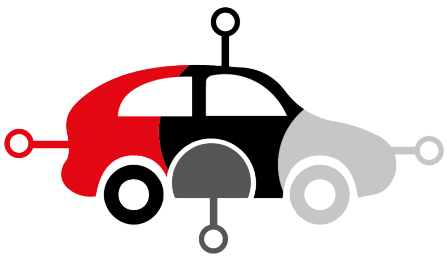
ERNW
providing security.

ABSCHLUSSBERICHT
ZUM TEILVORHABEN

Version:	1.0
Datum:	26. November 2021
Klassifikation:	Internal
Verbundprojekt:	DEvelopment For SEcured Autonomous Driving (DEFEnD)
Verbundpartner:	ERNW Enno Rey Netzwerke GmbH
Förderkennzeichen:	16KIS0887
Projektlaufzeit:	1. September 2018 - 31. Mai 2021
Autor(en):	Hannes Mohr



DEFEnD

DEvelopment For SEcured
Autonomous Driving

ERNW
providing security.

Inhaltsverzeichnis

ERNW
providing security.

Abbildungsverzeichnis

1 Gesamtüberblick

1.1 Aufgabenstellung

Das Ziel des Teilvorhabens im Projekt DEFEnD ist es, die Verknüpfung von IT-Security mit funktionalen Sicherheitsaspekten für autonomes Fahren zu ermöglichen.

Dabei wurde erforscht wie sich mit Hilfe von Metamodellen Security Aspekte frühzeitig und zuverlässig in Projekte einbringen lassen. Hierbei wurde in enger Kooperation mit den Projektpartnern eine Vereinigung von Security Anforderungen und Elektrik/Elektronik (E/E) Architekturen erarbeitet. Die beispielhafte Darstellung einiger Teilergebnisse wurde durch einen Demonstrator erbracht.

Speziell im Teilvorhaben wurden Methoden und Modelle entwickelt, die eine Einhaltung notwendiger Security-relevanter Vorkehrungen bereits innerhalb der Entwicklungsphase frühzeitig sicherstellen sollen. Von besonderem Interesse sind hierbei die definierten Schnittstellen und deren Kommunikation untereinander, sowie die in den einzelnen Komponenten verwendeten Technologien. Insbesondere wurde auf Grundlage der Erfahrung innerhalb der IT-Security Branche durch die ERNW eine detaillierte Analyse möglicher Angreiferszenarien und der sich ergebenden Angriffsvektoren erstellt werden.

In Kombination mit einer auf die notwendigen Technologien und Uses Cases abgestimmten Risikoanalyse leiten sich daraus technische Maßnahmen ab. Ebenso sollen die Vertrauensbeziehungen herausgestellt und, wo möglich, technisch abgesichert werden. Von speziellem Interesse ist hierbei die frühzeitige Evaluierung des zu erstellenden Designs und der damit verknüpften Angriffsoberfläche. Durch sorgfältige Planung dieses Designs, soll zum einen eine Verringerung des Entwicklungsaufwands erzielt werden, in der Hauptsache aber inhärenten Sicherheitsproblemen von Anfang an begegnet werden.

Ziel dieses Teilvorhabens im Projekt DEFEnD war es also, Konzepte für Security by Design zu erarbeiten, die in Security Engineering Prozessen möglichst zielführend und effizienzsteigernd eingesetzt werden können.

Dazu gehören die Verwendung standardisierter Vorgehen zur Bedrohungsmodellierung sowie die Erarbeitung eines Security Modells, das gängige Fragestellungen möglichst einfach beantwortbar macht. Dabei wurde eine enge Verzahnung des Modells mit typischen Entwurfsprozessen von Elektrik/Elektronik (kurz: E/E) Architekturen angestrebt.

1.2 Voraussetzungen

Das Projekt fand im Rahmen der Fördermaßnahme „KMU-innovativ: Informations- und Kommunikationstechnologien (IKT)“ statt. Die Partner aus KMUs und Forschungseinrichtungen haben innerhalb des Projekts die Förderschwerpunkte „Sichere und vertrauenswürdige IKT-Systeme“ sowie „IT-Sicherheit in Anwendungsfeldern“ vorangetrieben.

Dem Projekt stand dabei keine Architektur eines autonom fahrenden Fahrzeugs zur Verfügung, anhand derer eine Risikoanalyse hätte durchgeführt werden können, da derartige Architekturen noch nicht existieren.

Dadurch war es unter anderem notwendig den benötigten Detailgrad zu erarbeiten, der für eine sinnvolle Analyse eines solchen Systems notwendig ist.

1.3 Planung und Ablauf

Das Gesamtprojekt war in die folgenden Arbeitspakete unterteilt:

- AP 1: Szenarien und Security-Anforderungen für autonomes Fahren
- AP 2: Security-Requirements-Engineering und Auswirkungsanalyse
- AP 3: Methoden und Werkzeuge
    - AP 3.1: Security by Design – Modellbasierte Methoden und Konzepte
    - AP 3.2: Methoden und Prozesse für Security-Validierung und -Test
    - AP 3.3: Tooling für Security-Methoden und -Konzepte
- AP 4: Evaluation und Demonstratoren
    - AP 4.1: Implementierung des Demonstrators
    - AP 4.2: Validierung der Projektergebnisse
- AP 5: Analyse, Modellierung und Überführung rechtlich-technischer Vorgaben

ERNW hat zu den Arbeitspaketen AP 1, AP 2 und AP 4 unterstützend beigetragen. Das Unterarbeitspaket 3.1 wurde von ERNW geleitet. An AP 5 hatte ERNW keine direkte Beteiligung.

Wegen Verzögerungen auf Verbundsebene, kam es zu einer Projektverzögerung von etwa 3 Monaten. Das Projekt wurde kostenneutral zum 31ten Mai 2021 verlängert.

1.4 Ausgangssituation, verwendete Standards und Quellen

Während des Projekts sind insbesondere folgende Standards berücksichtigt worden:

- SAE J3061 (2016) – „Cybersecurity Guidebook for Cyber-Physical Vehicle Systems"
- SAE AS5506C (2017) – „Architecture Analysis & Design Language (AADL)"
- ISO 26262:2018 – „Road vehicles — Functional safety"
- ISO/SAE DIS 21434:2020 – „Road vehicles — Cybersecurity engineering"

Weitere verwendete Quellen werden im Anhang genannt.

Die wissenschaftliche Ausgangssituation wird im Gesamtbericht detailliert beschrieben.

1.5 Zusammenarbeit mit anderen Stellen

Während des gesamten Projekts wurde in stetigem Austausch mit den Konsortialpartner zusammengearbeitet. Darüber hinaus fand keine Zusammenarbeit mit weiteren externen Stellen im Projektkontext statt.

1.6 Grundlagen und Einordnung

Sicherheitsbetrachtungen im Sinne von Safety sind in der Regel funktional motiviert. Der zufällige Defekt oder ein Ausfall stehen im Mittelpunkt der Betrachtung. Security-relevante Angriffe hingegen haben im Allgemeinen einen motivierten, gezielten Angreifer als Ausgangspunkt, der willentlich einen Defekt herbeiführt oder einen nicht antizipierten Zustand des Systems ausnutzt. Bei einem derartigen Angriff werden oftmals nicht vorhergesehene Zustände herbeigeführt, die am gewünschten, funktionalen Verhalten vorbeiarbeiten und einen inneren Umstand oder Ablauf der zugrundeliegenden Technik geschickt ausnutzen.

Eine rein auf Funktionen und Logik beruhende Analyse, wie in Safetybetrachtungen eher üblich, ist daher nicht zielführend. Ebenso ist eine Einschätzung der Wahrscheinlichkeit eines bestimmten Angriffs nicht quantitativ messbar. Es kann lediglich eine qualitative Aussage über die Komplexität eines Angriffs getroffen werden. Aber während die Entdeckung und erstmalige Ausnutzung eines Angriffs aufgrund seiner Komplexität als mehr oder weniger schwierig klassifiziert werden kann, kann durch das Bekanntwerden von Schadcode, der für eine bestimmte Schwachstelle entwickelt wurde, der Angriff wiederum trivial nachzubilden sein. Eine statistische Einschätzung der Eintrittswahrscheinlichkeit, beispielsweise durch Versuchsreihen, die die Wiederholungen bis zum Eintritt eines unerwünschten Ereignisses messen, sind daher im Security Bereich nicht analog wie zur Safety durchführbar.

Vor dem Hintergrund des autonomen Fahrens ist der potentielle Verlust der Kontrolle über das Fahrzeug, beziehungsweise des sicheren Betriebs von höchster Bedeutung. Dadurch ergeben sich zwei zentrale Fragestellungen bei der Verknüpfung von Safety und Security.

Es muss zum einen weitestgehend sichergestellt werden, dass Security Schwachstellen keinen negativen Einfluss auf den sicheren Betrieb des Fahrzeugs nehmen. Zum anderen muss Sorge getragen werden, dass Security Maßnahmen keinen negativen Einfluss auf Randbedingungen aus Safety Überlegungen haben.

Korrespondierend zum Safety-by-Design Konzept (SDC) wird daher ein Safety-and-Security-by-Design Konzept (SSDC) angestrebt. Dem Design werden entsprechend Methoden beigefügt, mit denen die erreichten Level bezüglich Safety und Security bestmöglich verifiziert und somit validiert werden sollen.

Dieses Teilarbeitspaket befasst sich mit der Entwicklung von Metamodellen zur Beschreibung von autonomen Fahrfunktionen, E/E Architekturen und Fahrzeugsystemen sowie der Aufstellung, Modellierung, Analyse und Bewertung von Angreiferszenarien und Angriffsvektoren.

Damit soll die Grundlage für eine nachverfolgbare, semantische Verknüpfung zur Erkennung von Angriffen, Zustandsüberwachung, Verschlüsselung/Authentifizierung und Plausibilisierung unterschiedlicher Informationsquellen implementierbar werden.

ERNW
providing security.

2 Ergebnisse AP 1

Die ERNW hat während des Arbeitspakets eine beratendende Funktion mit Blick auf die betrachteten Use Cases eingenommen um die Sicherheitsperspektive in die funktionalen Betrachtungen der Partner frühzeitig mit einfließen zu lassen.

3 Ergebnisse AP 2

In diesem Arbeitspaket hat die ERNW beratend mitgewirkt um die Erfahrungen aus dem Security Bereich in die Betrachtungen der Use Cases einfließen zu lassen.

Es wurden Analysen anhand bestehender gängiger Safety und Security Modelle auf die Use Caess innheralb des Toolings der Partner durchgeführt.

Zudem wurde der notwendige Detailgrad der Betrachtungen erstmals herausgearbeitet und in Arbeitspaket 3 präzisiert.

4 Ergebnisse AP 3

An der direkten Entwicklung der Werkzeuge dieses Arbeitspakets hat die ERNW keinen Anteil.

Die ERNW hat das Arbeitspaket *Security by Design – Modellbasierte Methoden und Konzepte* geleitet.

Hierbei wurden grundlegende Methoden zur Verknüpfung von Safety und Security mit Bezug zu den erabeiteten Use Cases entworfenen. Wo möglich wurde auf bestehende Mehthoden zurückgegriffen, beziehungsweise wurden diese entsprechend erweitert.

5 Ergebnisse AP 4

An diesem Arbeitspaket war die ERNW lediglich unterstützend tätig.

Speziell wurde an der Untersuchung der Datenflüsse und der Konzeptionierung der Beispiele mit Bezug zur technischen Umsetzung der Plattform mitgearbeitet.

6 Ergebnisse AP 5

An diesem Arbeitspaket hatte die ERNW keine aktive Beteiligung.

7 Detaillierte Teilergebnisse AP 3

Wichtige Teilergebnisse werden im Folgenden detaillierter beschrieben.

7.1 E/E Architekturen

Die Komplexität von E/E Architekturen hat über die letzten Jahre und Jahrzehnte stark zugenommen und wird durch die Einführung neuer Funktionen, wie autonomem Fahren weiter zunehmen. Entsprechend wachsen auch die verknüpften Anforderungen an das Design, die Entwicklung und Risikobewertungen dieser Architekturen.

7.1.1 Beispielarchitekturen und Beispielangriffe

Da derzeit kein autonomes Fahrzeug in Marktreife existiert dessen Architektur als Grundlage dienen könnte, wurden anhand publizierter Angriffe einige beispielhafte Architekturen skizziert. Eine dieser Architekturen ist in Abbildung 1 zu sehen.

Die gezeigte Architektur wurde innerhalb des FZI-Tools modelliert. Anhand dessen wurden mögliche Angriffsvektoren innerhalb des Tools erstellt und bewertet.

7.1.2 Angriffspfade basierend auf Beispielarchitektur

Im folgenden werden anhand der Komponenten der Beispielarchitektur zwei Möglichkeiten für Angriffe mit hohem Schadenspotential gezeigt.

In beiden Beispielen wird als Endziel der besonders Safety-kritische CAN-BUS mit seinen ECUs verwendet. Das Abschicken beliebiger Nachrichten auf diesem CAN-BUS hat ein hohes Schadenspotential für Insassen und andere Verkehrsteilnehmer.

In beiden hier gezeigten Fällen sind die Verkettungen von Schwachstellen die ausgenutzt werden relativ komplex. Durch die unterschiedlichen initialen Eintrittsvektoren mittels USB Schnittstelle und GSM ergeben sich mittels CVSS Version 3.1 (siehe [6]) als zugrundeliegender Metrik, eine entsprechend unterschiedlich hohe Kritikalität.

Nichtlokaler Angriff (GSM)

(Fake) Base Station $\xrightarrow[\text{GSM}]{\text{RCE}}$ TCU $\xrightarrow{\text{uds message}}$ $\Rightarrow$ CGM $\xrightarrow[\text{vulnerability}]{\text{Cross CAN}}$ GCM $\xrightarrow{\text{arbitrary message}}$ $\Rightarrow$ ECU

Eingangsangriff: Zum Beispiel [10] und [12]

(Base Score: 9.8 CRITICALVector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H)

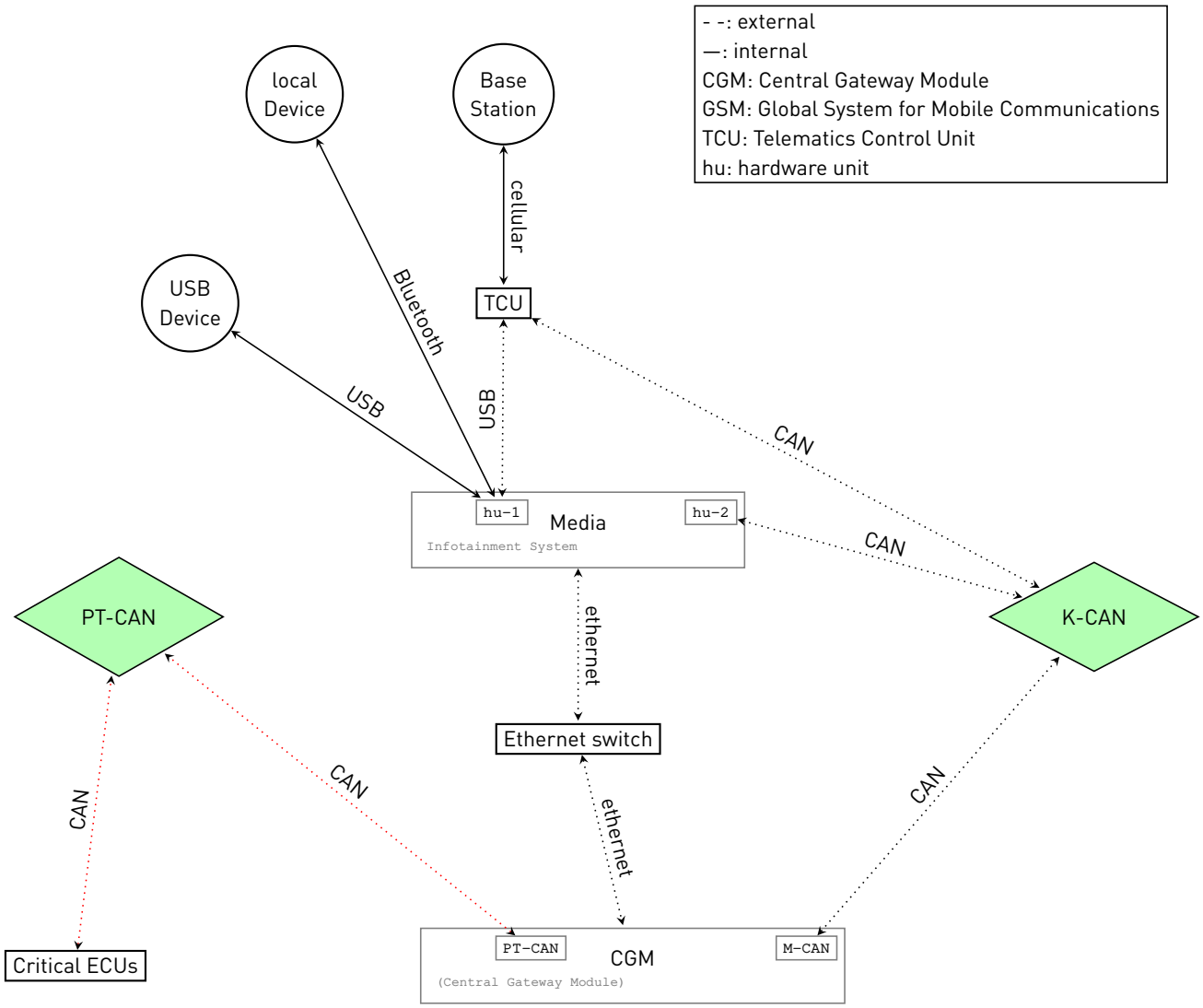
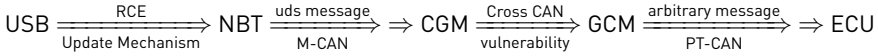


Abbildung 1: Beispielarchitektur mit ausreichend Komplexität, um realistische Angriffsszenarien zu untersuchen, analog zu [9]. Subkomponenten wie Software auf hu-1, beispielsweise Webbrowser, sind nicht mit abgebildet.

Lokaler Angriff (USB)



Eingangsangriff: Zum Beispiel [11] und [13]

(Base Score: 7.8 HIGHVector: CVSS:3.0/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H)

7.1.3 Risikobewertung

Um die begrenzten Ressourcen, die zur Absicherung eines Systems aufgebracht werden können möglichst sinnvoll zu verwenden, sollten iterativ Risikobetrachtungen durchgeführt werden. Diese helfen besonders kritische Komponenten zu identifizieren und Maßnahmen zu identifizieren und zu implementieren, die die Gesamtsicherheit des Systems verbessern. Es existiert eine Vielzahl etablierter Methoden diese Analysen durchzuführen, auf die hier nicht näher eingegangen wird.

Im Fokus der Betrachtung soll ein allgemeines von speziellen Metriken unabhängiges Vorgehen erarbeitet werden. Es soll Safety und Security Beziehungen berücksichtigen und miteinander verbinden.

Die Kriterien an die Betrachtung sind deckungsgleich mit denen einer regulären Risikoanalyse, wie sie beispielsweise in [4] dargestellt wird, mitsamt der üblichen Akteure, wie sie im folgenden kurz aufgeführt werden.

Begriffe

In der vorgestellten Methodik werden die folgenden allgemeingültigen Begrifflichkeiten verwendet:

`Betreiber (Owner)` betreut, entwickelt und verwaltet Komponenten.

`Gegenmaßnahmen (Countermeasures)` werden ergriffen um Schwachstellen zu entgegnen und Risiken zu minimieren

`Risiken (Risks)` Setzen sich aus Eintrittswahrscheinlichkeit und Schadenspotential zusammen

`Schwachstellen (Vulnerabilities)` können ausgenutzt werden

`Bedrohungen (Threats)` Abstrakte Schadensfallklasse

`Angreifer (Attackers)` unbekannte Dritte die Interesse haben einen Schaden auszulösen

`Angriffsvektoren (Threat Vectors)`

`Komponenten (Assets)` schützenswerter Betrachtungsgegenstand einer Analyse

7.1.4 Verknüpfung von Risikobewertungen mit Safetyanforderungen

Safety Anforderungen konnten im Wesentlichen aus zwei Motivationen heraus verwendet werden.

Zum einen dienen sie der Identifikation und *Schärfung der Ziele* möglicher Security Angriffe. Hierbei werden besonders kritische Komponenten, die aus Safety Sicht eine hohe Relevanz haben, als besonders schützensswert identifiziert. Dies stellt die positive Verknüpfung von Safety und Security dar.

Zum anderen müssen definierte Maßnahmen, die aus Security Überlegungen stammen, einem *Abgleich gegen Safety Rahmenbedingungen* unterzogen werden. Einige Security Bedingungen, wie beispielsweise Echtzeitverhalten kritischer Fahrfunktionen können im Gegensatz zu Security Anforderungen stehen. Hier muss im Einzelfall entschieden werden, ob und wie ein derartiger Konflikt behoben werden kann. Unterstützendes Tooling sollte daher im Stande sein, die aufgedeckten Probleme zu dokumentieren und greifbar zu machen.

Allgemein sollte das Tooling die folgenden Eigenschaften erfüllen:

- Abbildung von Schutzgegenständen
- Verknüpfung von Schutzbedarf und Schutzgegenstand
- *Verknüpfung von Safety und Security*
- Abbildung von Datenflüssen
- Abbildung von Maßnahmen
- Klassifizierung und Tracking von Maßnahmen
- Abbildung von Bedrohungen und Angriffsvektoren
- Klassifizierung von Bedrohungen und Angriffsfaden

Das Zusammenspiel dieser Anforderungen wird im Folgenden abstrakt beschrieben. Eine Umsetzung dieser Voraussetzungen ist im FZI-Tool, dem KIT-Tool, sowie Coodex in verschiedenen Ausprägungen realisiert.

7.1.5 Metamodell - Durchführung Risikoanalyse und Angriffspfade

Der Ablauf der generalisierten Methodik wird im Folgenden anhand der Abbildung 2 dargestellt. Die einzelnen Schritte, in der Grafik mit (n) annotiert, werden in der unten stehenden Aufzählung erläutert. Dieser Prozess ist iterativ und begleitend zur Entwicklung durchführbar.

Die Architektur sollte sich innerhalb der Entwicklung verfestigen, aber auch nachdem ein Fahrzeug in Prduktion geht, muss beispielsweise anhand neuer Erkenntnisse (wie dem Bekanntwerden neuer Schwachstellen einzelner Komponenten) oder auch Veränderungen der Funktionen durch Softwareupdates, eine Neubewertung stattfinden.

(0): Aufstellung von Safetybetrachtungen und Maßnahmen anhand bekannter Methoden, wie beispielsweise, HARA [7], TARA [8] und FADEC

(1): Die Architektur gibt allgemein die Senken der Risikobetrachtung vor, da sie alle Komponenten enthält.

(1'): Aus den Safetybetrachtungen und Modellierungen heraus, können diejenigen Komponenten identifiziert werden, die eine besonders kritische Funktion inne haben. Durch eine derartige Gewichtung dieser Komponenten, kann ein Safety-motivierter Betrachtungsschwerpunkt gesetzt werden der zur Absicherung des Gesamtsystems beiträgt.

(2): Die Architektur gibt die Quellen, also den Ursprung möglicher Angriffe vor. Dies gilt sowohl für externe Schnittstellen, wie etwa Funkschnittstellen aber auch interne Schnittstellen, wie etwa Debugports.

(3): Die Senken definieren über ihre Kritikalität, beispielsweise anhand ihrer Vertraulichkeits-, Integritäts- und Verfügbarkeitsanforderungen, verknüpft mit den Informationen aus den Safetybetrachtung die Gewichtung der einzelnen Pfade. Ein Angriffspfad mit einem hochkritischen Ziel, also mit entsprechend hohem Schadenspotential, erlangt somit eine größere Bedeutung in der Betrachtung.

(3'): Die Quellen definieren, anhand der Art der Quelle, also der Exposition des jeweiligen Startpunkts eines möglichen Angriffs, die Gewichtung der verknüpften Angriffspfade.

(4): Die Gesamtheit aller Quellen, Senken, Komponenten und Datenflüsse definiert den Graphen aller möglichen Angriffspfade

(5): Die Gewichtung der einzelnen der Quellen und Senken wird innerhalb der Graphen annotiert, dadurch kann eine Abschätzung beispielsweise der Komplexität des Angriffs oder seines Schadenpotentials generiert werden. Diese Informationen dienen zur Klassifizierung von Angriffspfaden.

(6): Aus den enstandenen Graphen lassen sich nun Angriffspfade extrahieren. Diese können mit Hilfe der annotierten Gewichtungen klassifiziert und anhand einer austauschbaren Metrik bewertet und sortiert werden.

(7): Für einzelne Angriffspfade, beziehungsweise assoziierte Risiken, können nun Maßnahmen definiert werden. Maßnahmen können sowohl architektureller Natur sein, aber auch auf Komponenten- oder Datenflussebene angesiedelt sein.

(8): Die Maßnahmen nehmen nach ihrer Implementierung wiederum Einfluss auf die Bewertung der identifizierten Pfade.

(9): Die Maßnahmen werden nach Implementierung ebenfalls wieder in im Modell der Architektur modelliert werden. Dies dient auch dazu mögliche negative Effekte der Maßnahmen im Bezug auf Safety Anforderungen zu erkennen.

Bei Befolgung dieses Prozesses ergeben sich aufgrund der Kommunikationsbeziehungen, der festgestellten Datenquellen, sowie der identifizierten Senken, auf natürliche Weise Schutzzonen. Die Schutzzonen ergeben sich zudem aus Überlegungen die die Vertrauenswürdigkeit der Komponenten, die Kritikalität ihrer Funktion, sowie die notwendigen Kommunikationsbeziehungen und sonstige technische Rahmenbedingungen mit einbeziehen.

Eine Methode zur Annotation der Daten, je nach Stärke und Art ihres Schutzbedarfs ist im Abschnitt 7.2 sowie den zugrundeliegenden Veröffentlichungen ([3],[2]) beschrieben.

Die resultierenden, identifizierten Angriffspfade können innerhalb des Entwicklungsprozesses verwendet werden, um beispielsweise innerhalb eines Secure Development Lifecycles (SDC), als Quality Gate zu dienen. Somit kann beispiels-

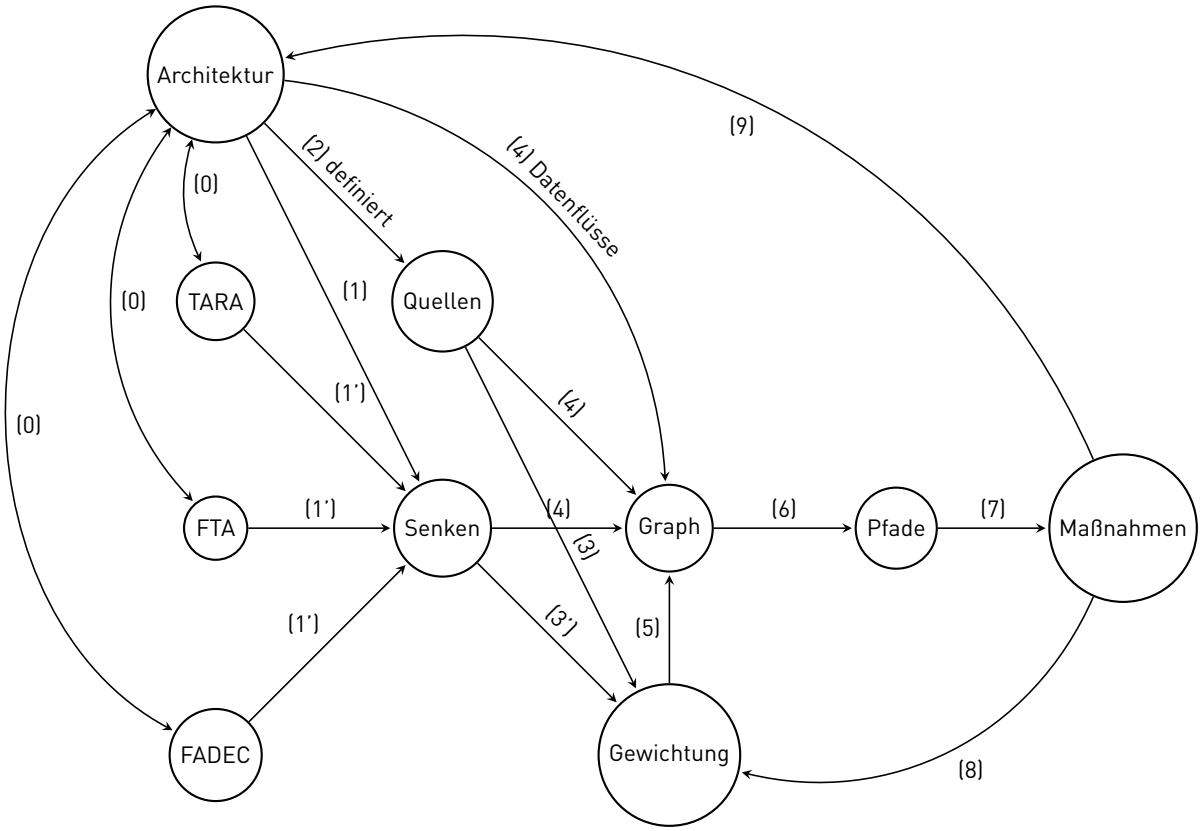


Abbildung 2: Metamodell zum Ablauf der Riskobewertung mitsamt Aufstellung von Angriffspfaden und Einführung von Maßnahmen.
Quellen entsprechen Angriffszielen, Senken entsprechen Startpunkten von Angriffen.

weise verlangt werden, dass keine hochkritische Schwachstelle existiert, die einen Angriffspfad von einer externen Quelle, hin zu einer kritischen Senke erlaubt.

Bekannt gewordene Schwachstellen können an einzelnen Komponenten annotiert werden, um Pfade entsprechend neu zu gewichten. Sollte eine Möglichkeit der Kontrolle beispielsweise eines zentralen Gateways durch einen Angriff bekannt werden, kann dieser Angriff mit einbezogen werden.

Dadurch erfüllt das Modell eine begleitende Funktion während des V-Modells ([16]).

Eine Methode zur Synthetisierung von Angriffsvektoren ausgehend von der Fault Tree Analysis (FTA) wird in Abschnitt 7.3 beschrieben.

Modellierung und Bewertung von Bedrohungen und Risiken

Die einzige Bedingung an die Bewertungsmethodik ist, dass die Methodik eine Metrik beinhaltet, die es erlaubt eine Priorisierung der einzelnen Bedrohungen durchzuführen. Es existiert eine Vielzahl gängiger Methoden, die jeweils mit eigenen Vor- und Nachteilen behaftet sind. Für eine kurze, unvollständige Auflistung dieser Methoden siehe 9.2.2.

Innerhalb des erarbeiteten Toolings des Partners wurden mehrere Metriken implementiert.

Maßnahmen

Eine begleitende Methodik eines Secure Development Lifecycles, sollte es entsprechend erlauben, Maßnahmen festzulegen und diese Maßnahmen zu katalogisieren. Diese Maßnahmen können grundsätzliche Leitlinien sein, wie beispielsweise die im Microsoft SDL genannten (siehe 9.1.1) oder die von Security Maßnahmen mit Fokus auf Hardware die von OWASP gesammelt werden, siehe 9.1.2.

Je nach Technologie und Funktion müssen umfangreiche Listen erstellt werden, die auf einzelne Bedrohungen eingehen. Ein Beispiel für eine solche Liste von Hardware Security Bedrohungen, verknüpft mit entsprechenden Maßnahmen, wurde beispielsweise durch die ENISA (in [4]) veröffentlicht. Ein Beispiel für eine stärker spezialisierte Liste mit Fokus auf Supply Chain Angriffen, lässt sich in [5] finden.

Um einzelne Bedrohungen zu adressieren müssen entsprechende Maßnahmen getroffen werden. Die Befolgung derartiger Maßnahmen die sich an *Security Best Practices* orientieren, ist für eine sichere Implentierung essentiell.

Ein Prozess zur Risikoanalyse sollte derartige Maßnahmen nachvollziehbar und auditierbar abbilden. Es sollte außerdem möglich sein, die mit den Maßnahmen veknüpften Annahmen kenntlich zu machen. Dies soll sicherstellen, dass bei inzwischen unzutreffenden Annahmen auch die Wirkung der Maßnahme als hinfällig erkannt werden kann. Bei zunehmender Komplexität der Systeme ist eine Standartisierung dieses Vorgehens in aller Regel notwendig. Hierduch soll insbesondere Nachhvollziehbarkeit gewährleistet werden. So kann zum Beispiel die Verwendung eines bestimmten Verschlüsselungsverfahren mit einer bestimmten Schlüsselänge, nur für eine bestimmte Dauer als ausreichend sicher angenommen werden.

Eine derartige Katalogisierung von Maßnahmen wurde beispielhaft innerhalb des Toolings des FZI implementiert.

7.1.6 Begrenzung von Datenfüssen als Maßnahme

Aus der initialen Betrachtung der funktionalen Bedingungen der frühen Designphasen ergeben sich die notwendigen Datenflüsse. Eine allgemeingültige Grundregel bei der Konzeption von Systemen mit Security als Designziel ist die Minimierung von Datenflüssen.

Es ist somit notwendig dafür zu sorgen, dass Komponenten oder Teilsysteme nur über zwingend erforderliche Kommunikationsschnittstellen verfügen. Als Beispiel darf in Abbildung 1 keine direkte Kommunikation zwischen dem Infotainment System und Systemen im PowerdriveTrain (PT) CAN Netzwerk bestehen. Die Abschottung der Systeme zueinander sollte dabei so grundlegendend wie möglich umgesetzt werden. Für die Entscheidung, wann Komponenten voneinander zu trennen sind, sind *Schutzzonen* zu definieren, die sich aus der *Risikobewertung* der einzelnen Komponenten ergeben.

Im Fall der Beispielarchitektur, muss also beachtet werden, dass das Infotainment System eine hohe Zahl stark exponierter Komponenten beinhaltet. Es existiert eine Bluetooth, USB und auch Telekommunikationsschnittstelle. Ebenfalls ist die Komplexität der für die Bedienung verwendeten Komponenten äußerst hoch. Eine große Anzahl an Drittthersteller Chipsätzen sowie komplexe Softwarekomponenten wie Webbrowser, die absehbar ebenfalls von Drittherstellern stammen, werden hier typischerweise eingesetzt. Es ist daher davon auszugehen, dass ein motivierter Angreifer Schwachstellen in diesen Komponenten finden wird.

Somit ist es in diesem Fall leicht ersichtlich, dass Infotainmentkomponenten und kritische ECUs zu verschiedenen Schutzzonen gehören.

Es gibt andererseits Funktionen, wie die Lautstärke der Audiowiedergabe vom Geräuschpegel des Motors und damit der Drehzahl abhängen. Derartige Datenflüsse müssen in entwiclungsbegleitendem Tooling modellierbar sein.

Dieses Prinzip gilt auch auf Komponentenebene, wie im Beispiel des Demonstrators gezeigt wird. Verwendet eine Komponente zur Hinderniserkennung die Umrisse von Passanten, kann es sinnvoll sein, nur Daten mit verpixelten Gesichtern oder eben nur die Umrisse an diese Komponente zu übergeben. Dies entspricht dem Prinzip der Datensparsamkeit, und kann helfen Privacy Probleme bei komplexen, schwerer abzusichernden Komponente zu verhindern.

Innerhalb des Projekts wurde eine generative, architekturunabhängige Methode zur Synthetisierung von Konfigurationen entwickelt, die genau diese Eigenschaften erfüllet. Sie wird im folgenden Abschnitt (Abschnitt 7.2) beschrieben.

7.2 Methode zur generativen Datenflussminimierung und Datenabsicherung

Während des Projekts wurde in Kooperation mit den Konsortialpartnern ein Tool entwickelt, das ausgehend von einer logischen Beschreibung von Datenflüssen für eine gegebene Architektur Maßnahmen implementiert, die zu einer Minimierung der Datenflüssen führen.

Das Tool ermöglicht eine abstrakte Darstellung der Datenflüsse und der implizit notwendigen Befugnisse von Einzelkomponenten. Dem Einsatz des Tools, sollte zuvor also eine Optimierung der Funktionen mit Blick auf die Minimierung der Abläufe, der Trennung von Systemen, und ähnlichen Grundprinzipien vorangehen. Die tatsächliche Hardwareplattform, im Sinne der verwendeten Chipsätze und Plattformen bleibt in dieser Designphase offen. Dadurch wird sichergestellt, dass bei einer Migration auf eine neue Plattform, die die gleichen logischen Funktionen erfüllt, diese initiale

Bewertung gültig bleibt. Das Tool bietet anschließend die Möglichkeit in einem weiteren Schritt Konfigurationen zu generieren, die die Einhaltung des definierten, minimalen Datenflusses bestmöglich einhalten.

Es kann durch Beschränkungen der Plattform dazu kommen, dass Abweichungen vom definierten Minimalzustand auftreten. In diesem Fall wird ein Minimum der Abweichung gefunden. Ebenso kann es sein, dass die gewählten Randbedingungen nicht erfüllbar sind.

Durch diese Eigenschaft wird die Möglichkeit geschaffen werden, verschiedene Hardwareplattformen zu vergleichen.

Während im Demonstrator die Security relevanten Regeln für eine bestimmte Familie von Chipsätzen synthetisiert wurden, ist die Implementierung weiterer Konfigurationsgeneratoren denkbar. Das Tool könnte erweitert werden, so dass es Gateway- und Firewall-Konfigurationen erstellt. Es entspricht somit dem *Compliance as Code* Ansatz.

Ein Überblick über die Methodologie des vorgeschlagenen Design Prozesses ist in Abbildung 4 gegeben.

7.2.1 Detailbeschreibung

Für diesen Meilenstein wurde im Rahmen von AP3.1 von Seiten des KIT ein erstes formales Modell zur Beschreibung

- einer zur betrachtenden E/E-Architektur,
- die diesbezüglich relevanten Informationsflussanforderungen und
- die angedachte Implementierung dieser speziellen Informationsflussanforderungen

entwickelt. Instanzen dieses Modells können durch die Definition mathematischer Mengen, Relationen und Funktionen beschrieben werden. Um die Verifikation zur unterstützen, die insbesondere, aber nicht ausschließlich zur Erfüllung der der ISO-26262-Anforderungen nötig ist, ist ein Verfahren zur automatischen Konsistenzprüfung dieser Modellinstanz definiert worden. Konkret geprüft wird dabei die Konsistenz zwischen den Informationsflussanforderungen und der angedachten Implementierung auf der konkreten E/E-Architektur. Hierbei muss sichergestellt werden, dass

1. jeder Informationsfluss, der gemäß Informationsflussanforderungen notwendig ist, durch die angedachte Implementierung auch ermöglicht wird,
2. kein Informationsfluss, der gemäß Informationsflussanforderungen weder notwendig noch akzeptabel ist, stattfinden kann.

Der erste Punkt ist eine notwendige Bedingung zur Erfüllung der Nominalfunktionalität, hat allerdings auch aus Safetyperspektive eine gewisse Relevanz. So muss zum Beispiel sichergestellt sein, dass nach ISO 26262 integrierte Sicherheitsmechanismen zur Ausübung aller ihrer geforderten Interaktionen mit dem System in der Lage sind und nicht fälschlicherweise eine logische Isolation stattfindet. Der zweite Punkt hat eine unmittelbare Relevanz aus Safety- und Securitysicht.

Um diese Konsistenzprüfung durchzuführen, erfolgt eine Transformation der Modellinstanz in zwei dedizierte Graphen (G_α und G_β). Grundsätzlich repräsentieren Knoten hierbei die Inputs und Outputs der sogenannten Features, die eine bestimmte Funktion repräsentieren. G_β verfügt zusätzlich über Hilfsknoten, die als Teil der Analyse eingefügt werden. Konsistenzanforderung (1) ist genau dann erfüllt, wenn G_α einen gerichteten Pfad von jeder Quelle zur Senke eines geforderten Informationsflusses enthält. Konsistenzanforderung (2) ist genau dann erfüllt, wenn jeder Pfad von einer möglichen Informationsflussquelle zu einer -senke einem geforderten oder akzeptierten Informationsfluss (gemäß Informationsflussanforderungen) entspricht.

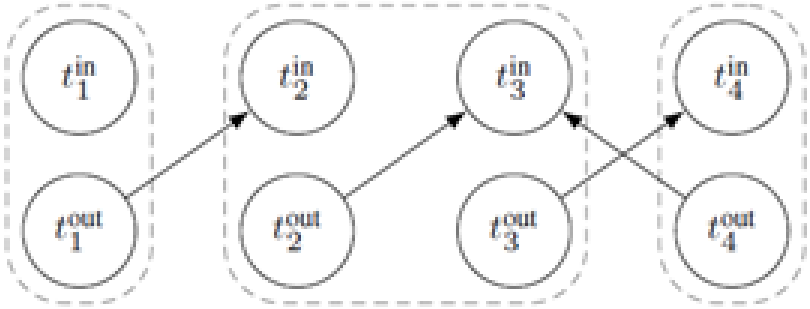


Abbildung 3: G_α für eine beispielhafte Modellinstanz.

Abbildung 3 zeigt ein beispielhaftes G_α für ein Modell, das nominale Informationsflüsse von t_1 nach t_2, von t_2 nach t_3, von t_3 nach t_4 sowie von t_4 nach t_3 erlaubt.

Logische Schutzzonen, ergeben sich aus diesem Model implizit und sind im Schaubild angedeutet.

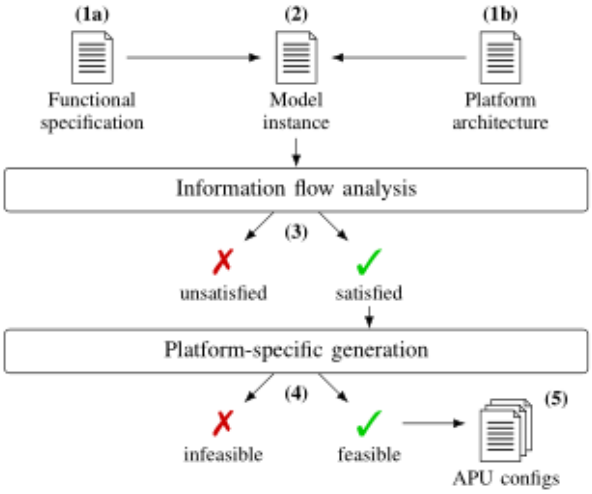


Abbildung 4: Design Methodologie - Konfigurationssynthetisierung

7.3 Methode zur Verknüpfung Fehlerbäumen und Angriffsvektoren

Die Ereignisse des Fehlerbaums, Beispiel für Fahrtenabwicklung, siehe Abbildung 5, werden daraufhin untersucht, ob sie Security-relevant sind. Dies beinhaltet zum einen die Frage ob sie durch einen Security Vorfall eintreffen können (mechanisches Versagen fällt hier beispielsweise in aller Regel weg) zum anderen die Frage nach ihrer Relevanz, also ob sie ein valides Angriffsziel darstellen. Eine fehlerhafte Warnung, dass der Reifendruck zu niedrig ist, ist in aller Regel kein Ereignis, dass ein Angreifer herbeizuführen versuchen wird.

Das FZI Tool bietet die Möglichkeit diese Angriffspfade zu modellieren und die Bewertung der Angriffe durchzuführen. Dadurch lässt sich im Gesamtprozess eine Verwertung und Weiterverwendung der innerhalb der Safetyuntersuchungen durchgeführten Arbeiten erreichen, somit wird die angestrebte Verzahnung von Safety und Security ermöglicht. Aufgrund der bekannten und im Tool zu diesem Zeitpunkt modellierten Implementierung lassen sich die beteiligten Hardwarekomponenten identifizieren, die für den Fehlerfall betroffen sein müssen. Im nächsten Schritt wird der Fehlerbaum erneut verwendet, um die Bedingungen, die im Fehlerbaum dokumentiert sind mit einzubeziehen. Wenn beispielsweise auf logischer Ebene eine Ortsabhängigkeit gegeben ist, lässt sich diese ebenfalls durch eine Manipulation der GPS Daten, oder der beteiligten Komponenten, beziehungsweise der verwendeten Daten herbeiführen. Sowohl eine eventuell höhere Anzahl der verwendeten Komponenten, als auch eine durch die Bedingungen identifizierte höhere Komplexität, werden abgebildet und nehmen Einfluss auf die resultierende Risikobewertung.

Der Fehlerbaum wird weiter entwickelt, indem die leittechnische Hardware detaillierter abgebildet wird. Falls die Attack Trees Hinweise auf die Angriffsziele und auf die Barrieren, deren Ausfall hier modelliert wird, enthalten, wird eine entsprechende Verknüpfung hergestellt.

Insbesondere sollen in der Fehlerbaumanalyse bestimmte, besonders kritische Ausfälle und Komponenten detailliert modelliert werden, somit kann die Safety Analyse helfen Ziele mit hoher Priorität zu identifizieren.

Der zugrundeliegende Prozess ist in Abbildung 6 schematisch dargestellt.

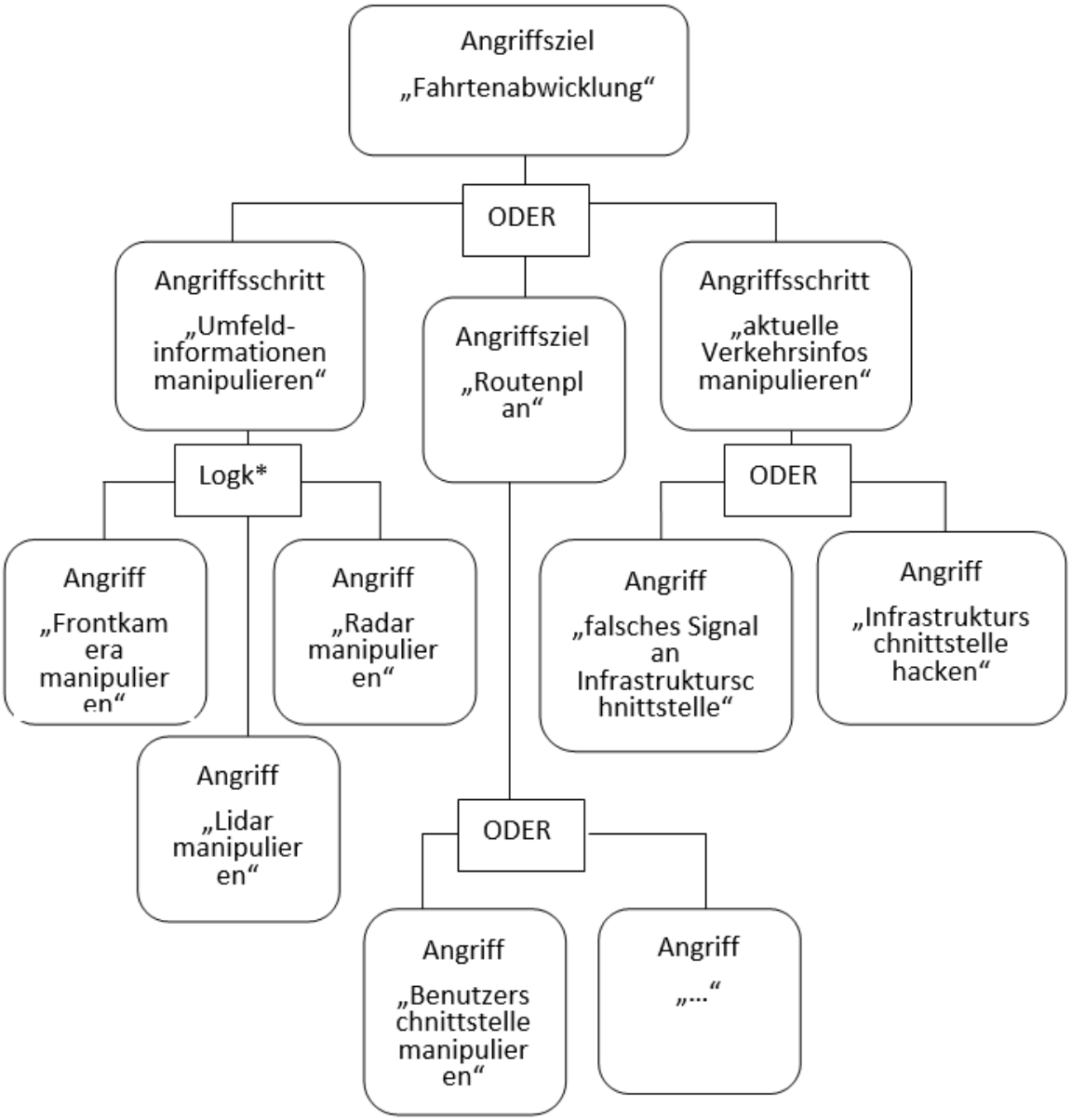


Abbildung 5: Fehlerbaum für Fahrtenabwicklung

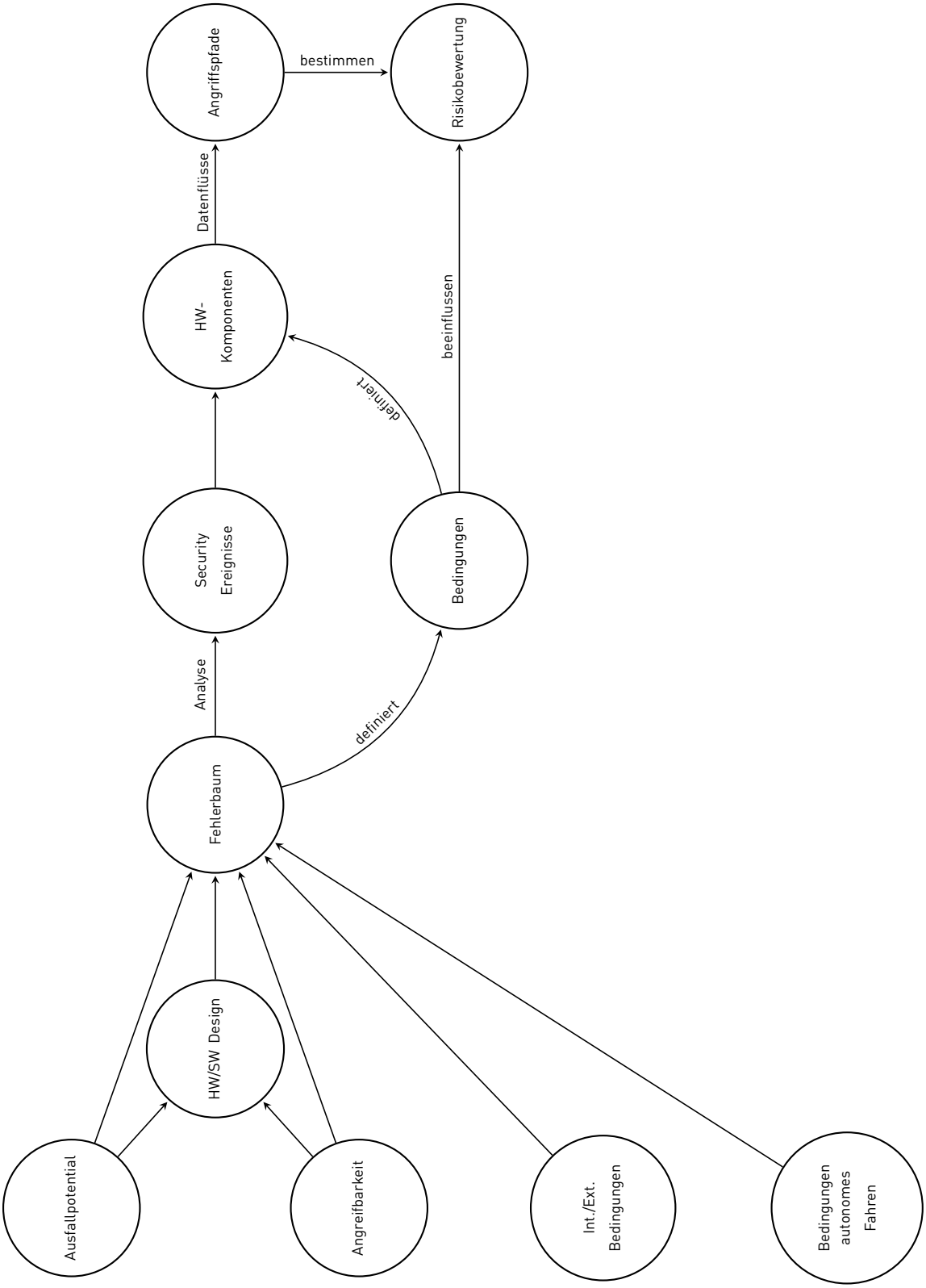


Abbildung 6: Abbildung: Prozess Angriffspfade aus Fehlerbaum

8 Verwertung und Einordnung

8.1 Wissenschaftliche Verwertbarkeit und Anschlussfähigkeit der Arbeiten

Die entwickelte Methodik, die sich durch Tooling unterstützen lässt, kann verwendet werden, um Angriffspfade und Gegenmaßnahmen zu modellieren. Es wurde dargestellt, dass die Verknüpfung von Safety und Securitybetrachtungen, innerhalb des V-Modells prinzipiell durchführbar ist. Dabei wurden insbesondere die Auswirkungen von Safetyanforderungen auf Securitymaßnahmen betrachtet und ein entsprechender Prozess definiert. Ebenso wurde definiert, dass die möglichen negativen Effekte von Securityanforderungen mit den Safetyanforderungen abgeglichen werden sollten. Daraus ergeben sich konkrete Anforderungen an die Katalogisierung von Komponenten, Datenflüssen und Schnittstellen, sowie deren Risikoanalyse.

Es wurde außerdem mit den Partner eine Methode entwickelt, die es erlaubt aus funktionalen Betrachtungen heraus Konfigurationen für bestimmte Hardwarekonfigurationen zu generieren, die zum einem minimalen Datenfluss führen. Die Konfigurationsgeneratoren sind derzeit auf wenige Hardwarebeispiele beschränkt, lassen sich aber prinzipiell auf beliebige weitere Technologien ausweiten.

Innerhalb des Gesamtarbeitspakets wurde gezeigt, dass das entwickelte Tooling, beziehungsweise die Weiterentwicklung des vorhandenen Toolings, den Anforderungen gerecht werden kann. Zudem wurde der notwendige Detailgrad einer ausreichenden Betrachtung eines Teilsystems anhand von bekannten Angriffen nichtautonomer Fahrzeuge definiert. Dieser Detailgrad hat sich im vorhandenen Tooling als abbildbar herausgestellt. Es ist davon auszugehen, dass dies auch für künftige Architekturen gültig ist. Die dargestellten Überlegungen beruhen nicht auf einer bestimmten E/E-Architektur und sind nur genau so detailliert dargestellt wie es für die gezeigten Beispiele notwendig war. Ein Grund hierfür ist, dass die momentan verwendeten Architekturen nicht öffentlich zugänglich sind. Des Weiteren unterscheiden sich die Architekturen je nach Hersteller, so dass eine Spezialisierung auf einen einzelnen Hersteller oder eine bestimmte Architektur zu einer Minderung der Allgemeingültigkeit führen würde.

Das Teilvorhabens im Projekt DEFEnD hat somit zur Entwicklung von Konzepten für Security by Design in Kombination mit funktionalen Sicherheitsaspekten beigetragen. Es wurde gezeigt, dass eine enge Verzahnung der Security Methoden mit dem Entwurf von Elektrik/Elektronik Architekturen und den dort bereits angeordneten Safety Fragestellungen vereinbar ist. Einer der Mehrwerte des Ansatzes besteht darin, dass ein sequentieller Entwurf bei dem nachgelagert Safety, Security und Rechtsprechung betrachtet werden, erheblich höhere Aufwände und Kosten hätte, als wenn dies bereits in frühen Entwurfsphasen erfolgt. Methoden, Prozesse und Konzepte die diese Anforderungen im Systementwurf berücksichtigen können, wurden in enger Kooperation mit den Projektpartnern in Modellbeschreibungen (Meta-Modellen) prototypisch realisiert.

Eine vollständige Modellierung eines modernen Systems war weder durchführbar noch zielführend. Es ist zudem unklar wie Architekturen für autonomes Fahren künftig konzipiert sein werden. Ziel war es daher exemplarisch zu zeigen, dass mit den entworfenen Methoden eine Modellierung mit Safety/Security Bezug durchführbar ist. Die erarbeitete Methodik wurde daher allgemein gehalten und erweiterbar gestaltet, um eine zukünftige, flexible Verwendung zu ermöglichen. Die dargestellten Methoden und Anwendungen sind daher so konzipiert, dass die Fallbeispiele sich zwar auf eine spezielle Architektur beziehen, die Methoden und Werkzeuge es aber zulassen beliebige Architekturen zu modellieren und zu betrachten. Für zukünftige Arbeiten erscheint eine Anwendung der Methodik auf bereits implementierte oder zu implementierende Gesamt- oder Teilarchitekturen durchführbar. Eine weitergehende Untersuchung mit Hinblick auf realitätsnahe Datensätze und gegebenenfalls auch Einbindung von Daten aus gängigen Modellierungslösungen wäre ebenfalls wünschenswert.

8.2 Verwertbarkeit im Projektanschluss

Wichtig für ERNW als Beratungsunternehmen war im Projekt die wissenschaftliche Verwertung der Projektergebnisse und der Aufbau von weiterem Wissen in diesem Fachbereich. Als herstellerunabhängiges Beratungsunternehmen ist für ERNW eine direkte kommerzielle Verwertung der aus dem Projekt entstehenden Produktkonzepte nicht möglich. Vorgesehen ist jedoch eine Verwertung der Forschungsergebnisse in Folgeprojekten und die Einbringung der gewonnenen Erfahrungen in Form von Schulungen oder Beratungsdienstleistungen.

8.3 Bekanntgewordener Fortschritt an anderen Stellen

Der während des Projektverlaufs bekanntgewordene Fortschritt an anderen Stellen wird im Abschlussbericht des Gesamtvorhabens im Detail beschrieben.

Während der Laufzeit wurden keine Forschungsergebnisse bekannt die mit den hier geschilderten Methoden im Widerspruch stehen.

8.4 Bezug zum zahlenmäßigen Nachweis

Der mit Abstand größte Anteil der entstandenen Kosten ist durch die entstandenen Personalkosten gegeben.

Für die Beiträge in den Arbeitspaketen war keine Neuanschaffung von Hardware oder ähnlichem notwendig.

Geringe Beträge waren zur Anschaffung noch nicht vorhandener Standards aufgewendet.

Geplante Reisekosten für Konferenzbeträge sind aufgrund der Corona-Pandemie entfallen.

8.5 Veröffentlichungen

Eine Veröffentlichung eines Teilaspekts von Arbeitspaket 3.1, fand im Rahmen eines Konferenzbeitrags [3] statt. Weitere Veröffentlichungen sind derzeit nicht geplant.

9 Appendix

9.1 Maßnahmen

9.1.1 Microsoft SDL

Microsoft geht bei SDL von folgenden Grundsätzen aus:

`Secure by design` Schon in der Planungsphase sollte auf die Sicherheitsbelange der Software eingegangen werden.

`Secure by default` Trotz sorgfältigster Planung sollte ein Entwickler von dem Vorhandensein von Sicherheitslücken ausgehen. Aus diesem Grund sollten die Standardeinstellungen (z. B. erforderliche Privilegien) möglichst niedrig gewählt werden und selten benutzte Features standardmäßig deaktiviert werden.

`Secure in deployment` Die mitgelieferten Dokumentationen und Tools sollen die Administratoren dabei unterstützen, die Software möglichst optimal einzurichten.

`Communications (Software)` Die Entwickler sollten offen mit möglichen Sicherheitslücken umgehen und den Endanwendern schnell Patches oder Workarounds zur Verfügung stellen.

`Privacy by design` Schon in der Planungsphase sollten Datenschutzbelange der Software berücksichtigt werden.

`Privacy by default` Die Standardeinstellungen der Software sollten konservativ gewählt werden.

`Privacy in deployment` Datenschutzmechanismen sollten offengelegt werden, um es Administratoren zu ermöglichen, die internen Datenschutzrichtlinien des Unternehmens umzusetzen.

`Communications (Privacy)` Datenschutzerklärungen sollten transparent formuliert werden. Ein Team für Datenschutzvorfälle sollte eingerichtet werden.

9.1.2 OWASP Maßnahmen Hardware

Beispielsweise kann die Befolgung der OWASP embedded application security guidelines [14] gefordert sein.

Die OWASP embedded application security top 10 listet die folgenden Maßnahmen:

- E1 – Buffer and Stack Overflow Protection
- E2 – Injection Prevention
- E3 – Firmware Updates and Cryptographic Signatures
- E4 – Securing Sensitive Information
- E5 – Identity Management
- E6 – Embedded Framework and C-Based Hardening
- E7 – Usage of Debug Code and Interfaces

- E8 – Transport Layer Security
- E9 – Data collection Usage and Storage - Privacy
- E10 – Third Party Code and Components

Diese Liste findet zum Beispiel auch zur Klassifizierung von Findings in Bug Bounty Programmen (siehe beispielsweise [1]) Anwendung.

9.2 Bewertungen und Metriken

9.2.1 Verbreitung von Security Standards

Es existiert eine Vielzahl von Security Standards die für die Bewertung der Einzelschritte einer tatsächlichen Risikoanalyse verwendet werden können. Abbildung 7 gibt exemplarisch die am häufigsten verbreiteten Standards nach [15] an.

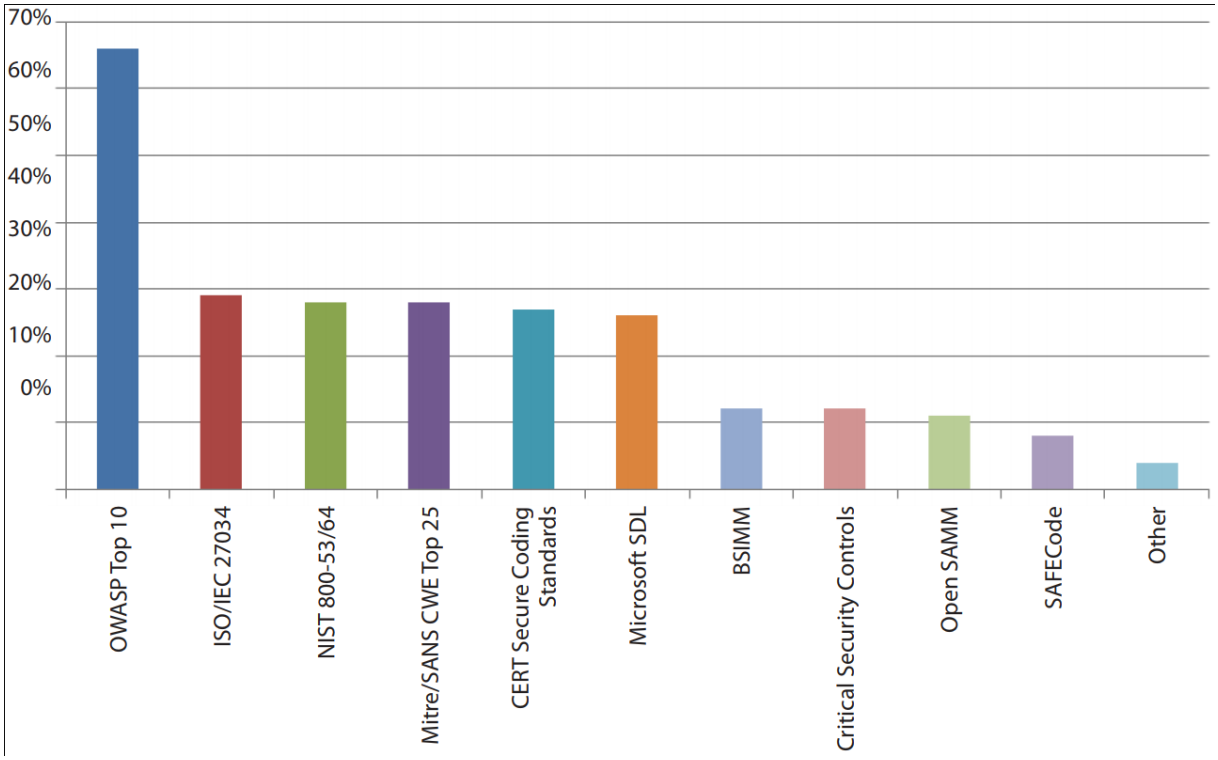


Abbildung 7: Verbreitung von Security Standards und Modellen nach [15].

9.2.2 Auflistung verbreiteter Methoden zur Bedrohungs- und Risikomodellierung

- STRIDE
- VAST
- TRIKE
- DREAD
- CVSS
- PASTA
- OCTAVE

- LINDDUN
- HEAVENS

ERNW
providing security.

10 Literaturverzeichnis

[1] bmwgroup. *How to report vulnerabilities - BMW GROUP. SECURITY.* 2020. URL: https://www.bmwgroup.com/en/general/Security.html#ace-1537276 (Zugriff am 20. Aug. 2020).

[2] Tobias Dörr, Timo Sandmann und Jürgen Becker. „A Formal Model for the Automatic Configuration of Access Protection Units in MPSoC-Based Embedded Systems". In: *2020 23rd Euromicro Conference on Digital System Design (DSD)*. 2020, S. 596–603. DOI: 10.1109/DSD51259.2020.00098.

[3] Tobias Dörr, Timo Sandmann, Hannes Mohr et al. „Employing the Concept of Multilevel Security to Generate Access Protection Configurations for Automotive On-Board Networks". Englisch. In: *2021 24th Euromicro Conference on Digital System Design (DSD)*. 24th 24th Euromicro Conference on Digital System Design. 2021 (Palermo, Italien, 1.–3. Sep. 2021). Institute of Electrical and Electronics Engineers (IEEE), 2021, S. 107–114. DOI: 10.1109/DSD53832.2021.00026.

[4] ENISA. *Hardware Threat Landscape and Good Practice Guide*. Techn. Ber. https://www.enisa.europa.eu/publications/hardware-threat-landscape, 2021.

[5] ENISA. *Threat Landscape for Supply Chain Attacks*. Techn. Ber. https://www.enisa.europa.eu/publications/threat-landscape-for-supply-chain-attacks, 2021.

[6] Inc FIRST. *Common Vulnerability Scoring System version 3.1*. Techn. Ber. https://www.first.org/, 2019.

[7] *ISO26262 - Road vehicles — Functional safety — Part 1: Vocabulary*. Techn. Ber. ISO, Dez. 2018.

[8] et. al. Jackson Wynn Joseph Whitmore. „Threat Assessment & Remediation Analysis (TARA)". In: (2011).

[9] keenlab, tencent. *Experimental Security Assessment of BMW Cars: A Summary Report*. 2018. URL: https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Assessment_of_BMW_Cars_by_KeenLab.pdf (Zugriff am 20. Aug. 2020).

[10] Mitre. *Vulnerability Details : CVE-2018-9311*. 2018. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-9311 (Zugriff am 20. Aug. 2020).

[11] Mitre. *Vulnerability Details : CVE-2018-9312*. 2018. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-9312 (Zugriff am 20. Aug. 2020).

[12] Mitre. *Vulnerability Details : CVE-2018-9318*. 2018. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-9318 (Zugriff am 20. Aug. 2020).

[13] Mitre. *Vulnerability Details : CVE-2018-9320*. 2018. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-9320 (Zugriff am 20. Aug. 2020).

[14] OWASP. *OWASP Embedded Application Security*. 2020. URL: https://owasp.org/www-project-embedded-application-security/ (Zugriff am 20. Aug. 2020).

[15] SANS Institute. *2015 State of Application Security: Closing the Gap*. 2015. URL: https://www.sans.org/reading-room/whitepapers/analyst/2015-state-application-security-closing-gap-35942 (Zugriff am 14. Sep. 2020).

[16] Christoph Schmittner, Zhendong Ma, Carolina Reyes et al. „Using SAE J3061 for Automotive Security Requirement Engineering". In: Bd. 9923. Sep. 2016, S. 157–170. DOI: 10.1007/978-3-319-45480-1_13.

Berichtsblatt

1. ISBN oder ISSN -	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht

3. Titel
DEvelopment For SEcured Autonomous Driving (DEFEnD)
Abschlussbericht zum Teilvorhaben

4. Autor(en) [Name(n), Vorname(n)]	5. Abschlussdatum des Vorhabens
Mohr, Hannes	31.05.2021
	6. Veröffentlichungsdatum 30.11.2021
	7. Form der Publikation online
8. Durchführende Institution(en) (Name, Adresse) ERNW Enno Rey Netzwerke GmbH Carl-Bosch Str. 4 69115 Heidelberg	9. Ber. Nr. Durchführende Institution -
	10. Förderkennzeichen 16KIS0887
	11. Seitenzahl 31
12. Fördernde Institution (Name, Adresse) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	Literaturangaben: 16
	14. Tabellen 0
	15. Abbildungen 7

16. Zusätzliche Angaben
- Verwendete Standards: 4 Eine Auflistung befindet sich im Schlussbericht unter Kapitel 1.4

17. Vorgelegt bei (Titel, Ort, Datum)
-

18. Kurzfassung

Das Teilvorhabens im Projekt DEFEnD hat somit zur Entwicklung von Konzepten für Security by Design in Kombination mit funktionalen Sicherheitsaspekten beigetragen. Es wurde gezeigt, dass eine enge Verzahnung der Security Methoden mit dem Entwurf von Elektrik/Elektronik Architekturen und den dort bereits angeordneten Safety Fragestellungen vereinbar ist. Einer der Mehrwerte des Ansatzes besteht darin, dass ein sequentieller Entwurf bei dem nachgelagert Safety, Security und Rechtsprechung betrachtet werden, erheblich höhere Aufwände und Kosten hätte, als wenn dies bereits in frühen Entwurfsphasen verzahnt erfolgt. Methoden, Prozessen und Konzepte, die diese Anforderungen im Systementwurf berücksichtigen können wurden in enger Kooperation mit den Projektpartnern in Modellbeschreibungen (Meta Modelle) prototypisch realisiert.

19. Schlagwörter
Safety, Security, autonomes Fahren

20. Verlag	21. Preis
-	-

Document Control Sheet

1. ISBN or ISSN -	2. type of document (e.g. report, publication) Schlussbericht
3. title DEvelopment For SEcured Autonomous Driving (DEFEnD) Abschlussbericht zum Teilvorhaben	
4. author(s) (family name, first name(s)) Mohr, Hannes	5. end of project 31.05.2021
	6. publication date 30.11.2021
	7. form of publication online
8. performing organization(s) (name, address) ERNW Enno Rey Netzwerke GmbH Carl-Bosch Str. 4 69115 Heidelberg	9. originator's report no. -
	10. reference no. 16KIS0887
	11. no. of pages 31
12. sponsoring agency (name, address) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. no. of references 16
	14. no. of tables 0
	15. no. of figures 7
16. supplementary notes - Verwendete Standards: 4 Eine Auflistung befindet sich im Schlussbericht unter Kapitel 1.4	
17. presented at (title, place, date) -	
18. abstract The subproject conducted by ERNW within the DEvelopment For Secured Autonomous Driving (DEFEnD) project strived to combine functional safety and security concepts within the development of autonomous cars. The project was able to show that a combination of established security and safety concepts and methods is in fact possible when designing E/E architectures. One of the benefits of the approach lies within the projected reduction of operational and development costs by avoiding a sequential approach to the problem. Instead, starting from early development stages, methods, processes and concepts can be employed that parallelize Safety, Security and Legal considerations. To this end and in collaboration with the associated partners, numerous methods, processes and concepts were developed and prototypically realized within model descriptions and meta models.	
19. keywords Safety, Security, autonomes Fahren	
20. publisher -	21. price -