Prof. Dr. Christof Ebert
(christof.ebert@vector.com)
ist Geschäftsführer der Vector Consulting Services GmbH und Professor an der Universität Stuttgart.

Dr. Eduard Metzker
(eduard.metzker@vector.com)
ist Solution Manager für Cyber Security bei der Vector Informatik GmbH.

Integrierte Entwicklung von Safety- und Security-Anforderungen

Die systematische Entwicklung von Safety-Anforderungen ist heute unabdingbar bei der Entwicklung von eingebetteten Systemen. Die steigende Anzahl von Funktionen mit Anbindung an Backend- und Cloud-Dienste verlangt, dass *Safety* zunehmend zusammen mit *Security* betrachtet wird. Hierfür fehlt noch eine durchgängige Vorgehensweise. In diesem Beitrag wird eine Methodik für eine integrierte semiformale Entwicklung von Safety- und Security-Anforderungen vorgestellt. Die Vorteile dieser Safety/Security Requirements Engineering-Methodik werden am Beispiel eines ADAS (Advanced Driver Assistance System) illustriert und das Potenzial einer Werkzeugunterstützung wird demonstriert.

Anforderungen für Safety und Security

Überlebensnotwendige Cloud-Dienste wie Notfallsysteme werden durch Überlastattacken lahm gelegt. Hacker dringen in Zugsteuerungen ein und verursachen Fehlfunktionen. Industrieanlagen werden durch Trojaner sabotiert. Implantierte Medizintechnik wird über die Diagnoseschnittstelle manipuliert und versagt ihren Dienst. Wer kennt diese Schlagzeilen nicht?

Die Sabotage von kritischer Infrastruktur ist nicht nur das Ziel von Terroristen, sondern leider auch zunehmen von irregeleiteten Fachleuten. Der eine will damit die Ohnmacht unserer Gesellschaft demonstrieren und der andere seine eigene Macht. Immer mehr eingebettete Systeme sind sicherheitskritisch.

Kompromittierte Software kann zum Ausfall kritischer Funktionen führen. Eine Insulinpumpe, wie sie bei vielen Personen implantiert ist, kann von einem informierten Hacker so beeinflusst werden, dass sie eine falsche Menge abgibt – mit möglicherweise tödlichen Konsequenzen.

Es ist höchste Zeit, Sicherheitsanforderungen nicht nur aus Softwarefehlern und zufälligen Fehlfunktionen abzuleiten, sondern die Aufmerksamkeit ganz gezielt auf Angriffe im Sinne der Informationssicherheit zu lenken.

Funktionale Sicherheit und Informationssicherheit wachsen zunehmend zusammen. Funktionale Sicherheit benötigt eine verlässliche Informationssicherheit, egal ob es sich um Fahrzeuge, Medizintechnik oder Automatisierung handelt. Dabei stehen Anforderungen an die Sicherheit und damit ein nachvollziehbares und konsequentes und durchgängiges Requirements Engineering im Mittelpunkt.

Das Ziel der integrierten Entwicklung von Safety und Security ist die Entwicklung von Funktionen, die möglichst robust auf technische und menschliche Fehler sowie Angriffe von außen reagieren und so das Risiko von Gefährdungssituationen auf ein akzeptables Maß begrenzen. Doch wie können funktionale Sicherheit und Informationssicherheit in solchen Systemen aus Sicht des Requirements Engineering erfolgreich realisiert werden?

Der Begriff „Sicherheit“ hat in der deutschen Sprache zwei Bedeutungen. Sicherheit bedeutet die Abwesenheit von Gefahr. Die Nutzung eines Produkts oder einer Funktion darf keine Gefahr für Gesundheit oder Leben des Nutzers und seiner Umwelt verursachen. Der Produzent – und damit die Entwickler einer Funktion – muss sicherstellen, dass diese Funktion zur Verfügung steht, wenn sie benötigt wird.

Die gleiche Bedeutung hat die Absicherung gegen den Fall, dass Funktionen ausgeführt werden, ohne dass dies notwendig ist. Dabei kann ein Maßnahmenkatalog dazu beitragen, dass die gewünschte Funktion unter verschiedenen Randbedingungen erhalten bleibt. Entsprechende Überwachungsmaßnahmen stellen die ordnungsgemäße Ausführung sicher. Wir nennen den Begriff daher im

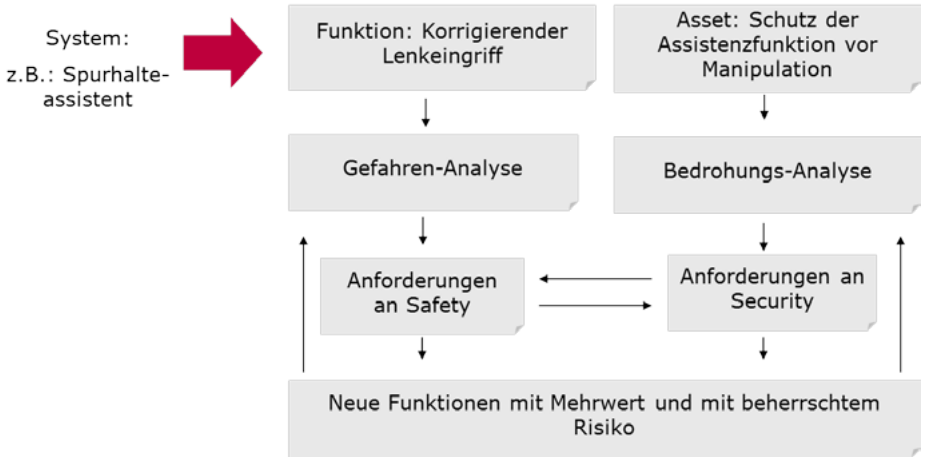

Abb. 1: Requirements Engineering identifiziert Wechselwirkungen und Abhängigkeiten von Safety und Security

Deutschen auch „funktionale Sicherheit“ oder im Englischen „Safety“.

Abwesenheit von Gefahr bedeutet aber auch Vertrauen. Ich kann mich darauf verlassen, dass die Nachricht von einem Übermittler auch wirklich von ihm stammt. Oder ich möchte eine Nachricht übermitteln und dabei sicher sein, dass sie niemand beeinflussen kann. Dieser Begriff der Sicherheit stellt im näheren Sinne das Themengebiet „Security“ dar. Es geht um Integrität, Authentizität und Vertraulichkeit von Informationen, weshalb man im deutschen auch von „Informationssicherheit“ spricht.

Qualität in einem System bedeutet grundsätzlich, dass das System alles das tut, was von ihm erwartet wird [Ebe14]. Die Verknüpfung von Safety und Security erweitert diese klassische Definition dahingehend, dass das System sowohl bei Ausfällen, menschlichen und technischen Fehlfunktionen und auch bei einem böswilligen Angriff nichts tut, das von ihm nicht erwartet wird. Wir werden in diesem Beitrag zur besseren Nachvollziehbarkeit der verschiedenen Semantik und damit auch der unterschiedlichen methodischen Vorgehensweisen die Begriffe „Safety“ und „Security“ verwenden.

Funktionale Sicherheit – also „Safety“ – ist ohne Informationssicherheit – also „Security“ – in Zukunft nicht mehr ausreichend. Das zeigt sich an zwei wesentlichen Trends: die stetig steigende Vernetzung von Funktionen sowie die Entwicklung zunehmend autonom agierender Diagnose- und Assistenzsysteme.

Beide Trends haben als Folge eine wachsende Anzahl von potenziell sicherheitsrelevanten Funktionen. Als relativ neuer Trend ist die wachsende Vernetzung von Fahrzeugfunktionen mit Back-End-Diensten von Fahrzeugherstellern oder IT-Cloud-Diensten erkennbar.

Diese Vernetzung eröffnet vollkommen neue Dienste und Geschäftsmodelle wie beispielsweise die Nachrüstung von Funktionen per Software-Update oder die Kommunikation zwischen Geräten, beispielsweise IT und Automatisierungstechnik in der Industrie 4.0. Diese Verknüpfung ganz unterschiedlicher Komponenten und deren Netze mit offenen externen Netzen vergrößern die Bedrohungen in Bezug auf Security.

Die Abhängigkeiten und Wechselwirkungen zwischen Safety- und Security-Anforderungen müssen im Requirements Engineering methodisch sauber und systematisch betrachtet werden. Die beschriebenen Zusammenhänge sind in Abbildung 1 visualisiert. Diese müssen in einer integrierten Vorgehensweise für die Entwicklung und Verifikation von Anforderungen beherrschbar gemacht werden.

Momentan geschieht dies eher unsystematisch, da die Industriestandards für Safety und Security noch stark voneinander isoliert aufgestellt sind. Wir wollen in diesem Beitrag auf die Integration der Vorgehensweise im Rahmen eines systematischen Requirements Engineering eingehen.

Im nachfolgenden Abschnitt werden zunächst Ausschnitte der Vorgehensweise für Safety-Anforderungen dargestellt. Wir beschränken diese Ausschnitte bewusst bis zur Erstellung der funktionalen Sicherheitsanforderungen, da die frühen Phasen prägend für den Rest der Vorgehensweise sind. Diese bilden die Grundlage zur Integration der Vorgehensweise für Security-Anforderungen die im darauffolgenden Abschnitt vorgestellt wird. Danach werden die integrierte Vorgehensweise am Beispiel eines ADAS (Advanced Driver Assistance System) illustriert und das Potenzial für eine Werkzeugunterstützung dargestellt.

Entwicklung und Verifikation von Safety-Anforderungen

Zunächst wird das betrachtete System abgegrenzt und in seinen Grundfunktionen beschrieben. Zusätzlich werden relevante Betriebsszenarien definiert. In der Gefahren- und Risiko-Analyse werden ausgehend von den Grundfunktionen und möglichen Fehlfunktionen sogenannte Gefährdungsereignisse abgeleitet.

Unter Beachtung der relevanten Betriebsszenarien wird das Risiko des Gefährdungsereignisses mit dem sogenannten Safety Integrity Level quantifiziert. Für jedes relevante Gefährdungsereignis werden anschließend noch relativ grobe Safety-Anforderungen, sogenannte Safety-Ziele, identifiziert.

In den nachfolgenden Designschritten werden die Safety-Ziele auf funktionale und technische Sicherheitsanforderungen heruntergebrochen und den Elementen des System-Designs zugeordnet, die für die Realisierung der Anforderungen zuständig sind. Gemeinsam bilden diese das funktionale Sicherheitskonzept. Alle weiteren Safety-Prozessschritte wie beispielsweise quantitative und qualitative Sicherheitsanalysen befinden sich außerhalb des Scopes dieses Beitrags. Im folgenden Abschnitt ergänzen wir diese Methodik um Vorgehensweisen zur Entwicklung und Verifikation von Security-Anforderungen.

Entwicklung und Verifikation von Security-Anforderungen

Analog zur Definition der Grundfunktionen des Systems werden auf dieser Basis die schützenswerten Assets und deren Besitzer identifiziert [CCRA12]. Ein Asset für einen Fahrzeugbesitzer ist beispielsweise, dass seine Privatsphäre nicht dadurch verletzt wird, dass seine Bewegungsprofile durch Dritte ausgewertet werden können.

Bei der Bedrohungs- und Risikoanalyse werden – ausgehend von den Assets – Angreifer identifiziert, welche das Potenzial haben, eine feindliche Aktion auf einen Angriffspunkt des Systems durchzuführen und damit eine Bedrohung für das Asset darstellen. Das Risiko ergibt sich aus dem

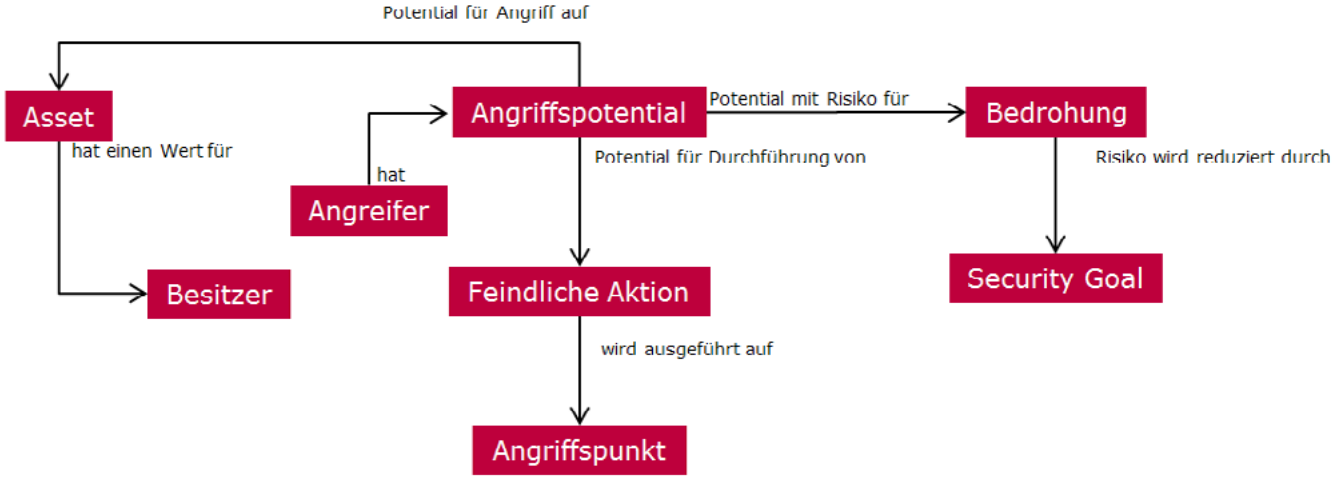

Abb. 2: Zusammenhänge bei der Bedrohungs- und Risikoanalyse

Angriffspotenzial sowie dem Schweregrad der Bedrohung. Die Besitzer der Assets wollen das Risiko der Bedrohungen mithilfe der Umsetzung geeigneter Security Goals auf ein akzeptables Maß reduzieren (siehe Abbildung 2).

In den nachfolgenden Designschritten werden die Security-Ziele auf funktionale Security-Anforderungen heruntergebrochen und den Elementen des System-Designs zugeordnet, die für die Realisierung der Anforderungen zuständig sind. Gemeinsam bilden diese das Funktionale Security-Konzept.

Analog zur Safety-Vorgehensweise befinden sich alle weiteren Security-Prozessschritte außerhalb des Fokus dieses Beitrags. Abbildung 3 stellt die betrachteten Security-Aktivitäten und ihre Beziehung zum Gesamtprozess sowie zu den Safety-Aktivitäten vereinfacht dar.

Fallstudie: ADAS-System

Beispielhaft wird im Folgenden ein Teilsystem eines ADAS (Advanced Driver Assistance System) für ein Automobil betrachtet. Dieses Teilsystem ist ein Spurhalte-Assistent (SHA), welcher den Fahrer vor dem unbeabsichtigten Verlassen der Spur warnt und falls nötig einen Lenkeingriff vornimmt, um das Fahrzeug zurück in die Spur zu lenken. Das System verfügt über eine Reihe von Sensoren zur Erfassung des Umfelds sowie eine Reihe von Aktuatoren für Fahreingriffe und die Benachrichtigung des Fahrers. In Abbildung 4 ist die vorgegebene Systemstruktur dargestellt.

An diesem Beispiel werden die Gemeinsamkeiten und Unterschiede bei Entwicklung von Safety- und Security-Anforderungen in den frühen Phasen verdeutlicht.

Aufgrund der direkten Einwirkungen in die Fahrdynamik hat das System Anforderungen an die funktionale Sicherheit, beispielsweise, um unbeabsichtigte Lenk- und Bremsmanöver zu vermeiden. Die Vernetzung ganz unterschiedlicher Informationsquellen aus der Sensorfusion wie Radar und Kamerasystemen sowie verschiedenen Steuergeräten im Fahrzeug verlangt nach einer sorgfältigen Absicherung gegen Eingriffe von außen.

Diese Anforderungen beeinflussen sich gegenseitig in ihrer methodischen Umsetzung und müssen daher gemeinsam entwickelt werden. Das wollen wir anhand der beschriebenen Methodik demonstrieren.

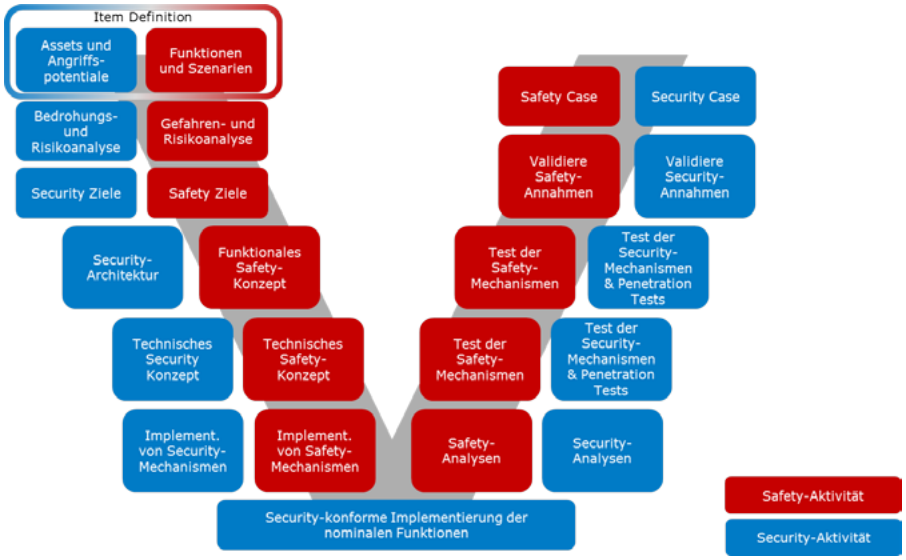

Abb. 3: Gemeinsame Betrachtung von Safety- und Security-Anforderungen

Identifikation von Grundfunktionen und Ableitung von Safety-Zielen

Im Rahmen der Systemdefinition werden zunächst die Grundfunktionen identifiziert und im Rahmen der Gefahren- und Risikoanalyse die Safety-Ziele abgeleitet.

Für das SHA-Beispiel wurde u. a. die folgende Grundfunktion F1 identifiziert: „Der Spurhalteassistent startet einen automatischen Lenkeingriff nachdem das Verlassen der Spur erkannt wurde und eine Warnzeit verstrichen ist“. Als für das SHA-System relevante Betriebssituation wurde u. a. die folgende relevante Betriebssituation BS1 identifiziert: “ Fahren auf Landstraßen, entgegenkommender Verkehr, Geschwindigkeit > 50km/h“. Im Rahmen der Gefahren- und Risikoanalyse wurde die Fehlfunktion FF1 für F1 identifiziert: „Das Gegenlenken wird durchgeführt, allerdings in die falsche Richtung“.

Aus dem Zusammenspiel von F1, FF1 und BS1 ergibt sich das Gefährdungsereignis H1:“ Er erfolgt ein Zusammenstoß mit dem entgegenkommenden Verkehr“. Um die Relevanz von H1 zu bewerten, werden als Kriterien die Auftretenswahrschein-

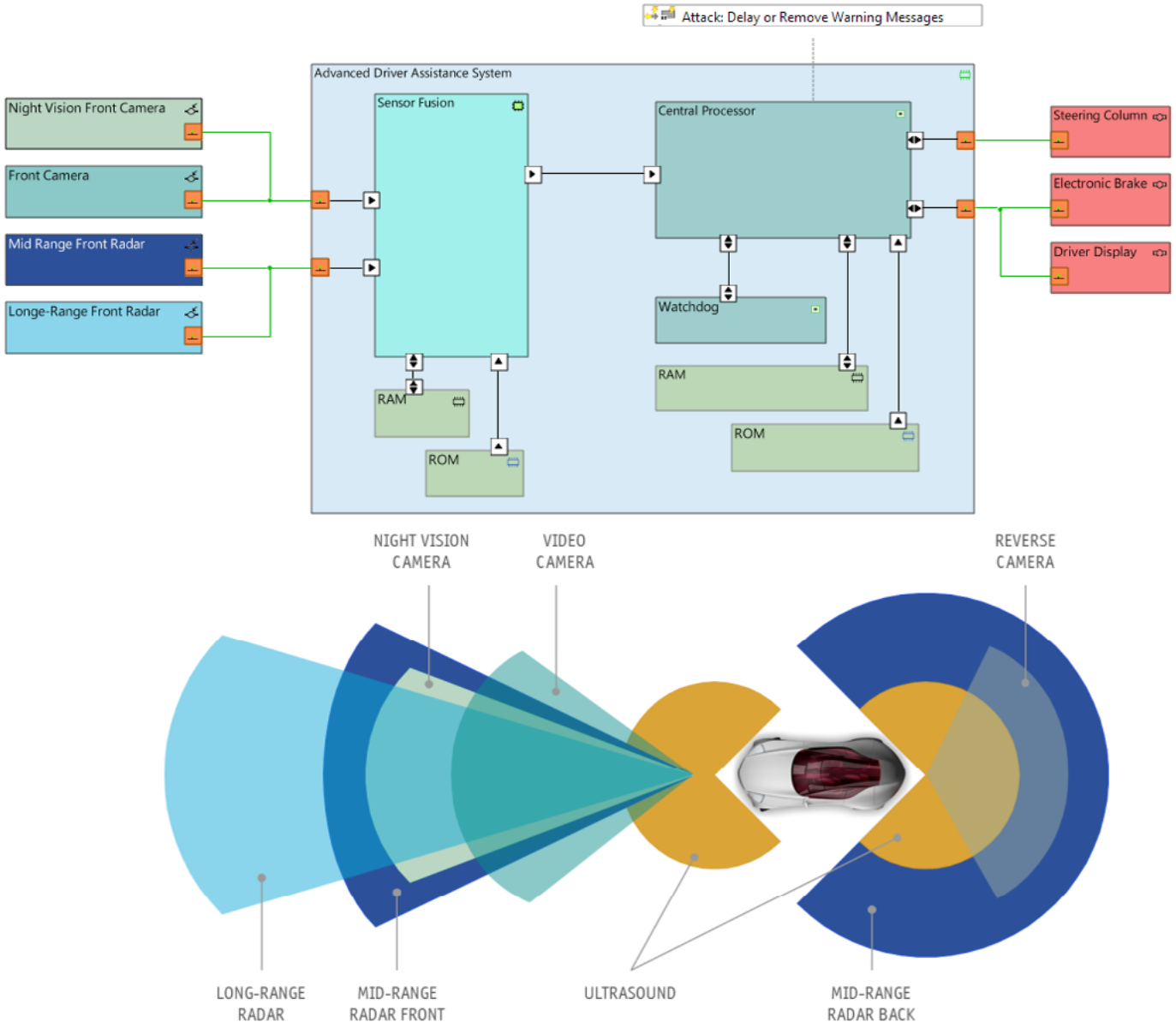

Abb. 4: Grobe Systemstruktur des ADAS (Teilsystem Spurhalteassistent)

lichkeit E, der Schweregrad S und die Kontrollierbarkeit C herangezogen. Zur Vermeidung von H1 wird das Sicherheitsziel SG1 definiert: „Vermeide nicht plausiblen Lenkeingriff des SHA-Systems".

Identifikation von Assets und Ableitung von Security-Zielen

Im Rahmen der System-Definition werden zunächst die schützenswerten Assets identifiziert und im Rahmen der Bedrohungs- und Risikoanalyse die Security-Ziele abgeleitet.

Im Rahmen der Asset-Definition wird unter anderem das Asset A1 definiert: „Der Spurhalteassistent verhält sich innerhalb seiner Spezifikation". Asset-Besitzer B1 ist der Fahrzeughersteller. Der Angreifer AT1 ist ein erfahrener Hacker. Ein Angreifer dieser Ausprägung verfügt typischerweise über die prinzipiellen Fähigkeiten des Reverse Engineering von Hardware und Software und die Möglichkeit über Remote-Schnittstellen privilegierte Schadsoftware zu installieren, die mit dem Rest des Systems kommuniziert.

AT1 kann die feindliche Aktion FA1 durchführen: „Unterdrücke Nachrichten an die Aktuatoren beim Verlassen der Fahrspur durch Manipulation der Funktionssoftware." Als Angriffspunkt AP1 wurde die Manipulation der Software des ADAS-Systems durch ein nicht autorisiertes Software-Update identifiziert.

Das Angriffspotenzial des Angreifers wird anhand von mehrdimensionalen Kriterien bewertet. Zu diesen Kriterien zählen unter anderem das Wissen des Angreifers über das System, die technische Ausrüstung des Angreifers sowie die benötigte Zeit für das Ausführen eines Angriffs. Der Angriff auf das Asset führt zu der Bedrohung T1: „Das Fahrzeug verhält sich wie bei ausgeschaltetem System, ohne dass dies dem Fahrer bewusst ist. Dadurch kann es zu Unfällen kommen, die dem Hersteller angelastet werden".

Der Schweregrad dieser Bedrohung kann nun anhand verschiedener Security-relevanter Kriterien bewertet werden. Dazu gehört das Potenzial der Bedrohung Safety-Ziele zu verletzen, aber auch das Potenzial der Bedrohung für operative und finanzielle Schäden des Asset-Besitzers B1.

Als Security-Goal zur Begegnung der Bedrohung wird SCG1 abgeleitet: „SW- und HW-Security-Schutzmechanismen müssen sicherstellen, dass Software- und Konfigurationsparameter für das SHA-

	Safety	Security
Ausgangspunkt	System und Top-Level Funktionen	System und zu schützende Assets
Ermittlung der Top-Level-Anforderungsebene	Risiko-basierte Ableitung von Safety-Zielen über Betrachtung der Fehlunktionen der Systemfunktionen	Risiko-basierte Ableitung von Security Zielen über Betrachtung von möglichen Angriffen auf Assets
Risikobewertung	Risiko für ein Gefährdungsereignis wird bewertet über Auftretenswahrscheinlichkeit, Schweregrad sowie Kontrollierbarkeit des Gefährdungsereignisses durch den Benutzer	Risiko einer Bedrohung wird bewertet über AngriffsPotenzial des Angreifers und Schweregrad der Bedrohung. Der Schweregrad teilt sich auf in die Bewertungsdimensionen Funktionale Sicherheit, Wirtschaftliche Kosten und Privatsphäre.

Tab.: Vergleich der Herangehensweise bei Safety- und Security-Anforderungen

System nicht verfälscht oder verzögert werden können, ohne dass dies unerkannt bleibt.“ In der Tabelle sind die Gemeinsamkeiten und Unterschiede bei der Ermittlung von Safety- und Security-Anforderungen zusammenfassend dargestellt.

Nach der Ableitung der Safety- und Security-Ziele kann die weitere Verfeinerung auf die funktionale und technische Anforderungseben erfolgen. Zu jeder Phase ist es sinnvoll, die Anforderungen durch Allokation an der (vorläufigen) funktionalen bzw. technischen Architektur zu spiegeln.

Potenzial für Werkzeugunterstützung

Bei klassischen Werkzeugketten wie sie heute bei der Entwicklung von Systemen eingesetzt werden, sind die Prozesse, die zu Beginn beschrieben wurden, stark fragmentiert. Anforderungsentwicklung, Funktions-Design sowie Safety- und Security-Analysen werden mit verschiedenen Werkzeugen durchgeführt, wobei die Schnittstellen oft keinen verlustfreien Austausch ermöglichen. Dadurch ist beispielsweise die Konsistenz der Safety- und Security-Ziele und der daraus abgeleiteten Anforderungen mit dem System-Design nur mit großem Aufwand über den gesamten Lebenszyklus hinweg sicherzustellen.

Als Schritt hin zu einem integrierten Safety- und Security-Konzept und der weiterführenden Verfeinerung der Anforderungen sollten die Safety- und Security-Ziele der vorläufigen Architektur zugeordnet werden. Dies hat den Vorteil, dass bereits früh eine Verfolgbarkeit von Anforderungen auf die Konzeptebene gegeben ist. Ebenso können dadurch potenzielle Widersprüche und Konflikte zwischen Safety- und Security-Anforderungen an System-Elemente früh erkannt und aufgelöst werden.

Dazu ist es vorteilhaft, wenn Werkzeuge in der Lage sind, die Anforderungsentwicklung und das System-Design integriert zu unterstützen. Dieses ermöglicht die in Abbildung 5 dargestellte grafische Visualisierung der Allokation. Mithilfe solcher Darstellungen sind Widersprüche rasch zu erfassen.

In Abbildung 2 haben wir die Zusammenhänge von Security-Zielen mit feindlichen Aktionen und Assets gezeigt. Solche Zusammenhänge lassen sich gerade in integrierten Werkzeugen für mustergestütz-

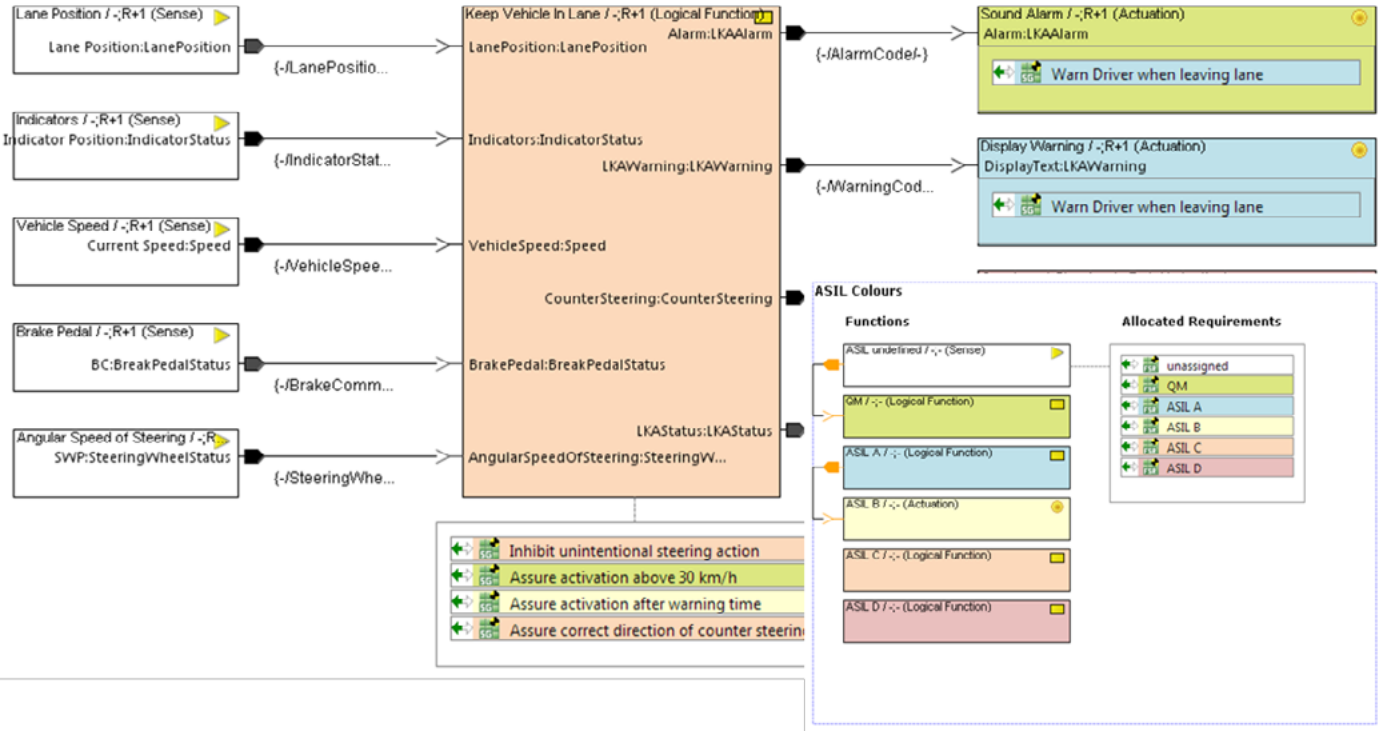

Abb. 5: Allokation von Safety-Zielen auf System-Elemente

te Konsistenzprüfungen leicht formalisieren, um Lücken und Unvollständigkeiten in Anforderungen zu erkennen. Wenn dies während der Entwicklung von Anforderungen geschieht, hat dies das Potenzial den Aufwand für manuelle Reviews zu reduzieren.

Zusammenfassung und Ausblick

Requirements Engineering bei sicherheitskritischen Systemen muss Anforderungen an die funktionale Sicherheit und an die Informationssicherheit systematisch parallel entwickeln und bewerten. Die bisher etablierte Trennung der zwei „Disziplinen“ mit jeweils eigenen Standards und Vorgehensweisen ist nicht länger tragfähig, da Abhängigkeiten und gegenseitige Beeinflussungen übersehen werden.

Zudem ist eine getrennte Vorgehensweise ineffizient, da viele Funktionen mehrere Male angefasst werden müssen. Methodische Vorgehensweisen für Konsistenzsicherung, Traceability, Verifikation, Modellierung und Testorientierung sollten gemeinsam angewandt werden. Eine semiformale Spezifikation von Safety- und Security-Anforderungen zusammen mit geeigneter Werkzeugunterstützung trägt dazu bei, die Komplexität und den Aufwand beherrschbar zu halten. ■

Literatur

[Ebe14] Ebert, C.: Systematisches Requirements Engineering. Dpunkt-Verlag, Heidelberg, Germany, 5. komplett überarbeitete Auflage, 2014.

[CCRA12] CCRA, „ISO/IEC 15408: Common Criteria for Information Technology Security Evaluation“, 2012.

URLs zum Beitrag:

1. Buch “Systematisches Requirements Engineering” Kurz-URL: www.vector.com/RE-Buch
2. Beiträge zu Informationssicherheit und funktionaler Sicherheit: Kurz-URL: www.vector.com/security, www.vector.com/safety