Ergebnisse des Model Checkers – bezogen auf das Modell – vollständig. Für die Angriffspfade bedeutet dies, das alle aus dem Modell ableitbaren Pfade, identifiziert werden. Die Vollständigkeit des Modells hängt von der Qualität der übergebenen Schwachstellen ab. Da diese aus Publikationen extrahiert und auf ihre Richtigkeit geprüft wurden, können sie als korrekt und realistisch angesehen werden. Wird außerdem die Annahme getroffen, dass das Rechtemodell eine passende Abstraktion der Realität darstellt, ist der Ansatz in der

Lage alle Angriffspfade zu finden, die sich aus einer gegebenen Schwachstellenmenge ergeben können. Um dies exemplarisch zu überprüfen werden dem Softwarewerkzeug Schwachstellen eines durchgeführten Angriffs bereitgestellt und die automatisierte Analyse durchgeführt. Bei dem ausgewählten Angriff handelt es sich um das böswillige Zünden der Airbags, das in der Publikation [3] beschrieben und in der Datensammlung [5] gelistet ist. Die dazugehörigen Schwachstellen sind mit den Zeilen 1 und 2 in Tabelle 7.3 nochmals aufgeführt.

Tabelle 7.3: Auszug der verwendeten Schwachstellen, aus der Datensammlung [5], mit den jeweiligen Rechteleveln und der CVSS-Ausnutzbarkeit.

Beschreibung	Erforderliche Rechte	Erlangte Rechte	Element	CVSS-Ausnutzbarkeit
Brute Forcing Security Access	1	2	Airbag-ECU	2,84
Exploiting vulnerability to deploy Airbag	2	4	Airbag-ECU	2,07
Missing authorization for using OBD	1	5	OBD	0,92
General relay of data on P2P	1	2	P2P	0,92
Relay of diagnostic messages by gateway	2	4	Gateway	2,84
General relay of data on CAN	1	2	CAN	2,84
Missing authorization for using actuator	2	2	Actuator	2,52

Die Zeilen 3 bis 7 entsprechen grundlegenden Schwächen der E/E-Architektur in Abbildung 7.11, die bei der Analyse von bekannten Angriffen abgeleitet wurden und ebenfalls über die Schwachstellendatenbank in das Softwarewerkzeug geladen werden. Unter Verwendung dieser Menge von Schwachstellen und dem oben beschriebenen Rechtemodell erzeugt das Werkzeug – deterministisch – den Angriffsgrafen in Abbildung 7.12. Neben den beschriebenen Schwachstellen, ist dem Softwarewerkzeug über die SGM-Tabelle die Gefährdung *Unintended Airbag dentonation* übergeben, die sich durch das nicht autorisierte Auslösen einer Diagnosefunktion ergeben kann und durch die orangefarbenen Blätter in Abbildung 7.12 repräsentiert ist.

Der Angriffsgraf weist neben den Schwachstellen, weitere Kanten auf, die sich durch die Modellierungssprache des gewählten Model-Checkers ergeben (vgl. Anhang D.2). Aufgrund dessen weist die Ausgabe des Model-Checkers zwei Kantentypen auf, die nicht im TS von Abschnitt 7.4.1 beschrieben sind. Die Kante *jump0_7_5* zeigt beispielsweise, dass der Angreifer vom E/E-Architekturelement *OBD* (0) zum Diagnose CAN (7) wechselt. Ermöglicht wird dies anhand der ersten Schwachstelle, womit der Angreifer das höchste Rechtelevel (5) auf dem OBD-Port erlangt und damit Zugriff auf die angeschlossenen Verbindungen (*CAN_D* und *ACL*) erhält. Somit spiegelt diese Kante implizit das Rechtemodell aus Abschnitt 7.3 wieder. Anschließend wird durch die Kanten *attackerPosition7_5* aufgezeigt, dass der Angreifer Zugriff auf das Element 7 (*D_CAN*) besitzt und von diesem Zeitpunkt an, Schwachstellen dieser Komponente ausnutzen kann.

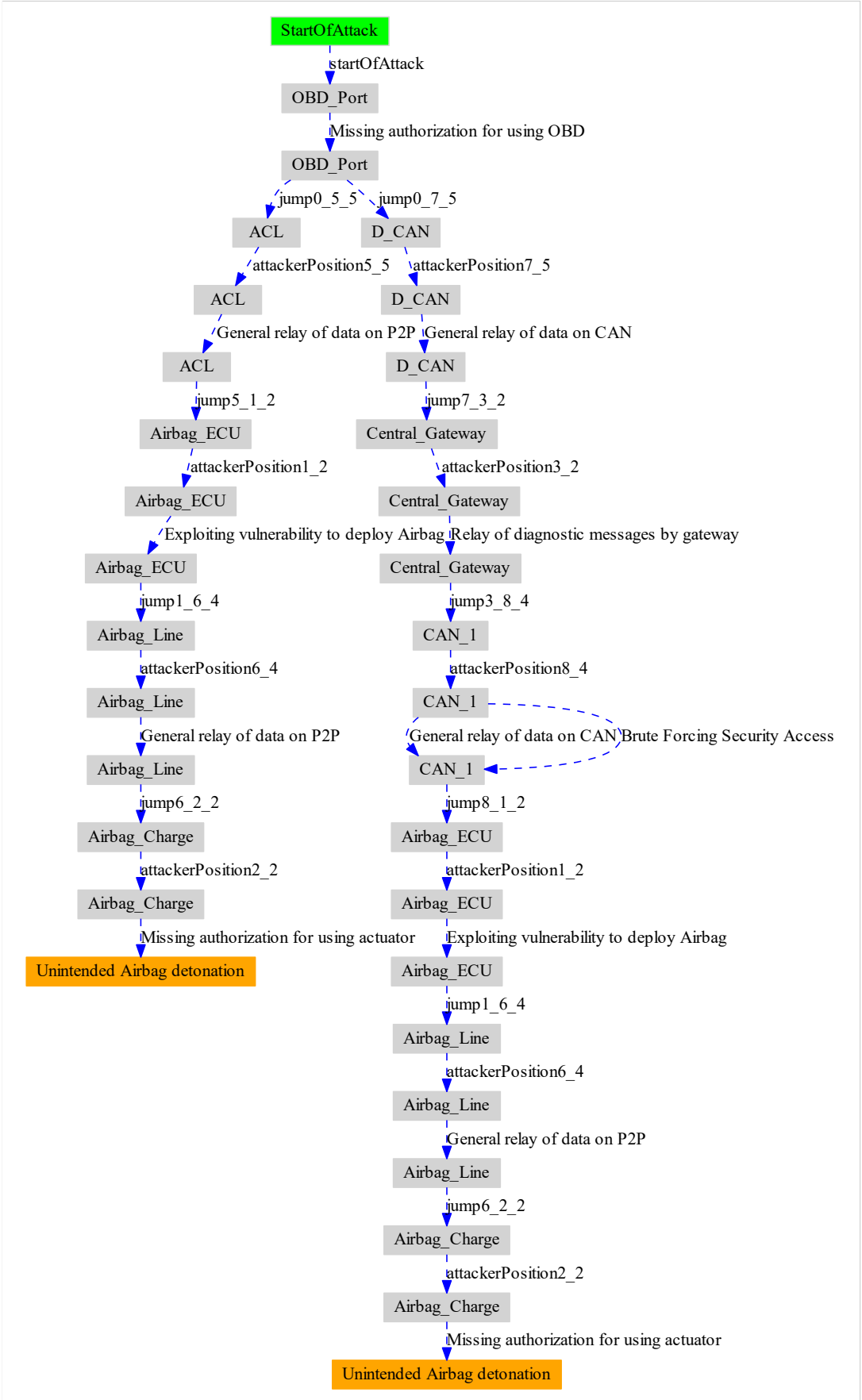


Abbildung 7.12: Erzeugter Angriffsgraf für das Airbag-Beispiel mit dem Startpunkt des Angreifers *StartofAttack* in Grün und dem Angreiferziel *Unintended Airbag detonation* in Orange.

Basierend auf diesem Grafen extrahiert das Softwarewerkzeug drei Angriffspfade, die in Tabelle 7.4 dargestellt sind und die mit der Veröffentlichung [4] der wissenschaftlichen Gemeinschaft vorgestellt sind.

Der zweite Pfad entspricht jenem Pfad, der in der Realität [3] durch die Angreifer verfolgt wurde. Der Ansatz hat somit einen Pfad gefunden, der in der Realität stattgefunden hat und der durch Gegenmaßnahmen verhindert hätte werden können.

Tabelle 7.4: Der vom ASTMT erzeugte Analyse-Report mit den drei identifizierten Angriffspfaden, den Schwachstellen in Klammern und dem jeweiligen Risikowert, bei einer Bewertung der Gefährdung *Unintended Airbag detonation* mit $S_3 = 10^3$.

Risikowert	Beschreibung des Angriffspfades
5,16	The attack path is: From OBD_Port with [Missing authorization for using OBD] ⇒ D_CAN with [General relay of data on CAN] ⇒ Central_Gateway with [Relay of diagnostic messages by gateway] ⇒ CAN_1 with [General relay of data on CAN] ⇒ Airbag_ECU with [Exploiting vulnerability to deploy Airbag] ⇒ Airbag_Line with [General relay of data on P2P] ⇒ Airbag_Charge with [Missing authorization for using the actuator] ⇒ Unintended airbag detonation.
3,76	The attack path is: From StartOfAttack with [Missing authorization for using OBD] ⇒ D_CAN with [General relay of data on CAN] ⇒ Central_Gateway with [Relay of diagnostic messages by gateway] ⇒ CAN_1 with [Brute Forcing Security Access] ⇒ with [General relay of data on CAN] ⇒ Airbag_ECU with [Exploiting vulnerability to deploy Airbag] ⇒ Airbag_Line with [General relay of data on P2P] ⇒ Airbag_Charge with [Missing authorization for using the actuator] ⇒ Unintended airbag detonation.
3,15	The attack path is: From OBD_Port with [Missing authorization for using OBD] ⇒ ACL with [General relay of data on P2P] ⇒ Airbag_ECU with [Exploiting vulnerability to deploy Airbag] ⇒ Airbag_Line with [General relay of data on P2P] ⇒ Airbag_Charge with [Missing authorization for using the actuator] ⇒ Unintended airbag detonation.

Ebenso interessant ist der dritte Pfad, der die *ACL*-Verbindung anstatt die *CAN_D* aufweist. Dieser Pfad wurde in der Publikation [3] nicht verfolgt, eine Analyse des angegriffenen Airbag-Systems zeigte allerdings, dass dieser Pfad möglich ist. Bemerkenswert ist hierbei, dass die Publikation keine ACL-Schwachstelle aufführte und das Softwarewerkzeug in der Lage war, einen noch unbekannten und nachvollziehbaren Pfad zu identifizieren. Im Hinblick auf den zeitlichen Aufwand für eine manuelle Analyse und Identifizierung von Angriffspfaden ist dieses Ergebnis hervorzuheben und unterstützt die Annahme für eine sinnvolle Anwendung des ASTMT. Ebenso ist der berechnete Risikowert nachvollziehbar, obwohl die Länge des Pfades kürzer ist als jene über *D_CAN*. So ist das Ausnutzen beziehungsweise das Verwenden einer P2P-

Verbindung – wie es die ACL-Verbindung ist – aufwendiger als eine standardisierte CAN-Verbindung. Dies spiegelt sich in der Schwachstellenbewertung und nachfolgend im Risikowert wider.

Der erste identifizierte Pfad mit dem Risikowert 5,16 zeigt eine Grenze des Modellierungsansatzes auf. So wird die Schwachstelle *General relay of data on CAN* zweimalig angewendet, obwohl das zweite Auftreten ($CAN\_1 \rightarrow Airbag\_ECU$) möglicherweise nicht zur Auslösung der Airbags führt. Das mehrmalige Anwenden der Schwachstelle ist mit der identischen Komponetenart (CAN) zu erklären und kann bei Systemen mit mehreren gleichartigen Komponenten zu Permutationen von Angriffspfaden führen, die nicht sinnvoll erscheinen. Dieses Problem kann in der Modellierung verhindert werden, indem für das Ausnutzen einer Schwachstelle beschrieben wird, welche Schwachstellen zuvor ausgenutzt sein müssen. Dies lässt die Komplexität des Modells allerdings ansteigen und die Gefahr besteht, dass Angriffspfade ausgelassen werden, da nicht alle Permutationen betrachtet werden. Grundsätzlich kann außerdem die Annahme getroffen werden, dass der Aufwand für ein manuelles Aussortieren von Angriffspfaden geringer ist als eine vollständige und manuelle Modellierung der Angriffsgrafen. Hierfür sollte zukünftige überprüft werden, wie die numerischen Risikowerte interpretiert werden können. Zum aktuellen Zeitpunkt ist nur ein relativer Vergleich der Risikowerte sinnvoll, da noch nicht ausreichend Daten zur Verfügungen stehen, um die numerischen Werte qualitativ bewerten zu können.

7.7.2 Allgemeine Performance

Neben der Bewertung der Angriffspfadgenerierung wurde ein Performance Evaluierung in zwei Schritten durchgeführt. Im ersten Schritt wurde eine allgemeine Performance Evaluierung für grundlegende Strukturen in Fahrzeugen vollzogen. Im zweiten Schritt ist eine Bewertung an dem Fallbeispiel « Airbag-System » in Abbildung 7.12 durchgeführt. Als Testsystem diente ein Lenovo Thinkpad mit Intel(R) Core(TM) i7-4800MQ @ 2,70GHz, 16GB DDR3 Arbeitsspeicher und Windows 10. Es wurden pro Testfall drei Durchläufe aufgenommen und die gemessenen Ergebnisse gemittelt.

Für die allgemeine Performancebewertung wurde überprüft, ob der Prototyp in der Lage war, die in Abschnitt 2.2 gezeigten Topologien praktikabel analysieren zu können. Hierzu wurde eine Stern-, Bus- und Ring-Topologie mit jeweils sieben ECUs generiert, die jeweils durch einen oder mehrere CAN-Busse verbunden waren. Für die Analyse der Laufzeit und des Speicherverbrauchs wurde jede Topologie mit unterschiedlichen Schwachstellenmengen evaluiert. Hierzu wurden Testfälle mit $10^1, 10^2, 10^3$ und 10^4 generischen Schwachstellen erzeugt, wobei aktuell eine Schwachstellenanzahl im Bereich von 10^1 bis 10^2 als realistisch angesehen werden kann. Dies lässt sich aus der identifizierten Anzahl an Schwachstellen erklären, die in der aufgestellten Schwachstellendatenbank [13] gelistet sind. Schwachstellen im Bereich von 10^3 bis 10^4

sind zukünftig denkbar, da sich die Fahrzeugsysteme kontinuierlich den IT-Systemen angleichen und die Anzahl an Komfortfunktionen und Schnittstellen wächst [211].

Die generischen Schwachstellen für die Bewertung der allgemeinen Performance wurden automatisiert generiert. Dies war notwendig, da die aufgebaute Datensammlung [13] aktuell noch keine ausreichende Anzahl an Schwachstellen aufweist. Die Schwachstellen wurden nicht zufällig generiert, sondern in der Weise, dass sich der größtmögliche Zustandsraum ergibt, was einer Worst-Case-Betrachtung entspricht. Hierbei entfallen jeweils 80% der Schwachstellen auf die ECUs und 20% auf den CAN-Bus. Für einen Testfall mit einer Anzahl von 10^4 Schwachstellen bedeutet dies, dass jede ECU $8 \cdot 10^3$ und jeder CAN-Bus $2 \cdot 10^3$ Schwachstellen besitzt. Als Bedrohung wurde das SGM-Leitwort « Auslösen » einer ECU-Funktion gewählt, was der CTL-Spezifikation $AG\ !(RL >= 2 \wedge I = 1)$ entspricht. Außerdem wurden der Eintrittspunkt und das Angriffsziel so gewählt, dass die größte netzwerkinterne Distanz zwischen den Steuergeräten entsteht. Grund hierfür ist es, möglichst lange Angriffspfade zu erzwingen. Für den Ringbus bedeutet dies, dass der Angreifer bei ECU 1 seinen Angriff beginnt und mit ECU 7 sein Angriffsziel erreicht. Abbildungen 7.13 bis 7.16 zeigen die Ergebnisse der Messung des ASTMT.

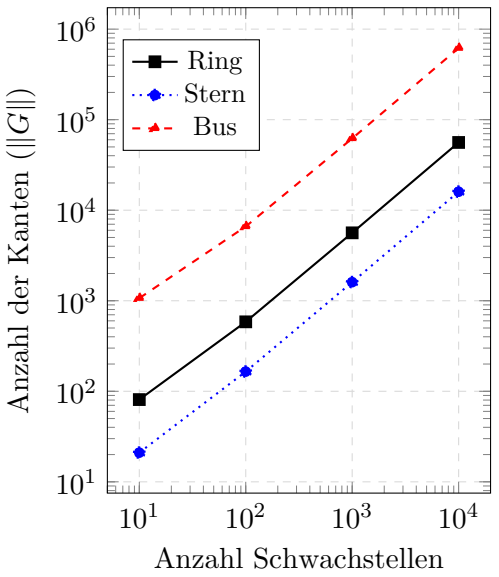


Abbildung 7.13: Anzahl der Kanten im erzeugten Angriffsgrafen bezogen auf die Menge der Schwachstellen.

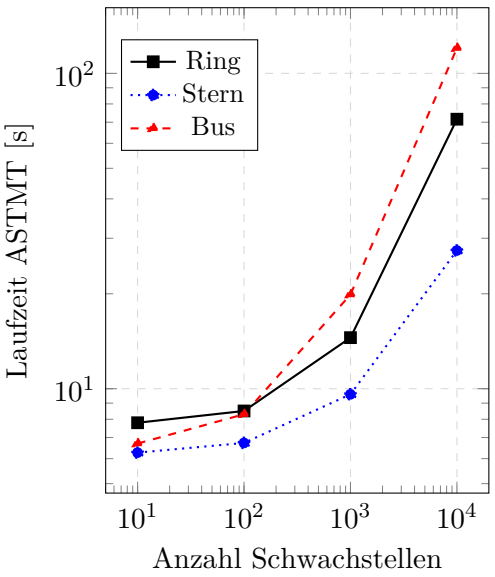


Abbildung 7.14: Laufzeit des ASTMT bezogen auf die Menge der Schwachstellen.

Die Laufzeit des ASTMT für die Bustopologie zeigt ab der Marke von 10^3 Schwachstellen einen deutlichen Anstieg der Messkurve. So wächst die benötigte Laufzeit für die Bustopologie stärker als bei den beiden anderen Topologien. Dies ist zum einen damit zu begründen, dass die interne Darstellung eines Bussystems einer vollständigen Vernetzung aller Bussteilnehmer entspricht, was zu einer großen Menge von Angriffspfaden führen kann. Damit weisen E/E-Architekturen mit zahlreichen Bussystemen eine hohe Komplexität auf und sind daher aufwendiger zu analysieren.

Abbildung 7.15 zeigt die Messung des Arbeitsspeicherverbrauchs des Model-Checkers (ITS-Tools), der bei der Erzeugung des Zustandsraums benötigt wird. In diesem Fall zeigt sich kein signifikanter Unterschied zwischen den drei Topologien. Außerdem ist der benötigte Speicher für die Abbildung von 10^4 Schwachstellen mit maximal 287 MB in einem vertretbaren Bereich und ermöglicht es auch größere Schwachstellenmengen verwenden zu können.

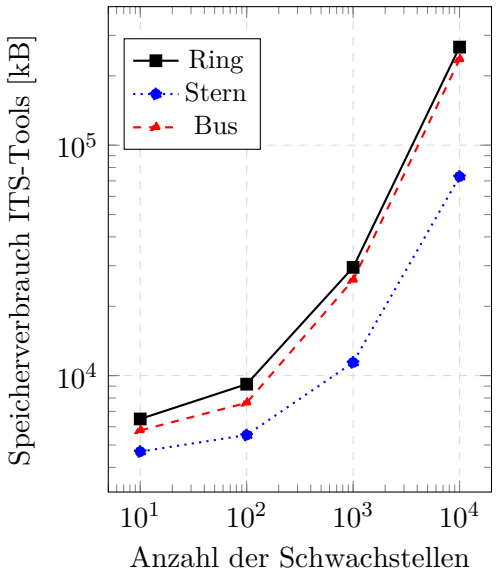


Abbildung 7.15: Speicherverbrauch der ITS-Tools bezogen auf die Menge der Schwachstellen.

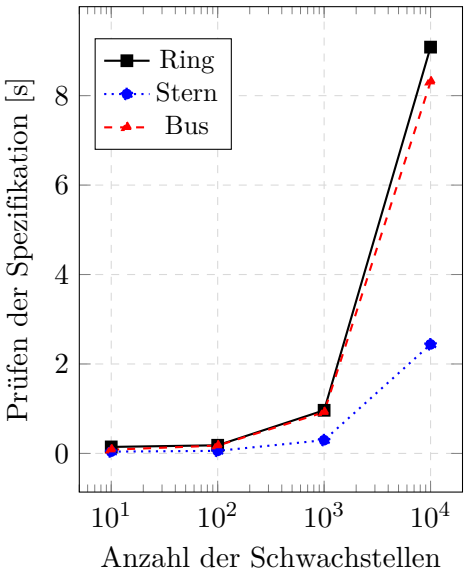


Abbildung 7.16: Die von den ITS-Tools benötigte Zeit zur Prüfung der Spezifikation, bezogen auf die Menge der Schwachstellen.

In gleicher Weise verhält es sich mit dem zeitlichen Aufwand des Model-Checkers für die Prüfung der Spezifikation. So ergibt sich beispielsweise für die Bus-Topologie mit 10^4 Schwachstellen, dass innerhalb von 8,3 s eine Verletzung der Spezifikation erkannt ist und der kürzeste Angriffspfad ausgeben werden kann. Es zeigt sich außerdem,

dass die Bus- und Ring-Topologie die größte Laufzeit aufweisen (Abbildung 7.16). Dies ist nachvollziehbar, da beide Topologien eine höhere Verbindungsanzahl als die Stern-Topologie zeigen, die zwischen jedem Busteilnehmer und Sternkoppler ausschließlich eine Verbindung aufweist. Der zeitliche Unterschied hingegen ist zwischen der erfolgreichen Überprüfung der Spezifikation und der Laufzeit des ASTMT signifikant höher (Abbildungen 7.14 und 7.16). Dies ist mit dem Umstand zu erklären, dass die Extraktion des Gegenbeispiels aus der internen Darstellung der ITS-Tools in eine Grafenstruktur sehr aufwendig sein kann. So muss für die Ausgabe aus der internen Mengendarstellung ein konkreter graf erzeugt werden, was eine Tiefensuche über die Kanten- und Knotenmenge erfordert, die eine hohe Komplexität aufweist und im aktuellen Falle nicht parallelisiert ist. Allerdings ermöglicht diese interne Darstellung, dass sowohl der Speicherverbrauch als auch der zeitliche Aufwand für die Prüfung der Spezifikation in einem praktikablen Bereich liegen.

7.7.3 Performance Evaluierung anhand des Airbag Fallbeispiels

Neben einer Performance Evaluierung für verschiedene Arten von Vernetzungen im Fahrzeug, ist der Ansatz ebenfalls an einem Fallbeispiel evaluiert.

Tabelle 7.5: Ergebnisse der Bedrohungsanalyse des Airbagsystems mit der SGM-Vorlage sowie den Ausfällen aus Tabelle 6.2. G-ID=1, entspricht der Gefährdung *ungewolltes Zünden der Airbags* aus Tabelle 6.2.

B-ID	G-ID	kann ausgelöst werden durch	Signal oder (Diagnose-) Funktion	für Komponente oder Teilsystem	Eintrittspunkt
1	1	auslösen	Zündfunktion	Airbag-Charge	OBD-Port
2	1	auslösen	Zündfunktion	Airbag-Charge	4G
3	1	auslösen	Zündfunktion	Airbag-Charge	Bluetooth
4	1	manipulieren	Unfall-Botschaft	Airbag-Charge	Impact Sensor
6	1	...	...	...	...

Neben der Voraussetzung, dass die E/E-Architektur in Abbildung 7.11 und die dazugehörige SGM-Tabelle (Tabelle 7.5) manuell vom Anwender bereitgestellt sind, werden die Schwachstellen aus der Datensammlung [5] automatisiert eingebunden. Die Datensammlung ist als Tabelle aufgebaut, bei der die Zeilen dem jeweiligen Angriff mit dessen Schwachstelle und die Spalten der in Abschnitt 7.2 beschriebenen Kategorien entsprechen. Zum Zeitpunkt der Arbeit enthält die Tabelle 413 Zeilen, die jeweils einem Angriff und einer entsprechenden Schwachstelle entsprechen. Für den Testfall in der ersten Spalte von Tabelle 7.6 bedeutet dies, dass die Zeilen 1-50 aus der Datensammlung entnommen und automatisiert für die E/E-Architektur appliziert werden. Hierzu wird für jede Schwachstelle verglichen, ob diese auf die vorliegende E/E-Architektur angewendet werden kann. Dies ist möglich, da die Datenbank für jede

Schwachstelle das zugehörige E/E-Architekturelement aufzeigt, bei der sie entdeckt wurde. Hierfür wird eine hierarchische Einteilung verwendet, welche in der Taxonomie [13] beschrieben ist (Abbildung E.2 und Abbildung E.3). Zeigt die Datenbank beispielsweise eine Schwachstelle für eine CAN-Verbindung, so wird diese Schwachstelle für jede CAN-Verbindung in der zu analysierenden E/E-Architektur appliziert. Da die Schwachstellendatenbank und das Softwarewerkzeug – für die Modellierung der E/E-Architektur – die identischen Kategorien aus der Taxonomie einsetzen, ist die automatisierte Zuordnung von Schwachstellen auf E/E-Architekturelemente möglich. Für eine detaillierte Erläuterung der automatisierten Zuordnung von Schwachstellen ist auf die Beschreibung des Softwarewerkzeugs in Anhang E verwiesen.

Mit Tabelle 7.6 sind die Ergebnisse der Performance Evaluation des Softwarewerkzeugs ASTMT zusammengefasst (Anhang E). Hierfür werden die zuvor beschriebene E/E-Architektur und SGM-Tabelle als Eingangsartefakte verwendet. Außerdem sind acht Analyseläufe mit einer ansteigenden Anzahl von Schwachstellen durchgeführt. So entspricht die letzte Spalte in Tabelle 7.6 mit 413 Schwachstellen (Zeilen in der Datenbank) dem Einbinden der vollständigen Schwachstellendatenbank. Als Ergebnis sind die erzeugten Angriffsgrafen G_i und die dafür benötigte Laufzeit dargestellt. Ein Angriffsgraf entspricht dem Gegenbeispiel des Model Checkers und die [V,E] den Ecken (V) und Kanten (E) des Angriffsgrafen. Der Index i der Angriffsgrafen (G_i) beschreibt den ausgewählten Eintrittspunkt. So entspricht der Angriffsgraf G_{OBD} dem Gegenspiel des Model Checkers bei dem der Angreifer vom OBD-Port aus seinen Angriff startet. Für die Laufzeitmessung und die Extraktion des Gegenbeispiels wurde eine Zeitgrenze von 14 Stunden festgelegt. Hierbei wurde angenommen, dass dies einem praktikablen Zeitfenster entspricht, da eine Analyse « über Nacht » akzeptabel ist.

Tabelle 7.6: Ergebnisse der Bedrohungsanalyse des Airbagsystems mit dem ASTMT. Aufgetragen sind die durch den Model-Checker erzeugten Angriffsgrafen G_i [Ecken,Kanten] sowie die Laufzeit t_i des ASTMT für die Eintrittspunkte OBD, BT, 4G und Impact (Abbildung 7.11). Rot hervorgehoben sind jene Ergebnisse, welche die Maxima in einer Messgruppe repräsentieren.

Datenbankeintrag	1-50	1-100	1-150	1-200	1-250	1-300	1-350	1-413
G_{OBD} in $[V, E]$	851, 1.048	851, 1.389	1.401, 2.492	1.823, 3.470	1.823, 3.528	1.823, 3.646	2.835, 6.134	4.143, 8.822
G_{BT} in $[V, E]$	7.010, 8.608	7.010, 11.438	20.836, 36.725	34.282, 64.651	34.282, 66.646	34.282, 67.787	225.835, 486.479	333.292, 708.952
G_{4G} in $[V, E]$	265, 322	265, 426	872, 1.523	3.924, 7.384	3.924, 7.609	5.393, 10.630	26.897, 57.931	36.803, 78.284
G_{Impact} in $[V, E]$	43, 48	43, 64	57, 80	57, 84	57, 84	57, 84	57, 100	57, 108
t_{OBD} in $[s]$	5,2	4,6	5,1	5,5	6,1	6,1	6,6	7,9
t_{BT} in $[s]$	12,3	12,1	64,7	202,3	200,1	202,2	18.621,3	41.077,3
t_{4G} in $[s]$	4,7	5	5,3	8,8	8,1	9,9	128,3	236,9
t_{Impact} in $[s]$	3,8	4,4	3,9	4,3	4,4	4,4	4,6	4,4

Die Ergebnisse in Tabelle 7.6 zeigen teilweise kein Verhältnis zwischen der Menge der eingelesenen Datenbankelemente und der Größe des Angriffsgrafen G_i. So bleiben beispielsweise für G_{OBD} die Ecken und Kante für Datenbankeinträge *1-200*, *1-250* und *1-300* gleich. Dies ist damit zu erklären, dass die Datensammlung keine gleichmäßige Verteilung der Schwachstellen aufweist. Das führt dazu, dass für bestimmte Komponenten- oder Verbindungstechnologien – in einem Datenbankintervall – keine applizierbaren Schwachstellen vorhanden sind.

Bis auf zwei Ausnahmefälle liegen die Laufzeiten unter vier Stunden. Der Eintrittspunkt *Bluetooth (BT)* mit den Datenbankeinträgen *1-350* und *1-413*, weist allerdings lange Laufzeiten auf, die in einem Stundenbereich von ≥ 10 Stunden liegen. Dies ist mit der Problematik zu begründen, dass die Extraktion des großen Gegenbeispiels G_{OBD} aus der symbolischen Darstellung des Model-Checkers aufwendig ist und mehrere Stunden dauern kann. So zeigen entsprechende Angriffsgrafen (G_i) eine hohe Anzahl von Ecken und Kanten (V,E), die zu 339.863 beziehungsweise 424.536 Pfaden bei der Angriffspfadextraktion führen. In diesem Fall ist die hohe Anzahl von Pfaden der Vernetzungskomplexität geschuldet, die sich zwischen der Bluetooth-Schnittstelle und dem Airbag-Steuergerät ergibt.

7.7.4 Diskussion und kritische Auseinandersetzung

Die Ergebnisse der Evaluierung zeigen eine hohe Anzahl von Angriffspfaden, die der Anwender betrachten muss. Dies ist als kritisch anzusehen, da keinem Anwender eine Analyse von 1000 oder mehr Angriffsgrafen zugemutet werden kann. Aufgrund dessen sind alle Angriffspfade mit einem Risikowert versehen und absteigend sortiert[6], sodass der Anwender eine priorisierte Liste von Angriffspfaden zur Verfügung gestellt bekommt. Die Angriffspfade können außerdem minimiert werden, wenn auf eine feinere Zuordnung der Schwachstellen auf die E/E-Architekturelemente gesetzt wird. So ist für die Evaluation die höchste Abstraktionsstufe der Taxonomie [13] gewählt, was beispielsweise für Komponenten eine Einteilung in nur vier Klassen (ECU, Gateways, Sensor und Aktuator) bedeutet. Die Unterteilung der Schwachstellen wie beispielsweise Motor- oder Airbag-ECU ist nicht implementiert. Das ASTMT ist allerdings dafür vorbereitet, die Zuordnung von Schwachstellen auf der zweiten Hierarchieebene der Taxonomie [13] durchführen zu können, was zu einer geringeren Anzahl von Schwachstellen pro E/E-Architekturelement führt. Hiermit wird das erzeugte Modell verkleinert und die Anzahl der Angriffspfade verringert.

Eine weitere Möglichkeit ist die Unterbindung langer Angriffspfade bei deren Extraktion aus dem Angriffsgrafen. Dies würde nicht nur die Länge der Pfade verkürzen, sondern ebenso den Aufwand für die Risikobewertung. So könnte bei Überschreiten einer maximalen Länge die weitere Suche entlang des Angriffspfades abgebrochen wer-

[6] Die Aufwände für das Bilden der Risikowerte nach Abschnitt 7.6.3 und das absteigende Sortieren aller Pfade ist bei der Evaluation mitbetrachtet.

den. Als maximale Länge kann ein fester oder ein sich adaptierender Wert in Betracht gezogen werden. Für Ersteres kann die Sammlung von Cyber-Angriffen hilfreich sein. So zeigen die bisherigen Angriffe eine maximale Anzahl von sechs Teilschritten und damit einen Ausnutzen von nicht mehr als sechs Schwachstellen. Diese Nummer kann als Richtwert für die maximale Pfadlänge dienen. Eine sich adaptierende Länge kann durch Einbezug des kürzesten Pfades erlangt werden. Dieser wird nach der Verifikation der Spezifikation vom Model-Checker ausgegeben und kann weiter verarbeitet werden. Da der kürzeste Pfad vom erzeugten Modell abhängig ist, orientiert sich ein daraus abgeleiteter Grenzwert ebenso an den Gegebenheiten des Modells, was sinnvoll erscheint.

Bezogen auf die verhältnismäßig langen Laufzeiten bei der Evaluation des Prototyps sollte die Reduktion dieser Zeiten im Vordergrund stehen. Hierbei sollte sich auf die effiziente Extraktion des Gegenbeispiels aus der internen Darstellung des Model-Checkers konzentriert werden. Erste Untersuchungen zeigen auf, dass die Verifikation der Spezifikation im Teil-Minutenbereich liegen kann und zugleich die Extraktion des Gegenbeispiels aus dem binären Entscheidungsdiagramm mehrere Stunden benötigt. Aufgrund dieser Situation und der Tatsache, dass der eingesetzte Model-Checker eine Tiefensuche für die Extraktion des Gegenbeispiels einsetzt, sollte dieser Vorgang beschleunigt werden. Hierzu kann eine verschachtelte Tiefensuche eingesetzt werden, um eine Parallelisierung zu erreichen. Erste Ansätze werden von Barnat et al. [28] gezeigt. Außerdem sollte geprüft werden, inwieweit die Extraktion des Gegenbeispiels aus der internen Darstellung des Model Checkers und die vorgestellte Risikobewertung mit Algorithmus 1 in einem Zuge stattfinden kann, um nur einmalig eine Tiefensuche durchlaufen zu müssen.

Im Vergleich zur Arbeit von Salfer et al. [131] zeigen sich ähnliche Laufzeiten, die allerdings nicht vollständig verglichen werden können, da die verwendeten Beispiele nicht gänzlich identisch sind. Ebenso wird bei Salfer et al. [131] nach einer gewissen Tiefe die Tiefensuche abgebrochen und der nächste Pfad gesucht. Außerdem wird nicht gezeigt wie die erzeugten Ergebnisse dem Anwender präsentiert werden. Aufgrund dessen ist unklar, ob in den gezeigten Laufzeiten eine Berichterzeugung beinhaltet ist.

Neben der gezeigten Performance Evaluierung sollten im Weiteren die Vollständigkeit und Korrektheit des erzeugten Angriffsgraf und den darin enthaltenen Angriffspfaden in einer empirischen Studie evaluiert werden. Hinsichtlich der Vollständigkeit kann zum aktuellen Zeitpunkt die Aussage getroffen werden, dass der Angriffsgrafen vollständig bezogen auf das Systemmodel M ist. Die Genauigkeit hängt hingegen vom Modellierungsansatz der Schwachstellen ab. Die Analyse in Abschnitt 7.7.1 und weitere Beispielanwendungen mit geringer Ecken- und Kantenzahl (≤ 300) zeigten nur eine geringe Anzahl von nicht-nachvollziehbaren Angriffspfaden.

7.8 Zusammenfassung

In diesem Kapitel wurde die methodische Erweiterung der SGM durch eine automatisierte Risikobewertung aufgezeigt. Der vorgestellte Ansatz fokussiert sich dabei auf Angriffspfade, um mehrstufige Angriffe betrachten zu können. Damit adressiert das Vorgehen die fehlende Betrachtung von mehrstufigen Angriffen in der Risikobewertung, welche die Mehrheit der vorgestellten Bedrohungsanalysemethoden aufzeigen (Abschnitt 3.4). Darüber hinaus zeigen jene Methoden, die mehrstufige Angriffe identifizieren können, keine Automatisierung dieses Vorgangs. Bezogen auf die Komplexität, die in einem Fahrzeugnetzwerk besteht, ist dies von Nachteil. Der hier entwickelte Ansatz ist in der Lage Angriffspfade automatisiert zu erzeugen, wobei die E/E-Architektur weiterhin manuell modelliert werden muss. Gesamtheitlich reduziert die gezeigte Automatisierung den gesamten Analyseaufwand. Mit diesem Vorgehen ist die Möglichkeit gegeben, mit einer gesteigerten Anzahl von Bedrohungen umzugehen, die durch die gemeinsame Betrachtung von Safety und Security entstehen können. Außerdem werden Vorteile der Model-Checking Techniken, bezogen auf die effiziente Erzeugung des Zustandsraums, ausgenutzt [27]. Zusätzlich sind die ausgegebenen Angriffspfade – bezogen auf das Systemmodell und die Spezifikation – vollständig. Dies eröffnet die Möglichkeit Analyseergebnisse zu vergleichen, da sich die Angriffspfade ausschließlich ändern, sollte das Systemmodell oder die Spezifikation geändert werden. So kann – bei einem Festhalten der Spezifikation – überprüft werden, wie sich eine Änderung des Systemmodells M auswirkt. Hiermit ist die Möglichkeit gegeben, Deltaanalysen durchführen zu können und zu bestimmen, welchen Einfluss eine neue Schwachstelle auf eine bestehende E/E-Architektur hat. Das Fehlen solch einer Fähigkeit zur Betrachtung von Änderungen wurde bei einer Untersuchung von Lisova et al. [124] als ein Defizit etablierter Bedrohungsanalysen identifiziert. Die erweiterte SGM ist grundsätzlich in der Lage, diese Unterschiede aufzuzeigen.

Aktuell verwendet der Ansatz lediglich eine geringe Anzahl von bekannten Schwachstellen, die sich jedoch mit generischen Schwachstellen erweitern lassen. Eine erste Möglichkeit hierfür wäre, die Zuordnung von CWE-Einträgen als Schwachstellen auf Fahrzeugkomponenten oder Kommunikationsverbindungen. Außerdem sollten potenzielle Schwachstellen, die während der manuellen Analyse (SGM-Tabelle) aufgedeckt wurden, in die Datensammlung eingepflegt werden. Mit diesem Vorgehen steigt die Schwachstellenzahl stetig und die Vollständigkeit der Analyseergebnisse nimmt von Analyse zu Analyse zu.

Der Ansatz bindet zum aktuellen Zeitpunkt keine Gegenmaßnahmen in die Betrachtung mit ein. Zur Erreichung dieses Ziels können allerdings zwei Ansätze verfolgt werden. Beim ersten Ansatz werden die Gegenmaßnahmen mit der Schwachstellenmenge kombiniert und in den Model-Checking Prozess miteinbezogen. Für eine gemeinsame Analyse von Schwachstellen und passenden Gegenmaßnahmen müssten in den Beendigungen zur

Ausnutzung der Schwachstelle die jeweiligen Gegenmaßnahmen gekoppelt werden. Dies müsste in einer Weise geschehen, dass die Schwachstelle nur ausgenutzt werden kann, wenn die dazugehörigen Gegenmaßnahmen als überwunden markiert sind. Der zweite Ansatz hingegen filtert jene Schwachstellen aus der Schwachstellenmenge heraus, die durch eine Gegenmaßnahme abgedeckt werden. Hierdurch wird die applizierbare Menge von Schwachstellen reduziert, was gleichzeitig den Zustandsraum verkleinert und den Model-Checking Vorgang beschleunigt. Nachteilig an dieser Lösung ist allerdings, dass die Gegenmaßnahmen nicht im Angriffsgrafen auftauchen und nachträglich eingefügt werden müssen, sollte ein Graf mit Gegenmaßnahmen benötigt werden.

8

Ergebnisse und Ausblick

In diesem Kapitel werden die Ergebnisse der Arbeit zusammengefasst und ein Ausblick auf mögliche zukünftige Arbeiten gegeben. Hierzu sind die Forschungsfragen nochmals aufgegriffen und die zugehörigen Antworten auf den Punkt gebracht. Als Ausgangspunkt der Arbeit diente eine Untersuchung existenter Bedrohungsanalyse-Methodiken, die sich eine gemeinsame Betrachtung von Safety und Security als Ziel setzte. Hierbei wurde festgestellt, dass Methoden – aus der Fahrzeugdomäne – nur in geringfügigem Umfang Safety-Artefakte in die Security-Analyse einbeziehen. Aufbauend auf dieser Situation wurden die zugrundeliegenden Beziehungen zwischen Safety und Security im Fahrzeug modelliert und eine neuartige Bedrohungsanalyse entwickelt. Hierbei entsprechen die folgenden Punkte den zentralen **Ergebnissen** der Arbeit.

- Eine Analyse und Bewertung kombinierter Safety- und Security-Analysen von CPS in Kapitel 3, die im Detailgrad über existente Analyse wie [124] und [41] hinausgeht.
- Eine detaillierte Aufbereitung der zugrundeliegenden Beziehungen zwischen Safety- und Security für CPS in Form eines Metamodells in Abschnitt 4.2, was über die in der Literatur bekannten Arbeiten [21, 68, 179] hinausgeht.
- Die Konzeption und Bereitstellung einer Bedrohungsanalyse-Methodik [2] (SGM) für eine kombinierte Safety- und Security-Analyse in Abschnitt 4.3, die Safety-Artefakte für Security-Analyse weiterverwendet und in der Literatur bisher noch nicht gezeigt ist.
- Ein empirischer und exemplarischer Nachweis für die Wirksamkeit der SGM in Kapitel 5 ([2, 3]) und dem dazu notwendigen Vorgehen, was für kombinierte Safety- und Security-Analysen von der Literatur bisher nicht betrachtet ist.

- Eine Integration der SGM in einen Penetrationstest [3] in Kapitel 6, um die Phase der Informationsbeschaffung zu steigern. Als Ergebnis konnte eine Schwachstelle in einem safety-kritischen System entdeckt werden, die bis dato unbekannt war.
- Eine Taxonomie [13] zur Einordnung von Cyber-Angriffen auf Fahrzeuge für die formale Modellierung in Abschnitt 7.2, welche in der Literatur bisher noch nicht gezeigt wurde.
- Eine Konzeption und prototypische Evaluierung der automatisierten Risikoanalysemethodik für die SGM in Abschnitt 7.7, die in dieser Form in der Literatur nicht gezeigt ist. Der Prototyp zeigt außerdem, wie bekannte Security-Schwachstellen automatisiert für eine Risikobewertung eingebunden und mit Model-Checking Techniken ausgewertet werden können. Im Vergleich zu bestehenden Ansätzen unterscheidet sich das Vorgehen insbesondere durch das formale Systemmodell aus dem die Angriffspfade automatisiert abgeleitet und bewertet werden.

Die entwickelte Bedrohungsanalyse Methodik SGM kombiniert die Safety- mit der Security-Analyse in einer Weise, dass Safety-Artefakte wiederverwendet werden können. Letztere entsprechen den identifizierten Gefährdungen, welche deduktiv auf ihre Security-Kausalfaktoren analysiert werden. Diese stellen sich als Cyber-Bedrohungen dar, die Safety-Verletzungen auslösen können. Die zur Identifizierung entwickelte Methodik entstammt der Safety-Domäne, ermöglicht allerdings, security-bezogene Sachverhalte zu identifizieren. Die hierfür notwendigen Zusammenhänge sind durch ein Metamodell beschrieben, das die Koppelung zwischen Safety und Security aufzeigt. Hiermit wird die Fragestellung beantwortet, wie eine Identifikation von Cyber-Bedrohungen gestaltet sein muss, um safety-relevante Cyber-Bedrohungen identifizieren zu können. Darüber hinaus führt die Koppelung von Gefährdungen mit Security-Kausalfaktoren zu neuen und in der Safety-Analyse nicht betrachteten Gefährdungsfällen. Hiermit ist eine Steigerung der Anzahl von Gefährdungssituationen gegeben, die im weiteren Safety-Prozess analysiert und abgeschwächt werden können, was zu einer Steigerung der Betriebssicherheit führt und damit die Hauptforschungsfrage *Wie lassen sich Bedrohungsanalysen aus der Informationssicherheit (IT) mit Gefährdungsanalysen kombinieren, um die Betriebssicherheit von Personenkraftwagen zu steigern?* (HF) adressiert. Die entwickelte SGM vermittelt – neben der Wiederverwendung von Safety-Artefakten – außerdem ein gemeinsames Domänenverständnis und fördert den Austausch zwischen beiden Studienfeldern.

Die sinnvolle Anwendung der Bedrohungsanalyse-Methodik wurde anhand einer empirischen Evaluation mit Safety- und Security-Ingenieuren aufgezeigt. Hierzu analysierten zwei Testgruppen ein safety-kritisches System, wobei beiden Gruppen eine durchgeführte Safety-Analyse und die darin identifizierten Gefährdungen zur Verfügung standen. Die Testpersonen der Safety-Gruppe waren dabei in der Lage, gleichwertige Ergebnisse bei der Identifikation von safety-relevanten Bedrohungen zu erreichen wie

die Security-Gruppe. Damit kann die Aussage getroffen werden, dass Safety-Ingenieure die SGM einsetzen können, um Security-Ingenieure bei einer gemeinsamen Safety- und Security-Analyse zu unterstützen. Hierbei ersetzen die Safety-Ingenieure nicht die Security-Ingenieure, sondern unterstützen durch die Identifikation von safety-relevanten Bedrohungen. Diese können anschließend an die Security-Analysten übergeben werden, die anknüpfend Gegenmaßnahmen entwickeln können. Hiermit ist gezeigt, dass die SGM bei der Bedrohungsidentifikation unterstützt, was die Teilforschungsfrage 1 *Wie können safety-relevante Cyber-Bedrohungen systematisch identifiziert werden?* adressiert. Bei der exemplarischen Anwendung der SGM auf ein Airbagsystem das in vielen gängigen Fahrzeugen eingesetzt wird, konnte außerdem eine Schwachstelle aufgedeckt werden, die das Zünden von Airbags ermöglicht. Dies untermauert den Nutzen der SGM und stellt unabhängig davon einen wertvollen Beitrag für die Automotive Security Community dar. Der Community wurden mit den Veröffentlichungen [8, 11, 12] außerdem fahrzeugspezifische Maßnahmen präsentiert, welche die Airbag-Schwachstelle abschwächen.

Aufgrund der Tatsache, dass bei der Koppelung der Safety- und Security-Analyse ein Teil der zur Risikobewertung eingesetzten Parameter disjunkt sind, wurden neue Kriterien bestimmt. Eines dieser Kriterien ist der Aufwand des Angreifers, den er für die erfolgreiche Durchführung des Angriffs benötigt. Dies ist abhängig von der Ausnutzbarkeit der jeweiligen Schwachstellen, dem Aufwand für den Zugriff auf diese Schwachstelle sowie der Anzahl der auszunutzenden Schwachstellen. Zur Bestimmung dieser Teilparameter wurde eine umfangreiche Analyse existenter Cyber-Angriffe durchgeführt und eine Taxonomie entwickelt, welche die einheitliche Extraktion und Bestimmung der Parameter erlaubt. Auf Basis dieser Taxonomie und der daraus entstanden Datenbank mit insgesamt 413 Angriffsschritten [5], konnte außerdem ein automatisiertes Werkzeug für die Risikobewertung entwickelt werden. Das grundlegende Konzept orientiert sich an einem modellbasierten Ansatz, der die informationsverarbeitenden Komponenten einer E/E-Architektur unterschiedlichen Klassen zuweist und auf dessen Basis relevante Schwachstellen zuordnet. Hierdurch wird ein formales Modell erzeugt, welches die Vernetzung von Fahrzeugkomponenten mit deren Schwachstellen widerspiegelt. Durch den Einsatz von Model-Checking-Techniken wird es anschließend möglich, alle Angriffspfade zu bestimmen, die zwischen einem Eintrittspunkt und dem Angreiferziel existieren. Hiermit wird der Analyst bei der Durchführung unterstützt und es wird ihm ermöglicht, ebenso mit einer gesteigerten Anzahl von Cyber-Bedrohungen umgehen zu können. Dies wird insbesondere durch die Automatisierung erreicht, was den Aufwand für die gemeinsame Betrachtung von Safety- und Security-Situationen verringert und damit die Teilforschungsfrage 2 *Wie kann der gesteigerte Aufwand für den Anwender reduziert werden, der sich durch die gemeinsame Betrachtung von Safety und Security ergibt?* beantwortet. Mittels modifizierter Grafenalgorithmen, der Einbindung der Ausnutzbarkeitswerte für die Schwachstellen und des Safety-Schadenswertes, ist

es außerdem möglich jeden Angriffspfad zu bewerten und eine priorisierte Liste zu erstellen. Da der Safety-Wert direkt in die Risikobestimmung mit einfließt wird Teilforschungsfrage 3 *Wie lässt sich eine Priorisierung und Risikobewertung identifizierter Cyber-Bedrohungen hinsichtlich Safety verwirklichen?* beantwortet, welche die Frage nach einer safety-bezogenen Risikobewertung für Cyber-Bedrohungen stellt.

Gesamtheitlich kann gesagt werden, dass die vorgestellte Bedrohungsanalyse-Methodik eine Koppelung von Security und Safety ermöglicht und durch die Betrachtung von Security-Kausalfaktoren für Gefährdungen ein Beitrag zur Steigerung der Betriebssicherheit geleistet wird. Die dazu entwickelte Risikobewertung basiert auf einem modellbasierten Ansatz und lässt sich konzeptionell auf andere Domänen übertragen. Außerdem bietet die vorgestellte Sammlung und Taxonomie von Cyber-Angriffen auf Fahrzeuge ein Fundament für weitere Arbeiten im Automotive Security Umfeld.

In einem Fokus **zukünftiger Arbeiten** sollte die Erweiterung der Methodik stehen. Insbesondere die Übertragung der SGM auf andere Domänen sollte verfolgt werden. Ein erster Schritt in diese Richtung wurde bereits mit einem Projekt erreicht, bei dem die SGM und das Six-Step Model (SSM) kombiniert wurden, um eine gemeinsame Safety- und Security-Analyse in der Chemieindustrie zu etablieren. Hierzu wurde mit dem CSE-Institut [49] kooperiert und die kombinierte Methodik erfolgreich in einem Arbeitskreis von Herstellern für Anlagentechnik sowie für Messsensorik und Aktorik vorgestellt. Das positive Feedback führte zur Entscheidung für eine wissenschaftliche Veröffentlichung, die voraussichtlich Ende 2021 den Prozess zur Veröffentlichung durchlaufen haben wird. Inhaltlich werden neben der Kombination aus SGM und SSM ebenso neue Leitwörter präsentiert, die Domänen spezifisch sind. Zur Übertragung auf andere Domänen sollten daher weitere spezifische Leitwörter festgelegt und evaluiert werden.

In diesem Kontext wäre es ebenfalls sinnvoll, wenn neue Forschungsvorhaben ihre präsentierten Methodiken nach aussagekräftigen Kriterien evaluieren würden, um einen direkten Vergleich zwischen Methodiken zu ermöglichen, was zum aktuellen Zeitpunkt dieser Arbeit nicht gegeben ist. Hierzu kann das in Kapitel 5 gezeigte Vorgehen als ein möglicher Ausgangspunkt dienen. Darüber hinaus sollten auch die SGM in weiteren Projekten eingesetzt, gemessen und mit den bekannten Ansätzen verglichen werden. Nur so kann ein Reifeprozess etabliert werden, der dazu führt, dass die SGM auch zukünftig sinnvoll anzuwenden ist. Ein weiterer Fokus sollte auf die Einbindung von Gegenmaßnahmen gesetzt werden, die zum aktuellen Zeitpunkt nicht betrachtet sind. Bei solch einer Integration sollte außerdem ein Vorgehen gezeigt werden, das Wechselwirkungen mit Safety-Maßnahmen aufdeckt und auflöst. Hierzu ist es allerdings notwendig, die – im Security-Konzept – festgelegten Maßnahmen zurückführen zu können, da insbesondere die Safety-Maßnahmen zum Zeitpunkt einer Bedrohungsanalyse nicht bekannt sind.

Zur Einbindung von bekannten Security-Maßnahmen sollten in der Bedrohungsanalyse grundsätzlich zwei Ansätze verfolgt werden. Beim ersten Ansatz werden die Gegenmaßnahmen mit der Schwachstellenmenge kombiniert und in den Model-Checking Prozess miteinbezogen. Der zweite Ansatz hingegen filtert jene Schwachstellen aus der Schwachstellenmenge heraus, die durch eine Gegenmaßnahme abgedeckt werden. Hierdurch wird die applizierbare Menge von Schwachstellen reduziert, was gleichzeitig den Zustandsraum verkleinert und den Model-Checking Vorgang beschleunigt. Generell sollte im Fokus weiterer Arbeiten die Reduktion der Laufzeiten im Vordergrund stehen. Hierbei sollte sich auf die effiziente Extraktion des Gegenbeispiels aus der internen Darstellung des Model-Checkers konzentriert werden. Hierzu kann beispielsweise eine verschachtelte Tiefensuche eingesetzt werden, um eine Parallelisierung der Extraktion zu erreichen. Außerdem können lange Pfade bei der Tiefensuche durch eine Längenbeschränkung unterbunden werden. So könnte bei Überschreiten eines Limits die weitere Suche entlang des Pfads abgebrochen werden. Hierzu kann eine feste oder sich adaptierende Längenbeschränkung dienen. Darüber hinaus kann durch ein Bilden von Schwachstellengruppen, die Anzahl der zu prüfenden Schwachstellen reduziert werden, um die Berechnung weiter zu beschleunigen. Eine geeignete Gruppierung kann beispielsweise anhand der notwendigen und zu erreichenden Rechte gebildet werden. Hierbei werden jene Schwachstellen zusammengefasst, die gleiche Rechtelevel für die Vor- und Nachbedingungen aufweisen.

Gesamtheitlich kann gesagt werden, dass die hier vorgestellten Maßnahmen, die SGM in einer Weise erweitern können, sodass eine Analyse zukünftiger E/E-Architekturen mit hohen Schwachstellenzahlen möglich ist. Dies ist im Hinblick auf den weiteren Aufbau der Angriffsdatenbank von Relevanz, die zukünftig eine Vielzahl von Schwachstellen aufzeigen könnte. Da sich die vorliegende Arbeit auf den Entwicklungsschritt « Bedrohungsanalyse » fokussiert, die Interaktionen zwischen Safety und Security sich allerdings weiter durch den Entwicklungsprozess ziehen, sollten zukünftig auch hierfür Lösungen gefunden werden. Zu diesem Zweck müssen gegebenenfalls Zielkonflikte erkannt und aufgelöst, sowie weitere Synergiepotenziale zwischen Safety und Security ausgenutzt werden. Die SGM hat hierzu einen Beitrag geleistet, das Gesamtproblem und -optimierungspotenzial stellt sich allerdings größer da. So sollte bei der Entwicklung des Safety- und Security-Konzeptes ebenso eine Schnittstelle zum Austausch von Artefakten etabliert werden. Insbesondere die Identifikation von Synergieeffekten und Wechselwirkungen zwischen Safety- und Security-Maßnahmen sollte betrachtet werden, um nachgelagerte Aufwände zu minimieren. Außerdem können Synergieeffekte für die in der Entwicklung geforderten Testabläufe erzielt werden.

Literaturverzeichnis

[14] Sridhar Adepu und Aditya Mathur. „Distributed Detection of Single-Stage Multipoint Cyber Attacks in a Water Treatment Plant“. In: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security - ASIA CCS '16*. Herausgegeben von Xiaofeng Chen, XiaoFeng Wang und Xinyi Huang. ACM Press, 2016, Seiten 449–460. ISBN: 9781450342339. DOI: 10.1145/2897845.2897855 (siehe Seiten 59, 63, 64, 224, 225).

[15] Amer Aijaz, Bernd Bochow, Florian Dötzer, Andreas Festag, Matthias Gerlach, Rainer Kroh und Tim Leinmüller. „Attacks on Inter Vehicle Communication Systems-an Analysis“. In: *Proceedings WIT* (2006) (siehe Seite 49).

[16] Ravi Akella, Han Tang und Bruce M. McMillin. „Analysis of information flow security in cyber--physical systems“. In: *International Journal of Critical Infrastructure Protection* 3.3-4 (2010), Seiten 157–173. DOI: 10.1016/j.ijcip.2010.09.001 (siehe Seiten 59, 65).

[17] Akers. „Binary Decision Diagrams“. In: *IEEE Transactions on Computers* C-27.6 (1978), Seiten 509–516. ISSN: 0018-9340. DOI: 10.1109/TC.1978.1675141 (siehe Seite 145).

[18] Christopher J Alberts, Sandra G Behrens, Richard D Pethia und William R Wilson. *Operationally critical threat, asset, and vulnerability evaluation (OCTAVE) framework, Version 1.0*. Technischer Bericht. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 1999 (siehe Seite 59).

[19] Ian Alexander. „Initial industrial experience of misuse cases in trade-off analysis“. In: *Proceedings IEEE Joint International Conference on Requirements Engineering*. 2002, Seiten 61–68 (siehe Seiten 59, 222).

[20] Husain Aljazzar, Manuel Fischer, Lars Grunske, Matthias Kuntz, Florian Leitner-Fischer und Stefan Leue. „Safety analysis of an airbag system using probabilistic FMEA and probabilistic counterexamples“. In: *2009 Sixth International Conference on the Quantitative Evaluation of Systems*. IEEE. 2009, Seiten 299–308 (siehe Seite 129).

[21] Lautenbach Aljoscha und Islam Mafijul. *HEAVENS -- HEAling Vulnerabilities to ENhance Software Security and Safet: Deliverable D2 - Security Models*.

http://autosec.se/wp-content/uploads/2018/03/HEAVENS_D2_v2.0.pdf. f2016 (siehe Seiten 59, 70, 179, 238–240).

[22] Abdullah Altawairqi und Manuel Maarek. „Attack Modeling for System Security Analysis“. In: *Computer Safety, Reliability, and Security*. Herausgegeben von Stefano Tonetta, Erwin Schoitsch und Friedemann Bitsch. Band 10489. Lecture Notes in Computer Science. Springer International Publishing, 2017, Seiten 81–86. ISBN: 978-3-319-66283-1. DOI: 10.1007/978-3-319-66284-88 (siehe Seiten 59, 61, 66).

[23] Jason Andress. *The basics of information security: understanding the fundamentals of InfoSec in theory and practice: Second Edition*. Syngress, 2014. ISBN: 9780128008126 (siehe Seite 91).

[24] Julieth Patricia Castellanos Ardila und Barbara Gallina. „Towards efficiently checking compliance against automotive security and safety standards“. In: *2017 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. 2017, Seiten 317–324 (siehe Seite 55).

[25] A. Avizienis, J.-C. Laprie, B. Randell und C. Landwehr. „Basic concepts and taxonomy of dependable and secure computing“. In: *IEEE Transactions on Dependable and Secure Computing* 1.1 (2004), Seiten 11–33. ISSN: 1545-5971. DOI: 10.1109/TDSC.2004.2 (siehe Seite 21).

[26] Christel Baier, Joost-Pieter Katoen u. a. *Principles of model checking*. MIT press Cambridge, 2008. ISBN: 9780262026499 (siehe Seiten 144–147, 150).

[27] Roberto Baldoni, Emilio Coppa, Daniele Cono D'elia, Camil Demetrescu und Irene Finocchi. „A survey of symbolic execution techniques“. In: *ACM Computing Surveys (CSUR)* 51.3 (2018), Seiten 1–39 (siehe Seiten 145, 177).

[28] Jiri Barnat, Vincent Bloemen, Alexandre Duret-Lutz, Alfons Laarman, Laure Petrucci, Jaco van de Pol und Etienne Renault. „Parallel model checking algorithms for linear-time temporal logic“. In: *Handbook of Parallel Constraint Reasoning*. Springer, 2018, Seiten 457–507 (siehe Seite 176).

[29] Kristian Beckers, Dominik Holling, Isabelle Côté und Denis Hatebur. „A Structured Hazard Analysis and Risk Assessment Method for Automotive Systems-A Descriptive Study“. In: *Reliability Engineering & System Safety* (2016). ISSN: 09518320 (siehe Seiten 109, 113).

[30] Ali Behfarnia und Ali Eslami. „Risk assessment of autonomous vehicles using bayesian defense graphs“. In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE. 2018, Seiten 1–5 (siehe Seite 154).

[31] Kai Borgeest. *Elektronik in der Fahrzeugtechnik*. Springer Fachmedien Wiesbaden, 2014. ISBN: 978-3-8348-1642-9. DOI: 10.1007/978-3-8348-2145-4 (siehe Seite 275).

[32] Sarah Boslaugh. *Statistics in a nutshell: A deskop quick reference*. 2. Aufl. O'Reilly, 2013. ISBN: 9781449316822 (siehe Seiten 109, 118, 119, 257).

[33] Sergey Brin und Lawrence Page. „The anatomy of a large-scale hypertextual web search engine". In: *Computer networks and ISDN systems* 30.1-7 (1998), Seiten 107–117 (siehe Seite 164).

[34] Florent Brissaud, Anne Barros, Christophe Bérenguer und Dominique Charpentier. „Reliability Study of an Intelligent Transmitter". In: *15th ISSAT International Conference on Reliability and Quality in Design* (2009) (siehe Seiten 59, 63, 64).

[35] Randal E Bryant. „Graph-based algorithms for boolean function manipulation". In: *IEEE Transactions on Computers* 100.8 (1986), Seiten 677–691 (siehe Seite 145).

[36] A. Burns, J. McDermid und J. Dobson. „On the Meaning of Safety and Security". In: *The Computer Journal* 35.1 (1992), Seiten 3–15. DOI: 10.1093/comjnl/35.1.3 (siehe Seite 39).

[37] Giovanni Cagalaban, Taihoon Kim und Seoksoo Kim. „Improving SCADA control systems security with software vulnerability analysis". In: *Proceedings of the 12th WSEAS international conference on Automatic control, modelling & simulation*. 2010, Seiten 409–414 (siehe Seite 49).

[38] Yue Chen, Barry Boehm und Luke Sheppard. „Value driven security threat modeling based on attack path analysis". In: *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*. 2007, 280a–280a (siehe Seite 154).

[39] Pengsu Cheng, Lingyu Wang, Sushil Jajodia und Anoop Singhal. „Aggregating CVSS Base Scores for Semantics-Rich Network Security Metrics". In: *Proceedings, 2012 31st International Symposium on Reliable Distributed Systems*. IEEE Computer Society, 2012, Seiten 31–40. ISBN: 978-1-4673-2397-0. DOI: 10.1109/SRDS.2012.4 (siehe Seite 52).

[40] Yulia Cherdantseva und Jeremy Hilton. „A reference model of information assurance & security". In: *2013 International Conference on Availability, Reliability and Security*. IEEE. 2013, Seiten 546–555 (siehe Seiten 59, 69).

[41] Sabarathinam Chockalingam, Dina Hadžiosmanović, Wolter Pieters, André Teixeira und Pieter van Gelder. „Integrated safety and security risk assessment methods: A survey of key characteristics and applications". In: *International Conference on Critical Information Infrastructures Security*. 2016, Seiten 50–62 (siehe Seiten 30, 59, 61, 62, 66, 71, 179, 227).

[42] Thomas Chowdhury, Eric Lesiuta, Kerianne Rikley, Chung-Wei Lin, Eunsuk Kang, BaekGyu Kim, Shinichi Shiraishi, Mark Lawford und Alan Wassyng. „Safe and Secure Automotive Over-the-Air Updates“. In: *Developments in Language Theory*. Herausgegeben von Mizuho Hoshi und Shinnosuke Seki. Band 11088. Lecture Notes in Computer Science. Springer International Publishing, 2018, Seiten 172–187. ISBN: 978-3-319-98653-1. DOI: 10.1007/978-3-319-99130-612 (siehe Seite 59).

[43] Gianfranco Ciardo und Radu Siminiceanu. „Structural Symbolic CTL Model Checking of Asynchronous Systems“. In: *Computer aided verification*. Herausgegeben von Warren A. Hunt und Fabio Somenzi. Band 2725. Lecture Notes in Computer Science. Springer, 2003, Seiten 40–53. ISBN: 978-3-540-40524-5. DOI: 10.1007/978-3-540-45069-64 (siehe Seite 145).

[44] Edmund M Clarke, William Klieber, Miloš Nováček und Paolo Zuliani. „Model checking and the state explosion problem“. In: *LASER Summer School on Software Engineering*. Springer. 2011, Seiten 1–30 (siehe Seite 144).

[45] Sielaff Clemens. *ZodiacGraph: A general-purpose, circular node graph GUI using Qt*. https://github.com/clemenssielaff/ZodiacGraph. 2017 (siehe Seite 295).

[46] International Electrotechnical Commission. *Hazard and Operability Studies (HAZOP studies): ISO/IEC 62882*. 2005 (siehe Seite 92).

[47] International Electrotechnical Commission. *IEC 61508: 2010*. 2010 (siehe Seiten 21, 59).

[48] Frank Crawley und Brian Tyler. *HAZOP: Guide to best practice*. Elsevier, 2015 (siehe Seite 62).

[49] CSE. *CSE-Institute*. https://cse-institut.de/?lang=en. 2020 (siehe Seite 182).

[50] Jin Cui und Giedre Sabaliauskaite. „US2: An Unified Safety and Security Analysis Method for Autonomous Vehicles“. In: *Advances in Information and Communication Networks*. Herausgegeben von Kohei Arai, Supriya Kapoor und Rahul Bhatia. Band 886. Springer International Publishing, 2019, Seiten 600–611. ISBN: 978-3-030-03401-6. DOI: 10.1007/978-3-030-03402-3-42 (siehe Seiten 59, 68, 70).

[51] CVE. *Common Vulnerabilities and Exposures (CVE): The Standard for Information Security Vulnerability Names*. https://cve.mitre.org/ (siehe Seiten 30, 142).

[52] *CVE-2017-14937*. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-14937. 2017 (siehe Seiten 26, 31, 137).

[53] CVSS Special Interest Group. *Common Vulnerability Scoring System v3.0: Specification Document*. Herausgegeben von Inc. FIRST.org. `https://www.first.org/cvss/specification-document`. 2019 (siehe Seite 142).

[54] Burzin Daruwala, Salvador Mandujano, Narasimha Kumar Mangipudi und Hao-chi Wong. „Threat analysis for hardware and software products using HazOP“. In: *Proceedings of the international Conference on Computational and information Science*. 2009, Seiten 446–453 (siehe Seite 91).

[55] Joost CF De Winter. „Using the Student's t-test with extremely small sample sizes“. In: *Practical Assessment, Research, and Evaluation* 18.1 (2013), Seite 10 (siehe Seiten 118, 121).

[56] Marco Di Natale. *SAFURE - SAFety and security by design for interconnected mixed-critical cyber-physical systems: D2.1 D2.1 Architecture models and patterns for safety and security (Alpha)*. `https://safure.eu/downloads/SAFURE-D2.1-PU-M12.pdf`. 2016 (siehe Seiten 59, 71).

[57] DIN. *Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1: General requirements*. 2011 (siehe Seiten 41, 42).

[58] Claudia Eckert. *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. 10. Auflage. De Gruyter Studium. De Gruyter Oldenbourg, 2018. ISBN: 9783110551587 (siehe Seite 47).

[59] Stefan Engeser und Regina Vollmeyer. *Tätigkeitsanreize und Flow-Erleben*. na, 2005 (siehe Seite 259).

[60] ENISA. *Good Practices for Security of Smart Cars*. `https://www.enisa.europa.eu/publications/smart-cars`. 2019 (siehe Seite 30).

[61] Michael Eraut. *Developing professional knowledge and competence*. Falmer Press, 1994. ISBN: 9780750703314 (siehe Seite 245).

[62] Clifton A. Ericson II. *Hazard analysis techniques for system safety*. Wiley-Interscience, 2005. ISBN: 0471720194 (siehe Seiten 45–47, 91, 92, 95).

[63] Torkildson Erik Nilsen, Jingyue Li, Stig Ole Johnsen und Jon Arne Glomsrud. „Empirical studies of methods for safety and security co-analysis of autonomous boat“. In: *Safety and Reliability-Safe Societies in a Changing World* (2018) (siehe Seiten 59, 67, 230).

[64] John W Essam und Michael E Fisher. „Some basic definitions in graph theory“. In: *Reviews of Modern Physics* 42.2 (1970), Seite 271 (siehe Seite 160).

[65] T. S. ETSI. „102 165-1: CYBER Methods and Protocols. Part 1: Method and Pro Forma for Threat, Vulnerability“. In: *Risk Analysis (TVRA). Technical Specification. European Telecommunications Standards Institute* (2017) (siehe Seite 58).

[66] Tom Fawcett. „An introduction to ROC analysis“. In: *Pattern recognition letters* 27.8 (2006), Seiten 861–874 (siehe Seite 112).

[67] Donald G Firesmith. „A taxonomy of security-related requirements“. In: *International Workshop on High Assurance Systems (RHAS'05)*. Citeseer. 2005, Seiten 29–30 (siehe Seiten 59, 61).

[68] Donald G. Firesmith. *Common concepts underlying safety security and survivability engineering*. 2003 (siehe Seiten 81, 84, 179).

[69] Inc. FIRST.org. *CVSS v3.0 Specification Document*. `https://www.first.org/cvss/v3.0/cvss-v30-specification_v1.9.pdf`. 2019 (siehe Seiten 52–54).

[70] Ivo Friedberg, Kieran McLaughlin, Paul Smith, David Laverty und Sakir Sezer. „STPA-SafeSec: Safety and security analysis for cyber-physical systems“. In: *Journal of Information Security and Applications* 34 (2017), Seiten 183–196. ISSN: 22142126. DOI: `10.1016/j.jisa.2016.05.008` (siehe Seiten 28, 29, 59–61, 217–219, 221).

[71] Marcel Frigault und Lingyu Wang. *Measuring network security using bayesian network-based attack graphs*. IEEE, 2008 (siehe Seite 154).

[72] Hossam A. Gabbar. *Modern formal methods and applications*. Springer, 2006. ISBN: 9781402042225. DOI: `10.1007/1-4020-4223-X` (siehe Seite 146).

[73] Robert Bosch GmbH. *CAN Specification Version 2.0*. `https://info.isl.ntt.co.jp/crypt/eng/psec/dl/iso/psec-kem_v2.1_20080118e.pdf`. 1991 (siehe Seite 41).

[74] Patrice Godefroid. „Using partial orders to improve automatic verification methods“. In: *Computer-aided verification*. Herausgegeben von Edmund M. Clarke. Band 531. Lecture Notes in Computer Science. Springer, 1991, Seiten 176–185. ISBN: 3-540-54477-1. DOI: `10.1007/BFb0023731` (siehe Seite 145).

[75] Deepak Gopalakrishna, Vince Garcia, Ali Ragan, Tony English, Shane Zumpf, Rhonda Young, Mohamed Ahmed, Fred Kitchener, Nayel Urena Serulle, Eva Hsu u. a. *Connected vehicle pilot deployment program phase 1, concept of operations (ConOps), ICF/Wyoming*. 2015 (siehe Seite 44).

[76] graphviz.org. *Graphviz - Graph Visualization Software*. `https://www.graphviz.org/`. 2019 (siehe Seite 293).

[77] graphviz.org. *The DOT Language*. `https://www.graphviz.org/doc/info/lang.html`. 2019 (siehe Seite 293).

[78] David A. Grimes und Kenneth F. Schulz. „Descriptive studies: what they can and cannot do“. In: *The Lancet* 359.9301 (2002), Seiten 145–149. DOI: `10.1016/S0140-6736(02)07373-7` (siehe Seite 110).

[79] Jatin Gupta. *Application of Hazard and Operability (HAZOP) Methodology to Safety-Related Scientific Software*. The Ohio State University, 2014 (siehe Seite 46).

[80] Dzana Hanic und Amer Surkovic. *An Attack Model of Autonomous Systems of Systems*. 2018 (siehe Seiten 59, 60).

[81] Denis Hatebur und Maritta Heisel. „A UML profile for requirements analysis of dependable software“. In: *International Conference on Computer Safety, Reliability, and Security*. 2010, Seiten 317–331 (siehe Seite 115).

[82] Jürgen Hedderich und Lothar Sachs. *Angewandte Statistik: Methodensammlung mit R*. 15., überarb. u. erweiterte Aufl. Springer Spektrum, 2015. ISBN: 3662456915 (siehe Seiten 110, 118, 119, 121, 263).

[83] Olaf Henniger, Ludovic Apvrille, Andreas Fuchs, Yves Roudier, Alastair Ruddle und Benjamin Weyl. „Security requirements for automotive on-board networks“. In: *9th International Conference on ITS Telecommunications (ITST)*. 2009, Seiten 641–646. DOI: 10.1109/ITST.2009.5399279 (siehe Seiten 49, 59, 68, 231–233).

[84] Olaf Henniger, Alastair Ruddle, Hervé Seudié, Benjamin Weyl, Marko Wolf und Thomas Wollinger. „Securing vehicular on-board it systems: The evita project“. In: *VDI/VW Automotive Security Conference*. 2009 (siehe Seiten 29, 30, 59, 67, 68, 164, 232, 233, 236).

[85] Pete Herzog. *Open Source Security Testing Methodology Manual (OSSTMM)*. Herausgegeben von Institute for Security and Open Methodologies. https://www.isecom.org/OSSTMM.3.pdf. 2010 (siehe Seite 125).

[86] Martin Hillenbrand. *Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen*. Band 4. Steinbuch series on advances in information technology. KIT Scientific Publishing, 2012. ISBN: 3866448031 (siehe Seite 44).

[87] Hannes Holm und Khalid Khan Afridi. „An expert-based investigation of the Common Vulnerability Scoring System“. In: *Computers & Security* 53 (2015), Seiten 18–30. ISSN: 01674048. DOI: 10.1016/j.cose.2015.04.012 (siehe Seite 30).

[88] Tobias Hoppe und Jana Dittman. „Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy“. In: *Proceedings of the 2nd workshop on embedded systems security (WESS)*. 2007, Seiten 1–6 (siehe Seite 133).

[89] Giles Howard, Michael Butler, John Colley und Vladimiro Sassone. „Formal analysis of safety and security requirements of critical systems supported by an extended STPA methodology“. In: *2017 IEEE European Symposium on Security and Privacy Workshops*. 2017, Seiten 174–180 (siehe Seiten 59, 60).

[90] Kyle Ingols, Richard Lippmann und Keith Piwowarski. „Practical attack graph generation for network defense“. In: *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*. 2006, Seiten 121–130 (siehe Seite 154).

[91] International Electrotechnical Commission u. a. *Analysis Techniques for System Reliability: Procedure for Failure Mode and Effects Analysis (FMEA)*. International Electrotechnical Commission, 2006 (siehe Seiten 59, 66).

[92] Mafijul Md. Islam, Aljoscha Lautenbach, Christian Sandberg und Tomas Olovsson. „A Risk Assessment Framework for Automotive Embedded Systems“. In: *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*. Herausgegeben von Jianying Zhou. ACM, 2016, Seiten 3–14. ISBN: 9781450342889. DOI: 10.1145/2899015.2899018 (siehe Seiten 59, 70, 238).

[93] ISO. *11898-1: 2015--Road vehicles--Controller area network (CAN)--Part 1: Data link layer and physical signalling* (siehe Seite 41).

[94] ISO. *ISO 17987-1:2016 Road vehicles --- Local Interconnect Network (LIN) --- Part 1: General information and use case definition*. `https://www.iso.org/standard/61222.html`. 2016 (siehe Seite 41).

[95] ISO. *ISO 26021 Road vehicles -- End-of-life activation of on-board pyrotechnic devices*. 2009 (siehe Seiten 130, 134, 137, 278).

[96] ISO. *ISO/IEC 7498-1:1994: Information technology --- Open Systems Interconnection --- Basic Reference Model: The Basic Model*. `https://www.iso.org/standard/20269.html` (siehe Seite 41).

[97] ISO. *ISO/SAE DIS 21434: Road vehicles --- Cybersecurity engineering*. `https://www.iso.org/standard/70918.html` (siehe Seiten 27, 157, 163).

[98] ISO. *Road vehicles - Unified diagnostic services (UDS): Part 1: Specification and requirements*. 2013 (siehe Seiten 130, 134, 275, 276, 279).

[99] ISO. *Road vehicles -- Functional safety -- Part 3: Concept phase*. `http://www.iso.org/iso/catalogue_detail?csnumber=51358` (siehe Seiten 21, 42–44, 47, 54, 55, 72, 82–84, 92, 96, 105, 162, 235, 240).

[100] ISO/IEC. *Common Criteria for Information Technology Security Evaluation: Part 1: Introduction and general model: ISO/IEC 15408-1*. 1. Jan. 2009 (siehe Seiten 27, 30, 52, 67, 69, 75, 234, 238).

[101] Somesh Jha, Oleg Sheyner und Jeannette Wing. „Two formal analyses of attack graphs“. In: *Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE*. 2002, Seiten 49–63 (siehe Seite 49).

[102] Pontus Johnson, Robert Lagerström, Mathias Ekstedt und Ulrik Franke. „Can the common vulnerability scoring system be trusted?“ In: *IEEE Transactions on Dependable and Secure Computing* 15.6 (2016), Seiten 1002–1015. ISSN: 1545-5971 (siehe Seiten 142, 163).

[103] Michael N Johnstone. „Modelling misuse cases as a means of capturing security requirements“. In: (2011) (siehe Seite 62).

[104] Steven A. Julious. „Sample size of 12 per group rule of thumb for a pilot study“. In: *Pharmaceutical Statistics* 4.4 (2005), Seiten 287–291. ISSN: 1539-1604. DOI: 10.1002/pst.185 (siehe Seite 121).

[105] Jessica Jung, Kai Hoefig, Dominik Domis, Andreas Jedlitschka und Martin Hiller. „Experimental Comparison of Two Safety Analysis Methods and Its Replication“. In: *2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. IEEE, 2013, Seiten 223–232. ISBN: 978-0-7695-5056-5. DOI: 10.1109/ESEM.2013.59 (siehe Seite 109).

[106] D. Jungnickel. *Graphs, networks, and algorithms*. 4. ed. Band 5. Algorithms and computation in mathematics. Springer, 2013. ISBN: 9783642322778 (siehe Seite 157).

[107] Dürrwang Jürgen. *Automotive-Security-Threat-Modelling-Tool*. https://github.com/IEEM-HsKA/Automotive-Security-Threat-Modelling-Tool. 2020 (siehe Seite 298).

[108] Vikash Katta, Peter Karpati, Andreas L. Opdahl, Christian Raspotnig und Guttorm Sindre. „Comparing two techniques for intrusion visualization“. In: *IFIP Working Conference on The Practice of Enterprise Modeling*. 2010, Seiten 1–15 (siehe Seiten 59, 62).

[109] Vikash Katta, Christian Raspotnig, Peter Karpati und Tor Stalhane. „Requirements Management in a Combined Process for Safety and Security Assessments“. In: *Proceedings, 2013 Eight International Conference on Availability, Security and Reliability*. IEEE Computer Society, Conference Publishing Services, 2013, Seiten 780–786. ISBN: 978-0-7695-5008-4. DOI: 10.1109/ARES.2013.104 (siehe Seiten 59, 222, 223).

[110] Auguste Kerckhoffs. *La cryptographie militaire, ou, Des chiffres usités en temps de guerre: Avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef*. Librairie militaire de L. Baudoin, 1883 (siehe Seite 135).

[111] Trevor A. Kletz. „Hazop --- past and future“. In: *Reliability Engineering & System Safety* 55.3 (1997), Seiten 263–266. ISSN: 09518320. DOI: 10.1016/S0951-8320(96)00100-7 (siehe Seite 91).

[112] Matthias Kohl. „Performance measures in binary classification“. In: *International Journal of Statistics in Medical Research* 1.1 (2012), Seiten 79–81 (siehe Seite 112).

[113] H. Kopetz. „Automotive electronics“. In: *Proceedings of the 11th Euromicro Conference on Real-Time Systems*. IEEE Computer Soc. Pr, 1999, Seiten 132–140. ISBN: 0-7695-0240-7. DOI: 10.1109/EMRTS.1999.777459 (siehe Seite 21).

[114] Barbara Kordy, Ludovic Piètre-Cambacédès und Patrick Schweitzer. „DAG-based attack and defense modeling: Don't miss the forest for the attack trees". In: *Computer science review* 13 (2014), Seiten 1–38 (siehe Seiten 48, 49).

[115] Eleftherios Koutsofios und Stephen C North. „Drawing graphs with dot". In: (1996) (siehe Seite 293).

[116] Siwar Kriaa, Ludovic Pietre-Cambacedes, Marc Bouissou und Yoran Halgand. „A survey of approaches combining safety and security for industrial control systems". In: *Reliability Engineering & System Safety* 139 (2015), Seiten 156–178. ISSN: 09518320. DOI: 10.1016/j.ress.2015.02.008 (siehe Seiten 59, 61, 62, 67).

[117] R. Kriesten, J. Dürrwang, M. Richter, M. Tucci und M. Shetliffe. „An automotive public key infrastructure design for limited embedded hardware resources". In: *ITS World Congress* 2017 ().

[118] Hans Josef Kullmann. *Produkthaftungsgesetz: Gesetz über die Haftung für fehlerhafte Produkte (ProdHaftG), 4*. Erich Schmidt Verlag GmbH & Co, 2004. ISBN: 978-3503078066 (siehe Seite 21).

[119] HG Lawley. „Operability studies and hazard analysis". In: *Chem. Eng. Prog.* 70.4 (1974), Seiten 45–56 (siehe Seite 91).

[120] C. Y. Lee. „Representation of Switching Circuits by Binary-Decision Programs". In: *Bell System Technical Journal* 38.4 (1959), Seiten 985–999. ISSN: 00058580. DOI: 10.1002/j.1538-7305.1959.tb01585.x (siehe Seite 145).

[121] Clemmer Lee. *Information Security Concepts +1: Confidentiality, Integrity, Availablity, and Authenticity*. https://www.brighthub.com/computing/smb-security/articles/29153.aspx. 2009 (siehe Seite 94).

[122] Nancy Leveson. „A new accident model for engineering safer systems". In: *Safety science* 42.4 (2004), Seiten 237–270 (siehe Seiten 59, 60).

[123] Nancy Leveson. *Engineering a safer world: Systems thinking applied to safety*. Engineering systems. MIT Press, 2012. ISBN: 9781628703399 (siehe Seiten 59, 60).

[124] Elena Lisova, Irfan Sljivo und Aida Causevic. „Safety and Security Co-Analyses: A Systematic Literature Review". In: *IEEE Systems Journal* (2018), Seiten 1–12. ISSN: 1932-8184. DOI: 10.1109/JSYST.2018.2881017 (siehe Seiten 59–61, 63, 65, 67, 72, 107, 121, 177, 179).

[125] Dave Macdonald. *Practical hazops, trips and alarms*. Practical professional books from Elsevier. Elsevier, 2004. ISBN: 0750662743 (siehe Seite 46).

[126] Georg Macher, Eric Armengaud, Eugen Brenner und Christian Kreiner. „A Review of Threat Analysis and Risk Assessment Methods in the Automotive Context“. In: *International Conference on Computer Safety, Reliability, and Security*. 2016, Seiten 130–141 (siehe Seiten 59, 62, 66–68, 71).

[127] Georg Macher, Eric Armengaud, Eugen Brenner und Christian Kreiner. „Threat and Risk Assessment Methodologies in the Automotive Domain“. In: *Procedia Computer Science* 83 (2016), Seiten 1288–1294 (siehe Seiten 59, 71, 72, 241, 242).

[128] Georg Macher, Andrea Höller, Harald Sporer, Eric Armengaud und Christian Kreiner. „A combined safety-hazards and security-threat analysis method for automotive systems“. In: *International Conference on Computer Safety, Reliability, and Security*. 2015, Seiten 237–250 (siehe Seiten 30, 59, 71, 241–243).

[129] Georg Macher, Harald Sporer, Reinhard Berlach, Eric Armengaud und Christian Kreiner. „SAHARA: a security-aware hazard and risk analysis method“. In: *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2015, Seiten 621–624 (siehe Seiten 59, 71, 242).

[130] Martin Salfer und Claudia Eckert. „Attack Graph-Based Assessment of Exploitability Risks in Automotive On-Board Networks“. In: *Proceedings of the 13th International Conference on Availability, Reliability and Security ARES 2018*. ACM, 2018, 21:1–21:10 (siehe Seiten 153, 154).

[131] Martin Salfer und Claudia Eckert. „Attack Graph-Based Assessment of Exploitability Risks in Automotive On-Board Networks“. In: *Proceedings of the 13th International Conference on Availability, Reliability and Security ARES 2018*. ACM, 2018, 21:1–21:10 (siehe Seite 176).

[132] Aditya P. Mathur und Nils Ole Tippenhauer. „SWaT: a water treatment testbed for research and training on ICS security“. In: *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. 2016, Seiten 31–36. DOI: 10.1109/CySWater.2016.7469060 (siehe Seite 64).

[133] Sjouke Mauw und Martijn Oostdijk. „Foundations of attack trees“. In: *Information Security and Cryptology-ICISC 2005*. Springer, 2006, Seiten 186–198 (siehe Seite 49).

[134] Robin McDermott, Raymond J. Mikulak und Michael Beauregard. *The basics of FMEA*. SteinerBooks, 1996 (siehe Seite 66).

[135] Peter Mell und Tim Grance. *Use of the common vulnerabilities and exposures (cve) vulnerability naming scheme*. 2002 (siehe Seite 30).

[136] Peter Mell, Karen Scarfone und Sasha Romanosky. „Common vulnerability scoring system“. In: *IEEE Security & Privacy* 4.6 (2006), Seiten 85–89 (siehe Seite 30).

[137] Microsoft, Herausgeber. *The STRIDE Threat Model*. https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx. 2005 (siehe Seiten 60, 110, 237).

[138] Mohammad Modarres und Se Woo Cheon. „Function-centered modeling of engineering systems using the goal tree -- success tree technique and functional primitives“. In: *Reliability Engineering & System Safety* 64.2 (1999), Seiten 181–200. ISSN: 09518320 (siehe Seiten 59, 63, 64).

[139] Jean-Philippe Monteuuis, Aymen Boudguiga, Jun Zhang, Houda Labiod, Alain Servel und Pascal Urien. „SARA: Security Automotive Risk Analysis Method“. In: *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*. 2018, Seiten 3–14 (siehe Seiten 59, 71).

[140] M. Mulazzani. „Reliability versus safety“. In: *IFAC Proceedings SAFECOMP*. Band 85. 2016, Seiten 141–146 (siehe Seite 21).

[141] Suvda Myagmar, Adam J. Lee und William Yurcik. „Threat modeling as a basis for security requirements“. In: *Symposium on requirements engineering for information security (SREIS)*. Band 2005. 2005, Seiten 1–8 (siehe Seiten 59, 69).

[142] Thor Myklebust, T. Stålhane, G. K. Hanssen und B. Haugset. „Change Impact Analysis as required by safety standards, what to do“. In: *Probabilistic Safety Assessment & Management conference (PSAM12), Honolulu, USA*. 2014 (siehe Seite 42).

[143] Mead Nancy, Shull Forrest, Vemuru Krishnamurthy und Villadsen Ole. *A Hybrid Threat Modeling Method*. Herausgegeben von Software Engineering Institute. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=516617 (siehe Seite 92).

[144] NIST. *National Vulnerability Database (NVD)*. https://nvd.nist.gov/ (siehe Seite 142).

[145] NIST. *Special Publication 800-115: Technical Guide to Information Security Testing and Assessment*. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf (siehe Seite 125).

[146] Tomas Olovsson. „HEAling Vulnerabilities to ENhance Software Security and Safety (HEAVENS)“. In: (2016) (siehe Seiten 30, 59, 69, 70, 237, 238, 240).

[147] OWASP. *Open Web Application Security Project - Testing Guide v4*. https://www.owasp.org/images/1/19/OTGv4.pdf (siehe Seite 125).

[148] OWASP. *OWASP Top Ten Project*. https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. 2019 (siehe Seite 66).

[149] Packet Storm. *Exploits*. https://packetstormsecurity.com. 2019 (siehe Seite 142).

[150] V. Page, M. Dixon und I. Choudhury. „Security risk mitigation for information systems“. In: *BT Technology Journal* 25.1 (2007), Seiten 118–127. DOI: 10.1007/s10550-007-0014-8 (siehe Seite 94).

[151] David Lorge Parnas und Jan Madey. „Functional documents for computer systems“. In: *Science of Computer Programming* 25.1 (1995), Seiten 41–61. ISSN: 01676423 (siehe Seite 38).

[152] Daniel Pereira, Celso Hirata, Rodrigo Pagliares und Simin Nadjm-Tehrani. „Towards Combined Safety and Security Constraints Analysis“. In: *International Conference on Computer Safety, Reliability, and Security*. Band 10489. Springer International Publishing, 2017, Seiten 70–80. ISBN: 978-3-319-66283-1. DOI: 10.1007/978-3-319-66284-87 (siehe Seiten 59, 61).

[153] Sándor Plósz, Christoph Schmittner und Pál Varga. „Combining Safety and Security Analysis for Industrial Collaborative Automation Systems“. In: *International Conference on Computer Safety, Reliability, and Security*. Springer International Publishing, 2017, Seiten 187–198 (siehe Seiten 59, 68, 70).

[154] Bruce Potter. „Microsoft SDL threat modelling tool“. In: *Network Security* 2009.1 (2009), Seiten 15–18. ISSN: 13534858 (siehe Seiten 59, 150, 229, 238, 285).

[155] Sam Procter, Eugene Y. Vasserman und John Hatcliff. „SAFE and secure: Deeply integrating security in a new hazard analysis“. In: *Proceedings of the 12th International Conference on Availability, Reliability and Security*. 2017, Seite 66 (siehe Seiten 59, 61).

[156] Rapid7. *Vulnerability and Exploit Database*. Herausgegeben von Rapid7 (siehe Seite 142).

[157] Raspotnig, Christian, Karpati, Peter, Katta und Vikash. *CHASSIS guideline*. http://hdl.handle.net/1956/6172. 2012 (siehe Seiten 28, 59, 61, 222, 223).

[158] Christian Raspotnig, Peter Karpati und Vikash Katta. „A combined process for elicitation and analysis of safety and security requirements“. In: *Enterprise, business-process and information systems modeling*. Springer, 2012, Seiten 347–361 (siehe Seiten 59, 61, 62, 222).

[159] Christian Raspotnig, Vikash Katta, Peter Karpati und Andreas L. Opdahl. „Enhancing CHASSIS: A Method for Combining Safety and Security“. In: *Proceedings, 2013 Eight International Conference on Availability, Security and Reliability*. IEEE Computer Society, Conference Publishing Services, 2013, Seiten 766–773. ISBN: 978-0-7695-5008-4. DOI: 10.1109/ARES.2013.102 (siehe Seiten 59, 62).

[160] Christian Raspotnig und Andreas Opdahl. „Supporting Failure Mode and Effect Analysis: A Case Study with Failure Sequence Diagrams“. In: *Requirements engineering: foundation for software quality*. Herausgegeben von Björn Regnell und Daniela Damian. Band 7195. Lecture Notes in Computer Science. Springer, 2012, Seiten 117–131. ISBN: 978-3-642-28713-8. DOI: 10.1007/978-3-642-28714-510 (siehe Seiten 59, 62).

[161] Balwant Rathore, Mark Brunner, Miguel Dilaj, Omar Herreragh, Piero Brunati, Rama Subramaniam, Subash Raman und Umesh Chavan. *Information Systems Security Assessment Framework (ISSAF): Draft 0.2*. Herausgegeben von OSSIG. www.oissg.org/files/issaf0.2.1.pdf (siehe Seite 125).

[162] Konrad Reif. *Automobilelektronik: Eine Einführung für Ingenieure*. 5. Auflage. Springer Vieweg, 2014. ISBN: 3658050489 (siehe Seiten 40, 41, 275).

[163] Falko Rheinberg, Yvette Manig, Reinhold Kliegl, Stefan Engeser und Regina Vollmeyer. „Flow bei der Arbeit, doch Glück in der Freizeit: Zielausrichtung, Flow und Glücksgefühle“. In: *Zeitschrift für Arbeits-und Organisationspsychologie* (2007) (siehe Seite 259).

[164] Falko Rheinberg, Regina Vollmeyer und Stefan Engeser. *Die Erfassung des Flow-Erlebens*. na, 2003 (siehe Seiten 114, 122, 259, 261, 262).

[165] Mark Richards. *Software architecture patterns: Understanding common architecture patterns and when to use them*. First edition. O'Reilly Media, 2015 (siehe Seite 296).

[166] R. W. Ritchey und P. Ammann. „Using model checking to analyze network vulnerabilities“. In: *Proceedings 2000 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2000, Seiten 156–165. ISBN: 0-7695-0665-8. DOI: 10.1109/SECPRI.2000.848453 (siehe Seite 153).

[167] Alastair Ruddle, David Ward, Benjamin Weyl, Sabir Idrees, Yves Roudier, Michael Friedewald, Timo Leimbach, Andreas Fuchs, S. Grgens, Olaf Henniger u. a. *Deliverable d2. 3: Security requirements for automotive on-board networks based on dark-side scenarios*. 2009 (siehe Seiten 59, 163, 232–236).

[168] Bernhard Rumpe. *Modeling with UML*. Springer International Publishing, 2016. ISBN: 978-3-319-33932-0. DOI: 10.1007/978-3-319-33933-7 (siehe Seite 84).

[169] Giedre Sabaliauskaite und Sridhar Adepu. „Integrating Six-Step Model with Information Flow Diagrams for Comprehensive Analysis of Cyber-Physical System Safety and Security“. In: *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, 2017, Seiten 41–48. ISBN: 978-1-5090-4636-2. DOI: 10.1109/HASE.2017.25 (siehe Seiten 59, 64, 65).

[170] Giedre Sabaliauskaite, Sridhar Adepu und Aditya Mathur. „A Six-Step Model for Safety and Security Analysis of Cyber-Physical Systems“. In: *Critical information infrastructures security*. Herausgegeben von Grigore Havarneanu, Roberto Setola, Hypatia Nassopoulos und Stephen Wolthusen. Band 10242. Lecture Notes in Computer Science. Springer, 2017, Seiten 189–200. ISBN: 978-3-319-71367-0. DOI: 10.1007/978-3-319-71368-716 (siehe Seiten 29, 59, 63, 64).

[171] Giedre Sabaliauskaite, Lin Shen Liew und Jin Cui. „Integrating autonomous vehicle safety and security analysis using stpa method and the six-step model“. In: *International Journal on Advances in Security* 11.1&2 (2018), Seiten 160–169. ISSN: 1942-2636 (siehe Seiten 59, 64, 65, 225, 226).

[172] Giedre Sabaliauskaite, Lin Shen Liew, Fengjun Zhou und Jin Cui. „Designing Safe and Secure Mixed Traffic Systems“. In: *2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, 2019, Seiten 222–227. ISBN: 978-1-5386-8540-2. DOI: 10.1109/HASE.2019.00041 (siehe Seiten 59, 65).

[173] SAE. *Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*. http://standards.sae.org/wip/j3061/. 2016 (siehe Seiten 54, 55, 58, 59).

[174] Chris Salter, O. Sami Saydjari, Bruce Schneier und Jim Wallner. „Toward a secure system engineering methodolgy“. In: *Proceedings of the 1998 workshop on New security paradigms*. 1998, Seiten 2–10 (siehe Seite 49).

[175] Frank Saunders, Jason Rife, Sai Vaddi und Victor Cheng. „Information flow diagram analysis of a model cyber-physical system: Conflict detection and resolution for airport surface traffic“. In: *IEEE Aerospace and Electronic Systems Magazine* 28.12 (2013), Seiten 26–35. ISSN: 0885-8985. DOI: 10.1109/MAES.2013.6693666 (siehe Seiten 59, 65).

[176] Riccardo Scandariato, Kim Wuyts und Wouter Joosen. „A descriptive study of Microsoft's threat modeling technique“. In: *Requirements Engineering* 20.2 (2015), Seiten 163–180 (siehe Seiten 31, 109, 112, 113, 121).

[177] Jörg Schäuffele und Thomas Zurawka. *Automotive Software Engineering: Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. 5., überarb. u. ak. Aufl. 2013. SpringerLink : Bücher. Springer Vieweg, 2013. ISBN: 3834824704 (siehe Seite 76).

[178] Ulrich Schiefele und Emmanouil Roussakis. „Die Bedingungen des Flow-Erlebens in einer experimentellen Spielsituation“. In: *Zeitschrift für Psychologie/Journal of Psychology* 214.4 (2006), Seiten 207–219 (siehe Seite 259).

[179] Christoph Schmittner, Thomas Gruber, Peter Puschner und Erwin Schoitsch. „Security application of failure mode and effect analysis (FMEA)“. In: *Computer Safety, Reliability, and Security*. Springer, 2014, Seiten 310–325 (siehe Seiten 59, 66, 179, 227, 229, 231).

[180] Christoph Schmittner, Zhendong Ma, Erwin Schoitsch und Thomas Gruber. „A Case Study of FMVEA and CHASSIS as Safety and Security Co-Analysis Method for Automotive Cyber-physical Systems“. In: *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*. 2015, Seiten 69–80 (siehe Seiten 59, 62, 63, 223, 230).

[181] Bruce Schneier. „Attack trees“. In: *Dr. Dobb's journal* 24.12 (1999), Seiten 21–29 (siehe Seite 49).

[182] E. Seier. *Comparison of tests for univariate normality. INTERSTAT 17*. 2002 (siehe Seite 118).

[183] Howard J. Seltman. „Experimental design and analysis“. In: *Department of Statistics at Carnegie Mellon* (2009) (siehe Seiten 109, 110, 120).

[184] *SESAMO: Security and Safety Modeling for Embedded Systems*. `http://www.sesamo-project.eu/`. 2019 (siehe Seiten 59, 71).

[185] S. S. Shapiro und M. B. Wilk. „An Analysis of Variance Test for Normality (Complete Samples)“. In: *Biometrika* 52.3/4 (1965), Seite 591. ISSN: 00063444. DOI: `10.2307/2333709` (siehe Seite 118).

[186] Adam Shostack. „Experiences threat modeling at microsoft“. In: *Modeling Security Workshop. Dept. of Computing, Lancaster University, UK*. 2008 (siehe Seite 66).

[187] Forrest Shull, Janice Singer und Dag I. K. Sjøberg. *Guide to Advanced Empirical Software Engineering*. Springer London, 2008. ISBN: 978-1-84800-043-8. DOI: `10.1007/978-1-84800-044-5` (siehe Seiten 109, 121).

[188] Guttorm Sindre und Andreas L. Opdahl. „Capturing security requirements through misuse cases“. In: *NIK 2001, Norsk Informatikkonferanse 2001* (2001) (siehe Seite 62).

[189] Guttorm Sindre und Andreas L. Opdahl. „Eliciting security requirements with misuse cases“. In: *Requirements Engineering* 10.1 (2005), Seiten 34–44. ISSN: 0947-3602. DOI: `10.1007/s00766-004-0194-4` (siehe Seite 52).

[190] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang und Prateek Mittal. „Darts: Deceiving autonomous cars with toxic signs“. In: *arXiv preprint arXiv:1802.06430* (2018) (siehe Seite 39).

[191] Vladimir Sklyar. „Application of Reliability Theory to Functional Safety of Computer Control Systems“. In: *Reliability: Theory & Applications* 12.1 (44) (2017) (siehe Seite 21).

[192] Thitima Srivatanakul, John A. Clark und Fiona Polack. „Effective security requirements analysis: Hazop and use cases“. In: *Information Security*. Springer, 2004, Seiten 416–427 (siehe Seiten 59, 61).

[193] Tor Stålhane und Guttorm Sindre. „A Comparison of Two Approaches to Safety Analysis Based on Use Cases“. In: *Conceptual modeling - ER 2007*. Herausgegeben von Christine Parent, Klaus-Dieter Schewe, Veda C. Storey und Bernhard Thalheim. Band 4801. Lecture Notes in Computer Science. Springer, 2007, Seiten 423–437. ISBN: 978-3-540-75562-3. DOI: 10.1007/978-3-540-75563-029 (siehe Seite 109).

[194] Tor Stålhane und Guttorm Sindre. „Safety Hazard Identification by Misuse Cases: Experimental Comparison of Text and Diagrams“. In: *Model driven engineering languages and systems*. Herausgegeben von Krzysztof Czarnecki, Ileana Ober, Jean-Michel Bruel, Axel Uhl und Markus Völter. Band 5301. Lecture Notes in Computer Science. Springer, 2008, Seiten 721–735. ISBN: 978-3-540-87874-2. DOI: 10.1007/978-3-540-87875-950 (siehe Seite 109).

[195] Tor Stålhane, Guttorm Sindre und Lydie Du Bousquet. „Comparing Safety Analysis Based on Sequence Diagrams and Textual Use Cases“. In: *Active Flow and Combustion Control 2018*. Herausgegeben von Rudibert King. Band 141. Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer International Publishing, 2019, Seiten 165–179. ISBN: 978-3-319-98176-5. DOI: 10.1007/978-3-642-13094-614 (siehe Seite 109).

[196] Stefanie Strobl, David Hofbauer, Christoph Schmittner, Silia Maksuti, Markus Tauber und Jerker Delsing. „Connected cars --- Threats, vulnerabilities and their impact“. In: *Proceedings 2018 IEEE Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2018, Seiten 375–380. ISBN: 978-1-5386-6531-2. DOI: 10.1109/ICPHYS.2018.8387687 (siehe Seiten 59, 66).

[197] F. Swiderski und W. Snyder. *Threat Modeling*. Microsoft Press, 2004. ISBN: 9780735619913 (siehe Seite 69).

[198] William G. Temple, Yue Wu, Binbin Chen und Zbigniew Kalbarczyk. „Reconciling Systems-Theoretic and Component-Centric Methods for Safety and Security Co-analysis“. In: *Computer Safety, Reliability, and Security*. Herausgegeben von Stefano Tonetta, Erwin Schoitsch und Friedemann Bitsch. Band 10489. Lecture Notes in Computer Science. Springer International Publishing, 2017, Seiten 87–93. ISBN: 978-3-319-66283-1. DOI: 10.1007/978-3-319-66284-89 (siehe Seiten 59, 60, 66).

[199] Chee-Wooi Ten, Chen-Ching Liu und Govindarasu Manimaran. „Vulnerability Assessment of Cybersecurity for SCADA Systems“. In: *IEEE Transactions on Power Systems* 23.4 (2008), Seiten 1836–1846. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2008.2002298 (siehe Seite 49).

[200] *The Penetration Testing Execution Standard (PTES)*. http://www.pentest-standard.org/index.php/Main_Page (siehe Seiten 31, 125, 126).

[201] Yann Thierry-Mieg. *Guarded Action Language*. https://lip6.github.io/ITSTools-web/files/gal.pdf (siehe Seiten 150, 283, 284, 293).

[202] Yann Thierry-Mieg. „Symbolic Model-Checking Using ITS-Tools“. In: *Tools and algorithms for the construction and analysis of systems*. Herausgegeben von Christel Baier und Cesare Tinelli. Band 9035. Lecture Notes in Computer Science. Springer, 2015, Seiten 231–237. ISBN: 978-3-662-46680-3. DOI: 10.1007/978-3-662-46681-020 (siehe Seiten 145, 152, 156, 282, 296, 297).

[203] Jeffrey M. Thompson, Michael W. Whalen und Mats Per Erik Heimdahl. „Requirements capture and evaluation in Nimbus: The light-control case study“. In: *Journal of Universal Computer Science* 6.7 (2000), Seiten 731–757 (siehe Seite 38).

[204] Walter F. Tichy. „Hints for reviewing empirical work in software engineering“. In: *Empirical Software Engineering* 5.4 (2000), Seiten 309–312 (siehe Seite 110).

[205] Upstream Security Ltd. *Upstream Security Global Automotive Cybersecurity Report 2019*. https://www.upstream.auto/upstream-security-global-automotive-cybersecurity-report-2019/. 2019 (siehe Seite 163).

[206] Siddhartha Verma, Thomas Gruber, Peter Puschner und Christoph Schmittner. *Combined Approach for Safety and Security*. 2019. DOI: 10.13140/RG.2.2.15021.13283 (siehe Seiten 59, 67).

[207] Lingyu Wang, Massimiliano Albanese und Sushil Jajodia. „Attack Graph and Network Hardening“. In: *Network Hardening*. Herausgegeben von Lingyu Wang, Massimiliano Albanese und Sushil Jajodia. Band 35. SpringerBriefs in computer science. Springer International Publishing, 2014, Seiten 15–22. ISBN: 978-3-319-04611-2. DOI: 10.1007/978-3-319-04612-93 (siehe Seiten 48, 49).

[208] Stijn Winsen. „Threat modelling for future vehicles: on identifying and analysing threats for future autonomous and connected vehicles“. Dissertation. University of Twente (siehe Seite 221).

[209] Rune Winther, Ole-Arnt Johnsen und Bjørn Axel Gran. „Security assessments of safety critical systems using HAZOPs“. In: *Computer Safety, Reliability and Security*. Springer, 2001, Seiten 14–24 (siehe Seiten 62, 91, 94).

[210] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell und Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012 (siehe Seiten 110, 121).

[211] Marko Wolf, André Weimerskirch und Thomas Wollinger. „State of the Art: Embedding Security in Vehicles“. In: *EURASIP Journal on Embedded Systems* 2007.4 (2007), Seiten 1–16. ISSN: 1687-3955. DOI: 10.1155/2007/74706 (siehe Seite 171).

[212] Michael Wolf. *Combining safety and security threat modeling to improve automotive penetration testing*. https://oparu.uni-ulm.de/xmlui/bitstream/123456789/13119/4/Thesis_MW_Oparu.pdf. 2019. DOI: 10.18725/OPARU-13062 (siehe Seiten 59, 67).

[213] Zhu Xin-feng, Wang Jian-dong, Li Bin, Zhu Jun-wu und Wu Jun. „Methods to tackle state explosion problem in model checking“. In: *2009 Third International Symposium on Intelligent Information Technology Application*. Band 2. IEEE. 2009, Seiten 329–331 (siehe Seite 144).

[214] Wenjun Xiong, Melek Gülsever, Koray Mustafa Kaya und Robert Lagerström. „A Study of Security Vulnerabilities and Software Weaknesses in Vehicles“. In: *Nordic Conference on Secure IT Systems*. 2019, Seiten 204–218. DOI: 10.1007/978-3-030-35055-013 (siehe Seiten 142, 163).

[215] Chen Yan, Wenyuan Xu und Jianhao Liu. „Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle“. In: *Def Con* 24.8 (2016), Seite 109 (siehe Seite 39).

[216] Kaya Yilmaz. „Comparison of Quantitative and Qualitative Research Traditions: epistemological, theoretical, and methodological differences“. In: *European Journal of Education* 48.2 (2013), Seiten 311–325. ISSN: 01418211. DOI: 10.1111/ejed.12014 (siehe Seite 21).

[217] William Young und Nancy Leveson. „Systems thinking for safety and security“. In: *Proceedings of the 29th Annual Computer Security Applications Conference*. Herausgegeben von Charles N. Payne. ACM Digital Library. ACM, 2013, Seiten 1–8. ISBN: 9781450320153. DOI: 10.1145/2523649.2530277 (siehe Seiten 59, 60).

[218] Markus Zoppelt und Ramin Tavakoli Kolagari. „SAM: A Security Abstraction Model for Automotive Software Systems“. In: *Security and safety interplay of intelligent software systems*. Herausgegeben von Brahim Hamid, Barbara Gallina, Asaf Shabtai, Yuval Elovici und Joaquin Garcia-Alfaro. Band 11552. LNCS sublibrary. SL 4, Security and cryptology. Springer, 2019, Seiten 59–74. ISBN: 978-3-030-16873-5. DOI: 10.1007/978-3-030-16874-25 (siehe Seite 82).

Indexe

Abbildungsverzeichnis

Tabellenverzeichnis

Auflistung

Anhänge

A

Bedrohungsanalysen

In diesem Abschnitt sind detaillierte Ergänzungen zu den in Abschnitt 3.2 und Abschnitt 3.3 beschrieben Methodiken gegeben. Ziel des Kapitels ist es ein tieferes Verständnis für – die verwandten Arbeiten – zu vermitteln, um die Argumente in Kapitel 3 besser nachvollziehen zu können.

A.1 Ergänzende Erläuterungen zu STAP-SafeSec

STAP-SafeSec folgt einem Top-Down Vorgehen das aus 7 Schritten besteht, über zwei Schleifen gekoppelt und in Abbildung A.1 dargestellt ist. Die äußere Schleife zeigt, dass STPA-SafeSec als ein iterativer Ansatz konzipiert ist, der während der Lebensdauer eines Systems kontinuierlich angewendet werden sollte.

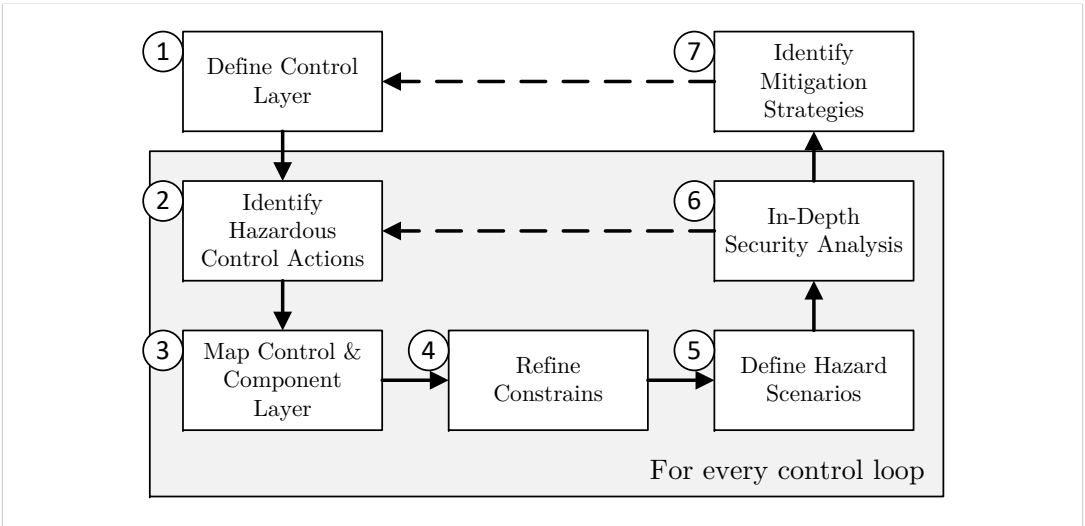


Abbildung A.1: Übergeordnete Analyse-Phasen von STPA-SafeSec. Die sieben Phasen zeigen das Abgrenzen des Analyse-Umfanges, das Identifizieren der Abweichungen des Systemverhalten und die Auswahl von Abschwächungsmaßnahmen. Die gestrichelten Pfeile entsprechen den Iterationen die STPA-SafeSec in der Analyse vorsieht [70].

In der ersten Phase wird die Regelschicht für das Gesamtsystem definiert, die aus einem oder mehreren Regelkreisen besteht. Diese werden analysiert, um Gefahrenszenarien aufzudecken, die durch Systemfehler oder gefährliche Handlungen (engl. hazardous control actions) verursacht werden können. Letztere werden in der zweiten Phase aufgedeckt. Außerdem wird jeder Regelkreis separat analysiert, wodurch die Komplexität beherrschbar sein soll (siehe graue Umrahmung in Abbildung A.1) [70]. In der dritten Phase werden die auf der Controller-Ebene (engl. control layer) identifizierten Systemeinschränkungen auf die physische Komponentenschicht (engl. component layer) übertragen. Dies dient zur Identifikation konkreter Systemfehler, welche die Gefährdungssituationen auslösen und zu Abweichungen gegenüber des spezifizierten Systemverhaltens führen können. An dieser Stelle betrachtet STPA-SafeSec – neben den Ursachen aus dem Safety-Umfeld – ebenfalls Ursachen, die durch Verletzungen von Security-Anforderungen entstehen können. Diese werden in Phase 4 anschließend weiter verfeinert. Mittels der Anwendung von Kausalfaktoren werden in Phase 5 Szenarien abgeleitet, welche zu gefährlichen Kontrollaktionen führen können und bei STPA-SafeSec um Security-Aspekte erweitert werden. Hierbei können die Kausalfaktoren als generische Fehler oder Bedrohungen gesehen werden, welche den Analysten bei der Analyse anleiten. Anschließend wird die abstrakte Regelebene auf eine implementierungsspezifische Komponentenebene übertragen, die mehr Details aufzeigt. Dies ermöglicht die Einschränkungen und Kausalfaktoren weiter zu detaillieren, welche für eine tiefer gehende Security-Analyse in Phase 6 verwendet werden können. Als Ergebnis aus diesem Schritt werden in Phase 7 Abschwächungsmaßnahmen vorgeschlagen, die einen sicheren Betrieb gewährleisten sollen [70].

Die Kernbeiträge von STPA-SafeSec gegenüber STPA sind die Einführung eines generischen Komponentenebenendiagramms in Phase 5, welches die abstrakte Kontrollschicht um eine detailliertere Komponentenebene erweitert [70]. Letztere führte das generische Kontrollstruktur-Diagramm ein, um bei der Identifizierung der Regelkreise zu unterstützen. Die Komponentenebene in STPA-SafeSec erweitert die Visualisierung der Systemimplementierung um Knoten mit Regelalgorithmen oder Sensoren. Außerdem gehören die Netzwerknoten mit deren physikalischen Verbindungen und die eingesetzten Protokolle zur Komponentenebene dazu. Der zweite Beitrag entspricht der Erweiterung der Kausalfaktoren hinsichtlich Security, die mit dem Kausalfaktor-Diagramm von STPA verwendet werden können, um gefährliche Kontrollaktionen zu identifizieren, welche durch Verletzen der Security-Eigenschaften entstehen können. Hierzu ist eine Menge von Faktoren abgeleitet, welche die Auswirkungen von Cyber-Angriffen auf die Integrität und Verfügbarkeit aufzeigen. Die Kausalfaktoren stehen dem Analysten als Liste zur Verfügung und unterstützen ihn bei der Analyse. Außerdem ermöglicht dies ein Ableiten von Angreifer-Fähigkeiten auf Komponentenebene, die notwendig sind, um einen böswilligen Effekt auslösen zu können [70].

Die detaillierten Schritte zu STPA-SafeSec sind in Abbildung A.2 aufgezeigt.

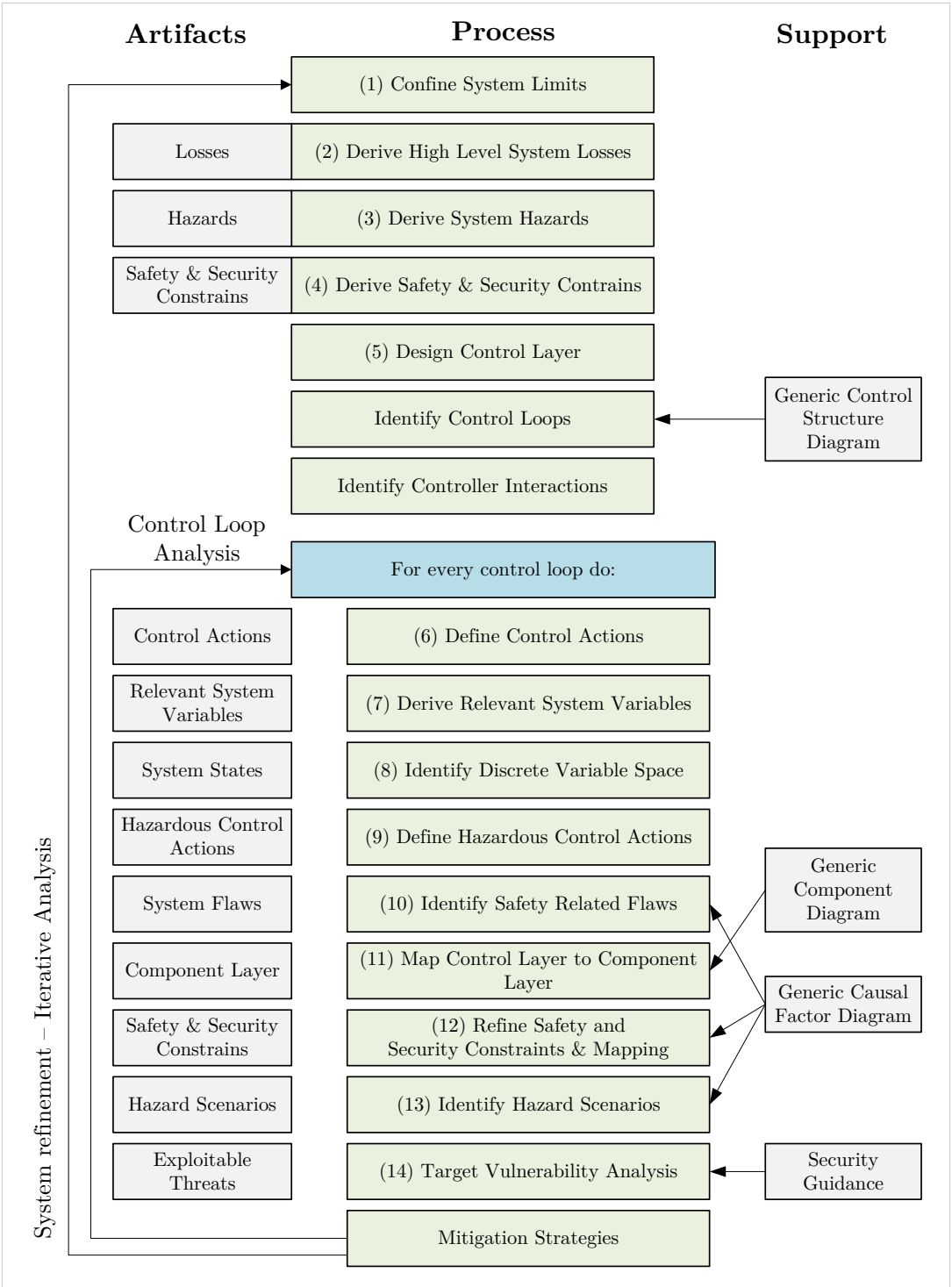


Abbildung A.2: Detailliertere Analyse-Schritte von Abbildung A.1 für die integrierte Safety- und Security-Analyse [70]. Die Abbildung zeigt mit den linken Rechtecken die erzeugten Artefakte (engl. Artifacts), mit den mittleren Rechtecken den jeweiligen Analyse-Schritt (engl. Process) und mit den rechten Rechtecken die unterstützenden Artefakte (engl. Support).

Begonnen wird mit dem Festlegen des Analyseumfanges in Schritt 1. Nachfolgend werden in den Schritten 2 bis 3 die Safety- und Security-Anforderungen festgelegt.

Hier werden Artefakte ermittelt, welche auf Schwächen des Systemdesigns hindeuten und es ermöglichen, die Systemverluste zurückzuverfolgen. Anschließend bestimmt STPA-SafeSec in Schritt 4 systemweite Anforderungen, die sich auf Safety und Security beziehen.

In Schritt 5 von Abbildung A.2 wird die Kontrollebene erzeugt und grafisch die Regelkreise und Wechselwirkungen zwischen den Controllern erfasst. Außerdem werden Anforderungen hinsichtlich Safety und Security auf die unterschiedlichen Aspekte der Kontrollstruktur übertragen. Letztere stellt die Basis der nachfolgenden Schritte dar. Die Kontrollschleifenanalyse (Schritte 6 bis 14 in Abbildung A.2) fokussiert sich auf die Identifikation von Kausalfaktoren, welche die Safety- oder Security-Anforderungen verletzen. In Schritt 6 startet die sorgfältige Analyse der Kontrollschleifen, um die Gefährdungsszenarios zu identifizieren und Abschwächungsstrategien bestimmen zu können. Letztere dienen zur Einhaltung der initial festgelegten Safety- und Security-Anforderungen. Hierzu werden in den Schritten 6 bis 8 die Kontrollaktionen definiert, die relevanten Systemvariablen abgeleitet sowie der diskrete Variablen-Raum der Kontrollschleife bestimmt. Diese Informationen ermöglichen es anschließend die gefährlichen Kontrollaktionen in Schritt 9 bestimmen zu können.

Basierend auf der Menge der gefährlichen Kontrollaktionen, dem generischen Kausalfaktor-Diagramm und der Menge der Bedrohungen hinsichtlich der Integrität und der Verfügbarkeit, werden in Schritt 10 Safety-relevante Fehler identifiziert. Schritt 11 ordnet anschließend die Kontrollebene jeder Kontrollschleife der entsprechenden Komponentenebene zu. Hier wird visualisiert auf welchen physikalischen Komponenten, welche Algorithmen vorliegen. Außerdem wird eine Zuordnung zwischen den abstrakten Kommunikationspfaden und dem konkret genutzten Netzwerk erzeugt. Bezogen auf die erzeugte Komponentenebene werden im Schritt 12 die Safety- und Security-Anforderungen verfeinert und der Komponentenebene zugeordnet. Dies geschieht, indem der Analyst die Safety-Anforderungen in der Kontrollebene mit Security-Anforderungen aus der Komponentenebene ergänzt. Ist dies abgeschlossen, werden in Schritt 13 die Gefahrensituationen (engl. hazardous scenarios) aufgedeckt. Diese beschreiben wie genau Safety-Anforderungen verletzt werden können.

In Schritt 14 wird anschließend die Komponentenebene verwendet, um eine detaillierte Security-Schwachstellenanalyse durchzuführen. Anhand der kritischsten Security-Verletzungen auf der Kontrollebene und deren Übertragung auf die Komponentenebene kann die Auswahl der Komponenten für eine tiefer gehende Schwachstellenanalyse priorisiert werden. Letztere deckt die potenziell ausnutzbaren Bedrohungen auf, welche als Ergebnisse der Bedrohungsanalyse gesehen werden können. Zusammen mit festgelegten Situationen können außerdem die durch Cyber-Angriffe verursachten Systemverluste nachverfolgt werden. Das ermöglicht nach Ansicht der Autoren das Identifizieren der effektivsten Abschwächungen. Als effektiv wird hierbei eine Menge von Abschwächungsmaßnahmen angesehen, die alle Pfade in einem Situationenbaum abdeckt. An

dieser Stelle soll daraufhin gewiesen werden, das STPA-SafeSec alle Analyseergebnisse in Baumstrukturen dargestellt. Die Autoren argumentieren diese Entscheidung mit der Aussage, dass Baumstrukturen die dominierende Notation in Safety (Fehlerbäume) und in Security (Angriffsbäume) sind [70]. Neben dieser Darstellung der Ergebnisse können nach Ansicht der Autoren bestimmte Artefakte von STPA-SafeSec zur Kommunikation mit anderen Parteien wie dem Management verwendet werden [70]:

1. Gefährdungsszenarien können zur Kommunikation für die Notwendigkeit von Maßnahmen mit der Unternehmensleitung dienen.
2. Abschwächungsstrategien können an jedem Baumknoten angewendet werden. So sind alle Szenarien in einem bestimmten Baum abgedeckt, wenn jeder Pfad vom Blatt bis zur Wurzel in mindestens einem Knoten abgeschwächt ist.
3. Die Unterknoten können verfeinert werden, um aufzuzeigen, auf welche Weise eine Anforderung verletzt wurde. Auf diese Weise können Angriffsbäume verwendet werden, um Hazard-Szenarien weiter in der Cyber-Security-Domäne zu verfeinern.
4. Die Baumstruktur erhöht die Sichtbarkeit für notwendige und tiefer gehende Security-Analysen.

Hinsichtlich der Anwendbarkeit von STPA-SafeSec kann auf eine Beispielanwendung aus der Domäne der Energieversorgung (engl. Smart Power Grid) verwiesen werden [70]. Eine Anwendung im Bereich von Fahrzeugen-Systemen ist zum Zeitpunkt dieser Arbeit nicht gezeigt.

A.1.1 Ergänzende Erläuterungen zu CHASSIS

In der ersten Phase von CHASSIS und in Abbildung A.3 (Schritt 1 bis 3) dargestellt, werden die Systemfunktionen und Services festgelegt, was einer Festlegung funktionaler Anforderungen entspricht [208]. Diese können anhand von Funktions- und Umgebungsbeschreibungen abgeleitet werden. Benutzer, Funktionen und Dienste werden in Use-Case-Diagrams (UCDs) (Schritt 1) dokumentiert und mit textuellen Beschreibungen der Anwendungsfälle mit Textuell-Use-Cases (T-UCs) in Schritt 2 festgehalten.

Die SD werden anschließend in Schritt 3 verwendet, um den Inhalt der UCDs zu verfeinern und Objekte und deren Wechselwirkungen anhand von Sequenzen zu modellieren. Sollte für ein System bereits eine Architektur und Funktionen definiert sein, so müssen bei dieser Aktivität die entstandenen Artefakte möglicherweise überarbeitet werden, um in CHASSIS weiterverwendet werden zu können. Die zweite Phase (Schritt 4-6) fokussiert sich auf die Erhebung und Analyse der Safety- und Security-Anforderungen. Hierzu werden in Schritt 4 von Abbildung A.3 MUCD erzeugt, die auf den T-UC aus Schritt 2 basieren und durch böswillige Handlungen eines Angreifers sowie dem erwarteten Schaden erweitert werden. Zur Identifizierung der MUC werden Satzanleitungen

mit HAZOP Leitwörtern als Vorlagen verwendet und in einer Brainstorming-Sitzung mit Domain- und Sicherheitsexperten eingesetzt.

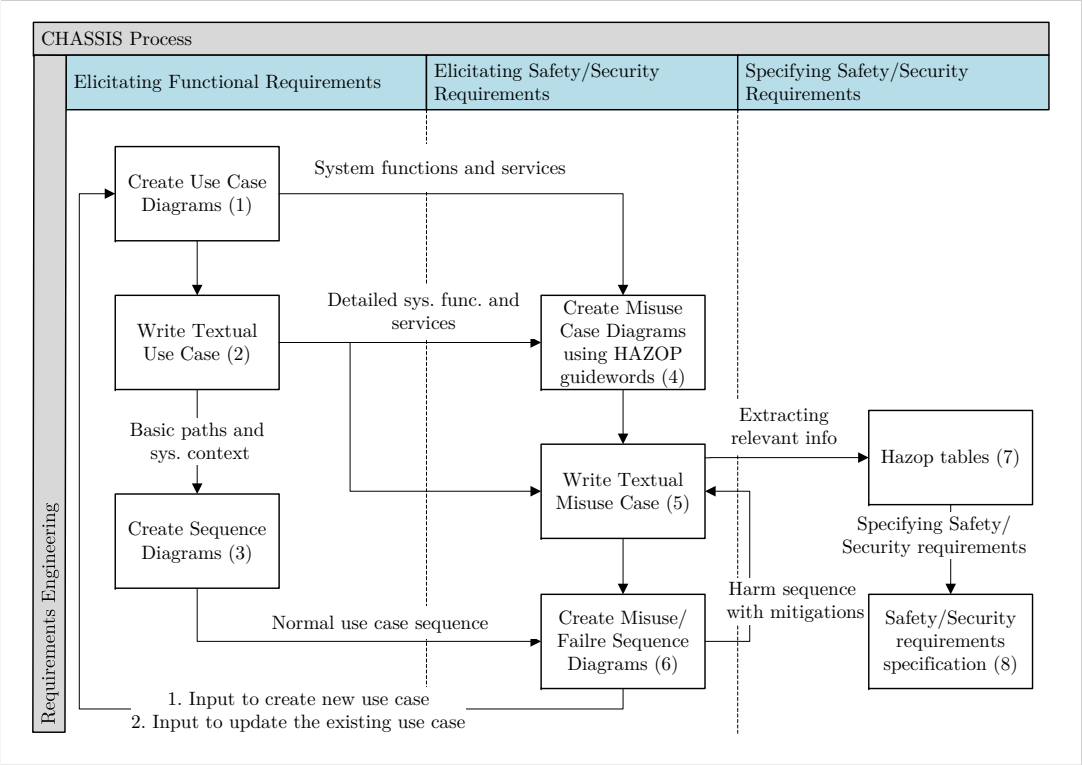


Abbildung A.3: Vereinheitlichter Prozess des CHASSIS-Ansatzes mit acht Methoden Schritten, aufgeteilt in drei Phasen [158].

Nach der Identifizierung potenzieller Missbrauchsfälle – bezogen auf Safety und Security – werden die potenziellen Verursacher identifiziert [157, Seiten 5, 12]. Die Entität *Missbraucher* (engl. misuser) umfasst nach CHASSIS alle menschlichen Benutzer sowie externe und interne Systeme, die ausfallen oder das betrachtete System gefährden können [109]. Neben der grafischen Darstellung werden in Schritt 5 Sicherheitsmissbrauchsfälle zusätzlich als Textuell-Misuse-Case (T-MUC) festgehalten. Diese Beschreibungen zeigen weitere Details zu den Akteuren und Missbrauchern. In Schritt 6 verwenden Stakeholder anschließend FSD und MUSD, um die Schadenszenarien zu verfeinern. Letztere stehen für die potenziellen Security-Probleme und die erzeugten FSD beschreiben die Safety-bezogenen Ursachen. Der Zweck dieser Diagramme ist die Beschreibung von Ereignisketten und Interaktion mit einem System, die zu einem Missbrauch führen können. Ein beispielhaftes Sequenz-Diagramm ist in Abbildung A.4 gezeigt.

Die entstandenen Sequenz-Diagramme werden erörtert, um die Schwächen in Systemen und Services aufzudecken. An dieser Stelle werden erste Ideen und Ansätze für Gegenmaßnahmen in den Diagrammen festgehalten [109]. Zur Darstellung von Konflikten zwischen definierten Maßnahmen übernimmt CHASSIS eine Notation, die von den Forschern Alexander et al. [19] vorgeschlagen wurde. Relevante Informationen aus den

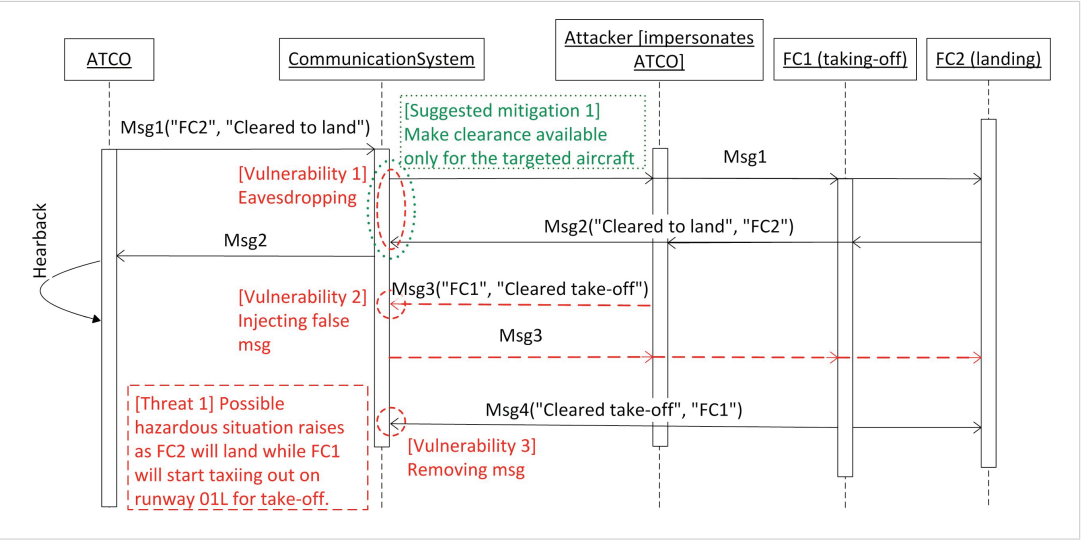


Abbildung A.4: Beispiel eines Misuse-Case-Sequenz-Diagram (MUSD), das bei CHASSIS verwendet wird, um die Handlungen eines Angreifers darzustellen. Hierzu ist der Angreifer als Objekt (Attacker) in das MUSD eingefügt [157]. Weiterhin zeigen die rot-gestrichelte Kreise Schwachstellen, die der Angreifer ausnutzen kann sowie grün-gepunktete Rechtecke, die Vorschläge für Abschwächungen darstellen.

FSD und MUSD, die in Schritt 6 erzeugt wurden, werden zu Schritt 5 zurückgeleitet und identifizierte Gegenmaßnahmen werden als funktionale Anwendungsfälle festgelegt. Diese neuen Anwendungsfälle werden zusätzlich nach Schritt 1 zurückgeleitet, was in einer Iteration der ersten Stufe resultiert. Sollten die T-MUC finalisiert sein, werden in Schritt 7 HAZOP-Tabellen vorbereitet, mit deren Hilfe in der dritten Stufe von Abbildung A.3 die entsprechenden Safety- und Security-Anforderungen festgelegt werden können. Der Prozess ist abgeschlossen, sobald alle Use-Cases und Misuse-Cases bezogen auf die Schäden und Gegenmaßnahmen untersucht wurden. An dieser Stelle weisen die Autoren darauf hin, dass die wichtigste Informationsquelle für die Analyse das Domänen- und Fachwissen der beteiligten Personen ist.

CHASSIS ist initial mit einer Fallstudie aus dem Flugverkehr – Air Traffic Management (ATM) – anhand eines Remote-Tower-Beispiels evaluiert worden [109, 157]. Außerdem wurde der Ansatz durch die Forscher Schmittner et al. auf ein Steuergerät (ECUs) angewendet [180]. Bei Letzterem wurde der Anwendungsfall *Aufspielen einer Firmware über eine Funkschnittstelle* betrachtet. Hier konnte mit dem Leitwort *EARLY* eine Verletzung der Verfügbarkeit aufgedeckt werden. Die identifizierte Bedrohung entsprach der Nicht-Verfügbarkeit einer ECU, sollte ein Firmware Over-the-Air (FOTA) Update eingespielt werden, obwohl sich die Ziel-ECU noch im Normalbetrieb befindet. Mit dem Leitwort *OTHER THAN* konnte eine Bedrohung identifiziert werden, bei der das Firmware-Update auf eine falsche ECU eingespielt werden kann. Mit dem Leitwort *MORE* wurde anschließend eine Bedrohung aufgedeckt, bei der ein Update mehrmals eingespielt wird und die Verfügbarkeit der Ziel-ECU nicht mehr gegeben ist.

A.2 Ergänzende Erläuterungen zu Six-Step Model

Im folgenden werden die detaillierten Methodenschritte des Six-Step Model (SSM) aufgezeigt. Hierzu dient Abbildung A.5, welche die Methodenschritte 1 bis aufzeigt. Im ersten Schritt werden die System-Funktionen beschrieben und ein Ziel-Baum (GT) konstruiert, der die Beziehungen zwischen den Haupt- und Nebenfunktionen identifiziert, was einem GTST-MLD entspricht. Das bedeutet, dass die Wurzel des GT die Ziel-Funktion widerspiegelt und die nachfolgenden Blätter die Haupt- und Nebenfunktionen darstellen.

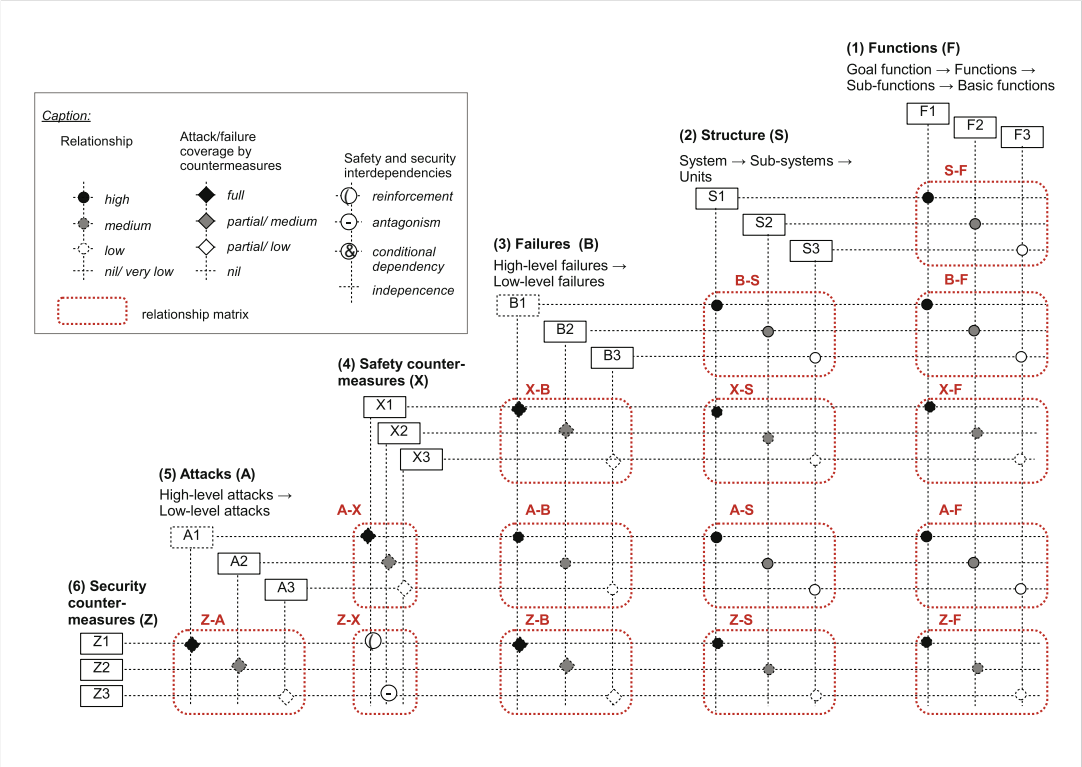


Abbildung A.5: Six-Step Model (SSM) mit den Beziehungsmatrizen, welche den Ergebnissen aus den Schritten 1 bis 6 entsprechen [14]. Der Ansatz unterscheidet vier Typen von Abhängigkeiten: i) *Bedingte Abhängigkeiten*: Security ist eine Voraussetzung für Safety und umgekehrt, ii) *Stärkung*: Safety- und Security-Gegenmaßnahmen können sich gegenseitig verstärken, iii) *Antogonismus*: Safety und Security können sich gegenseitig schwächen, iv) *Unabhängigkeit*: Keine Interaktion zwischen Safety und Security.

Im zweiten Schritt wird anhand des Erfolgsbaumes die Struktur des Systems festgelegt. Außerdem werden die Beziehungen zwischen der Systemstruktur und den Funktionen identifiziert. Hierzu wird das System in seine Teil- und Unterstützungssysteme zerlegt. Anschließend wird das MLD verwendet, um die Beziehungen zu modellieren. Daraus lässt sich die erste Beziehungsmatrix (Struktur-Funktionen) in Abbildung A.5 aufstellen, welche die Beziehungen zwischen den Teil-Systemen, Komponenten und den System-Funktionen identifiziert.

In Schritt 3 werden anschließend die System-Ausfälle identifiziert und in das Modell integriert. Ergebnis in diesem Schritt sind die Beziehungsmatrizen *Ausfälle und Struktur (B-S)* sowie *Ausfälle und Funktionen (B-F)* (siehe Abbildung A.5). Letztere zeigt, wie Ausfälle System-Funktionen beeinflussen können, die B-S Matrix hingegen zeigt, welche Systemeinheiten von einem Ausfall betroffen sind.

In Schritt 4 werden die Safety-Gegenmaßnahmen zum Modell hinzugefügt. Außerdem werden Beziehungen zwischen den Maßnahmen und den Ausfällen in der Matrix X-B, den Maßnahmen und Strukturen in der Matrix X-S sowie die Beziehungen zwischen den Maßnahmen und Funktionen in Matrix X-F festgehalten. Die in der Matrix X-B von Abbildung A.5 gezeigten Raute-Symbole spezifizieren die Abdeckung der Safety-Gegenmaßnahmen. Weiße Rauten entsprechen einer geringen, graue einer mittleren und schwarze Rauten einer vollen Abdeckung. Die X-S Matrix zeigt somit wie Safety-Gegenmaßnahmen die System-Struktur beeinflussen können und ob neue Elemente hinzugefügt werden müssen, falls Safety-Gegenmaßnahmen diese benötigten. Matrix X-F zeigt anschließend wie System-Funktionen durch die Safety-Gegenmaßnahmen beeinflusst werden.

In Schritt 5 werden die Angriffe zum Modell hinzugefügt und deren Zusammenhänge mit den übrigen Elementen identifiziert. Der Ansatz betrachtet ausschließlich Cyber-Angriffe und keine physischen Angriffe. Zur Bestimmung möglicher Angriffe können Angriffsbäume eingesetzt werden [14, 171]. Als Ergebnisse in diesem Schritt sind vier Beziehungs-Matrizen zu nennen. Matrix A-X zeigt die Zusammenhänge zwischen den Angriffen und Safety-Maßnahmen sowie dem Grad der Abdeckung von Safety-Maßnahmen bezogen auf Cyber-Angriffe. Matrix A-B zeigt die Beziehungen zwischen den Angriffen und Ausfälle, Matrix A-S repräsentiert hingegen, wie sich Angriffe auf Teile des Systems auswirken können. Matrix A-F zeigt außerdem, welchen Einfluss Cyber-Angriffe auf Funktionen haben können. Sind die Matrizen erzeugt und damit Schritt 5 abgeschlossen, sollte zu Schritt 3 und 4 zurück gegangen werden, um iterativ festzustellen, ob sich Änderungen aufgrund der Angriffe aus Schritt 5 ergeben haben.

In Schritt 6 werden anschließend die Security-Gegenmaßnahmen eingefügt und die Beziehungen zwischen den Elementen im Modell bestimmt. Dies resultiert in fünf Beziehungsmatrizen um die Zusammenhänge zwischen Security-Gegenmaßnahmen und den Angriffen (Z-A), Safety-Gegenmaßnahmen (Z-X), Ausfällen (Z-B), Strukturelementen (Z-S) sowie Funktionen (Z-F) aufzuzeigen. Matrix Z-X in Abbildung A.5 beschreibt außerdem die Abdeckung der Angriffe durch Safety-Maßnahmen. Stellt sich hierbei heraus, dass ein Angriff durch eine Safety-Gegenmaßnahme (Matrix A-X) vollständig abgedeckt ist, so besteht nach Ansicht der Autoren keine Notwendigkeit für das Hinzufügen einer Security-Maßnahme [14]. Mit Matrix Z-B wird zusätzlich gezeigt, ob Ausfälle durch Security-Gegenmaßnahmen abgedeckt werden. In Kombination mit der X-B Matrix kann Z-B verwendet werden, um den Grad der Abdeckung durch Safety- und Security-Gegenmaßnahmen festzustellen. Matrix Z-X dient anschließend

der Analyse zur Konsistenz von Safety- und Security-Maßnahmen, indem die Abhängigkeiten zwischen Gegenmaßnahmen aufgezeigt werden. Ist Schritt 6 abgeschlossen, wird wie bei Schritt 5, ein Teil der vorherigen Schritte wiederholt. Hier sind die Schritte 3 bis 5 zu wiederholen, um Änderungen aus Schritt 6 zu überprüfen.

A.2.1 Erweiterung Six-Step Model

Abbildung A.6 zeigt den erweiterten Ablauf des SSM mit den eingeführten IFDs.

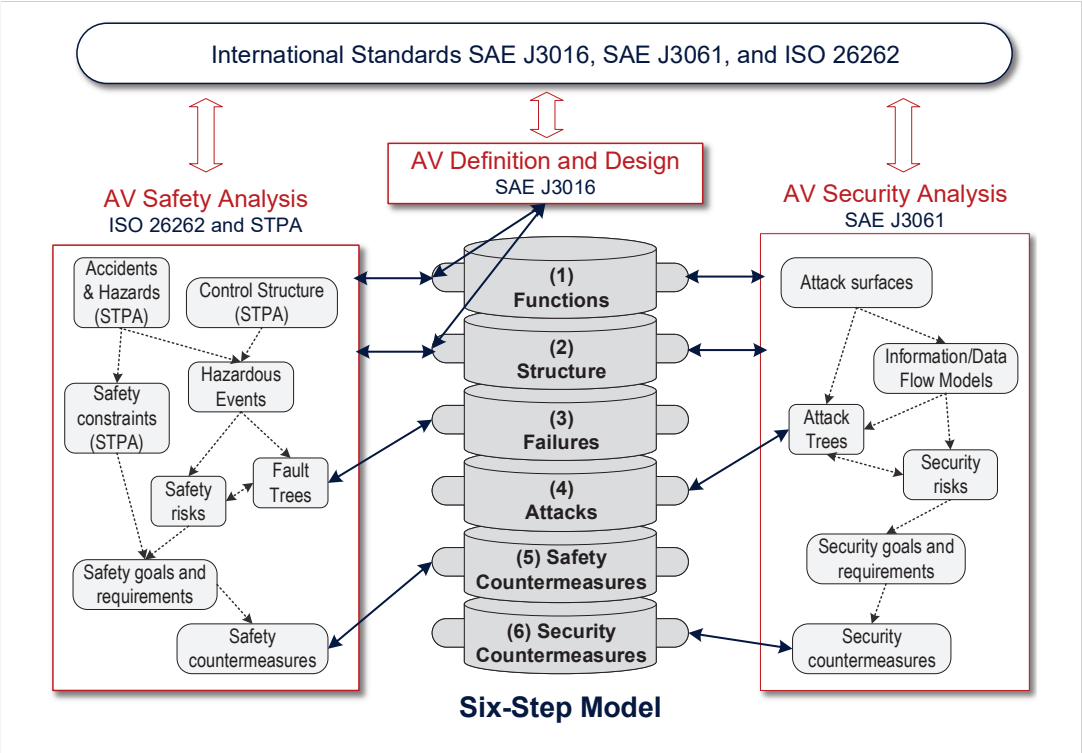


Abbildung A.6: Erweitertes Six-Step Model (SSM) mit IFD im Kontext von autonomen Fahrzeugen [171]. Die linke Seite zeigt den Ablauf der Safety-Analyse und die Rechte den der Security. Für die Security-Analyse werden Informationsfluss-Graphen und Angriffsbäume vorgeschlagen.

In Schritt 3 wird die Gefährdungsanalyse durchgeführt und in Schritt 4 die Bedrohungen für das Fahrzeug analysiert. Im letzteren werden die IFDs eingesetzt, um Angriffe zu identifizieren und damit Angriffsbäume konstruieren zu können. Anschließend werden in Schritt 5 von Abbildung A.6 die Safety-Gegenmaßnahmen festgelegt und in Schritt 6 die Security-Gegenmaßnahmen ausgewählt. Auch hier weisen die Autoren darauf hin, dass Änderungen in den Schritten 5 und 6 ein Zurückkehren zu Schritt 2 impliziert und die Schritte 3 bis 6 wiederholt werden müssen [171].

A.3 Ergänzende Erläuterungen zu FMVEA

Abbildung A.7 zeigt das zugrundeliegende Modell von FMVEA. Es sind die Elemente dargestellt, welche die für die Security-Analyse benötigt werden. Dies sind: die

Schwachstelle (eng. Vulnerability), der Bedrohungsagent (engl. Threat Agent) sowie der Bedrohungsmodus (engl. Threat Mode). Konsequenzen die von Ausfällen oder Bedrohungen resultieren, werden unter dem Begriff Auswirkung (engl. Effect) verstanden.

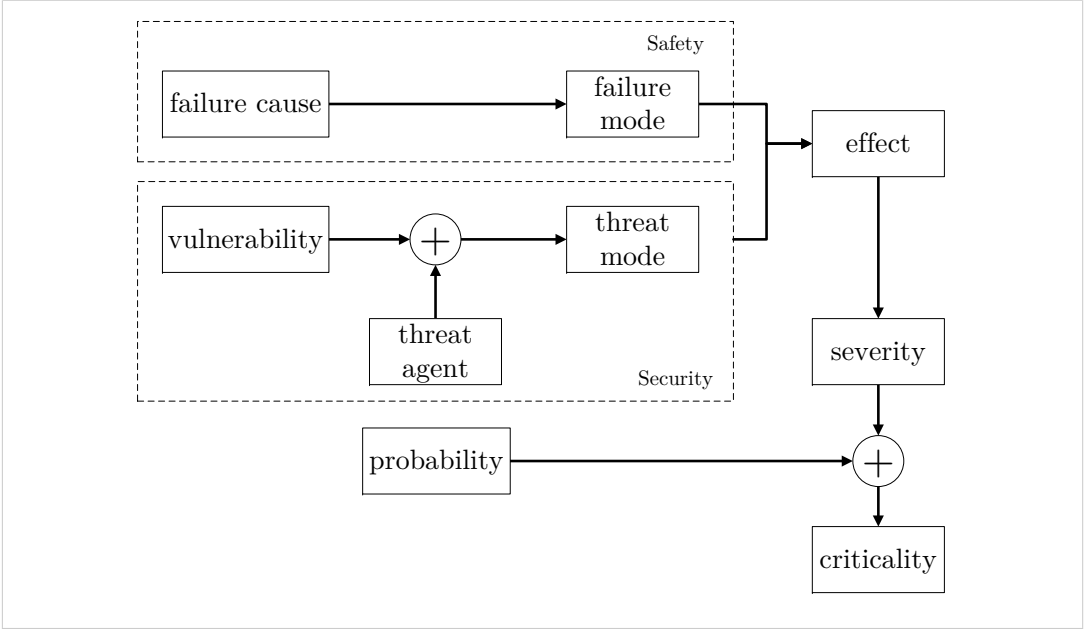


Abbildung A.7: Ursachen-Wirkungs-Modell der FMVEA definiert und auf FMEA basiert [179].

Die Eintrittswahrscheinlichkeit (engl. Probability) beinhaltet, nach Abbildung A.7, die Ausfall- und Angriffswahrscheinlichkeiten bezogen auf Safety und Security. Die Bedrohungsmodi (engl. Threat Modes) werden anhand von STRIDE klassifiziert und daran die verletzten Security-Attribute abgeleitet. Die genannten Begrifflichkeiten finden sich ebenfalls im Ablauf-Diagramm der Methode und zeigen die folgenden Einzelschritte (siehe Abbildung A.8) [41]:

1. Eine funktionale Analyse auf Systemebene, um eine Komponentenliste zu erstellen.
2. Die Auswahl einer Komponente, die der Analyse unterzogen werden soll.
3. Identifikation von Ausfall- und Bedrohungsmodi für die Komponente.
4. Festlegung der Auswirkungen für die identifizierten Ausfall- und Bedrohungsmodi.
5. Bestimmung der Schwere der zuvor erkannten Auswirkungen.
6. Identifizierung möglicher Ausfallursachen (Safety), Schwachstellen (Security) und Angreifer (Security).
7. Festlegen der Eintrittswahrscheinlichkeiten für Ausfälle und Angriffe anhand der identifizierten Ursachen.

8. Berechnung des Risikos.

Das zu analysierende System wird bei FMVEA in Komponenten aufgeteilt und die Ausfallmodi für jede Komponente identifiziert und im dritten Schritt die Safety-Ausfälle und die Security-Bedrohungen – getrennt voneinander – analysiert. Im linken Pfad von Abbildung A.8 wird hierzu für jeden Ausfallmodus die Auswirkungen, die dazugehörige Schwere und die potenziellen Ursachen untersucht. Die Häufigkeiten und Eintrittswahrscheinlichkeiten der Ausfallraten werden dabei abgeschätzt.

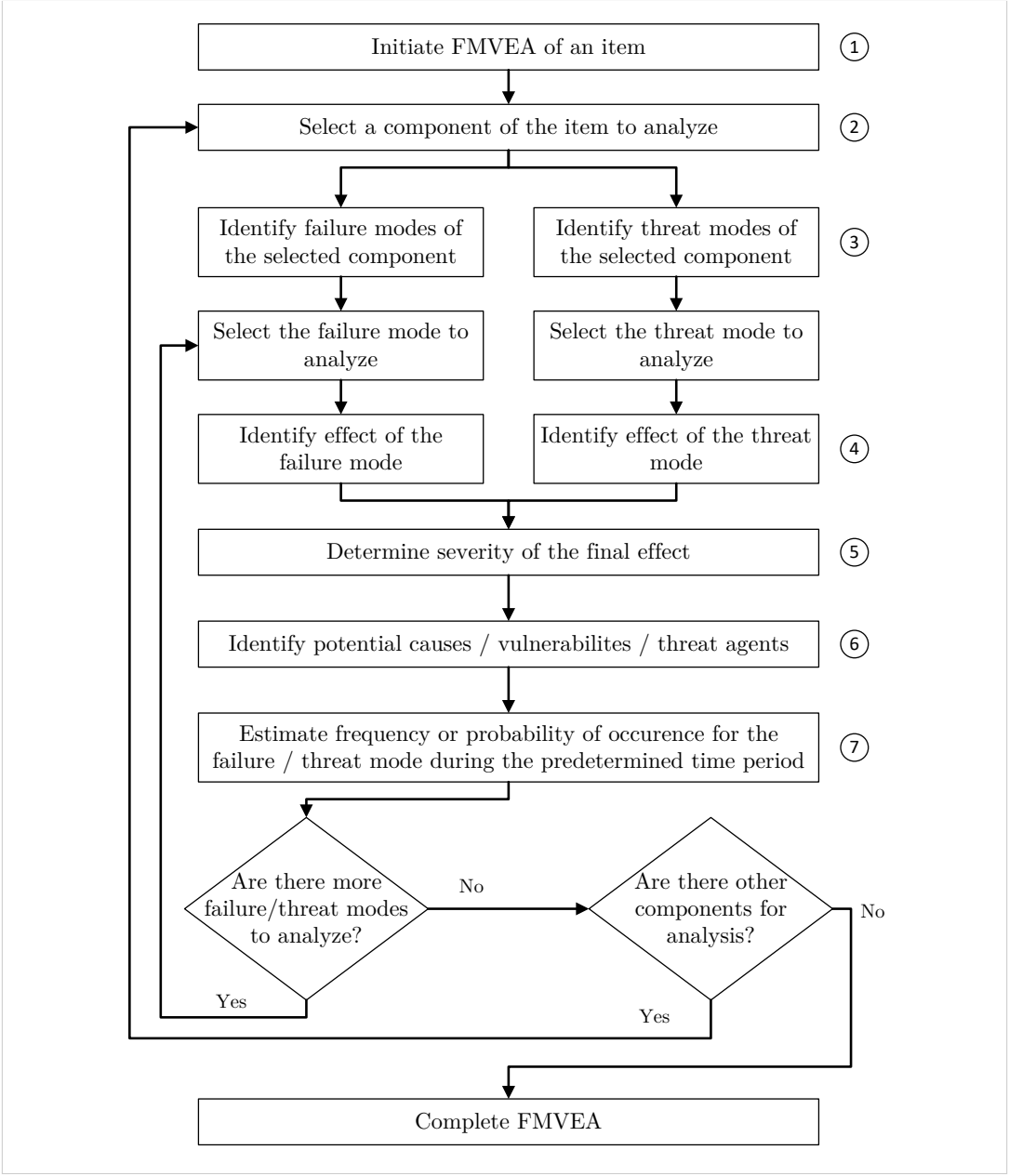


Abbildung A.8: Ablauf-Diagramm der FMVEA Methode, das zwei parallele Pfade aufzeigt [179]. Der linke Pfad zeigt dabei den Ablauf der Safety-Analyse und der Rechte, das Vorgehen für die Security-Analyse. Für beide Pfade müssen jeweils sieben Schritte durchlaufen werden, welche das Festlegen des Analyseumfanges (1-2), das Identifizieren der Gefahren beziehungsweise der Bedrohungen (3-4), die Risikobewertung (5+7) und dem Aufdecken der zugrundeliegenden Ursachen (6) entsprechen.

Für den rechten Pfad von Abbildung A.8, welcher der Security-Analyse entspricht, nutzt FMVEA die STRIDE-Methodik und das TMT [154] von Microsoft, um Cyber-Bedrohungen zu identifizieren. Der Ansatz setzt für die Security-Analyse auf Datenfluss-Diagramme, die in drei Abstraktionsebenen unterschieden werden. Auf der höchsten Abstraktionsstufe, dem Kontext-Datenfluss-Diagramm sind die Komponenten aufgeführt, die zur Realisierung einer bestimmten Funktion notwendig sind. In der nächsten

Stufe werden die Ausfall- und Bedrohungsmodi zugewiesen, die aus Katalogen entnommen werden, die vor der Anwendung von FMVEA erzeugt wurden. Das kann durch eine andere und vorgelagerte Analyse-Methode oder durch historisches Wissen erreicht werden. Die Ausfallmodi beschreiben hierbei die Safety-Aspekte, die Bedrohungsmodi hingegen die Security-Aspekte [63]. Auf der dritten und detailliertesten Stufe zeigt das Datenfluss-Diagramm die konkreten Funktionen des Systems, anhand derer die potenziellen Ursachen für die Ausfälle oder Bedrohungen bestimmt werden. Ist diese Phase abgeschlossen, werden nachfolgend die dazugehörigen und potenziellen Ursachen aufgedeckt. Für Safety sind es Fehler im System, für Security sind es Schwachstellen und Schwächen, die das System besitzen kann. Somit ist die wesentliche Voraussetzung für eine Verletzung der Security das Vorhandensein einer Schwäche oder einer Schwachstelle im System. Letzteres ist bei FMVEA vergleichbar zu einer Ausfallursache, bekannt aus der Safety. Anschließend werden für zufällige Ursachen (Safety) statistische Daten zur Bestimmung der Wahrscheinlichkeit verwendet, für böswillige Ursachen, die einer Bedrohung entsprechen, werden Eintrittswahrscheinlichkeiten anhand der Bedrohungs- und Systemeigenschaften bestimmt [180].

Zur Unterstützung der Bestimmung von Angriffswahrscheinlichkeit und der Schwere der Auswirkungen definiert FMVEA eigene Begrifflichkeiten. So entspricht der in Abbildung A.8 gezeigte Begriff *Threat Agent* einem aktiven Element (Person), welches versucht eine Schwachstelle auszunutzen. Im Allgemeinen ist dies mit dem Begriff des Angreifers gleichzusetzen. Der Term *Threat Mode* hingegen beschreibt, auf welche Weise eine Schwachstelle ausgenutzt werden kann. Dies hängt bei FMVEA vom System und von den Fähigkeiten des Angriffes ab. Vergleichbar mit den Auswirkungen eines Ausfalles (Safety) stellt der *Threat Effect* die Konsequenzen bezogen auf den Betrieb, die Funktion oder den Status des Systems dar. Somit drückt der Term die Verletzung von Qualitätsattributen des Systems aus. Die Festlegung der Schwere einer Attacke kann nach Ansicht der Autoren durch Safety-Fachleute geschehen, die Eintrittswahrscheinlichkeit jedoch nicht, da diese für Security und Safety unterschiedlich definiert sind. Letztere entspricht auf der Security-Seite der Angriffswahrscheinlichkeit (engl. Attack Probability). Sie beschreibt, wie wahrscheinlich es ist, das ein Angreifer (Threat Agent) Aktionen durchführen wird, die Konsequenzen auslösen können. Die für die Bestimmung der Angriffswahrscheinlichkeit notwendigen Kriterien sind in Tabelle A.1 aufgeführt.

A.4 Ergänzende Erläuterungen zu EVITA

Ergänzend zu den Erläuterungen in Abschnitt 3.3.1 zeigt Abbildung A.9 die bei EVITA definierte Architektur, welcher der Analyse zugrunde liegt.

Tabelle A.1: Kriterien für die Bestimmung der Angriffswahrscheinlichkeit bei FMVEA, basierend auf [179]. Die in den Klammern aufgezeigten Werte entsprechen der quantitativen Bewertung.

Kategorie	Klasse		
Angreifer-Motivation	Gelegenheitsziel (1)	Mäßig interessiert (2)	Hauptziel (3)
Angreifer-Fähigkeiten	Gering (1)	Mittel (2)	Hoch (3)
Erreichbarkeit des Systems	Keine (1)	Privates Netzwerk (2)	Öffentliches Netzwerk (3)
Ungewöhnlichkeit des Systems	Eingeschränkt (1)	Kommerziell verfügbar (2)	Standard (3)

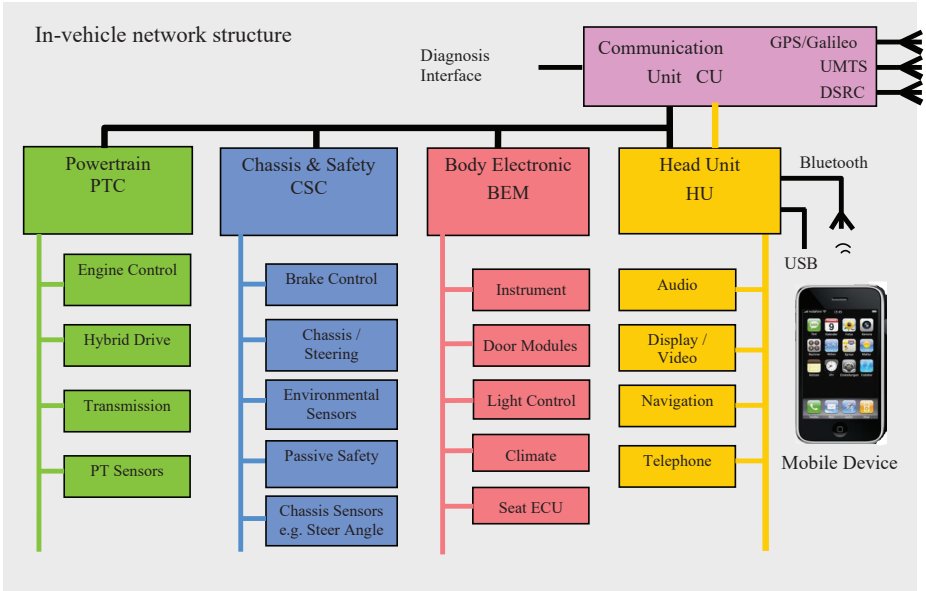


Abbildung A.9: E/E-Architektur, die dem EVITA Projekt zugrunde liegt [83]. Die Architektur zeigt vier Domänen mit dem jeweiligen Domänen-Controllern und ein Gateway (Communication Control Unit) zur Kommunikation mit der Umwelt. Der graue Bereich markiert dabei das Fahrzeug-interne Netzwerk, der weiße Bereich die Fahrzeug-externe Kommunikation wie beispielsweise Mobilfunk.

Anforderungen an diese Architektur sind unter anderem die sichere Erzeugung, Speicherung und Verarbeitung von Security-kritischem Material wie zum Beispiel kryptografischen Schlüsseln. Diese Anforderungen werden auf vier grundlegende Schutzziele heruntergebrochen:

- Privacy: Dem Schutz von privaten Daten des Fahrers sowie dem geistigen Eigentum des Herstellers.
- Financial: Dem Schutz von Finanztransaktionen und Zugang zum Fahrzeug in Bezug auf Diebstahl.

- Operational: Die Gewährleistung des korrekten und effizienten Betriebs von Fahrzeugsystemen und der V2V-Kommunikation.
- Safety: Die Gewährleistung der Betriebssicherheit des Fahrzeuges.

Diese Schutzziele werden anschließend mit Anwendungsfällen (engl. Use-Cases) kombiniert, um die Cyber-Bedrohungen und Security-Maßnahmen ableiten zu können. Das Verständnis über die Betriebsumgebung und das Systemverständnis wird hierbei durch die Anwendungsfälle bereitgestellt [167]. Das Projekt sieht dabei die folgenden fünf relevanten Anwendungsfälle [83, 84]:

- Kommunikation zwischen Fahrzeugen (V2V),
- Kommunikation zwischen dem Fahrzeug und der Infrastruktur (V2I),
- Einbindung von Mobilgeräten wie Smartphones oder USB Sticks,
- Aftermarket-Anwendungen,
- Werkstatt- und Diagnoseprozeduren.

Auffällig hierbei ist, dass stets eine Kommunikation mit dem Fahrzeug-internen Netzwerk und der Außenwelt betrachtet wird. So ist beispielsweise im ersten Anwendungsfall eine Kommunikation zwischen Fahrzeugen notwendig und im vorletzten ein Informationsfluss zwischen dem Fahrzeug und einer externen Einheit, wie es ein OBD-Adapter oder Diagnosetester sein kann. Für die Bedrohungsanalyse startet der Ansatz mit einer funktionalen Beschreibung. Diese wird von den Anwendungsfällen abgeleitet und als Pfade beschrieben. Die funktionalen Pfade werden anschließend auf eine konkrete Architektur übertragen und mit weiteren Details, wie zeitlichen Beschränkungen, angereichert. Hieraus können Security-Anforderungen abgeleitet werden, die nach der CIA-Triad-Nomenklatur [167] beschrieben werden und sich folgendermaßen darstellen [84]:

- Integrität von Hardware Security Module (HSM),
- Integrität und Authentizität von Software und Daten im Fahrzeug,
- Integrität und Authentizität der Fahrzeug-internen Kommunikation,
- Vertraulichkeit von Software und Daten im Fahrzeug,
- Vertraulichkeit der Fahrzeug-internen Kommunikation,
- Nachweis der Integrität der eingesetzten Plattform sowie der Authentizität gegenüber anderen Entitäten,
- Zugriffskontrolle zu Fahrzeug-internen Daten und Ressourcen.

Anhand der in Abbildung A.9 gezeigten Architektur, den definierten Anwendungsfällen sowie den vier Schutzzielen, werden die Cyber-Bedrohungen identifiziert und der

Schaden bestimmt. Hierzu definiert das Projekt einen einheitlichen Aufbau des Baumes, der in Abbildung A.10 gezeigt ist.

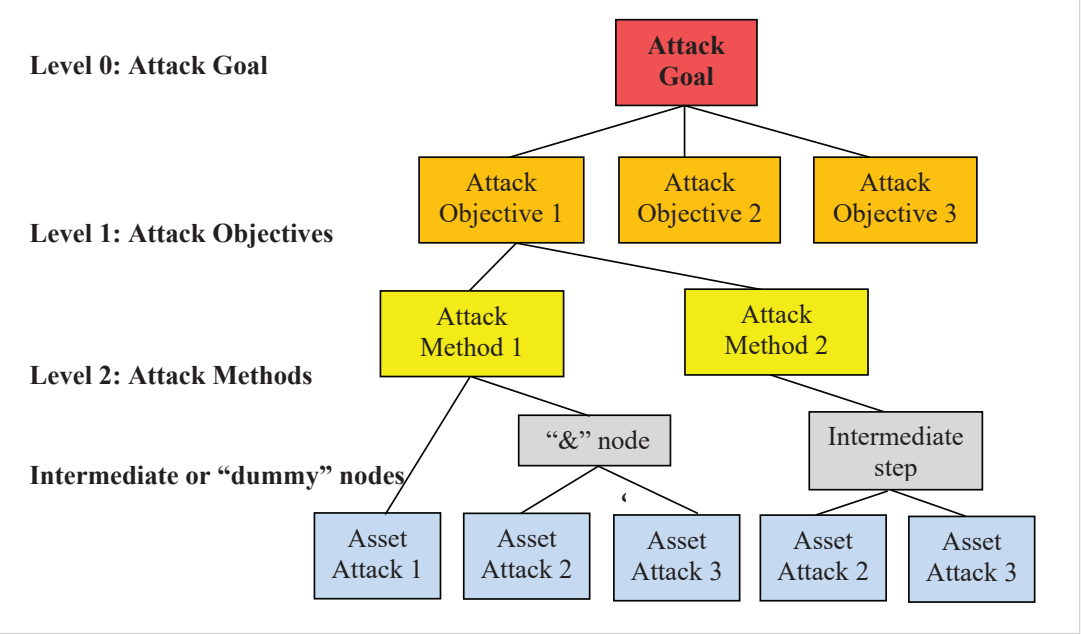


Abbildung A.10: Generischer Angriffsbaum, der im Projekt zur Dokumentation der Bedrohungen verwendet wird. Die Wurzel des Baumes entspricht dem Hauptziel des Angriffes und die Ebene darunter, den Teilzielen, die der Angreifer erreichen möchte. Die gelben Rechtecke auf der dritten Ebene, entsprechen der Angriffstechnik, die für den Angriff notwendig ist. Die unterste Ebene zeigt außerdem die durch den Angriff verletzten Vermögenswerte auf [83, 167].

Die Wurzel des Baumes (Level 0) in Abbildung A.10 repräsentiert das Ziel des Angriffes und Level 1 die Teilziele, die ein Angreifer erreichen möchte. Level 2 beschreibt als Nächstes die Angriffsmethoden, die eingesetzt werden können. Hierbei kann es notwendig sein, dass mehrere Angriffe zugleich ausgeführt werden müssen, was einem UND-Knoten entspricht oder im sequentiellen Fall zu einem Zwischenschritt (engl. intermediate step) führt. Die tiefste Ebene wird durch die Blätter des Baumes repräsentiert, welche die Verletzung der relevanten Vermögenswerte aufzeigen und in Abbildung A.10 durch die hellblauen Rechtecke dargestellt sind. Den Blättern werden außerdem die Wahrscheinlichkeiten für einen erfolgreichen Angriff auf die Vermögenswerte angeheftet. Die dazu notwendigen Eintrittswahrscheinlichkeiten werden anhand des Angriffspotentials abgeleitet, welches für jedes Blatt bestimmt werden muss. Je höher das Angriffspotential, desto geringer ist die Wahrscheinlichkeit des Eintretens eines Angriffes. Das Angriffspotential kann somit als Aufwand beziehungsweise Schwierigkeitsgrad für die Durchführung eines Angriffs interpretiert werden.

Das Vorhaben bestimmt anhand der folgenden fünf Kategorien das Angriffspotential [83] (für eine detaillierte Ausführung, siehe [84]):

- Benötigte Zeit (engl. Elapsed time),

- Erfahrung (engl. Expertise),
- Wissen über das System (engl. Knowledge of system),
- Gelegenheit für den Angriff (engl. Window of opportunity),
- Ausrüstung (engl. Equipment)

Mittels der Summe der vergebenen Werte für das Angriffspotential wird nach Tabelle A.2 die Angriffswahrscheinlichkeit bestimmt.

Tabelle A.2: Zuordnung des Angriffspotentials zur jeweiligen Angriffswahrscheinlichkeit. Das Angriffspotenzial entspricht dem Aufwand zur Identifizierung und Ausnutzung des Angriffsszenarios, basierend auf [167, Seite 89].

Werte	Angriffspotenzial	Angriffswahrscheinlichkeit P
0-9	Grundlegend	5
10-13	Grundlegend erhöht	4
14-19	Moderat	3
20-24	Hoch	2
≥ 25	Überdurchschnittlich hoch	1

Die gezeigten Kategorien zur Bestimmung des Angriffspotentials sowie die Metriken zur Überführung in einen quantitativen Wert sind aus der ISO 15048 [100] abgeleitet und werden aufsteigend von den Blättern des Baumes bis zu Level 1 propagiert. Hierbei werden jeweils die Wahrscheinlichkeitswerte der Kinder eines Knotens betrachtet. Sind die Kinder (P_i) durch eine *ODER*-Verknüpfung verknüpft, resultiert die kombinierte Angriffswahrscheinlichkeit A zu $A_{ODER}(P_i) = \max\{P_i\}$. Sind die Kinder hingegen mit einer *UND*-Verknüpfung verknüpft, so ergibt sich die kombinierte Angriffswahrscheinlichkeit der Kinder P_i zu $A_{UND}(P_i) = \min\{P_i\}$ und entspricht damit einem Übernehmen des kleinsten Wertes aller Kinder des gleichen Knotens. Für gemischte Verknüpfungen von Kindern eines Knoten werden die Rechenvorschriften kombiniert [167]. Beispielsweise ergibt sich so die kombinierte Angriffswahrscheinlichkeit (A) für einen Knoten, bei dem zwei Kinder $\{P_1, P_2\}$ mit *UND* und ein Kind $\{P_3\}$ mit *ODER* verknüpft ist, zu $A_{ODER,UND}(P_i) = \max[\min\{P_1, P_2\}, P_3]$.

Zur Bestimmung der Schwere des Schadens bezieht sich EVITA auf die vier oben genannten Schutzziele: Safety, Privatsphäre (engl. Privacy), Finanziell (engl. Financial) und Betriebsfähigkeit (engl. Operational). Der Schweregrades S_i wird anhand der fünf Klassen in Tabelle A.3 bestimmt und ein quantitativer Wert von 0 bis 4 zugeordnet.

Tabelle A.3: Festlegung des Schweregrades S_i der Bedrohungen, durch Festlegung des Schadens für die vier Schutzziele Safety, Datenschutz, Finanziell und der Betriebsfähigkeit [167].

Schweregrad S_i	Safety	Datenschutz	Finanziell	Betriebsfähigkeit
0	Keine Verletzungen	Kein nicht-autorisierter Zugriff auf Daten	Kein finanzieller Verlust	Kein Einfluss auf die Betriebsleistung
1	Leichte Verletzungen	Zugriff auf anonyme Daten	Geringer Verlust (≈10 €)	Auswirkungen für den Fahrer nicht erkennbar
2	Schwere Verletzungen	Identifizierung von Fahrzeug und Fahrer	Moderate Verluste (≈100 €)	Fahrer bemerkt Leistungseinbuße
3	Lebensbedrohliche Verletzungen	Nachverfolgbarkeit des Fahrzeugs oder Fahrer	Hohe Verluste (≈1000 €)	Signifikante Leistungseinbuße
4	Lebensbedrohliche Verletzungen bei mehreren Fahrzeugen	Nachverfolgbarkeit von Fahrer und Fahrzeug bei mehreren Fahrzeugen	Hohe Verlsute bei mehreren Fahrzeugen	Signifikante Leistungseinbußen bei mehreren Fahrzeugen

Ist der Wert für den Schaden hinsichtlich der Schutzziele bestimmt und der Wert für die kombinierte Angriffswahrscheinlichkeit A berechnet, so kann das Risiko bestimmt werden. Hierzu stehen qualitative Klassen von $R0$ bis $R7$ zur Verfügung, wobei $R0$ das kleinste und $R7+$ das höchste Risiko darstellt. EVITA unterscheidet beim Risiko zwischen dem Schutzziel Safety (S_S) und den Schutzzielen Privacy (S_P), Financial (S_F) und Operational (S_O). Werden Letztere drei und damit nicht die Safety verletzt, so muss zur Risikobestimmung Tabelle A.4 verwendet werden.

Tabelle A.4: Bestimmung des Risikos anhand der Schwere des Schadens (S_i) und der kombinierten Angriffswahrscheinlichkeit (A) für Cyber-Bedrohungen, welche nicht die Safety verletzen [167].

Risiko-Level	Kombinierte Angriffswahrscheinlichkeit A				
	$A = 1$	$A = 2$	$A = 3$	$A = 4$	$A = 5$
$S_i = 1$	R0	R0	R1	R2	R3
$S_i = 2$	R0	R1	R2	R3	R4
$S_i = 3$	R1	R2	R3	R4	R5
$S_i = 4$	R2	R3	R4	R5	R6

Falls ein Blatt im Angriffsbaum das Schutzziel Safety (S_S) beeinflusst, so bezieht EVITA die Kontrollierbarkeit (Controllability) aus der ISO 26262 [99] für die Bewer-

tung des Risikos mit ein. Der Wert für die Kontrollierbarkeit beschreibt, inwieweit ein Fahrer die Gefahrensituation beherrschen und zu welchem Grad er die Schwere (engl. Severity) des Unfalles mildern kann. Die festgelegten Werte für die Schwere des Schadens (S_i) der jeweiligen Bedrohungsklasse, das kombinierte Angriffspotential (A) und die Kontrollierbarkeit (C) der Bedrohungen werden in Tabelle A.5 gegenübergestellt und der finale Risikowert R bestimmt.

Tabelle A.5: Zugrundeliegende Metriken die zur Bewertung des Angriffspotentials verwendet werden, sollte die Cyber-Bedrohung die Safety (S_S) verletzen [84, 167].

Kontrollierbarkeit C	**Schweregrad** S_s	**Kombinierte Angriffswahrscheinlichkeit** A				
		$A=1$	$A=2$	$A=3$	$A=4$	$A=5$
C=1	$S_s=1$	R0	R0	R1	R2	R3
	$S_s=2$	R0	R1	R2	R3	R4
	$S_s=3$	R1	R2	R3	R4	R5
	$S_s=4$	R2	R3	R4	R5	R6
C=2	$S_s=1$	R0	R1	R2	R3	R4
	$S_s=2$	R1	R2	R3	R4	R5
	$S_s=3$	R2	R3	R4	R5	R6
	$S_s=1$	R3	R4	R5	R6	R7
C=3	$S_s=1$	R1	R2	R3	R4	R5
	$S_s=2$	R2	R3	R4	R5	R6
	$S_s=3$	R3	R4	R5	R6	R7
	$S_s=4$	R4	R5	R6	R7	R7+
C=4	$S_s=1$	R2	R3	R4	R5	R6
	$S_s=2$	R3	R4	R5	R6	R7
	$S_s=3$	R4	R5	R6	R7	R7+
	$S_s=4$	R5	R6	R7	R7+	R7+

A.5 Ergänzende Erläuterungen zu HEAVENS

Mit Abbildung A.11 ist zugrundeliegende Modell für HEAVENS gezeigt, das bereits in Abschnitt 3.3.2 angedeutet wurde.

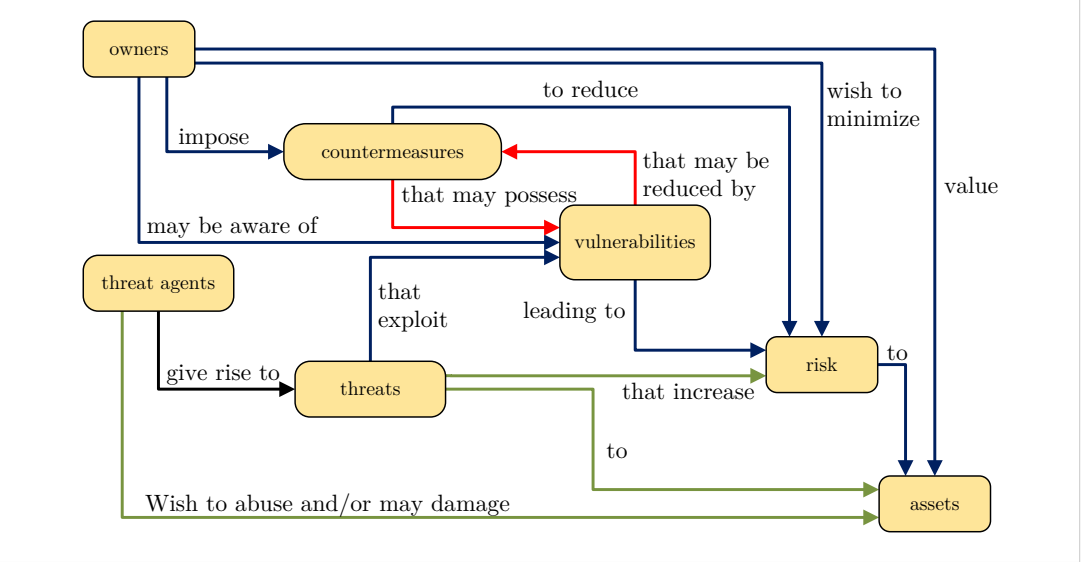


Abbildung A.11: Zugrundeliegendes Modell für die Bedrohungsanalyse und der Risikobewertung nach dem HEAVENS-Ansatz [146, Seite 48].

Darüber hinaus zeigt Abbildung A.12 die drei Schritte, die für eine Analyse mit HEAVENS durchgeführt werden müssen.

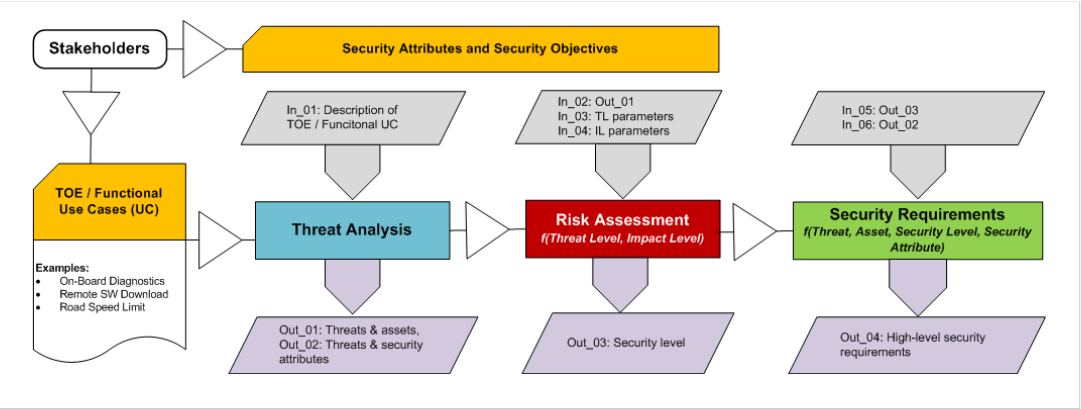


Abbildung A.12: Abläufe für die Identifikation der Anwendungsfälle, Durchführung der Bedrohungsanalyse und Risikobewertung sowie dem Ableiten der Security-Anforderungen [146, Seite 51].

Im ersten Schritt werden die funktionalen Anwendungsfälle für das TOE definiert. Von diesen werden die Vermögenswerte, Bedrohungen und dazugehörigen Security-Attribute abgeleitet. Hierzu verwendet das Projekt einen für E/E-Architekturen modifizierten STRIDE-Ansatz (siehe [137]). Die Autoren argumentieren die Anpassung mit der Situation, dass STRIDE mit einem Fokus auf Software-Systeme entwickelt wurde und

nicht für E/E-Architekturen. Es ist allerdings vorteilhaft, das der Ansatz bedrohungs- und angreiferzentriert ist und somit eine Fokussierung auf die Identifikation von Konsequenzen anstatt auf einzelne Angriffe gesetzt wird [146]. STRIDE bedient sich einer Menge von Schlüsselwörtern, um eine strukturierte Bedrohungsanalyse zu ermöglichen und die Anwendung der Methodik zu unterstützen. Als Werkzeug für die Bedrohungsanalyse setzt das Projekt Microsoft's TMT-Werkzeug ein, das in dem rot-markierten Methoden-Schritt von Abbildung A.13 Anwendung findet [92, 154].

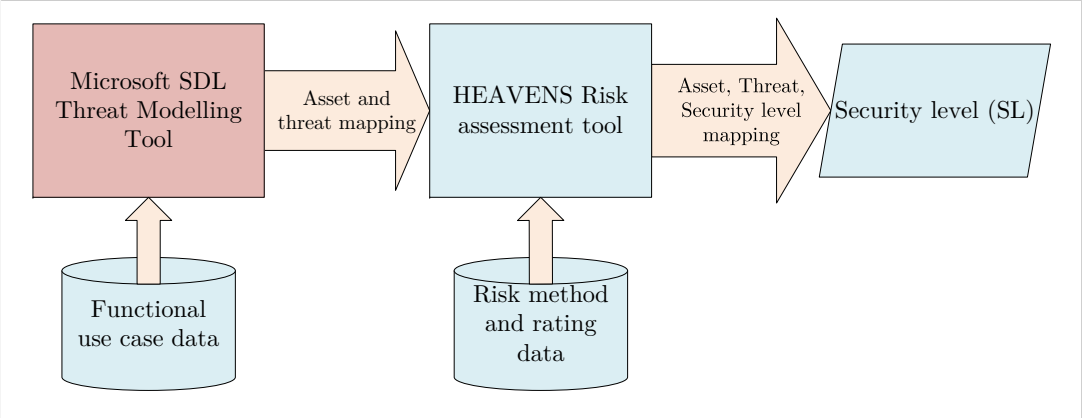


Abbildung A.13: Der von HEAVENS vorgeschlagene Ablauf zur Bedrohungsanalyse mit Werkzeugunterstützung, basierend auf dem TMT von Microsoft und dem im Projekt entwickelten Risikobewertungswerkzeug [21, S.69].

Nach der Identifikation der Bedrohungen werden die dazugehörigen Assets sowie die relevanten Security-Attribute festgelegt und zugeordnet. Diese werden im zweiten Schritt in die Risikobewertung übernommen und es werden Werte bestimmt, die einem SL zugeordnet werden (siehe zweites Rechteck in Abbildung A.13). Für die Risikoanalyse setzt HEAVENS jedoch nicht die zu STRIDE gehörige Risikobewertungsmetrik DREAD ein, sondern verwendet Metriken aus dem EVITA-Projekt, die auf der ISO 15408 [100] basieren. Nach Meinungen der Autoren ist DREAD wegen inkonsistenter Bewertungen und hoher Subjektivität nicht für die automotive Domäne geeignet [146, Seite 49]. Die eigentliche Risikobewertung setzt sich aus drei Teilschritten zusammen: 1) der Bestimmung eines Bedrohungslevel (TL), das der Eintrittswahrscheinlichkeit einer Bedrohung entspricht, 2) der Festlegung der Schadenshöhe (IL) ausgelöst durch die Bedrohung sowie 3) der Berechnung des endgültigen Risikolevels (SL) für jede Bedrohung. Zur Festlegung der Bedrohungslevel orientiert sich das Projekt an den Metriken der Common Criteria (CC) [100], wobei nicht der gesamte Umfang verwendet wird. Übernommen wurden die Kategorien: Angreiferwissen (*Expertise*), Wissen über das Analyse-Objekt (*Knowledge about TOE*), die Gelegenheit für die Durchführung einer Manipulation (*Window of opportunity*) sowie die notwendige Ausrüstung (Equipment). Der Parameter *Elapsed Time* und *Motivation of the attackers* werden nicht von HEAVENS übernommen, da nach Ansicht der Autoren, beide keine primären Parameter bei der Bedrohungsbewertung sind und schwierig abzuschätzen

seien. HEAVENS setzt eine lineare Skalierung und eine einheitliche Gewichtung der Bewertungsparameter ein [21, Seite 56]. Diese werden, für jedes Bedrohungs-Asset-Paar, aufsummiert und anhand von festgelegten Wertebereichen einem Bedrohungslevel zugeordnet. Zur Bestimmung der Schadenshöhe werden vier Schadenskategorien betrachtet, die den Schutzzielen: *Safety*, *Financial*, *Operational*, *Privacy and Legisation* entsprechen. Die vier verwendeten Schadenskategorien beziehen sich somit auf Stakeholder wie dem OEM, Flottenbesitzer, Fahrzeugbesitzer und weitere. Die sich durch einen Angriff für den Angreifer ergebenden Vorteile werden hingegen nicht betrachtet [21, Seite 61]. HEAVENS gewichtet die Schutzziele *Safety* und *Financial* bei der Schadensfestlegung höher als die Schutzziele *Operational* und *Privacy and Legislation*, um eine Priorisierung der ersten beiden Schutzziele zu erreichen (siehe Abbildung A.14).

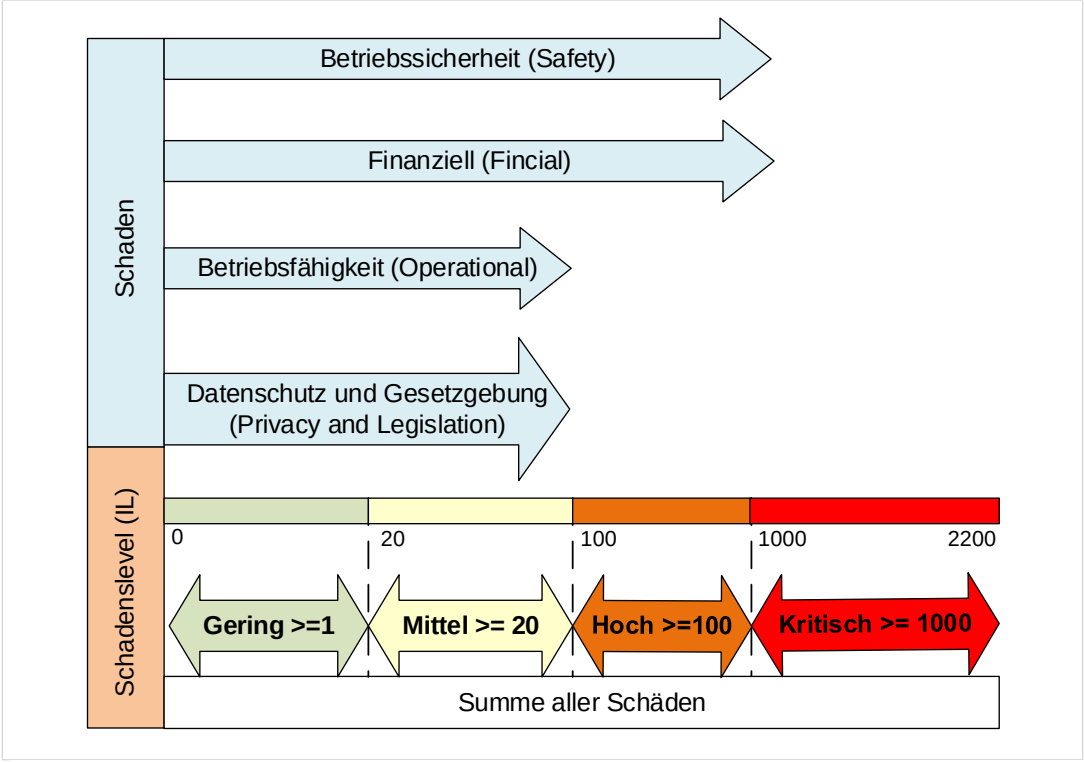


Abbildung A.14: Zuordnung der aufsummierten Schadenswerte zu einem qualitativen Parameter, der einem Schadenslevel (Impact Level) entspricht. Zu sehen sind vier Abstufungen: Gering, Mittel, Hoch und Kritisch basierend auf [21, Seite 66].

Durch Aufsummieren der Werte für die vier Schadenskategorien (Safety, Fincial, Operational, Privacy and Legislation) von Abbildung A.14, ergibt sich der gesamte Schadenswert, der als Impact Level (IL) bezeichnet wird. Mittels diesem und dem Bedrohungslevel TL lässt sich unter Anwendung von Tabelle A.6 das Security-Level (SL) bestimmen.

Die Autoren weisen auf die Problematik hin, dass einem Asset eine Reihe von Bedrohungen zugeordnet sein können, die unterschiedliche SLs aufweisen. In diesem Falle schlägt HEAVENS vor, das höchste vorkommende SL dem Asset zuzuordnen,

Tabelle A.6: Zuordnung von TL und IL zu dem Risikolevel SL [21, Seite 68].

Security Level SL		**Schadenslevel IL**				
		0	1	2	3	4
Bedrohungslevel TL	0	QM	QM	QM	QM	Gering
	1	QM	Gering	Gering	Gering	Mittel
	2	QM	Gering	Mittel	Mittel	Hoch
	3	QM	Gering	Mittel	Hoch	Hoch
	4	Gering	Mittel	Hoch	Hoch	Kritisch

was einem Maximalwert-Ansatz entspricht. Anschließend wird das SL auf ein ASIL übertragen, um den Safety-Bezug herzustellen. Die Zuordnung der Metriken ist in Tabelle A.7 gezeigt.

Tabelle A.7: Zuordnung zwischen dem SL von HEAVENS und dem ASIL aus der ISO 26262 [99].

HEAVENS-SL	**ASIL**
QM	QM
Low	ASIL A
Medium	ASIL B
High	ASIL C
Critical	ASIL D

Die Autoren weisen daraufhin, dass die in Tabelle A.7 vorgeschlagene Zuordnung zwischen dem HEAVENS-Model und der ISO 26262 nicht verpflichtend ist. Grund hierfür sei, dass Safety und Security unterschiedliche Studienfelder sind und eine eindeutige Zuordnung somit nicht gegeben werden kann [21, Seite 70]. Elementare Unterschiede sehen die Autoren bei der Risikobewertung von Gefährdungen und Bedrohungen. So seien Safety-Metriken wie *severity*, *exposure*, *controllability* statischer als die Metriken, welche zur Bedrohungsanalyse eingesetzt werden [146, Seite 75].

A.6 Ergänzende Erläuterungen zu SAHARA

Aufbauend auf den Erläuterungen in Abschnitt 3.3.3 zeigt Abbildung A.15 das detaillierte Ablaufdiagramm von SAHARA mit den einzelnen Analyseschritten.

Als Vorbedingung für die in Abbildung A.15 gezeigten Schritte wird von einer durchgeführten HARA und Bedrohungsanalyse ausgegangen. Letztere entspricht der STRIDE-Methodik von Microsoft, um die Bedrohungen zu identifizieren. Somit bestimmt SAHARA den Security-Einfluss auf Systemebene durch Anwendung des STRIDE-Ansatzes, der von einem Security-Spezialisten angewendet wird. Die hierbei erzeugte Bedrohungsliste wird im ersten Teilschritt von SAHARA auf ihrer Safety-Relevanz hin

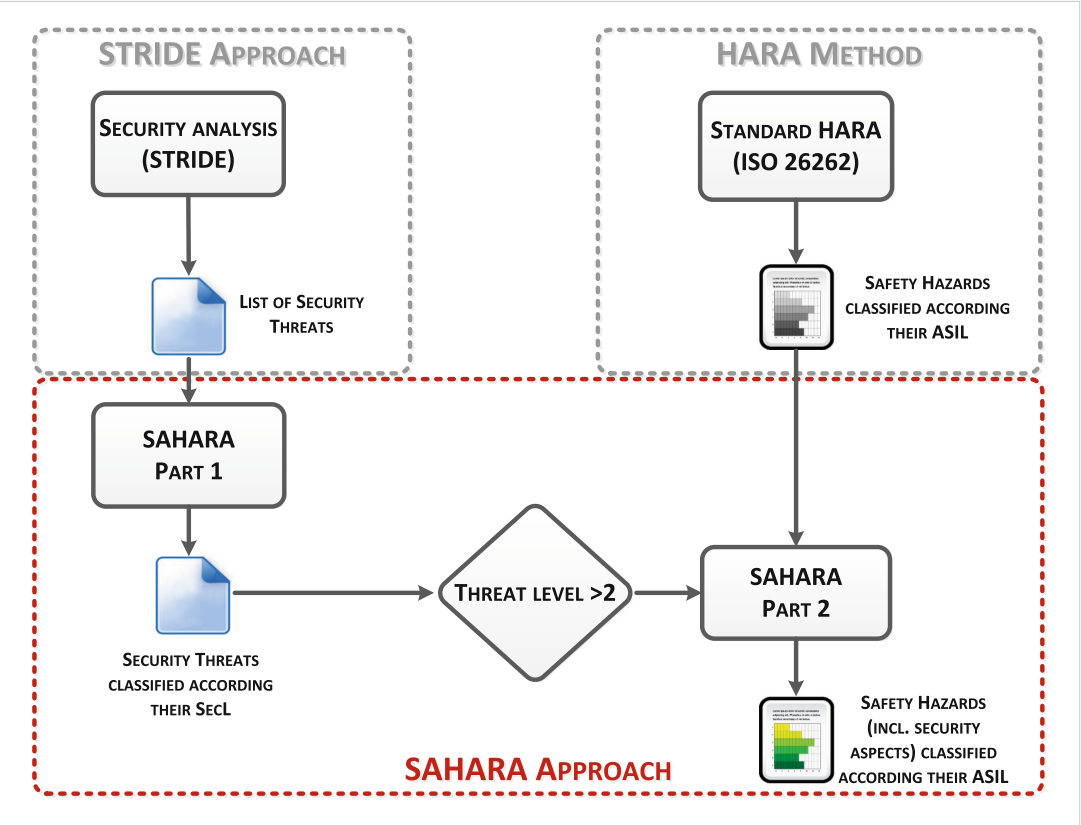


Abbildung A.15: Überblick über die SAHARA-Methode, die zeigt das identifizierte Cyber-Bedrohungen erst in der Risikobewertung auf ihre safety-relevanz überprüft werden, sofern ein Bedrohungslevel größer als 2 vergeben wird [127, 128].

bewertet. Dies geschieht in analoger Weise zur Bewertung mit den ASIL-Parametern aus der ISO 26262, was nach Ansicht der Autoren ein Vorteil der Methodik ist. Als Ergebnis wird das sogenannte Security-Level (SecL) festgelegt, das sich anhand der Gleichung (A.1) berechnen lässt.

$$\text{SecL} = \begin{cases} 0 & \text{if } T = 0 \\ > 0 & \text{if } T = 3 \\ 4 & \text{if } 5 - K - R + T \geq 7 \\ 3 & \text{if } 5 - K - R + T = 6 \\ 2 & \text{if } 5 - K - R + T = 5 \\ 1 & \text{if } 5 - K - R + T = 4 \end{cases} \tag{A.1}$$

Der erste Teilwert, der zur Berechnung von Gleichung (A.1) benötigt wird, ist die Kritikalität der Bedrohung (T). Aus Sicht der klassischen Risikobewertung entspricht dieser Wert dem Einfluss (Impact) auf den jeweiligen Vermögenswert (Asset). Bei SAHARA wird dieser anhand von Tabelle A.8 bestimmt.

Tabelle A.8: Kritikalität der Bedrohung (T), basierend auf [128].

Level	Kritikalität (T) der Bedrohung	Beispiel
0	kein Security-Einfluss	keinen Security-relevanten Einfluss
1	moderate Security-Relevanz	störende Manipulationen, teilweise eingeschränkte Verfügbarkeit des Dienstes
2	hohe Security-Relevanz	Beschädigung von Waren, Rechnungsmanipulation, Nichtverfügbarkeit von Dienstleistungen, möglicher Eingriff in die Privatsphäre
3	hohe Security und eventuell Safety-Relevanz	maximaler Security-Einfluss und lebensbedrohlicher Missbrauch sind möglich

Zur Abschätzung der Eintrittswahrscheinlichkeit einer Bedrohung verwendet SAHARA die für die Durchführung eines Angriffes notwendigen Ressourcen (R) sowie das notwendige Wissen (K). Ersteres wird anhand von Tabelle A.9 bestimmt.

Tabelle A.9: Zur Ausführung der Bedrohungen notwendigen Ressourcen (R), basierend auf [128].

Level	Notwendige Ressourcen (R)	Beispiel
0	kein zusätzliches Werkzeug oder Alltagsprodukt	Zufälliges Benutzen der Benutzeroberfläche, Sicherungsleiste, Schlüssel, Münze, Handy, etc.
1	Standardwerkzeug	Schraubenzieher, Multi-Meter, Multi-Werkzeug
2	fortschrittliche Werkzeuge	CAN-Sniffer, Oszilloskop
3	fortgeschrittene Werkzeuge	Debugger, Flash-Tools, Buskommunikationssimulatoren

Die Festlegung des notwendigen Wissens (K) ergibt sich bei Anwendung von Tabelle A.10. Ergibt sich bei Anwendung der genannten Metriken und Gleichung (A.1) ein SecL > 2, wird die Bedrohung an die Gefährdungsanalyse weitergereicht und entsprecht auf ihren Safety-Einfluss analysiert [127, 129]. Wird festgestellt, dass Hazards ausgelöst werden können, werden die safety-relevanten Bedrohungen anhand der ISO 26262-Metriken bewertet (siehe Part 2 in Abbildung A.15). Die Autoren sehen dies als ein primäres Unterscheidungsmerkmal gegenüber anderen Methoden, da Bedrohungen identifiziert werden, die zu einer Verletzung der Safety-Ziele führen können und im Safety-Konzept betrachtet werden müssen.

Tabelle A.10: Zur Ausführung der Bedrohungen notwendiges Wissen (K), basierend auf [128].

Level	Notwendige Ressourcen	Beispiel
0	keine Vorkenntnisse (Black-Box-Ansatz)	durchschnittlicher Fahrer, unbekannte Komponenten
1	technisches Vorwissen (Gray-Box-Ansatz)	Techniker, Grundkenntnisse der Komponenten
2	Domänenwissen (White-Box-Ansatz)	Person mit technischer Ausbildung und Interessenschwerpunkten, Komponenten bekannt

B

Unterlagen der Evaluierung

In diesem Abschnitt sind die Unterlagen und Auswertungen zur Teilnehmerbefragung von Kapitel 5 gezeigt. Außerdem ist mit B.2.4 ein weiterer Hypothesentest präsentiert, der unabhängig von der Stichprobenverteilung ist und den Hypothesentest aus Abschnitt 5.3 ergänzt.

B.1 Ergebnisse der Teilnehmer Befragung

In diesem Unterabschnitt wird die Selbsteinschätzung der Teilnehmer zu den Disziplinen: Automotive Software Engineering, Funktionale Sicherheit, Automotive Bussysteme, Echtzeitsysteme, Sicherheit, Gefahren- und Gefahrenanalyse für die Teams *Security*, *Safety* und *Unerfahren* aufgezeigt und bewertet. und bewertet. Hierzu stellen die grauen Balken in den Whisker-Plots die Ergebnisse der Selbsteinschätzung dar und die schwarzen Balken zeigen die Ergebnisse der Befragung. Letztere sind in Anhang B.2.4 aufgeführt. Der Vergleich zwischen Selbseinschätzung und Befragung erlaubt den Grad der Erfahrung der Teilnehmer festzustellen. Zur Messung der Teilnehmererfahrung ist die bewährte Skala von Eraut [61] verwendet.

B.1.1 Team Security

Die Umfrageergebnisse in Abbildung B.1 zeigen, dass Team *Security* ein gutes Verständnis im Bereich des Software-Engineerings besitzt. Dieses Ergebnis ist nicht überraschend, da diese Kategorien zu den üblichen Arbeitsbereichen der Sicherheitsingenieure gehört. So können 8 der 9 Teilnehmer als kompetent bewertet werden, 5 der Befragten zeigen sogar ein Wissenslevel auf Expertenniveau. Interessant hierbei ist, dass sich die Teilnehmer schlechter eingeschätzt haben, als die Befragung herausfand. So zeigt Abbildung B.1, dass sich kein Teilnehmer als Experte sieht.

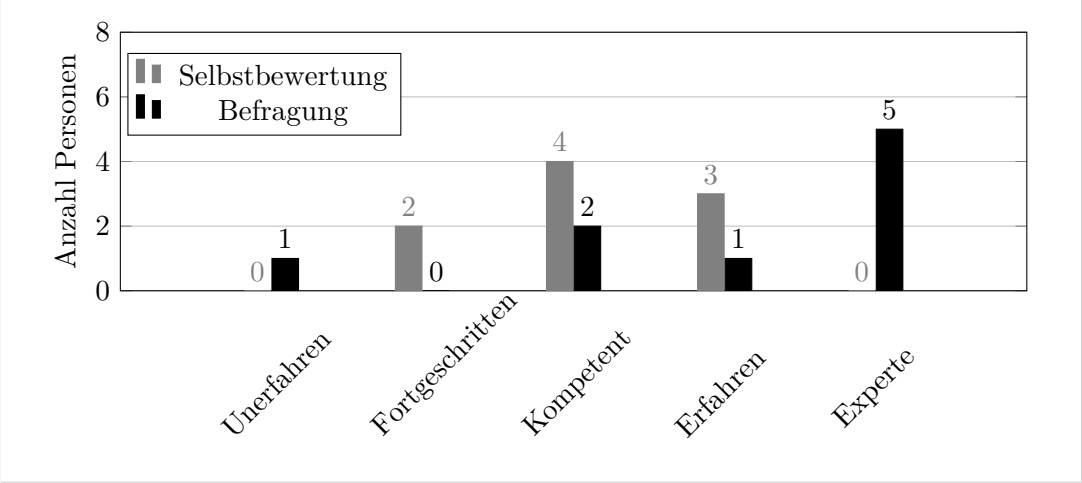


Abbildung B.1: Ergebnisse der Befragung für die Erfahrungen im Bereich Software-Engineering für das Team *Security*.

Im Gegensatz dazu sind die Ergebnisse für das Wissen um die funktionale Sicherheit (Abbildung B.2) für Security-Ingenieure unerwartet hoch. So konnten 7 der 9 Teilnehmer bei der Befragung das Experten-Level erreichen. Dies lässt sich zum Teil dadurch erklären, dass die Security-Ingenieure aufgrund der Geschäftsfelder ihres Arbeitgebers täglich mit Themen der funktionalen Sicherheit in Kontakt stehen.

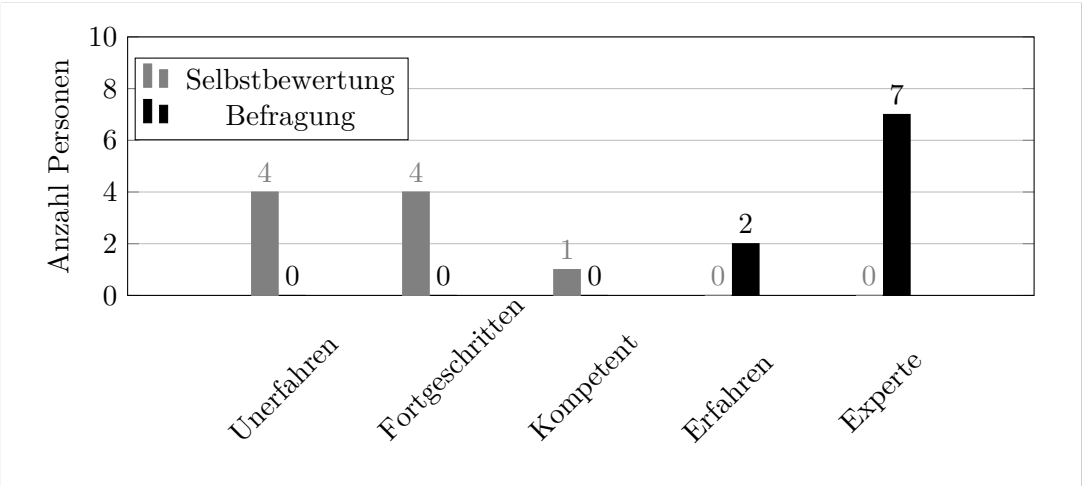


Abbildung B.2: Ergebnisse der Befragung für die Erfahrungen im Bereich funktionale Safety für das Team *Security*.

Hinsichtlich der Ergebnisse für den Wissensstand in automobilen Bussystemen für Team *Security* zeigt Abbildung B.3, das eine überdurchschnittliche Kenntnis in der Gruppe besteht. So lassen sich 4 Teilnehmer als Kompetent bewerten, 3 weitere als mindestens Erfahren.

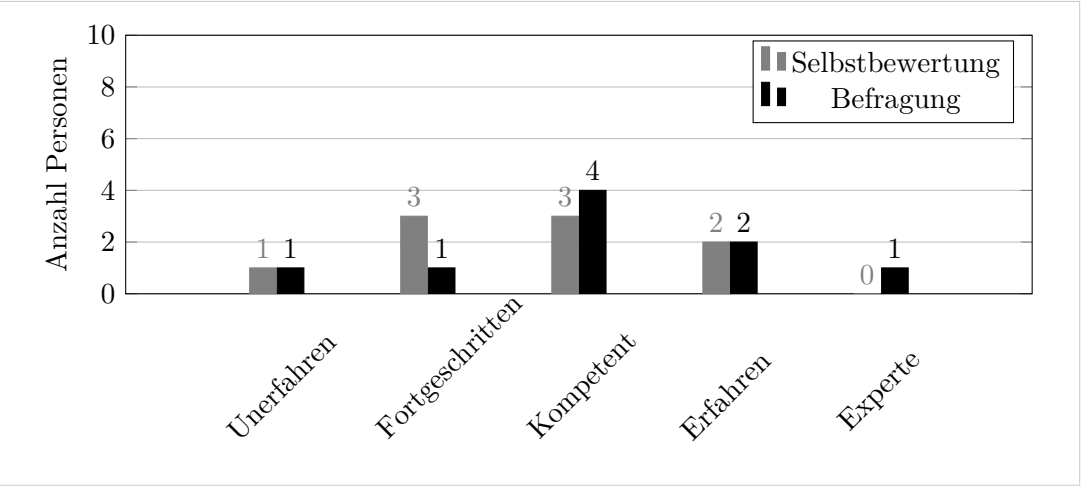


Abbildung B.3: Ergebnisse der Befragung für die Erfahrungen im Bereich automotive Bussysteme für das Team *Security*.

Für die Erfahrungen mit Echtzeitsystemen zeigt Abbildung B.4, dass die Hälfte der Gruppe *Security* in die Klasse *Unerfahren* eingeordnet werden kann und die andere Hälfte mindestens als Profis zu bewerten sind.

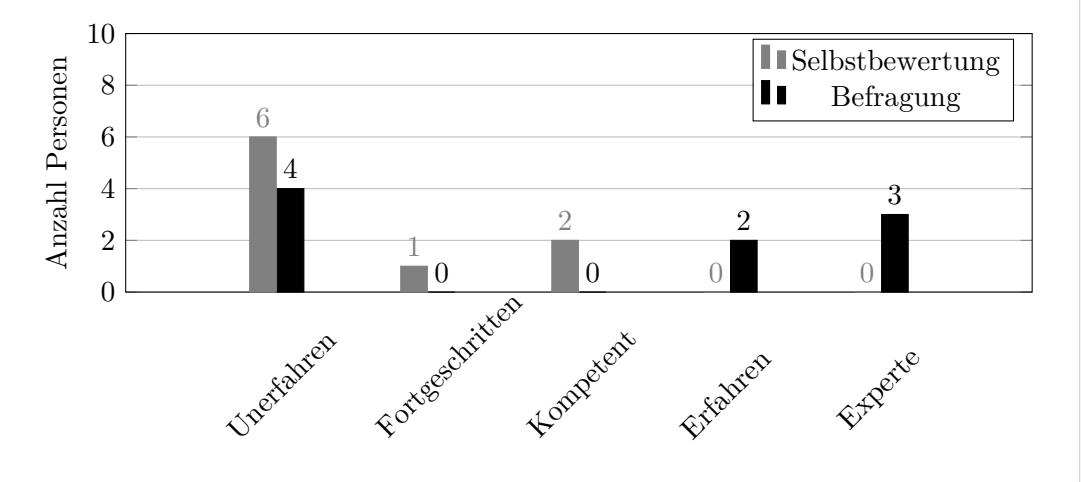


Abbildung B.4: Ergebnisse der Befragung für die Erfahrungen im Bereich Echtzeitsysteme für das Team *Security*.

Darüber hinaus zeigen die in Abbildung B.5 erreichten Werte, dass Team *Security* in der Mehrheit nur geringes Wissen im Bereich der Gefährdungsanalyse besitzt. Interessant ist, dass 5 Teilnehmer hingegen angaben, keine Erfahrungen in diesem Gebiet zu besitzen, allerdings in der Lage waren, unsere Fragen zur Gefährdungsanalyse richtig zu beantworten (Erfahren und Experte).

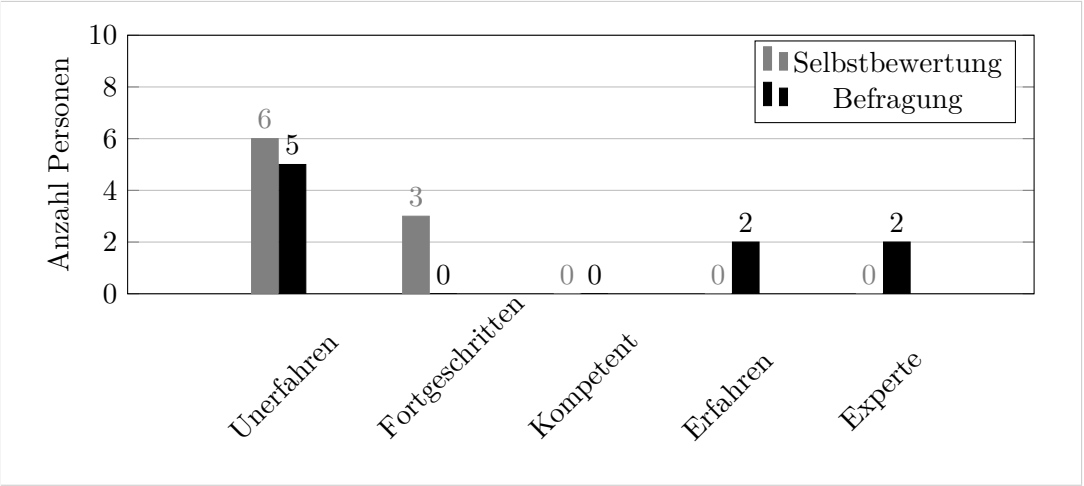


Abbildung B.5: Ergebnisse der Befragung für die Erfahrungen mit Gefährdungsanalysen für das Team *Security*.

Im Gegensatz dazu zeigt Abbildung B.6 ein eindeutiges Bild für die Erfahrungen mit Bedrohungsanalysen. Obwohl die Teilnehmer in der Selbsteinschätzung angaben, dass sie selbst keine oder nur wenig Erfahrung mit Bedrohungsanalyse besitzen, sind 4 Teilnehmer als Erfahren und weitere 4 als Experten zu bewerten. Somit können fast alle Teilnehmer als Sicherheitsexperten gesehen werden. Da die Selbsteinschätzung ein – für ein Security-Team – ungewöhnliches Ergebnis aufzeigt, wurde ein Dialog mit den Security-Ingenieuren geführt. Hier stellte sich heraus, dass alle Teilnehmer bereits mehrfach Bedrohungsanalysen durchgeführt hatten, ihre pessimistische Selbsteinschätzung allerdings mit fehlenden Messkriterien für die Qualität der Analyseergebnisse zu begründen ist.

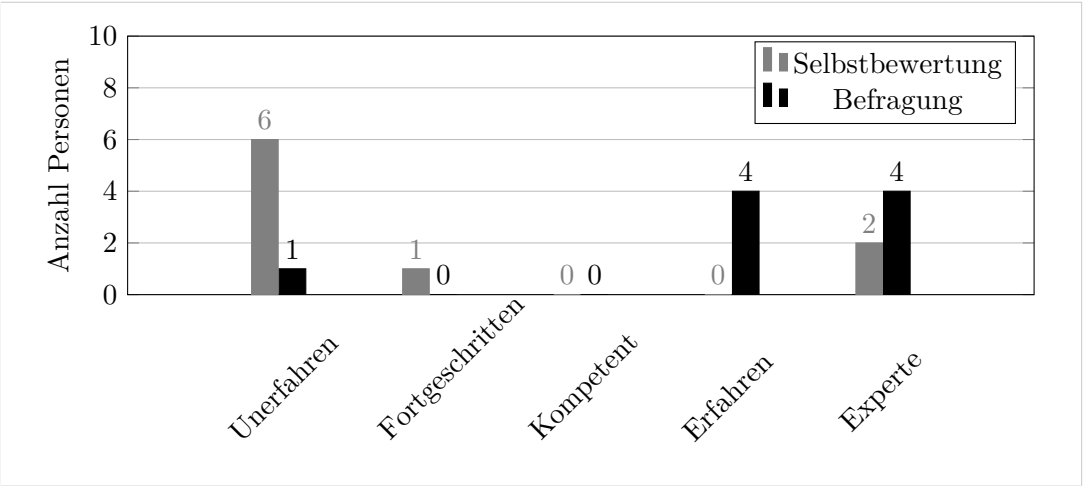


Abbildung B.6: Ergebnisse der Befragung für die Erfahrungen mit Bedrohungsanalysen für das Team *Security*.

Die Annahme, dass die Teilnehmer von Team *Security* als Sicherheitsexperten benannt werden können, wird durch die Messergebnisse in Abbildung B.7 weiter gestützt. Hier zeigt sich, dass mehr als die Hälfte des Teams Erfahrung im Bereich Security besitzt.

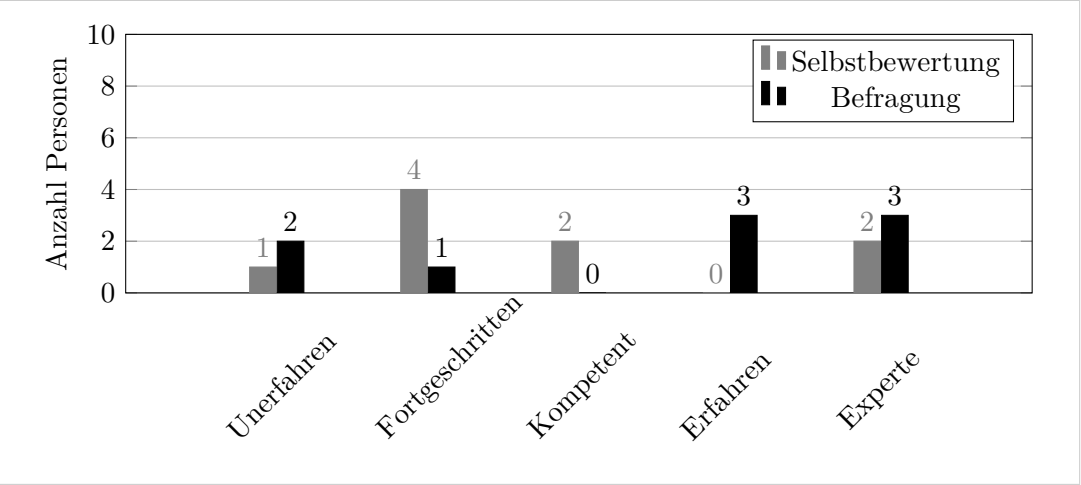


Abbildung B.7: Ergebnisse der Befragung für die Erfahrungen in der Security-Domäne für das Team *Security*.

Zusammenfassend lässt sich sagen, dass aus der Selbsteinschätzung und der Befragung abgeleitet werden kann, dass das Team *Security* über einen guten technischen Hintergrund in den Bereichen Software-Engineering, funktionale Sicherheit, Automotive-Bussysteme sowie Bedrohungsanalyse besitzt, allerdings weniger Wissen im Bereich der Gefährdungsanalyse vorweisen kann.

B.1.2 Team Safety

Für die Teilnehmer des Team *Safety* wurde ein hoher Kenntnisstand im Bereich Software-Engineering (Abbildung B.8) und der funktionalen Safety (Abbildung B.9) festgestellt. Dies ist kein überraschendes Ergebnis, handelt es sich doch um typische Arbeitsbereiche der Safety-Ingenieure. So konnten insbesondere für die Kategorie Software-Engineering 11 Experten ermittelt werden, was 79% des gesamten Teams darstellt. Ähnlich wurde für die funktionale Sicherheit 6 Gruppenmitglieder als Erfahren und 7 als Experten bestimmt (Abbildung B.9).

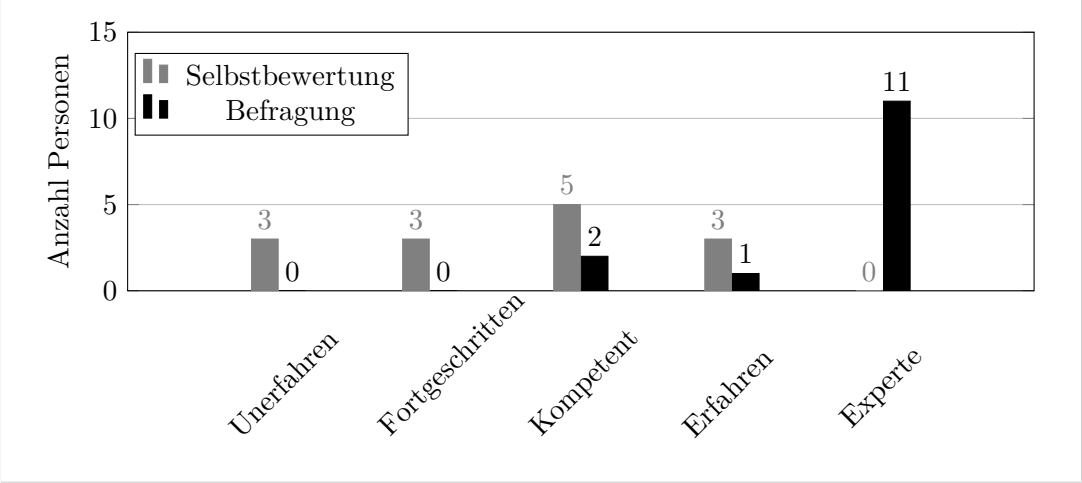


Abbildung B.8: Ergebnisse der Befragung für Erfahrungen im Software-Engineering für das Team *Safety*.

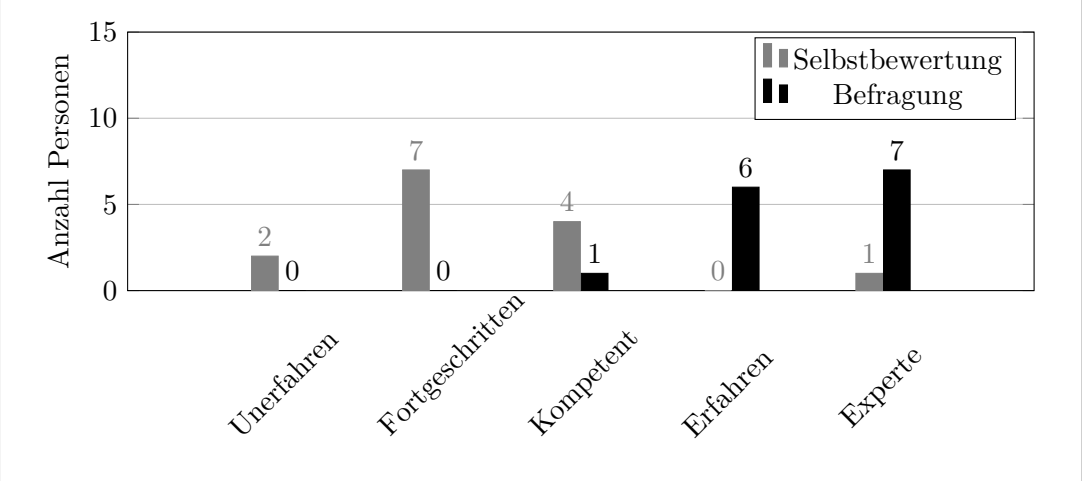


Abbildung B.9: Ergebnisse der Befragung für Erfahrungen im Bereich funktionaler Safety für das Team *Safety*.

Für die Erfahrung mit automobilen Bussystemen zeigt Abbildung B.10 einen Kenntnisstand, der vom fortgeschrittenen Anfänger bis zum Experten reicht. Eine Häufung mit 6 Teilnehmern zeigt sich bei der Klasse *Kompetent*. Daher werden die Teilnehmer als vertraut mit den Grundlagen automobiler Bussysteme sowie dem CAN angenommen.

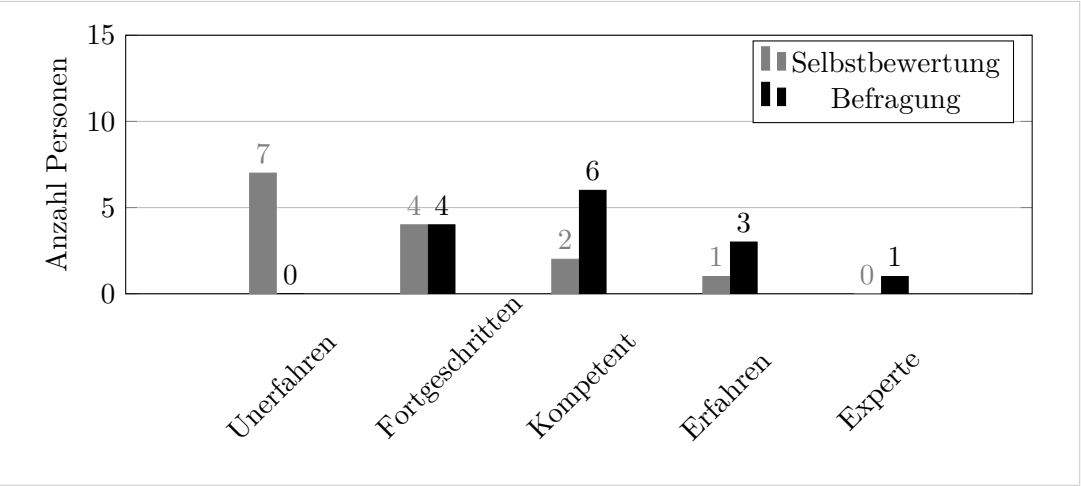


Abbildung B.10: Ergebnisse der Befragung für Erfahrungen mit automotiven Bussystemen für das Team *Safety*.

Für die Erfahrung mit Echtzeitsystemen zeigen die Ergebnisse, dass fast kein Teilnehmer glaubt, Kenntnisse in diesem Bereich zu haben. Dies ist ein interessantes Ergebnis, da die gemessenen Wissensstände in Abbildung B.11 einen durchschnittlichen bis hohen Erfahrungsstand in Echtzeitsystemen aufzeigen. Insbesondere erreicht mehr als die Hälfte des Teams das Niveau kompetent oder Experte. Daher kann angenommen werden, dass Team *Safety* in der Lage ist, typische Probleme zeitkritischer Funktionen zu verstehen, wie es beispielsweise verspätete Meldungen sein können.

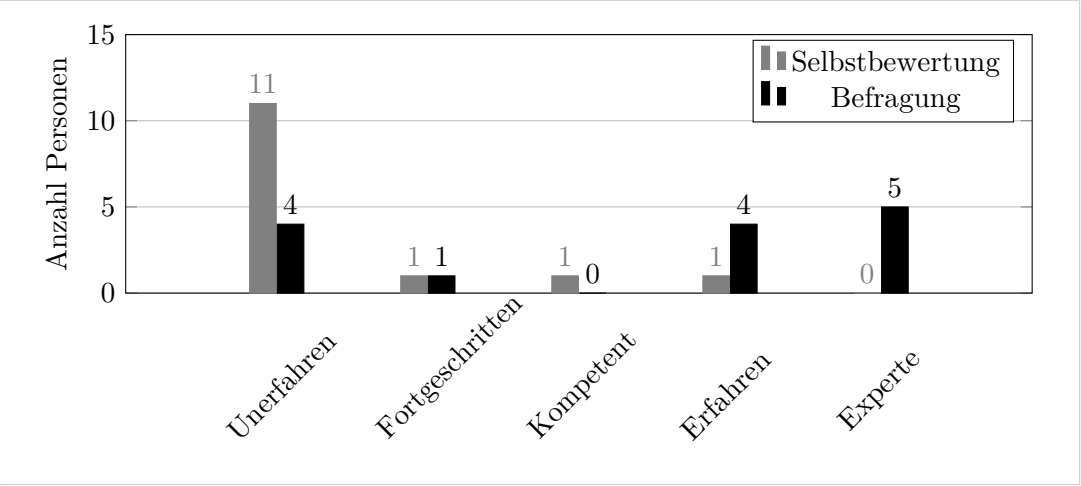


Abbildung B.11: Ergebnisse der Befragung für Erfahrungen mit Echtzeitsystemen für das Team *Safety*.

Wie erwartet, erreichte die Team *Safety* einen hohen Wissensstand für den Bereich der Gefährdungsanalyse. So zeigt Abbildung B.12, dass 3 Mitglieder der Gruppe Erfahren und 8 Experten sind, was zusammen 79 % des gesamten Teams darstellt. Hiermit kann gesagt werden, dass die Mitglieder keine Probleme bei der Durchführung einer

Gefährdungsanalyse haben werden. Gleichwohl dieses Ergebnis zu erwarten war, lässt dies die Annahme zu, dass das Design der Befragung valide ist.

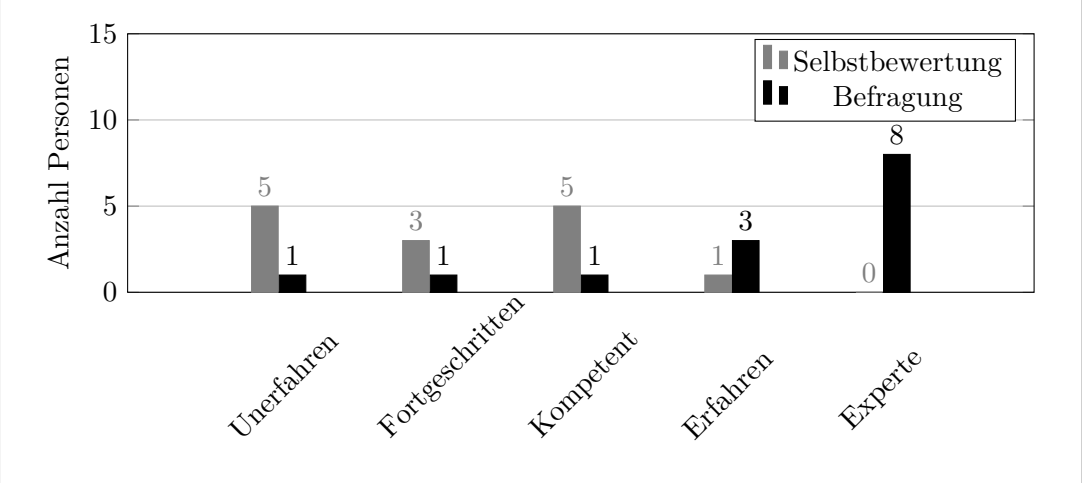


Abbildung B.12: Ergebnisse der Befragung für Erfahrungen mit Gefährdungsanalysen für das Team *Safety*.

Die wichtigsten Ergebnisse für die Evaluierung der SGM zeigen Abbildung B.13 mit dem Kenntnisstand hinsichtlich Security und Abbildung B.14 mit der Bewertung der Erfahrungen von Bedrohungsanalysen. So zeigen die Messergebnisse in Abbildung B.13, das 10 Gruppenmitgleiter als unerfahren im Security-Bereich zu bewerten sind. Außerdem kann kein Mitglied als Erfahren bezeichnet werden.

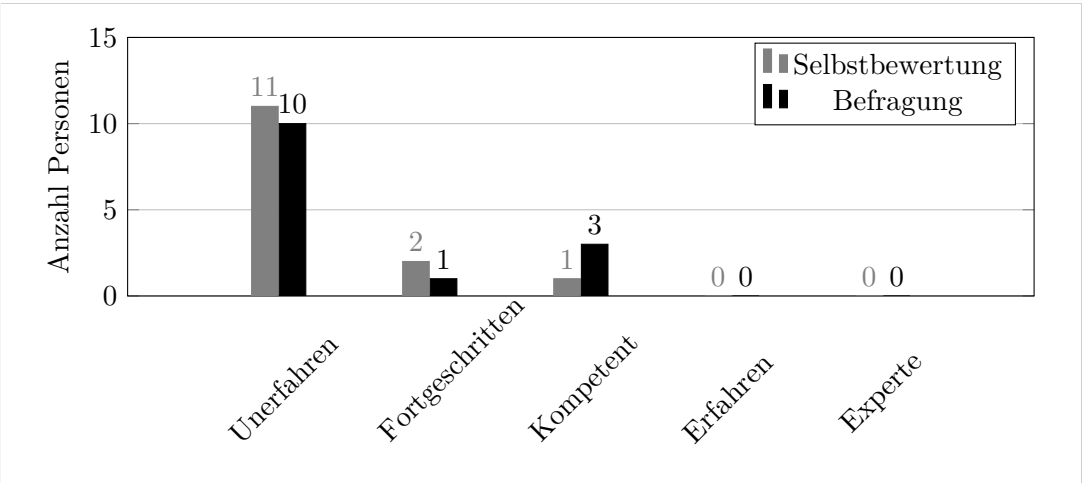


Abbildung B.13: Ergebnisse der Befragung für Erfahrungen im Security-Bereich für das Team *Safety*.

Als ungewöhnlich können die Ergebnisse in Abbildung B.14 angesehen werden. Hier zeigt das Team *Safety* mit 6 Erfahrenen und 5 Experten eine starke Kompetenz bei Bedrohungsanalysen. Aufgrund dessen wurden diese Teilnehmer gesondert befragt, ob sie jemals eine Bedrohungsanalyse durchgeführt hatten. Die Antwort hierzu war, dass niemand aus dem Safety-Team je eine Bedrohungsanalyse durchgeführt hatte. Eine

Begründung für diesen Umstand könnte sein, dass das Vorgehen für die Durchführung von Gefahren- und Bedrohungsanalysen Analogien aufweisen. Insbesondere, wenn es um das Identifizieren von Missbrauchsfällen und Designabweichungen geht.

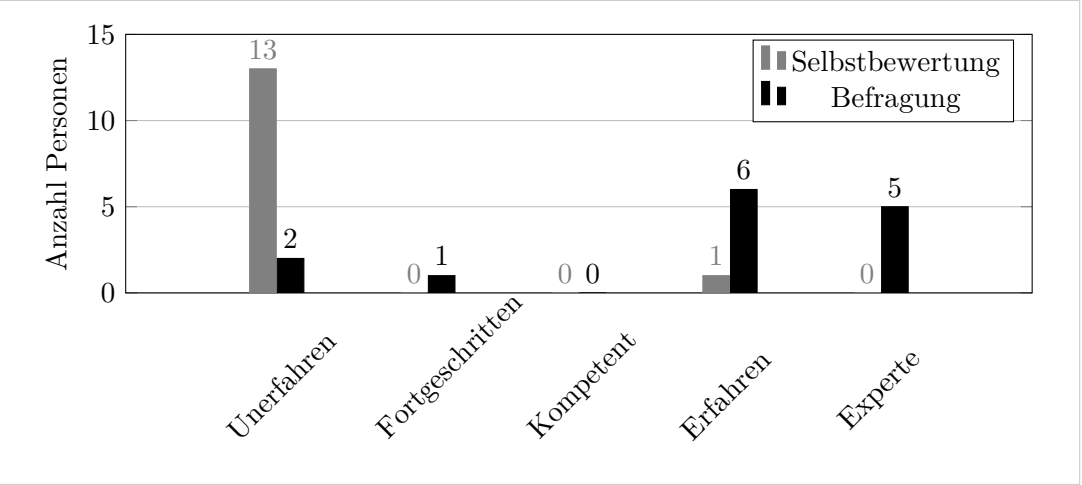


Abbildung B.14: Ergebnisse der Befragung für Erfahrungen mit Bedrohungsanalysen für das Team *Safety*.

Zusammenfassend lässt sich aus der Selbsteinschätzung und der Auswertung des Fragebogens ableiten, dass im Durchschnitt alle Teilnehmer von Team *Safety* einen guten technischen Hintergrund in automobilen Bussystemen, gute Kenntnisse im Bereich der funktionalen Sicherheit und nur wenige Kenntnisse im Security-Bereich und der Bedrohungsanalyse besitzen.

B.1.3 Team Unerfahren

Auch wenn die Teilnehmer von Team *Unerfahren* nicht in den Bereichen Software-Engineering und funktionale Safety gearbeitet haben, zeigen Abbildung B.15 und Abbildung B.16, dass die Teilnehmer ein Verständnis in diesen Bereichen besitzen. Dies wird zum Teil durch ihre Studienrichtungen erklärt. Außerdem haben vier der Teilnehmer ihren Master-Abschluss in Fahrzeugtechnik erworben, was die gesteigerte Anzahl von Experten in Abbildung B.15 und Abbildung B.16 erklärt.

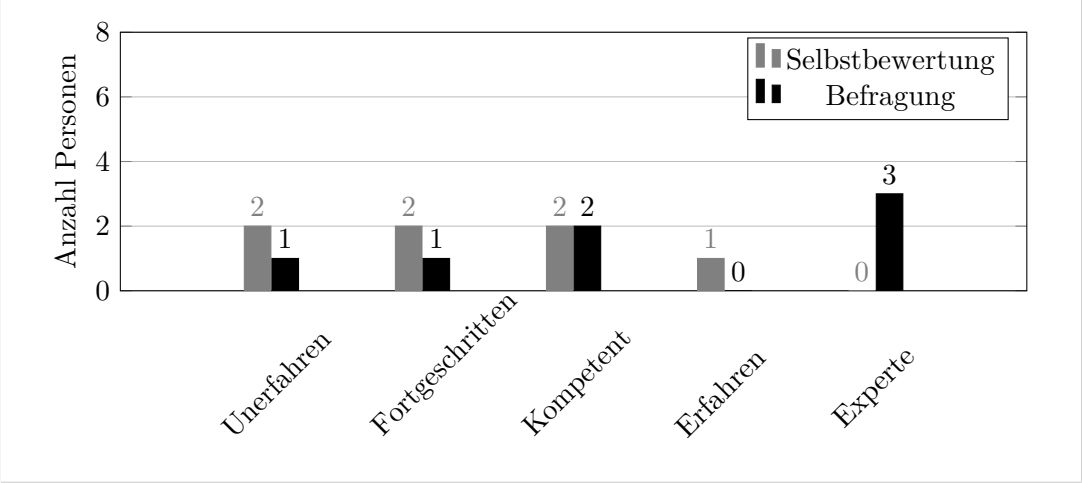


Abbildung B.15: Ergebnisse der Befragung für die Erfahrungen im Bereich Software-Engineering für das Team *Unerfahren*.

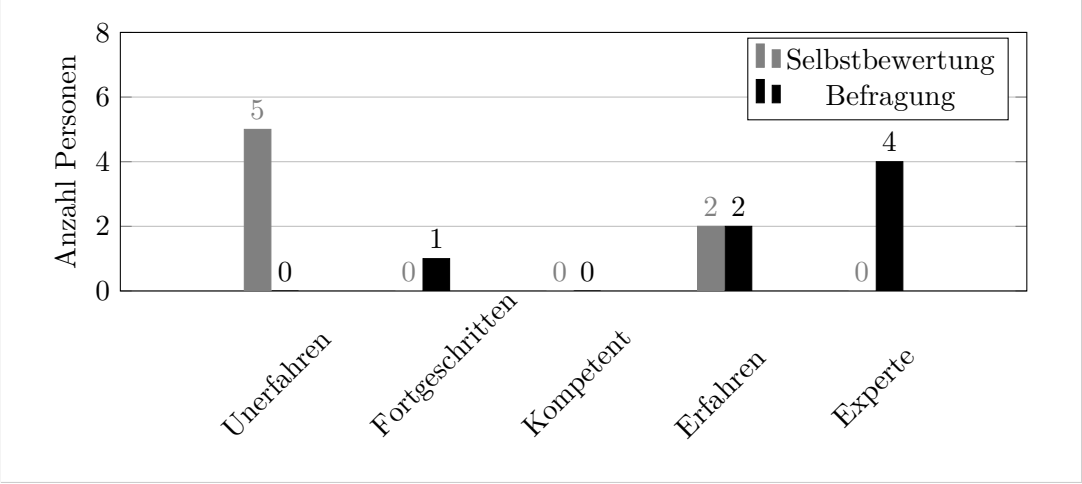


Abbildung B.16: Ergebnisse der Befragung für die Erfahrungen im Bereich funktionale Safety für das Team *Unerfahren*.

Hinsichtlich deren Wissens um automobile Bussysteme wie beispielsweise dem CAN-Bus (Abbildung B.17), liegt die Erfahrung des Teams *Unerfahren* in der Bandbreite von kompetent bis Experte. Aufgrund dessen kann angenommen werden, dass die Teilnehmer mit dem CAN-Bus vertraut sind. Dies ist daher von Bedeutung, da das Kontextdiagramm (Abbildung 5.4) mehrere CAN-Buse aufweist.

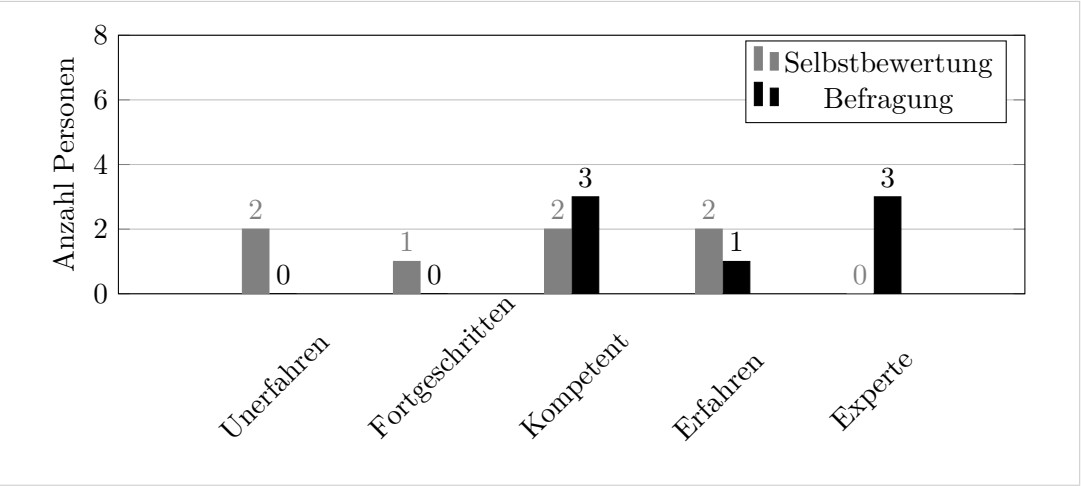


Abbildung B.17: Ergebnisse der Befragung der Erfahrungen mit automotiven Bussystemen für das Team *Unerfahren*.

Für die Erfahrung mit Echtzeitsystemen zeigen die Ergebnisse, dass mehr als die Hälfte des Teams als Experten bezeichnet werden kann (Abbildung B.18). Daher kann davon ausgegangen werden, dass die Teilnehmer von Team *Unerfahren* in der Lage sind, die Auswirkungen von Manipulationen zeitkritischer Funktionen auf das Gesamtsystem zu verstehen.

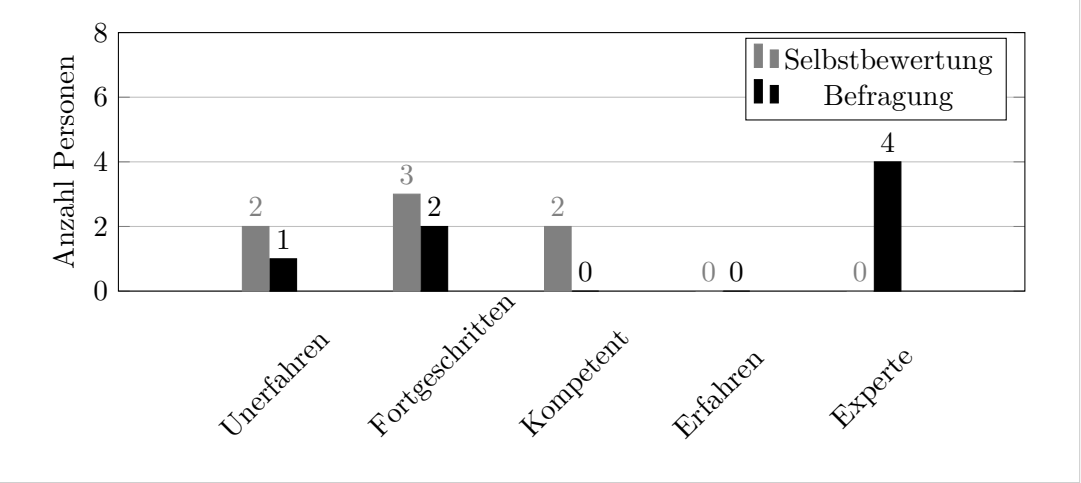


Abbildung B.18: Ergebnisse der Befragung für die Erfahrungen mit Echtzeitsysteme für das Team *Unerfahren*.

Ein interessantes Ergebnis, das bei der Gruppe *Unerfahren* gemessen wurde, ist die Erfahrung mit Gefährdungsanalysen in Abbildung B.19. Die Teilnehmer glaubten nicht, das sie Erfahrungen auf diesem Gebiet besitzen, die Messergebnisse zeigen allerdings, dass fast alle von ihnen in der Lage waren, die Fragen richtig zu beantworten. Daher kann angenommen werden, dass diese Teilnehmer in der Lage sind, eine Gefährdungsanalyse durchzuführen. Außerdem kann davon ausgegangen werden, dass das Team *Unerfahren* mit Nachwuchsingenieuren im Bereich Safety vergleichbar ist.

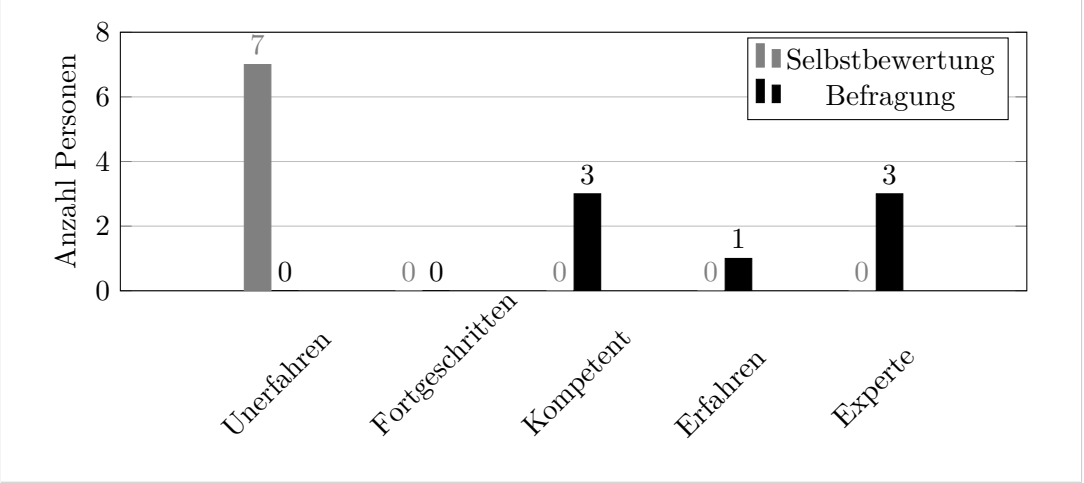


Abbildung B.19: Ergebnisse der Befragung für die Erfahrungen mit Gefährdungsanalysen für das Team *Unerfahren*.

Ein ähnliches Bild zeigt Abbildung B.20 mit den Ergebnissen für die Messung der Erfahrungen mit Bedrohungsanalysen. Die Gruppe glaubte, dass sie selbst keine oder weniger Erfahrung mit Bedrohungsanalysen besitzen. Die ausgewerteten Antworten zeigen jedoch, dass 6 Teilnehmer von Team *Unerfahren* als Experten hinsichtlich Bedrohungsanalysen eingestuft werden können. Hiermit können wir annehmen, dass die Gruppe Kompetenzen im Bereich der Bedrohungsanalysen besitzt.

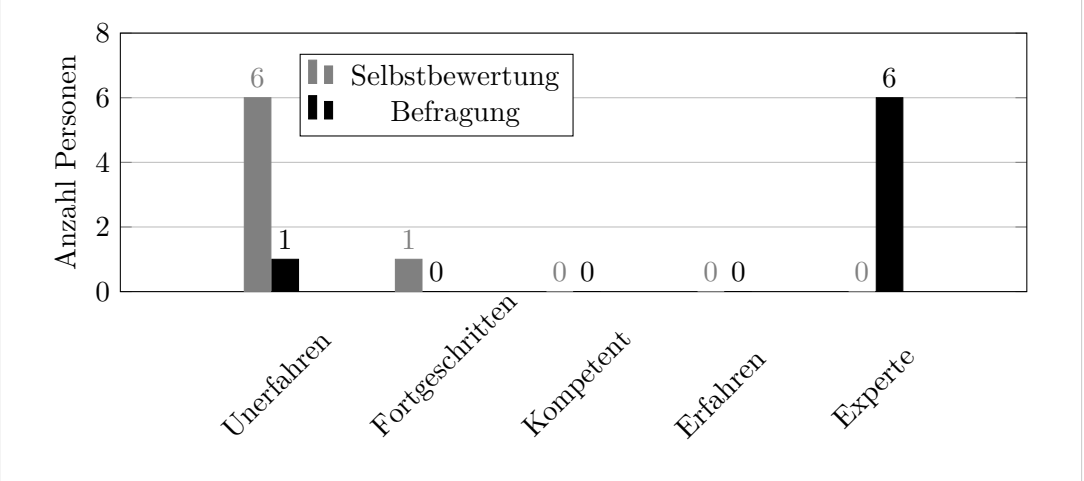


Abbildung B.20: Ergebnisse der Befragung für die Erfahrungen mit Bedrohungsanalysen für das Team *Unerfahren*.

Mit Abbildung B.21 zeigt sich allerdings, dass Team *Unerfahren* aus Teilnehmern besteht, die grundsätzlich wenige Kenntnisse im Security-Bereich besitzen. Daher kann angenommen werden das die Gruppe alleinig Erfahrung mit Bedrohungsanalysen besitzt, jedoch weniger Erfahrung in der übergeordneten Security-Domäne aufweist.

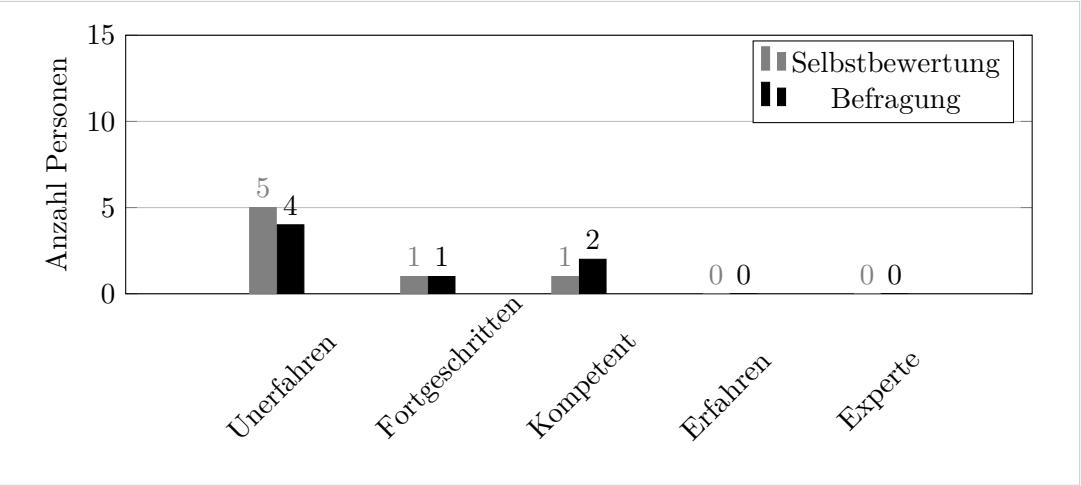


Abbildung B.21: Ergebnisse der Befragung für Erfahrungen im Security-Bereich für das Team *Unerfahren*.

Zusammenfassend lässt sich sagen, dass sich aus der Selbsteinschätzung und der Auswertung der Fragen zeigt, dass im Durchschnitt alle Mitglieder der Gruppe *Unerfahren* über einen guten technischen Hintergrund bezüglich automobilen Bussystemen, gute Kenntnisse im Bereich der funktionalen Sicherheit und durchschnittliche Kenntnisse im Security-Bereich aufweisen.

B.2 Ergebnisse des Experimentes

In diesem Abschnitt werden die gemessenen Ergebnisse für FI, TP und FP für Team *Security*, *Safety* und Team *Unerfahren* vorgestellt. Die erzielten Werte sind für jede Gruppe in drei Whisker-Plots zusammengefasst und erläutert (Abbildungen B.22 und B.24 und Anhang B.2.2). Darüber hinaus ist für jeden Teilnehmer aus Team *Security* und *Safety* das Flow-Erleben gemessen und in den Tabellen B.2, B.5 und B.6 aufgezeigt. Für Team *Unerfahren* wurde das Flow-Erleben nicht gemessen, da ausschließlich die Mitarbeiter des befragten Zulieferers nach ihrem Workflow während des Experimentes befragt wurden. Diese Entscheidung ist damit zu begründen, das letztere täglich Fahrzeugsysteme – bezogen auf ihr Studienfeld – analysieren und in ihrer grundsätzlichen Denk- und Arbeitsweise vergleichbar sind.

B.2.1 Ergebnisse für Team Security

Anhand der Auswertung der Ergebnisse von Team *Security* ergibt sich der Whisker-Plot [32, Seiten 106–108] in Abbildung B.22. Die Ergebnisse sind dabei bezogen auf die vom Teilnehmer ausgefüllten Zeilen (FI), nicht korrekt beschriebenen Schutzzielen (FP) sowie den als korrekt bewerteten Schutzzielen (TP).

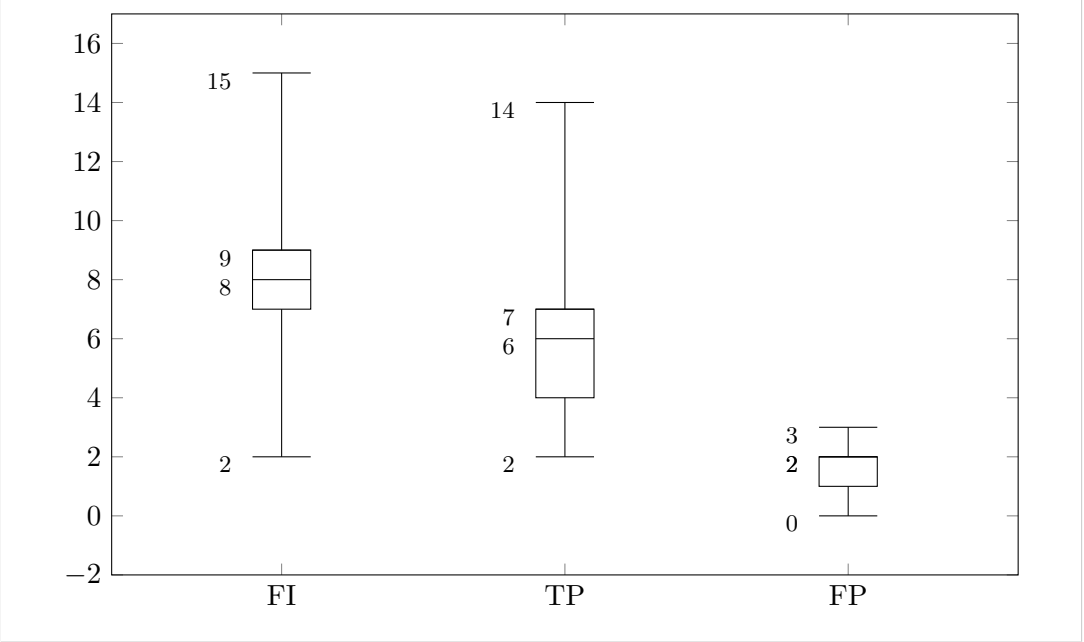


Abbildung B.22: Whisker-Plot von Team *Security* für die gemessene Anzahl der ausgefüllten Zeilen (FI), der korrekt identifizierten Bedrohungen (TP) sowie der als falsch bewerteten Bedrohungen (FP). Zur Identifikation der Bedrohungen, setzte Team *Security* nicht die SGM ein.

Wie Abbildung B.22 aufzeigt, identifizierte Team *Security* im Maximum eine Anzahl von 15 Bedrohungen (FI) und im Minimum eine Anzahl von 2. Der Median für identifizierte Bedrohungen liegt bei 8. Von diesen waren maximal 14 Bedrohungen korrekt (TP), wobei im schlechtesten Fall nur 2 Bedrohungen als korrekt bewertet wurden. Der Median für das Team lag bei 6 richtigen Bedrohungen. Hinsichtlich der Ergebnisse für die Präzision, Effizienz, Sensitivität und Produktivität zeigt Tabelle B.1 die erreichten Werte, für jeden Teilnehmer aus dem Security-Team.

Tabelle B.1: Die im Experiment für jeden Teilnehmer aus Team *Security* erreichten Ergebnisse hinsichtlich der Präzision (PPV), Effizienz, identifizierte Bedrohungen (FI) und Produktivität (PRO).

Teilnehmer	1	2	3	4	5	6	7	8	9	∅	σ
PPV	1	0,89	0,63	0,57	0,75	0,75	0,778	0,93	0,75	78,29%	0,13
Effizienz	8	36	32	28	16	32	36	60	32	31,11	13,57
PRO	0,13	0,6	0,53	0,47	0,27	0,53	0,6	1	0,53	51,85%	0,52

Aus Tabelle B.1 lässt sich ableiten, dass Team *Security* im Durchschnitt einen Wert von 78,29 % für die Präzision bei einer Standardabweichung von $\sigma = 0{,}13$ erreicht hat. Für die Effizienz wurden im Mittel 31,11 identifizierte Bedrohungen pro Stunde mit einer Standardabweichung von $\sigma = 13{,}57$ gemessen. Für die FI wurden im Mittel 7,778 identifizierte Bedrohungen mit einer Standardabweichung von $\sigma = 0{,}22$ gemessen. Für

die Produktivität ergab sich im Durchschnitt schließlich ein Wert von 51,85 % sowie eine Standardabweichung von $\sigma = 0{,}5$.

Neben der Messung von FI, FP und TP wurden drei weitere Werte erfasst, die von Rheinberg et. al. [164] vorgeschlagen sind, um das Flow-Erleben zu messen. Diese Skala hat sich als sinnvoll erwiesen, wenn der Anreiz zur Durchführung einer Tätigkeit gemessen werden möchte [59, 163, 178]. Konkret sind das der *Flow*, der widerspiegelt, wie enthusiastisch ein Teilnehmer bei der Durchführung seiner Aufgabe ist, die *Bedenken* die der Teilnehmer über die Aufgabe hat und schließlich wie stark sich ein Teilnehmer durch die Aufgabe gefordert fühlt (*Herausforderung*). Die Erfassung dieser Werte erfolgte parallel zum Experiment und wurde mit Teil 2 des Fragebogens festgehalten (Anhang B.2.4). Auf diese Weise konnte bestimmt werden, wie gestresst sich die Teilnehmer während des Experiments fühlten. Hieraus lässt sich ableiten wie viel Aufwand die SGM erfordert. Die Ergebnisse für jeden Teilnehmer des Teams *Security* sind in Tabelle B.2 zusammengefasst.

Tabelle B.2: Rückmeldung der Teilnehmer des Teams *Security* auf das Flow-Erleben nach Rheinberg et. al. [164]. Gemessen wurde, wie stark sich der Teilnehmer im Flow befand, wie groß seine Bedenken waren und wie stark er sich herausgefordert fühlte.

Teilnehmer	1	2	3	4	5	6	7	8	9	∅
Flow	1,7	6	4,4	4,1	2,2	1,7	2,8	5,7	6	3,8
Bedenken	1	3,7	2,7	2,7	4,7	1	3	2,3	3	2,7
Herausforderung	1	5	5	5	4	4	4	4	4	4

Bezogen auf die von Team *Security* gemeldeten Werte, liegen die Teilnehmer beim *Flow* und bei der *Herausforderung* in der Nähe des Mittelwertes des Intervalls $[1, 7]$. Damit befanden sich die Teilnehmer in einem durchschnittlichen Arbeitsablauf und fühlten sich durchschnittlich gefordert. Hinsichtlich der *Bedenken* ist die Abweichung zum Mittelwert mit 0,8 größer und bedeutet, dass die Teilnehmer weniger Bedenken bei der Durchführung der Aufgabe hatten.

B.2.2 Ergebnisse für Team Safety

Als zweite Teilnehmergruppe ist Team *Safety* ausgewertet. Die Ergebnisse hierzu sind in Anhang B.2.2 dargestellt. Sie zeigen, ähnlich zu den Ergebnissen in Abbildung B.24 des Teams *Unerfahren* eine geringe Streuung im Whisker-Plot für die Werte FI, TP und FP, was sich mit der erhöhten Unterstützung durch die SGM-Vorlage erklären lässt.

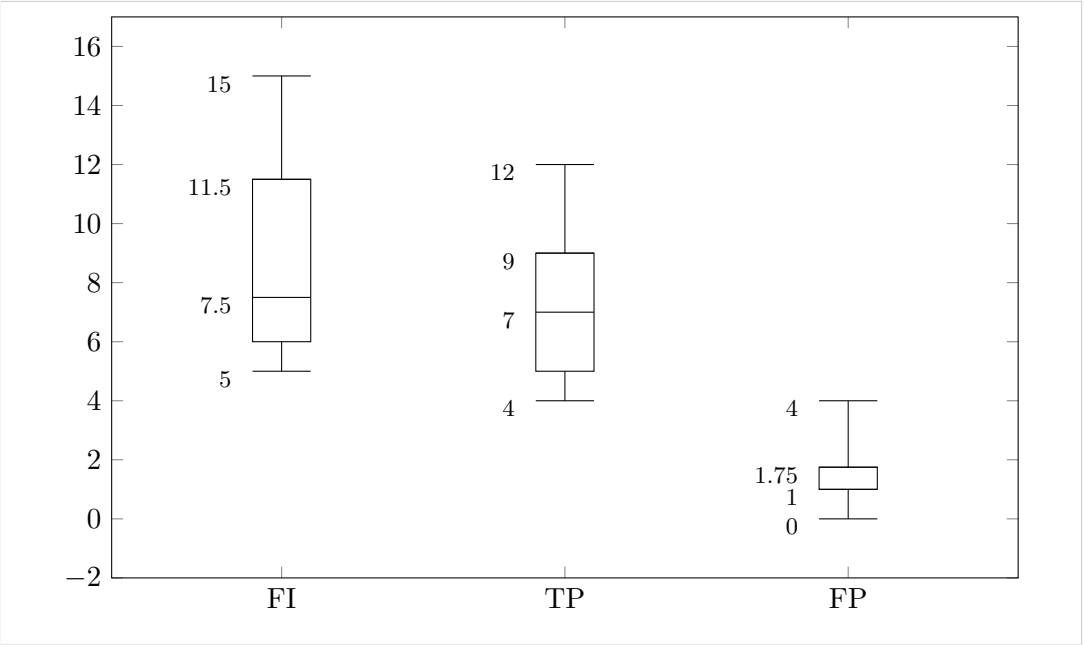


Abbildung B.23: Whisker-Plot von Team *Safety* für die gemessene Anzahl der ausgefüllten Zeilen (FI), der korrekt identifizierten Bedrohungen (TP) sowie der als falsch bewerteten Bedrohungen (FP). Team *Safety* setzte die SGM zur Bedrohungsidentifikation ein.

Für die Anzahl identifizierter Bedrohungen (FI) erreichte das Team im Maximum 15, im Minimum 5 und im Median 7,5 Bedrohungen. Das Team erreicht damit die gleiche maximale Anzahl von Bedrohungen wie das Team *Security*, die nicht die SGM einsetzten. Im Minimum wurden außerdem 5 Bedrohungen identifiziert, was im Vergleich mit Team *Security* eine Verbesserung um 3 Bedrohungen bedeutet. Die davon als korrekt befunden Bedrohungen (TP) sind allerdings mit der Maximalanzahl von 12 korrekten Einträgen um 2 Einträge geringer als bei Team *Security*. Andererseits ist das Safety-Team mit 4 korrekten Bedrohungen im Minimum besser als das Security-Team. Der Median ist mit einer Anzahl von 7 korrekt identifizierten Bedrohungen bei beiden Teams identisch. Als letzte Messgröße – den als falsch bewerteten Bedrohungen (FP) – erreicht das Safety-Team eine Maximalanzahl von 4, im Median 1,75 und im Minimum 0 falsche Bedrohungen.

Wie bereits für Team *Security*, zeigt an dieser Stelle Tabelle B.3 die Messwerte für Präzision (PPV), Effizienz und Produktivität (PRO) der einzelnen Teilnehmer von Team *Safety*.

Tabelle B.3: Die im Experiment für die Teilnehmer 1-8 aus Team *Safety* erreichten Ergebnisse hinsichtlich der Präzision (PPV), Effizienz und Produktivität (PRO).

Teilnehmer	1	2	3	4	5	6	7	8
PPV	1	0,9	0,86	0,83	0,67	0,75	1	1
Effizienz	36	40	28	24	24	48	20	32
PRO	0,6	0,67	0,47	0,4	0,4	0,80	0,33	0,53

Sowie die verbleibenden Messergebnisse für die Teilnehmer 9-14 des Safety-Teams mit Tabelle B.4.

Tabelle B.4: Die im Experiment für die Teilnehmer 9-14 aus Team *Safety* erreichten Ergebnisse hinsichtlich der Präzision (PPV), Effizienz und Produktivität (PRO).

Teilnehmer	9	10	11	12	13	14	∅	σ
PPV	0,8	0,77	0,73	0,83	0,92	0,80	84,75 %	0,11
Effizienz	20	52	60	24	52	20	34,29	13,32
PRO	0,33	0,87	1	0,4	0,87	0,33	57,14 %	0,22

Bezogen auf Tabelle B.4 lässt sich entnehmen, dass für Team *Safety*, im Durchschnitt ein Wert von 84,75 % für die Präzision erreicht wurde, unter einer Standardabweichung von $\sigma = 0{,}11$. Für die Effizienz wurde im Mittel 34,29 Schutzziele pro Stunde mit einer Standardabweichung von $\sigma = 13{,}32$ gemessen. Schließlich erreichte Team *Safety* im Durchschnitt eine Produktivität von 57,14 % bei einer Standardabweichung von $\sigma = 0{,}22$, was einem höheren Wert im Vergleich zu Team *Security* entspricht.

Wie für das Security-Team wurde auch für das Safety-Team das Flow-Erleben –während des Experimentes – nach Rheinberg et. al [164] gemessen und ausgewertet. Die Ergebnisse hierzu sind in Tabelle B.5 dargestellt.

Tabelle B.5: Rückmeldung der Teilnehmer 1-8 des Teams *Safety* auf das Flow-Erleben nach Rheinberg et. al. [164]. Gemessen wurde, wie stark sich der Teilnehmer im Flow befand, wie groß seine Bedenken waren und wie stark er sich herausgefordert fühlte.

Teilnehmer	1	2	3	4	5	6	7	8
Flow	5,6	4,4	5,1	3,5	3,1	5,0	2,1	4,4
Bedenken	2,6	1,3	1	1	3,3	1	1	2
Herausforderung	3	4	4	4	4	3	2	5

Sowie die Ergebnisse des Flow-Erlebens für die Teilnehmer 9-14 des Safety-Teams (Tabelle B.6).

Tabelle B.6: Rückmeldung der Teilnehmer 9-14 des Teams *Safety* auf das Flow-Erleben nach Rheinberg et. al. [164]. Gemessen wurde, wie stark sich der Teilnehmer im Flow befand, wie groß seine Bedenken waren und wie stark er sich herausgefordert fühlte.

Teilnehmer	9	10	11	12	13	14	∅
Flow	5	5,2	4,2	5,5	6,1	3,5	4,5
Bedenken	1	3	3	7	1,3	1	2,1
Herausforderung	1	3	2	4	4	4	3,4

Im Vergleich zu den Werten des Teams *Security* gaben die Teilnehmer des Teams *Safety* an, dass sie sich im Durchschnitt noch mehr im Arbeitsablauf befanden, noch weniger Bedenken bei der Aufgabe hatten und sich noch weniger von der Aufgabe herausgefordert fühlten. Dieses Ergebnis stützt die Annahme, dass unter Anwendung der SGM die Analysten strukturiert durch die Bedrohungsanalyse geführt werden und sich dadurch weniger gefordert fühlen.

B.2.3 Ergebnisse des Teams Unerfahren

Als letztes Team wird Team *Unerfahren* ausgewertet, das den wissenschaftlichen Mitarbeitern der Hochschule Karlsruhe entspricht. Die Ergebnisse dieser Gruppe sind in Abbildung B.24 aufbereitet. Sie zeigen im Generellen eine geringere Streuung der Ergebnisse bezogen auf die des Security-Teams in Abbildung B.22. Auch hier sind die erzielten Werte für die Klassen FI, FP und TP aufgetragen.

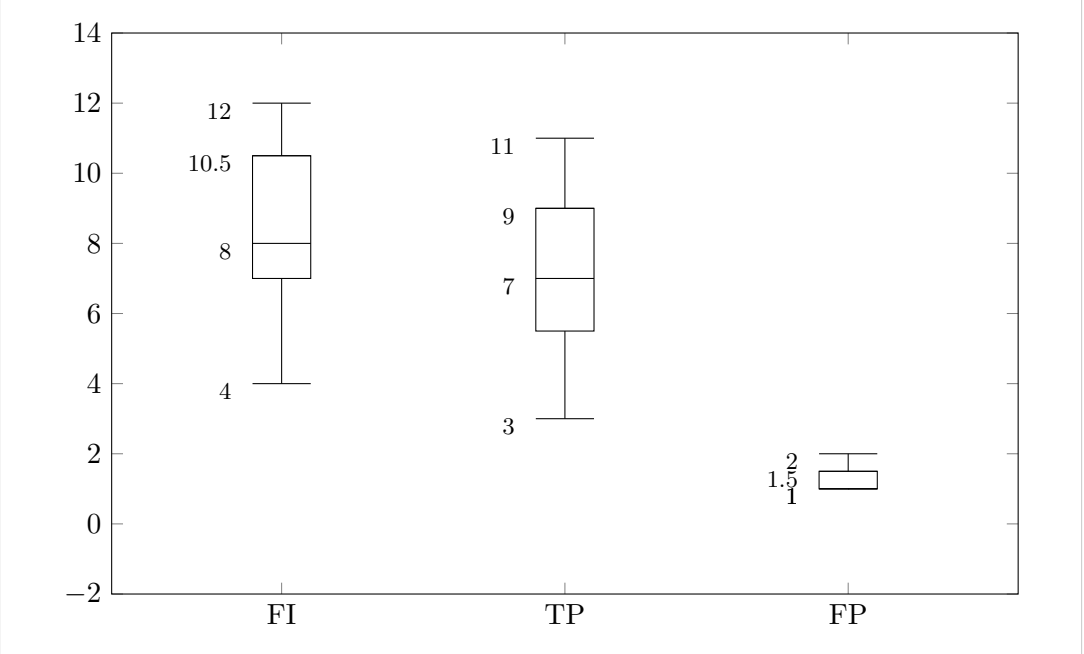


Abbildung B.24: Whisker-Plot für die Messergebnisse der ausgefüllten Zeilen (FI), der korrekten Bedrohungen (TP) und der als Falsch bewerteten Bedrohungen (FP) für das Team *Unerfahren*.

Mit Abbildung B.24 zeigt sich, dass das Team *Unerfahren* eine maximale Anzahl von 12 Zeilen (FI), im Minimum 4 und im Median 8 Bedrohungen während der Analyse identifiziert hat. Die maximale Anzahl ist damit geringer als bei Team *Security*, der Median allerdings bei beiden Teams identisch. Außerdem zeigt sich eine geringe Streuung in der FI-Klasse. Dies gilt ebenfalls für die Ergebnisse der TP Messung die mit einer maximalen Anzahl von 11 abermals geringer als die des *Security*-Teams ist. Der Median mit dem Wert 7 und die minimale Anzahl von korrekt identifizierten Bedrohungen ist hingegen größer als bei Team *Security*. Bei der Auswertung der nicht korrekt identifizierten Bedrohungen (FP) zeigt sich mit dem Maximalwert von 2 und dem Minimalwert von 1 eine kleinere Spanne als bei Team *Security*. Dies ist als positiv zu werten, da das Team *Unerfahren* weniger inkorrekte Zeilen ausfüllte als das Security-Team, was auf eine verbesserte Führung, während der Identifikation der Bedrohung hindeutet.

Wie bei den beiden anderen Teams wurde ebenfalls für Team *Unerfahren* Präzision (PPV), Effizienz und Produktivität (PRO) gemessen. Die Ergebnisse sind in Tabelle B.7 gezeigt.

Tabelle B.7: Die im Experiment für die Teilnehmer aus Team *Unerfahren* erreichten Ergebnisse hinsichtlich der Präzision (PPV), Effizienz, und Produktivität (PRO).

Teilnehmer	1	2	3	4	5	6	7	∅	σ
PPV	0,7143	0,9	0,75	0,818	0,917	0,876	0,857	83,33 %	0,0662
Effizienz	28	40	16	44	48	32	28	33,71	9,56
PRO	0,467	0,667	0,267	0,733	0,800	0,533	0,4667	56,19 %	0,0254

Bezogen auf Tabelle B.7 lässt sich sagen, dass Team *Unerfahren* im Durchschnitt einen Wert von 83,33 % für die Präzision erreicht hat, bei einer Standardabweichung von σ =0,0662. Für die Effizienz wurden im Mittel 33,71 Bedrohungen pro Stunde mit einer Standardabweichung von σ =9,56 gemessen. Schließlich wurde für die Produktivität im Durchschnitt ein Wert von 56,19 % mit einer Standardabweichung von σ =0,0254 festgestellt.

B.2.4 Wilcoxon-Test

Neben dem t-Test in Abschnitt 5.3 ist ebenso ein Wilcoxon-Test für die Auswertung der Hypothesen durchgeführt worden [82, Seite 527]. Gegenüber dem t-Test ist der Wilcoxon-Test unabhängig von der zugrundeliegenden Verteilung der Stichprobe. Tabelle B.8 fasst die Ergebnisse des Testes zusammen.

Tabelle B.8: Ergebnisse für den rechtsseitigen Wilcoxon-Test mit einem Signifikanzniveau von $\alpha = 10$ %. Die Nullhypothesen 1 bis 4 sowie die Schwellwerte (μ) der Testhypothesen (H_1) sind aus Abschnitt 5.1 abgeleitet. Die letzte Spalte beschreibt, ob die jeweilige Testhypothese akzeptiert oder abgelehnt ist.

Nullhypothese	**Akzeptiertes μ für H_1**	**Status H_0**	**p-Wert des t-Tests für H_0**	**Status H_1**
$H_{0,Saf}^{TP}$	$> 6{,}11$	abgelehnt	0,09	akzeptiert
$H_{0,Saf}^{FP}$	$> 1{,}67$	akzeptiert	0,12	abgelehnt
$H_{0,Saf}^{PPV}$	> 78.29 %	abgelehnt	0,01	akzeptiert
$H_{0,Saf}^{PRO}$	$> 51{,}85$ %	akzeptiert	0,30	abgelehnt

Die hierbei erzielten Testergebnisse stimmen mit jenen des t-Testes in Abschnitt 5.3 überein, sodass eine Beeinflussung der Testergebnisse durch eine falsche Annahme der Stichprobenverteilung ausgeschlossen werden kann.

B.3 Fragebogen zur Selbstwertung

Fragenbogen

Dieser Fragebogen dient ausschließlich der Sammlung von Daten über das Hintergrundwissen sowie der Stimmungslage der Teilnehmer.

Teilnehmernummer: ☐

I. Hintergrundwissen

1. In welcher Fachrichtung/Vertiefung hast du deinen Hochschulabschluss erworben?

_____

2. Wie viele Jahre arbeitest du schon bei der [Unternehmen]?

_____

3. In welcher Fachrichtung/Vertiefung arbeitest du bei der [Unternehmen]?

_____

4. Wie lange bist du in dieser Fachrichtung/Vertiefung schon tätig?

_____

5. In welchen der nachfolgenden Fachgebiete hast du bereits Kenntnisse erlangt?

- ☐ Automotive Software Engineering
- ☐ Funktionale Sicherheit (ISO 26262)
- ☐ Bedrohungsanalysen (Threat Analyse)
- ☐ Risikoanalysen (Risk Assessment)
- ☐ Kryptographie
- ☐ Embedded Systems

Weitere relevante Gebiete:

_____

6. Hast du bereits Tätigkeiten im Bereich der Funktionalen Sicherheit ausgeübt (z.B. Risikoeinschätzung)?

Bitte wähle eine der folgenden Optionen aus: ☐ Ja ☐ Nein

7. Hast du bereits Tätigkeiten im Bereich von Security ausgeübt?

Bitte wähle eine der folgenden Optionen aus: ☐ Ja ☐ Nein

7.1. Wenn du bei der vorherigen Frage mit JA geantwortet hast, dann beantworte bitte, ob du schon einmal an einer Bedrohungsanalyse (Threat Analysis) teilgenommen hast?

Bitte wähle eine der folgenden Optionen aus: ☐ Ja ☐ Nein

8. Ist dir die HAZOP (Hazard and operability study) Methode bekannt?

Bitte wähle eine der folgenden Optionen aus: ☐ Ja ☐ Nein

Falls du die Methode kennst, beschreibe diese bitte kurz:

Bitte verwende die folgende Skala für die Beantwortung der Fragen 9-15:

1 für Neuling: Ich habe minimales Wissen aus der Literatur ohne praktische Anwendung.
2 für fortgeschrittener Anfänger: Ich habe Wissen aus der Anwendung von Teilbereichen des Gebietes.
3 für Kompetenter: Ich habe Erfahrung und gutes technisches Verständnis in diesem Bereich.
4 für Gewandter: Ich habe ein tiefgehendes Verständnis in diesem Bereich.
5 für Experte: Ich bin ein Experte auf dem Gebiet.

9. Wie würdest du dein Wissen auf dem Gebiet der Funktionalen Sicherheit einschätzen?

Bitte verwende nur eine Antwortmöglichkeit:

Neuling ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 Experte

10. Wie würdest du dein Wissen auf dem Gebiet der Bussysteme von Autos einschätzen?

Bitte verwende nur eine Antwortmöglichkeit:

Neuling ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 Experte

11. Wie würdest du dein Wissen auf dem Gebiet der Echtzeitsysteme einschätzen?

Bitte verwende nur eine Antwortmöglichkeit:

Neuling ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 Experte

12. Wie würdest du dein Wissen auf dem Gebiet der Hazard Analyse einschätzen?

Bitte verwende nur eine Antwortmöglichkeit:

Neuling ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 Experte

13. Wie würdest du dein Wissen auf dem Gebiet des Software Engineerings einschätzen?

Bitte verwende nur eine Antwortmöglichkeit:

Neuling ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 Experte

14. Wie würdest du dein Wissen im Gebiet der Bedrohungsanalyse (Threat Analyse) einschätzen?

Bitte verwende nur eine Antwortmöglichkeit:

Neuling ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 Experte

15. Wie würdest du dein Wissen im Gebiet der Security einschätzen?

Bitte verwende nur eine Antwortmöglichkeit:

Neuling ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 Experte

Bitte kreuze für die Fragen 16 bis 22 jeweils eine einzige Antwort an.

---

16. Was ist ein ASIL? Bitte wähle die treffendste Antwort aus. (*Bitte keine Mehrfachnennung*)

▢ Der Automotive Safety Integrity Level (ASIL) ist eine Risikoklassifikation nach dem ISO 26262 Standard. Diese erlaubt es die Budgets für Funktionale Sicherheit angemessen zu verteilen.

▢ Der Automotive Safety Integrity Level (ASIL) erlaubt eine Klassifizierung von Risiken nach dem ISO 26262 Standard. Die Risiken werden in Gruppen von ASIL A zu ASIL D eingeordnet, wobei ASIL D die höchste Risikoklasse darstellt. Diese Risikoklassen erfordern die Abschätzung der notwendigen Investitionen zur Risikominimierung.

▢ Der Automotive Safety Integrity Level (ASIL) erlaubt eine Klassifizierung von Risiken im Rahmen der Spezifizierung des ISO 61508 Standards durch den ISO 26262 Standard. Die verschiedenen Risikogruppen erlauben eine für die Automotive Branche spezifische Klassifizierung von Risiken.

▢ Der Automotive Safety Integrity Level (ASIL) ist eine von dem Standard ISO 26262 vorgeschriebene Klassifizierung von Risiken. Die Risikoklassen werden von Experten für Funktionale Sicherheit vergeben um eine gerechte Verteilung des Budgets für die funktionale Sicherheit zu erreichen.

▢ Ich kann die Frage nicht beantworten.

17. Was ist ein CAN Bus? Bitte wähle die treffendste Antwort aus. (*Bitte keine Mehrfachnennung*)

▢ Ein CAN-Bus (Controller Area Network) ist ein serielles Bussystem, welches für die Vernetzung von Steuergeräten in Fahrzeugen eingesetzt wird. Die Steuergeräte können mittels des Busses kommunizieren ohne das Aufkommen von Kollisionen.

▢ Ein CAN-Bus (Controller Area Network) ermöglicht durch CSMA/CA eine kollisionsfreie Arbitrierung des Busses. Die gleichberechtigen Steuergeräte sind in diesem Bus topologisch miteinander verbunden.

▢ Ein CAN-Bus (Controller Area Network) ermöglicht eine bitweise Arbitrierung des Busses durch ein multiple access Verfahren. Diese Verfahren erlaubt es die Dringlichkeit von Busnachrichten zu priorisieren.

▢ Ein CAN-Bus (Controller Area Network) ist ein System zur Verteilung von Nachrichten von Steuergeräten in Automobilen. Der Bus erlaubt es durch eine geschickte Arbitrierung die Nachrichten in wichtig und weniger wichtig zu unterscheiden.

▢ Ich kann die Frage nicht beantworten.

18. Was ist harte Echtzeit? Bitte wähle die treffendste Antwort aus. (*Bitte keine Mehrfachnennung*)

▢ Bei Anforderungen von harter Echtzeit führt eine Überschreitung von Antwortzeiten zu einem Systemversagen. Für die Umsetzung von harten Echtzeitanforderungen werden deterministische Betriebssysteme benötigt.

▢ Bei Anforderungen von harter Echtzeit führt eine Überschreitung von Antwortzeiten zu Unfällen. Diese Systeme werden umgesetzt durch den Einsatz von präzisen Betriebssystemen, welche bestimmte Antwortzeiten garantieren können.

▢ Bei Anforderungen von harter Echtzeit können keine zeitlichen Abweichungen garantiert werden. Die Umsetzung dieser Anforderungen benötigen verlässliche Betriebssysteme die präzise Antwortzeiten umsetzen können.

▢ Harte Echtzeit Anforderungen erlauben keine Überschreitung von Antwortzeiten, weil dies ein Systemversagen zur Folge haben kann. Betriebssysteme die diese Antwortzeiten garantieren können sind zur Umsetzung von harten Echtzeitanforderungen unerlässlich.

▢ Ich kann die Frage nicht beantworten.

19. Ein Fault Tree ist? Bitte wähle die treffendste Antwort aus. (*Bitte keine Mehrfachnennung*)

▢ Ein Fault Tree ermöglicht es verschiedene low-level Ereignisse in Bezug zu setzen um ein unerwünschtes Systemverhalten zu analysieren. Für das unerwünschte Verhalten werden mögliche Ursachen ermittelt. Anschließend werden diese weiter verfeinert bis eine Reihe von low-level Ereignissen ermittelt werden.

▢ Ein Fault Tree ist eine deduktive Methodik zur Failure Analyse in welcher ein unerwünschter Zustand eines Systems analysiert wird. Der Einsatz von Boolean Logik ermöglicht es eine Reihe von low-level Ereignissen in Bezug zu setzen und mögliche Ursachen für den unerwünschten Zustand zu ermitteln.

▢ Ein Fault Tree ist eine deduktive Methodik zur Failure Analyse in welcher ein unerwünschter Zustand eines Systems analysiert wird. Dieser Zustand wird in Bezug zu verursachenden Ereignissen gesetzt mit AND und OR Gates. Diese Verfeinerung wird durchgeführt bis zu atomaren Ereignissen.

▢ Ein Fault Tree erlaubt es mögliche Ursachen für einen Failure zu bestimmen, durch den Einsatz von Boolean Operatoren. Diese werden im Rahmen einer Failure Analyse genutzt um deren mögliche Ursachen in Bezug zu setzen.

▢ Ich kann die Frage nicht beantworten.

20. Was ist die UML? Bitte wähle die treffendste Antwort aus. (*Bitte keine Mehrfachnennung*)

▢ Eine Modellierungssprache für Software die man beim Requirements Engineering und der Entwicklung von Software Architekturen einsetzt.

▢ Ein Standard zur semi-formalen Spezifizierung von Software. Der UML Standard ermöglicht das Verhalten und die Struktur von Software präzise darzustellen.

▢ Ein Standard zur semi-formalen Spezifizierung von Software. Dieser enthält verschiedene Diagrammtypen, unterteilt in die Spezifizierung von Verhaltens- und Strukturdiagrammen von Software.

▢ Ein Standard zur semi-formalen Spezifizierung von Klassendiagrammen. Diese Diagramme können die konzeptionelle und implementationsnahe Sicht auf Software dokumentieren.

▢ Ich kann die Frage nicht beantworten.

21. Was verstehst du unter einer Bedrohungsanalyse? Bitte wähle die treffendste Antwort aus. (*Bitte keine Mehrfachnennung*)

▢ Ist eine Methode zur systematischen Identifikation von Safety und Security Problemen, bei der eine Liste von technischen Sicherheitsmaßnahmen generiert wird um ein System sicherer zu gestalten.

▢ Ist eine Methode zur systematischen Identifikation und Bewertung von Bedrohungen (Threats), den ein System ausgesetzt sein kann.

▢ Ist eine Methode zum Abschätzen des Aufwandes von Security Maßnahmen. Die Methodik hilft bei der Festlegung des Budgets für Security Maßnahmen.

▢ Ist eine Methode zur systematischen Identifikation und Bewertung von Bedrohungen (Threats) und Schwachstellen, die ein System besitzen könnte.

▢ Ich kann die Frage nicht beantworten.

22. In welcher Beziehung stehen die Begriffe Besitzer (Owner), Vermögenswert (Asset), Bedrohung (Threat), Gefährdung und Risiko (Risk)? Bitte wählen Sie die treffendste Antwort aus. (*Bitte keine Mehrfachnennung*)

▢ Ein Asset ist alles, was für den Besitzer des Assets einen Wert darstellt. Weiterhin gibt es Bedrohungen die dem Asset schaden können. Für alle identifizierten Bedrohungen wird ein einziger Risikowert bestimmt. Gefährdungen werden in diesem Kontext nicht betrachtet.

▢ Anhand einer Liste von Bedrohungen wählt der Besitzer diejenigen Assets aus, die für ihn einen Wert darstellen. Außerdem dient die Liste zur systematischen Identifikation von Gefährdungen mit anschließender Festlegung des Risikos.

▢ Ein Asset ist ein monetärer Wert welcher für den Besitzer des Assets von Wert ist. Weiterhin gibt es Gefährdungen, die dem Asset schaden können. Anhand der identifizierten Gefährdungen kann das Risiko bestimmt werden.

▢ Assets beschreiben die Gesamtheit aller Objekte die für einen Besitzer von Wert sind. Weiterhin gibt es Bedrohungen die dem Asset schaden können. Für jede Bedrohung wird ein eigener Risikowert bestimmt. Gefährdungen werden in diesem Kontext nicht betrachtet.

▢ Ich kann die Frage nicht beantworten.

B.4 Fragebogen für das Flow-Erfahren

II. Stimmungslage

Versetze dich bitte noch einmal in die Bedrohungsanalyse zurück und beantworte die Fragen 1 bis 16.

Bitte keine Mehrfachnennungen.

1. Ich fühle mich optimal beansprucht.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

2. Meine Gedanken bzw. Aktivitäten laufen flüssig und glatt.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

3. Ich merke gar nicht, wie die Zeit vergeht.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

4. Ich habe keine Mühe, mich zu konzentrieren.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

5. Mein Kopf ist völlig klar.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

6. Ich bin ganz vertieft in das was ich gerade mache.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

7. Die richtigen Gedanken kommen wie von selbst.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

8. Ich weiß bei jedem Schritt, was ich zu tun habe.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

9. Ich habe das Gefühl, den Ablauf unter Kontrolle zu haben.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

10. Ich bin völlig selbstvergessen.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

11. Es steht etwas für mich Wichtiges auf dem Spiel.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

12. Ich darf jetzt keine Fehler machen.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

13. Ich mache mir Sorgen über einen Misserfolg.

Trifft nicht zu ☐ ☐ ☐ ☐ ☐ ☐ ☐ Trifft zu

14. Verglichen mit allen anderen Tätigkeiten, die ich sonst ausübe, ist die jetzige Tätigkeit…

leicht ☐ ☐ ☐ ☐ ☐ ☐ ☐ schwer

15. Ich denke, meine Fähigkeiten auf diesem Gebiet sind…

niedrig ☐ ☐ ☐ ☐ ☐ ☐ ☐ hoch

16. Für mich persönlich sind die jetzigen Anforderungen…

zu gering ☐ ☐ ☐ ☐ ☐ ☐ ☐ zu hoch

C

Fallstudie

In diesem Abschnitt sind weitere Erläuterungen zur aufgedeckten Schwachstelle des Airbag-Systems aus Kapitel 6 aufgezeigt. Das beinhaltet insbesondere die detaillierten Schritte zur Zündung der Airbags.

C.1 Diagnose im Fahrzeug

Damit Steuergeräte, Aktuatoren und Sensoren auf ihre grundlegenden Funktionalitäten geprüft werden können, besteht die Möglichkeit, diese zu diagnostizieren [162, Seite 417]. Hierzu speichern ECUs Fehler, die über Fehlercodes aus Fehlerspeichern ausgelesen werden können. Ebenso können durch bestimmte Diagnosefunktionen Aktuatoren angesteuert und damit in der Werkstatt überprüft werden. Die Diagnosefunktionen ermöglichen auch das Aufspielen von Firmware auf Steuergeräte, um sie aktualisieren zu können. Hierzu wird eine Diagnosekommunikation zwischen einem Diagnosetester und der zu überprüfenden Einheit aufgebaut. Die Kommunikation setzt dabei auf definierte Diagnoseprotokolle, die den oberen Schichten des ISO/OSI-Schichtenmodells entsprechen [162, Seite 423]. Das hierzu eingesetzte Protokoll ist mit der ISO 14229:2013 [98] spezifiziert und ist unter dem Namen Unified Diagnostic Services (UDS) bekannt. Die Kommunikation mit UDS findet hauptsächlich über den OBD-Port statt, der die Verbindung zwischen dem Diagnosetester und dem Fahrzeugnetzwerk ermöglicht. Für der Aufbau der Kommunikation zwischen dem Diagnosetester und dem Steuergerät legt der Tester die *physikalische Adresse* des Steuergerätes auf die Leitung (request) und das Steuergerät antwortet (response) auf diese [31, Seiten 245–248]. Anschließend kann der Tester vordefinierte Diagnosedienste aufrufen, die mit der ISO 14229 [98] normiert sind. Die Dienste sind dabei in zwei Klassen eingeteilt, die « Standarddienste », die als *Default Session* bezeichnet werden und die « Nichtstandarddienste » die als *Non-Default Session* beschrieben werden. Letztere sind Diagnosedienste, die kritische Messdaten lesen und schreiben oder Funktionen ausführen, die nur in einem kontrollierten Umfeld, wie einer Werkstatt, ausgeführt werden sollten. Ein Beispiel

hierfür ist das Entlüften der Bremszylinder, was zu einer fehlenden Bremsleistung während des Entlüftens führt. Aufgrund dessen muss für diese Funktionalitäten von einer Standardsitzung in eine spezielle Diagnosesitzung gewechselt werden, was in Abbildung C.1 beschrieben ist.

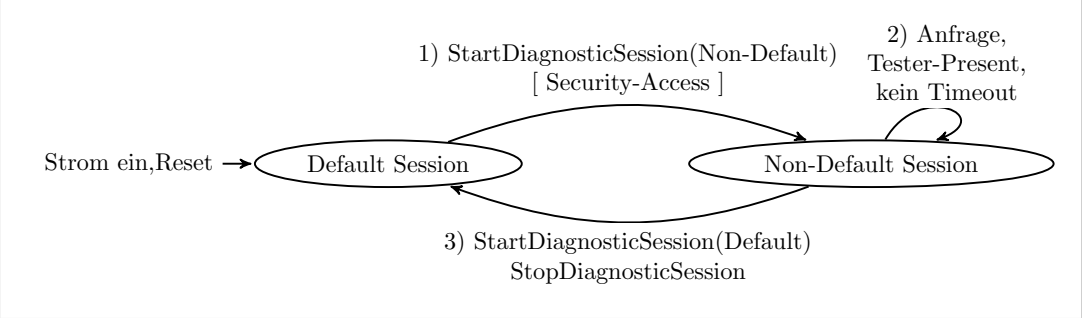


Abbildung C.1: Zustandsautomat für die Diagnosesitzungen nach UDS, die in der ISO 14229 [98] definiert sind.

Der Zustandsübergang 1) zeigt dabei die Anfrage des Testers für den Wechsel aus einer Standardsitzung in eine Nichtstandardsitzung (*Non-Default Session*), um kritische Funktionen ausführen zu können. Damit dieser Zustandswechsel vom Steuergerät erlaubt wird, muss ein Autorisierungsmechanismus durchlaufen werden. Dieser wird als *Security Access (SA)* bezeichnet und basiert auf dem sogenannten « Challenge and Response » Verfahren. Der *Seed* entspricht einer Zufallszahl, die einem geheimen Algorithmus übergeben wird und der daraus einen *Key* erzeugt. Unter der Annahme, dass zwei Entitäten den gleichen geheimen Algorithmus und einen identischen *Seed* besitzen, können beide den gleichen *Key* berechnen. Hiermit ist es möglich eine Authentifizierung zu etablieren, die anhand des Ablaufes in Abbildung C.2 beschrieben ist. Hierzu fordert der Diagnosetester vom Steuergerät einen *Seed* an und erhält die Zufallszahl Z_G. Diese wird im Diagnosetester mit dem geheimen Algorithmus F_{Dia} in den Key K_{Dia} übersetzt und an das Steuergerät zurückgesendet. Ebenso berechnet das Steuergerät aus der Zufallszahl einen Key ($F_{St}(Z_G) = K_{St}$). Anschließend überprüft das Steuergerät, ob der erhaltene Key K_{Dia} gleich dem berechneten Key K_{St} ist ($K_{Dia} == K_{St}$). Ist dies der Fall, wird der Security Access gewährt, da angenommen wird, dass der angefragten Entität vertraut werden kann. Hierbei gilt die Annahme, dass ausschließlich vertrauenswürdige Entitäten den geheimen Algorithmus F besitzen.

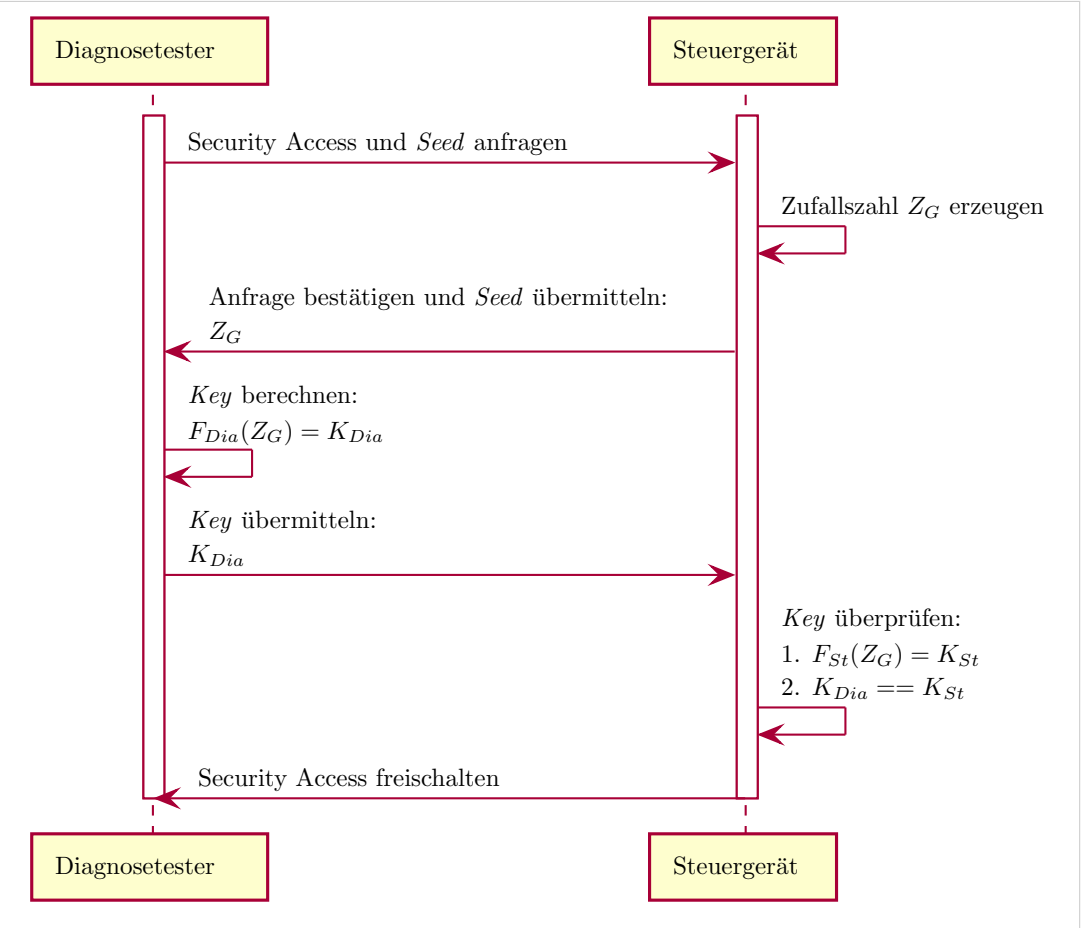


Abbildung C.2: Sequenzdiagramm für den SA, um von einer Standarddiagnosesitzung in eine Nichtstandardsitzung wechseln zu können.

Wurde der SA korrekt beantwortet, so kann der Diagnosetester Anfragen für erweiterte Diagnosedienste stellen. Dies ist mit der Schleife 2) in Abbildung C.1 repräsentiert. Die Sitzung bleibt dabei erhalten, solange korrekte Anfragen oder die Botschaft *TesterPresent* innerhalb eines festen Zeitfensters an das Steuergerät gesendet wird. Wird hingegen ein Timeout erkannt oder die Anfrage für den Wechsel in die Standardsitzung (*StartDiagnosticSession(Default)*, *StopDiagnosticSession*) empfangen, wechselt das Steuergerät in die Standarddiagnosesitzung zurück.

C.2 Ablauf der Airbag-Zündung nach ISO 26021

Die Vorgehensweise zur Ausnutzung der ausgewählten Airbageinheit (PCU) wird in sieben Schritten beschrieben und ist in Abbildung C.3 dargestellt. Die Schritte basieren auf der Auslösung von Airbags über UDS, die aus der Norm ISO 26021 entnommen werden können.

Im ersten Schritt wurde der UDS-Dienst *Read Data By Identifier* verwendet, um Informationen vom Ziel abzurufen. Dies lieferte wertvolle Informationen, wie zum Beispiel die Anzahl der verbauten PCUs im Fahrzeug, den CAN-Identifier zur Kommunikation

sowie die Fahrzeug-Identifizierungsnummer (VIN). Obwohl ein Fahrzeug über mehr als eine PCU verfügen kann, zeigte keines der hier getesteten Fahrzeuge mehr als eine PCU.

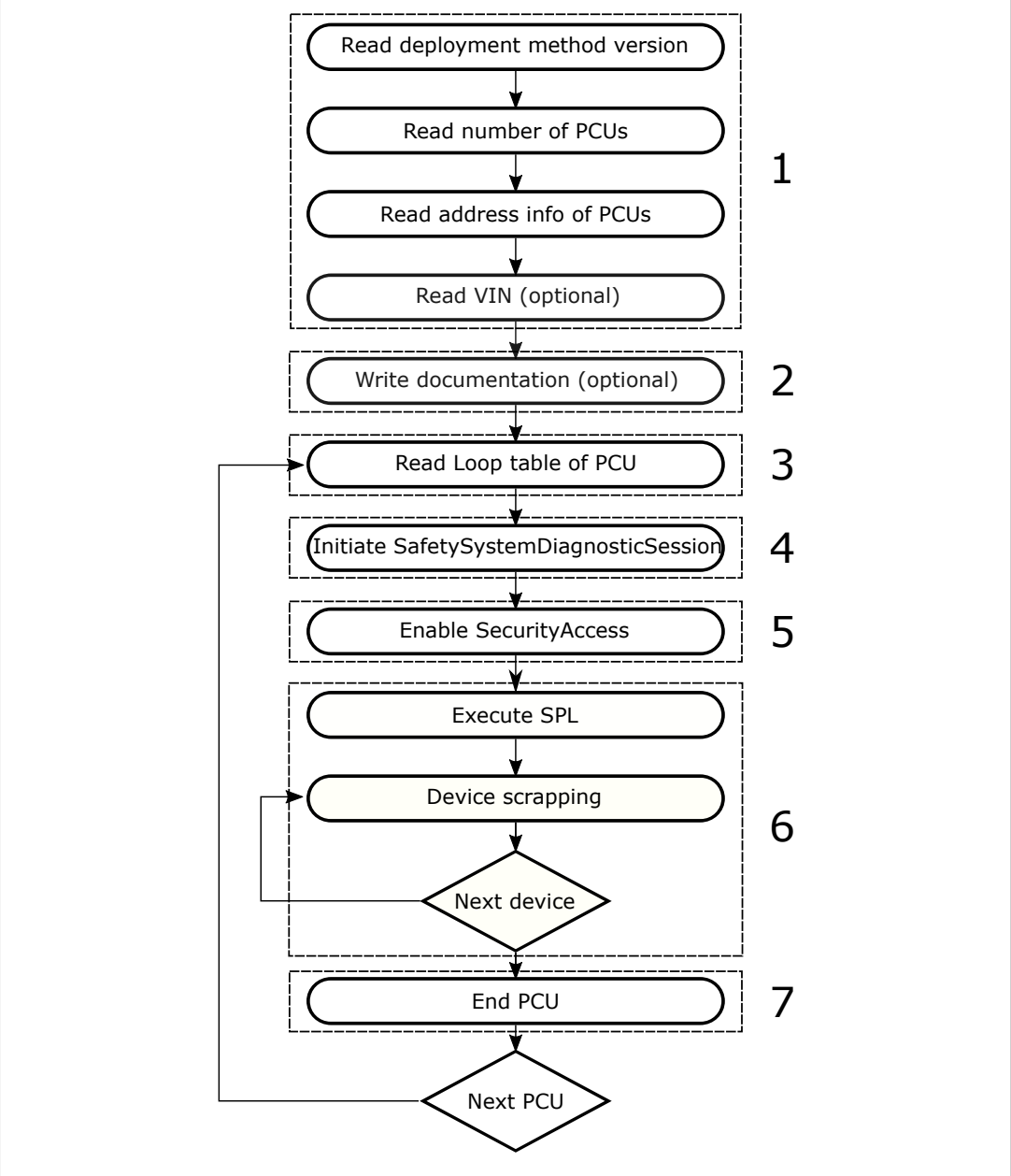


Abbildung C.3: Die während der Penetrationstests durchgeführten Schritte, die sich auf das Ablaufdiagramm nach der ISO-Norm 26021 [95] stützen.

Der zweite und optionale Schritt dient dazu, Informationen in der PCU zu speichern. Hierzu kann der Dienst *Write Data By Identifier* verwendet werden. Dieser ermöglicht es, dass Datum der Airbag-Auslösungen oder Informationen über die Person, die den Recycling-Einsatz durchführt, zu speichern. Da ein Penetrationstest allerdings einen bösartigen Angriff simuliert wurde diese Funktion nicht verwendet, da es unwahrscheinlich ist, dass ein Angreifer absichtlich Spuren hinterlassen wird.

Mit Schritt 3 und dem Dienst *Read Data By Identifier* war es möglich, alle verfügbaren Sprengkapseln (*loops*) auszulesen. Dies beinhaltet unter anderem den aktuellen Status der Sprengkapseln. Hiermit konnte im Nachgang validiert werden, ob eine Sprengkapsel erfolgreich ausgelöst wurde. Dies war durch Änderungen der internen Statuscodes sichtbar und wurde vom Prüfstand überwacht.

In Schritt 4 wurde der Dienst *Diagnostic Session Control* genutzt, um Zugriff auf erweiterte Diagnosefunktionen durch den Wechsel in eine *Safety System Diagnostic Session* zu erhalten. Dies erforderte allerdings, dass bestimmte Betriebsbedingungen erfüllt sind, die zu Beginn des Testes nicht bekannt waren. Unglücklicherweise waren die Fehlerantworten (Negative Response Codes (NRCs) [98]) der PCU unspezifisch, den die PCU sendete die Antwort *conditions not correct*. Dies ermöglichte somit keinen Hinweis auf die Hintergründe der Ablehnung von gesendeten Anfragen. Durch die Änderung der physikalischen Signale und einer Bussimulation, gemäß der NRC in Kombination mit Szenarien, die für den Fahrzeugzustand während eines Recycling-Einsatzes sinnvoll sind, konnten die Betriebsrandbedingungen identifiziert werden. So sendete die PCU beispielsweise den Fehlerantwort *RPM zu hoch*, was eine Anpassung der Botschaften ermöglichte. Basierend auf diesen Antworten konnten die Randbedingungen weiter modifiziert werden, um Kombination von Fahrzeugzustandswerten aufzudecken, die von der PCU akzeptiert werden. Eine weitere Einschränkung, die hierbei festgestellt wurde, ist die Notwendigkeit des zyklischen Sendens der Botschaft *Diagnostic Tester Present*. Ohne diese regelmäßige Nachricht entstand eine Zeitüberschreitung und die PCU verließ die Diagnosesitzung. Dieses Problem ließ sich allerdings durch ein zyklisches Senden der *Diagnostic Tester Present* Nachricht lösen. An dieser Stelle sei hervorgehoben, dass dieses eine weitere Schwachstelle der PCU darstellt. Denn der Diagnosetester wurde durch keine ausreichend sichere Maßnahme authentifiziert, sodass ein Wiedereinspielen von Botschaften des Diagnosetesters möglich war.

In Schritt 5 schützt der Diagnosedienst *Security Access* den Zugriff auf bestimmte Funktionen, um sicherzustellen, dass ein Diagnosetester berechtigt ist, auf diese Funktionen zuzugreifen. Hierzu verwendet der Dienst ein Seed-and-Key Verfahren. Um mehr Informationen über diesen Algorithmus zu erhalten, wurden mehrere Seeds angefragt und ein Brute-Force-Angriff durchgeführt. Diese ermöglichte die Rekonstruktion des Algorithmus, der sich als das Einerkomplement darstellte. Nach der Bestimmung des richtigen Algorithmus wurde die Berechnung in das Angriffsskript implementiert, um den Angriff zu automatisieren.

In Schritt 6 wurde der Dienst *Routine Control* verwendet, um die Zündung (Scrapping Program Module Loader (SPL)) der Sprengkapseln auszuführen. Hierzu wird der Programmcode des Zündmoduls (Scrapping Program Module (SPM)) in einen ausführbaren Programmcode konvertiert und in den RAM der PCU geladen. Ohne diese Vorbereitung ist das Auslösen der Airbags nicht möglich. Sind diese Vorbereitungen allerdings getätigt, kann mit dem Diagnosedienst *Routine Control* eine

bestimmte Sprengkapsel gezündet werden. Dies wird durch das Senden der Kennung der Sprengkapsel (*loop ID*) im 5. Byte der *Routine Control* Botschaft ermöglicht. Hierbei können auch mehrere pyrotechnische Ladungen gezündet werden, indem die Botschaft abermals jedoch mit einer anderen *loop ID* gesendet wird.

Im letzten Schritt wird die Zündung der Sprengkapseln beendet. Hierzu kann mit dem Diagnosedienst *EcuReset* das Steuergerät neugestartet werden. Um allerdings gleiche Vorbedingen für einen erneuten Angriff(Testfall) herzustellen wurde das Airbag-Steuergerät Spannungsfrei geschaltet. Dies führt zu einem Verlust der Daten im Random-Access Memory (RAM) und stellte sicher, das keine weitere Sprengkapsel zufällig gezündet wird.

D

Generierung von Angriffspfaden

In diesem Abschnitt sind Artefakte für die Modellierung und Generierung von Angriffspfaden aus Kapitel 7 näher beschrieben. Das ist der Ablauf zur Applikation von Schwachstellen in D.1 und eine beispielhafte Modellierung mit GAL in D.2.

D.1 Ablauf der Erzeugung des Systemmodells M

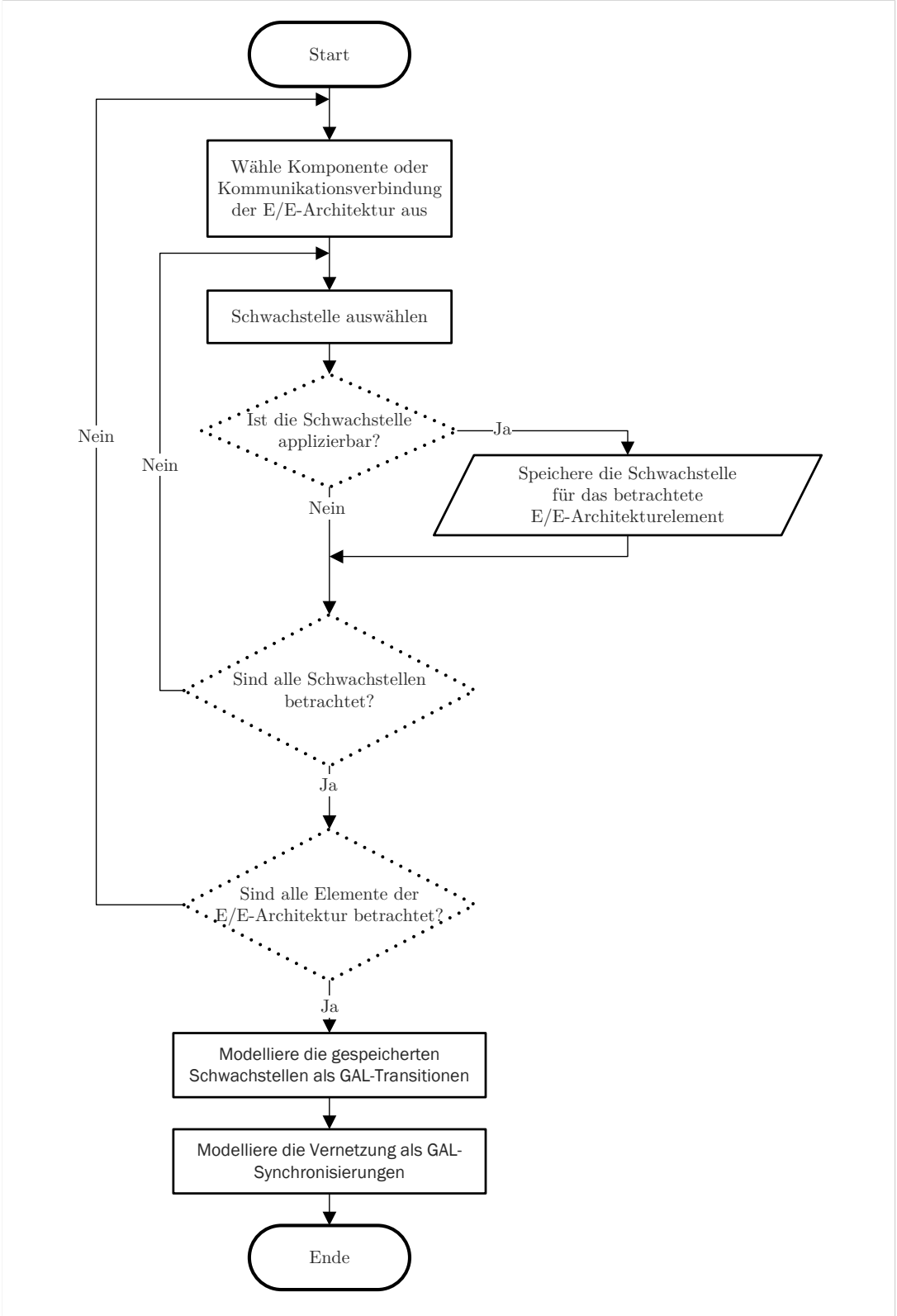


Abbildung D.1: Ablaufdiagramm zur Modellierung einer E/E-Architektur mit dessen Schwachstellen, unter der Verwendung der GAL für den Model-Checker ITS-tools [202].

D.2 Modellierung mit der GAL

Die konzeptionelle Modellierung mit der Guarded Action Language (GAL) [201] ist mit Abbildung D.2 gezeigt. Hierbei werden zwei Konzepte eingesetzt. Diese sind Typendeklarationen beziehungsweise Systeme, welche mit dem Schlüsselwort *gal* beschrieben sind und Variablen sowie Transitionen beinhalten. Nach dem Schlüsselwort folgt ein Name, der in der Abbildungen entweder durch den Platzhalter *Element* oder als Angreifer (*Attacker*) beschrieben sind. Ein *Element* entspricht dabei einem Stellvertreter aus der E/E-Architektur und kann als Komponente oder Kommunikationsverbindung Instanziiert werden.

Als weiteres GAL-Artefakt sind die Synchronisierungen (*synchronization*) zu nennen die ebenfalls in Abbildung D.2 aufgezeigt sind. Sie dienen der Synchronisierung von instanziierten Systemen und ermöglichen es, Zustände zwischen den Artefakten auszutauschen. Im konkreten Fall wird damit die Position des Angreifers in der E/E-Architektur und seine erlangten Rechte synchronisiert. Dies ist notwendig, da der Angreifer zwischen den E/E-Architekturelementen und unter verschiedenen Rechtelevel (RL) wechseln kann. Genauer gesagt um zwischen den fünf RL, die in Abschnitt 7.3 definiert sind. Wird eine Synchronisation ausgelöst, so führt sie atomar alle darin enthaltenen Aktionen sequentiell aus und ermöglicht es von einem Quellzustand (s_i) zu einem Nachfolgezustand (s_{i+1}) zu wechseln. Die Bezeichnungen in den Synchronisierungen entsprechen den instanziierten GAL-Artefakten und deren Transitionen. Für den Angreifer entspricht beispielsweise *Attacker.attackerPosRL(D)*, das dieser an Position *D* mit dem Rechtelevel *RL* sitzt. Hierbei trennt der Punkt in der Bezeichnung das instanziierte Artefakt von dessen Transitionen, was dem objektorientierten Prinzip ähnelt. Das Angreifer-Artefakt dient im Systemmodell primär der Speicherung des aktuell angegriffenen Elementes und welche Rechte der Angreifer auf diesem Element bereits erlangt hat.

Der Initialzustand ist mit der Synchronisierung *startOfAttack* repräsentiert, bei dem der Angreifer Zugriff (*Element.attackerAccess(1)*) besitzt. Dies entspricht dem Eintrittspunkt, welcher in der sechsten Spalte der SGM-Tabelle festgehalten ist (Tabelle 4.2). Als ein Beispiel hierfür kann der OBD-Port verstanden werden, welchen der Angreifer nutzt, um seinen ersten Angriffsschritt auszuführen.

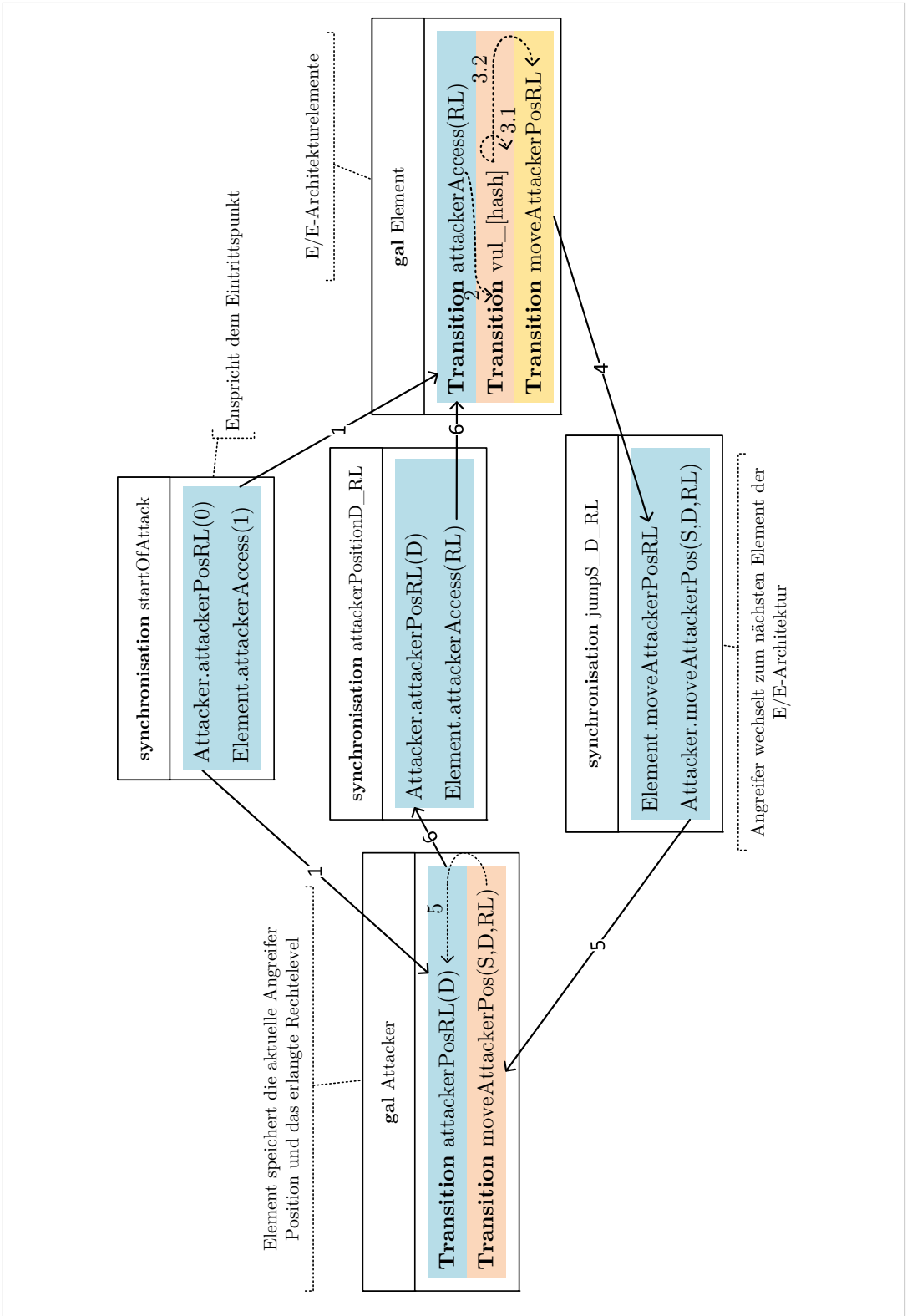


Abbildung D.2: Konzeptionelle Darstellung der Modellierung mit der Guarded Action Language (GAL) [201], um die E/E-Architekturelemente und deren Vernetzung mit dem Rechtekonzept aus Abschnitt 7.3 in ein gemeinsames Systemmodell zu übertragen. Modelliert ist der Angreifer (Attacker) sowie die E/E-Architekturelemente (Element). Die verbleibenden drei Synchronisierungen (synchronisation) beschreiben, wie ein Angreifer Schwachstellen auf einem Element ausnutzen und wie er von einem zu einem anderen wechseln kann. Die nummerierten Pfeile zeigen dabei die Reihenfolge, wie die jeweiligen GAL-Artefakte miteinander agieren können.

Beim Wechseln auf ein Element wird durch die Transition *attackerAccess(RL)* das auf dieser Komponente erreichte Rechtelevel (RL) gesetzt. Der Angreifer besitzt damit Zugriff auf das E/E-Architekturelement unter einem bestimmten Rechtelevel, was ihn befähigt die jeweiligen Schwachstellen im Element auszuführen. Dies ist durch die Transition *vul_[hash]* repräsentiert (Abbildung D.2). Der Platzhalter *[hash]* beschreibt dabei eine eindeutige Zeichenfolge, welche die Schwachstelle kennzeichnet und durch einen Hash-Algorithmus erzeugt wird. Hiermit ist es im Nachgang möglich, weitere Details der Schwachstelle zuzuordnen, die für das Systemmodell nicht von Relevanz waren. Das kann beispielsweise die volle textuelle Beschreibung der Schwachstelle sein.

Tritt der Fall ein, dass die RL des Angreifers mit der Schwachstellentransition übereinstimmen, so wird letztere ausgelöst. Dies hat zur Folge, dass der Angreifer durch das Ausnutzen der Schwachstelle ein höheres RL erreichen kann. An dieser Stelle müssen allerdings zwei Situationen unterschieden werden. Zum einen kann der Angreifer Rechte erreichen, die es ihm erlauben weitere Schwachstellen im Element auszulösen. Zum anderen besitzt das Element womöglich eine Schwachstelle, die einen Wechsel des Elementes ermöglicht, was in Abbildung D.2 mit der Markierung 3.2 gekennzeichnet ist. Ist dies der Fall, so wird eine weitere Transition ausgelöst (*moveAttackerPosRL*), die dem Wechsel des Elementes und dem Übergang 4 in Abbildung D.2 entspricht. Der Wechsel des Elementes wird durch die Synchronisierung *jumpS_D_RL* koordiniert, wobei die Großbuchstaben in der Bezeichnung dem Quellelement (S), dem Zielelement (D) und dem jeweiligen RL beim Wechsel entsprechen. Das Quellelement ist jenes, das durch die Transition (*moveAttackerPosRL*) ausgelöst wurde. Dieses Vorgehen erlaubt auch die Modellierung von Broadcast-Verbindungen wie dem CAN, da der Angreifer von einem Element – wie dem CAN – zu allen angebundenen Zielelementen (nicht-deterministisch) wechseln kann. Hiermit unterscheidet sich das Vorgehen beispielsweise vom Microsoft Threat Modeling Tool [154], das nicht fähig ist, Broadcast-Verbindungen zu modellieren. Die Informationen über den Wechsel des Elementes wird anschließend durch die Transition *Attacker.moveAttackerPos(S,D,RL)* in der Angreifer-Instanz (*Attacker*) gespeichert. Hierzu übergibt die Transition das Quellelement (S), von dem gewechselt wurde, das Zielelement (D) auf dem der Angreifer aktuell sitzt sowie das Rechtelevel (RL), welches der Angreifer auf dem Zielelement erlangt hat. Dies entspricht dem Übergang 5 in Abbildung D.2, was dazuführt, dass die Transition *attackerPosRL(D)* des Angreifer-Artefakts aktiviert wird. Diese löst mit Übergang 6 die letzte Synchronisierung (*attackerPositionD_RL*) aus, welche die zuvor gespeicherten RL an das Zielelement weiterreicht und bei diesem die Variable für den erreichten Zugriff (RL) setzt. Von nun an wiederholen sich die Schritte 2 bis 6, bis alle Elemente durchlaufen und der Zustandsraum vollständig aufgebaut ist.

D.2.1 GAL Beispiel

Im Folgenden wird die Modellierung der E/E-Architektur des Airbags-Systems (Abbildung D.3) mit der GAL des Model-Checkers ITS-tools beschrieben.

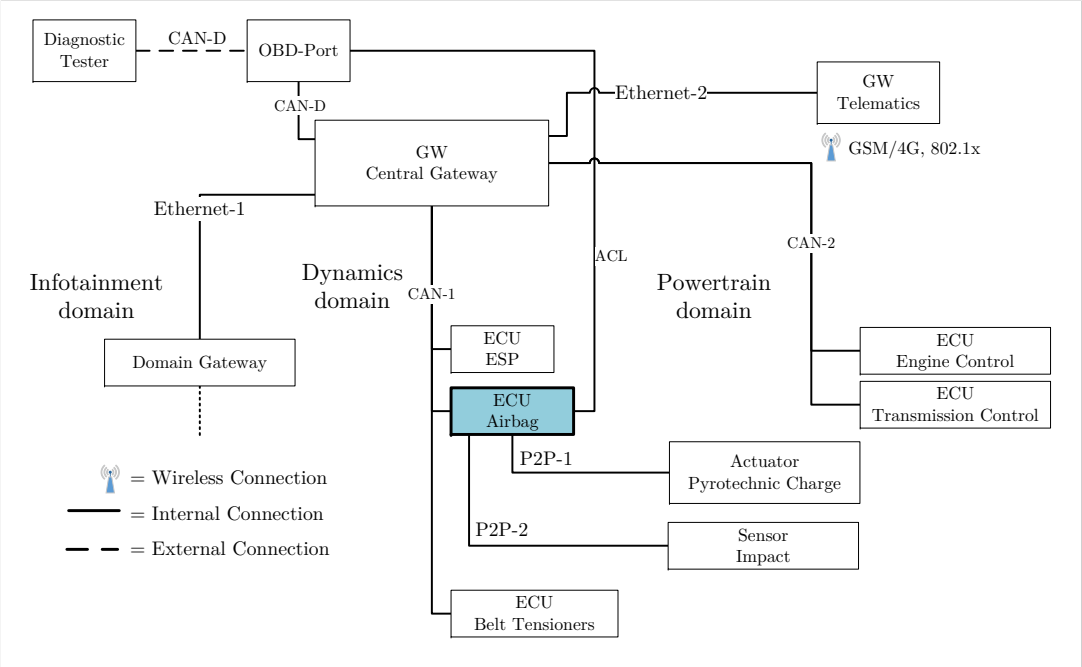


Abbildung D.3: E/E-Architektur des Airbag-Systems.

Auflistung D.1: Schwachstelle die als GAL-Transition fomuliert ist

```
1 transition vul_4113837603 [access == 1 && acc_priv == 2&& visit == 0 ] {
2 acc_priv = 5;
3 c = 1;
4 a = 1;
5 ia = 1;
6 jump = 5;
7 self."moveAttPos5";
8 visit = 1;
9 }
```

Zeile 1 der Listing D.1 zeigt den Rumpf der Transition mit dem Namen der Schwachstelle *vul_4113837603*. Dieser setzt sich aus dem Prefix *vul_* und einer eindeutigen Zeichenfolge zusammen, um eine spätere Zuordnung der Schwachstellen und deren Eigenschaften zu ermöglichen. Gefolgt wird dies vom Wächter (engl. guard) der Transition in den eckigen Klammern, welche die Bedingungen beschreiben die erfüllt sein müssen ([access == 1 && acc_priv == 2 && visit == 0]).

Neben diesen wird auch der Angreifer modelliert, der entweder seine Rechte auf einer Komponente ausweitet oder von einer Komponente zur nächsten wechselt. Hierzu fließt die Vernetzung der Komponenten und deren Kommunikationsverbindung ein. Diese können P2P oder Broadcast-Verbindungen wie der CAN sein. Jede Komponente

oder Kommunikationsverbindung enthält eine eindeutige ID. Der Angreifer wird wie in Listing D.2 dargestellt, modelliert.

Auflistung D.2: Modellierung des Angreifers unter Verwendung der GAL.

```
1 gal Attacker($A = 0) {
2 int req = 0;
3 int pos = $A;
4
5 transition attPos(id_t $id)[pos == $id] label "attPos"($id) {
6 //keine Aktion notwendig
7 }
8 transition attPos1(id_t $id)[pos == $id && req == 1] label "attPos1"($id) {
9 //keine Aktion notwendig
10 }
11 transition attPos2(id_t $id)[pos == $id && req == 2] label "attPos2"($id) {
12 //keine Aktion notwendig
13 }
14 transition attPos3(id_t $id)[pos == $id && req == 3] label "attPos3"($id) {
15 //keine Aktion notwendig
16 }
17 transition attPos4(id_t $id)[pos == $id && req == 4] label "attPos4"($id) {
18 //keine Aktion notwendig
19 }
20 transition attPos5(id_t $id)[pos == $id && req == 5] label "attPos5"($id) {
21 //keine Aktion notwendig
22 }
23 transition moveAttPos(id_t $src, id_t $dst, req_t $req)[pos == $src] label "
       moveAttPos"($src, $dst, $req) {
24 pos = $dst;
25 req = $req;
26 }
27
28 }
```

Die Vernetzung lässt sich folgendermaßen modellieren:

Auflistung D.3: Modellierung der Vernetzung und Synchronisierung (GAL).

```
1
2 composite eeArchitecture
3 {
4 Attacker att($A=5);
5 OBD OBDPort ($ID=0);
6 CAN DCAN ($ID=7);
7 ECU AirbagECU ($ID=1);
8 Ethernet Ethernet1 ($ID=8);
9 Gateway ZGateway ($ID=4);
10 P2P AirbagLine ($ID=9);
11 Aktor AirbagCharge ($ID=3);
12 Sensor CrashSensor ($ID=2);
13 P2P SensorLine ($ID=12);
14 CAN CAN2 ($ID=11);
15 ECU EngineECU ($ID=5);
16 CAN CAN1 ($ID=10);
17 WirelessEndPoint WirelessEntryPoint ($ID=6);
18 Bluetooth BluetoothInterface ($ID=13);
19
20 synchronization startOfAttack {
21 att."attPos"(0);
22 OBDPort."attAccess"(2); // Angreifer hat Zugriff auf den OBD-Port
23 }
24 synchronization attackerPosition8_1 {
25 att."attPos1"(8);
26 Ethernet1."attAccess"(1);
27 }
28 .
29 .
30 .
31 synchronization jump0_7_1{
32 att."moveAttPos"(0,7,1); //Angreifer wechselt die Position von Element 0 zu 7 mit
       dem RL von 1
33 OBDPort."moveAttPos1";
34 }
35 .
36 .
37 .
38 main eeArchitecturetecture;
39 property p1 [never] : (AirbagCharge:acc_priv >= 2 || AirbagCharge:jump >= 2) &&
       AirbagCharge:ia == 1; //Spezifikation
40 }
```

E

Prototyp des Softwarewerkzeuges für die SGM

Das Ziel der hier vorgestellten Arbeit ist die Steigerung der Betriebssicherheit durch die Einbindung von Bedrohungsanalysen bei der Betrachtung von Gefährdung. Durch diese gemeinsame Analyse entsteht allerdings der Umstand, dass eine größere Anzahl von Situationen betrachten werden muss, die potenziell gefährlich sind. Um diesem entgegenzuwirken und dem Security-Analysten bei dieser Aufgabe zu unterstützen, wurde in Kapitel 7 ein Ansatz aufgezeigt, der Cyber-Bedrohungen computergestützt identifiziert und bewerten werden kann. Ein Teil dieses Ansatzes erfordert allerdings eine Überführung der E/E-Architektur und dessen Schwachstellen in ein formales Modell. Dieser Vorgang ist komplex und muss von Experten durchgeführt werden, welche die notwendigen Kenntnisse besitzen. Selbst in diesem Falle kann sich die Modellierung aufwendig gestalten und bereits für kleine E/E-Architekturen mit wenigen Steuergeräten und Verbindungstechnologien unpraktikabel werden. Aufgrund dessen, ist ein Softwarewerkzeug entwickelt worden, welches bei den unterschiedlichen Schritten der SGM unterstützt. Konkret sind das

- die Übertragung der E/E-Architektur in eine Graphenstruktur,
- die Aufnahme der SGM-Tabelle und deren Inhalte,
- das Importieren von Schwachstellen aus einer Schwachstellendatenbank und deren Zuweisung auf die E/E-Architektur,
- das Erzeugen des formalen Systemmodells M und der Spezifikation φ mit der GAL,
- die Interaktion mit dem Model-Checker und der Übernahme ausgegebener Gegenbeispiele,

- die Extraktion der Angriffspfade und deren Risikowerte aus dem Gegenbeispiel,
- die Priorisierung und Berichterstattung in Form textueller Angriffspfade.

Im ersten Schritt kann der Anwender des Werkzeuges eine E/E-Architektur modellieren. Hierbei ist die Annahme getroffen, dass die Architektur jener entspricht, die ebenso für die Gefährdungsanalyse verwendet wurde. Um diesen Vorgang zu unterstützen, stehen dem Anwender eine Bibliothek von Fahrzeugkomponenten und Kommunikationsverbindungen zur Verfügung, die in Abbildung E.1 auf der linken Seite erkennbar sind.

Abbildung E.1: Benutzeroberfläche des ASTMT mit der Komponentenbibliothek links und der Zeichenfläche für die E/E-Architektur in der rechten Bildhälfte.

Die Kommunikationsverbindungen folgen dabei der Einteilung nach Abbildung E.3. Hierbei kann der Anwender entweder das unspezifische Element *Kommunikationsverbindung* oder die konkretisierten Elemente aus *Ebene 1* wählen. Die Auswahlmöglichkeit der Kategorie *Schnittstelle* als höchste Abstraktionsebene ist damit zu begründen, dass

in der Konzeptphase womöglich nicht bekannt ist, welche konkreten Kommunikationsverbindungen eingesetzt werden. In diesem Falle kann der Anwender das Element *Kommunikationsverbindung* verwenden. Ist die Verbindungsart allerdings bekannt, so kann *Ebene 1* eingesetzt werden.

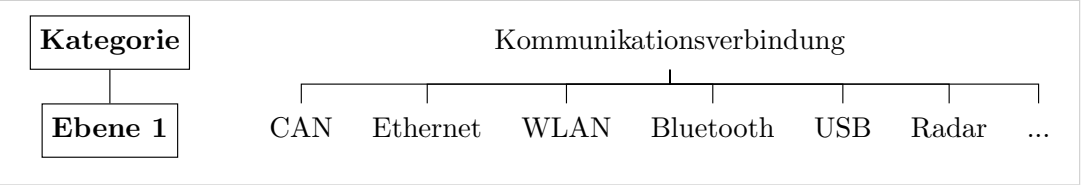


Abbildung E.2: Einteilung der Kommunikationsverbindungen nach der entwickelten Taxonomie in [13, Seite 12]. Hierbei entspricht die oberste Ebene (*Kommunikationsverbindung*) der höchsten Abstraktionsstufe.

In ähnlicher Weise verhält es sich für die Fahrzeugkomponenten. Hierbei kann der Anwender jedoch aus einer Kategorie und zwei Abstraktionsebenen auswählen. Die höchste Abstraktion in Abbildung E.3 besitzt die Kategorie *Komponente*, gefolgt von den Unterscheidungen zwischen ECU, Sensor und Aktuator (Ebene 1 in Abbildung E.3). Diese können weiter konkretisiert werden wie beispielsweise eine Motor-ECU oder einem Licht Sensor. Diese sind in der zweiten Ebene von Abbildung E.3 aufgeführt.

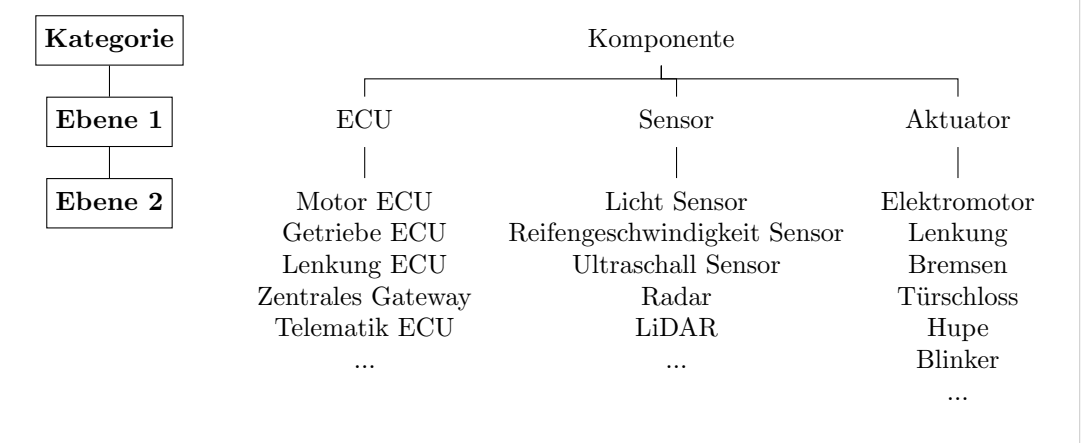


Abbildung E.3: Darstellung der unterschiedliche Abstraktionsebenen für Fahrzeugkomponenten, basierend auf [13, Seite 17].

Hierbei ist die gezeigte Einteilung im Einklang mit der entwickelten Schwachstellentaxonomie [13] und dem Metamodell in Abschnitt 4.1.2. Dies ermöglicht es die im Metamodell definierten Zusammenhänge auf die eingelesene Schwachstellen zu übertragen. Ein weiterer Vorteil hierbei zeigt sich bei der Übernahme der SGM-Tabelle und deren Inhalte. So kann dem Anwender eine interaktive SGM-Tabelle präsentiert werden, welche mit der fünften (Komponente/Teilsystem) und sechsten Tabellenspalte (Eintrittspunkt) bereits Komponenten und Kommunikationsverbindungen vorschlägt. Hierbei werden ausschließlich jene vorgeschlagen die in der gezeichneten

E/E-Architektur angeben wurden. Dies unterbindet fehlende Zuordnungen, da der Anwender nur angegebene Komponenten- oder Verbindungsbeschreibungen eintragen kann. Das gleiche gilt für die dritte Spalte der SGM-Tabelle, welche die Leitwörter beinhaltet. Auch diese werden dem Anwender vorgeschlagen, sodass direkt nach der Auswahl IT-Vermögenswerte und die Model-Checking Spezifikation mit Tabelle 7.2 abgleitet werden können. Die interaktive Tabelle ist in Abbildung E.4 in der rechten Bildhälfte hervorgehoben und im unteren Bildabschnitt ist das Ausgabefenster der Analyse zu erkennen. Hierbei wird der berechnete Risikowerte und der dazugehörige Angriffspfad angezeigt.

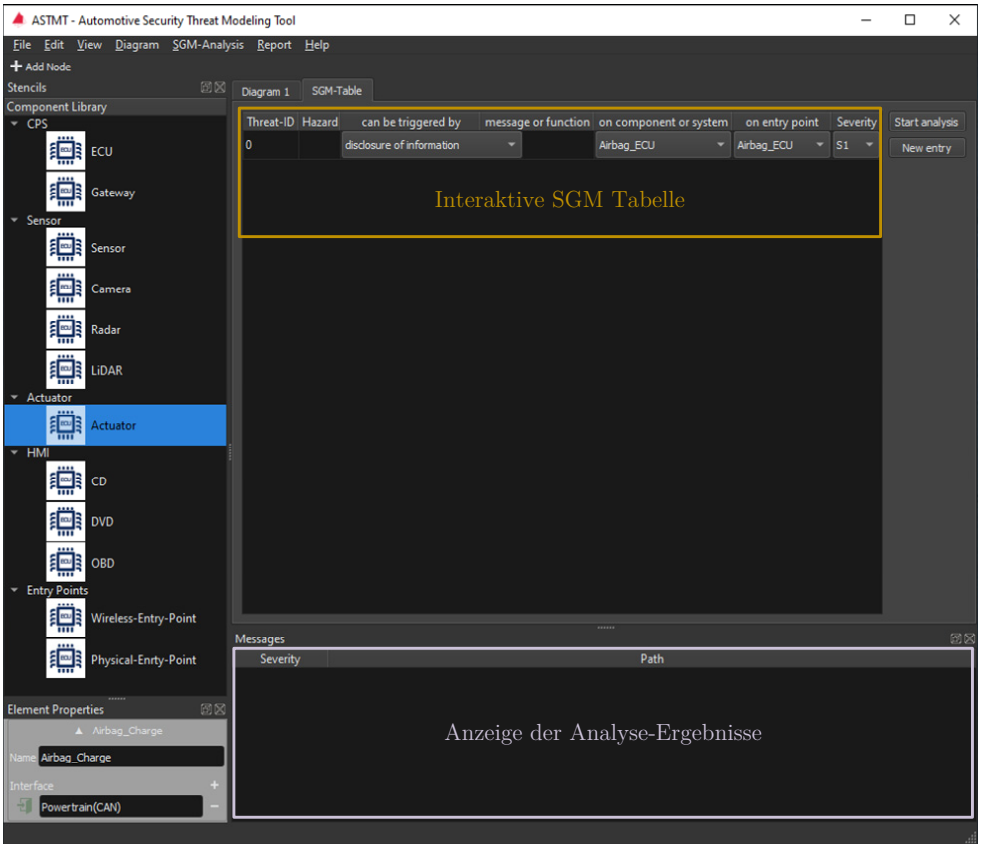


Abbildung E.4: Benutzeroberfläche des ASTMT mit der interaktiven SGM-Tabelle und dem Ausgabefenster für die Analyse-Ergebnisse in der rechten Bildhälfte.

Neben der Übernahme der Bedrohungsbeschreibung anhand eine interaktiven SGM-Tabelle, liest das Softwarewerkzeug (ASTMT) ebenso die bekannten Schwachstellen aus einer Datenhaltung ein (Abbildung E.5). Die Datenhaltung entspricht dabei der gleichen Struktur wie die entwickelte Schwachstellentaxonomie. Zur Zuweisung der Schwachstellen auf die E/E-Architektur wird das Vorgehen aus Abbildung D.1 eingesetzt. Ist dies abgeschlossen, erzeugt das ASTMT eine formales Systemmodell M und die dazu passende Spezifikation φ.

Das erzeugte GAL Modell wird anschließend dem Model-Checker [201] übergeben und validiert. So wird der Erzeugung des formalen Models und die Interaktion mit dem Model-Checker computergestützt bereitgestellt, sodass kein Experte für diese Schritte notwendig ist. Ebenfalls wird das Einlesen und Aufbereiten des Gegenbeispiels durch das ASTMT übernommen. Dieses wird durch den *Graphviz-Importeur* [76] eingelesen und repräsentiert einen DOT-Graphen [77, 115], der mit Schwachstelleninformationen angereichert wird. Auf diesem Graph wird im Anschluss die modifizierte Tiefensuche aus Abschnitt 2.5.4 angewendet, um die Angriffspfade zu extrahieren und deren Eintrittswahrscheinlichkeiten bestimmen zu können. Auch dieser Schritt wird ohne Eingriff des Anwenders durch das ASTMT durchgeführt. Ebenso wird ein Analysebericht erzeugt, der sich als textuelle Form eines Angriffspfades mit dessen Eintrittswahrscheinlichkeit darstellt.

Neben den externen Elementen, die mit dem ASTMT interagieren, zeigt das Kontextdiagramm in Abbildung E.5 außerdem zwei Personengruppen, die mit dem ASTMT in Aktion stehen. Zum einen der Anwender, welcher die E/E-Architektur und die Bedrohung anhand der ausgefüllten SGM-Tabelle an das Werkzeug übergibt, sowie die Angriffspfade und den Risikobericht exportiert. Die zweite Personengruppe, die ebenso mit dem ASTMT interagiert, sind Domänenexperten, welche Wartungsarbeiten übernehmen. Dies ist beispielsweise die Aktualisierung des Angreifermodells und der Regelsätze im ASTMT. Letztere entsprechen der Ableitung von Schwachstelleneigenschaften zu Rechteleveln, die für das Rechtemodell in Abschnitt 7.3 benötigt werden. Dies wird relevant, sollte die Taxonomie erweitert oder ein neues Rechtemodell eingesetzt werden. Ebenso beinhaltet dies die Regelsätze zur Transformation der Leitwörtern in die Spezifikation φ für den Model-Checker. Darüber hinaus werden die Domänenexperten ebenso für die Aktualisierung der Schwachstellendatenbank benötigt, um stets die neusten Security-Probleme betrachten zu können.

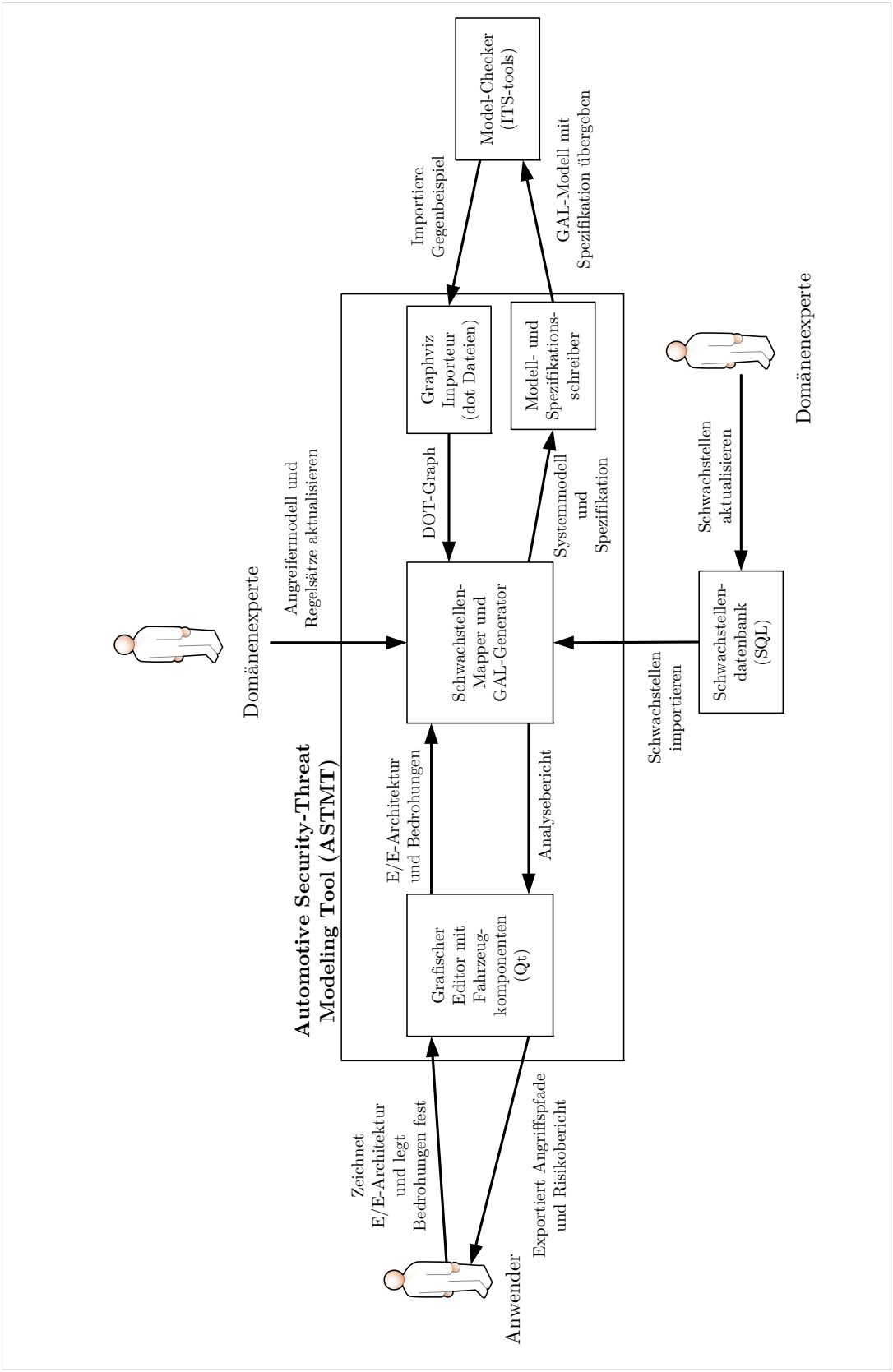


Abbildung E.5: Kontextdiagramm, welches aufzeigt wie externe Personen (Anwender und Domänen Experten) mit dem ASTMT agieren. Außerdem ist ein Überblick über interne und externe Elemente geben, die mit dem ASTMT Daten austauschen. Die externen Elemente sind hierbei die Structured Query Language (SQL)-Datenbank mit den Schwachstellen und der Model-Checker ITS-tools, welcher das Systemmodell validiert und das Gegenbeispiel erzeugt. Die dargestellten Pfeile beschreiben dabei in welche Richtung Informationen oder Daten zwischen zwei Elementen ausgetauscht werden.

Wie genau die einzelnen Bausteine des entwickelten ASTMT miteinander agieren und welche Technologie hierzu eingesetzt werden, zeigt der nächste Abschnitt mit der Softwarearchitektur des Werkzeuges.

E.1 Softwarearchitektur

Das ASTMT ist in der Sprache C++14 entwickelt und nutzt die Bibliotheken Boost C++ 1.69 sowie Qt in der Version 5.11.1. Letzteres dient primär zur Darstellung der Benutzeroberfläche und kommt ebenso für den ZodiacGraph von Clemens Sielaff [45] zum Einsatz, auf dem das ASTMT aufbaut. Die Oberfläche ist dreigeteilt und bietet dem Anwender auf der linken Seite eine Bibliothek für die Fahrzeugkomponenten sowie deren Kommunikationsverbindungen (Abbildung E.6).

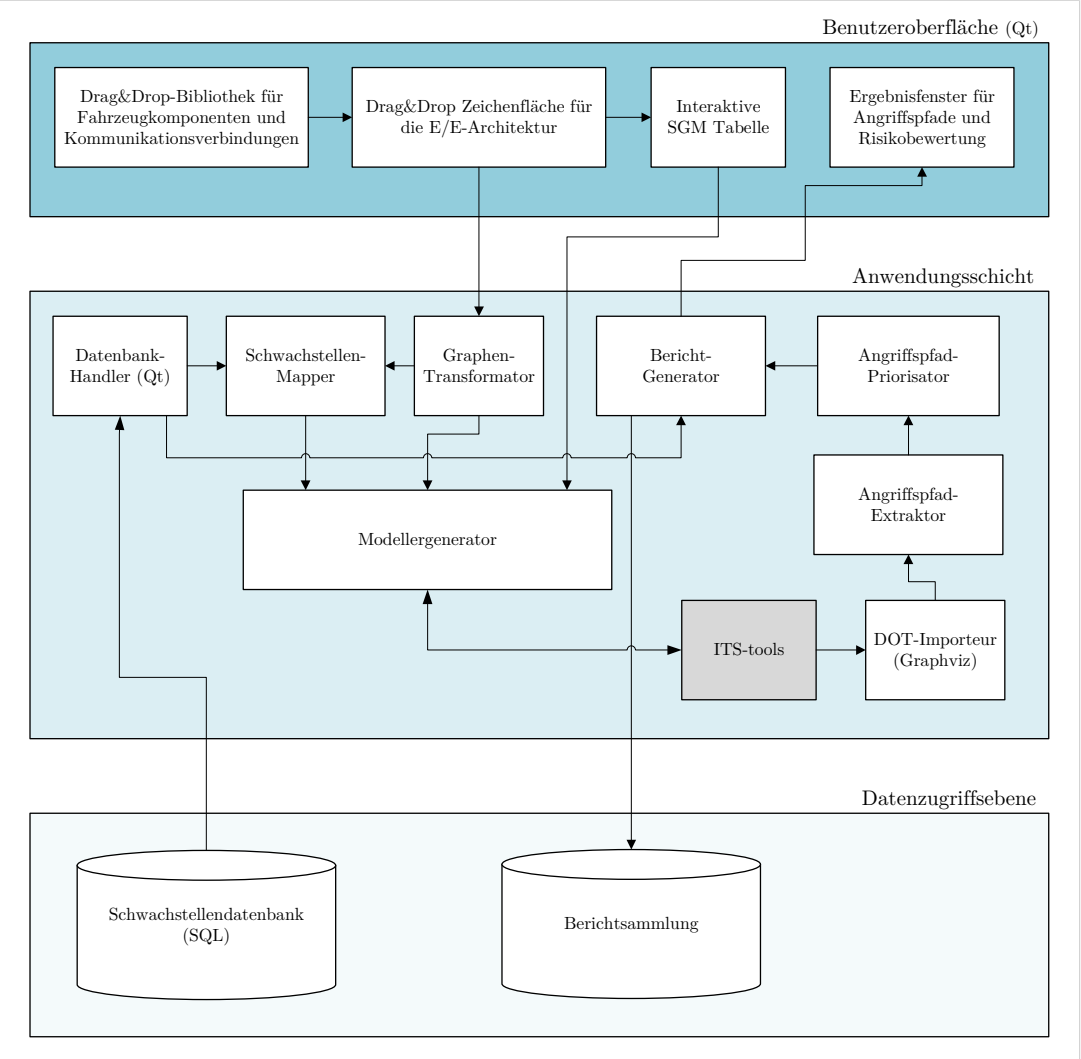


Abbildung E.6: Softwarearchitektur des ASTMT mit der Benutzeroberfläche, der Anwendungsschicht und der Datenzugriffsebene.

Jedes Element lässt sich dabei per Drag&Drop auf die Zeichenfläche ziehen, um die E/E-Architektur erstellen zu können. Außerdem zeigt die Oberfläche ein Eingabefenster für die SGM-Tabelle und das Ausgabefenster für die Analyseergebnisse. Hierbei ist die Nutzeroberfläche von der Datenverarbeitung getrennt. Die Softwarearchitektur folgt dabei einem Ebenenmuster [165, Seite 6], bei der die oberste Ebene durch die Benutzeroberfläche, die darunter liegende durch die Anwendungsschicht und die Datenhaltung als letzte Ebene in Abbildung E.6 repräsentiert ist.

Schicht zwei beinhaltet entsprechend die datenverarbeitenden Elemente. Sie empfängt von der darüber liegenden Schicht die in der Zeichenfläche abgelegten E/E-Architekturelemente, eine Adjazenzmatrix sowie die Einträge der SGM-Tabelle. Der Graphen-Transformator wandelt die ersten beiden Artefakte in einen gerichteten Graphen ($G = (U, E)$) um, damit die Suchalgorithmen angewendet werden können (Algorithmus 1). Hierbei repräsentieren die Knoten des Graphen die Komponenten U und die Kanten E die Kommunikationsverbindungen. Der *Schwachstellen-Mapper* übernimmt anschließend den Graphen und ordnet die passenden Schwachstellen aus der Datenbank zu. Hiermit entsteht ein Modell der E/E-Architektur mit applizierten Security-Schwachstellen, die ein Angreifer ausnutzen kann. Dieses und die Inhalte der interaktiven SGM-Tabelle werden vom Modellgenerator übernommen und das Systemmodel M sowie die Spezifikation φ erzeugt. Ersteres wird aus dem Modell der E/E-Architektur und φ aus den Inhalten der SGM-Tabelle erstellt (Tabelle 7.2).

Der Modellgenerator übergibt das erzeugte GAL-Modell an den Model-Checker ITS-Tools [202] und löst die Validierung des Modells aus. Wird ein Gegenbeispiel erzeugt, so liest der *DOT-Importeur* dieses als DOT-Graph ein und überführt es in einen gerichteten Graphen. Hierbei wird aus den Zuständen und den Zustandsübergängen im Gegenbeispiel ein Graph erzeugt, der die E/E-Architekturelemente, die ausgenutzten Schwachstellen sowie deren Eintrittswahrscheinlichkeiten vereint. Dieser wird anschließend an den *Angriffspfad-Extraktor* übergeben, welcher mit Algorithmus 1, die Angriffspfade generiert und deren Eintrittswahrscheinlichkeit nach Gleichung (7.1) bestimmt. Anknüpfend legt der *Angriffspfad-Priorisator* das Risiko jedes Angriffspfades nach Gleichung (7.3) fest und erzeugt eine absteigende Liste der Angriffspfade. Diese wird dem *Bericht-Generator* übergeben, welcher die konkrete Beschreibung der Schwachstelle zuordnet. Hiermit entsteht ein textueller Angriffspfad, der vom Security-Analysten nachvollzogen werden kann. Dieser wird zum einen an die Datenzugriffsebene in Abbildung E.6 übergeben, um den Bericht abzuspeichern. Zumanderen werden die Angriffspfade an die Benutzeroberfläche übergeben, die diese im *Ergebnisfenster* darstellt, womit der Analysevorgang abgeschlossen ist.

E.1.1 Parallelisierung der Analyse

Da die Identifikation der Angriffspfade aufwendig sein kann, wurde bereits erwähnt, dass der Vorgang parallelisiert ist, um die Laufzeit zu verbessern. Hierzu zeigt das UML-

Sequenzdiagramm in Abbildung E.7 das abstrahierte Vorgehen zur Parallelisierung. Zu Beginn zeichnet der Anwender die E/E-Architektur im Modellierungseditor (Benutzeroberfläche) und startet deren Analyse. Das ASTMT überführt nun die gezeichnete Architektur in den Graphen G. Im Folgenden wird G mittels einer Tiefensuche (Depth-first search (DFS)) reduziert, um nur jene Komponenten aufzuweisen, die in den Pfaden zwischen dem Eintrittspunkt und dem Angreiferziel liegen. Die hierfür modifizierte Tiefensuche, identifiziert alle Pfade die zwischen dem Eintrittspunkt und dem Angreiferziel existieren und formt damit G_R. Diesem Graphen werden anschließend die relevanten Schwachstellen zugeordnet.

Im nächsten Schritt wird in einer Schleife das Systemmodell M und die Spezifikation φ erzeugt. Diese wird anschließend mit dem Model-Checker überprüft, was dem Aufruf *Starte Model-Validierung* in Abbildung E.7 entspricht. Die ITS-tools [202] erzeugen nun ein Gegenbeispiel, falls die Spezifikation φ verletzt wurde. Dieses wird eingelesen, in den Graphen A_i überführt und anschließend transponiert (A_i^T). Die Sinnhaftigkeit für letzteres wurde mit Abbildung 7.9 erläutert. Hierbei wurde aufgezeigt, dass durch die Transponierung die ehemaligen Endzustände zu Startzuständen werden und damit eine Parallelisierung der Suche mit Algorithmus 1 möglich ist. Hierzu wird für jeden Startknoten in A_i^T eine eigenständige Programminstanz (Thread) ausgeführt, welche den Pfad zwischen dem Startknoten und dem Endknoten erforscht. Dies ist in Abbildung E.7 mit dem Rechteck *par*, hervorgehoben. Hat jede Programminstanz die Suche abgeschlossen, so ist der Graph A_i^T durchlaufen und die Suche abgeschlossen. Die aufgedeckten Pfade werden nun gespeichert und die nächste Bedrohung aus der SGM-Tabelle wird selektiert. Diese Prozedur wiederholt sich, bis alle Bedrohungen beziehungsweise Zeilen in der Tabelle abgearbeitet wurden. Ist dies der Fall, werden alle gespeicherten Angriffspfade zu einer Datensammlung zusammengefügt und die Risikowerte bestimmt. Sind die Risikowerte bestimmt, werden anknüpfend die Angriffspfade absteigend sortiert und der Analysebericht geniert. Dieser wird, wie Abbildung E.7 aufzeigt, anschließend vom Anwender abgerufen beziehungsweise im ASTMT angezeigt.

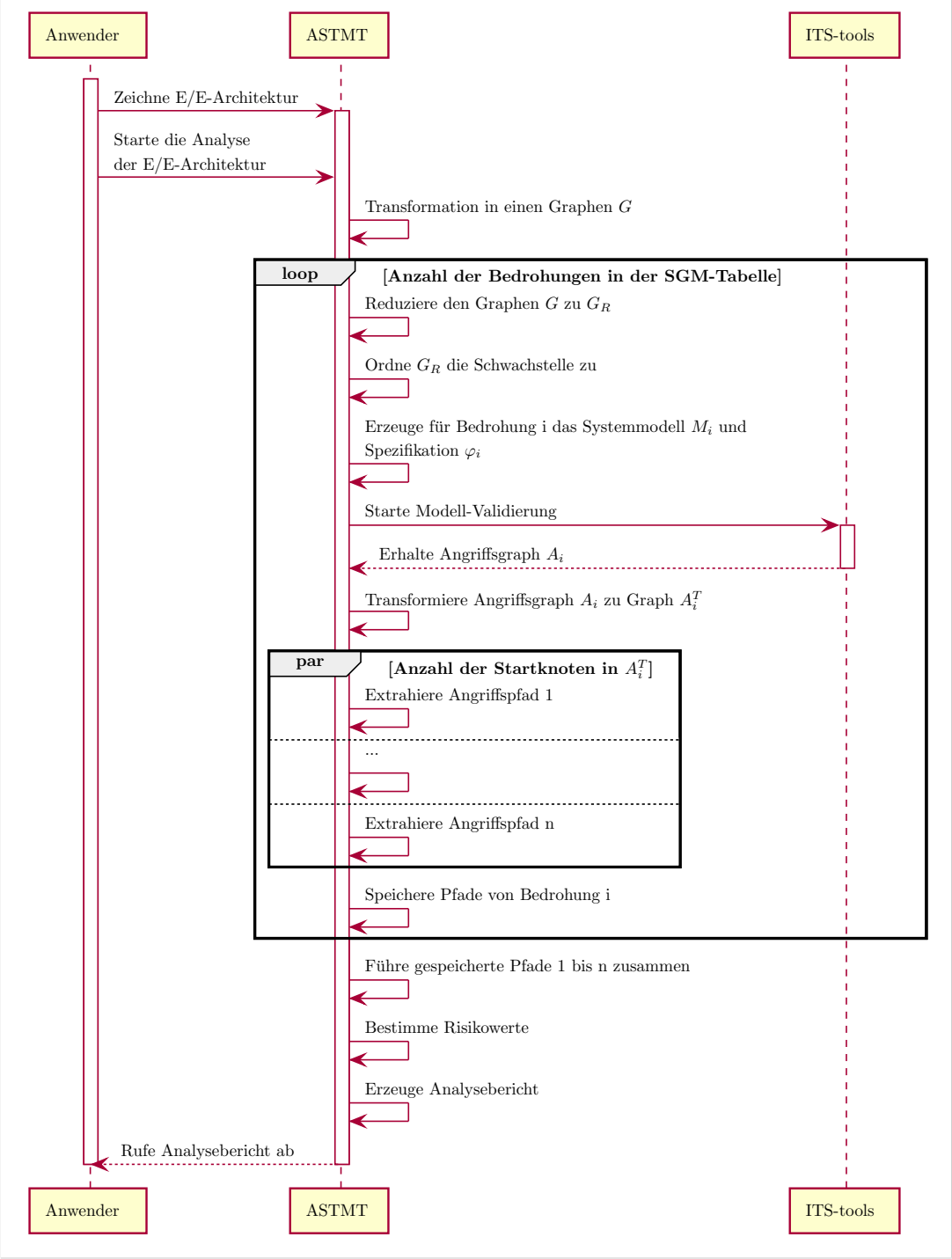


Abbildung E.7: UML-Sequenzdiagramm, das die Parallelisierung der Pfadsuche mit Algorithmus 1 beschreibt.

Der Quellcode des ASTMT steht unter [107] zur Verfügung.