VDI

Fahrzeug- und
Verkehrstechnik

VDI-Fachtagung

Automotive Security

VDI-Berichte 2310

VDI-BERICHTE
Herausgeber: VDI Wissensforum GmbH

VDI Fahrzeug- und Verkehrstechnik

VDI-Fachtagung

Automotive Security

Nürtingen, 27. und 28. September 2017

VDI-Berichte 2310

Bibliographische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter http://dnb.ddb.de abrufbar.

Bibliographic information published by the Deutsche Nationalbibliothek
(German National Library)
The Deutsche Nationalbibliothek lists this publication in the Deutsche National-bibliographie
(German National Bibliography); detailed bibliographic data is available via Internet at http://dnb.ddb.de.

Der VDI-Bericht, der die Vorträge der Tagung enthält, erscheint als nichtredigierter Manuskriptdruck. Die einzelnen Beiträge geben die auf persönlichen Erkenntnissen beruhenden Ansichten und Erfahrungen der jeweiligen Vortragenden bzw. Autoren wieder.

Printed in Germany.

ISSN 0083-5560

ISBN 978-3-18-092310-9

Inhalt

Cyber-Abwehr

Vorwort

Steuergeräte oder Fahrzeugnetze sind hochgradig durch mögliche Hackerattacken gefährdet, ein Thema dass die Automobilhersteller massiv beschäftigt. Zahlreiche Live-Hacks auf IT-Sicherheits Veranstaltungen haben gezeigt, wie verwundbar vernetze Fahrzeuge sind. Doch welche Auswirkungen haben diese Demonstrationen auf den Endkunden? Die VDI-Fachtagung „Automotive Security“ wendet sich im kommenden Jahr erneut den aktuellen Themen rund um den Datenschutz und die Datensicherheit im Fahrzeug zu. Ein Augenmerk liegt hier unter anderem auf dem Umgang mit Kundenwünschen versus der gesetzlichen und technischen Lage. Wie gehen Automobilhersteller unter Beachtung dieser Fragen bei der Entwicklung von Steuergeräten und Betriebssystemen vor? Wie kann man den Datenschutz im sowohl im Inneren des Systems sicherstellen als auch das gesamte System vor externen Angriffen schützen, ohne gewollte Funktionen zu blockieren? Dem vernetzten Fahrzeug kommt darüber hinaus eine ganz besondere Rolle zu, da es im Gegensatz zum Smartphone kein optionales Mobilgerät oder Hilfsmittel zur Überwindung einer Distanz darstellt, sondern immer öfter auch ein Arbeitsgerät oder gar Arbeitsplatz ist. Auch aus diesem Grund gibt es viele sicherheitsrelevante Aspekte, die immer stärker in den Fokus rücken. Für die VDI-Fachtagung am 27. und 28. September 2017 hat der Programmausschuss der Tagung deshalb erneut ein spannendes Themenfeld rund um das Motto „Security und Datenschutz im Fahrzeug von morgen“ abgesteckt.

VDI Wissensforum GmbH

Divide & Conquer: Effiziente und übersichtliche Risikoanalyse und -behandlung

Dr.-Ing. **Jens Köhler**, Dr.-Ing. **Sebastian Labitzke**,
ITK Engineering GmbH, Rülzheim

1. Einleitung

Aus Kostengründen kann in Security-Konzepten nicht jeder erdenkliche Security-Mechanismus integriert werden. Vielmehr müssen die Security-Mechanismen angewendet werden, die gegen tatsächlich auftretende Angriffe mit inakzeptablen Auswirkungen schützen. Ziel ist somit ein sogenanntes „angemessenes Sicherheitsniveau“ durch adäquate Security-Konzepte zu gewährleisten. Das Sicherheitsniveau wird in der Regel im Rahmen einer Bedrohungs- und Risikoanalyse in Form von Security-Anforderungen an eine Systemarchitektur [1] und verwendete Kommunikationsprotokolle festgelegt [2,3].

In der Automobilindustrie etablieren sich hierfür derzeit Analysemethodiken, die oft ähnliche Ansätze verfolgen, jedoch im Detail differieren und so zu identifizierten Risiken führen, die sich hinsichtlich Nachvollziehbarkeit, Genauigkeit, Vergleichbarkeit und Wiederverwendbarkeit unterscheiden. Beispielsweise stellen einige Methodiken den Angreifer ins Zentrum der Analyse [4], während andere die Schwachstellen des Systems [5] oder die System-Assets [6], die es zu schützen gilt, fokussieren. Weiterhin existieren stark strukturierte und geführte Risikoanalysemethodiken neben „kreativen“ Risikoanalysen, die oft Angriffspfade beschreiben und von der Kreativität des Risikoanalysten abhängen. Die Vor- und Nachteile der Methodiken abzuwiegen, fällt oft aufgrund der Komplexität der Risikoanalyseproblematik an sich und teils subjektiven individuellen Vorlieben und Erfahrungen schwer.

In diesem Beitrag wird der Einfluss von Design-Entscheidungen beim Entwurf einer Risikoanalysemethodik auf die Qualität der mit der Methodik angefertigten Risikoanalysen untersucht. Insbesondere wird diesbezüglich der Einfluss von in der Risikoanalyse berücksichtigten Eingabeparametern wie etwa Angreiferfähigkeiten, Angreifermotivation und Schwachstellenbewertungen auf Qualitätsfaktoren wie „Genauigkeit“ und „Wiederverwendbarkeit“ der Risikoanalyseergebnisse untersucht.

Darauf aufbauend wird ein Angreiferfähigkeiten-zentrischer Ansatz zur Modularisierung von Risikoanalysen aufgezeigt, der, statt Angreifertypen und deren Motivation, Angreiferfähigkeiten in den Fokus setzt und effizient anwendungsunabhängige und wiederverwendbare Teilergebnisse produziert. Die Wiederverwendbarkeit ermöglicht aufeinander aufbauende Risi-

koanalysen und sorgt so für eine Verringerung der Komplexität und damit einer Steigerung der Nachvollziehbarkeit sowie einer Senkung der Fehleranfälligkeit. Damit wird die weitere Verwendung der Ergebnisse zur Entwicklung von Architekturen und Protokollen mit einem angemessenen Sicherheitsniveau erleichtert.

2. Qualität von Risikoanalyseergebnissen

Die Qualität von Risikoanalyseergebnissen hängt stark von der angewendeten Risikoanalysemethodik ab. Qualität von Risikoanalyseergebnissen hat mehrere Dimensionen, von denen einige im Folgenden eingeführt werden.

- **Genauigkeit:** Die Genauigkeit einer Risikoanalyse ist dadurch definiert, wie akkurat die identifizierten Risiken die Realität wiederspiegeln. In diesem Zusammenhang ist sowohl die Maximierung der als relevant identifizierten Risiken, die tatsächlich relevant sind (Vollständigkeit, Minimierung der false-negatives) als auch die Minimierung der als relevant identifizierten Risiken, die in der Realität keine Relevanz haben (Minimalität, Minimierung der false-positives) von Belang. Risikoanalysemethodiken sollten immer die Zielsetzung haben, möglichst vollständige Ergebnisse zu liefern, um alle Risiken abzudecken. Gleichermaßen sollten Risikoanalysemethodiken zu möglichst keinen irrelevanten Ergebnissen führen, um Kosten für die Implementierung unnötiger Sicherheitsmechanismen zu minimieren.
- **Nachvollziehbarkeit:** Die Nachvollziehbarkeit von Risikoanalyseergebnissen ist dadurch definiert, wie leicht es Dritten fällt, die Ergebnisse der Risikoanalyse und die Begründung der Ergebnisse zu verstehen. Die Qualitätsdimension „Nachvollziehbarkeit“ ist sowohl für das Review von Risikoanalysen durch unabhängige, bei der Erstellung der Risikoanalyse nicht beteiligten Parteien, als auch für Arbeiten, die auf der Risikoanalyse aufbauen, relevant. Dazu zählen insbesondere die Konzeption von Security Architekturen und die Wiederverwendung/Erweiterung der Risikoanalyseergebnisse für zukünftige Anwendungen und Systeme.
- **Vergleichbarkeit:** Die Vergleichbarkeit von Risikoanalyseergebnissen ist dadurch definiert, dass die einzelnen identifizierten Risiken bzgl. ihrer Kritikalität verglichen werden können. Dies betrifft sowohl die Risiken einer in sich geschlossenen Risikoanalyse, als auch Risiken aus mehreren verschiedenen Risikoanalysen die nach derselben Methodik durchgeführt wurden. So macht beispielsweise eine Risikoakzeptanzschwelle „alle Risiken, die höher als ‚mittel‘ kategorisiert sind, müssen behandelt werden“ nur Sinn, wenn zur Bewertung der Risiken dieselben Maßstäbe angesetzt werden. Dabei muss beachtet werden, dass Vergleichbarkeit nur gegeben sein kann, wenn alle Ein-

flussfaktoren auf die Höhe eines Risikos objektiv, nach einem festen Schema bewertet werden. Beispielsweise würden zwei Security-Experten zu unterschiedlichen Ergebnissen kommen, wenn sie beide unterschiedliche Auffassungen des Eintrittswahrscheinlichkeitsbegriffes „hoch“ hätten und nicht klar definiert ist, wann eine Eintrittswahrscheinlichkeit als „hoch“ angesetzt werden darf.

- **Wiederverwendbarkeit:** Risikoanalyseergebnisse können sowohl bei ähnlich aufgebauten Systemen, als auch bei Systemen, die aufeinander aufbauen gegebenenfalls wiederverwendet werden. Beispielsweise sollten optimaler Weise Ergebnisse einer Risikoanalyse, die für einen CAN-Bus angefertigt wurde, in die Risikoanalyse von auf dem CAN-Bus aufbauenden Funktionen integrierbar sein und die Bedrohungen, die zu einer Kompromittierung des CAN-Busses führen, nicht erneut analysiert werden müssen. Risikoanalyseergebnissen sind genau dann gut wiederverwendbar, wenn die Ergebnisse der neuen Risikoanalyse durch die Wiederverwendung nicht negativ beeinflusst werden. Durch die Wiederverwendung von Teilergebnissen kann Arbeitsaufwand eingespart werden und die Vergleichbarkeit von Risikoanalysen erhöht werden.
- **Effizienz**: Die Effizienz von Risikoanalysemethodiken beschreibt den nötigen Arbeitsaufwand im Verhältnis zu dem Umfang des analysierten Systems. In diesem Kontext ist es wünschenswert, dass die Risikoanalysemethodik sinnvolle „Leitplanken“ vorgibt anhand derer sich der Risikoanalyst orientieren kann und somit effizienter zu einem aussagekräftigen Ergebnis kommt. Weiterhin sollten möglichst viele Schritte auch ohne Security Expertenwissen durchführbar sein, um Expertenstunden und damit Kosten zu minimieren.

3. Eingabefaktoren für Risikoanalysemethodiken

Ein Risiko setzt sich zusammen aus der Eintrittswahrscheinlichkeit und dem Schadenspotential. Beispielsweise können Risiken, die einen hohen Schaden nach sich ziehen, durchaus als unkritisch betrachtet werden, wenn die Eintrittswahrscheinlichkeit des Risikos nur sehr gering ist. Sowohl für die Bewertung der Eintrittswahrscheinlichkeit als auch für die Bewertung des Schadenspotentials spielen viele Einflussfaktoren eine Rolle, von denen einige im Folgenden eingeführt werden. In Abschnitt 4 wird diskutiert in wie weit es sinnvoll ist, die Einflussfaktoren bei der die Risikoanalyse einzubeziehen.

Eintrittswahrscheinlichkeit: Damit ein Angriff gegen ein System zustande kommt, muss sowohl eine Schwachstelle im System existieren, die ein Angreifer ausnutzen kann, als auch

ein Angreifer existieren, der fähig und gewillt ist die Schwachstelle auszunutzen (Bedrohung).

- **Bedrohungseintritttswahrscheinlichkeit**: Die Bedrohung auf das System wird durch den Angreifer verkörpert. Die Wahrscheinlichkeit, dass ein Angreifer das System tatsächlich angreift, hängt von mehreren Faktoren ab. Zu diesen zählen insbesondere:
    - **Zugang zum System**: Um bestimmte Angriffe durchzuführen, benötigen Angreifer einen Zugang zum System (bspw. Zugang zum CAN-Bus). Die Wahrscheinlichkeit, dass dieser Zugang besteht, hat Einfluss auf die Eintrittswahrscheinlichkeit des Angriffs.
    - **Erfahrung**: Für die meisten Angriffe benötigt der Angreifer spezifisches Systemwissen oder zumindest Erfahrungen im Reverse Engineering. Um beispielsweise Schwachstellen in proprietären Systemen zu entdecken sind entweder Insider-Informationen nötig oder es müssen Reverse Engineering-Techniken, wie beispielsweise Fuzzing, beherrscht werden.
    - **Ressourcen**: Für viele Angriffe werden neben Systemzugang und Erfahrungswerten Ressourcen in Form von Rechenleistung, Speicher oder Geld benötigt. Beispielsweise benötigen Angreifer zum Brechen von kryptographischen Mechanismen oft erhebliche Rechenkapazität, auch wenn diese im Allgemeinen nicht mehr als sicher gelten.
    - **Angreifer Motivation**: Selbst, wenn ein Angreifer die notwendigen Fähigkeiten und Ressourcen hat einen Angriff durchzuführen, fehlt ihm gegebenenfalls die Motivation dies zu tun, sodass ein Angriff unwahrscheinlich wird.
- **Schwachstelleneintrittswahrscheinlichkeit**: Ob der Angreifer eine Schwachstelle für einen Angriff nutzt, hängt davon ab, ob überhaupt eine Schwachstelle existiert, wie leicht es ihm fällt die Schwachstelle zu entdecken und ob die Schwachstelle zur Erreichung der Ziele des Angreifers geeignet ist.
    - **Angriffsfläche**: In manchen Fällen kann a-priori genau bestimmt werden, ob eine Schwachstelle im System existiert. Es existieren jedoch auch Fälle, in denen a-priori nicht genau bestimmt werden kann ob eine Schwachstelle existiert oder nicht, sodass eine Argumentation über Wahrscheinlichkeiten nötig wird. Beispielsweise kann für Software meistens nicht gänzlich ausgeschlossen werden, dass sie Schwachstellen enthält. Die Wahrscheinlichkeit, dass eine unbekannte Schwachstelle im System existiert, hängt maßgeblich davon ab, wie komplex das System ist, das heißt, wie groß die Fläche ist, auf der ein Angreifer Schwachstellen ausmachen kann.

- **Bekanntheitsgrad der Schwachstelle**: Ein weiterer Einflussfaktor darauf, ob ein Angreifer eine Schwachstelle ausnutzt, ist, ob sie ihm kenntlich ist. Ist beispielsweise eine Schwachstelle bereits in der Common Vulnerability & Exposure Datenbank [7] verzeichnet, so ist die Wahrscheinlichkeit höher, dass sie durch einen Angreifer ausgenutzt wird, als dass die Schwachstelle noch gänzlich unbekannt ist.
- **Bekanntheitsgrad der Teilsysteme**: Die Wahrscheinlichkeit, dass eine Schwachstelle im Laufe des Lebenszyklus eines Produkts aufgedeckt wird, hängt maßgeblich davon ab, ob Standardkomponenten im Produkt verwendet werden. Setzt ein Produkt beispielsweise das verbreitete kryptographische Kommunikationsprotokoll TLS [8] ein, so ist es gegebenenfalls auch von Schwachstellen in TLS betroffen, die bei der Untersuchung von anderen Produkten aufgefallen sind und das Produkt kann so in den Fokus von Angreifern rücken.
- **Skalierbarkeit der Ausnutzung**: Je nach Ziel des Angreifers kann die Tatsache entscheidend sein, ob ein Angriff skalierbar ist, also auf eine große Anzahl von Systemen automatisiert ausgeführt werden kann. Wenn ein Angreifer nur so seine Ziele erreichen kann, eine Schwachstelle jedoch keinen skalierbaren Angriff zulässt, ist die Wahrscheinlichkeit, dass ein Angreifer die Schwachstelle ausnutzt gering.

Schadenspotential: Schäden durch bösartige Angriffe können verschiedenen Auswirkungen auf den Risikoträger haben, die sich je nach Unternehmen unterscheiden. Im Folgenden werden exemplarisch einige im Automotive Kontext gängigen Schadensklassen aufgelistet:

- **Direkte finanzielle Schäden**: beispielsweise Rückrufaktionen, direkte Schadensersatzforderungen oder Reparaturkosten**.**
- **Rechtliche Schäden**: beispielsweise Verletzungen des Datenschutzrechts oder Verletzung der Sorgfaltspflichten aus dem Produktsicherheitsgesetz.
- **Vertragliche Schäden**: beispielsweise Vertragsstrafen oder Aufkündigungen
- **Schäden am geistigen Eigentum**: beispielsweise durch Produktfälschungen oder Stärkung von Konkurrenzprodukten
- **Reputationsschäden**: beispielsweise bedingt durch verringerte Effizienz, Verlässlichkeit und Langlebigkeit der Produkte oder Bedrohung der Safety.
- **Skalierbarkeit der Ausnutzung**: Falls skalierbare Angriffe auf eine Produktlinie möglich sind, steigt das Schadenspotential stark an, da nicht einzelne Produkte, sondern die ganze Produktlinie betroffen ist. Beispielsweise ist das Schadenspotential für einen

Denial-of-Service Angriff auf ein einzelnes Fahrzeug, der über eine Bluetooth-Verbindung ausgeführt wird, weitaus geringer als das Schadenspotential eines Denial-of-Service-Angriffs über die LTE Verbindung, der die ganze Fahrzeugflotte betrifft.

4. Zusammenhang Qualitätsdimensionen & Eingabefaktoren

Die identifizierten Risikoeinflussfaktoren können prinzipiell alle in Risikoanalysemethodiken berücksichtigt werden. Dabei dient in der Praxis jeder Einflussfaktor der Minimierung von Risikoeinschätzungen, um eine Annäherung der realistisch bestehenden Risiken zu erreichen und die Menge der Security Mechanismen, die implementiert werden müssen, möglichst zu reduzieren. Es ist zu beachten, dass mit jedem betrachteten Einflussfaktor Fehleinschätzungen einhergehen können, sodass Risiken, die bestehen, nicht betrachtet werden. Dieser Dualismus zwischen Unter-/Überapproximation der real existierenden Risiken ist in Bild 1 visualisiert. Weiterhin können durch die Berücksichtigung von Einflussfaktoren Abhängigkeiten zum konkreten Anwendungsfall geschaffen werden, die eine Wiederverwendung der Ergebnisse erschweren. In der Praxis hat der Einbezug von bestimmten Einflussfaktoren in der Risikoanalysemethodik somit Auswirkungen auf die Qualität der Risikoanalyseergebnisse. Im Folgenden wird diskutiert, welche Einflussfaktoren kritische Nebenwirkungen auf die Ergebnisse von Risikoanalysen haben können.

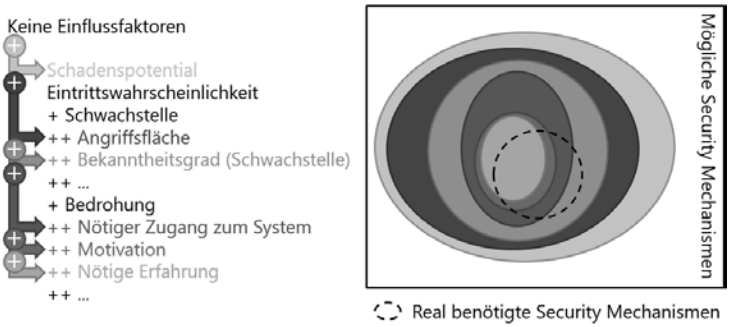

Bild 1: Über-/Unterapproximation von real existierenden Risiken durch Berücksichtigung von Einflussfaktoren © ITK Engineering GmbH

Im Folgenden werden zunächst die Auswirkungen von Einflussfaktoren auf die **Genauigkeit** der Risikoanalyseergebnisse diskutiert. Ziel einer guten Risikomanagementmethodik ist es, Security Anforderungen zur Erreichung eines angemessenen Sicherheitsniveaus zu identifizieren, das erreicht ist, wenn Security-Anforderungen minimal bezüglich aller denkbaren und vollständig bezüglich aller relevanten Risiken sind, die Risikoanalyseergebnisse also „genau“

sind (siehe Abschnitt 2). Aufgrund der gegebenen Komplexität und unvollständiger Informationsbasis ist es in der Praxis schwer bis unmöglich ein angemessenes Sicherheitsniveau für ein Szenario scharf zu identifizieren. Durch Berücksichtigung von Einflussfaktoren kann bei der Erstellung einer Risikoanalysemethodik der Fokus jedoch mehr auf Minimalität oder auf Vollständigkeit ausgeprägt werden.

Diskussion – Erfahrung: Die nötige Erfahrung eines Angreifers zur Durchführung eines Angriffs beschränkt die Anzahl von Angreifern, die einen Angriff das erste Mal durchführen können. Sobald der Angriff von einem erfahrenen Angreifer durchgeführt wurde und entsprechende Informationen/Anleitung publiziert wurden, können die Angriffe allerdings auch weniger erfahrene Angreifer durchführen. Das kann die Genauigkeit von Risikoanalyseergebnissen negativ beeinflussen, falls Risiken aufgrund der nötigen Angreifer-Erfahrung als unkritisch eingestuft wurden, die eigentlichen weitflächigen Angriffe dann aber von relativ unerfahrenen Nachahmungstätern durchgeführt werden.

Diskussion - Motivation: Die Motivation des Angreifers wird oft herangezogen, um die Notwendigkeit von wirkungsvollen Security-Maßnahmen zu hinterfragen und so Kosten einzusparen. Allerdings ist eine Bewertung der Angreifermotivation in der Praxis oftmals *spekulativ* und stark beeinflusst durch die Vorstellungskraft des Risikoanalysten. Weiterhin ist Angreifermotivation *volatil* im Vergleich zu anderen Einflussfaktoren. Beispielsweise stellen mittlerweile Verschlüsselungstrojaner in der IT-Welt eine der wichtigsten Bedrohungen dar, während sie vor 5 Jahren zwar auch schon existierten, aber von Angreifern nicht oft eingesetzt wurden.

Diskussion - Zugang zum System: Die Einschätzung, ob ein Angreifer bestimmte Zugänge zum System besitzt, ist in der Praxis oft schwierig. Diesem Problem kann begegnet werden, in dem die angenommenen Zugänge des Angreifers als „Security Annahmen" formuliert werden, unter denen die Risikoanalyseergebnisse Gültigkeit besitzen. Derartige Security Annahmen sind auch bei der Ausarbeitung des Security Konzepts nötig. Beispielsweise muss oft die Annahme getroffen werden, dass kryptographische Verfahren wie RSA sicher sind. Die Security Annahmen, die während der Risikoanalyse getroffen werden, sollten mit den Security Annahmen, die bei der Entwicklung des Konzepts getroffen wurden, vereint dokumentiert werden.

Eine weitere Methode zur Bestimmung der Wahrscheinlichkeit, mit der ein Angreifer bestimmte Zugänge zu Teilsystemen besitzt, ist die Wiederverwendung der Bedrohungs-/ Schwachstellenanalyse der entsprechenden Teilsysteme, die beschreiben, wie wahrscheinlich es ist, dass ein Angreifer in der Lage ist, das Teilsystem zu kompromittieren. Um diese

Methode anzuwenden ist es nötig, dass Bedrohungs-/Schwachstellenanalysen *anwendungsfallunabhängig* durchgeführt werden (siehe Diskussion – Wiederverwendbarkeit).

Diskussion – Bekanntheitsgrad der Schwachstelle: Der Bekanntheitsgrad einer Schwachstelle hat kurzfristig großen Einfluss auf die Eintrittswahrscheinlichkeit eines entsprechenden Angriffes. Allerdings ist der Bekanntheitsgrad von Schwachstellen volatil und kann sich beispielsweise durch die Veröffentlichung von Informationen durch erfolgreiche Angreifer schnell ändern.

Neben der Genauigkeit ist auch die **Wiederverwendbarkeit** von Risikoanalyseergebnissen stark abhängig von den Einflussfaktoren, die in der Methodik berücksichtigt werden. Ermittelte Schadenspotentiale sind zwangsläufig vom Anwendungsfall abhängig und können in keinem Fall wiederverwendet werden. Eintrittswahrscheinlichkeiten dagegen schon, sofern die Risikoanalysemethodik keine anwendungsfallspezifischen Überlegungen in die Ergebnisse einfließen lässt. Beispielsweise können Risikoanalyseergebnisse, die für einen CAN-Bus über den unkritische, für den Angreifer weitgehend uninteressante Funktionen realisiert werden, nicht für einen CAN-Bus, auf dem kritische Funktionen aufbauen, übernommen werden, falls die anwendungsfallspezifische Angreifermotivation mit in die Bewertung eingeflossen ist.

Um die **Vergleichbarkeit** von Risikoanalyseergebnissen sicherzustellen, ist es bei der Auswahl von Einflussfaktoren wichtig, dass diese objektiv bewertbar sind. Idealerweise sollte jeder Einflussfaktor anhand einer objektiven Skala vom Risikoanalyst bewertbar sein, ohne dass individuelle Auffassungen über die Bedeutung der einzelner Bewertungsmöglichkeiten Einzug finden. Beispielsweise kann für den Faktor Angriffsfläche in Bezug auf Software Exploits die Anzahl der Codezeilen oder die Anzahl der Schnittstellen nach außen als objektiver Maßstab dienen. Eine weitere Möglichkeit, die Vergleichbarkeit von Risiken sicherzustellen, ist die Wiederverwendung der Risikoanalyseergebnisse von Teilsystemen für mehrere übergeordnete Systeme (siehe Wiederverwendbarkeit). Beispielsweise können Risikoanalyseergebnisse für einen CAN-Bus für die Risikoanalysen mehrerer Funktionen, die auf dem CAN-Bus realisiert werden, wiederverwendet werden und somit ausgeschlossen werden, dass die Risiken, die durch Schwachstellen der CAN-Bus Kommunikation entstehen, für jede Funktion unterschiedlich bewertet wird und somit die Ergebnisse nicht mehr vergleichbar sind.

Um **Nachvollziehbarkeit** der Risikoanalyseergebnisse sicherzustellen, muss bei der Entwicklung der Methodik darauf geachtet werden, dass die Komplexität der Risikoanalyseergebnisse soweit wie möglich reduziert wird. Dabei ist neben der strukturierten Darstellung der Ergebnisse die weitgehend unabhängig von der eigentlichen Methodik ist auch die Ergebnisstruktur, die von der Methodik vorgegeben wird, von Belang. Beispielsweise sehen

viele Methodiken Systeme als monolithisch an und bewerten sie als eine große Komponente. Während für Security Bewertungen immer das Gesamtsystem betrachtet werden sollte, führt dies schon bei Systemen mittlerer Komplexität zu Risikoanalyseergebnissen, die weder Außenstehende noch der Risikoanalyst selbst in der Gänze durchdringen und nachvollziehen kann. Dementsprechend sollten Risikoanalysemethodiken die Modularisierung vom Kern auf unterstützen und es erlauben Systeme in Teilsysteme zu unterteilen, um die Ergebnisse der Teilsysteme in einer Gesamtsystembetrachtung zusammenfließen zu lassen (siehe auch „Wiederverwendbarkeit“).

5. Angreiferfähigkeiten-zentrische Risikoanalysemethodik

Basierend auf den bisherigen Erkenntnissen wird im Folgenden das in Bild 2 dargestellte Rahmenwerk für Risikoanalysen vorgestellt. Ausgehend von den System- und Feature-Beschreibungen sowie den Organisationszielen werden schützenswerte Features und entsprechende Misuse-Cases identifiziert. Jeder Misuse-Case geht mit einem entsprechenden Schadenspotential einher, das sich für das Unternehmen ergibt. Um einen Misuse-Case herbeizuführen muss ein Angreifer Schutzziele bzgl. Informationswerten aushebeln. Beispielsweise muss die Integrität von CAN-Nachrichten untergraben werden, um eine Fahrzeugfunktion fälschlicherweise auszulösen. Dem Aushebeln der Schutzziele geht immer die Ausnutzung einer Schwachstelle voraus. Dazu muss eine entsprechende Schwachstelle vorhanden sein und der Angreifer muss ggf. notwendige Fähigkeiten (Zugang zum System, notwendige Ressourcen) zur Ausnutzung der Schwachstelle besitzen. Dieses Herunterbrechen der Misuse-Cases auf Schwachstellen im System und nötige Angreiferfähigkeiten kann beispielsweise durch die Modellierung in Form von Angriffsbäumen geschehen. Basierend auf dieser Modellierung können aus der Wahrscheinlichkeit, dass eine Schwachstelle existiert und der Wahrscheinlichkeit, mit der Angreifer die notwendigen Fähigkeiten besitzen diese auszunutzen, die Eintrittswahrscheinlichkeit des Misuse-Cases ermittelt werden. Gepaart mit dem Schadenspotential, das hinter dem Misuse-Case steht, ergibt sich somit ein Risiko.

Zur Bestimmung der Bedrohungseintrittswahrscheinlichkeit und der Schwachstelleneintrittswahrscheinlichkeit werden Bewertungsrichtlinien für jeden Einflussfaktor festgelegt, die eine objektive Bewertung jedes Faktors erlauben. Über einheitliche Bewertungsmodelle können die resultierenden Wahrscheinlichkeiten abgeleitet werden und so die Vergleichbarkeit der Risikoanalyseergebnisse sichergestellt werden.

Basierend auf den Erkenntnissen, die im vorherigen Abschnitt dargelegt wurden, sind bewusst die folgenden Designentscheidungen getroffen:

- Vermeidung von spekulativen und volatilen Einflussfaktoren: Für nicht modellierte Einflussfaktoren werden implizit worst-case Annahmen getroffen und so ein sicheres Sys-

tem durch fehlerhafte Risikoanalyseergebnisse vermieden. Für die Angreifermodellierung werden nur die Faktoren „Zugang zum System“ sowie „Ressourcen“ und für die Schwachstellenmodellierung „Angriffsfläche“, „Bekanntheitsgrad der Teilsysteme“ sowie „Skalierbarkeit der Angriffe“ berücksichtigt. Insbesondere wird hierfür auf Bewertungsgrößen, wie beispielsweise „Erfahrung“ und „Motivation“ aus den oben diskutierten Gründen verzichtet.

- Klare Trennung zwischen Modellierung der Eintrittswahrscheinlichkeit und des Schadenspotentials: Die anwendungsfallspezifischen Modelle zur Bestimmung des Schadenspotentials werden klar abgegrenzt, um die Modelle zur Bestimmung der Eintrittswahrscheinlichkeit anwendungsfallunabhängig zu halten.
- Keine Berücksichtigung anwendungsfallspezifischer Einflussfaktoren zur Bestimmung der Eintrittswahrscheinlichkeit: Eintrittswahrscheinlichkeiten können somit in anderen Risikoanalysen wiederverwendet werden, sofern sie dort relevant sind. Weiterhin ermöglicht die Wiederverwendung eine Modularisierung der Risikoanalysen und senkt somit die Komplexität der Ergebnisse, was zu einer Steigerung der Nachvollziehbarkeit führt. Beispielsweise kann bei der Bewertung von Angriffen auf eine bestimmte Fahrzeugfunktion über den CAN-Bus auf die Risikoanalyse des CAN-Busses verwiesen werden, statt Angriffe auf den Bus aufzuschlüsseln und so die Ergebnismenge unnötig zu vergrößern. Durch Nicht-Berücksichtigung von anwendungsspezifischen Einflussfaktoren kommt es zu einer Überapproximation von Risiken. Dies führt zu einem sichereren Gesamtsystem, da mehr Security-Mechanismen im Konzept implementiert werden, steigert allerdings auch die Kosten. Um dem entgegenzuwirken, können in der Risikobehandlung kostenkritische Risiken im Nachgang zur Risikoanalyse anwendungsfallspezifisch betrachtet und ggf. herabgestuft werden.
- Wiederverwendung von Teilergebnissen: Teilergebnisse können und sollen nach Möglichkeit wiederverwendet werden, um Nachvollziehbarkeit, Vergleichbarkeit und Effizienz von Risikoanalysen zu erhöhen. Die Kompromittierung eines Teilsystems stellt einen Misuse-Case in der Risikoanalyse des Teilsystems dar. In der Risikoanalyse des übergeordneten Systems entspricht die Wahrscheinlichkeit, dass ein Angreifer die Fähigkeit besitzt ein Teilsystem zu kompromittieren damit der Wahrscheinlichkeit, dass der Misuse-Case im Teilsystem eintritt, und kann der Risikoanalyse des Teilsystems entnommen werden.

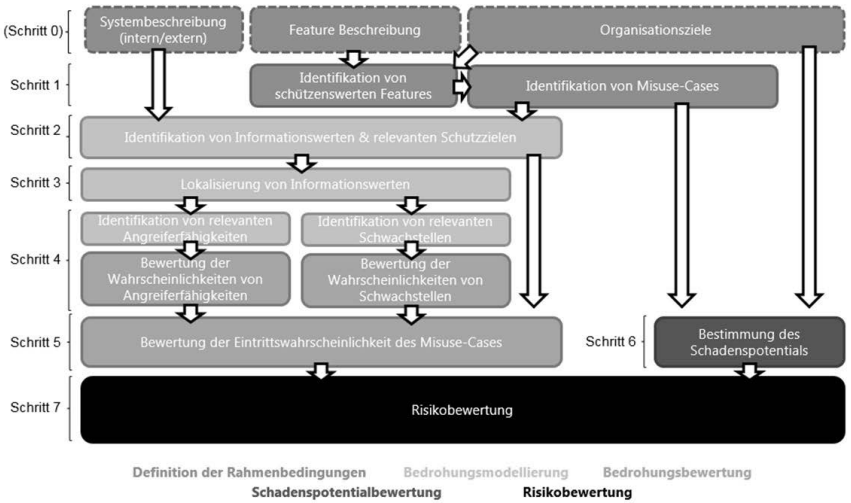

Bild 2: Angreiferfähigkeiten-zentrische Risikoanalysemethodik © ITK Engineering GmbH

6. Fazit

Der Beitrag hat gezeigt, dass eine Vielzahl von Einflussfaktoren relevant für die Bewertung von Risiken sind und damit auch für Risikoanalysemethodiken sein können. Durch das Hinzunehmen von zusätzlichen Einflussfaktoren wird dabei die Relevanz von Bedrohungen und Schwachstellen zunehmend minimiert. Dadurch werden Risiken in ihrer Kritikalität herabgestuft, weniger Security Maßnahmen müssen getroffen werden und Aufwände können eingespart werden. Demgegenüber steht der Genauigkeitsverlust, der sich durch Fehleinschätzungen der einzelnen Einflussfaktoren für konkrete Risiken zunehmend einschleicht. Weiterhin beeinflusst die Berücksichtigung von weiteren Einflussfaktoren Qualitätsmerkmale wie Wiederverwendbarkeit und Nachvollziehbarkeit. Somit existiert ein Trade-Off zwischen Kostenersparnispotential und Security, der bei der Entwicklung einer Risikoanalysemethodik sorgsam ausgewogen werden muss.

Diesbezüglich hat der Artikel aufgezeigt, dass insbesondere für spekulative und volatile Einflussfaktoren wie etwa die Angreifermotivation stark überlegt werden sollte, ob diese in die Risikoanalyse einfließen sollten. Weiterhin sollte beachtet werden, dass viele Einflussfaktoren Annahmen über den Anwendungsfall des Systems erzwingen. Die Anwendungsfälle für viele Systeme sind jedoch einer ständigen Evolution unterworfen und teilweise werden Systeme ganz bewusst für gänzlich andere Anwendungsfälle eingesetzt. Um Aufwände zur An-

passung/erneuten Durchführung der Risikoanalyse zu vermeiden, sollten zumindest Teile der Risikoanalysemethodik wie etwa die Schwachstellen und Bedrohungsanalyse anwendungsfallunabhängig gehalten werden um eine Wiederverwendung und eine Komposition von Risikoanalyseergebnissen zu ermöglichen.

[1] Jens Köhler, Henry Förster. "Trusted Execution Environments im Fahrzeug" *ATZelektronik* 11.5 (2016): 38-43.

[2] Jens Köhler, Jochen Breidt. "Sichere Fernwartung von Steuergeräten in Landmaschinen" *74. Internationale Tagung LAND.TECHNIK* (2016).

[3] Jens Köhler. "Secure Remote Diagnostics for Electronic Control Units in Off-Highway Machines" 2nd International VDI Conference: Connected Off-Highway Machines (2017).

[4] Rosenquist, Matthew. "Prioritizing information security risks with threat agent risk assessment." Intel Corporation White Paper (2009).

[5] Wynn, Jackson, et al. Threat assessment & remediation analysis (TARA): Methodology description version 1.0. No. MTR110176. MITRE CORP BEDFORD MA, 2011.

[6] Caralli, Richard A., et al. Introducing octave allegro: Improving the information security risk assessment process. No. CMU/SEI-2007-TR-012. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2007.

[7] Common Vulnerabilities and Exposures - https://cve.mitre.org/

[8] Dierks, Tim. "The transport layer security (TLS) protocol version 1.2." (2008).

Kriminelle Energie

Treibstoff für das Automotive Security Engineering

Dipl.-Ing. **Jürgen Belz**, PROMETO GmbH, Paderborn

Kurzfassung

Gäbe es keine Hacker, würde niemand auch nur einen Cent in Automotive Cyber Security ausgeben. Nun geben Unternehmen Geld für Security aus. Dabei stellt sich die Frage, ob die Mittel auch tatsächlich zielgerichtet eingesetzt werden und die im Markt erhältlichen Sicherheitslösungen tatsächlich ihr Geld wert sind. Als Security-Berater weiß ich: ein simples Schwarzweißdenken wird der Frage nicht gerecht. Man muss differenziert vorgehen und bereit für einen Perspektivwechsel sein. In meinem Vortrag greife ich das Thema Automotive Cyber Security aus dem Blickwinkel eines Hackers auf. Im Ergebnis steht die Erkenntnis, dass Entwicklungsbereiche kriminelle Energie brauchen, um sichere Produkte zu entwickeln.

Hacker sind nicht automatisch kriminell

Hacking wird allgemein mit digitalem Vandalismus, Erpressung, Datendiebstahl und anderen Straftaten assoziiert. Übersehen wird allerdings, dass die Wortbedeutung eine andere ist. Der Begriff entstand in den 50ern des letzten Jahrhunderts am MIT, als man eine alte Telefonanlage zur Steuerung einer Modelleisenbahn umfunktionierte. Die meisten Menschen hätten in einer Telefonanlage lediglich eine Telefonanlage gesehen und keine Eisenbahnsteuerung. Hacker haben eine besondere Gabe, alternative Anwendungsmöglichkeiten zu sehen, die den meisten Menschen verborgen bleiben. Hacking ist im Wortsinn die Gabe, sehr ungewöhnliche und alles andere als naheliegende Lösungsansätze zu finden.

Security Engineering Prozesse alleine bieten keinen ausreichenden Schutz

Die Entwicklungsprozesse der meisten Unternehmen zielen auf die sog. best practices ab, also auf bewährte Konstruktionsprinzipien. Sie sind in Normen und Standards festgeschrieben. Die Norm und damit der Normalfall ist es nun mal, dass ein Entwickler bei der Lösung einer Entwicklungsaufgabe – egal wie begabt oder untalentiert er ist – die gleichen Methoden aus einem Standard wählt wie jeder andere Entwickler und sie auch genauso gleich anwendet wie jeder andere auch. Alles andere wäre abnormal. Wer sich abnormal verhält, wird bestraft, denn man gilt – ausgesprochen oder nicht – irgendwie als Perversling. Der starre

und vor allem lineare Ansatz der Unternehmensprozesse ist exakt das Gegenteil von der besonderen Gabe der Hacker, alternative Anwendungsmöglichkeiten zu sehen. In der Folge entfalten die Security Engineering Prozesse in der Praxis eine deutlich geringere Wirkung als erhofft. Die Tatsache, dass Security Engineering Prozesse keinen besonders signifikanten Beitrag bei der Entstehung sicherer Produkte leisten, bedeutet nicht zwingend, dass man auf diese Prozesse verzichten sollte, im Gegenteil. Allerdings sollte man sich der Grenzen der Prozesse klar bewusst sein.

Das Minimum an Engineering-Prozessen

Abbildung 1 zeigt ein einfaches Beispiel eines Lasten- und Pflichtenheftsatzes, das in PTC Integrity abgebildet wurde. In der oberen Hälfte findet sich das Lastenheft eines OEMs wieder, der einfach nur ein sicheres Produkt haben will. Die untere Hälfte zeigt das korrespondierende Pflichtenheft mit vier Pflichten. Mit PTC Integrity lässt sich auf einfache Weise die prozessorale Forderung nach bilateraler Traceability erfüllen – erkennbar am Trace Status rechts unten in der Abbildung. Meine Empfehlung lautet stets, die Security Prozesse in die normalen Entwicklungsprozesse einzubetten und nicht nebenläufig Security vom Rest des Geschehens abzukoppeln. Security stellt auch und vor allem ein inhaltliches Element des Requirements-Engineerings dar.

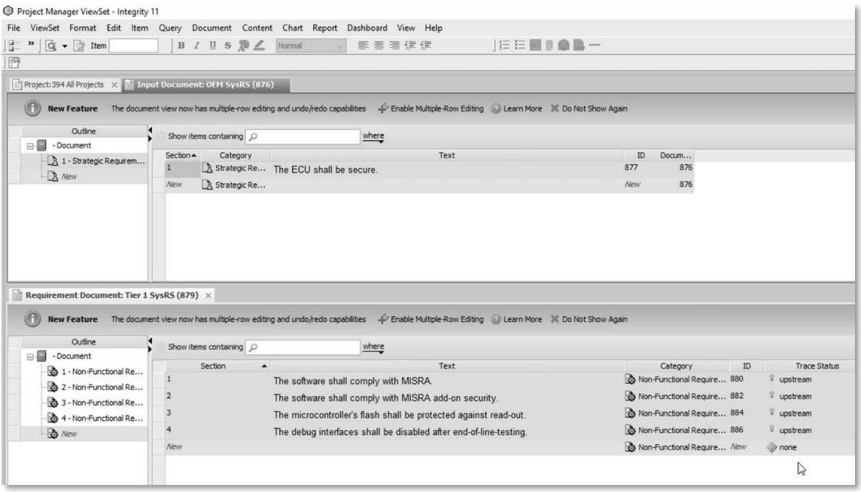

Bild 1: Beispiel Lasten- und Pflichtenheft

Die vier Pflichten stellen zentrale Elemente dar, die in der Entwicklung Berücksichtigung finden sollten. Das sind zum einen Anforderungen an die Gestaltung von Software, die über Methoden der Statischen Code-Analyse nachgewiesen werden (Pflichten 1 und 2). Auch sollte man tunlichst verhindern, dass die Software einer ECU ausgelesen und mit Methoden des Reverse Engineering von Dritten nachvollzogen werden kann (Pflicht 3). Ebenso ist es im Sinne der Sicherheit, wenn die Zugänge auf interne Bereiche der ECU versperrt werden (Pflicht 4). Aus unserer Beratungspraxis wissen wir, dass gerade der letzte Punkt einen Konflikt auslöst, denn die Ansätze von Design for Manufacturing sind vollkommen entgegengesetzt zu den Anforderungen der Security. Schließlich möchte man ja an Reparatur-Arbeitsplätzen oder bei Rückläufern auf die Zustände im Inneren der ECU zugreifen. In der Security möchte man bewusst diese Zugänge dichtmachen. In der Praxis gewinnt zumeist der, der die geringsten sichtbaren Kosten verursacht, also Design for Manufacturing und damit der Hacker.

Naivität, Arroganz, Ignoranz und Selbstüberschätzung

Ich höre häufig, dass die Schnittstellen einer ECU nicht offengelegt werden. Folglich sei Design for Manufacturing unproblematisch, denn Hacker kommen da sowieso nicht hinein, weil sie ja nichts über die verwendeten Mechanismen wissen. Diese Plattitüde kann an Naivität kaum übertroffen werden. Nach meiner Erfahrung kann ich sagen, dass diese Haltung sich zumeist als Folge eines fundamental falschen Feindbildes ausprägt: der Hacker, das singuläre Wesen. Diese Simplifizierung auf eine Person ist insofern tückisch, weil es vielfach das Bild eines Super-IT-Cracks suggeriert, den es so in der Wirklichkeit gar nicht gibt. Analysiert man die in der Presse ausführlich dargestellten Sicherheitslücken in PKWs u.a. der Marken BMW, Fiat Chrysler, GM, Mitsubishi, Tesla und VW, dann sind zwei Hacker-Gruppen erkennbar. Zum einen sind es Forscher, die Sicherheitslücken veröffentlichen. Zum anderen nutzen manche IT-Sicherheitsunternehmen Veröffentlichungen zu Marketing-Zwecken. Die dritte Gruppe sind gewinnorientierte Unternehmen, die zum Teil unabhängig von juristischen Aspekten der Legalität wirtschaftliche Interessen verfolgen. Diese Gruppe zeigt zumeist keinen Bedarf an zu viel Öffentlichkeit, da dies schlecht fürs Geschäft ist. Diese Gruppe wird übrigens gerne in den Diskussionen übersehen. Von Unternehmen aus dieser Gruppe kann man beispielsweise Geräte mit Aktualitätsgarantie über Fahrzeug-Hersteller und Modellreihen hinweg beziehen, um Tachos zu manipulieren. Die eigentliche Straftat in diesem Bereich begehen dann Kleinkriminelle, die weder über eine ausreichende Sachkunde in IT verfügen noch diese überhaupt brauchen. Es ist die Sorte von Kleinkriminellen, die früher mit einem Schraubendreher Autotüren aufgebrochen und Gegenstände entwendet hat. Wie in der nor-

malen IT auch, liegt es im wirtschaftlichen Interesse der Anbieter, Anwendern möglichst einfach zu nutzende Werkzeuge an die Hand zu geben. Das Internet ist voll mit Anleitungen und Werkzeugen, die sich für kriminelle Aktivitäten nutzen lassen. Die Werkzeuge runter zu laden und für einen konkreten Anwendungsfall hin nutzbar zu machen, dauert häufig nur wenige Stunden. Auch vernetzen sich Hacker, tauschen sich aus und arbeiten somit team- und lösungsorientiert zusammen.

Unternehmen übersehen gerne, dass die meisten Taten von Insidern ausgeführt werden. Grob schreibt man zwei Drittel aller Taten Insidern zu, wie sich in Abbildung 2 erkennen lässt. Die Abbildung zeigt zwei unterschiedliche Statistiken. Die Gesamtsumme >100% erklärt sich übrigens dadurch, dass manchmal der Innentäter und der Außentäter zusammenarbeiten. Bei gezielten Angriffen verlassen sich die Angreifer nicht nur auf ihr technisches Können, sondern sie wissen auch menschliche Schwächen wie das Bedürfnis nach Anerkennung, Habgier, Alkoholsucht und sexuelle Vorlieben auszunutzen. Diesen Aspekt der Cyber Security nennt man Social Engineering, was auch im Automotive-Bereich bereits zu größeren finanziellen Schäden geführt hat.

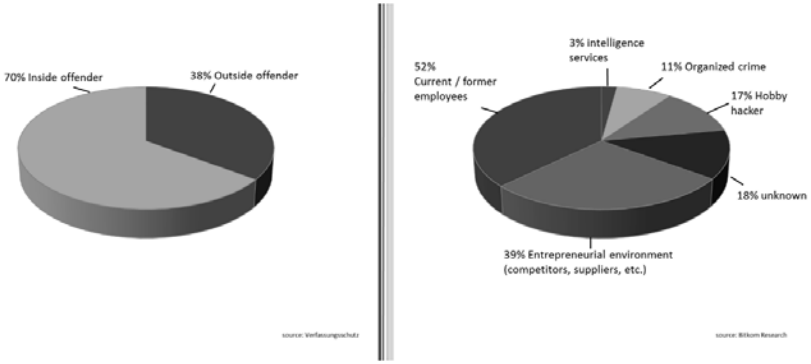

Bild 2: Tätergruppen

Wer Insider ist, verfügt grundsätzlich über die notwendige Expertise um ein System zu hacken. Automotive-Unternehmen können die Expertise, die für einen Hackerangriff erforderlich ist, nicht beeinflussen, auch wenn sie gerne in ihrer Überheblichkeit etwas anderes behaupten. Schließlich lassen sich die Experten für ein Thema schnell über XING, LinkedIn, Facebook und andere soziale Medien finden. Mit Methoden des Social Engineerings macht man deren Expertise dann nutzbar.

Betrachtet man die gängigen Automotive-Security-Risikoassessments wie EVITA, erkennt man eine weitere Illusion, nämlich Einflussnahme der Unternehmen auf die Ausrüstung, die für einen Hack erforderlich wird. Dabei wird übersehen, dass die meisten Werkzeuge sich für wenig Geld beschaffen lassen. Auch beliebt ist die Nutzung der Werkzeuge des Arbeitgebers „nach Feierabend“. Allerdings sollte man wissen, dass im Bereich der Automotive-Security nur wenige Werkzeuge existieren, die sich direkt einsetzen lassen. Man muss gelegentlich schon mal zum Lötkolben greifen und auch einiges skripten. Die Fähigkeit, sich die erforderlichen Betriebsmittel zu bauen, gehört übrigens zu den wesentlichen Merkmalen der Hacker. Man kann also viel über die Qualität der wahren Expertise eines Automotive Security Experten erfahren, wenn man ihn nach seinen Werkzeugen befragt.

Security bedeutet, Angriffe unwirtschaftlich zu machen

Hacker benötigen für ihre Arbeit Expertise, Ausrüstung und Zugänge zum Auto. Als Konsequenz der zuvor gemachten Ausführungen zu Expertise und Ausrüstung bleibt einzig, den Zugang zum Auto so beschwerlich wie möglich zu machen. Das setzt aber voraus, dass Entwickler detailliert und ständig aufs Neue lernen, wie Hacker sich Zugänge verschaffen. Analysiert man Expertise, Ausrüstung und Zugang genauer, dann zeigt sich die in Abbildung 3 illustrierte Tatsache, dass die zur Verfügung stehenden Ressourcen in Summe sich von günstig bis teuer eingruppieren lassen. Als Konsequenz steht der Aufwand dem Nutzen gegenüber. Security ist folglich gegeben, wenn der wirtschaftliche Vorteil aus Aufwand und Nutzen nicht gegeben ist. Das Perfide dabei ist, dass der tatsächliche Aufwand mit den Fähigkeiten der Hacker stark variiert.

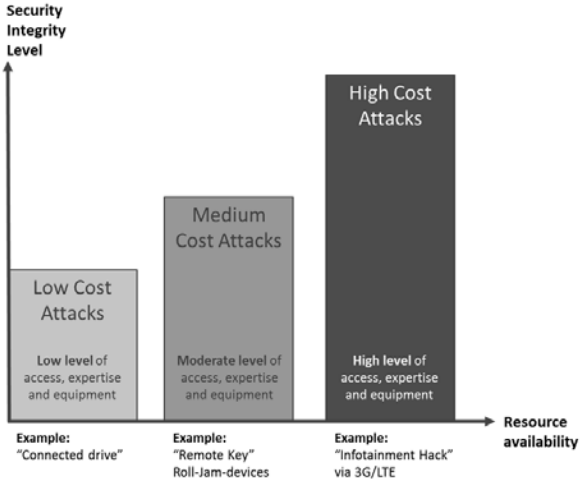

Bild 3: Expertise, Ausrüstung und Zugang

Abbildung 3 listet drei Beispiele auf, auf die ich hier kurz eingehen möchte. Die „low cost attacks" waren rein mit einem Laptop aus der Ferne möglich. Durch haarsträubende Anfängerfehler des OEMs konnte ein Angreifer, der nur Grundlagen der IT beherrschen musste, die Remote-Dienste am Fahrzeug nutzen. Bei der „medium cost attack" reicht etwas Wissen über Funktechnik und frei käufliche Hardware für 50€ aus, um sich Zugang zu Autos zu verschaffen, die über Funkschlüssel verriegelt werden. Bei der „high cost attack" ersetzt man das Handy-Funknetz durch einen Rogue Access Point. So lassen sich etwaige Schwachstellen der im Auto verbauten Mobilfunkchips ausnutzen. Zusammenfassend sei gesagt, dass mit jedem Beispiel der Anspruch an die Fertigkeiten und gleichermaßen die Kosten steigen. Ob der Aufwand sich lohnt, hängt einzig und allein vom Business Modell der Hacker und nicht von den etwaigen einfältigen Realfiktionen der OEMs ab.

Risiko-Bewertung

Es erscheint naheliegend, Security-Risiken wie auch sonstige Geschäftsrisiken in Hinblick auf Eintrittswahrscheinlichkeit und Auswirkungen zu untersuchen, so wie Tabelle 1 dies zeigt.

Tabelle 1: Risikobewertungsschema

Gesamtes Risikoausmaß				
Auswirkung	Hoch	Mittel	Hoch	Kritisch
	Mittel	Niedrig	Mittel	Hoch
	Niedrig	Note	Niedrig	Mittel
		NIEDRIG	MITTEL	HOCH
	Wahrscheinlichkeit			

Ein branchenspezifischer Ansatz, eine Risikobewertung in Bezug auf Automotive zu machen, wurde im Rahmen des EVITA-Projektes erarbeitet. EVITA steht übrigens für **E**-Safety **V**ehicle **I**n**T**rusion Protected **A**pplications. Die Tabelle 2 zeigt ein beliebiges Beispiel dieser Bewertungsmethode für irgendeine Funktion. Bei EVITA ist die Wahrscheinlichkeit als Attack Potential und die Auswirkung als Severity gekennzeichnet. Beide besitzen jeweils fünf Kriterien. Ihre Quantifizierung führt zu einer differenzierteren Betrachtung der Risiken.

Tabelle 2: Beispiel einer Risikobewertung

Attack Potential:		Severity:	
Elapsed Time:	2	safety:	5
Expertise:	4	privacy:	2
Knowledge:	5	financial:	2
Opportunity:	2	operational:	2
Equipment:	2	controllability:	4

Die Risiko-Bewertung ist nach innen ins Unternehmen gerichtet und befriedigt Prozessvorgaben. Die Sache hat einen Schönheitsfehler, denn kein Hacker interessiert sich für die Risiko-Bewertung. Dass die EVITA-Kriterien vielleicht etwas problematisch in Hinblick auf die Natur des Hackers sowie in Hinblick auf Wissen und Ausrüstung sind, wurde bereits deutlich. EVITA ist auch noch aus einem anderen Grund problematisch, denn die Bewertung wird von den sogenannten Experten im Unternehmen durchgeführt. Diese sind aber selten Hacker und damit auch keine Experten in der Beurteilung der Risiken. Als Folge können die Risikobewertungen auch nur laienhaft ausgeführt sein. Trotzdem rate ich dazu, Risiken im Rahmen einer Threat Analysis and Risk Assessment (TARA) zu bewerten. Schon jetzt ist erkennbar, dass Fortbildungen im Bereich Hacking erforderlich sind. Für Security-Risiken gilt das Gleiche wie für Risiken im Projektmanagement oder sonst wo. Risiken zeigen lediglich potentiel-

le Bedrohungen. Man ist also gut beraten, sinnvolle Maßnahmen zu ergreifen, so dass aus einem Risiko kein eingetretenes Ereignis, also ein Problem wird.

Sinnvolle Maßnahmen

Die Kunst im Risiko-Management besteht offensichtlich darin, geeignete Maßnahmen zu ergreifen, damit ein Risiko nicht eintritt und zum Problem mutiert. Im Moment wird gerade viel Geld damit verdient, CAN FD in die Fahrzeugarchitekturen einzuführen, weil damit CAN-Netzwerke sicher sind. Hacker hätten so keine Chance, zumindest nach Einschätzung der Security-Experten, die entsprechende Lösungen anbieten. Nehmen wir diese Einschätzung sportlich und widerlegen sie.

Wir veranstalten Trainings im Bereich Automotive Cyber Security. Am ersten Tag im Einsteiger-Training analysieren wir genauer den Toyota Prius Hack von Charly Miller und Chris Valasek, der über das Forbes Magazine brillant in Szene gesetzt wurde. Bei diesem Hack sitzen die beiden auf der Rücksitzbank und greifen auf das CAN-Netzwerk des Fahrzeugs zu.

Dieser Hack wie auch viele andere der vergangen Hacks beruhen auf dem gleichen Prinzip, nämlich der Übernahme der Kontrolle über das CAN-Netzwerk im Fahrzeug. Dabei löst der Hacker die gewünschten Funktionen aus, in dem er die dazu erforderlichen Kommandos auf dem CAN Bus losschickt. Die Steuergeräte, die z.B. Hupe, Lenkung, Bremse steuern, führen diese Kommandos bedingungslos aus, weil sie keine Überprüfung der Authentizität und Berechtigung durchführen. Abbildung 4 zeigt diese Architektur, wobei das „Hacker Device“ im Falle des Prius ein Notebook war. Bei anderen Hacks wie z.B. beim Jeep war es ein gekapertes Mobilfunk-Steuergerät und bei der GM Corvette konnte man das Netzwerk über einen Dongle übernehmen, der von einer Versicherung bzw. einem Car Sharing Unternehmen ins Auto gebracht wurde.

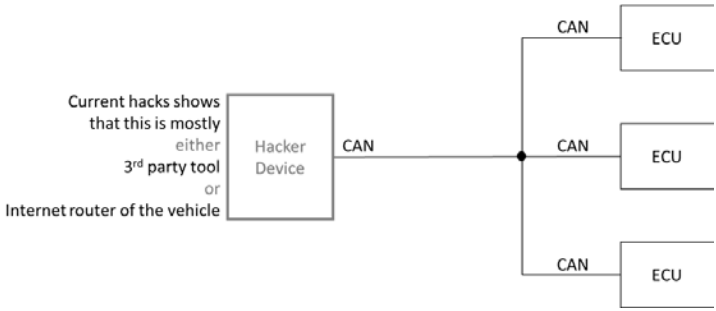

Bild 4: CAN Architektur eines Hacks

Die Teilnehmer des Einsteiger-Trainings sind innerhalb eines Tages in der Lage, mit dem Toyota Prius den gleichen Schabernack zu treiben. Dazu brauchen sie hardwareseitig einen geeigneten Linux Rechner, CAN-USB-Umsetzer und etwas handwerkliches Geschick, sich physikalisch mit den CAN-Netzen des Autos zu verbinden. Da ich intensiv Google-Hacking behandle, finden die Teilnehmer die CAN-Matrix des Prius im Internet. Das vorherrschende Format ist von Vector Informatik das CAN DB. Um dieses Format nutzbar zu machen, braucht es Konverter, die sich beispielsweise bei GitHub finden lassen. Nun lassen sich die üblichen Netzwerk-Analysatoren und auch Skriptsprachen verwenden, um die CAN DB zu überprüfen und zu nutzen. Alles andere hängt nur noch von der kriminellen Energie des Einzelnen und seinen Skriptfähigkeiten ab.

Das Prius-Netzwerk ist verhältnismäßig einfach. Andere Hersteller machen es komplexer. Sie glauben, dass Nachrichtenzähler oder gar die Verwendung von CAN FD eine höhere Sicherheit bringen. Wenn man sich mit Hackern unterhält, die auf Netzwerke spezialisiert sind, zeigt sich schnell, dass das Versprechen mit der höheren Sicherheit nur ein leeres Versprechen sein kann. Die Probleme mit der Authentizität und der Berechtigung sind prinzipbedingt und lassen sich nicht über CAN FD lösen.

Darüber hinaus bedarf wirksame Sicherheit Kryptografie und damit Rechenleistung sowie hinreichend große Datenmengen. Die CAN-Netze in modernen Fahrzeugen verfügen nur über kleine Datenmengen und sind heute üblicherweise nahezu voll ausgelastet. Die Datenrate eines klassischen CAN unterscheidet sich nicht von der einer CAN FD. Der einzige Vorteil von CAN FD ist, dass in der Theorie die Nutzdatenrate erhöht wird. In der Praxis macht das nicht für alle Nachrichten Sinn, denn sowohl Datenrate als auch die Nachrichten-Zyklen

wirken limitierend. Auch bieten die üblichen Automotive-Microcontroller nicht genügend Rechenleistung um kryptografisch überhaupt etwas Sinnvolles zu machen.

Vor dem Hintergrund der zu geringen Rechenleistung, zu niedrigen Bandbreite und vor allem vor dem Hintergrund der prinzipbedingt fehlenden Möglichkeit zur Authentifizierung muss jeder Versuch, Sicherheit über CAN FD zu bringen, ins Leere laufen. Die selbsternannten Experten, die hier Sicherheit durch CAN FD versprechen, dürften, und das meine ich wohlwollend, als Scharlatane bezeichnet werden.

Wahre Experten findet man spielerisch

Experten aller Fachrichtungen sind in diesen Tagen rar. Das gilt auch für Security-Experten. Man findet sie im wahrsten Sinne des Wortes spielerisch. Wir als Beratungshaus raten Unternehmen, den Mitarbeitern eine Reihe von Herausforderungen zu stellen. Beispielhaft zeigt Abbildung 5 einen Assembler-Code. Wer detailliert die Schwachstelle erklären kann, ist eine Runde weiter. Wer viele Runden meistert, eignet sich für eine Tätigkeit im Schwerpunkt Cyber Security. Der Bundesnachrichtendienst (BND) und andere Behörden machen es übrigens auch so. Wir bieten Unternehmen nicht nur Herausforderungen, sondern vor allem auf diesem Wege eine kostenoptimale Personalbeschaffung.

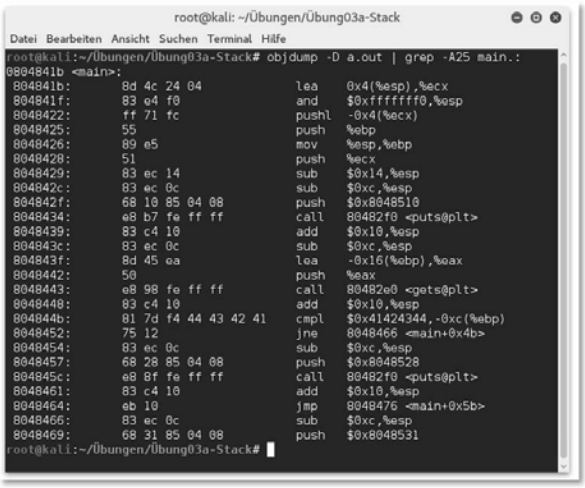

Bild 5: Herausforderung

Automotive Cyber Security ist ein Spiel

Security sollte weniger als Teil eines Entwicklungs-Prozesses, sondern vor allem als Wettkampf verstanden werden, den man gewinnt oder verliert. Bei den meisten Sportarten gilt: wenn man den Gegner kennt, kann man den Kampf gegen ihn gewinnen – das Überraschungsmoment muss ausgenutzt werden! Hackern sollte der Zugang zum Auto erschwert werden – dies geht nur, wenn Entwickler detailliert und stets aufs Neue lernen, wie Hacker vorgehen. Automotive Security ist kein Selbstzweck, sondern es braucht die kriminelle Energie der Hacker als Treibstoff. Wer im Wettkampf *Hacker gegen Automobilindustrie* bestehen will, muss vor allem ständig neue Angriffstechniken lernen und praktizieren.

Die Anpassung der Prozesse

Das aktuelle Wissen um Angriffstechniken leistet einen Beitrag zur Automotive-Security von rund 50%. Die anderen 50% sind hälftig dem Security-Entwicklungsprozess sowie den Verifikations- und Validierungsmaßnahmen inklusive Reviews zuzuordnen. Es schadet Automobilisten nicht, auch einen Blick in die IT zu riskieren. Besonders hervorgetan hat sich hier Microsoft, die mittlerweile über sehr gute und vor allem auch nach außen transparente Prozesse verfügen. Der grüne Anteil Top-Level-Prozess in Abbildung 6 zeigt den klassischen Verlauf einer Entwicklung. Allerdings hat man dort das normale Vorgehen um einige Security Aspekte ergänzt. Für Entwickler gilt, dass sie als Eingangsvoraussetzung ein Security-Training absolviert haben müssen, was blau hervorgehoben wurde.

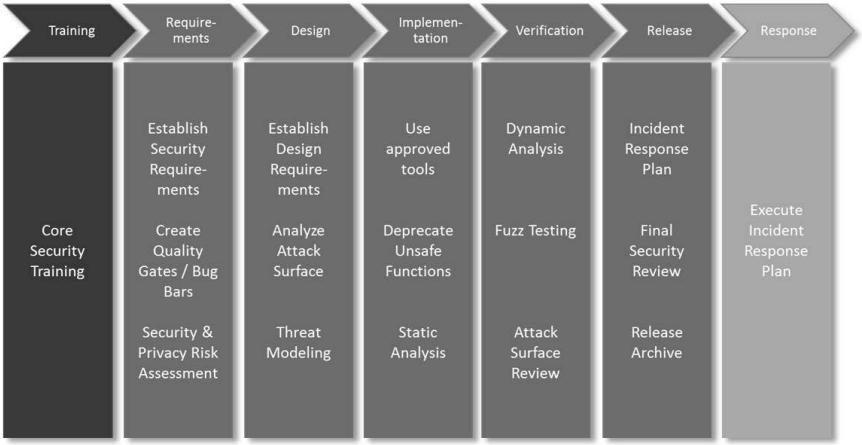

Bild 6: Microsofts Top Level Security Entwicklungsprozess

Eine wichtige Erkenntnis der Security lautet, dass es keine sicheren Systeme gibt. Diese Erkenntnis ist im Automotive-Bereich bereits durch Funktionale Sicherheit bekannt. Allerdings ist die Konsequenz, dass man einen Notfallplan braucht, den man im Fall der Fälle ausführt. Automobilisten werden vor allem die zeitlichen Aspekte eines Hacks zum Verhängnis. Anstelle der üblichen Reaktionszeit in Monaten muss ein Security-Problem innerhalb von Tagen gefixt werden. Das geht nur, wenn man sich im Vorfeld genügend Gedanken gemacht hat und diese Gedanken sich in einem Notfallplan wiederfinden. Abbildung 6 markiert den Notfall-Prozess gelblich als incident response plan. Seine Erstellung muss aber mit der Produktfreigabe abgeschlossen sein. Deshalb findet er sich auch im Release-Prozess wieder.

Zusammenfassung

Viele Hersteller vertrauen bei den eingeleiteten Security-Maßnahmen allein auf formale Security Prozesse. Sie sind weder effektiv noch effizient, um sich vor Hackern zu schützen. In der Konsequenz ist dieses Handeln bestenfalls naiv und fahrlässig. Warum das so ist, erschließt sich schnell, wenn man Automotive Cyber Security aus dem Blickwinkel eines Hackers betrachtet.

Die Entwicklungsprozesse der meisten Unternehmen verlangen den beteiligten Entwicklern das Anwenden eines standardisierten Vorgehens ab. Der starre und vor allem lineare Ansatz der Unternehmensprozesse ist exakt das Gegenteil von der besonderen Gabe der Hacker, alternative Anwendungsmöglichkeiten zu sehen. Dies wurde u.a. anhand des Beispiels Eisenbahnsteuerung mittels Telefonanlage erläutert. In der Folge entfalten die Security Engineering Prozesse in der Praxis eine deutlich geringere Wirkung als erhofft.

Die Tatsache, dass Security Engineering Prozesse keinen besonders signifikanten Beitrag bei der Entstehung sicherer Produkte leisten, bedeutet nicht zwangsweise, dass man auf diese Prozesse verzichten sollte, im Gegenteil. Allerdings sollte man sich der Grenzen der Prozesse klar bewusst sein. Im Kern sollte nie versucht werden, ein absolut sicheres Produkt zu entwickeln, sondern zur Produktfreigabe sollte ein Notfallplan existieren.

Ein Entwickler wird kein sicheres Produkt entwickeln können, solange er keine konkrete Vorstellung davon hat, was Sicherheit bedeutet. Vor diesem Hintergrund ist eine weitere Empfehlung, für alle Entwickler Security-Schulungen durchzuführen. Aus der IT wissen wir, dass Security nicht als Teil eines Entwicklungs-Prozesses, sondern vor allem als Wettkampf zwischen Herstellern und Hackern gesehen werden sollte, den man gewinnt oder verliert. Erst, wenn man den Gegner kennt, kann man gegen ihn den Kampf gewinnen!

Weder Können noch Ausrüstung der Hacker lassen sich von den Herstellern beeinflussen. Was bleibt, ist den Zugang zum Auto zu erschweren. Dies geht nur, wenn Entwickler detailliert und stets aufs Neue lernen, wie Hacker vorgehen. Automotive Security ist kein Selbstzweck, sondern es braucht die kriminelle Energie der Hacker als Treibstoff. Wer im Wettkampf *Hacker gegen Automobilindustrie* bestehen will, muss vor allem ständig neue Angriffstechniken lernen und praktizieren.

Neben dem Grundverständnis für Security bei allen Entwicklern werden eigene wenige Experten im Unternehmen gebraucht. Diese lassen sich dadurch finden, in dem man sie herausfordert. Auch die Personalentwicklung hat etwas Spielerisches. Allerdings gelingt das nur, wenn bei diesen Personen „kriminelle Energie“ vorhanden ist und diese sich auch frei entfalten kann.

How to Prepare Automotive for Future Challenges

ISO 21434 – A Standard for Cybersecurity Engineering

Dr. **Markus Tschersich**,
Continental Teves AG & Co. oHG, Frankfurt am Main

Die Automobilindustrie ist im digitalen Wandel und verschiedene Megatrends werden in der näheren Zukunft allgegenwertig sein. Elektromobilität, autonomes Fahren und Big Data bieten für Endkunden und die Wirtschaft viele neue Möglichkeiten. Neue und erweiterte technische Möglichkeiten führen heutzutage aber unweigerlich zu mehr Gefahren und Angriffspotenziale der Cybersecurity. Beim elektrischen Laden beispielsweise besteht eine Datenverbindung zu der kritischen Infrastruktur des Energiesektors, Autonome Fahrzeuge biete eine sehr große Bandbreite an Funktionen, die von einem Angreifer missbraucht werden können und die Sammlung und Verarbeitung von Daten in Fahrzeugen bedarf auch den sicheren Schutz dieser. Darüber hinaus erhöht sich aufgrund der immer weiter wachsenden Komplexität der Technologien die Integration der Wertschöpfungskette.

Infolgedessen ist eine verstärkte Harmonisierung der Prozesse in und zwischen beteiligten Unternehmen notwendig, um weiterhin die hohe Qualität der Produkte, insbesondere auch im Sinne der funktionalen Sicherheit, zu gewährleisten. Neue und bestehende Risiken müssen adäquat über die komplette Lebenszeit eines Fahrzeuges adressiert werden. Dies bedarf sowohl ein gemeinsames Verständnis von Cybersecurity-Risiken als auch Strategien, um solche Risiken gemeinsam und auf abgestimmte Art und Weise zu reduzieren oder zu lindern. Mit dem Industriestandard ISO 26262 für die funktionale Sicherheit hat die Automobilindustrie bereits ihre Kompetenzen im Integrieren einer wichtigen Disziplin in den Entwicklungsprozess über die komplette Wertschöpfungskette gezeigt. Eine gemeinsame Arbeitsgruppe von Experten der ISO und der SAE ist aktuell dabei das gleiche Ziel für Cybersecurity im Entwicklungsprozess zu erreichen. Die Arbeit an dem Standard „ISO-SAE AWI 21434 Road vehicles – Cybersecurity Engineering" wird sicherstellen, dass die Schutzziele der Cybersecurity ein integraler und elementarer Bestandteil des gesamten Entwicklungsprozesses der Automobilindustrie werden.

Im Zentrum dieses Ansatzes steht das Risikomanagement zur Identifizierung und Bewertung von Cybersecurity Risiken sowie das Ableiten entsprechender Gegenmaßnahmen. Der Standard wird ein Prozessrahmenwerk definieren, das beschreibt, wie diese Risiken über die Phasen des Entwicklungsprozesses und des Produktlebenszyklus und zwischen den Organisationen der Wertschöpfungskette anzuwenden sind.

Das Standardisierungsvorhaben ist im Oktober 2016 gestartet und auf 36 Monate geplant. Mit einer finalen Veröffentlichung des Standards ist daher Anfang 2020 zu rechnen.

Combining the Strengths of Elektrobit Secure Onboard Communication with Argus Intrusion Detection and Prevention System

Gilad Barzilay, Argus Cyber Security, Tel Aviv, Israel;
Martin Böhner, Elektrobit, Erlangen

1. Introduction

After conducting several discussions in AUTOSAR and with customers regarding aspects of the Secure Onboard Communication (SecOC)[1] specification, Argus and Elektrobit propose a joint deployment of Elektrobit SecOC and Argus Intrusion Detection and Prevention (IDPS) to overcome security challenges that were raised during the discussions.

While extensively detailing the techniques to secure a message, some security aspects remain out of the scope of the AUTOSAR specification, namely key management and freshness. In addition, the specification does not necessarily apply to all in-vehicle communications and when applied correctly, might be subject to the limitation of the CAN message size. Each of these aspects pose security challenges.

This paper outlines these challenges and proposes a joint approach of the Elektrobit SecOC mechanism and Argus IDPS to leverage the advantages of both technologies and greatly enhance the security of in-vehicle communications.

2. How does SecOC profit from additional security?

2.1 Specification Gaps in SecOC scope

While specifying many aspects of SecOC in detail, AUTOSAR intentionally excluded aspects like key management or freshness value management from standardization. These aspects must be defined by each OEM individually, which involves addressing all the unique security aspects of each individual solution. Elektrobit offers solutions in plain AUTOSAR and in OEM

[1] See AUTOSAR „Specification of Module Secure Onboard Communication“: https://www.autosar.org/fileadmin/files/standards/classic/4-2/software-architecture/safety-and-security/standard/AUTOSAR_SWS_SecureOnboardCommunication.pdf

variants. Nevertheless, for customers that are new to this subject, the challenges are quite demanding.
In a nutshell SecOC uses a symmetric, MAC based approach to make an explicit statement about the authenticity and integrity of transferred messages. Therefore, one or more corresponding symmetric keys are needed by all ECUs configured according to the SecOC specification.
To mitigate the risk of replay attacks, a freshness value is integrated in the scheme. The AUTOSAR standard refers to counter or time-based freshness values as typical options but, due to different OEM solutions, the topics are otherwise intentionally excluded from standardization.
This leaves two fundamental security-critical aspects to be defined by OEMs when implementing a SecOC scheme: Freshness value management and key management. Some typical challenges that need to be addressed are discussed below.

2.1.1 Freshness value management

Time management is a special form of counter management. To setup a SecOC scheme based on time values, you need a synchronized and secure time base in all ECUs participating in the SecOC group. The challenge is to ensure a secure time synchronization without relying on mechanisms of SecOC.
If an initial time value is transferred with a similar SecOC mechanism that is based on symmetric cryptography, all ECUs involved have access to the key used to authenticate timestamp messages. Any ECU could potentially impersonate the sender of a timestamp message and send spoofed timestamp messages to other ECUs. Therefore, a compromised ECU may attempt to do any of the following:

- Move time backwards (even slightly) to extend the message acceptance window.
- Alter time substantially in one network segment to take it out of sync with other network segments.
- Effectively disable the freshness mechanism by moving it to its maximum value. Typically, since a rollover of the value is not supported to prevent replay attacks, once the freshness reaches its maximum value, it will remain there indefinitely. Once the freshness value is fixed, replay attacks cannot be prevented.

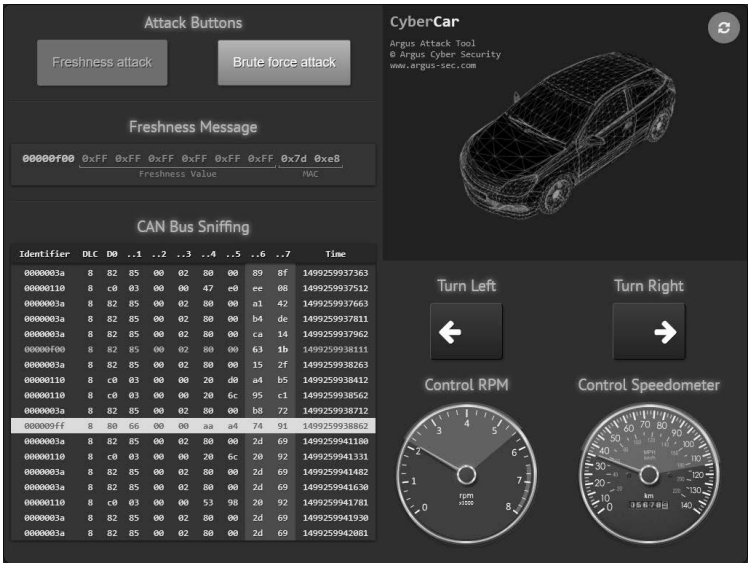

Fig 1: Argus and Elektrobit demonstration of the combined solution presented at Escar US 2017

2.1.2 Key Management

There are many strategies, to distribute and manage symmetric keys in relevant ECUs. Many less complicated strategies involve complicated logistics that can be difficult to maintain. Typically, in order to handle symmetric keys in different ECUs, key exchange mechanisms based on additional asymmetric algorithms are used. These mechanisms allow for key exchanges between ECUs in the vehicle and might be triggered once, frequently or upon request (e.g., when out of sync).

Implementations of such key exchange mechanisms are frequently expensive in terms of computational power which may lead to certain speed optimizations that jeopardize security goals:

- An attacker may be able to listen in on the process of ECU replacement and extract the new key from the in-vehicle traffic (i.e., even if it is encrypted by another “known” key).
- An attacker may force a key exchange to take place. This can cause denial of service or may allow the attacker to listen in on the key material.

2.2 Application of SecOC

The Elektrobit SecOC implementation is a mechanism that can give an explicit statement about integrity and authenticity of messages. However, each message that is to be protected by SecOC must be pre-selected and configured accordingly. For messages without a SecOC configuration, no statement regarding authenticity or integrity can be given.

2.3 MAC truncation

When implementing SecOC over CAN networks (as opposed to implementation over a CAN-FD network), there are only 8 bytes of data available per message (as opposed to 64 bytes in CAN-FD). To perform message authentication, those 8 bytes must include the MAC and the message data.
As a result of this stringent limitation, short MACs have to be used (e.g., a 2-4 byte MAC makes the scheme vulnerable to brute force attacks – for 2 byte MACs, an in-vehicle brute force attack would be successful in a matter of hours[2], for 4 byte MAC - a brute force attack would take about 10 days. In this context, brute force attacks could lead to the successful injection of one valid, but malicious, message on the bus if those messages are valid for such time slots.
Some examples of the potential effects of such a single valid, but malicious, message may be:

[2] See M. Dworkin: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, U.S. Department of Commerce, Information Technology Laboratory (ITL), National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, NIST Special Publication 800-38B, 2005, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-38B.pdf

- Malicious single messages may manipulate the freshness / timestamp value. This builds upon the analysis presented above but it does not require the attacker to know the SecOC key in order to pull off the attack.
- Malicious single messages may cause cyber physical damage to the vehicle. For example:
    - The message may simulate a pre-crash message which tightens the seat belts and cuts off the fuel pump.
    - The message may be able to unlock the doors or depressurize the anti-lock braking system (i.e., bleeding the brakes) in order to disable the brakes.

Attempting to execute this attack at scale on hundreds or thousands of vehicles across a large fleet can significantly reduce the time an attacker needs to successfully guess one valid message.

3. How SecOC may be enhanced by Argus IDPS

Compared to SecOC, Argus IDPS can provide a different kind of statement regarding the security of the system. Monitoring all messages on the bus, Argus IDPS does not require explicit configuration of single messages and can detect anomalies in the traffic based on comparing each message to a predefined behavioral model of expected in-vehicle traffic behavior.

Part of the behavioral model of an Argus IDPS may include, among other things, properties of protocols, the logic of message sequences and bus traffic in general. This may include timing and content properties of messages as well as expected consistency of messages and plausibility tests regarding different messages in the vehicle. For example, an Argus IDPS may analyze diagnostic traffic in order to validate it and make sure it is consistent with the current state of the vehicle or it may detect deviations from a predefined cycle time of a periodic message.

These qualities can be used to enhance the security of SecOC schemes and the adjacent protocols necessary.

3.1 Detection of Attacks

The example attacks on the SecOC scheme mentioned above typically require the attacker to inject messages onto the bus or modify the traffic in such a way which is not in accordance with how the in-vehicle traffic is expected to behave in normal circumstances.
Therefore, Argus IDPS, which is deployed at a central location in the vehicle (e.g., on the gateway, depending on the vehicle architecture), may detect such attempts to manipulate the SecOC based on different message timing and content models as well as consistency and other plausibility tests.
The behavioral model of Argus IDPS may include properties of the freshness value and key management protocols. Therefore, Argus IDPS may be configured to detect replay attacks, attempts to manipulate the freshness values, manipulations of the key management and brute force attacks. This enables the Argus IDPS to detect attacks on relevant protocols and messages necessary for the proper functioning of a SecOC scheme.
Since all messages transmitted over the in-vehicle network are monitored by Argus IDPS, it can detect attacks on messages that are not covered by the SecOC scheme as well as attacks aimed at the SecOC scheme itself.

4. Cyber Security across the fleet

Security logs from both Argus IDPS and the Elektrobit SecOC mechanism, as well as other security mechanisms in the vehicle, may be collected and sent to a central aggregation and analysis hub.
This information can then be used to better understand the cyber security health status of the fleet and assist in the cyber security incident management process.
The information is accessible to the customer (e.g., OEM and/or Tier 1s) through a web-based dashboard that enables the examination of different measurements and indicators regarding the cyber health of the vehicles in the fleet: rReal-time and historical information about cyber-attacks, updating the security policy of the fleet over the air (OTA), a heat map of attacks, and more.
Additional technical information such as forensic data can also be analyzed by cybersecurity experts and vehicle engineers. Another integral part of the dashboard is the ability to view the configuration of the security layers and modify/update it through OTA fleet updates.

Fig 2: The cyber health of the fleet is presented on a web-based dashboard

5. Proposal of a joint setup of Elektrobit SecOC and Argus IDPS

Working together, the Elektrobit SecOC implementation and Argus IDPS are able to provide comprehensive coverage of all network communications:

- Validate the authenticity and integrity of messages with Elektrobit SecOC. Secure relevant protocols and messages necessary for a proper SecOC scheme with Argus IDPS
- Enhance the security of all messages with Argus IDPS (regardless of whether they are configured in accordance with SecOC or not).
- Understand and respond to attacks in real time with over the air security updates through the monitoring and analysis of vehicle data generated by Argus IDPS and Elektrobit SecOC and other sources

6. Conclusions

In this paper, we discuss the advantages of both SecOC and Argus IDPS as well as highlight some of the risk-prone aspects, and the related security challenges, posed by the current state of standardization in AUTOSAR.
To overcome these challenges, Argus and Elektrobit propose a combined solution of the Elektrobit SecOC mechanism and Argus IDPS to leverage the security benefits of each technology. This combined solution can greatly enhance the overall level of security of in-vehicle communications.

Fleet SIEM als Bestandteil eines integrierten Automotive Cyber Security Management System

Fleet SIEM as a part of an integrated Automotive Cyber Security Management System

Dipl.-Kfm. **Ingo Dassow**, Dipl.-Inf. **Richard Bensch**,
Deloitte GmbH Wirtschaftsprüfungsgesellschaft, Berlin

Kurzfassung

Das moderne Fahrzeug – ein Rechenzentrum auf Rädern – besteht aus einer Vielzahl von hochvernetzten Komponenten und Funktionen. Die Vernetzung ist nicht auf die Kommunikation im Fahrzeug beschränkt, sondern schließt auch die Kommunikation mit dem OEM-Backend, anderen Fahrzeugen (Car2Car) und der Infrastruktur (Car2x) ein. Komponenten und Funktionen des vernetzten Fahrzeugs protokollieren kritische und sicherheitsrelevante Ereignisse und liefern dabei eine Vielzahl nützlicher Daten. Jedoch sind bestehende Sicherheitsüberwachungssysteme möglicherweise nicht ausreichend, um relevante Sicherheitsereignisse und/oder Vorfälle zu identifizieren. Für den Umgang mit der sich ständig weiterentwickelnden Bedrohungslandschaft ist eine moderne Automotive Cyber Security erforderlich. Um diese Herausforderung erfolgreich zu bewältigen, ist ein Schwerpunkt auf das Bedrohungsmanagement zu legen, das in den automobilen Entwicklungsprozess integriert werden muss. Security Information und Event Management (SIEM) ist ein bewährtes und häufig angewandtes Werkzeug in Unternehmen, um kritische Sicherheitsereignisse zu protokollieren. Es bietet ein Mittel, um das gesamte Bedrohungslebenszyklusmanagement zu implementieren. Daher könnte die Integration von SIEM in das Fahrzeug ein tragfähiger Ansatz sein, um Automotive Cyber Security-Bedrohungen zu bewältigen, die im Zusammenhang mit dem vernetzen Fahrzeug stehen. Diese Arbeit stellt ein mögliches Vorgehen für die Integration von SIEM im Kontext des vernetzten Fahrzeugs dar.

Abstract

The modern vehicle - a data center on wheels - is made up of a large number of highly connected ECUs and functions. Networking is not restricted to in vehicle communication but also includes communication with the OEM backend, other vehicles (Car2Car) and the infrastructure (Car2x). ECUs and functions of connected vehicle log critical and security-relevant events and thereby deliver a variety of useful data. However, existing monitoring systems may not be sufficient to identify relevant safety and/or security events. A modern automotive cyber security approach is required to deal with the ever-evolving threat landscape. In order to meet this challenge successfully, an emphasis must be placed on threat management, which has to be integrated into the automotive development process. Security information and event management (SIEM) is a proven and frequently used instrument in companies to log critical security events. It provides a means to implement the entire threat life cycle management. Therefore, the integration of SIEM into the vehicle could be a viable approach to tackle cyber security threats that could occur in the associated vehicle. This paper presents the procedure for the integration of SIEM in the context of the connected vehicle.

1. Motivation

Es gibt schnelle Fortschritte und die Industrie verändernde Entwicklungen im Kontext des vernetzen Fahrzeugs. Die Entwicklung der individuellen Mobilität wird u.a. durch das Internet der Dinge vorangetrieben und schließt die Kommunikation mit dem OEM-Backend, anderen Fahrzeugen (Car2Car) und der Infrastruktur (Car2x) ein. Mit der Steigerung der Funktionalität erhöht sich auch die Komplexität der E/E-Architektur, was auch im Hinblick auf die Automotive Cyber Security eine Herausforderung darstellt. Die (neuen) vernetzten Komponenten und Funktionen des Fahrzeugs protokollieren kritische und sicherheitsrelevante Ereignisse und liefern dabei eine Vielzahl nützlicher Logging-Daten. Die Verwendung der Logging-Daten in einem fahrzeugspezifischen Security Information und Event Management (Fleet SIEM) ermöglicht die Erkennung und Behandlung von Sicherheitsvorfällen im Kontext des vernetzten Fahrzeugs sowie wesentliche weiterführende Handlungsoptionen durch die Anreicherung mit Fahrzeugdaten aus Enterprise-Systemen. In der Enterprise IT ist SIEM ein bewährtes Werkzeug für den Umgang mit Sicherheitsinformationen und (kritischen) Sicherheitsereignissen. Mittels Fleet SIEM können anhand definierter Regeln (im Kontext von SIEM werden die Regeln „Use Case“ genannt) Angriffsvektoren identifiziert werden. Mittels der Erkennung von Angriffen durch ein Fleet SIEM können, abgeleitet aus einer Bedrohungsanalyse, risikobasierte Maßnahmen für den Umgang mit den Sicherheitsvorfällen (engl. Incidents) erarbeitet werden.

Im Folgenden wird zunächst der Aufbau und die Entwicklung eines klassischen SIEM in der Enterprise IT (siehe Kapitel 2) beschrieben, während Kapitel 3 einen generischen Prozess für die Entwicklung von Automotive Cyber Security beschreibt. Kapitel 4 stellt die Prozesse für die Automotive Cyber Security im Betrieb von Fahrzeugen dar. Während das Kapitel 5 eine konzeptionelle Beschreibung für die Entwicklung eines Fleet SIEM und dem Betrieb in Fahrzeugen liefert.

2. SIEM

SIEM ist als eine Komposition eines Security Information Management (SIM) und eines Security Event Management (SEM) definiert. SIM ist für alle logbezogenen Aufgaben wie die Protokollsammlung und -analyse verantwortlich. Auf der anderen Seite spezialisiert sich SEM auf die Echtzeit-Überwachung von Netzwerken und schafft damit die Möglichkeit, adäquat und schnell auf Sicherheitsvorfälle zu reagieren. Durch die Kombination von SIM und SEM entsteht ein SIEM-System, das eine effiziente Verwendung einer Vielzahl von Logging-Daten aus verschiedensten Quellen ermöglicht. Typischerweise besteht ein SIEM-System aus den folgenden Bestandteilen:

- Datenaggregation: Aggregiert Daten aus mehreren Quellen, bspw. Netzwerk, Servern, Datenbanken, Anwendungen etc. und bietet die Möglichkeit, überwachte Daten zu konsolidieren.
- Korrelation: Sucht auf Basis von Use Cases nach gemeinsamen Attributen und verknüpft Ereignisse miteinander zu sinnvollen Bündeln.
- Alarmierung: Die automatisierte Analyse der korrelierten Ereignisse und die Erstellung von Alarmen können zu einem Dashboard übermittelt werden.
- Dashboards: Werkzeug für die Darstellung von Ereignissen. Dient weiterhin als Hilfsmittel für die (manuelle) Erkennung von Vorfällen, die sich nicht durch eine automatische Analyse erkennen lassen.
- Erstellung von Compliance-Berichten: Automatische Erstellung von Compliance-Berichten auf Basis existierender Sicherheits-, Governance- und Audit-Prozesse.
- Retention: Die langfristige Speicherung von historischen Daten zur Erleichterung der Korrelation von Daten über einen definierten Zeitraum und zur Bereitstellung der für die Compliance-Anforderungen erforderlichen Retention. Langfristige Aufbewahrung ist bei forensischen Untersuchungen von entscheidender Bedeutung, da es nicht immer möglich ist, dass die Entdeckung in Echtzeit erfolgt.
- forensische Analyse: Stellt ein Werkzeug für die Analyse von Logs aus verschiedenen Quellen und Zeitperioden auf Basis bestimmter Kriterien zur Verfügung.

Traditionell wird SIEM zum Schutz von Institutionen gegen Cyber-Attacken eingesetzt. Unter Berücksichtigung des Bedrohungsmanagements verbessert SIEM die Erkennung von Sicherheitsvorfällen mittels der Korrelation von Ereignissen aus verschiedenen Quellen. Darüber hinaus können die gesammelten Ereignisprotokollinformationen im Rahmen der Implementierung von SIEM eine forensische Analyse signifikant unterstützen. Damit unterstützt SIEM die Beantwortung folgender Fragen: Woher wissen Sie, wann ein Angriff im Gange ist? Wie werden Sie den Angriff erkennen? Was muss getan werden, um auf den Sicherheitsvorfall zu reagieren?

3. Prozesse und Methoden für die Entwicklung von Automotive Cyber Security im Fahrzeugentwicklungsprozess

Für die kosteneffiziente Entwicklung und Implementierung von Automotive Cyber Security und damit der Schaffung der notwendigen Voraussetzung für ein funktionsfähiges SIEM ist es wichtig, dass der Prozess für die Erarbeitung von Automotive Cyber Security zeitnah nach

der Definition der fachlichen Anforderungen an den Entwicklungsgegenstand (E/E-Architektur, Komponenten und Funktionen) erfolgt. Im Rahmen der Entwicklung der Automotive Cyber Security werden unter Maßnahmen sowohl präventive Maßnahmen, detektive Maßnahmen als auch korrektive Maßnahmen verstanden. Mittels der frühzeitigen Integration der notwendigen Automotive Cyber Security-Anforderungen und -Maßnahmen in den weiteren Entwicklungsschritten wird sichergestellt, dass die Anforderungen und Maßnahmen nicht zu einem späteren Zeitpunkt der Entwicklung mit möglichen Auswirkungen auf Zeit, Qualität und Budget integriert werden müssen. In den folgenden Teilabschnitten dieses Kapitels sind die einzelnen Phasen der Automotive Cyber Security im Rahmen des Entwicklungsprozesses im V-Modell beschrieben.

3.1 Definition der Features

Auf Basis der Features der zu entwickelnden E/E-Architektur werden die fachlichen Anforderungen, Technologien, Schnittstellen etc. herausgearbeitet. Durch Anwendung einer aktuellen Bedrohungs- und Risikolage an die herausgearbeiteten Feature-Anforderungen werden grundlegende Anforderungen an die Automotive Cyber Security herausgearbeitet. Das daraus resultierende Anforderungsset ist Grundlage für die Erarbeitung von Automotive Cyber Security-Maßnahmen sowohl für die E/E-Architektur als auch für die gezielte Entwicklung von Funktionen und Komponenten.

3.2 Bedrohungs- und Risikoanalyse

Die Bedrohungs- und Risikoanalyse (engl. Threat and Risk-Assessments; TARA) ist eine Kerntätigkeit in der Konzeptphase der Automotive Cyber Security. In der Bedrohungsanalyse (engl. Threat Analysis) werden die für den Entwicklungsgegenstand einschlägigen Bedrohungen identifiziert und klassifiziert. Für die Risikobewertung ist es notwendig, dass die identifizierten Assets des Entwicklungsgegenstandes identifiziert werden. Für die jeweiligen Assets ist eine Bewertung der Auswirkungen im Schadensfalls vorzunehmen (bspw. in den folgenden Kategorien: Safety, finanzielle Auswirkungen, Reputationsschaden, Verletzung von Compliance, Datenschutzverletzung). Mittels der Kombination aus der Eintrittswahrscheinlichkeit der Ausnutzung einer Bedrohung auf Assets des Entwicklungsgegenstandes und der Auswirkung im Schadensfall für die einzelnen Assets ergeben sich Risiken.

3.3 Erarbeitung von Anforderungen und Maßnahmen

Für die in der Bedrohungs- und Risikoanalyse (vgl. 3.2) ermittelten Risiken gilt es, bei Bedarf (bspw. mittlere und hohe Risiken) weitere Security-Anforderungen und -Maßnahmen zu er-

arbeiten. Die Anforderungen und Maßnahmen ermöglichen einen Umgang mit den identifizierten Risiken. Für die Berücksichtigung der erarbeitenden Automotive Cyber Security sind diese in die weiteren Entwicklungsprozesse einzusteuern. Die Maßnahmen sind dann entsprechend in den Konzepten bzw. Lastenheften des Entwicklungsgegenstands zu integrieren, während die Anforderungen bei der Erarbeitung von Konzepten und Lastenheften entsprechend zu berücksichtigen sind.

3.4 Implementierung und Integration

Nach Abschluss der konzeptionellen Phase folgt der Umsetzungsprozess mit der Implementierung der Anforderungen aus der Konzeptphase, dazu zählen auch die Anforderungen an die Automotive Cyber Security sowie gesonderte Security-Funktionen.
Im Anschluss an die Implementierung erfolgt die Integration von verschiedenen Komponenten zu einem Teilsystem der E/E-Architektur, während die verschiedenen Teilsysteme zu einem Gesamtfahrzeugsystem integriert werden.

3.5 Validierung

Ein integraler Bestandteil des Entwicklungsprozesses ist die Prüfung, ob die erzeugten Funktionalitäten auf Basis der definierten Anforderungen erfüllt sind. In einem Bottom-up-Prozess (Funktion → Komponente → Teilsystem → E/E-Architektur/Gesamtfahrzeug) des V-Modells werden alle Entwicklungsgegenstände getestet, um sie im Anschluss zu integrieren. Im Kontext der Automotive Cyber Security werden typischerweise Testumfänge auf Basis des Ergebnisses der Bedrohungs- und Risikoanalyse durchgeführt.

3.6 Freigabe

Der letzte Schritt ist die Freigabe des Entwicklungsgegenstands aus Sicht der Automotive Cyber Security. Dazu erfolgt eine Bewertung der Ergebnisse aus der Validierung (Security-Tests) unter der Berücksichtigung der Ergebnisse aus der Konzeptphase (Bedrohungs- und Risikoanalyse sowie der erarbeiteten Maßnahmen). Bei der Erfüllung der Anforderungen aus der Konzeptphase, sowie dem Nachweis der Beseitigung aller Schwachstellen aus Security-Tests ist ein Entwicklungsgegenstand entsprechend aus Automotive Cyber Security-Perspektive freizugeben.

4. Automotive Cyber Security im Betrieb von Fahrzeugen

Um die Automotive Cyber Security während des Betriebs und der Instandhaltung des Fahrzeugs zu gewährleisten, sind geeignete Prozesse zu implementieren. Dazu ist eine kontinu-

ierliche Überwachung von potenziellen Schwachstellen, Bedrohungen und Risiken mit Auswirkung auf das Fahrzeug bzw. das angeschlossene Ökosystem notwendig. Durch die Extraktion der gewonnenen Informationen über Risikofaktoren aus der Risikobewertung der aktuellen Risiko- und Bedrohungslandschaft kann ein kontinuierlicher Sicherheitszustand des Fahrzeugs gewährleistet werden. Dementsprechend geben die nächsten Abschnitte einen Überblick darüber, welche Aspekte von Betriebs- und Wartungsprozessen mit vernetzen Fahrzeugen zu beachten sind.

4.1 Cyber Threat Intelligence

Unter der Cyber Threat Intelligence werden die aufbereiteten und in Kontext gesetzten Informationen über Bedrohungen für die (Automotive) Cyber Security verstanden. Sie ist für die Erkennung von Vorfällen und deren Bearbeitung hilfreich. Für die Durchführung von Cyber Threat Intelligence werden Daten über Bedrohungen und Schwachstellen aus einer Vielzahl von internen und externen Ressourcen verwendet. Durch das Importieren von Daten aus mehreren Quellen und Formaten, die Korrelation dieser Daten und das Exportieren in die vorhandenen Sicherheitssysteme oder Ticketing-Systeme eines OEM trägt Cyber Threat Intelligence zur Identifikation von neuen Risiken im Betrieb eines Fahrzeugs bei. Für eine effiziente Durchführung von Cyber Threat Intelligence ist der Betrieb durch ein Sicherheitsoperationszentrum (engl: Security Operations Center, SOC) zu leisten.

4.2 Incident Response-Prozess

Für die Lösung von Sicherheitsvorfällen (engl. Incidents) ist es notwendig, dass bereits vor dem eigentlichen Vorfall ein strukturierter Ansatz für die Lösung von Sicherheitsvorfällen existiert. Ein Incident Response-Prozess beinhaltet typischerweise die folgenden Aspekte für die Lösung eines Sicherheitsvorfalls:

- operative Arbeitsanweisungen für häufige Sicherheitsvorfälle (Incident Playbooks)
- Kommunikationsmatrizen zur Organisation der Kommunikation und Eskalation
- KPIs zur Effizienzmessung von Security Incident Management-Prozessen

Für die effiziente Erarbeitung einer Lösung von Sicherheitsvorfällen ist die Steuerung der Lösung eines Sicherheitsvorfalls in einem SOC anzusiedeln. Durch die Verwendung eines SOC besteht die Möglichkeit, dass ein zentraler Überblick über alle existierenden Sicherheitsvorfälle vorliegt und bei der Lösung von mehreren Sicherheitsvorfällen Synergien genutzt werden können.

4.3 Vulnerability (Schwachstellen) Management

Ein etabliertes und gelebtes Vulnerability Management erlaubt ein effizientes Verwalten der identifizierten Schwachstellen, dazu ist eine fortlaufende Pflege und Aufmerksamkeit erforderlich. Für eine optimales Vulnerability Management ist es essenziell, dass der Vulnerability Management-Prozess mit anderen Prozessen verknüpft ist. Mit entsprechenden Berichten liefert ein Vulnerability Management u.a. einen Überblick über die folgenden KPIs:

- Zahl der Schwachstellen pro System (Funktionen, Komponenten, Fahrzeuge, etc.)
- (Durchschnitts-)Alter der Schwachstellen
- Anteil der Systeme (Funktionen, Komponenten, Fahrzeuge, etc.)
- Zahl der Schwachstellen im Zeitraum x
- Wahrscheinlichkeit der Ausnutzung einer Schwachstelle
- Schwere einer möglichen Ausnutzung einer Schwachstelle
- ggf. weitere unternehmensspezifische KPIs

So ermöglichen diese und weitere KPIs unter der Berücksichtigung der aktuellen Bedrohungs- und Risikolandschaft eine Bewertung und Entscheidung für den Umgang mit den Schwachstellen.

4.4 Patch Management

Für die Beseitigung von Sicherheitsvorfällen ist es gemeinhin notwendig, dass dafür eine aktualisierte Version der zugrundeliegenden Software installiert wird. Auf Grund der Tatsache, dass sich ein Fahrzeug in öffentlichem Raum bewegt, sind im Rahmen des Patch Managements geeignete Strategien für das Ausrollen eines Patches zu entwickeln. Dazu werden auch Over the Air-Software-Updates eine immer wichtigere Rolle spielen. Jedoch ist bei der Durchführung des Patch Management zu beachten, dass die Fahrzeugkomponenten unterschiedliche Aktualisierungsrichtlinien haben und somit ECUs, Apps oder Funktionen nicht in der gleichen Weise aktualisiert werden können.

5. Fleet SIEM

Für die Erarbeitung und Implementierung eines SIEM im Kontext des vernetzten Fahrzeugs (Fleet SIEM) ist es notwendig, dass das Vorgehen für die Entwicklung eines SIEM in den Prozess der Fahrzeugentwicklung integriert wird (vgl. Kapitel 2 und 3). Dazu wird im Abschnitt 5.1 anhand eines auf den Kontext angepassten Vorgehens erläutert, wie ein SIEM im

Fahrzeug entwickelt werden kann. Der Abschnitt 5.2 beschreibt Strategien für die Umsetzung von Fleet SIEM, während der Abschnitt 5.3 den Betrieb eines Fleet SIEM näher erläutert und darstellt.

5.1 Vorgehen für die Entwicklung eines Fleet SIEM

Im Folgenden werden die einzelnen Schritte für die Entwicklung eines Fleet SIEM beschrieben, dazu werden die Beziehungen bzw. Synergien zu dem bestehenden Vorgehen im Entwicklungsprozess der Automotive Cyber Security dargestellt:

- Identifikation von Bedrohungen: Im initialen Schritt der Entwicklung eines Fleet SIEM müssen die Bedrohungen identifiziert und aufbereitet werden. Die Bedrohungen sind für die Erarbeitung der Use Cases im nächsten Schritt erforderlich. Im Vorgehen für die Entwicklung von Automotive Cyber Security (vgl. Abschnitt 3.2) werden die Bedrohungen bereits identifiziert, daher sind für die Identifikation von Bedrohungen keine Veränderungen am bisherigen Vorgehen notwendig.
- Identifikation von existierenden Datenquellen: Erfassung des Ist-Stands an existierenden Datenquellen im Fahrzeug. Die Erfassung dient im späteren Verlauf als Grundlage für einen Soll-Ist-Vergleich u.a. von vorhanden Datenquellen und weiteren Voraussetzungen für die Implementierung von Fleet SIEM.
- Definition von Use Cases: Für den Umgang mit den Bedrohungen mittels Fleet SIEM sind Use Cases zu erarbeiten. Die Use Cases definieren die Regeln für die Erkennung von potenziell eintretenden Bedrohungen, dazu sind u.a. die folgenden Aspekte zu erarbeiten:
    - notwendige Datenquellen für die Implementierung des Use Case
    - notwendige Datenformate
    - Korrelation zwischen den Datenquellen
    - Definition von operativen Vorgaben für den Umgang mit den Use Cases
    - Umfang der Maßnahmen, die bei der Erkennung auszulösen sind
    - KPI für die Auswertung bspw. für die Darstellung mittels Dashboards
- Identifikation von Entwicklungsbedarfen: Auf Basis der definierten Use Cases und der bereits existieren Datenquellen werden die Entwicklungsbedarfe abgeleitet, dazu sind bspw. die folgenden Fragen zu beantworten:

- Welche weiteren Funktionen sind für die Datenerhebung notwendig?
- Ist eine Übertragung von existierenden Daten im Fahrzeug notwendig?
- Wo findet die Auswertung der Daten statt?
- Ist eine weitere Komponente im Fahrzeug erforderlich?
  - Werden die Daten ausschließlich im Fahrzeug ausgewertet?
  - Werden Daten auch in das Backend des OEM transferiert und wenn ja, welche?

Die letzten drei genannten Schritte sind aktuell nicht Teil des Entwicklungsprozesses für Automotive Cyber Security, jedoch sind diese im weiteren Sinne als Erarbeitung von Anforderungen und Maßnahmen interpretiert worden (vgl. Abschnitt 3.3) und können daher ohne weiteren Aufwand in das Vorgehen aufgenommen werden.

- Entwicklung, Integration und Test: Analog zu Abschnitt 3.4 (Implementierung und Integration) werden die im vorherigen Punkt identifizieren Entwicklungsbedarfe umgesetzt und in die Bordnetz-Architektur integriert. Weiterhin wird die Implementierung des Fleet SIEM entsprechend getestet. Im Rahmen dieses Schrittes werden auch die ggf. notwendigen Funktionalitäten im Backend implementiert, integriert und getestet.
- Übergabe in den Betrieb: Das fertige und funktionsfähige Fleet SIEM geht in den Betriebsmodus über, dieser wird in Abschnitt 5.3 beschrieben.

5.2 Strategien für die Implementierung von Fleet SIEM

Für die Erarbeitung und Umsetzung von Fleet SIEM können verschiedene Strategien gewählt werden, so besteht ein Zusammenhang zwischen der Komplexität der Umsetzung und damit dem Ressourcenaufwand (Zeit, Material und Budget) und dem Abdeckungsgrad der Bedrohungen durch entsprechende Use Cases. Dieser Abschnitt liefert drei unterschiedliche Strategien für die Implementierung eines Fleet SIEM.

5.2.1 Implementierung auf Basis vorhandener Daten (im Backend)

Durch die verschiedenen Dienste im vernetzten Fahrzeug, bspw. Connected Lösungen inkl. OEM Backend, Entertainment, Navigation und Companion-Applikationen auf den jeweiligen Smartphones der Kunden, existieren bereits eine Vielzahl von Datenquellen, die für die Implementierung von Fleet SIEM Use Cases verwendet werden können. Die Herausforderung besteht darin, dass die verschiedenen existierenden Datenquellen an einem Punkt zentralisiert und ausgewertet werden müssen. Daher besteht die Möglichkeit, dass in einer initialen

Implementierung die Daten im Backend entsprechend für Use Cases verwendet und wo möglich durch weitere Daten im Fahrzeug angereichert werden können.

Dieses Vorgehen liefert eine erste Möglichkeit für die Implementierung von Fleet SIEM, jedoch stößt diese schnell an Grenzen, was die Umsetzbarkeit von potenziellen Use Cases angeht, da notwendige Datenquellen nicht ohne Weiteres (d.h. die Einbringung der Implementierung der Datenquellen in den Entwicklungsprozess) zur Verfügung stehen werden.

5.2.2 Implementierung einer Softwarelösung im Fahrzeug

Wie bereits in 5.2.1 aufgezeigt, hat die ausschließliche Strategie der Wiederverwendung von existierenden Datenquellen ihre Grenzen, was die Betrachtung des kompletten Spektrums an Bedrohungen angeht. Daher ist es notwendig, dass im Fahrzeug weitere Datenquellen für die Umsetzung von Use Cases implementiert werden. Im Sinne einer effizienten Nutzung von vorhandenen Ressourcen ist es notwendig, die Use Cases dahingehend zu betrachten, wo sie ausgewertet werden sollen. Dazu sind die folgenden Fragestellungen konkret im Zusammenhang mit der jeweiligen Fahrzeugarchitektur zu klären:

- Use Cases, die direkt im Fahrzeug ausgewertet werden und aus denen ggf. entsprechende Maßnahmen abgeleitet werden können. In diesem Fall erfolgt ausschließliche eine Meldung an das Backend für die Auswertung mittels Dashboards.
- Use Cases die nur partiell im Fahrzeug ausgewertet werden können und daher auch teilweise im Backend ausgewertet werden können. Für diese Use Cases gilt es, eine entsprechende Partitionierung zu finden, dass die richtigen, aber nicht zu viele Daten in das Backend übertragen werden. Die Meldung erfolgt weiterhin in Dashboards im Backend des OEM.
- Use Cases, die ausschließlich im Backend ausgewertet können. Trotz des Umstands, dass diese Art von Use Cases ausschließlich im Backend ausgewertet werden können, ist darauf zu achten, dass keine unnötigen Daten über die Luftschnittstelle transportiert werden.

Im Vergleich zur Strategie der Wiederverwendung von existierenden Daten (vgl. 5.2.1) bietet dieser Ansatz eine wesentlich höhere Abdeckung mit möglichen Use Cases für die existierenden Bedrohungen.

5.2.3 Implementierung einer hardwaregestützten Lösung im Fahrzeug

Die Implementierung und (partielle) Auswertung von Datenquellen für die Umsetzung von Use Cases kann durch die Implementierung der notwendigen Funktionen in Hardware unterstützt werden (Hardware/Software-Codesign). Hierbei gilt es, die punktuelle Unterstützung zu prüfen. Die Implementierung von Funktionen in Hardware hat den Vorteil, dass diese Funktionen eine bessere Performance und geringe Leistungsaufnahme im Vergleich zu einer reinen Software-Lösung haben. So hingegen liefert die Implementierung in Software die Vorteile, dass die Stückkosten geringer und die Wartung/Änderbarkeit durch Updates einfacher durchführbar sind.

5.3 Betrieb eines Fleet SIEM in einem SOC (Security Operation Center)

Wird durch die Analyse und Korrelation der Daten des Fahrzeugs ein Alarm ausgelöst, so wird ein Event erzeugt. Ein Event ist typischerweise in einem Dashboard sichtbar (vgl. Kapitel 2). Mittels eines Dashboards hat das Team des SOC einen Überblick über den aktuellen Stand der durch Fleet SIEM abgedeckten Bedrohungslage.

Die Darstellung der aktuellen Bedrohungslage mittels Fleet SIEM kann in den folgenden Teilprozessen der Entwicklung und des Betriebs der Automotive Cyber Security verwendet werden:

- Incident Response Management:
  - Alarmierung: Die automatisierte Analyse der korrelierten Ereignisse und die Erstellung von Alarmen.
  - Forensische Analyse: Stellt ein Werkzeug für die Analyse von Logs aus verschiedenen Quellen zur Verfügung und unterstützt damit bei der Analyse.
  - Erstellung von Compliance-Berichten: Automatische Erstellung von Compliance-Berichten bei meldepflichtigen Ereignissen.

- Vulnerability Management:
  - Identifikation von neuen Schwachstellen durch die Identifikation von neuen Angriffen mittels Meldung durch Fleet SIEM
  - Hilfsmittel für die (manuelle) Erkennung von neuen Schwachstellen, die sich nicht durch eine automatische Analyse erkennen lassen.

6. Auswertung

Mittels des dargestellten Vorgehens liefert der Einsatz eines Fleet SIEM einen wertvollen Beitrag zur Steigerung der Cyber Security im vernetzten Fahrzeug, sowohl in der Entwicklung als auch im Betrieb eines Fahrzeugs. Im konkreten Fall liefert die Verwendung von Fleet SIEM den folgenden Mehrwert für die Automotive Cyber Security:

- Steigerung der Messbarkeit von Automotive Cyber Security – bspw. mit KPIs und Metriken
    - Mittels Fleet SIEM wird die Datenlage der verschiedenen Datenquellen im Fahrzeug aufbereitet und damit ein Überblick über die Ausnutzung von Schwachstellen bzw. über die Durchführung von Angriffen gegeben. Die aufbereitete Datenlage kann sowohl im Incident Response-Prozess (vgl. Abschnitt 4.2) als auch im Vulnerability Management (vgl. Abschnitt 4.3) verwendet werden.
- Effektiveres Management (Reduzierung und Priorisierung) von Sicherheitsvorfällen im Incident Management bei gleichzeitiger Erstellung von detaillierten Berichten
    - Auf Basis der aufbereitenden Datenlage ist eine Entscheidung bzw. Priorisierung im Incident Response-Prozess (vgl. Abschnitt 4.2) und im Vulnerability Management (vgl. Abschnitt 4.3) möglich.
- Berücksichtigung der Besonderheiten des vernetzten Fahrzeugs für die Entwicklung und den Betrieb eines SIEM im vernetzten Fahrzeug
    - Die Entwicklung eines Fleet SIEM erfordert keine großen Anpassungen im Entwicklungsprozess, da die Entwicklung an bereits vorhandene Aktivitäten angedockt werden kann.

Referenzen

[1] The future of mobility - How transportation technology and social trends are creating a new business ecosystem, Deloitte University Press, 2017.

[2] Good Practices on the Security and Resilience of smart cars, European Union Agency for Network and Information Security (ENISA), 2016.

[3] Cybersecurity Guidebook for Cyber-Physical Vehicle Systems, Society of Automotive Engineers (SAE), 2016.

[4] Cybersecurity Best Practices for Modern Vehicles, National Highway Traffic Safety Administration (NHTSA), 2016.

[5] Risk Management Framework Applied to Modern Vehicles, C. McCarthy and K. Harnett; National Institute of Standards and Technology (NIST), 2014.

[6] Successful SIEM and Log Management Strategies for Audit and Compliance, SANS Institute, 2010.

[7] Magic Quadrant for Security Information and Event Management, Gartner, 2016.

[8] From Events to Incidents, SANS Institute, 2010.

CAN Transceivers with cyber security functions

NXP, the market-leading CAN transceiver manufacturer, has introduced ideas to secure the CAN lower layers that can be implemented in smart CAN transceivers

Dipl.-Ing. **Bernd Elend**, Dipl.-Ing. **Thierry Walrant**,
Dipl.-Ing. **Georg Olma**, NXP, Leuven, München, Hamburg

The modern connected car with various internal and external communication interfaces, up to 150 electronic control units (ECUs) and 100 million lines of code, is a cyber-physical system rather than a simple mechanical system. One challenge of seamless connectivity to the Internet and end-user devices is the exposure of the vehicle to malicious exploitation of vulnerabilities, such as buffer overflow exploits, malware and Trojans. The connected car's potential for attack (its "attack surface") is increasing as the amount of connectivity, electronics and software continues to increase.

A common method to mitigate these risks is Defense-in-Depth (DiD). DiD is a concept in which multiple layers of security countermeasures are placed through a system to provide redundancy in the event a single security countermeasure fails or a vulnerability is successfully exploited. This is important as the attacker will need to circumvent multiple countermeasures to launch a successful attack.

The responsibility to define what level of security is required lies with the vehicle manufacturer. Current state of the art solutions are cryptographic-based with secure key exchange, authentication and possibly encryption. Cryptographic checks of message authenticity are adding message latency and requiring considerable computing power. Thus, the disruption of applying these kinds of solutions can be prohibitive or lead to only partial implementation for protecting solely a low percentage of the CAN messages in a network. NXP therefore proposes an additional layer in the DiD concept, either complementing state of the art security solutions, or as a standalone solution for less critical, low cost ECUs, providing a basic-level of protection and hack containment.

Proposed is a distributed intrusion detection methodology, based on CAN network specific parameters, like identifiers of the CAN messages and the contribution to the overall network busload of an ECU. This method helps contain network attacks like spoofing, remote frame tampering and denial of service (flooding).

The method described is implemented solely in a smart CAN transceiver, operating fully independent and isolated from the microcontroller (MCU) – providing an inherent level of security, without neither impacting the message latency nor increasing the processor load. It can be introduced into a network in a stepwise approach, without impacting other ECUs. Such smart transceivers can be provided as drop-in replacements with today's standard CAN transceivers avoiding further hardware and software changes on the ECU and do not affect the operation of other ECUs. This makes the proposed approach a fast, low-effort and highly cost-effective way to introduce a basic level of security or fortify state of the art security solutions with a last layer of defense.

Spoofing, tampering, and flooding

Spoofing a CAN-ID means that a compromised ECU attempts to use an ID that it is not intended to be send by this ECU. This can be useful to pretend to be another ECU. This technique has been used in practical attacks on modern cars, see Figure 1. Spoofing in the body and comfort domain can be become safety relevant, as sudden unexpected actions can distract the driver tremendously, e.g. set radio volume to maximum, or turn lights on/off.

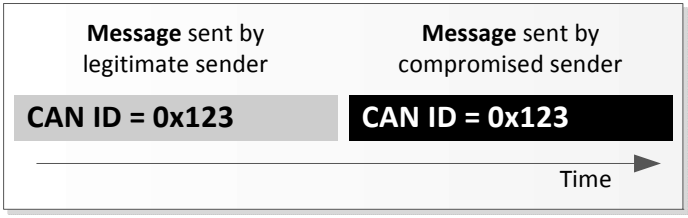

Fig. 1: Spoofing attack.

For the tampering attack, the attacker aims to adjust a message, which another ECU is currently sending on the bus. The attacker must also adjust the cyclic redundancy check (CRC) to match the tampered data. Before a successful tamper attack can be accomplished, the legitimate sender must be forced into the Error-passive state, or else it will publish an active error on the bus when the attacker causes a bit flip. The attacker can put the legitimate sender in Error-passive state by intentionally publishing errors on the bus for several times. The tampering attack is useful since it gives the attacker the power to tamper with the messages that are being sent on the bus, which may be of critical operation for the car. This kind of attack has been presented at several conferences, see Figure 2. The effects in the network are like caused by spoofing.

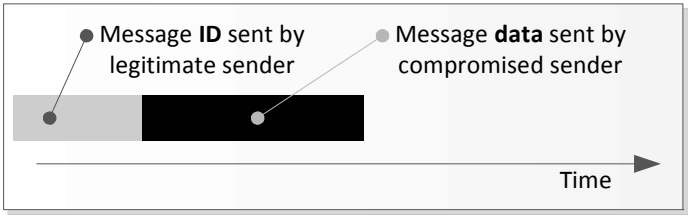

Fig. 2: Tamper attack.

Flooding the bus by continuously pumping the bus full of messages is a way to deny service, see Figure 3. This makes the bus unusable for all other ECU, which forces the entire vehicle into an emergency operating mode.

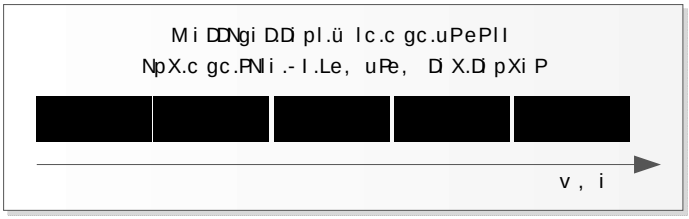

Fig. 3: Flooding attack.

Countermeasures

The methodology proposed by NXP can be implemented in smart CAN transceivers. All the countermeasures are based on parameters that the transceiver can perceive and are executed independently from the host, which might be compromised.

The first countermeasure, filtering messages based on CAN-IDs in the transmit path, is a way for the transceiver to protect the bus from a compromised ECU. If the ECU tries to send a message with an ID that is originally not assigned to it, the smart CAN transceiver can refuse to transmit this message on the bus by invalidating the message and deny subsequent transmissions. CAN ID-based filtering can be done using a white list of IDs that is user-configurable. For example, the IDs for Unified Diagnostic Services (UDS) as specified in ISO 14229 for off-board testers may be excluded from the whitelist. This would prevent a compromised ECU from starting a diagnostic session with another ECU in the vehicle to, for example, manipulate calibration values.

The second countermeasure against spoofing is the monitoring and invalidating messages on the bus based on the CAN-ID. This method enables every ECU to protect its own IDs in case a rouge ECU is not prevented from sending this ID; e.g. in case of an aftermarket device that is not under control of the car OEM and thus does not have a smart CAN transceiver with a configured transmission whitelist. When any ECU sends a message on the bus, the smart CAN transceiver of the legitimate ECU can actively invalidate that message by writing an active error frame to the bus. It can do this based of the same white list as the filtering in the transmit path. The compromised sender will repeat the spoofed message 16 times before Suspend-transmission behavior kicks in, limiting the bus load contribution, and finally another 16 repetitions will occur before the attacking ECU enters Bus-off state.

Preventing spoofing makes transferring a stolen cryptographic key to a rogue ECU useless, as the ECU cannot send the CAN IDs of the messages that it could authenticate with the stolen key!

Invalidating messages on the CAN network can also be used to prevent tampering. The smart CAN transceiver can check whether there was a valid message on the network, for which the local node has won arbitration, but stopped transmission (due to receiving a dominant bit while sending recessive). This is a clear sign that a compromised ECU has stepped into the transmission.

Limiting the number of transmitted messages per ECU of time can prevent flooding the network, when implemented at the sender side. In certain applications, a burst of messages on the CAN network is desirable, but this should only last for a certain amount of time. To prevent flooding, a leaky bucket mechanism can be used. In order, not to hamper diagnostic services, e.g. for uploading data, the contribution of messages with low priority IDs is neglected when filling the bucket. Flooding protection increases the availability of the network, also in case of babbling idiots.

Smart CAN transceivers with cyber security features are available

The proposed methodology is deployed on smart CAN transceivers, isolated from the host MCU and intended to be configured one-time at the Tier-1 production site and then locked, preventing future reconfiguration. An additional advantage of implementing in a CAN transceiver is it exploits the pervasiveness of the CAN transceivers in the in-vehicle network, ena-

bling a fast and cost-effective security upgrade of existing ECUs without touching the MCU and/or software.

NXP has developed a demonstrator to prove the concept. It is based on demo silicon in an SO8 package with standard transceiver pin-out.

Towards Cryptographic Agility in Automotive Systems

H. Gregor Molter, **Ahmad Sabouri**, **Marc Stöttinger**,
Continental Teves AG & Co. oHG, Frankfurt am Main

Abstract

The security of operations and information exchanges in modern vehicles mainly rely on the correctness and resilience of cryptographic algorithms and building blocks that are implemented as part of the platform. However, those cryptographic algorithms are prone to implementation bugs as well as loss of resilience due to aging. In this regard, the concept of crypto-agility as a mechanism to securely upgrade those building blocks while the vehicles are in the field, is an important topic to be considered before the vehicles leave the plants. In this paper, we investigate the possible failure cases that might occur to cryptographic algorithms and categorize them so that appropriate treatments can be designed. Moreover, we propose a procedure to handle each of the possible scenarios. Our work puts a step forward towards designing a vehicle-wide update mechanism that enables crypto-agility.

1 Introduction

In an interconnected system, such as modern vehicles, there are numerous possibilities for attackers to try out their chance for compromising the system. Security countermeasures employed to protect against various attacks are often built upon cryptographic solutions to ensure confidentiality, integrity and authenticity of applications and data. Besides attacks designed to bypass those countermeasures, it is not uncommon that the cryptographic building blocks of those countermeasures become the primary target of attackers. Efforts can be dedicated to find flaws in the design, implementation, and usage of cryptographic primitives. Moreover, those primitives are prone to aging as technological developments can weaken the computational infeasibility assumptions, such as for possibility of brute force attacks. A very recent example is the demonstration of ability to craft collisions for SHA-1 [9]. To cope with these kinds of downgrade in security level, cryptographic primitives may need to be enhanced, such as by increasing key-length, or replaced by an improved or a completely new algorithm. Therefore, the ability of a system to quickly adapt itself to these changing requirements, known as “crypto-agility”, seems to become an essential feature. Nevertheless, the concept of crypto-agility shall not be limited only to reconfiguration, but also should cover all

organizational chain of operations, e.g. generation and deployment of new keying materials, or interoperability with the legacy systems.

In this paper, we investigate the problem of securely updating cryptographic primitives. We take a detailed look into different scenarios of handling crypto-agility and propose a solution for them. The rest of this paper is organized as follows: Section 2 provides an overview of the related work for secure update of cryptographic algorithms. Afterwards, we introduce the preliminaries that are necessary to enable crypto-agility in Section 3. The core contribution of this paper is presented in Section 4, which delivers different update procedures and techniques for all the possible scenarios. In the end, we conclude the paper in Section 5.

2 Related Work

In the recent years, development of secure update solutions for the automotive domain has attracted a notable level of attention. For instance, the Uptane project[1] was announced to deliver a secure over-the-air update solution, which is resilient against several types of practical attacks [6]. The particular solution is a great step forward, however, it does not touch the distinctive aspects of updating cryptographic algorithms. Therefore, the concept of crypto-agility stays an important open question for such a system.

The concept of crypto-agility has been handled in the related works in two different ways. In the first approach, crypto-agility is introduced as a mechanism to achieve compatibility between different communicating parties when performing cryptographic operations - such as encryption / decryption, signature generation / verification, and integrity verification - by using different cryptographic algorithms. For instance, the secure communication protocols TLS [8,4] and IPSec [5] provide such ability by utilizing so-called cipher-suits. Other examples for enabling crypto-agility in terms of compatibility and legacy support by using cipher-suits would be smart meters [7] and the new specification of tachograph [10]. Both products support different cryptographic algorithms in their cipher-suites with different key sizes to ensure security in the future as well as compatibility.

The second approach is more close to the ability of securely updating or exchange cryptographic algorithms used in cryptographic building blocks when the device is in the field. To the best of our knowledge, for this case, no already mature or established examples are available in any civil industry branch. Directorate-General for Mobility and Transport (DG MOVE) of the European Commission specified requirements for updating cryptographic algorithms for the Cooperative Intelligent Transport Systems (C-ITS)[3]. In the proposed design

[1] https://uptane.github.io

the securing procedure is a chain of PKI certificates and a revocation strategy for invalid certificates [2].

Table *1*: Overview on related crypto-agility schemesTable 1 lists the different approaches on crypto-agility and indicates the maturity of the approaches as well as their intended usage.

Table 1: Overview on related crypto-agility schemes

Scheme	**Maturity**	**Intended Usage**	**Mechanism**
TLS	in-use for years	compatibility	cipher-suits
IPSec	in-use for years	compatibility	cipher-suits
Smart meters	in-use	future security	cipher-suits
Tachograph	concept	future security	cipher-suits
C-ITS scheme	concept	future security	chained certification scheme

3 Preliminaries to Enable Crypto-Agility

3.1 Agility-friendly Software Development

Achieving crypto-agility would be very difficult if the software using the crypto primitives does not enable this feature. This essentially means that the application must have been developed in a way that allows easy replacement of the underlying crypto algorithms without requiring major refactoring of the application code itself. Otherwise, updating a crypto algorithm may end up in a complete software update due to the change of all the references to the algorithm within the application code. For instance, high level programming languages such as Java and .NET take advantage of their object-oriented nature to enable cryptographic agility. Implementations of specific algorithms follow the definition of the abstract superset classes of the respective category of algorithms. More specifically, the polymorphism feature of object-oriented programming allows developing software that invoke different cryptographic algorithms of the same functionality without being concerned about the details underneath. One of the cornerstones of an effective update architecture of security primitives is the usage of an abstract crypto-API similar or equal to [1].

3.2 Protocol Versioning

Cryptographic primitives are not only used solely by one component but often employed as part of a protocol between some communicating parties within a vehicle or between some

components inside a vehicle and external systems. A crucial point about such a communication is that all parties must speak a common language. If there has been a change in the language of an involved party (e.g. updating one of the employed cryptographic primitives), there must be a mechanism in place to represent or determine such a change. More specifically, a versioning mechanism is unavoidable for the underlying cryptographic primitives. Such a versioning enables backward compatibility by allowing a component to recognize that the counter-part is still using an old or deprecated algorithms and thus adapt its language. Additionally, version numbers or algorithm identifiers facilitate offering secure alternative schemes for the same functionality that could be negotiated at the start phase of a protocol.

4 Secure Update Procedure

4.1 An Abstract View on the Update Process

A secure update procedure must be reliable as well, meaning that interrupts or unexpected conditions during the update process should not bring the vehicle in a broken state. In this section, we provide an abstract model of the procedure that must be followed to have a reliable and secure update. We distinguish between the case of an implementation bug and a deprecated algorithm. An implementation bug does not necessarily mean a flaw in the algorithm itself. Therefore, in most of the cases the problem can be addressed by patching the existing implementation and updating it. Typically, the fix does not require any further steps with regard to the key management, unless the key is compromised or leaked out due to the implementation bug. In this case, new keys must be set up between the communicating parties, after the bug is fixed.

The recommended steps for applying an update for a bug-fix are demonstrated in Fig. *1*. In the first step, the update package must be verified to ensure the integrity and authenticity. Moreover, a secure mechanism must be in place for version checking in order to prevent possible downgrade attacks. In case the verification fails, the process must be aborted and the driver shall be informed. In the next step, the existing algorithm must be backed up. The backup procedure must consider integrity, authenticity, freshness and sometimes confidentiality, especially when key related materials are included in the backup image. We recommend to perform a verification process to ensure that the backup is performed correctly before proceeding to the next step. When the backup is made, the update can be installed on the target algorithm. After the installation, it is necessary to verify if the process was performed correctly. This means to check if the installed algorithm is functioning correctly. Moreover, as we mentioned before, if the nature of the bug caused any leakage of the key, a key provisioning/agreement step may be considered to set up and verify a new key between

the communicating parties. In the last step, after ensuring that the update has been applied correctly, a cleanup process is necessary to remove the backup image and the confidential materials that might be left from the update process.

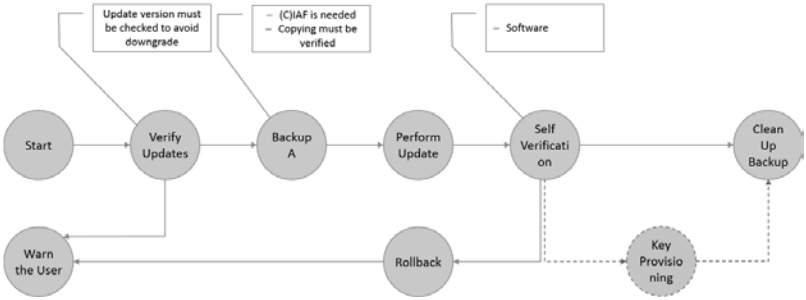

Fig. 1: Update Procedure for Fixing Implementation Bugs of an Existing Algorithm

The next type of update procedure concerns the case when an algorithm is found to suffer from some weaknesses in its design, which make it deprecated. Therefore the algorithm must be replaced by a new one. Such a procedure is demonstrated in Fig. *2*. Similar to the previous approach, the update process starts with verifying the update package to ensure its integrity and authenticity. Upon failure of this check, the process must be stopped and the driver shall be informed about it. If the verification of the update package was successful, the requirements for the installation of the new algorithm must be reviewed and checked on the device. This step is necessary because the new algorithm may need to co-exist with the deprecated one for a period of time to enable legacy support. Therefore, the device must be checked for the necessary additional resources that are needed by the new algorithm. Failing to meet the requirements causes the update process to stop. If the resources are available, the new algorithm can be installed in parallel to the old one. We do not consider a backup of the old one to be necessary here as we assume that the design of the system allows adding a new algorithm without interfering with the operation of the old ones. After installation of the new algorithm, it must be verified to ensure its correct operation. It is very likely that installation of a new scheme requires a key provisioning step due to the fact that it may require a different type of key than the existing one. The difference could be due to the length, cryptographic properties, or simply because the old key is compromised. The last step, which is the clean-up process, may not happen immediately but in the future. This is due to the previously

mentioned reason for keeping the deprecated algorithm for a period of time until it is phased out.

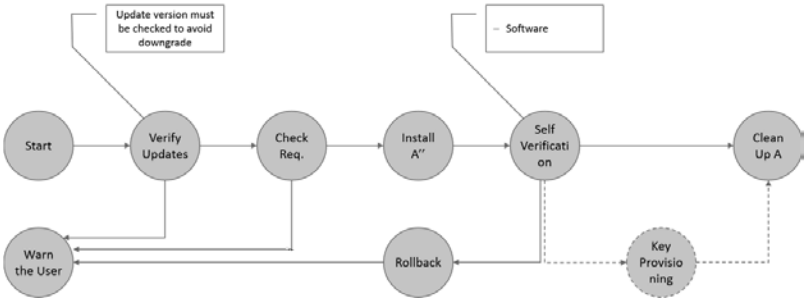

Fig. 2: Update Procedure for Replacing an Existing Algorithm with a New One

4.2 Foreseeable Scenarios

For the rest of our discussion, we envision four generic scenarios where an algorithm needs to be updated. These scenarios are introduced in this section. The urgency of an update is determined by the time distance to when the vulnerability can be exploited. Security issues could be categorized into *imminent* and *distant* issues. The former is often in form of implementation bugs that are discovered over night while the latter is mostly resulted from a series of analysis which show the underlying algorithms do not provide sufficient security in a foreseeable future. Another important aspect of the problem concerns the touch-points of the discovered vulnerabilities with the update process itself. Whether the security of the update process will be degraded due to discovered vulnerabilities or not, impacts the measures that must be taken for the imminent and distant issues. The combination of those two aspects will provide us with the following four scenarios:

- *Scenario I:* Algorithm *A* needs to be updated because it will be insecure at a *later* point of time, however, there exists another algorithm *B* which is secure and ensures the security of update process.
- *Scenario II:* Algorithm *A* needs to be updated because it is insecure at the point in time, however, there exists another algorithm *B* which is secure and ensures the security of update process.
- *Scenario III:* Algorithm *A* needs to be updated but it will be insecure at a *later* point in time. The algorithm itself is involved in ensuring the security of the update process.

- *Scenario IV:* Algorithm *A* needs to be updated but it will be insecure at the point in time. The algorithm itself is involved in ensuring the security of the update process.

Depending on the scenarios and the urgency for addressing the problem, different update strategies could be applicable. Over-The-Air (OTA) update is a cost-saving approach, which does not require the vehicle to visit a repair shop. Therefore, it is the preferred solution when it is possible. All the aforementioned scenarios can be handled in the secure environment of a workshop. However, in Scenario IV where the vulnerability impacts the update process itself, the changes cannot be securely applied remotely. Consequently, the only solution is to perform the updates in a trusted environment. Table *2* summarizes these information.

Table 2: Possible Update Strategies

	Distant security fix is needed	Imminent security fix is needed
Algorithm A is not involved in ensuring the security of the update process	OTA or Repair Shop	OTA or Repair Shop
Algorithm A is involved in ensuring the security of the update process	OTA or Repair Shop	Repair Shop

4.3 Implementation with Cipher-Suits

Having multiple algorithms providing the same functionality on a device fosters agility. When an issue is discovered in one of the algorithms, it might be possible to switch to another one until the problem is addressed adequately. Nevertheless, the details of the steps of the process can be different depending on the urgency of the required fix, and whether the functionality of the target algorithm is needed for the update process itself. Table *3* summarizes the different methods the must be taken for such a setting in various possible conditions.

Table 3: Update Steps Considering Using a Cipher-Suite

	Distant Security Fix		Imminent Security Fix	
	Bug in Algorithm	Imp. is Broken	Bug in Algorithm	Imp. is Broken
Algorithm A is not involved in ensuring the security of the update process	CS#1	CS#2	CS#3	CS#4
Algorithm A is involved in ensuring the security of the update process	CS#1	CS#2	CS#5	CS#5

Method CS#1: A distant security problem implies that the discovered issue will not be exploitable in the near future. The fact that whether *A* is needed for the update process or not, can influence the urgency and sensitivity of the fix but not the process itself. Fig. *3* demonstrates the schematic of the process. In the first step, a policy shall be enforced that the car uses the alternative algorithms *A'* whenever it is possible. If some communication partners do not support it, *A* can be still used without any concern for a while. Nevertheless, a deadline must be introduced by when the update must be applied. If all the communicating parties support *A'*, applying the updates on *A* may not require stopping any functionality. But if any of the functionalities has to be suspended while performing the update, it must be postponed to the time when the car is claimed by the driver to be in a safe state. It is important to mention that due to the distant nature of the problem, the updates on different communicating parties can be applied at a different point of time. After performing the update, the priority may be set back to use *A*. The update procedure shall follow the bug-fix approach, explained in Section 4.1.

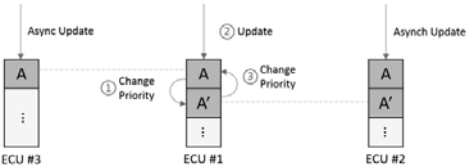

Fig. 3: Update Procedure for distant security problem due to an implementation bug - Cipher-Suite Approach

Method CS#2: Similar to the previous case, the identified problem does not require an immediate action and there is a reasonable window of time to react to the issue. Here also the fact that whether *A* is not needed for the update process or not, does not influence the process itself. The solution to this case should replace the deprecated algorithm *A* with the new one called *A''*. The update procedure for installation of a new algorithm was explained in Section 4.1. Enforcing the policy to employ the existing alternative algorithm, *A'*, on the device shall be considered in the first step. For those communicating parties that do not still support *A'*, the deprecated algorithm must be used. As depicted in Fig. *4*, the installation of the new algorithm *A''* must be performed in parallel to the deprecated one, until *A* is phased out. Therefore, it may not require stopping any functionality during the update process. In this case also, the updates can be applied at different points of time for the different parties.

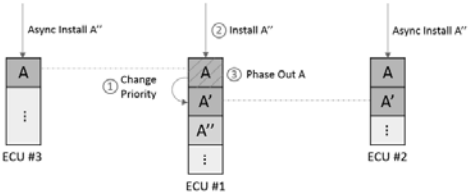

Fig. 4: Update Procedure for distant security problem due to algorithm deprecation - Cipher-Suite Approach

Method CS#3: When the discovered bug is considered to be an imminent security problem, immediate action needs to be taken to address the problem. In this case, we assume that the update package contains the bug fix for all the components using algorithm *A*. Therefore, it is possible to perform a synchronous update on all the components. The generic procedure explained in Section 4.1 for bug-fix is applicable in this scenario. The steps for this case are demonstrated in Fig. *5*. Enforcing the units to temporarily change to the alternative algorithm *A'* is an advantage. This can be performed without influencing the normal operation of the car. After the car is declared to be in a safe state, synchronous update must be performed on all the components using *A*. After a successful update, the priority shall be set back to use *A*.

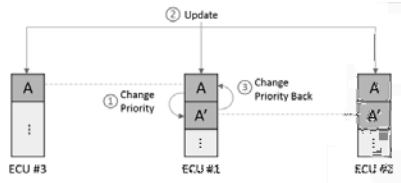

Fig. 5: Update procedure for imminent security problem due to an implementation bug when algorithm A is not needed for the update process - Cipher-Suite Approach

Method CS#4: In this case also, the identified weaknesses in the algorithm require immediate attention and must be addressed as soon as possible. Nevertheless, similar to the other cases, we assume that the update must not cause any interruption for the operation of the car, until it is declared to be in a safe sate. A deprecated algorithm must be replaced by a new one. Before getting the car to the safe state, those components that have an alternative algorithm in place, must switch to it when it is possible. Due to the imminent nature of the problem, we assume that the update package includes the fix for all the communicating parties and therefore allows a synchronous update (Fig. *6*). Similar to the other algorithm replacement procedure, the second method from the Section 4.1 must be employed. After installing the update on all the parties, it can be decided if *A'* or *A''* must be the primary algorithm to be used afterwards.

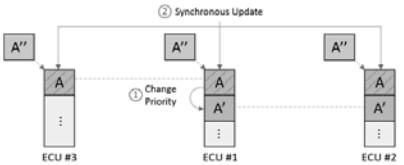

Fig. 6: Update procedure for imminent security problem due to algorithm deprecation when algorithm A is not needed for the update process - Cipher-Suite Approach

Method CS#5: This can be seen as one of the worst case scenarios. Any imminent problem that appears in an algorithm required during the update process, will not allow to perform a remote fix of the problem. This is because the update process will not be trustworthy anymore. In this case a trusted environment such as repair shop is needed to perform the necessary reconfiguration of the components.

4.4 Implementation without Cipher-Suits

The scenarios discussed in this section focus on the cases where algorithm *A* is the unique provider of a functionality and there is no alternative such as *A'* in the previous scenarios, which could be of help when a problem is identified. Therefore, the processes might be slightly different. Table 4 summarizes the process for all the possible cases. Below we review every case in details.

Table 4: Update Steps without Cipher-Suite

	Distant Security Fix		Imminent Security Fix	
	Bug in Algorithm	Imp. is Broken	Bug in Algorithm	Imp. is Broken
Algorithm A is not involved in ensuring the security of the update process	NCS#1	NCS#2	NCS#3	NCS#4
Algorithm A is involved in ensuring the security of the update process	NCS#1	NCS#2	NCS#5	NCS#5

Method NCS#1: The first update procedure described in Section 4.1 concerns the bug-fix scenario and can be used in this case. The distant nature of the security problem allows to have a relatively long time window for reaction. No matter whether the algorithm is needed for the update process or not, it can still be used for a while and fixes can be applied securely. In this scenario, nothing can be done before the car is brought to the safe sate. When the update is triggered by the user, then *A* can be patched and fixed. The update can be performed asynchronously on different components and typically does not require key renewal, as it is very unlikely that the key is affected due to a distant problem of an implementation bug. Figure 7 represents the update process for this scenario.

Fig. 7: Update procedure for distant security problem due an implementation bug - No Cipher-Suite Approach

Method NCS#2: Algorithm deprecation requires installation of a new algorithm. The second procedure of 4.1 is the recommended approach for performing updates in this case. Here also, the absence of an alternative algorithm providing the same functionality as *A* may prevent us from any corrective action before the car is brought to a safe state to apply the update, unless installation of the new algorithm can be performed without interrupting the normal functionality of the component. Nevertheless, the discovered problem is a distant one, and there must be reasonable window of time to react. Moreover, as the algorithm is still considered to be secure, it does not matter whether it is needed for the update process itself or not. As demonstrated in Fig. *8*, *A''* is installed on the component in parallel to *A*. After the installation, the priority shall be changed to use *A''*. This is only possible if the key is established between the component and its communication partners that also have installed *A''*. In this case a deadline must be set in order to phase out the deprecated algorithm.

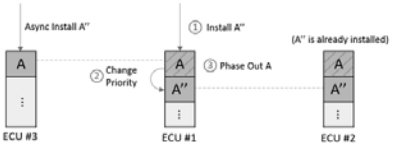

Fig. 8: Update procedure for distant security problem due an implementation bug - No Cipher-Suite

Method NCS#3: As previously mentioned, we assume the update package contains all the necessary fixes when an imminent security problem arises. The car must be brought to a safe state and the first procedure from Section 4.1 shall be followed to address the implementation bugs. Before that, the car must continue operation but the driver must be informed that urgent updates are pending to be applied. Since the algorithm *A* is not needed for the update process itself, a secure update can be envisioned. Fig. *9* depicts the update steps for this case. Upon the confirmation of the driver, a synchronous update is performed on all the communicating parties.

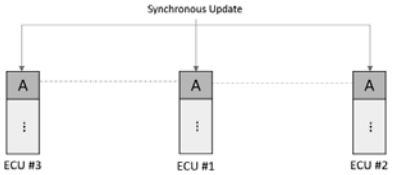

Fig. 9: Update procedure for distant security problem due an implementation bug - No Cipher-Suite Approach

Method NCS#4: Similar to the case of a bug fix, a synchronous update is necessary in order to replace the deprecated algorithm. Here also the security of the update process is not endangered because of the fact that the functionality of *A* is not needed for it. As depicted in Fig. *10* an atomic replacement of the deprecated algorithm is performed on all the components after the car is brought to a safe state. As always, installing a new algorithm may require a key agreement process between the communicating parties.

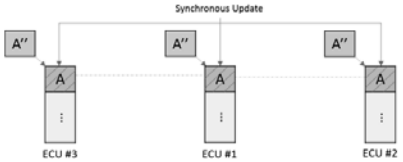

Fig. 10: Update Procedure for distant security problem due an implementation bug - No Cipher-Suite Approach

Method NCS#5: An imminent security problem in an algorithm which is needed to perform the secure update causes the update process to lose its trustworthiness. Therefore, the problem cannot be address without bringing the car to a trusted location such as a repair shop in order to perform manual reprogramming of the components. Similar to the case of having Cipher-Suites, a secure over-the-air update is not possible in this case.

5 Conclusion

We discussed the need for crypto-agility within a modern vehicle exploiting an increasing amount of cryptography in its ECUs and communication networks. If we take into account the past and recent advancements in the field of crypto-analysis, we can foresee that newly developed vehicle architectures will not be secure over their whole vehicle lifetime by statically using the same cipher-suits. Even more, cryptographic algorithms and schemes are prone to be implemented wrong. This can be observed in the long list of vulnerabilities and bug-fixes within open-source cryptographic libraries, e.g., OpenSSL. Thus, there is an additional need of being able to update the exploited cryptographic algorithms and schemes within a single ECU and within the whole vehicles including backend connectivity.

We addressed four different update scenarios in which most of the discussed procedures for updating cipher-suit and non-cipher-suit based cryptographic algorithms can be performed by using secure over-the-air (OTA) update functionality. In comparison to a manual update at a

garage, using OTA will save costs and ensures the possibility to roll-out patches in a timely manner, i.e. reducing the time of opportunity for an attacker.

The orchestration of the different update procedures for each electronic control unit used within the complex heterogeneous architecture of a modern vehicle is not an easy task. Especially, if one takes into account that different updates may be deployed from different stakeholders at different points in time. Future work will further address this complexity in terms of proposing an intermediate layer between the applications and the layer for cryptographic primitives to facilitate the complex orchestration for pending update procedures. In this regard, Autosar with its well established software architecture is an excellent candidate for integrating such an intermediate layer due to its modular design principles.

References

[1] AUTOSAR. Specification of Crypto Abstraction Library - 4.2.2. Technical report.

[2] WG5: Security & Certification C-ITS Platform. Final Report - ANNEX 1: Trust models for Cooperative - Intelligent Transport System (C-ITS) v1.1. Technical report, European Commission, 2015.

[3] WG5: Security & Certification C-ITS Platform. Final Report - ANNEX 3: Crypto Agility / Updateability in Cooperative - Intelligent Transport System (C-ITS) v1.1. Technical report, European Commission, 2015.

[4] Tim Dierks and Christopher Allen. RFC2246: The TLS Protocol. *IETF, January*, 1999.

[5] Paul Hoffman. RFC4308: Cryptographic Suites for IPsec. *IETF, January*, 2005.

[6] Trishank Karthik, Akan Brown, Sebastien Awwad, Damon McCoy, Russ Bielawski, Cameron Mott, Sam Lauzon, Andr´e Weimerskirch, and Justin Cappos. Uptane: Securing software updates for automobiles. In *14th Embedded Security in Cars Conference*, 2014.

[7] Helge Kreutzmann and Stefan Vollmer. Protection Profile for the Gateway of a Smart Metering System (Smart Meter Gateway PP). *German Federal Office for Information Security*, 2014.

[8] Eric Rescorla. *SSL and TLS: Designing and Building Secure Systems*, volume 1. Addison-Wesley Reading, 2001.

[9] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. The first collision for full SHA-1. Technical report, shattered.io, 2017.

[10] G07 Digital Citizen Security Unit. JRC Science and Policy Reports - Revision of the Digital Tachograph Security Framework. Technical report, Institute for the Protection and Security of the Citizen (Ispra), European Commission, 2015.

Physical and Mechatronic Security, Technologies and Future Trends for Vehicular Environment

Towards Counteracting Cloning in Automotive Industry

Prof. **Wael Adi**, Dipl.-Ing. **Ayoub Mars**,
IDA, Technische Universität Braunschweig

1. Abstract

Cloning spare parts and entities of mass products is an old and serious unsolved problem for automotive industry. The economic losses in addition to loss of know-how and IP theft as well as security and safety threats are huge in all dimensions. This presentation gives an overview on the traditional state of the art on producing clone-resistant electronic units in the last two decades. A survey is attempting to demonstrate the techniques so far known as Physically Unclonable Functions PUFs showing their advantages and drawbacks.

The necessity for fabricating “mechatronic-security” in vehicular environment is emerging to become a vital requirement for new automotive security regulations (legal regulations) in the near future. Automotive industry is facing a challenge to produce low-cost and highly safe and secure “networked” automotive units and systems. The emerging networked smart traffic management is offering new safety services and creating at the same time new needs and threats in a highly networked world. There is a crying need for automotive security for mass-products that approaches the level of the robust “biological security” for cars as dominating mobility actors in the modern smart life environment.

The use of possible emerging technologies allowing embedding low-cost practical mechatronic-security modules as digital alternative is proposed. Such digital clone-resistant mechatronic-units (as Electronic Control Units ECUs) may serve as smart security anchors for automotive environment in the near future. First promising initial results are presented.

2. Introduction

The Physically Unclonable Functions (PUFs) were introduced two decades ago for fabricating electronic unclonable units for secured identification or for intellectual property protection [1] to [31]. Due to the analog nature of all proposed PUF technologies, virtually all techniques proposed so far had limited use in real world applications due to economic cost factors and failing long term stability. The majority of such PUF techniques offer idealized DNA-like identity for a physical entity. However, all such technologies suffer from being highly inconsistent due to being sensitive to aging, temperature, power fluctuations and other operational conditions. The objectives of this presentation are to show the different known traditional unclonability technologies and their drawbacks for vehicular industry.

A secured “Mechatronic-Identity” is one of the most wanted requirements in automotive systems for producing unclonable or clone resistant units and spare parts. This technology seems to be in its childhood due to the expected high cost factors when embedded in vehicular units like Electronic Control Units (ECU) and seriously more problematic in mass-products as automotive wearing-parts. In the contemporary emerging world of Internet of Things (IoT), automotive units are becoming even a part of the worldwide communicating networked units. This requires however robust, consistent and low-cost technologies for safe and secure operation. This is admittedly a very challenging research and development area addressing the “Physical Security” issues which started recently to grow up exponentially in its importance in the research environment. Unclonable uniqueness is expected to play a major role as a security anchor in all our future applications; similarly as it is the case in our well established smart biological environment. Technology will possibly not attain the quality of biological DNA-identity. Modern technology can however, try to develop bio-inspired techniques.

The following sections include first an excursion in the world of PUF technologies followed by introduction to some basic concepts of mechatronic-security. Finally a summary of some research activities on *“Mechatronic-Security”* and “practical Digital-PUFs” at IDA, Institut für Datentechnik und Kommunikationsnetze of the Technical Universität Braunschweig are presented. This is an attempt to initiate hopefully fruitful discussions and mutual exchange of ideas and experiences between industrial and academic communities.

3. Basic Unclonability Concepts and Physically Unclonable Functions PUFs

Physical Unclonable Function (PUF) is a function embodied in a physical unique structure where its output looks like a random function, it is easy to evaluate but hard to predict even

for an attacker with physical access. PUFs are increasingly used in cryptographic systems especially for devices identification/authentication, secure memoryless key storage, IP protection and anti-counterfeiting. As unclonable entities, PUFs should fulfil the following security requirements:

Evaluable: For a given PUF, it should be easy to get a response R for any challenge C, R=PUF(C). The C-R pair is called a challenge-response pair CRP.

Uniqueness: PUFs should be unique, such that when challenging different PUFs with the same challenge C, each PUF generates a unique response which is also different from all other units. Fig. 1 illustrates this property.

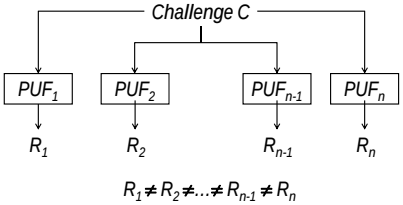

Fig. 1: PUFs uniqueness property

Robustness: Means that the PUF should generate consistent and repeatable responses within the whole system lifetime. Deviating responses should be impossible or tolerable by some adequate means.

Unclonability: It should be intractable for an adversary (including the manufacturer) even when physically attacked to create a physical or software clone of the PUF.

Unpredictability: The adversary should not efficiently be able to compute the response of a PUF to any challenge, even if he/she can adaptively obtain unlimited number of CRPs from the same PUF or other instances.

One-way: PUFs are difficult to invert such that given a PUF and one of its responses R, it is hardly infeasible to find the corresponding challenge C.

Tamper evident: The PUF should be tamper-evident. That is by any attempt to physically access the PUF, the PUF would changes its challenge/response behavior.

4. PUF instantiations

The idea of using the physical unique structure to identify units is similar to fingerprint identification of human beings. In the twentieth century, non-electronic PUFs have been

introduced; they were using random patterns on paper and optical entities to extract unique identities. Recently, many electronic technologies for PUFs instantiations have been proposed, Fig. 2 shows a classification table for the most known PUFs instantiations.

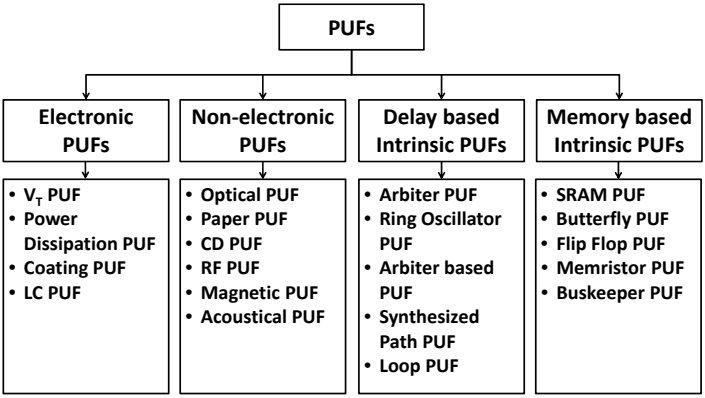

Fig. 2: Physical Unclonable Functions instantiations

3.1. Electronic PUFs

3.1.1. V_T PUF

V_T PUF was introduced in [1] , it consists of an addressable array of equally designed transistors driving resistive loads. The randomness is coming from the variations of the threshold voltage (V_T) of each transistor, the addressed transistor drives a resistive load, and the resulting random analog voltage sequence is converted to a binary identification sequence.

3.1.2. Power Dissipation PUF

Power Dissipation PUF was introduced in [2], it is based on the resistance variation of the power grid of chips which is connected to Power Ports (PP). The response consists of a set of voltage drops at a set of distinct locations. The measured electrical parameters are random and unique. Fig. 3 describes a sample instrumentation setup of Power Dissipation PUF for 6 PPs.

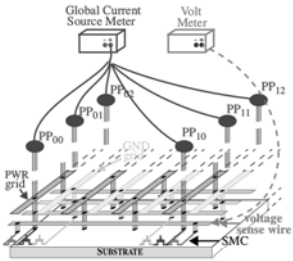

Fig. 3: Instrumentation setup of Power Dissipation PUF [2]

3.1.3. Coating PUF

Coating PUF was introduced in [3]. A network of metal wires (sensors) is laid out in a comb shape. The space between and above is filled with opaque material and randomly doped with dielectric particles. The unpredictable capacitance values are measured with comb-shaped sensors on the top metal layer of an integrated circuit.

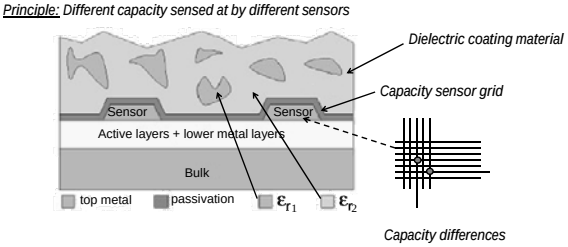

Fig. 4: Dielectric Coating PUF

Coating PUF represents possibly one candidate for fabricating low-cost clone resistant automotive mechatronic identities as would be shown later.

3.1.4. LC PUF

LC PUF was introduced in [4]. A passive resonator circuit (LC circuit) absorbs an amount of power when a RF electromagnetic field is generated. The power depends on the frequency and of the precise characteristics of the capacity and inductance of the LC circuit that uniquely identifies an LC circuit.

3.2. Non-Electronic PUFs

3.2.1. Optical PUF

Optical PUFs have been proposed first in [5], where reflective particle tags were developed for uniquely identifying strategic weapons. In [6], optical PUFs were proposed as Physical One-Way Functions (POWF). The concept of optical PUF is presented in Fig. 5. A Laser beam is directed to a light scattering material. Random and unique speckle pattern will arise; the pattern is captured by a camera for digital processing to generate a unique identity. To increase the number of CRPs, different laser orientations can be used.

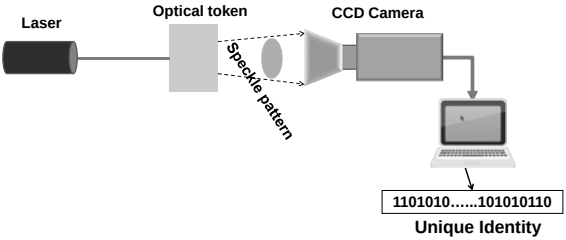

Fig. 5: Concept of Optical PUF

3.2.2. Paper PUF

Paper PUF was proposed in [7], It consists of scanning the unique and random fiber structure of regular or modified paper, the reflection of a laser beam on the random fiber structure is used as a fingerprint.

3.2.3. CD PUF

The data is stored as a series of lands and pits formed on the surface of the CD. It was observed that the measured lengths of lands and pits of a regular CD contain unpredictable random deviations which can be used as an identification profile [8].

3.2.4. RF DNA

In one implementation scenario, a physical token as a mixture of conducting and dielectric materials integrated with an RFID is produced. A high frequency wave (5-6 GHz) is propagated through the token. A low-cost antenna matrix receives a fingerprint of the token and use it as an identification profile [9].

3.2.5. Magnetic PUF

Magnetic PUF is based on determining the remanent noise in a magnetic medium by DC (Direct Current) saturation of a region thereof and measurement of the remaining DC magnetization. The remanent noise may then be digitized and recorded on the same magnetic medium to thereby "fingerprint" the magnetic medium [10].

3.2.6. Acoustical PUF

Acoustical PUF presented in [11], is based on the random delay in a fiber-glass delay line integrated in a device (DL701) used in televisions. When probing DL701 with one certain caustic wave, the internal medium of DL701 changes the wave response characteristics individually. In [11], it is shown that each individual DL701 produces a unique response. To create large number of CRPs, each unit is probed with different frequencies. Fig. 6 illustrates the principle of the acoustic delay line PUF's concept of DL701.

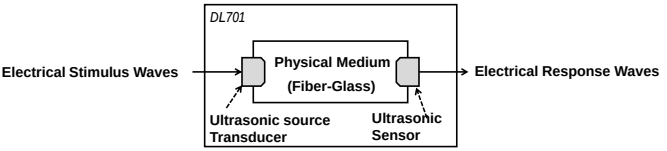

Fig. 6: Principle of DL701 Acoustic PUF

3.3. Delay-Based Intrinsic PUFs

3.3.1. Arbiter PUF

The initial proposal of Arbiter PUF was introduced in [12][13], it is based on the random delay or latency of each switch block, such that for two symmetrically designed units, and because of the manufacturing variations, there is a small random offset delay between the delays of the two units. For n switch blocks, there exist 2^n different delays selections as a stimulus. Fig. 7 illustrates the principle of an arbiter PUF.

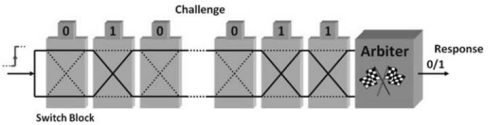

Fig. 7: Basic concept of arbiter PUF [14]

3.3.2. Ring Oscillator RO PUF

RO PUF is also based on random delays caused by manufacturing variations. It is based on a delay line where its output is inverted and fed backed to its input, resulting with an asynchronously oscillating loop. The PUF response is extracted by measuring the frequency of the RO, where a simple edge detector can detects the rising edges in the periodical oscillation and a counter counts the number of edges over a period of time [15]. Ring Oscillator structure is also practically deployed in Microsemi S devices as a true random entropy source for the Number Random Bit Generator (NRBG), it provides 384 full entropy conditioned bits to the Deterministic Random Bit Generator (DRBG) [16].

3.3.3. Arbiter-Based PUF

APUF was introduced in [17] to enhance the reliability of delay-based PUFs against temperature variations. APUF compensates the thermal-induced delay variation in the PUF circuits; a Voltage Controlled Current Starved (VCCS) inverter chain is regulated by adapting its control voltage by a complementary to absolute temperature (CTAT) reference generator.

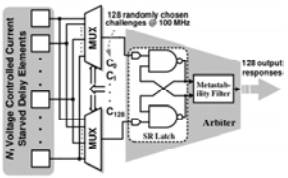

Fig. 8: Block diagram of APUF [17]

3.3.4. Loop PUF

Loop PUF was introduced in [18], it has nearly the same principle as the RO PUF. N identical and controllable delay chains are connected in a loop to create a ring oscillator, each delay chain contains M delay elements as shown in Fig. 9. The controller applies different combinations of the M control bits and measures the frequency or the delay of the oscillating loop.

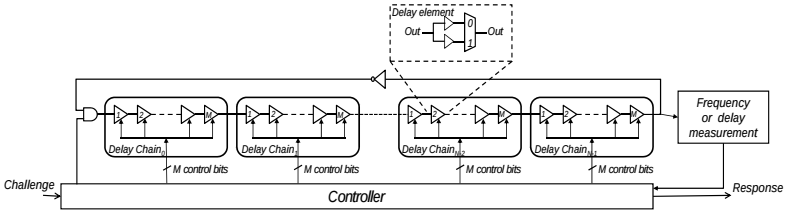

Fig. 9: Loop PUF structure

3.3.5. Sensitized Path PUF

SP-PUF was introduced recently in [19], an existing circuit is turned into an SP-PUF. It is measuring the delay differences in selected delay paths in existing designs by adding Race-Resolution Element (RRE) and multiplexers MuxA and MuxB as described in Fig. 10 [19].

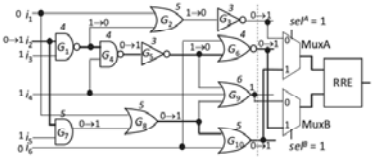

Fig. 10: Example of SP-PUF [19]

3.4. Memory based intrinsic PUFs

3.4.1. SRAM PUF

SRAM PUFs is one of the most practically used PUFs. was introduced in [20], this research was carried out at Phillips Research, before the company Intrinsic ID was founded. Intrinsic ID [21] created SRAM PUF, based on the behavior of standard SRAM memory in digital chips. It can differentiate devices such as microcontrollers from each other. Every SRAM cell has its own preferred state every time the SRAM is powered on, resulting from random differences in the threshold voltages. This randomness is expressed in the startup values of “uninitialized” SRAM memory. Hence an SRAM response yields a unique and random pattern of 0’s and 1’s. This pattern can represent a chip’s fingerprint, since it is unique for a particular SRAM and hence for a particular chip. In [14], security analysis of the

most popular intrinsic electronic PUFs have been done, resulting that SRAM PUFs have a Bit Error Rates (BER) of about 6% at +25 °C, 8% at -40°C and at +85°C. Also, the BER at 1.20V is about 6% also for 1.32V. SRAM PUFs are used in some Microsemi FPGAs such as SmartFusion®2 SoC FPGAs [16] and IGLOO2 [22] FPGAs that can be used for providing secure memoryless keys for cryptographic applications (PUF unit complexity is around 100 K-Gate Equivalent!). SRAM PUF needs its own protected design area/location and can't use the internal FPGA SRAM resources, as usable SRAMs are hard-resetted to all-zero after power up and hence its randomness is lost. Furthermore, SRAM PUF needs controllable power-up-event to enable the response generation which is not acceptable for most standard SRAM use applications.

3.4.2. Butterfly PUF

Since SRAM PUFs can't be used in FPGA environment without additional external resources, butterfly PUF was introduced in [23] to overcome the SRAM PUF drawbacks. It imitates the SRAM PUF in FPGA environment without the need for actual device power up, it consists of two cross-coupling latches. By using clear/preset functionality, a random state will be generated depending on the physical mismatch between the latches and the cross-coupling interconnect.

3.4.3. Latch PUF

Latch PUF [24] is very similar to Butterfly PUF, two NOR gates are cross-coupled to a simple NOR latch. Depending on the internal mismatch, it converges to a stable state.

3.4.4. Flip-Flop PUF

Flip-Flop PUF was proposed in [25], it is based on power up characteristics of uninitialized flip-flops. Flip-flops have the advantage of being able to easily spread over an IC which makes them difficult to locate by an attacker.

3.4.5. Memristor PUF

Memristor PUF was introduced in [26], memristors are emerging as next generation non-volatile memory technologies, a memristor is defined at logic 0 when $0 < w/D < O_L$ and for logic 1 when $O_H < w/D < 1$, the region $0 < w/D < O_L$ is undefined. Fig. 11 presents the memristor device model.

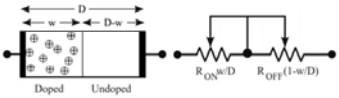

Fig. 11; Memristor device model [26]

The memristor PUF mechanism exploits the unpredictable state of the memristor within the undefined region, where the state depends on the duration of access time to the memristor and the value of supply voltage. Each physical memristor unit behaves individually differently to the same challenge that can be a Short Write Time (SWT) or Low Write Voltage (LWV).

3.4.6. Buskeeper PUF

Buskeeper PUF was proposed in [27], Buskeeper or Busholder is a week latch that usually has no control signals. It is intended to be used with on-chip buses that have multiple drivers, it is equivalent to a DFF with the enable signal connected to Vdd [27], also it has lower hardware complexity than Latches and DFFs. The principle of Buskeeper PUF is similar to all memory based intrinsic PUFs where the initial patterns are read at the memory start up.

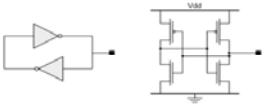

Fig. 12; Buskeeper cell structure [27]

5. Fuzzy extractors

PUFs are inconsistent in their behavior because of their sensibility to the environmental/operational conditions variations such as temperature, voltage, radiation and aging factors. To overcome the PUFs inconsistency, Fuzzy Extractors involving helper data algorithms are used to extract consistent responses from the PUFs. Fuzzy Extractors are designed to correct 25% of inconsistency errors or more. Fuzzy extractors induce however new security weakness that will be discussed in the following section. Fig. 13 describes the general constellation strategy of PUFs with fuzzy extractors.

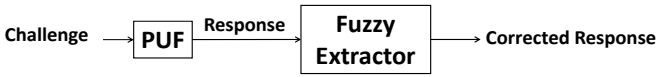

Fig. 13: Use of Fuzzy Extractor with PUFs

To get an impression about the complexity involved when using a practical PUF with fuzzy extractor, the hardware complexity of an intrinsic 16 K-bit SRAM PUF implemented in Micosemi FPGA should be around 100 K-Gates. This is still too high for automotive mass products.

6. Cloning Attacks on Physically Unclonable Functions

Two types of cloning are well known:

1. Mathematical/modelling cloning that aims to create an algorithm that behaves similarly as the targeted PUF.
2. Physical cloning characterizes the physical response of the targeted device and creates an identical physical response in a second instance of the same device type.

The process of cloning a PUF consists of two steps [28]:

- **Characterization:** a process in which the attacker gains knowledge of the challenge/response behavior of a PUF.
- **Emulation:** the process of recreating or modelling the unique response of a PUF, i.e. creating a PUF with identical challenge/response pairs.

In [29] many successful modelling attacks are presented on several known strong PUFs, such as Arbiter PUFs, XOR arbiter PUFs, Feed-Forward Arbiter PUFs, Lightweight Secure PUFs, and Ring Oscillator PUFs.

In [31], side-channel leakage attacks on PUFs and Fuzzy Extractors were shown. Attacks targeting weak PUFs and their fuzzy extractors were introduced. The presented analysis covers Arbiter PUFs and RO PUFs, by deploying side-channels mainly power consumption and electromagnetic emission. For fuzzy extractors, Simple Power Analysis (SPA) was used to attack both Code-Offset fuzzy extractor and Toeplitz hashing; hence extracting the cryptographic key derived from PUFs structure was possible.

In [28], cloning SRAM PUF by side channel analysis was treated. As SRAM PUFs use standard on-chip memory interfaces and buses, attacker can gain control of such interfaces, where he/she could read the memory content and hence clones the SRAM PUF. Invasive

de-capsulation with micro-probing can provide access to any memory content, especially if the memory IC is separated. For SRAM PUFs embedded in FPGA for example, this method is infeasible because of the huge number of interconnections. Several Side Channel Analysis (SCA) techniques can be used to extract the memory content or part of it, if the targeted device includes an inspection resistant memory. In [28], Photonic Emission Analysis (PEF) was used to extract the full content of an SRAM embedded in Atmel ATmega328P. PEF is passive, non-destructive and a semi-invasive SCA. The proposed attack uses a backside approach to clone the SRAM startup behavior exploiting the most known countermeasures seeking to detect malicious modifications on the front sides of the chips. The amount of lab time necessary to produce an initial clone was about twenty hours, and subsequent clones was produced more easily in less than three hours [28].

7. Mechatronic Security

7.1. Optimized Secured Mechatronic Identity Model

Physically linking an electronic unit to a mechanical unit such as in sensors or actuators, results with “mechatronic units”. The security of such entities starts by securely identifying such units as unique and unclonable entities. Therefore, a secured mechatronic identity is a first step towards protection against cloning in automotive environment. Fig. 14 shows a possible joint structural mechatronic identity. One ultimate implementation could be achieved by embedding an electronic unit such as some ECU in the physical structure when possible. The system includes now two identities which are joined together, namely the electronic one and the identity extracted from the body of the physical structure (as physical structure-DNA). In that case the resulting unit becomes unclonable as any invasive physical attack on the electronic unit would destruct the identity profile and hence destructs the secret to be cloned in the physical body.

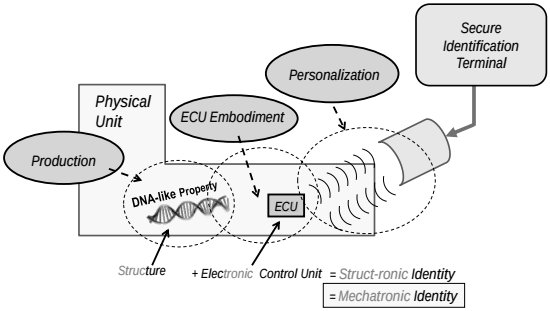

Fig. 14: Conceptual Mechatronic Security

7.2. Sample Implementation Scenario for a Mechatronic Identity

One possible DNA like identity profile seen as a structural PUF is the amorphous material structure of a physical body. Such structures exhibit properties which are impossible to predict, copy or to model. These can be adopted as adequate structural properties to be used as a part of a DNA-like identity chain.

To extract randomly a part of the internal structure, several randomly ECU-created stimulations can be sent through the structure's body. The responses to some randomly selected stimulus from a huge stimulus class may represent a DNA-like identity. If the number of stimulus possibilities exceeds say 2^{80}, then the trace of that structure is deemed as impossible to clone.

Possible stimulus technologies on physical structures can be (but not limited to): Optical, acoustic, electrical, chemical, etc. which may be seen jointly or individually as a Structural-DNA.

Fig. 15 shows a possible sample conceptual DNA-like profile-extraction-constellation by applying ultrasonic stimulation waves. The wireless link to the electronic control unit ECU is selected just to offer optimized security level against invasive attacks. The reason is that, any attempt to mechanically reach the ECU would destruct the secret structural response and hence destruct the DNA-like response and finally destruct the identity. That is, cloning the structure becomes impossible for any invasive attack!

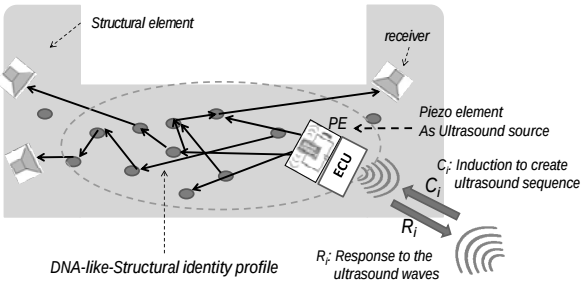

Fig. 15: Conceptual Identity Profile Extraction by Acoustic Waves

Fig. 16 shows a possible stimulation train scenario for the constellation of Fig. 15. The first attained experimental results of challenge–response pairs showed very promising results for creating DNA-like chain with very good cryptographic entropy exceeding 200 bits.

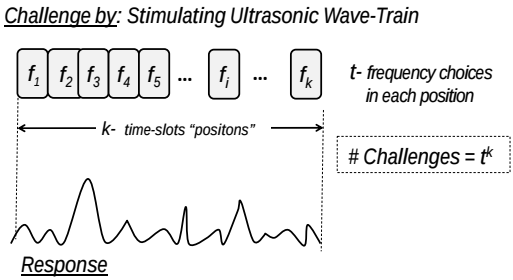

Fig. 16: Sample Acoustic Wave Stimulation and Response Sequences

When taking into consideration response post-processing for cancellation of aging, temperature and other operational condition factors, the entropy may get down remarkably. However, even entropy levels of say 40 bits would still offer acceptable mechatronic security. The reason is that the total identification entropy is the sum of both the structural and the ECU identification entropies. The digital ECU entropy is however highly scalable as would be seen in Section 6.

Fig. 17 shows a first experimental acoustic wave based identification response in two equally fabricated "Eyton" blocks stimulated at a speed of about 40 KHz ultrasonic stimulation

impulse. A challenging post-processing effort on the resulting patterns as those of Fig. 17 is required to attain aging-free (and consistent) identification profile for each structural type and form.

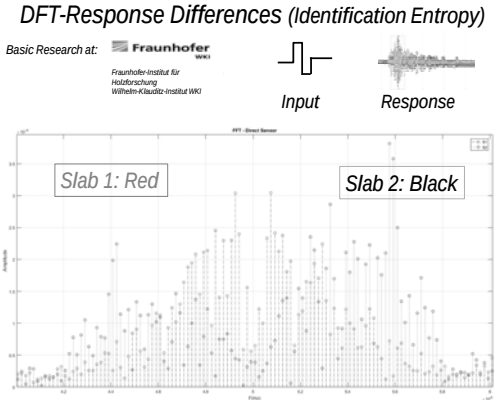

Fig. 17: Experimental Identification Response Differences after DFT Mapping

8. Proposed ECU Embedded Ultra-Low Cost Digital PUF Concept:

8.1. Secret Unknown Cipher SUC

A new digital PUF concept was developed at IDA, Technische Universität Braunschwig since 2008. The main idea behind the concept is to replace the traditional electronic analog PUF technologies mentioned in section 4 by some alternative digital PUF structures.

The key technique of the proposed digital-PUFs is by allowing the ECU to create own digital PUF structures internally. The result is a usable and practically implementable security however theoretically not perfect as the analog PUF.

The proposed digital PUF technique is only possible if a self-reconfiguring nonvolatile hard and software ECU technology is used. Such technologies are expected to be available in a form of System-on-Chip units SoCs in the near future.

Fig. 18 shows a proposed digital PUF creation/or ***"Mutation"*** concept as a Secret Unknown Cipher SUC. Notice that the security of the resulting system is not equivalent to "security by obscurity"!!. The reason is that the created/mutated cipher is a random-unknown-cipher and it is *not known to anybody*. The only secret which can be kept absolutely secure is the one

which nobody knows!!. The cipher is created by a single event trigger in a post-fabrication stage.

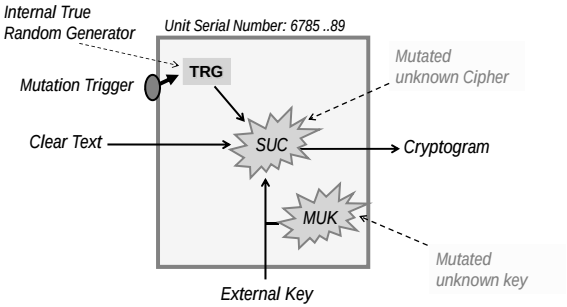

Fig. 18: The Secret Unknown Cipher (SUC) Principle

Fig. 19 illustrates a 3-way practical scenario for personalizing/creating a SUC by a non-predictable and non-repeatable single-event personalization process in a post-fabrication operation. The trusted authority loads a program called GENIE into each SoC unit and lets it create a cipher by the help of True Random Number Generator TRNG. The GENIE program is then deleted and a cipher which nobody knows is created as Secret Unknown Cipher SUC and the SoC becomes unique and unclonable. A record of say t-secret and random clear text and cipher text pairs X_i, Y_i are then set in a safe record by the trusted authority.

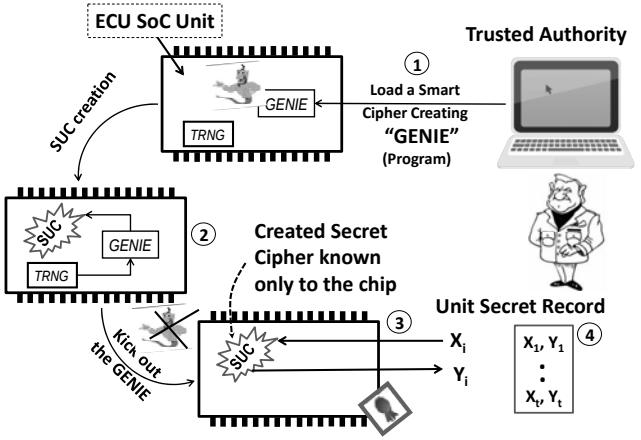

Fig. 19: Embedding Secret Unknown Cipher (SUC) Principle as a Digital PUF in ECU

Fig. 20 shows a possible generic two-way identification protocol of the created individual SUC in a certain ECU. It is simply by asking the unit to decrypt one randomly selected cryptogram from the secret record saved by the Trusted Authority TA. Notice that, the trusted authority cannot create two units with the same serial number!. Such unclonable checks may become the next regulatory-requirement to attain automated real-time secured car identification when necessary for future highest traffic safety and security.

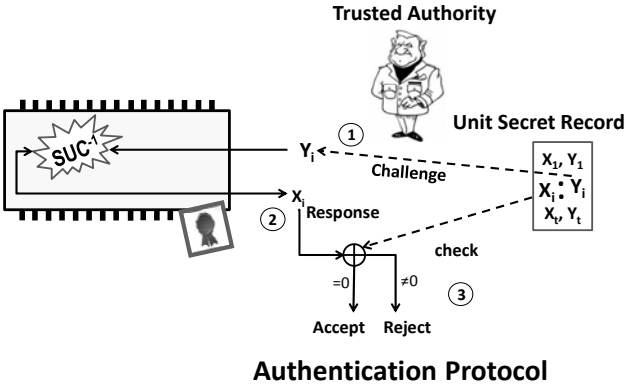

Fig. 20: Key Idea of SUCs in an ECU and a Primitive C-R identification Protocol

8.2. Sample Digital PUF Prototype as a SUC

Fig. 21 shows one sample layout implementation of a SUC having an entropy of 80 bits in a non-volatile Microsemi FPGA SoC technology.

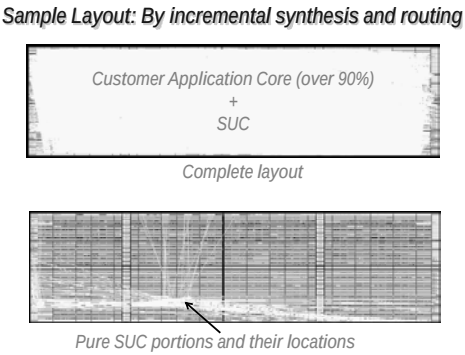

Fig. 21: Sample SUC implementation in a Microsemi SoC FPGA

The sample layout exhibits a very low complexity. In other words, if an ECU deploys such SoC FPGA units as microcontroller (Cost ca. 1 to 3 $), then embedding a secured digital

identity in the ECU unit may cost nothing other than a personalization process running for few milliseconds.

The sample accommodated SUC identity with 80-bits entropy has a hardware complexity of just 213 LUTs and 72 register bits. Fig. 22 shows different chip selection scenarios consuming at most 4.5% of the programmable hardware resources of the chip area.

Gate Complexity: 213 LUTs + 72 DFFs

SmartFusion2 SoC FPGA Families	*Resources usage*	
	LUTs % of usage	*DFFs % of usage*
M2S005	3,51	1,19
M2S010	1,76	0,6
M2S025	0,77	0,26
M2S050	0,37	0,12
M2S060	0,37	0,12
M2S090	0,24	0,08
M2S150	0,14	0,04

Cipher Cardinality:

- *The cardinality of this class of random cipher is 2^{274}*

Attack complexity:

- *Linear cryptanalysis: ≥ 2^{80}*
- *Differential cryptanalysis: ≥ 2^{80}*

Fig. 22: SUC implementation Complexity in a Microsemi SoC FPGA

Fig. 23 shows a possible practical realization scenario for a low-cost clone-resistant automotive mechatronic unit. The produced mechatronic units can only operate if the manufacturer approves them after fabrication!. Units fabricated by a third party are unclonable even if equally fabricated. Non agreed-on overproduction is then useless!

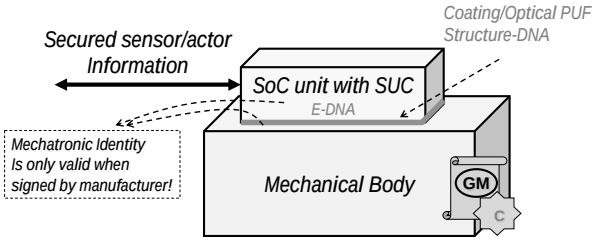

Fig. 23: Low-Cost Secured Automotive Mechatronic Unit

9. Conclusion

The emerging regulating vehicular security may need in the foreseen future that each vehicular unit should fulfil the following security requirements:

- Each car should accommodate unique, unclonable and remotely provable secured Identity. Even some vehicular ECUs may require having the same security level.
- The car manufacturer should not be able to create the same car identity (that is, car identity should be equivalent to the biological identity)

The necessary technology for such requirements is still costly and highly challenging in the contemporary technologies. Mechatronic security (Physical Security) is going to play a fundamental role as a security anchor in all future vehicular systems similarly as that of the biological-DNA as a robust, resilient and reliable identity.

References

[1] K. Lofstrom, W. R. Daasch, and D. Taylor, “IC identification circuit using device mismatch,” *Dig. Tech. Pap. - IEEE Int. Solid-State Circuits Conf.*, no. July, pp. 372–373, 2000.

[2] R. Helinski, D. Acharyya, and J. Plusquellic, “A physical unclonable function defined using power distribution system equivalent resistance variations,” *Proc. 46th Annu. Des. Autom. Conf. ZZZ - DAC ’09*, no. August 2009, p. 676, 2009.

[3] P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters, “Read-Proof Hardware from Protective Coatings,” pp. 369–383, 2006.

[4] J. Guajardo *et al.*, “Anti-counterfeiting, key distribution, and key storage in an ambient world via physical unclonable functions,” *Inf. Syst. Front.*, vol. 11, no. 1, pp. 19–41, 2009.

[5] Keith M. Tolk, “Reflective Particle Technology for Identification of Critical Components,” Sandia National Labs., Albuquerque, NM (United States). 1992.

[6] P. S. Ravikanth, “Physical One-Way Functions,” *Science (80-.).*, pp. 1–154, 2002.

[7] P. Bulens, “How to Strongly Link Data and its Medium : – the Paper Case –,” in *IET Information Security*.

[8] B. Sunar, “CDs Have Fingerprints Too,” in *CHES*, 2009, no. January 2014.

[9] G. Dejean and D. Kirovski, “RF-DNA : Radio-Frequency Certificates of Authenticity RF-DNA : Radio-Frequency Certificates of Authenticity,” no. January, 2014.

[10] R. S. Indeck and W. M. Marcel, “Method and apparatus for fingerprinting magnetic media,” US5365586, 1994.

[11] S. Vrijaldenhoven, “Acoustical physical uncloneable functions,” Netherlands, 2004.

[12] D. Lim, J. W. Lee, and B. Gassend, “Extracting Secret Keys from Integrated Circuits,” *IEEE Trans. Very Large Scale Integr. Syst.*

[13] J. W. Lee, Daihyun Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, “A technique to build a secret key in integrated circuits for identification and authentication applications,” *2004 Symp. VLSI Circuits. Dig. Tech. Pap. (IEEE Cat. No.04CH37525)*, pp. 176–179, 2004.

[14] S. Katzenbeisser, Ü. Koçabas, V. Rozic, A.-R. Sadeghi, I. Verbauwhede, and C. Wachsmann, “{PUFs}: Myth, Fact or Busted? {A} Security Evaluation of Physically Unclonable Functions ({PUFs}) Cast in Silicon,” pp. 283–301.

[15] R. Maes and I. Verbauwhede, “Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions,” no. 71369, pp. 3–37, 2010.

[16] Microsemi, “SmartFusion2 SoC FPGA Family.” [Online]. Available: https://www.microsemi.com/products/fpga-soc/soc-fpga/smartfusion2.

[17] S. Tao and E. Dubrova, “Reliable low-overhead arbiter-based physical unclonable functions for resource-constrained IoT devices.,” *CS2@HiPEAC*, pp. 1–6, 2017.

[18] Z. Cherif, J. Danger, S. Guilley, and L. Bossuet, “An Easy-to-Design PUF based on a Single Oscillator : the Loop PUF,” in *Digital System Design (DSD), 2012 15th Euromicro Conference*, 2012.

[19] M. Sauer and R. Ulrich, “Sensitized Path PUF : A Lightweight Embedded Physical Unclonable Function,” in *Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, pp. 680–685.

[20] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, “FPGA Intrinsic PUFs and Their Use for IP Protection,” *Cryptogr. Hardw. Embed. Syst. - CHES 2007*, pp. 63–80.

[21] Intrinsic ID, “Intrinsic ID.” [Online]. Available: https://www.intrinsic-id.com/.

[22] M. Corporation, “IGLOO2.” [Online]. Available: https://www.microsemi.com/products/fpga-soc/fpga/igloo2-fpga.

[23] S. S. Kumar, J. Guajardo, R. Maes, G. J. Schrijen, and P. Tuyls, “The Butterfly PUF protecting IP on every FPGA,” *2008 IEEE Int. Work. Hardware-Oriented Secur. Trust. HOST*, no. 71369, pp. 67–70, 2008.

[24] C. Tokunaga, D. Blaauw, and T. Mudge, “A 1.6pJ/bit 96% Stable Chip-ID Generating Circuit using Process Variations,” in *Solid-State Circuits Conference, 2007. ISSCC 2007*, 2007, pp. 404–406.

[25] R. Maes, P. Tuyls, and I. Verbauwhede, “Intrinsic PUFs From Flip-Flops on Reconfigurable Devices,” *Work. Inf. Syst. Secur.*, no. 71369, pp. 1–17, 2008.

[26] P. Koeberl, “Memristor PUFs : A New Generation of Memory-based Physically Unclonable Functions,” in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2013, pp. 1–4.

[27] P. Simons, “Buskeeper PUFs , a Promising Alternative to D Flip-Flop PUFs,” in *Hardware-Oriented Security and Trust (HOST), 2012 IEEE International Symposium*, 2002.

[28] C. Helfmeier and C. Boit, “Cloning Physically Unclonable Functions,” in *Hardware-Oriented Security and Trust (HOST), 2013 IEEE International Symposium*.

[29] U. Rührmair, F. Sehnke, J. Sölter, and G. Dror, “Modeling attacks on physical unclonable functions,” in *Proceedings of the 17th ACM conference on Computer and communications security*, 2010.

[30] U. Rührmair *et al.*, “PUF Modeling Attacks on Simulated and Silicon Data,” in *IEEE Transactions on Information Forensics and Security*, 2013, vol. 8, no. 11, pp. 1876–1891.

[31] D. Merli, D. Schuster, F. Stumpf, and G. Sigl, “Side-Channel Analysis of PUFs and Fuzzy Extractors.,” in *Trust*, 2011.

Methoden und Nachweise der Angriffssicherheit zur Integration offener Netzwerkverbindungen in Fahrzeugsystemen

Dipl.-Ing. **Heiko Fimpel**, Dr. **Falk Lindner**, M.Sc. **Alexander Winnicki**,
SILVER ATENA Electronic Systems Engineering GmbH, Hamburg

Kurzfassung

Assistenzfunktionen im Fahrzeug dienen der Entlastung des Fahrers von Überwachungs- und Kontrollaufgaben. Sie stellen ihm darüber hinaus Wetter-, Verkehrs- oder Navigations-Informationen zur Verfügung und erhöhen die Sicherheitsreserve in kritischen Fahrsituationen. Auch zukünftig werden sich die Assistenzfunktionen weiter entwickeln bis hin zum (teil)-automatisierten Fahren. Das umfangreiche Aufgabenspektrum solcher Funktionen kann nur durch softwarebasierte Systeme realisiert werden. Steigende Anforderungen an die Funktionalität, Verfügbarkeit und Autonomie der Systeme bedingen eine stetig zunehmende Interoperabilität der Funktionen untereinander sowie eine Ausweitung ihrer Konnektivität ins operative Umfeld des Transportmittels. Auf der Systemebene bedeutet dies eine intensive Datenkommunikation innerhalb des Fahrzeugs, von Fahrzeug zu Fahrzeug sowie zu externen (TCP/IP, Bluetooth, WiFi, LTE, …) Datennetzen. Neben den zahlreichen Vorteilen für die Insassen hinsichtlich erhöhter Funktionalität, mehr Sicherheit und Komfort steigt andererseits auch sein Gefährdungspotential aufgrund gezielter, vorsätzlicher Manipulationsangriffe auf die Systemintegrität von außen über die zahlreichen Kommunikationsschnittstellen.

Sowohl für Automobilhersteller als auch für Zulieferer wird es immer entscheidender, frühzeitig Security-Analysen in den Entwicklungsprozess zu integrieren. Die klassischen Methoden der Sicherheitsanalysen zur Bewertung der Funktionsintegrität greifen bei Systemen, die extern manipuliert werden können, zu kurz. Die bestehenden Methoden aus der funktionalen Sicherheit gehen von zufällig auftretenden Fehlern aus, die zum Versagen einer kritischen Funktion führen. Demgegenüber betrachten Security-Analysen die Systemintegrität unter dem Aspekt vorsätzlicher Angriffe auf das System, also die Option von Fehlfunktionen, die nicht zufällig, sondern gezielt die Schwächung der Funktionsintegrität hervorrufen. Solche Manipulationen haben keineswegs immer sicherheitskritische Relevanz. Oft beschädigen sie eher die Eigentumsrechte (Patente, Daten etc.) oder das Markenimage eines Herstellers. Solche Risiken lassen sich meist eindeutig identifizieren und getrennt von Sicherheitsanalysen behandeln. In Fällen in denen die Sicherheit des Produktes betroffen ist, muss die

Security-Analyse sehr früh und sehr eng mit den Methoden der funktionalen Sicherheit korreliert werden. Die Korrelation der beiden Analysemethoden sollte über den gesamten Entwicklungszyklus bestehen bleiben. Das strukturierte Vorgehen gewährleistet einerseits eine transparente (ggf. auch quantifizierbare) Bewertung der Angriffsrisiken und ihrer Auswirkungen sowie andererseits die systematische Erfassung möglicher Sicherheitslücken, die so im Systementwurf rechtzeitig berücksichtigt und geschlossen werden können.

Dieses Paper beschreibt ein Verfahren der SILVER ATENA zur systematischen Klassifizierung und Bewertung von Angriffsrisiken auf komplexe und vernetzte Systeme, die sicherheitskritische Funktionen erfüllen und schützenswerte Daten verarbeiten. Dabei werden zunächst Risiken, die nicht relevant für die Betriebssicherheit sind, von solchen getrennt, die die funktionale Sicherheit der Systeme kompromittieren. Für letztere bietet das Verfahren weitere Methoden zur Einstufung des Gefahrenpotentials sowie zur Identifikation von Maßnahmen zur Risikominimierung. Deren Ziel ist, entweder die Schwachstellen gegen Angriffe zu sichern, oder die Funktionen gegen vorsätzlich von außen eingebrachte Fehler zu immunisieren. Grundlage für die Methode bilden bereits existierende und erfolgreich angewandte Security-Methoden aus der Luftfahrt, die ursprünglich aus der ISO 27001 [9] abgeleitet wurden. Die ISO 27001 beschreibt grundsätzliche Vorgehensweisen und Rahmenbedingungen für die Gewährleistung der Informationssicherheit. Um aber für ein bestimmtes System das Risiko zu ermitteln bedarf es einer konkreten Methode.

Die Security-Analyse ermöglicht in Teilbereichen eine quantitative Einordnung der Angreifbarkeit, womit das Gefährdungspotential noch besser bewertet werden kann. Dabei ist die Anwendung des SILVER ATENA Verfahrens nicht auf Systeme des Transportsektors beschränkt, sondern kann überall dort eingesetzt werden, wo hochautomatisierte, hochvernetzte, sicherheitskritische Systeme nach Standards entwickelt werden, die von der IEC-61508 abgeleitet sind. Der Prozess ist an einem generischen anforderungsbasierten Entwicklungsprozess orientiert und kann daher auch an den spezifischen Kundenprozess adaptiert werden.

1. Einleitung

In den letzten Jahren ist mehrfach in der Praxis gezeigt worden, dass erfolgreiche Angriffe über drahtlose Schnittstellen auch im Automobilbereich möglich sind [3] (vgl. Fälle „ConnectedDrive“ (BMW) und „Onstar“ (GM)). Ein schwerwiegenderer Vorfall geschah im „U-Connect“-System von Fiat-Chrysler (vgl. „Jeep Hack“ von Miller und Valasek [4, 5, 6]). Hier

wurden Schwachstellen des Systems dazu genutzt, die Firmware des Unterhaltungssystems zu manipulieren. Allen Szenarien gemein ist, dass die entsprechenden Korrekturen der Dienste in eine große Anzahl von bereits verkauften Fahrzeugen integriert werden mussten. In Fällen, in denen Updates nicht über die Internetschnittstelle der Autos geliefert werden konnten, mussten sogar kostspielige Rückrufaktionen veranlasst werden [7]. Dementsprechend kann ein Verkennen der tatsächlichen Bedrohungslage zu umfangreichen rechtlichen und wirtschaftlichen Konsequenzen führen. Eine der aktuellen Herausforderungen im Automobilsektor ist es, einen Patch-Prozess zu entwickeln, um Sicherheitslücken nach der Auslieferung beheben zu können. In erster Linie sollte das Thema Security allerdings bereits in der Vorentwicklung und entwicklungsbegleitend berücksichtigt werden, um Risiken zu klassifizieren und rechtzeitig angemessene Schutzmaßnahmen zu integrieren (vgl. auch [8]).

Häufig werden Security- und Risikoanalysen von Systemen nach abgeschlossener Entwicklungsphase mittels Schwachstellenanalyse bzw. Penetrationstest durchgeführt. Das hat zur Folge, dass bei der Schwachstellenanalyse festgestellte Probleme eventuell nur noch schwer vollständig zu beseitigen sind, da die Entwicklungsphase bereits abgeschlossen ist. Größere eventuell nötige Änderungen der Systemarchitektur sind dann ausgeschlossen. Außerdem verbleiben Schwachstellen, die nicht gefunden werden, unerkannt im System.

Damit es nicht zu dieser Situation kommt, darf Security nicht als nachträgliche Entwicklungsphase verstanden werden, sondern erfordert eine frühestmögliche Einbindung in den Entwicklungsprozess. Dann können Systemarchitekturen, die im Kontext von Security problematisch sind, im Vorfeld ausgeschlossen werden. Im Idealfall beinhaltet die Entwicklung eines Systems vorher festgelegte und dem Anwendungsfall entsprechende Security-Anforderungen.

SILVER ATENA verwendet einen eigenen Entwicklungsprozess, bei dem die Security des Systems in den Prozess integriert ist. Diesem Entwicklungsprozess liegt die Normreihe ISA 62443 [1] zugrunde, welche für die Security von industriellen Automatisierungssystemen vorgesehen ist. Der industrielle Kontext dieser Norm bringt eine hohe Anwendbarkeit auch für branchenübergreifende technische Systementwicklung.
Anhand dieser Norm werden die Security-Anforderungen an das zu entwickelnde System aus einem Security-Level abgeleitet, der vorher systematisch bestimmt wird. Dadurch ergeben sich in der Security-Analyse vergleichbare und reproduzierbare Ergebnisse. Außerdem

werden die Security-Anforderungen an das System vor Beginn der Entwicklungsphase bestimmt, sodass sie während der Implementierung effektiv berücksichtigt werden können.

Im Folgenden werden der von SILVER ATENA angewandte Entwicklungsprozess sowie die zugrunde liegende Norm näher beschrieben. Der nächste Abschnitt erläutert die übergeordnete Struktur des Entwicklungsprozesses. Kapitel 2 stellt die verwendete Normreihe vor, und beschreibt ihre relevanten Aspekte. In Kapitel 3 wird das im Rahmen dieses Entwicklungsprozesses entstehende Verhältnis zwischen Security Engineering und System Engineering vorgestellt. Eine abschließende Diskussion mit Berücksichtigung der Vor- und Nachteile des präsentierten Entwicklungsprozesses ist in Kapitel 4 gegeben.

1.1. Struktur des SILVER ATENA Entwicklungsprozesses

Der von SILVER ATENA angewandte Entwicklungsprozess teilt sich auf in die Bereiche Security Engineering und System Engineering. Abbildung 1 stellt das Gesamtkonzept des Prozesses dar, auf dessen Teile wird im Folgenden noch genauer eingegangen.

Der Bereich Security Engineering befasst sich mit der Analyse des Risikos für das zu entwickelnde System, und der Formulierung der daraus resultierenden Security-Anforderungen. Die Bedeutung des Begriffs „Risiko“ wird im folgenden Kapitel noch genauer definiert. Es wird davon ausgegangen, dass für das zu entwickelnde System die Architektur bereits festgelegt ist, da diese für die Risikoanalyse benötigt wird. Zusätzlich zur Systemarchitektur benötigt das Security Engineering für die Risikoanalyse einen vom Kunden bestimmten Wert für das tolerierbare Risiko. Die Festlegung dieses Wertes wird im Folgenden noch näher beschrieben.

Aus der Risikoanalyse des Security Engineerings resultiert ein dem System zugewiesener Security Level. Für diesen Security Level definiert die ISA 62443 technische Security-Anforderungen, die zunächst in Anforderungen auf Systemebene übersetzt werden müssen. Diese Systemanforderungen werden dann als Output des Security Engineerings an den Bereich System Engineering zur Implementierung weitergereicht.

Das System Engineering befasst sich mit der Umsetzung der übergebenen Systemanforderungen. Entwicklungsbegleitend werden Testfälle sowie eine Dokumentation generiert, die zusammen mit dem entwickelten System zurück an das Security Engineering übergeben werden. Anhand dieses Inputs, und unter Berücksichtigung des vorher bestimmten Security

Levels, wird die Umsetzung der Security-Anforderungen mittels Methoden im Sinne von Verifizierung & Validierung überprüft.

Ziel dieser Tests ist in erster Linie eine Validierung der systematisch bestimmten Security-Anforderungen. Zusätzlich dazu sollen Schwachstellen identifiziert und bewertet werden. Ähnlich der Vorgehensweise der funktionalen Sicherheit können hier Iterationen mit dem Systems Engineering durchgeführt werden um ein für den Kunden zufriedenstellendes Maß an Security zu erzielen.. Ist dieser Punkt erreicht, wird ein Dokument generiert, welches die Security-Analyse, die resultierende Anforderungen und deren Umsetzung zusammenfassend darstellt. Dieses Dokument dient dem Kunden als Informations- und Orientierungshilfe hinsichtlich der in Auftrag gegebenen Security-Analyse und ihrer Umsetzung.

Den Bereich des Security Engineerings übernehmen wir als SILVER ATENA komplett. Unsere fundierten branchenübergreifenden Erfahrungen mit der Anwendung dieser Methodik erlauben eine effiziente und normgerechte Vorgehensweise.

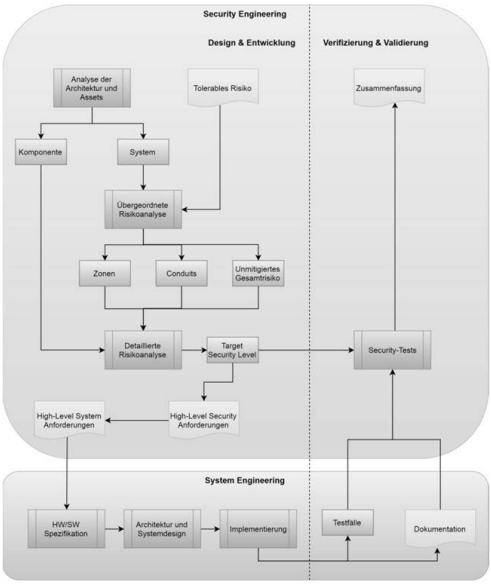

Bild 1: SILVER ATENA Security Engineering

Da wir abgesehen von den Anforderungen an das zu entwickelnde System auch Security-Anforderungen an den Entwicklungsprozess selbst stellen, übernehmen wir idealerweise auch die Aufgaben aus dem Bereich des System Engineerings. Optional sieht dieser Prozess auch die kundenseitige Übernahme dieser Aufgabe vor. In diesem Fall werden die Schnittstellen des Entwicklungsprozess mit SILVER ATENA abgestimmt, Anforderungen an den Prozess werden formuliert, und bei Bedarf wird der Entwicklungsprozess zur Erfüllung der Anforderungen entsprechend angepasst.

Das übergeordnete Ziel dieser Methodik ist es, den Entwicklungsprozess mit den in der Risikoanalyse abgeleiteten Security-Anforderungen so zu leiten, dass ein vorgesehener Security-Level effektiv umgesetzt wird. Dies erfordert eine enge Kooperation mit den zuständigen Systementwicklern.

2. Normgerechtes Security Engineering

Im Bereich der funktionalen Sicherheit haben sich normgerechte Entwicklung und Zertifizierung schon seit Jahrzehnten als Standardverfahren bei der Systementwicklung in unterschiedlichen Industriezweigen etabliert. Dies wäre auch im Kontext von Security wünschenswert, damit Entwicklungsprozesse der stetig ansteigenden Bedrohungslage entsprechen.

Standardisierte Vorgehensweisen erleichtern die Entwicklung sowie die Zertifizierung von Systemen, und wirken sich damit positiv auf die Umsetzung festgelegter Anforderungen aus. Dies trifft sowohl auf rechtliche, als auch auf industrielle oder wissenschaftliche Anforderungen und Richtlinien für die zu entwickelnden Systeme zu.

SILVER ATENA unterstützt die fortlaufenden Standardisierungs- und Zertifizierungsvorhaben im Bereich Security. In diesem Sinne orientiert sich unser hier beschriebener Entwicklungsprozess an der industriellen Normreihe ISA 62443 für die Security von industriellen Automatisierungs- und Steuerungssystemen. Unsere Erfahrungen mit dieser Normreihe belegen branchenübergreifend eine hohe technische Anwendbarkeit.

In den folgenden Abschnitten stellen wir diese Normreihe vor, benennen benötigte Eingangsgrößen, und erläutern die Anwendung im Rahmen unseres Security Engineerings.

2.1. Normreihe ISA 62443

Die Normreihe der ISA 62443 ist eine neuere und noch weniger bekannte US-amerikanische Industrienorm für die Security von industriellen Automatisierungs- und Steuerungssystemen, die aber bereits teilweise als europäische Norm (DIN IEC 62443-3-3) übernommen wurde. Sie besteht aus mehreren Teilen, die jeweils einen spezifischen Fokus im Kontext der Security von industriellen Automatisierungs- und Steuerungssystemen haben (siehe Abbildung 2). Im Folgenden wird die Struktur der Normreihe genauer beschreiben.

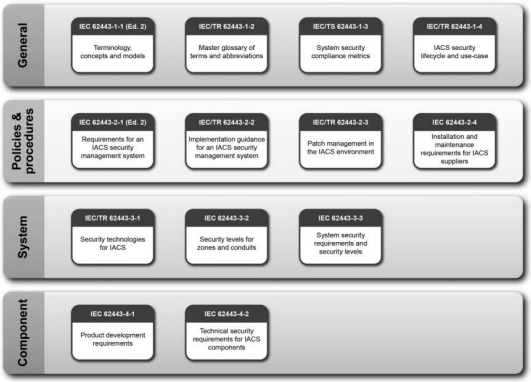

Bild 1: Struktur der Normreihe ISA 62443 (Quelle: isa.org)

- „Models and Concepts" (62443-1-1): Beschreibt die Konzepte und Modelle, die die Grundlage für alle Dokumente der Normreihe bilden.

- „Master Glossary" (62443-1-2): Definiert die Begriffe und Abkürzungen, die in der Normenreihe verwendet werden. Dort wo es möglich ist werden Definitionen von bestehenden Quellen herangezogen und bei Bedarf leicht angepasst um dem Kontext der industriellen Automatisierungs- und Steuerungssysteme zu entsprechen.

- „Cyber Security System Conformance Metrics" (62443-1-3): Hierbei handelt es sich um einen prozess-basierten Standard, der die Konformitätsmetriken bezüglich der Security des Systems sowie ihre Anwendung spezifiziert. Es wird allerdings keine Bewertung hinsichtlich Validität oder Genauigkeit durchgeführt.

- „Industrial Automation and Control System Security Management System" (62443-2-1): Definiert Anforderungen an die Entwicklung eines Security Management Systems für in-

dustrielle Automatisierungs- und Steuerungssysteme, und bietet entwicklungsbegleitende Empfehlungen.

- „Implementation Guidance for IACS Security Management System" (62443-2-2): Behandelt den Betrieb eines effektiven Cyber Security Programms für industrielle Automatisierungssysteme.
- „Security Technologies for Industrial Automation and Control Systems" (62443-3-1): Bietet eine aktuelle Beurteilung gängiger Cyber Security Tools, Gegenmaßnahmen, und Technologien, die effektiv auf moderne elektronisch-basierte industrielle Automatisierungs- und Steuerungssysteme angewendet werden können.
- „Security Risk Assessment and System Design" (62443-3-2): Legt Anforderungen fest für die Definition des zu betrachtenden Systems, die Unterteilung in Zonen und Conduits, die Risikobeurteilung, die Festlegung des Ziel Security Niveaus, und die Dokumentation der Security Anforderungen.
- „System Security Requirements and Security Levels" (62443-3-3): Legt detaillierte technische Systemanforderungen an industrielle Automatisierungs- und Steuerungssysteme fest.
- „Secure Product Development Lifecycle Requirements" (62443-4-1): Spezifiziert Anforderungen an einen sicheren Entwicklungs- und Lebenszyklus von Produkten für industrielle Automatisierungs- und Steuerungssysteme. Diese Anforderungen betreffen die Entwickler und Betreiber der Produkte, nicht aber die Benutzer.
- „Technical Security Requirements for IACS Components" (62443-4-2): Formuliert detaillierte technische Anforderungen an Komponenten industrieller Automatisierungs- und Steuerungssysteme, diese Anforderungen bauen auf den vorher aufgestellten Anforderungen auf.

Industrielle Automatisierungs- und Steuerungssysteme unterscheiden sich hauptsächlich in ihrer Größe von Systemen anderer technischer Industriezweige wie z.B. Automotive, Luftfahrt, etc. Bezüglich der zugrunde liegenden Systemarchitekturen und IT Infrastrukturen gibt es allerdings zahlreiche Gemeinsamkeiten. Ähnlich einer Industrieanlage müssen in einem modernen Fahrzeug (Auto, Zug, Flugzeug) zahlreiche Sensoren, Aktuatoren, und Steuergeräte miteinander kommunizieren um den sicheren (Safety & Security) Betrieb des jeweiligen Systems zu gewährleisten. Durch die zunehmende Komplexität dieser Systeme, z.B. in der Form von Fahrassistenzsystemen, steigt auch die Anzahl der verbauten Komponenten, sowie die zur Verfügung stehenden IT Ressourcen. Ein damit einhergehender Anstieg der Au-

tomatisierung der Verwaltung, Steuerung, und Kommunikation dieser Funktionalitäten wird unausweichlich.

Aufgrund der bereits erwähnten strukturellen Gemeinsamkeiten hinsichtlich der IT und des Automatisierungsgrades haben bereits etablierte Normen industrieller Automatisierungs- und Steuerungssysteme eine hohe Anwendbarkeit bzgl. der Security dieser Systeme. Aus diesem Grund greift SILVER ATENA auf die ISA 62443 zurück, und wendet diese im Kontext von Security branchenübergreifend an.

2.2. Notwendige Eingangsgrößen

Die Norm sieht es vor, für ein zu entwickelndes bzw. bestehendes System eine Security Analyse auf zwei Ebenen durchzuführen. Mit einer Systemarchitektur und einem tolerierbaren Risikowert als Eingangsgrößen wird zunächst eine übergeordnete Risikoanalyse durchgeführt. Die Systemarchitektur dient in dieser Analyse der Unterteilung des Systems in Zonen und Conduits mit ähnlichen Funktionen, Komponenten, bzw. mit ähnlich zu realisierenden Schutzmechanismen. Für diese Zonen und Conduits wird im Weiteren eine detailliertere Analyse durchgeführt.
Der tolerierbare Risikowert dient in der übergeordneten Analyse der Bestimmung eines unmitigierten Gesamtrisikos für das betroffene System, anhand dessen dann eine Priorisierung von Bedrohungsszenarien in der detaillierten Analyse erfolgen kann.

Idealerweise ist der tolerierbare Risikowert ein vom Kunden selbst bestimmter Wert, der das für das gewünschte System akzeptable Risiko quantifiziert. Erfahrungsgemäß haben viele Kunden aber selbst nicht die Erfahrung, solche Risikowerte ihren Bedürfnissen und Wünschen entsprechend zu bestimmen. Da auch die Norm diesbezüglich keine Hilfestellung bietet hat SILVER ATENA eigene Methoden entwickelt, um das tolerierbare Risiko den Kundenwünschen entsprechend zu erfragen, ohne jegliche Erfahrung mit der Norm oder mit Security im Allgemeinen vorauszusetzen.

Des Weiteren können bei der Bestimmung von Bedrohungsszenarien, die verschiedene Zonen, Conduits, oder Komponenten betreffen, externe Quellen zu Hilfe gezogen werden wie z.B. Bedrohungskataloge des Bundesamtes für Sicherheit in der Informationstechnik (BSI) [2]. Erfahrungsgemäß decken solche Bedrohungskataloge Basisbedrohungen relativ gut ab. Allerdings sollte sich die Analyse der Bedrohungsszenarien nicht ausschließlich auf Bedrohungskataloge stützen. Solche Kataloge sind meist generischer Natur, d.h. sie sind allge-

mein formuliert und decken nicht die spezifischen Bedrohungen für das jeweils betroffene System ab. Ebenso können sie für den jeweiligen Anwendungsfall weniger relevante Bedrohungsszenarien enthalten, deren Analyse und Vermeidung unnötigen Aufwand verursachen, und nicht zur Security des Systems beitragen.

SILVER ATENA führt daher für alle relevanten Schnittstellen eine jeweils spezifisch auf das betroffene System zugeschnittene detaillierte Bedrohungsanalyse durch, um die Angriffsfläche des Systems bestmöglich zu ergründen.

2.3. Anwendung der Norm

Wie bereits erwähnt sieht die Norm die Durchführung einer Risikoanalyse in zwei Teilen vor. Zunächst wird eine übergeordnete Analyse durchgeführt, in der das betroffene System in Zonen und Conduits aufgeteilt wird. Zonen und Conduits sind Gruppierungen von Komponenten mit ähnlichen Funktionen oder ähnlichem Schutzbedarf. Dabei werden Conduits jene Komponenten eingeordnet, die eine Kommunikation zwischen Zonen herstellen.
Außerdem wird in der übergeordneten Risikoanalyse ein Wert für das unmitigierte Gesamtrisiko bestimmt. Dieser wird mithilfe der Formel

$$R_u = H * K$$

berechnet, wobei *H* die maximale Häufigkeit eines der Bedrohungsszenarien und *K* die Kritikalität des betroffenen Schutzgutes beschreibt. Der unmitigierte Risikowert wird auch als worst-case Risiko bezeichnet. Angenommen es werden drei Häufigkeits- und Kritikalitätskategorien festgelegt. Dann stellt die in Tabelle 1 dargestellte Risikomatrix alle möglichen Werte für das unmitigierte Risiko dar.

Dieser Risikowert gibt einen ersten Hinweis bezüglich der Verwundbarkeit des Systems in seinem momentanen (evtl. konzeptionellen) Zustand, und erlaubt eine Priorisierung von Bedrohungsszenarien und Schutzgütern in der folgenden detaillierten Analyse. Des Weiteren gibt der unmitigierte Risikowert eine erste Einschätzung des zu erwartenden Aufwandes um der entsprechenden Bedrohungen entgegen zu wirken.

Die Häufigkeiten der erfassten Bedrohungsszenarien werden in der übergeordneten Security-Analyse näherungsweise bestimmt. Zur Bewertung der Kritikalität von Schutzgütern werden je nach Anwendungsfall unterschiedlich abgestufte Schadenskategorien festgelegt, diese reichen z.B. von geringfügiger Betriebsbeeinträchtigung über wirtschaftlichen Schaden bis hin zu Gefährdung von Personen.

Tabelle 1: Übergeordnete Risikomatrix

Häufigkeit / Kritikalität	Selten (1)	Wenig (2)	Häufig (3)
Niedrig (1)	1	2	3
Mittel (2)	2	4	6
Hoch (3)	3	6	9

Die Unterteilung der übergeordneten Security-Analyse des Gesamtsystems in Zonen und Conduits geschieht nur, sofern die Analyse für ein System, und nicht für eine einzelne Komponente durchgeführt wird. Die Norm gibt konkrete Vorgaben dazu, nach welchen Kriterien diese Unterteilung stattfinden soll. Grundsätzlich werden Komponenten mit ähnlicher Funktion oder ähnlichem Schutzbedarf zu Zonen bzw. Conduits zusammengefasst.

Für jede Zone und jedes Conduit aus der übergeordneten Risikoanalyse wird anschließend eine detaillierte Analyse durchgeführt und ein neues unmitigiertes Risiko berechnet. Für die genauere Bestimmung der Häufigkeiten von Bedrohungsszenarien in der detaillierten Analyse gibt die Norm keine eindeutig festgelegte Vorgehensweise vor. Hier hat SILVER ATENA ein systematisches Verfahren zur Bestimmung dieser Werte entwickelt, welches sich am „Common Vulnerability Scoring System“ CVSS [10] orientiert. Das Common Vulnerability Scoring System ist ein Industriestandard zur Bewertung des Schweregrades von möglichen oder tatsächlichen Sicherheitslücken in Computer-Systemen.

Dabei wird jedes Bedrohungsszenario anhand der Kriterien Angriffsvektor, Zugangsdaten, Zeitfenster, und Expertise jeweils mit einer entsprechenden Punktzahl bewertet. In der Summe ergibt sich eine Punktzahl, die auf eine abgestufte Häufigkeitsskala abgebildet wird. Die erreichte Punktzahl legt damit eine angenommene Häufigkeit für das jeweilige Bedrohungsszenario fest.

Mithilfe dieser Häufigkeiten und der zuvor bestimmten Kritikalitäten kann für alle Zonen und Conduits ein individueller Risikowert bestimmt werden. Aus allen anwendbaren Bedrohungsszenarien ist für eine Zone bzw. Conduit der maximal erreichte Risikowert ausschlaggebend.

Auf Grundlage dieser Risikowerte wird anhand der von der Norm vorgegebenen Berechnung für alle Zonen und Conduits ein Security-Niveau bestimmt. An dieser Stelle wird der tolerier-

bare Risikowert verwendet. Zur Erreichung des sich ergebenden Security-Niveaus gibt die Norm konkrete Security-Anforderungen vor, die umgesetzt werden müssen.

Diese Security-Anforderungen werden von SILVER ATENA in Systemanforderungen übersetzt. Die Übersetzung findet so statt, dass die zuständigen Entwickler die Systemanforderungen auch ohne Kenntnis oder Erfahrungen im Bereich Security umsetzen können.

3. Übergabe an System Engineering

Idealerweise wird auch das System Engineering durch SILVER ATENA übernommen. Ist dies aber nicht möglich, kann unser Verfahren trotzdem effektiv angewendet werden. Hierzu muss der Entwicklungsprozess von den Verantwortlichen so dokumentiert und an uns kommuniziert werden, dass eine Bewertung dieses Prozesses hinsichtlich der Security durchgeführt werden kann. Bei Bedarf werden dann Anforderungen an den Entwicklungsprozess selber gestellt und von den Verantwortlichen durchgesetzt. Diese Vorgehensweise wird durch die modulare Aufteilung des Gesamtkonzepts in Security Engineering und System Engineering ermöglicht. Des Weiteren wird davon ausgegangen, dass externe Entwickler ihren Entwicklungsprozess an etablierten Methoden orientieren, wie z.B. agilen Prozessen oder dem V-Modell.

Im Sinne der modularen Auslegung unseres Entwicklungsprozesses werden die Systemanforderungen, die wir aus den Security Anforderungen abgeleitet haben, an das System Engineering übergeben, sofern wir dieses nicht selbst übernehmen. Ziel ist es, dass die aus der Security-Analyse gefolgerten Anforderungen an das System möglichst auf System-naher Ebene an das System Engineering kommuniziert werden.

Eine weitere Schnittstelle zwischen den Bereichen Security Engineering und System Engineering ist die Übergabe des entwickelten Systems zusammen mit Testfällen und einer Dokumentation an das Security Engineering. Aus der Dokumentation muss ersichtlich sein, wie und ob die vorher übergebenen Systemanforderungen umgesetzt wurden. Das Security Engineering überprüft im Folgenden ebenfalls, ob die umgesetzten Systemanforderungen tatsächlich die in der Risikoanalyse festgelegten Security-Anforderungen effektiv realisieren. Dies geschieht auch im Rahmen von Security-Tests, in welche abgesehen von den Testfällen und der Dokumentation auch der in der Risikoanalyse bestimmte Security-Level mit einfließt. In der Regel liegt der Quellcode des entwickelten Systems SILVER ATENA nicht vor, daher können die Security-Tests lediglich als Blackbox-Tests durchgeführt werden.

Sollten die Security-Anforderungen an dieser Stelle nicht zufriedenstellend umgesetzt sein, wird zunächst die Ursache formuliert und an das Systems Engineering kommuniziert. Beispielsweise können Systemanforderungen nicht vollständig umgesetzt sein, oder Security-Anforderungen wurden nicht eindeutig bzw. nicht entsprechend ihrer Bedeutung in Systemanforderungen übersetzt. Diese Validierung und Kommunikation ist entscheidend für den Erfolg des Security Engineerings und kann ggf. in einer gemeinsamen Iteration mit dem Systems Engineering optimiert werden.

Sind die in der Risikoanalyse abgeleiteten Security Anforderungen zufriedenstellend innerhalb des System Engineerings umgesetzt, erfolgt im Rahmen des Security Engineerings eine nachvollziehbare Zusammenfassung der durchgeführten Risikoanalyse, der abgeleiteten Security Anforderungen, sowie der Umsetzung dieser Anforderungen (mithilfe der Dokumentation des System Engineerings).

4. Diskussion

Das übergeordnete Ziel unseres Entwicklungsprozesses ist die Unterstützung und Etablierung standardisierter und normgerechter Vorgehensweisen, die sowohl die Systementwicklung als auch die Zertifizierung im Kontext von Security im Automobilsektor und darüber hinaus vereinheitlichen und erleichtern sollen.

Die angesprochenen Beispiele erfolgreicher Angriffe gegen Automobilsysteme [3, 4, 5, 6] belegen einen nicht zu vernachlässigenden Handlungsbedarf im Bereich von Security. Durch die zunehmende Komplexität und Vernetzung von Fahrzeugen ist weiterhin mit einer Vergrößerung der Angriffsflächen zu rechnen. Diese Bedrohungslage erfordert Gegenmaßnahmen, die in die jeweiligen Entwicklungsprozesse integriert sind.

Unser Security Engineering ist modular aufgebaut, und vom System Engineering losgelöst. Dadurch genießt der Kunde eine hohe Flexibilität in Bezug auf die konkrete Umsetzung des Projektes. Das System Engineering kann vom Kunden selbst, von uns oder auch von Dritten Parteien übernommen werden. In jedem Fall findet eine Kommunikation zwischen dem System Engineering und dem von SILVER ATENA durchgeführten Security Engineering statt.
Wird das System Engineering allerdings extern durchgeführt und der Quellcode nicht vorgelegt, kann im Rahmen unserer Security-Tests nur ein Blackbox-Testing erfolgen. Wie schon ein der Einleitung erwähnt ist diese Art der Verifikation von Security mit einigen Nachteilen behaftet.

Die dem Security Engineering zugrunde gelegte Industrienorm ISA 62443, bringt zudem branchenübergreifend eine hohe technische Anwend- und Anpassbarkeit mit sich. Die bereits teilweise abgeschlossene Übernahme der ISA 62443 als europäische Norm (DIN IEC 62443-3-3) belegt eine zunehmende Kenntnis und Akzeptanz dieser Norm auch im europäischen Wirtschaftsraum.

Durch ihre abstrakten und von der Systemebene losgelösten Richtlinien werden Konflikte in der Anwendung im Vorfeld auf ein Minimum reduziert. Allerdings ist es möglich, dass Kunden nicht die Erfahrung besitzen, den nötigen Input für das Security Engineering zur Verfügung zu stellen (z.B. tolerierbares Risiko). Für diesen Fall hat SILVER ATENA Verfahren entwickelt, um den Kunden bei der Erfassung der benötigten Informationen zu unterstützen.

Des Weiteren bringt die generische industrielle Ausrichtung der Norm den Vorteil, dass auch kleinere unabhängige Systeme wie z.B. Fahrzeuge sehr gut erfasst werden. In modernen Fahrzeugen sind zahlreiche Mikroprozessoren, Sensoren, Aktuatoren, Steuergeräte, und Kommunikationsnetzwerke verbaut, die im laufenden Betrieb unterschiedliche Aufgaben verteilt ausführen. In dieser Hinsicht kann ein solches Fahrzeug als mikroskopische Industrieanlage verstanden werden, in der physische Prozesse mithilfe von IT Infrastrukturen gesteuert werden. Die von der Norm vorgesehene Unterteilung des Systems in Zonen und Conduits ist daher sehr gut realisierbar. Aus diesem Grund bietet sich speziell für den Bereich Automotive die Anwendung dieser bereits etablierten Industrienorm sehr erfolgsversprechend an. Unsere fundierten Erfahrungen mit der Anwendung der ISA 62443 in zahlreichen ursprünglich nicht dafür vorgesehenen Branchen bestätigen diesen Sachverhalt.

Ein weiterer Vorteil der ISA 62443 sind die konkreten Anforderungskataloge an erreichte Security-Niveaus, die von technischen Details losgelöst festlegen, welche Security-Anforderungen umzusetzen sind. Damit lässt die Norm bei der Übersetzung der Security-Anforderungen in Systemanforderungen sehr viel Freiraum. Dieser ermöglicht ein passgenaues Zuschneiden der Anforderungen auf das vorliegende System. Auch dies ist ein Punkt, der die branchenübergreifende Anwendbarkeit dieser Norm belegt.

Dadurch kann es allerdings auch zu Situationen kommen, in denen die korrekte Anwendung der Norm nicht eindeutig spezifiziert ist. Dies ist meist in neuen Anwendungsgebieten der Fall, bei denen Systemeigenschaften auftreten, die in der Norm nicht direkt berücksichtigt sind. Beispielsweise kann vom Kunden das Security Engineering lediglich für eine einzelne

Komponente anstatt für ein Gesamtsystem gewünscht sein. Die Norm ist dann trotzdem anwendbar, aber das Vorgehen muss in diesen Fall durch SILVER ATENA modifiziert werden.

Aufgrund der langjährigen Erfahrung im Bereich Security Engineering kann SILVER ATENA auf ein breites Methodenspektrum zurückgreifen um Spezifizierungslücken der Norm durch eigene Anpassungen bzw. Ergänzungen zu schließen.

5. Quellenverzeichnis

[1] ISA 62443 Security for Industrial Automation and Control Systems , http://isa99.isa.org/Public/Information/The-62443-Series-Overview.pdf

[2] Bundesamt für Sicherheit in der Informationstechnik, https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/itgrundschutzkataloge_node.html

[3] heise.de, BMW ConnectedDrive gehackt, https://www.heise.de/security/meldung/BMW-ConnectedDrive-gehackt-2533601.html

[4] Ingenieur.de, Hacker steuern fahrenden Jeep Cherokee aus der Ferne übers Inter-net, http://www.ingenieur.de/Themen/Automobil/Hacker-steuern-fahrenden-Jeep-Cherokee-Ferne-uebers-Internet

[5] Greenberg, A.; Hackers Remotely Kill a Jeep on the Highway – with me in it, https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[6] Miller, C; Valasek, C, Remote Exploitation of an Unaltered Passenger Vehicle, Black Hat USA 2015

[7] Wever, (2006) Die Zukunft des Autos – das Auto der Zukunft: Wird der Computer den Menschen ersetzen?, Arbeitspapier zum Vortrag auf dem Branchenforum Frei-er Autoreparaturmarkt am 15. Januar 2006, Dortmund, ISSN 1612-5355

[8] Robinson-Mallett, C.; Hansack, S.; A Model of an Automotive Security Concept Phase, Proceedings of the 10th Annual Cyber and Information Security Research Conference, 2015

[9] ISO 27001, Ausgabe 2013-10, Information technology - Security techniques - In-formation security management systems – Requirements

[10] Common Vulnerability Scoring System v3.0: Specification Document, https://www.first.org/cvss/specification-document

Cyber Resilience im Rahmen von vernetzten Fahrzeugen

Felix-Jacob Siever, B.Sc., Dipl.-Ing. **Jochen Sandvoß**,
MHP Management und IT-Beratung GmbH, Ludwigsburg

Einleitung

Durchschnittlich enthält ein Fahrzeug über 10 Millionen Zeilen Softwarecode und etwa 60 Mikroprozessoren. Eine Boeing der Bauart 787 Dreamliner enthält im Vergleich ungefähr 18 Millionen Zeilen Softwarecode.[1] In Anbetracht der Komplexität des Produktes sticht die ausgeprägte Digitalisierung des Fahrzeugs hervor. Fahrzeuge in der heutigen Zeit werden immer mehr zu fahrenden Computern die mit anderen Systemen, wie bspw. dem Mobiltelefon, einem Backend-System des Herstellers oder auch mit verschiedenen Plattformen von Drittanbietern kommunizieren. Grund für die steigende Vernetzung und Digitalisierung ist das stark wachsende Marktpotenzial solcher Fahrzeuge.[2]

Dafür verantwortlich ist unter anderem die Bereitschaft der Kunden, ein solches Fahrzeug zu besitzen und für den entsprechenden Mehrwert zu zahlen. Laut einer Studie von Deloitte können sich 76% der befragten Personen zwischen 18 und 30 Jahren vorstellen ein vernetztes Fahrzeug zu nutzen.[3] Aus Sicht der Hersteller und Anbieter von Mobility Services, ist vor allem aber die Möglichkeit in einem solchen vernetzten Fahrzeug Daten zu sammeln, diese selbst zu verwenden oder zu vermarkten ein ausschlaggebender Grund.

Die erhobenen Daten sind nichts anderes als Informationen und diese bilden den Grundbaustein für viele neue Themenbereiche wie ‚Industrie 4.0', Big Data' oder ‚Internet of Things' sowie weitere neue Geschäftskonzepte (z.B. im Bereich der Autoversicherung – „pay as you drive").

Durch die Vernetzung der Kraftfahrzeuge mit externen IT-Systemen entstehen neben den neuen Chancen, neue Risiken für Fahrer, andere Verkehrsteilnehmer und Hersteller. Einem Hackerkollektiv gelang es bspw. die vollständige Kontrolle über einen Jeep Cherokee zu

[1] Vgl. Anand 2015 o.S.

[2] Vgl. Viereckl et al. 2014 S.3f

[3] Vgl. Deloitte & Touche GmbH Wirtschaftsprüfungsgesellschaft 2015 S.8f

erlangen.[4] Der Zugriff konnte dabei sogar drahtlos erfolgen. 54% der Teilnehmer zwischen 18 und 30 Jahren geben in einer Studie an, Angst vor einem unerlaubten Zugriff in ihrem Fahrzeug zu haben.[3]

Solche Angriffe wirken harmlos, wenn man sich vorstellt dass künftig auch ganze Infrastruktursysteme (Ampeln, Straßenschilder etc.) und Fahrzeugflotten angegriffen werden könnten. Der wirtschaftliche Schaden, für Automobilhersteller und viele verwandte Branchen, der durch solche Zugriffe entstehen kann, ist enorm. Viel schlimmer als ein monetärer Schaden der durch solche Angriffe entstehen kann, ist die Gefahr für Leib und Leben der einzelnen Verkehrsteilnehmer.

Definition vernetztes Fahrzeug

Für den Begriff vernetztes Fahrzeug, nachfolgend und in der einschlägigen Literatur auch Connected Car genannt, existiert in der Fachwelt bisher keine einheitliche Definition.[5] In der Automobilbranche verwenden Hersteller sowie Zulieferer und andere Beteiligte unterschiedliche Definitionen. Eine Beschreibung, die alle spezifischen Eigenschaften eines vernetzten Fahrzeugs beinhaltet, ist nach dem Verfasser die folgende:

Ein vernetztes Fahrzeug ist ein Kraftfahrzeug, welches über eine Verbindung, deren Art nicht definiert ist, verfügt und mit Hilfe derer, mit verschiedenen Kommunikationspartnern der Außenwelt, interagiert.

Streng genommen ist somit ein Fahrzeug mit einem Funkschlüssel bereits ein vernetztes Fahrzeug („car to mobile device“). Nachfolgend werden die Kommunikationspunkte des Fahrzeugs mit der Außenwelt dargestellt, sowie die wichtigsten Kommunikationspartner kurz erläutert (Abbildung 1).

[4] Vgl. Miller und Valasek 2015 S.1ff

[5] Vgl. Vogt 2014 S.7

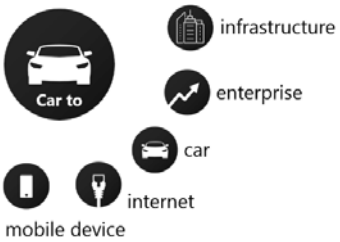

Bild 1: Car to X Kommunikation

‚Car to infrastructure' bezeichnet die Kommunikation zwischen dem Automobil und Infrastrukturelementen. Dazu zählen intelligente Verkehrszeichen, Lichtzeichenanlagen, Knotenpunkte der Kommunikationsinfrastruktur entlang der Verkehrswege (kurz RSUs, Roadside Units) oder Verkehrsleitzentralen, die eine Internetverbindung bereitstellen. Diese Art der Kommunikation ist besonders im Bereich des automatisierten Fahrens von besonderer Bedeutung. [6]

‚Car to car' bezeichnet die direkte Kommunikationsverbindung zwischen Fahrzeugen. Durch die Kommunikation von Fahrzeugen untereinander kann die Sicherheit im Straßenverkehr stark verbessert werden. Dies findet bspw. mittels direkter Warnhinweise (z.B. Unfallmeldungen oder glatte Straßen) statt, die von sich in der Nähe befindlichen Fahrzeugen versendet wurden. Dies hat zur Folge, dass nicht nur die Verkehrssicherheit verbessert, sondern auch der Verkehrsfluss optimiert werden kann.[7]

Unter ‚Car to enterprise' wird die Kommunikation des Autos mit kommerzieller Infrastruktur zusammengefasst. Dazu zählen unter anderem Parkhäuser, Werkstätten, Tankstellen, Hotels und andere POIs (points of interest, Orte von Interesse). Dadurch könnte die Bezahlung an Tankstellen und Parkhäusern automatisiert oder das drahtlose Auslesen der Steuergerätedaten im Automobil realisiert werden. [8] ‚Car to enterprise' umfasst ebenso die Kommunikation des Fahrzeugs mit den Backendsystemen des Fahrzeugherstellers (oder Drittfirmen wie Versicherungsgesellschaften).

[6] Vgl. Johanning und Mildner 2015 S.15; Limke und Wiesner o.S.

[7] Vgl. Vogt 2014 S.14ff; Limke und Wiesner o.S.

[8] Vgl. Johanning und Mildner 2015 S.16; Limke und Wiesner o.S.

Informationssicherheit in der Automobilbranche

Die Aufgabe der Informationssicherheit, als operatives Geschäftsfeld im Bereich der Automobilbranche, unterscheidet sich im Kern nicht von ihrer Aufgabe in anderen Branchen. Die Informationssicherheit hat die Aufgabe, die Schutzziele Vertraulichkeit, Verfügbarkeit und Integrität in informationsverarbeitenden und –lagernden Systemen sicherzustellen.[9] Dabei spielt es keine Rolle, ob es sich hierbei um technische oder nicht-technische Systeme handelt.

Durch die in den letzten Jahren stark gestiegene Vernetzung des Autos spielt nicht nur die funktionale Sicherheit eine große Rolle, sondern zunehmend auch die Angriffssicherheit.

Bei bisherigen Betrachtungen wurde die Sicherheit eines vernetzten Fahrzeugs vor allem durch die **funktionale Sicherheit** abgedeckt. Dies war nicht Aufgabe der Sicherheitsabteilung des Fahrzeugherstellers, sondern Aufgabe der jeweiligen Fachabteilung, die für ein oder mehrere Bauteile verantwortlich ist.

Aspekte der **Angriffssicherheit** können nicht mit herkömmlichen funktionalen Sicherheitsanalysen oder -betrachtungen abgedeckt werden. Der Unterschied zwischen Angriffssicherheit und der funktionalen Sicherheit ist die Analyse potentieller, beabsichtigter oder geplanter Angriffe, deren Zielsetzung es ist, negative Konsequenzen für Fahrer, Fahrzeug, Hersteller oder andere Beteiligte herbeizuführen. Aufgabe ist es, das Bedrohungspotential möglichst früh zu identifizieren und damit verbundene Risiken direkt zu reduzieren. Dies ist nur möglich wenn der Prozess der Gefährdungsanalyse im Prozess der Fahrzeugentwicklung fest implementiert wird. Das hat auch Auswirkungen auf die informationstechnische Organisation eines Automobilherstellers. Auf entsprechende Sicherheitslücken in der Software muss während des Entwicklungsprozesses reagiert, und Gegenmaßnahmen müssen entwickelt werden. In der Luftfahrt existieren schon länger solche entwicklungsbegleitenden Prozesse die aus den Standards ISO 27001 und ISO 27005 abgeleitet wurden.[10] Diese Prozesse gelten aber eher als träge und sind für die Automobilindustrie nur bedingt geeignet, da die Entwicklungszyklen in der Automobilindustrie wesentlich kürzer sind als die in der Flugzeugindustrie.

[9] Vgl. Bundesamt für Sicherheit in der Informationstechnik 2008 S.12f

[10] Vgl. Gemeinschaftstagung Automotive Security et al. 2015 S.169ff

Die gestiegene Vernetzung und Digitalisierung führt folglich zur Erschließung neuer Sicherheitsaspekte, da verschiedene Komponenten immer mehr miteinander interagieren, also gemeinsam betrachtet werden müssen. Reguläre Schwachstellen in Applikationen bekommen durch die Vernetzung eine größere Tragweite: Nicht nur IT-Systeme sondern auch die Fahrzeugfunktionen und damit Leib und Leben eines Nutzers (nachfolgend auch als ‚safety' bezeichnet) können bedroht werden. Dadurch ergeben sich verschiedenste Herausforderungen für Automobilhersteller, Zulieferer und Normungsgremien aus Sicht der Sicherheit.

Durch die Wahrung der Schutzziele Vertraulichkeit, Verfügbarkeit und Integrität wird also unmittelbar die ‚safety' aller Verkehrsteilnehmer gewährleistet (Abbildung 2).

Bild 2: Erweitertes Modell der Schutzziele

Resilience Ansätze in der Informationssicherheit

Aus den in den vorherigen Abschnitten beschriebenen Entwicklungen ergeben sich neue Anforderungen an die Informationssicherheit. Bisherige IT-Sicherheitskonzepte in dem Bereich waren sehr steif und nicht auf sich schnell ändernde Verhältnisse ausgelegt oder wurden hauptsächlich, wie zuvor beschrieben, von Seiten der funktionalen Sicherheit betrieben. Auch bezieht sich die traditionelle Sicherheit eher auf Prävention und nicht auf schnelle Gegenmaßnahmen.

Im Oxford English Dictionary wird resilience wie folgt definiert: „the quality or fact of being able to recover quickly or easily from, or resist being affected by, a misfortune, shock, illness etc.: robustness; adaptability" Resilience bedeutet also frei übersetzt Widerstandsfähigkeit.

Ein Cyber Resilience Konzept geht über die bisherigen meist präventiven, technischen Sicherheitsmaßnahmen hinaus – Was ist also Cyber Resilience?

Cyber Resilience beschreibt die Fähigkeit Angriffen vorzubeugen, diese zu erkennen und im Falle einer erfolgreichen Attacke schnellstmöglich wieder in den normalen Betrieb übergehen zu können – kurz „widerstandsfähig zu sein“

Durch die wachsende Vernetzung von Fahrzeugen werden Automobilhersteller immer mehr zu IT-Firmen, die komplexe verteilte Systeme mit hunderttausenden von mobilen Subsystemen (Fahrzeuge und Fahrzeugkomponenten) entwickeln und betreiben. Durch diese Wandlung ist es nötig, dass Hersteller ihre IT-Sicherheitsabteilungen erweitern und neu strukturieren. Auch Anpassungen an der derzeitigen Fahrzeugentwicklung werden nötig sein, um eine ausreichende Absicherung zu gewährleisten. Auf entsprechende Sicherheitslücken in der Software muss während und nach dem Entwicklungsprozess reagiert werden. Daraus resultieren entsprechende Gegenmaßnahmen. Dafür müssen **neue Prozesse und Methoden** entwickelt werden, um den neuen Bedrohungen effektiv entgegen zu wirken. Aufgabe ist es, das Bedrohungspotential möglichst **früh zu identifizieren** und damit verbundene **Risiken direkt zu reduzieren**. Dies ist nur möglich wenn der Prozess der Gefährdungsanalyse im Prozess der Fahrzeugentwicklung fest implementiert wird. Diese Veränderungen haben auch Auswirkungen auf die informationstechnische Organisation eines Automobilherstellers sowie auf die Zusammenarbeitsmodelle zwischen Zulieferern und Betreibern von Infrastruktursystemen (weitere Erläuterung erfolgt im Abschnitt Service Strategie).

Daher ist das Cyber Resilience Konzept ein umfassender kollaborativer Ansatz der seitens der Geschäftsführung aktiv betrieben werden muss und alle **Organisationseinheiten, Partner und Kunden** mit einbindet.

Um die neuen Cyber-Risiken des Geschäfts, gegen die Chancen und Wettbewerbsvorteile auszugleichen, ist eine unternehmensweite risikobasierte Strategie nötig. Diese muss die Schwachstellen, Bedrohungen und Risiken und deren Einfluss auf Vermögenswerte oder kritische Informationen, proaktiv und rückwirkend verwalten. Als Vermögenswerte werden an dieser Stelle die Fahrzeugflotte und alle damit verbundenen IT-Systeme betrachtet (Backend- und Drittsysteme).

Ein gutes Resilience Konzept sollte folgende Eigenschaften haben (Angelehnt an Rance 2015 S.8):

- Das Resilience Konzept beinhaltet ein klares Verständnis dessen, was einen kritischen Vermögenswert darstellt und wie dieser durch den Schutz von Informationen (Daten) zu schützen ist.
- Ebenso bietet es einen Überblick über die Hauptbedrohungen und Schwachstellen der zu schützenden Fahrzeugflotte und der damit verbundenen IT-Systeme (Hauptbedrohungen und Schwachstellen der Vermögenswerte).
- Es muss möglich sein, die Qualität der Cyber Resilience kontinuierlich zu messen und zu verbessern.
- Das Konzept soll eine unternehmensweite Kommunikationsbasis bilden und eine angemessene Anzahl von Regeln (in der einschlägigen Literatur und nachfolgend controls genannt) und Methoden zur Prävention, Erkennung und Korrektur von Angriffen bereitstellen.

Ein wesentlicher Output eines Resilience-Konzeptes sind konkrete Regeln und Methoden zur Prävention, Erkennung und Korrektur von Angriffen sowie Sicherheitsschwachstellen. Da die Implementierung von Security-Mechanismen in den jeweiligen Fahrzeugkonzepten von den Systemarchitekturen abhängig ist, müssen diese Regeln und Methoden stets lösungsoffenen definiert werden. Eine Vorgabe von konkreten technischen Lösungen wäre innovationshemmend und würde Lösungsräume einschränken. „Monokulturen“ würden entstehen, die zudem im Sinne eines Einzelangriffspunktes (single point of attack) sehr attraktiv für Angreifer wären. Konkret bedeutet das: Regeln müssen Systemunabhängig definiert werden („Verbindungen müssen verschlüsselt werden“). Systemabhängige Lösungen müssen im Rahmen der Cyber Resilience kontinuierlich an den jeweils aktuellen Stand der Technik angepasst werden.

Die Einführung eines Resilience Konzeptes wird in zwei Abschnitte unterteilt, die Entwicklung und die Implementierung. Beide Bereiche orientieren sich am Aufbau der IT Infrasturcture Libary (kurz ITIL). ITIL stellt eine gängige Praxis zur Umsetzung eines IT-Service-Managements dar und bildet damit eine gute Basis, für die Entwicklung eines Resilience Konzeptes. Die verschiedenen ITIL Phasen Strategie, Design, Überführung, Betrieb und Kontinuierliche Verbesserung müssen aus Sicht der Informationssicherheit betrachtet werden. Dabei werden entsprechende Maßnahmen, zur Umsetzung der Cyber Resilience entwickelt sowie besonders relevante Prozesse identifiziert. Da der Umfang dieser Veröffentlichung begrenzt ist, wird im Folgenden nur auf die unbedingt erforderlichen Punkte eingegangen.

Die Kernelemente eines Resilience Konzepts beziehen sich auf die Prävention und Erkennung von Angriffen, sowie die schnellstmögliche Wiederherstellung von Systemen nach einer

Kompromittierung. Für die Umsetzung dieser Kernelemente sind folgende ITIL-Prozesse besonders wichtig:

- Entwicklung des Resilience Konzepts (Service Planung)
  - Service Strategie
- Implementierung des Resilience Konzepts (Service Betrieb)
  - Event-, Incident- und Problem Management
- Kontinuierliche Serviceverbesserung (Prozessübergreifend)

In den kommenden Kapiteln werden die drei Kernprozesse (Event-, Incident- und Problem Management, Kontinuierliche Serviceverbesserung und Service Strategie) unter den Gesichtspunkten der Prävention, Erkennung und Wiederherstellung betrachtet. Ein besonderes Augenmerk wird dabei, auf die zu Beginn dieses Kapitels geschilderten, Eigenschaften eines Resilience Konzepts sowie die Erkennung von Angriffen gelegt.

Service Strategie

In Unternehmen, in denen die IT nach den Best Practice Ansätzen aus ITIL betrieben wird, spielt die ITIL Servicestrategie eine zentrale Rolle bei der Cyber Resilience. Die ITIL Service Strategie definiert eine Strategie für die Bereitstellung von Services für IT-Kunden.[11] Kunden können sich dabei innerhalb der Unternehmung befinden, oder auch extern. Auf Basis einer Analyse der Kundenbedürfnisse legt die Servicestrategie fest, wie die IT-Organisation befähigt werden kann, Kundenbedürfnisse strategieorientiert zu ermöglichen.

Im Rahmen der Cyber Resilience Strategie gilt es, im Kontext der ITIL Servicestrategie, die Relevanz (den Schutzbedarf) der Unternehmenswerte (assets) mit allen beteiligten Stakeholdern abzustimmen. Dabei muss ein Verständnis für die zum Schutz notwendigen Ressourcen geschaffen werden (Stichwort Budgetplanung). Gleichzeitig werden besonders kritische Informationen, Systeme, Prozesse oder Vorgänge identifiziert. Die Cyber Resilience Strategie gibt Regeln an, auf Basis deren gehandelt werden soll. Als Basis dient dafür eine Risikostrategie, die z.B. beschreibt welche Systeme aufgrund erhöhten Risikos redundant ausgelegt werden müssen.

Einen Überblick über die Hauptbedrohungen und Schwachstellen der zu schützenden Systeme und Daten zu erstellen, stellt für Automobilhersteller in der Regel kein großes Problem dar. Entsprechende Prozeduren sind meist fest etabliert. Somit ist es relativ leicht möglich, ein Verständnis dafür zu schaffen, was einen kritischen Wert darstellt und wie dieser durch

[11] Vgl. Olbrich 2008 S.146

den Schutz von Informationen (Daten) zu schützen ist. Als Endprodukt werden den zu schützenden System und Daten jeweils auf Basis ihrer individuellen Schutzwürdigkeit Controls zugewiesen. Diese Controls sind allerdings lediglich der Prävention von Angriffen dienlich.

Weitaus interessanter wird es, in Bezug auf die Cyber Resilience, bei der Identifikation von kritischen Prozessen:

- Wie kann meine IT-Organisation (Fahrzeugflotte samt aller damit verbundenen Systeme), im Fall eines erfolgreichen Angriffs, wieder in den normalen Betrieb übergehen? → Wiederherstellung nach Angriffen (kritischer Prozess)
- Wie kann meine Organisation befähigt werden Angriffe zu erkennen? → Erkennung von Angriffen (kritischer Prozess)

Die Wiederherstellung der Systeme in tausenden Fahrzeugen stellt Automobilhersteller vor eine große Herausforderung. Rückrufaktionen im großen Stil kosten den Hersteller viel Geld. Im Zweifelsfall kann die ‚safety' der Verkehrsteilnehmer nicht gewährleistet werden, da Softwareaktualisierungen nicht zeitnah eingespielt werden können. Fahrzeughersteller müssen verstärkt auf Technologien zur „remote Aktualisierung" setzen. Solche technologischen Lösungen finden bereits Ihre Anwendung. Die Mobilfunkschnittstelle des automatischen Notrufsystems eCall (ab 2018 in der EU für alle Neufahrzeuge verpflichtend) ermöglicht solche Aktualisierungen ‚Over-the-Air' (kurz OTA). Die Möglichkeit, Software aus der Ferne in ein Fahrzeug einzuspielen, birgt aber gewisse Risiken in Bezug auf die Informationssicherheit. Konkrete Lösungsansätze für diese Probleme gibt es bereits (siehe bspw. ESCRYPT GmbH).

Die Erkennung von Angriffen stellt einen der schwierigsten Teile eines Cyber Resilience Konzepts dar. Die vollständige Erkennung aller Angriffe, egal ob erfolgreich oder nicht, ist in der Praxis, realistisch gesehen, unmöglich. Ebenso wird häufig die Intention hinter Cyber-Attacken nicht verstanden und somit eine falsche Schlussfolgerung, auf den Erfolg oder Misserfolg bzgl. des Angriffs, gezogen. Man kann an dieser Stelle also nur von Schätzungen ausgehen. Umso wichtiger ist es, innerhalb der Cyber Resilience Strategie konkrete Handlungsfelder aufzuzeigen.

Durch die gestiegene Vernetzung und die große Anzahl von zu überwachenden Ereignissen, in diversen Logquellen eines Infrastruktursystems (Fahrzeuge und die damit verbundenen Backendsysteme), machen eine manuelle Überwachung faktisch unmöglich.

An dieser Stelle kommt das Security Information and Event Management (kurz SIEM) zum Einsatz. SIEM ist ein Ansatz des Sicherheits-Managements, der darauf abzielt, eine ganzheitliche Sicht auf die Sicherheit der IT eines Unternehmens zu haben. Konkret bedeutet das: Durch die zentrale automatisierte Analyse aller in einem Infrastruktursystem anfallenden Daten (aus Logquellen), lassen sich Verhaltensweisen erkennen, die nicht dem normalen ‚soll Verhalten' der Systeme entsprechen. Logquellen können in Bezug auf vernetze Fahrzeuge bspw. verschiedene Steuergeräte, Infrastruktursysteme und Backendsysteme sowie Firewalls sein. Anomalien werden von einer Managment Console als potenzielle Sicherheitsvorfälle angezeigt. Die Logik der Erkennung von Anomalien im Verhalten der überwachten Systeme, liegt in dem Regelset anhand dessen ein SIEM-System ein abnormales Verhalten festmacht. Die Validität der erkannten potenziellen Sicherheitsvorfälle durch ein SIEM-System ist dabei abhängig von der Logik der Regel selbst, sowie den gegebenen Validierungsmöglichkeiten anhand verschiedener Logquellen.

Aus Sicht der Resilience Strategie entsteht genau hier ein Handlungsfeld. Derzeit betreiben Automobilhersteller häufig Ihre eigenen SIEM Systeme nicht herstellerübergreifend und nicht mit Infrastrukturkomponenten verbunden - erstes wohlmöglich aus Angst Sicherheitslücken oder sonstige geheime Informationen preiszugeben - letzteres aufgrund der momentan fehlenden Möglichkeit. Des Weiteren werden häufig verschiedene SIEM-Systeme von einem Hersteller verwendet. Ein SIEM-System zur Überwachung der Fahrzeugflotte und ein SIEM-System zur Überwachung der übrigen Infrastruktur (Backendsysteme, Clients, Mailserver etc.). Dies ist teilweise den verschiedenen Technologien der SIEM-Anbieter geschuldet. Verschiedene Anbieter unterstützen nur gewisse Software-Agenten (die bspw. nicht auf Steuergeräte in Fahrzeugen zugreifen können), dadurch ist der Einsatz mehrerer SIEM-Systeme erforderlich.

Der maximale Mehrwert einer SIEM-Lösung im Bereich der vernetzten Fahrzeuge entsteht aber erst wenn möglichst viele hersteller- und plattformübergreifende Logquellen an ein zentrales System angebunden werden. Eine Veranschaulichung warum plattformübergreifende Anbindung von Logquellen wichtig ist erfolgt im Abschnitt „Event-, Incident- und Problem Management". Ein Beispiel für die Steigerung des Mehrwerts durch die Anbindung möglichst vieler Logquellen erfolgt im Abschnitt „kontinuierliche Verbesserung".

Event-, Incident- und Problem Management

Das ITIL Event Management stellt sicher, dass Services und Konfigurationselemente überwacht werden. Es kategorisiert eventuell eintretende Events (z.B. Störungen) und leitet entsprechende Maßnahmen ein.[12] Als Subprozess des Eventmanagements verwaltet das Incident Management (Störungsmanagement) alle Incidents (Störungen). Ziel ist es, den Betrieb für den Anwender so schnell wie möglich wiederherzustellen. Sich auf Basis der gleichen Ursache häufende Incidents sind Problems (Probleme), diesen versucht das Problem Management vorzubeugen, indem es proaktiv Incidents thematisch kategorisiert und entsprechende Gegenmaßnahmen einleitet. Ebenso ist es Aufgabe des Problem Managements, die Anzahl an nicht vermeidbaren Incidents, zu minimieren.[13]

Auch aus dem Blickwinkel der Cyber Resilience geht es beim Incident Management ebenso um die schnelle Wiederherstellung der Services. Dadurch können etwaige Schäden vermieden oder zumindest minimiert werden. Eine zentrale Rolle spielen das Event- und Problem Management als übergeordnete Instanz. Durch verschiedene Events, Incidents und Problems können Rückschlüsse auf Sicherheitslücken getätigt werden. Deshalb ist eine enge Zusammenarbeit zwischen Event-, Incident- und Problem Management sehr wichtig. Nur durch eine enge Zusammenarbeit ist eine Erkennung von Einfallstoren möglich. Um eine gute Zusammenarbeit zu gewährleisten, sind eine abgestimmte Vorgehensweise, sowie ein Verständnis für die Ursache von Störungen nötig. Die Vorgehensweise und das Verständnis beinhalten Eskalationsmechanismen und Indikatoren für sicherheitsrelevante Störungen.

Als zentrales System, welches Indikatoren für sicherheitsrelevante Störungen sammelt, spielt das SIEM eine essenzielle Rolle im Rahmen des Event-, Incident- und Problem Managements. Unter den Gesichtspunkten der Cyber Resilience (Prävention, Erkennung, Wiederherstellung) kommt das System dabei im Wesentlichen bei der Erkennung von Angriffen zum Einsatz. Durch die Erkennung können weitere Schritte zur Wiederherstellung von Systemen eingeleitet werden. Ebenso ist es möglich, durch die erfolgreiche Identifikation von Angriffen weitere präventive Maßnahmen (Systemhärtungen) zu ergreifen.

Um den optimalen Nutzen eines SIEM-Systems zu gewährleisten, muss eine plattformübergreifende Anbindung der Logquellen erfolgen. Ein einfaches Beispiel soll dies verdeutlichen:

[12] Vgl. Buchsein et al. 2007 S.202ff

[13] Vgl. Buchsein et al. 2007 S.204ff,212ff

Ein SIEM-System erhält Ereignisse aus den Logquellen einer Ampel, eines Verkehrsleitrechners und eines Fahrzeugs. Die Ereignisse werden durch das SIEM-System anhand eines vordefinierten Regelsets geprüft. Der Verkehrsleitrechner meldet, dass die Ampel auf rot geschaltet ist. Dieses Ereignis wird durch die Logdaten der Ampel bestätigt. In sich sind diese zwei Ereignisse schlüssig. Das Fahrzeug allerdings, meldet die Ereignisse „Ampel ist grün“ und „Fahrzeug fährt an“. Das SIEM System erkennt einen Widerspruch in den bereitgestellten Informationen und generiert einen potenziellen Security Incident. Ähnliche Konstellationen sind mit Fahrzeugen fremder Automobilhersteller denkbar. Je mehr unabhängige Logquellen ein Ereignis bestätigt wird, desto größer ist die Wahrscheinlichkeit einer korrekten Erkennung. Ziel ist es, dass sich Fahrzeugflotten, andere Verkehrsteilnehmer (z.B. ausgestattet mit Datenbrillen), sowie intelligente Infrastruktursysteme gegenseitig überwachen.

Auch unter dem Aspekt ‚safety‘ können an dieser Stelle Servicestörungen vermieden werden. Durch die Echtzeitanalyse der Daten könnten unmittelbar weitere Aktionen eingeleitet werden. In unten genannten Fallbeispiel, könnte dem Auto die Weiterfahrt verweigert werden. Das vermeidet einen Unfall.

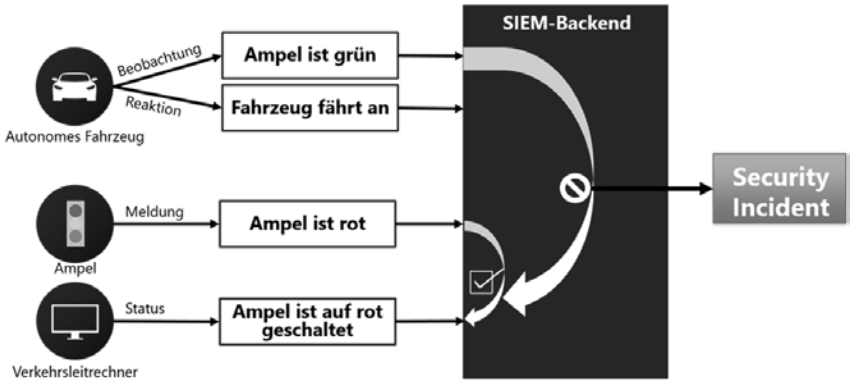

Bild 3: Fallbeispiel SIEM

Kontinuierliche Verbesserung

Ziel der kontinuierlichen Serviceverbesserung (Continual Service Improvement, CSI) nach ITIL ist es, das Wissen, welches aus Erfolgen und Misserfolgen generiert werden kann, ge-

zielt zur kontinuierlichen Verbesserung der Prozesse zu nutzen.[14] Nach ITIL kommen dabei u.a. Methoden der Norm ISO 20000, zur kontinuierlichen Verbesserung, zum Einsatz.

Im Zusammenhang der Cyber Resilience bedeutet CSI, dass die Informationssicherheit der betrachteten Services laufend verbessert und optimiert wird. Besonders wichtig im Kontext der Cyber Resilience, sind dabei die Prozesse zur Systemwiederherstellung, Prävention und der Erkennung von Angriffen.

Im Bereich der Prävention, sind die bereits im Abschnitt „Resilience Ansätze in der Informationssicherheit“ erläuterten Regeln (controls) kontinuierlich zu aktualisieren. Es geht darum die systemabhängigen Regeln an den aktuellen Stand der Technik anzupassen, um möglichst viele Angriffe zu abwehren zu können.

Auch im Bereich der Wiederherstellung gibt es Aktionspunkte im Bereich der kontinuierlichen Verbesserung. Desaster Recovery Pläne bzw. Rollout Pläne müssen stetig verbessert und angepasst werden. Denkbar sind auch Änderungen an Technologien, die zur Aktualisierung von Fahrzeugflotten genutzt werden.

Besonders wichtig, wie auch in den anderen ITIL-Prozessen, ist die Verbesserung der Erkennung von Angriffen. Ohne diesen Schritt können keine weiteren Erkenntnisse zur Prävention gewonnen werden. Bedrohungspotentiale können nicht ausreichend früh identifiziert werden und damit verbundene Risiken werden nicht unmittelbar reduziert.

Nachfolgend wird anhand von Abbildung 4 erläutert wieso der CSI-Prozess für ein SIEM-System essenziell ist. Die Erläuterung, basiert auf der Annahme, dass ein umfassender kollaborativer Ansatz eines SIEM-Systems aufgebaut wird. Das heißt, dass eine Hersteller- und Plattformübergreifende Anbindung von Logquellen stattfindet.

Zunächst wird auf einer begrenzten Datenbasis (ggf. zunächst nur die Fahrzeugflotte eines Herstellers und allen damit verbundenen Infrastruktursystemen), ein Regelset definiert. Durch dieses Regelset können Angriffe erkannt werden - ein Mehrwert entsteht. Ausgehend von diesem Mehrwert werden weitere Logquellen angeschlossen (bspw. Infrastruktursysteme und Fahrzeuge anderer Automobilhersteller). Dadurch steigt die Qualität der Datenbasis.

[14] Vgl. Olbrich 2008 S.156f

Auf der neuen Datenbasis können weitere Regelsätze definiert werden und die Validität der Regeln wird gesteigert. Als Folge, steigt wiederrum der Mehrwert des SIEM-Systems.
Ziel ist es, den Punkt zu erreichen, an dem die Anbindung an das SIEM-System auch für den einzelnen einen solchen Mehrwert bietet, dass sich weitere Teilnehmer aus eigener Intension anbinden.

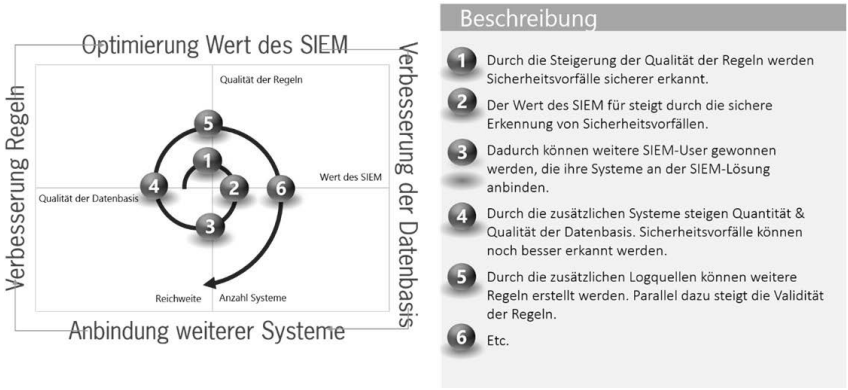

Bild 4: CSI SIEM

Fazit

Die Vernetzung des Automobils bietet Kunden sowie Herstellern erhebliche Mehrwerte und ermöglicht neue Geschäftsmodelle. Mit der Vernetzung des Fahrzeugs gehen jedoch auch entsprechende Risiken einher.

Die steigende Vernetzung von Fahrzeugen stellt alle Fahrzeughersteller vor die Herausforderung, zunehmend komplexe IT-Services für die Fahrzeuge zu entwickeln und sich mit den daraus resultierenden Schwierigkeiten auseinanderzusetzen. Diese entstehen durch die neue Verknüpfung zwischen herkömmlichen IT-Systemen und Fahrzeugsystemen. Dadurch könnten Schwächen in der Fahrzeuginfrastruktur und den damit verknüpften Systemen durch Angreifer ausgenutzt werden. Schwachstellen erhalten durch die Vernetzung eine größere Tragweite als bisher. Durch diese Entwicklung wirken sich die Schutzziele Vertraulichkeit, Verfügbarkeit und Integrität, direkt auf die Sicherheit (‚safety') der Verkehrsteilnehmer aus.

Die Sicherheit in vernetzten Fahrzeugen wird derzeit meist durch „bottom up" Verfahren aus den einzelnen Entwicklungsgruppen der Fahrzeugbauteile getrieben. Dadurch fehlt den Un-

ternehmen häufig die Fähigkeit Schwachstellen bauteilübergreifend zu identifizieren. Ebenso werden Absicherungen gegen Cyberangriffe überwiegend seitens der funktionalen Sicherheit realisiert. Dies ist jedoch nicht länger ausreichend. Das Resilience Konzept zeigt Punkte auf, an denen Sicherheitsmechanismen implementiert werden müssen, um eine breite Absicherung des Fahrzeugs zu ermöglichen. Sicherheit muss durch die Unternehmensführung kommuniziert und eingefordert werden. Ein Resilience Konzept ist ein kollaborativer Ansatz, der nicht nur die Aktion eines Fahrzeugherstellers fordert, sondern die Zusammenarbeit zwischen allen Beteiligten. Dafür müssen ein entsprechendes Bewusstsein, sowie Ressourcen und Prozesse geschaffen bereitgestellt und entwickelt werden.

(Cyber) Sicherheit muss ein integraler Bestandteil der IT-Services eines Fahrzeugherstellers werden. Nur wenn alle IT-Services des Herstellers nach den Prinzipien der Cyber Resilience (Prävention, Erkennung, Wiederherstellung) entwickelt und umgesetzt werden, ist die notwendige Widerstandsfähigkeit gegeben. Dies wurde anhand dem Beispiel der Security Information and Event Managements erläutert.

Ausblick:
Reguläre Schwachstellen in Applikationen bekommen eine größere Tragweite: Nicht nur IT-Systeme sondern auch die Fahrzeugfunktionen und damit Leib und Leben eines Nutzers können gefährdet werden. Dadurch ergeben sich verschiedenste Herausforderungen für Automobilhersteller, Zulieferer und Normungsgremien aus Sicht der IT-Sicherheit:

- Wie kann eine zentrale Datenbasis für ein „globales SIEM“ geschaffen werden (Plattform- und Herstellerübergreifend)?
- Wie können im Rahmen autonom fahrender Autos Infrastruktur- und Umgebungsdaten validiert werden?
- Wie kann die erforderliche vernetzte Infrastruktur (Ampeln, RSUs, etc.) abgesichert werden?
- Wie kann eine standardisierte Risikoevaluierung im und vor dem Entwicklungsprozess realisiert werden? (Im Hinblick auf die Implementierung neuer Funktionen)
- Wie kann eine Überwachung des Fahrzeugbordnetzes und dessen kooperierender Systeme realisiert werden, um etwaige Angriffe zu entdecken?
- Wie könnten Notfallpläne aussehen? (z.B. im Falle einer Kompromittierung einer kompletten Fahrzeugflotte)

- Wie können bauteilübergreifende Tests im Rahmen vernetzter Fahrzeuge standardisiert werden? (Penetrationstests des gesamten Fahrzeugs)
- Wie können Altsysteme abgesichert werden, wenn diese nachträglich vernetzt werden? (z.B. über OBD-Adapter oder Mobiltelefon)[15]

Die Bedeutung des Themas Connected Car wächst potentiell immer weiter, da das Marktpotential noch längst nicht ausgeschöpft ist.[16] Daraus resultiert auch die wachsende Bedeutung der IT-Sicherheit im Bereich vernetzter Fahrzeuge. Es ist nicht die Frage ob ein Automobilkonzern von einem Cyberangriff getroffen wird, sondern wann und wie stark, denn: „Sicher ist, dass nichts sicher ist. Selbst das nicht.“[17]

Das Literaturverzeichnis finden Sie in der digitalen Version dieses Dokuments (im Downloadbereich des VDI).

[15] Vgl. Krauß und Waidner 2015 S.387

[16] Vgl. Viereckl et al. 2014 S.3

[17] Joachim Ringelnatz, deutscher Schriftsteller

Programmiersprachen und Konzepte zur Entwicklung zuverlässiger und sicherer Automotive Software

Dipl.-Inf. **Oliver Schneider**, KIT, Karlsruhe

Kurzfassung

Ein Großteil der Securityprobleme bei Software entsteht nicht durch Logikfehler bei der Programmierung, sondern durch das versehentliche Verletzen von Sprachregeln. Die klassisch in der Automobilindustrie eingesetzte Programmiersprache C beispielsweise verbietet die Division durch Null. Wenn es einem Angreifer gelingt eine Division durch Null herbeizuführen, gibt es weder einen Programmabsturz noch eine kontrollierte Fehlerbehandlung. Stattdessen entsteht Undefined Behaviour (UB). UB bedeutet, dass das Programm nun nicht mehr dem programmierten Verhalten folgt, sondern zufälliges Verhalten zeigt. Dies erlaubt es in vielen Fällen dem Angreifer eingeschleusten Code auszuführen. Um UB entgegenzuwirken werden Standards wie MISRA-C eingesetzt, statische Analysen und Softwaretests durchgeführt. Des Weiteren wird durch Softwareentwicklungsprozesse dafür gesorgt, dass jede Zeile Code auch nachvollziehbar einem Eintrag in den Projektanforderungen zuordenbar ist. Diese „Add-ons“ sind jedoch nur Kompensationen, die die üblichen Programmierfehler verhindern, denn sie bieten keinerlei ganzheitliche oder beweisbare Sicherheit.

Es werden Verfahren und Programmiersprachen vorgestellt, die es erlauben hochperformante echtzeitfähige Software zu entwickeln und gleichzeitig die Fehlererkennung nicht erst im Test oder nach einem Angriff durchzuführen, sondern schon bevor die Software kompiliert wird.

1. Öffnung und Vernetzung von Systemen

Immer mehr Softwaresysteme die bisher durch eine physikalische Abschottung „gesichert“ waren, werden nach außen geöffnet. Die Energiesysteme der Zukunft enthalten Komponenten, die mit Apps auf Endnutzergeräten kommunizieren, um den Besitzern von Solaranlagen oder privaten Blockheizkraftwerken überall und live Informationen über ihre Anlagen geben. Fabriken werden im Rahmen von Industrie 4.0 stark intern, aber auch extern vernetzt. Dabei ist der Mehrwert durch schnellere Reaktionszeiten, bessere Planbarkeit, proaktive Wartung und vieles mehr gegeben. Im Automobilbereich aufkommende Systeme wie selbstfahrende Autos mit Car2Car Kommunikation, medialen vernetzten Systemen und mobile Wartungs- und Updatemöglichkeiten bieten viele Vorteile gegenüber den Varianten ohne Kommunikationsanbindung.

Diese Öffnung nach außen ist in den existierenden Softwaresystemen jedoch nicht vorgesehen. Die Software wird weiterhin entwickelt, als würde sie in einem abgeschlossenen System ausgeführt, und die Angriffs-„Sicherheit“ wird durch Add-ons wie Firewalls und VPNs erledigt. Es ist nicht nur unrealistisch zu erwarten, dass diese Systeme hiermit sicher sind, die Praxis zeigt, dass die beste Firewall wirkungslos ist [1][2], wenn es noch weitere Kommunikationskanäle gibt. Diese wird es in komplexen (und damit allen realen) Systemen immer geben. Analog zur Gebäudesicherheit, reicht es nicht nur einen Zaun um das System zu ziehen, sondern ist es zusätzlich notwendig jeden Raum (Softwarekomponente) und jeden Schrank (Funktion einen Softwarekomponente) einzeln abzusichern. [4]

2. Lücken der Programmiersprache C

Die Programmiersprache C (zurzeit meist verwendet in der Version C99) ermöglicht es Entwicklern sehr leicht sicherheitskritische Fehler in Programmcode zu schreiben. Ein Beispiel ist die fehlende Typsicherheit, die dazu führen kann, dass die Aussage „1 < -1“ wahr ist:

```
unsigned i = 1;
if (i < -1) {
    printf(„Klassische Mathematik gilt nicht in C“);
}
```

Hierbei wird automatisch die „-1“ in eine vorzeichenlose Ganzzahl umgewandelt, was auf den meisten Systeme `4294967295` ist. Da die Aussage `1 < 4294967295` logischerweise wahr ist, wird der Körper der If-Bedingung ausgeführt.

Die üblichste Quelle von Sicherheitslücken in Software sind Zeiger. Ein Zeiger in C ist eine Ganzzahl, die die Adresse des Speichers eines anderen Wertes enthält. Es ist ersichtlich, dass ein Angreifer durch das Manipulieren von Adressen auf beliebigen Speicher zugreifen und damit beliebige (und auch geheime) Daten auslesen oder modifizieren kann.

Ein prominentes Beispiel ist der Heartbleed Bug (http://heartbleed.com/), der es erlaubte, den kompletten Speicher eines Webservers auszulesen, selbst den Teil des Speichers, in dem Administratorpasswörter gespeichert sind. Dies führte dazu, dass viele Firmen (selbst Google) die Software ihrer Server aktualisieren mussten und alle ihre Nutzer baten ihre Passwörter zu ändern. Der Heartbleed Bug trat in der OpenSSL Bibliothek auf, die eigentlich für Verschlüsselung zuständig ist, aber in diesem Fall nicht nur die Verschlüsselung zwecklos machte, sondern zudem noch viel mehr Daten preisgab als die Verschlüsselung je schützte.

Zur Veranschaulichung des Zeigerfehlers „Buffer Overflow“ sei folgender C-Code gegeben, in dem ein Array (`arr`) von 7 Elementen erzeugt wird:

```
unsigned arr[7] = { 42, 99, 33, 15, 8, 17, 1 };
```

Auf Elemente des Arrays kann mit der Index Operation zugegriffen werden: `arr[2]` greift hierbei auf das Element mit dem Wert `33` zu. `arr[6]` greift auf das letzte Element zu. Die Sprache C erlaubt es nun auch auf `arr[7]` oder `arr[132]` zuzugreifen, obwohl diese nicht existieren. Während in diesen Fällen eine eindeutige Verletzung der Indexgrenzen stattfindet, ist dies bei einem Zugriff über eine Variable nicht klar, bevor das Programm ausgeführt wird. `arr[n]` ist erlaubt, solange `n >= 0` und `n < 7`. Wenn der Wert von `n` von einer Benutzereingabe abhängt (wie es im Heartbleed Bug der Fall war), so kann ein Angreifer Werte angeben, die dazu führen, dass die Indexoperation nicht mehr im Speicherbereich des Arrays liegt.

3. MISRA-C

Das Ziel des MISRA-C Standards [3] ist explizit nicht, dass C für die Automobilindustrie genutzt werden sollte, sondern dass trotz der Verwendung von C keine fatalen Softwareprobleme entstehen. Die Nutzung des Standards kann viele Probleme, die bei der Verwendung von C aufkommen, kompensieren oder vermeiden. Hierbei wird allerdings die Programmiersprache so stark eingeschränkt, dass die Entwicklung merklich darunter leidet. Zwar ist in der Mehrheit der Fälle der Code durch die Einhaltung einer Regel lesbarer und weniger fehleranfällig, allerdings benötigen die restlichen Fälle umständliche Umschreibungen, die die Bedeutung des Programmcodes verschleiern und damit selbst zu neuen Fehlerquellen werden.

Daher schreibt der Standard vor, nicht alle Regeln blind anzuwenden, was wiederum dazu führt, dass der Entwickler eine Entscheidung treffen muss, zwischen dem Erfüllen der Regel oder dem Ignorieren derselben auf Basis subjektiver Informationen.

Der Autor sieht zwar den Bildungseffekt des Standards für Entwickler, die noch nicht mit den obskuren Regeln der Programmiersprache C vertraut sind, jedoch bleibt also die Frage offen, weshalb C eingesetzt wird, wenn offensichtlich jeder Entwickler sich mit vielerlei Regeln zur fehlerfreien Entwicklung beschäftigen muss, statt sich um das ursprüngliche Problem der Entwicklung neuer Features zu kümmern.

4. Programmiersprachen für sichere Software

Die Programmiersprache Ada, entworfen 1983 und mittlerweile in ihrer 3. Version mit Ada 2012, wurde für sicherheitskritische Anwendungen im militärischen Bereich und der Luft- und Raumfahrt entwickelt. Schon ihre erste Version eliminiert die Notwendigkeit des MISRA Standards fast vollständig, da die Sprache kein Undefined Behaviour enthält und alle Sprachkonstrukte möglichst redundant aufgebaut sind, um versehentliche Fehler zu vermeiden.

Zur Demonstration wird auf das Beispiel des Arrays zurückgegriffen. Im Gegensatz zu Arrays in C, sind Arrays in Ada semantisch an ihre Länge gebunden. Das bedeutet, dass beim Zugriff

auf ein Element eines Arrays eine automatische Prüfung eingefügt wird, die sicherstellt, dass der Zugriff sich auch innerhalb des Arrays befindet. Wenn die Prüfung fehlschlägt, wird eine Ausnahmebehandlung gestartet.

```
declare
  arr: array(1..7) of Integer;
begin
  arr(n) := 19; -- löst CONSTRAINT_ERROR aus falls n > 7
  -- Wenn der `exception` block fehlt wird der Fehler
  -- automatisch an die aufrufende Funktion weitergereicht
exception
  when CONSTRAINT_ERROR => -- Fehlerfall hier behandeln
end
```

Der Entwickler kann entweder eine eigen Ausnahmebehandlung bereitstellen, oder einen kontrollierten Systemabsturz in Kauf nehmen. Es ist nicht möglich, die Prüfung zu vergessen, falsch zu implementieren oder abzuschalten.

Unabhängig voneinander haben die drei Firmen Apple, Google und Mozilla um ca. 2010 erkannt, dass trotz verschiedenster Fehlervermeidungsstrategien die Rate sicherheitskritischer Fehler ihrer Software steigt. Um hier entgegenzuwirken wurden die Programmiersprachen Swift (App Entwicklung), Go (Web-backend und Softwarewerkzeugentwicklung) und Rust (Systementwicklung) entwickelt, die seit einigen Jahren nun auch praktisch eingesetzt werden. Es wird hier jedoch nur Rust betrachtet, da die anderen beiden Sprachen nicht echtzeitfähig sind.

Rust nutzt die Kombination aus Ada's starker Typsicherheit zusammen mit der mathematischen Reinheit der funktionalen Programmiersprache Haskell. Dies erlaubt es zum Beispiel die Prüfung von Arraygrenzen von Ada noch weiter zu verbessern, indem statt des automatischen Einfügens von Prüfungen, der Entwickler gezwungen wird selbst die Grenzen zu prüfen:

```
let arr = [ 42, 99, 33, 15, 8, 17, 1 ];
// Fehler in der folgenden Zeile, `arr.get()` ist keine Zahl
let elem_plus_one = arr.get(3) + 1;
// Prüfung ob Element an Stelle 3 existiert
if let Some(&elem) = arr.get(3) {
        // Arbeiten mit Element
    let elem_plus_one = elem + 1;
}
// kein Zugriff auf Wert von `elem` außerhalb des `if let`
```

Während es sich intuitiv effektiver anhört, wenn der Compiler wie bei Ada automatisch Überprüfungen generiert, muss bedacht werden, dass versteckte Überprüfungen und damit auch Ausnahmebehandlungen übersehen werden können. Nicht vorgesehene Ausnahmebehand-

lungen führen im schlimmsten Fall zu einem kontrollierten Programmabsturz, was in sicherheitskritischen Systemen zu Notabschaltungen und Ausfällen führen kann.
Zudem unterstützt Rust die Möglichkeit der inkrementellen Integration in existierende C-Projekte. Dies erlaubt es Teile des Projektes in Rust neu zu schreiben und diversitär oder als Ersatz des C-Codes in das Gesamtprojekt zu integrieren. Damit kann die Sicherheit schrittweise erhöht werden, indem neue Softwaremodule in Rust geschrieben und existierende Module einzeln portiert werden.

5. Fazit

Standards wie MISRA-C sind sehr lehrreich für unerfahrene Entwickler und absolut notwendig, um in bereits vorhandenen C-Projekten überhaupt ein Mindestmaß an Sicherheit gewährleisten zu können. Ein Flickenteppich von Regeln ist jedoch kein Ersatz für eine Programmiersprache, deren Entwurf auf den aktuellen Kenntnissen der Softwareentwicklung basiert und mit dem Ziel entwickelt wurde, ganze Klassen sicherheitskritischer Fehler vollständig zu eliminieren. Daher sollte vor dem Beginn neuer Projekte die zu nutzende Programmiersprache evaluiert werden, da durch den ganzheitlichen Ansatz der sicheren Sprachen letztlich auch enorme Kosten eingespart werden können, die durch zu spät entdeckte Fehler entstehen. Existierende Projekte können inkrementell in neue Sprachen übersetzt werden und dabei kann mittels Softwarediversität das Vertrauen in die Sicherheitskonzepte der Sprachen gewonnen werden, bevor vollständig auf die neue Sprache umgestellt wird.

[1] GUO, Qinglai, et al. Comprehensive security assessment for a cyber physical energy system: a lesson from Ukraine's blackout. *Automation of Electric Power Systems*, 2016, 40. Jg., Nr. 5, S. 145-147.

[2] BSI - Die Lage der IT-Sicherheit in Deutschland 2014

[3] Motor Industry Software Reliability Association. (2013). *MISRA C 2012: Guidelines for the Use of the C Language in Critical Systems: March 2013*. Motor Industry Research Association.

[4] Keller, H. B., Schneider, O., Matthes, J., & Hagenmeyer, V. (2016). Reliable, safe and secure software of connected future control systems-challenges and solutions. *AT-AUTOMATISIERUNGSTECHNIK*, *64*(12), 930-947.