Grundseminar Winter 2018/19
Master Informatik

Sichere Kommunikationsnetze im Auto

Wilhelm Schumacher

Eingereicht am 17.02.2018

Zusammenfassung Die Integration des Autos in das Internet der Dinge ermöglicht die Umsetzung einer Vielzahl neuer Funktionen. Zum Beispiel können „Software Updates over the Air“ für bestimmte Komponenten des Autos durchgeführt werden. Auch könnten Ad-Hoc-Fahrzeug-Netzwerke aufgebaut werden, in denen ein Auto anderen Autos in der Nähe Informationen zur Verfügung stellt (z.B. aktuelle Geschwindigkeit/Position oder potentielle Gefahren auf der Straße) oder die Planung von Routen im Bereich der E-Mobilität könnte so realisiert werden. Diese Funktionen erfordern die Öffnung des internen Kommunikationsnetzes des Autos, damit Kommunikation mit anderen Autos (V2V), mit Infrastruktur an der Straße (V2I) und mit dem Backend des Herstellers (V2C) stattfinden kann. Für diese neue Art der Kommunikation mit der Außenwelt ist die aktuelle Architektur des Autos nicht mehr ausgelegt. Deswegen werden auf das Auto angepasste Sicherheitssysteme zum Schutz des Netzwerks (Security) benötigt. Teilbausteine von diesem neuen Sicherheitssystem könnten **Software Defined Networking (SDN)** und **Network Intrusion Detection Systeme (IDS)** sein. Diese Ausarbeitung gibt einen kurzen Überblick über die bestehende Netzwerkarchitektur des Autos sowie Angriffsmöglichkeiten und eine Einführung zu Network Intrusion Detection als auch Software Defined Networking im automotiven Kontext.

Schlüsselwörter Automotive Security · Kommunikationsnetze im Auto · Network Intrusion Detection · Anomaly Detection · Software Defined Networking

Inhaltsverzeichnis

Wilhelm Schumacher
E-Mail: Wilhelm.Schumacher@haw-hamburg.com
Department Informatik
Fakultät Technik und Informatik
Hochschule für Angewandte Wissenschaften Hamburg

1 Einleitung

Heutzutage besteht ein Fahrzeug der oberen Mittelklasse bereits aus einer Vielzahl verteilter Steuereinheiten (ECUs), die über interne Bussysteme (z.B. CAN) miteinander kommunizieren. Insgesamt umfasst das komplette Software-System mehrere Millionen Zeilen von Programmcode [vgl. BCK+12, Seite 1]. Laut [BCK+12, Seite 1] basieren 80% der Innovationen in Fahrzeugen auf der Einbindung von Computersystemen. Gleichzeitig zu den neuen Funktionen im Auto wächst aber auch die Anzahl an Schnittstellen zur Außenwelt inklusive Internet. Die Schnittstellen bieten das Potential für eine Vielzahl neuer Dienste rund um das Auto. Zum Beispiel Vehicle-to-Vehicle (V2V) Kommunikation, die es ermöglicht gefährliche Situationen frühzeitig zu erkennen oder Vehicle-to-Infrastructure (V2I) Kommunikation, die unter anderem Möglichkeiten zur besseren Steuerung des Verkehrsflusses bereitstellt. Auch Vehicle-to-Cloud (V2C) Kommunikation soll in Zukunft eine noch größere Rolle im Auto spielen und das Auto mit Daten versorgen sowie Daten aus dem Auto verarbeiten. Es könnten zum Beispiel mit Hilfe von Updates das Kartenmaterial des autonomen Fahrzeuges aktualisiert werden oder eine Ladestation reserviert werden, so dass diese auch frei ist, sobald man am Ziel ankommt. Sogar der Kalender des Fahrers könnte in die Planung der Ladezeiten des Autos miteinbezogen werden.

Durch die Integration des Internets wird das Auto zu einem attraktiven Angriffsziel für Hacker. Die zusätzlichen Schnittstellen, die durch die Vernetzung des Autos entstanden sind, könnten für potentielle Angriffe missbraucht werden. Dabei können Angriffe gravierende Auswirkungen auf den Menschen und die Umwelt haben, denn das Auto ist ein sicherheitskritisches System. Zum Beispiel demonstriert das Paper [siehe MV15] einen Angriff von einem Fahrzeug über Schnittstellen zum Internet. In dem Beispiel haben Forscher es geschafft, einen Jeep Cherokee (Baujahr 2014) zu hacken. Dafür wurde eine Sicherheitsschwachstelle im Uconnect-System des Herstellers Chrysler ausgenutzt. Anschließend konnte über die Multimedia-Schnittstelle auf das interne Kommunikationsnetz des Autos zugegriffen werden. Dadurch war es dann möglich den Jeep fernzusteuern und Funktionen wie das Abschalten des Motors von der Ferne auszulösen. In dem Paper [siehe CMK+11] wurde demonstriert, wie ein in großen Massen produzierter Sedan über verschieden Schnittstellen nach Außen komplett übernommen werden konnte. Zum Beispiel konnten Angriffe über ein „PassThru“ Gerät (verbindet sich mit dem Diagnoseport des Autos) oder über die Update Funktionalität des CD-Players ausgeführt werden. In [siehe RMM+10] wurde bei der Analyse einer Tire Pressure Monitoring System (TPMS) entdeckt, dass die ECU, die für Verarbeitung der Signale zuständig war, zum Absturz gebracht werden konnte. Auch über das Digital Radio (DAB) können Angriffe ausgeführt werden, weil DAB erlaubt auch Daten wie Bilder oder Texte zu übertragen [siehe Dav15]. Diese Beispiele zeigen, dass neue Sicherheitskonzepte notwendig sind um das interne Kommunikationsnetz besser vor Angriffen zu schützen bzw. das interne Kommunikationsnetzwerk selbst bessere Sicherheitsstrukturen benötigt.

Dazu werden Security-Mechanismen auf mehreren Ebenen benötigt. Zum Beispiel Systeme, die direkt am Eintrittspunkt vor Angriffen schützen, wie Firewall Systeme oder Proxy-Server und auch weitere Komponenten wie Verschlüsselung oder Sicherheitsupdates sind im internen Netzwerk notwendig. Zwei weitere mögliche Bausteine könnten dabei **Network Intrusion Detection** und **Software Defined Networking (SDN)** sein. Diese Systeme würden dann ergänzend zu den anderen Sicherheitsmechanismen eingesetzt werden, um vor allem Angreifer auch noch entdecken zu können, die bereits die anderen Sicherheitsmecha-

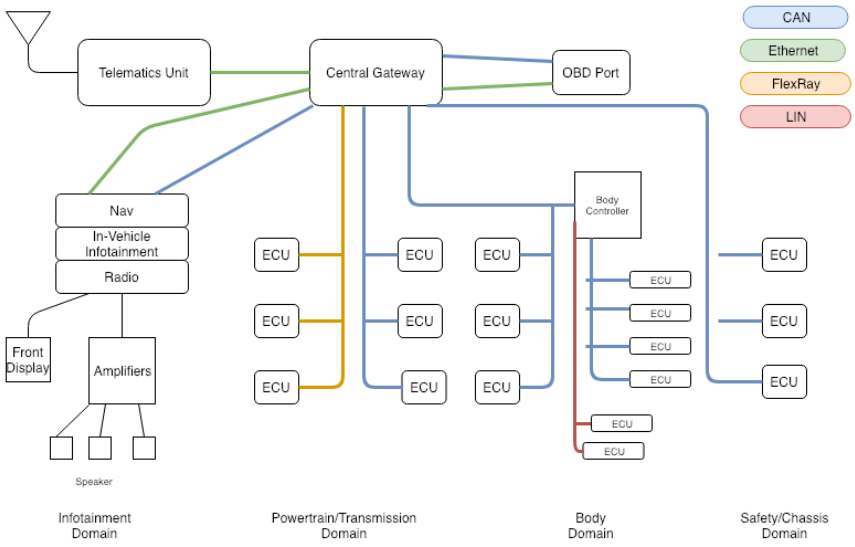
nismen überwunden haben und Teile des Netzes kontrollieren. Network Intrusion Detection [siehe BBK14] versucht Angriffe anhand bestimmter Muster zu erkennen. Dabei können Angriffe auch noch erkannt werden, nachdem die Firewall überwunden wurde und der Angreifer zum Beispiel bereits eine ECU kontrolliert. Der Ansatz Software Defined Networking [siehe KRV+15] integriert einen zentralen Controller in das Netzwerk, der die Kommunikation der einzelnen ECUs im Netzwerk steuern kann. Der Controller kann nun entscheiden welche Komponenten miteinander kommunizieren dürfen. So können gerade kleine und leistungsschwache ECUs besser geschützt werden. Sowohl IDS als auch SDN werden bereits über längere Zeit im Netzwerk Security Bereich eingesetzt. Diese Ausarbeitung hat als Ziel einen Überblick über die Netzwerkarchitektur des Autos und Angriffsmöglichkeiten zu bieten sowie eine kurze Einführung in die Themen SDN und IDS im Kontext des Autos zu geben. Die Arbeit findet innerhalb des SecVI Forschungsprojekt [siehe Sec] statt. Das SecVI Projekt wird in Verbindung mit Industriepartnern durchgeführt und vom Federal Ministry of Education und Research gefördert und hat als Ziel eine sichere Netzwerkarchitektur für das Auto zu entwickeln.

Das Kapitel **Aufbau des internen Kommunikationsnetzes** beschreibt den typischen Aufbau eines domänenbasierten Kommunikationsnetzes aus einem aktuellen Oberklasse Auto. Dabei hebt es auch die wichtigsten Komponenten innerhalb dieses Netzes einmal kurz hervor. Anschließend stellt das Kapitel **Angriffsmöglichkeiten** die Schnittstellen des Autos zur Außenwelt dar und kategorisiert die vorhanden Schwachstellen in der Architektur. Das Kapitel **Network Intrusion Detection** erläutert dann die Funktionswiese von Network Intrusion Detection genauer. Danach folgt im Kapitel **Software Defined Networking** noch ein Einblick in das Thema Software Defined Networking im Kontext des Fahrzeuges, bevor das letzte Kapitel **Zusammenfassung und Ausblick** die Arbeit noch einmal zusammenfasst. Weiterhin wirft dieses Kapitel auch ein Blick in die Zukunft hinsichtlich des weiteren Verlauf meines Masters.

2 Aufbau des internen Kommunikationsnetzes

Eine wichtige Komponente im internen Kommunikationsnetzwerk der aktuellen Oberklasse Autos sind die **Electrical Control Units (ECU)** [vgl. SNA+13, Seite 2-3]. Sie sind verantwortlich für die Steuerung bestimmter Funktionalitäten, wie zum Beispiel für die Lenkung, die Bremse oder auch für die Komfortfunktionen des Sitzes. Je nach Aufgabe der ECU, verfügt sie auch über sehr unterschiedliche Mengen an Rechenleistung. Typischerweise enthält ein modernes Auto über 70 verschiedene ECUs. Weiterhin befindet sich an einer zentralen Stelle im Netzwerk noch ein Central Gateway. Das Central Gateway bildet einen Eintrittspunkt in das Netz. Denn es ist sowohl mit der Telemetrie-Komponente als auch auch mit dem Diagnose Port OBD verbunden.

Sowohl das Central Gateway als auch viele weitere ECUs werden in Abbildung 1 dargestellt. Bei dieser Abbildung des internen Kommunikationsnetzes des Autos handelt es sich um eine domänenbasierte Architektur. Denn die ECUs werden anhand ihrer Funktionen in Domänen gruppiert und organisiert. Es gibt zum Beispiel Domänen für Infotainment, Powertrain oder Chassis (siehe Abbildung 1). Jede dieser Domänen befindet sich innerhalb eines eigenes Subnetzwerkes [vgl. HAF+09]. Da die einzelnen Steuereinheiten auch miteinander kommunizieren müssen und eine direkte Verbindung aller ECUs nicht handhabbar wäre, werden Bussysteme mit mehreren Teilnehmern im internen Netzwerk eingesetzt. Das



Quelle: basierend auf https://www.iot-now.com/2017/02/27/59018-securing-automotive-air-updates

Abb. 1 Die Abbildung stellt eine typische domänenbasierte Netzwerkarchitektur mit Central Gateway von einem aktuellen Oberklasse Auto dar.

am meisten verwendete Bussystem ist der **CAN-Bus (Controller Area Network)** [siehe Bos]. Das Bussystem erlaubt Datenraten von bis zu 1 Mbit/s und jeder Teilnehmer sendet seine Nachrichten im Broadcast an alle anderen Teilnehmer des Bussystems. Zur Kollisionsvermeidung wird das Carrier Sense Mulitiple Access / Collision Detection (CSMA/CD) Verfahren sowie ein Message Identifier (für Prioritäten) eingesetzt. Weitere Bussysteme, die im Auto zum Einsatz kommen, sind **LIN (Local Interconnect Network)** [siehe AUT], **MOST (Media Oriented Systems Transport)** [siehe COO] und **FlexRay** [siehe Fle]. Das LIN-Protokoll ist eine sehr kostengünstige Variante mit niedrigen Datendurchsatz. Dagegen ist das MOST-Protokoll eher auf die Übertragung von Multimediadaten mit hohen Datendurchsatz spezialisiert. FlexRay ist ein weiteres Bussystem mit höheren Datendurchsatz, der sich vor allem wegen höherer Produktionskosten und dem äußerst komplexer Aufbau nicht durchsetzen konnte.

Durch die Verwendung von Bussystemen besteht das interne Kommunikationsnetz der aktuellen Generation folglich aus mehreren Subnetzwerken (Abbildung 1). Dabei enthält jedes Subnetzwerk bestimmte Steuereinheiten (ECUs). Diese ECUs müssen auch über das eigene Bussystem hinaus mit Steuereinheiten auf einem anderen Bussystemen kommunizieren. Dazu werden Gateway ECUs verwendet. Gateway ECUs ermöglichen die Kommunikation zwischen verschiedenen Netzen im Fahrzeug. Dabei können in dieser Architektur ECUs, die für Entertainment- oder Komfortfunktionen zuständig sind, auch über das zentrale Gateway auf sicherheitsrelevante Bussysteme (z.B. Chassis Domäne) zugreifen. Zum Teil können sich auch Steuereinheiten aus unterschiedlichen Domänen in einem Subnetzwerk befinden. Dann würden sicherheitskritische Nachrichten über denselben Bus wie Komfort-

funktionsdaten oder Entertainmentdaten versendet werden.

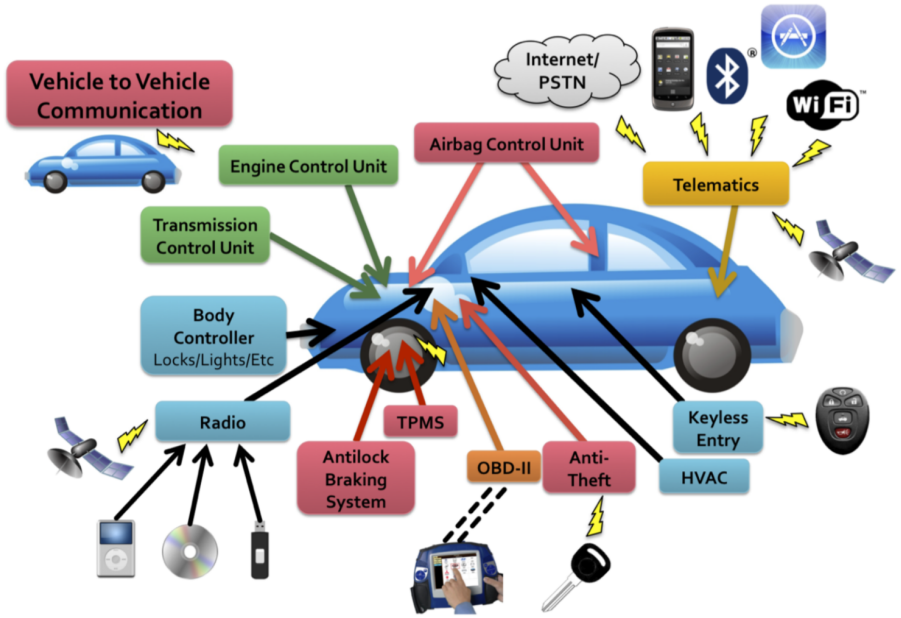
In der Zukunft spielt **Automotive Ethernet** in der Netzwerk Architektur des Autos eine immer größere Rolle [vgl. Lev18]. Denn Ethernet kann Anforderungen an wachsende Bandbreite, Flexibilität und Kosteneffizienz besser gerecht werden. In der abgebildeten Architektur wird Ethernet hauptsächlich für Infotainment, die Verbindung zum OBD Port und für die Verbindung zur Telematics Unit verwendet. Dieser Einsatzbereich stellt nur die Anfänge von Ethernet in der Netzwerkarchitektur des Fahrzeuges dar. Der erweiterte Einsatz von Ethernet würde die Architektur des Autos in Teilen verändern [siehe Rö18]. Dadurch würde viele sehr unterschiedliche Domänen IP-basiert über ein Ethernet Backbone abgewickelt werden. Die Integration von Ethernet in das Auto würde also auch zu geänderten Rahmenbedingungen für die Sicherheitsmechanismen des Autos führen. Zum Beispiel ermöglichen die größeren Frames von Ethernet den Einsatz einer großen Bandbreite von bereits etablierten Sicherheitsfunktionalitäten, die mit einem 8-Bit großen CAN-Bus Frame nicht eingesetzt werden konnten.

3 Angriffsmöglichkeiten

Innerhalb der letzten Jahre hat sich das interne Netzwerk des Autos von einem komplett abgeschotteten Netzwerk hin zu einem aktiven Teilnehmer im Internet der Dinge entwickelt. Durch neue Schnittstellen und Technologien können immer mehr Funktionen umgesetzt werden, bei denen Kommunikation mit anderen Teilnehmern benötigt wird. Zum Beispiel V2V- (Vehicle to Vehicle), V2I- (Vehicle to Infrastructure) oder V2C- (Vehicle to Cloud) Kommunikation. Dabei konsumiert das Auto Dienste von außen wie zum Beispiel Updates für eine bestimmte Komponente vom Backend des Herstellers. Gleichzeitig bietet das Auto aber auch Dienste nach außen hin an. Zum Beispiel können Informationen über den Standort des Autos zur Verkehrsflusssteuerung verwendet werden. Diese neue Denkweise in Diensten benötigt aber auch eigene und neue Sicherheitsmechanismen. Denn das bestehende interne Netzwerk bietet zu wenig Schutz gegenüber möglichen Angreifern. Angreifer könnten Ziele wie Sabotage des Autos, Diebstahl, Erpressung, elektronisches Tuning oder das Sammeln privater Daten verfolgen.

3.1 Schnittstellen zur Außenwelt

Abbildung 2 gibt einen Überblick über die IO-Channel, die in einem modernen Auto vorliegen [vgl. CMK$^+$11, Seite 2-4]. Generell können die IO-Channel anhand ihrer Reichweite in die drei Kategorien **Indirect Physical Access**, **Short-range Wireless Access**, und **Long-range Wireless Access** unterteilt werden und bilden die Angriffsoberfläche (Surface) des Autos. Zu den Indirect Physical Access Schnittstellen gehören der Diagnoseport OBD-II, USB-Schnittstellen, Dockingstations, CD-Player oder möglicherweise sogar ein direkter Ethernet-Anschluss. Diese Schnittstellen haben alle gemeinsam, dass direkter Zugriff auf das Fahrzeug benötigt wird, um einen Angriff ausführen zu können. Trotzdem können die Schnittstellen zum Teil auch indirekt aus der Ferne von Angreifern verwendet werden. Zum Beispiel wenn das Gerät, das auf dem OBD-II Port zugreift, vom Angreifer kontrolliert wird. Die Short-range Wireless Schnittstellen bieten Zugriff innerhalb einer Reichweite von 5-300 Meter an. Dazu zählen zum Beispiel Bluetooth, Remote Keyless Entry, Tire Pressure



Quelle: http://www.autosec.org/pubs/cars-usenixsec2011.pdf

Abb. 2 Die Abbildung zeigt Schnittstellen eines modernen Automobils mit der Außenwelt (sowohl Input Channel als auch Output Channel).

Monitoring System (TPMS), V2V Kommunikation und Internet Hotspots. Als Letztes gibt es noch die Long-range Wireless Schnittstellen, die Kommunikation über große Reichweiten ermöglichen, wie zum Beispiel Radio-Kanäle oder Mobile Daten, die sogar individuelle Adressierung erlauben.

3.2 Potentielle Schwachstellen

Informationssicherheit (Security) bedeutet für ein System, dass keine unautorisierte Veränderung und Gewinnung von Informationen zugelassen wird. Laut [siehe Eck14] müssen drei Basisanforderungen für die Informationssicherheit gewährleistet sein. Die erste Basisanforderung ist die **Informationsvertraulichkeit**. Informationsvertraulichkeit sagt aus, dass eine Information eines Systems geheim gehalten wird und nur mit entsprechender Autorisierung lesbar ist. Die aktuelle domänenbasierte Architektur des Autonetzwerkes offenbart in diesem Bereich einige Sicherheitsschwachstellen. Zum Beispiel in den Netzwerkprotokollen wie dem CAN-Protokoll. CAN Nachrichten können von jedem Teilnehmer auf dem Bussystem im Klartext gelesen werden. Die zweite Anforderung **Datenintegrität** bedeutet, dass unautorisierte Modifikationen an Datenobjekten verhindert werden müssen. Dazu werden zum Beispiel Vertrauenskette (Chain of Trust) zur Authentifizierung im internen Netzwerk benötigt. Die dritte Basisanforderung **Systemverfügbarkeit** besagt, dass Angriffe die Leistungsfähigkeit des Systems nicht beeinflussen dürfen. Auch für die Systemverfügbarkeit kann das CAN-Protokoll wieder als Beispiel dienen, denn eine korrupte ECU könnte ein Bussystem mit Nachrichten von hoher Priorität flooden und damit die Übertragung anderer Nachrichten verhindern. Weitere Sicherheitsschwachstelle stellen die Gateway ECUs

dar. Dadurch ist es ECUs aus einem Subnetz wie zum Beispiel Infotainment möglich mit Steuereinheiten aus anderen Domänen zu kommunizieren. Auch in den Schnittstellen nach außen sind einige Sicherheitslücken vorzufinden [siehe CMK+11]. Zum Beispiel kann ein korruptes WMA-File benutzt werden, um Bus Nachrichten durch den CD-Player zu senden.

4 Network Intrusion Detection

Eine Möglichkeit, um das Netz des Autos besser vor Angriffen zu schützen, ist Network Intrusion Detection [vgl. BBK14]. IDS-Systeme werden schon lange im Bereich der Netzwerksicherheit eingesetzt. Sie stellen nur einen Baustein innerhalb eines kompletten Sicherheitsmechanismus dar und können nicht alleine das ganze System sicher machen. Generell sind Network Intrusion Detection (IDS) Systeme in der Lage Angriffe auf ein Netzwerk anhand von bestimmten Mustern im Netzwerk-Traffic oder System zu erkennen. Dazu wird vordefiniertes Normalverhalten des Systems mit dem tatsächlichen Verhalten verglichen. Wenn dann ein untypisches Verhalten, das vom Normalzustand abweicht, festgestellt wird, löst das System einen Alarm aus. Zum Beispiel, wenn eine ECU plötzlich in sehr kurzen Zyklen Nachrichten verschickt. Anschließend wird ein Administrator/Anwender des Systems informiert, der geeignete Gegenmaßnahmen einleiten kann. In vielen Bereichen werden bereits umfangreiche Anwendungen für Network Intrusion Detection eingesetzt. Zum Beispiel bei der Betrugserkennung für Kreditkarten, im Cyber Security Bereich oder im Militär zum Überwachen des Feindes [vgl. BBK14].

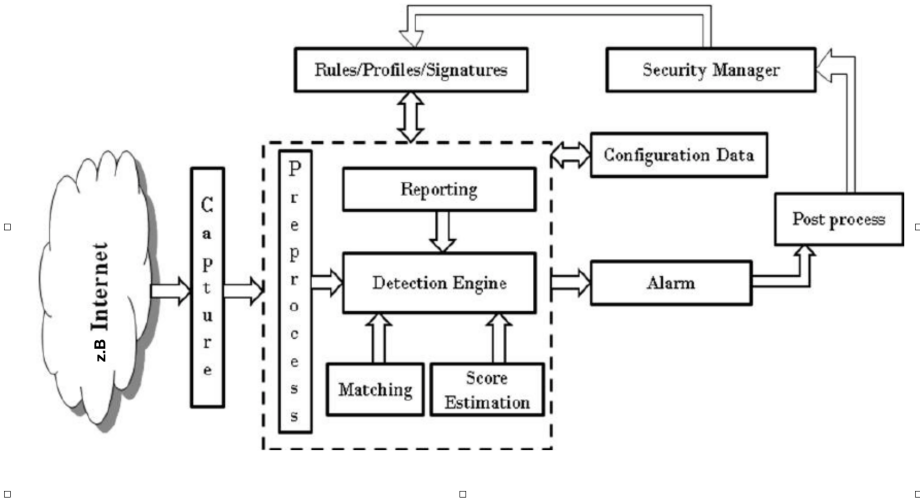
Häufig wird ein IDS (Intrusion Detection System) ergänzend zu einer Firewall eingesetzt. Denn ein großer Vorteil vom IDS ist, das Angriffe auch noch nach Überwinden der Firewall, die sich an den äußeren Schnittstellen befindet, entdeckt werden können. Im Netzwerk des Autos könnte zum Beispiel ein Eindringling, nachdem er bereits eine ECU übernommen hat, noch entdeckt werden. Wenn die ECU nun untypische Nachrichten an andere Kommunikationspartner, in höherer Frequenz oder mit falschem Aufbau verschicken würde, könnte dies vom IDS erkannt werden. Weiterhin wird bei der Position des IDS zwischen zwei verschiedenen Arten unterschieden. Es gibt einerseits das Host-basierte Intrusion Detection System als auch das Netzwerk-basierte Intrusion Detection System. Bei einem Netzwerk-basierten IDS werden Unregelmäßigkeiten vor allem in den Paketen aus dem Netzwerk gesucht. In der klassischen IT-Security befindet sich diese Art des IDS typischerweise an der Schnittstelle zum Internet um allen eingehenden Traffic untersuchen zu können. Dabei wird häufig auf Grund hoher Brandbreiten viel Performance benötigt. Im Gegensatz dazu befindet sich das Host-basierte IDS direkt auf dem zu schützenden System. In einem Host-basierten System werden vor allem interne Daten aus dem System gesammelt und analysiert. Die internen Daten können zum Beispiel aus dem RAM, dem Filesystem oder aus Logfiles stammen. Zusätzlich zum Netzwerk-basierten Ansatz und dem Host-basierten Ansatz gibt es auch noch Hybride Varianten. Dabei werden die beiden Systeme zusammen eingesetzt.

4.1 Unterschiedliche Erkennungsverfahren

Bei Intrusion Detection Systemen gibt es unterschiedliche Verfahren zur Erkennung von Unregelmäßigkeiten im System [vgl. BBK14, Seite 4]:

1. Misuse-based Verfahren
2. Anomaly-based Verfahren
3. Hybride Verfahren

Das Ziel von der Misuse-based Intrusion Detetection ist es, bekannte Angriffsarten sicher erkennen zu können. Dabei basiert die Erkennung auf einer Reihe von Regeln für bekannte Angriffe. Der große Vorteil an diesem Verfahren ist, dass alle bekannten Angriffe sicher erkannt werden und kein falscher Alarm (false positive) vom IDS ausgelöst wird. Der Nachteil wiederum ist, dass auch nur bekannte Angriffe erkannt werden können. Um sowohl unbekannte Angriffsarten als auch bekannte Angriffsarten erkennen zu können, gibt es das Anomaly-based Verfahren. Dieses Verfahren geht davon aus, dass sich das Verhalten von Angreifern erkennbar vom normalen Systemverhalten unterscheidet. Mit Hilfe eines normalen Aktivitätsprofil des Systems wird dann geprüft, ob das auftretende Verhalten signifikant vom dem Normalverhalten abweicht. Dabei kann dann aber auch ein untypisches Normalverhalten fälschlicherweise als Anomalie eingestuft werden. In diesem Fall würde ein falscher Alarm (false positive) vorliegen. Das Hybride Verfahren kombiniert beide Verfahren und kann so sowohl sicher bekannte Angriffe als auch unbekannte Angriffsarten erkennen.

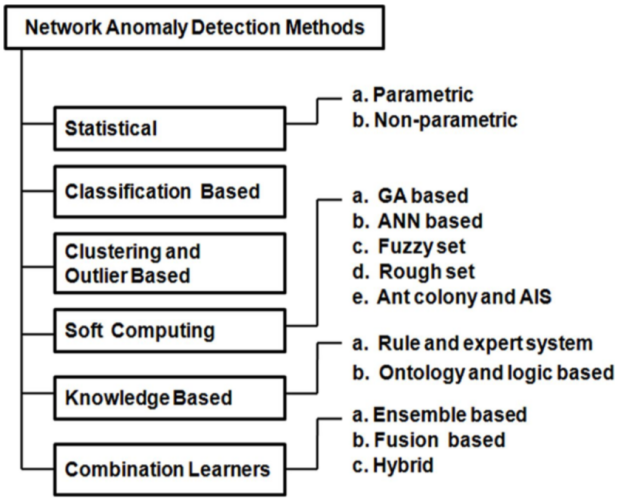


Quelle: https://www.researchgate.net/figure/A-generic-architecture-of-an-Incremental-ANIDS-can-signify-a-fault-in-some-component-of_fig1_233532205

Abb. 3 Die Abbildung zeigt wie ein Anomalie Network Intrusion Detection System typischerweise aufgebaut ist.

4.2 Aufbau eines Network Intrusion Detetection System

Ein typischer Aufbau eines Anomaly-based Network Intrusion Detetection System (ANIDS) ist in Abbildung 3 dargestellt [vgl. BBK14, Seite 5]. Im ersten Schritt **Capture** wird der von außen (z.B. Internet) ankommende Traffic gesammelt. Dabei wird der Traffic sowohl auf Paket-Ebene als auch auf Flow-Ebene aufgenommen. Anschließend gelangt der gesammelte Traffic zum Hauptteil des Systems, der **Anomaly Detection Engine**. Dort werden im **Preprocessing** die ganzen Pakete gefiltert oder verschiedene Netzwerkmetriken wie zum Beispiel die Anzahl der Pakete oder die Paketgröße gesammelt. In der Detection Engine wird dann zuerst eine **Misuse-based Erkennung** durchgeführt, um bekannte Angriffsarten direkt entdecken zu können. Die unbekannten Angriffe wiederum sollen danach von einer **Anomaly-based Erkennung** mit geeigneten Matching Mechanismus (Algorithmus z.B. Clustering) identifiziert werden. Das Matching ist dafür verantwortlich, ob die neue Instanz zu einer bekannten Klasse aus einem Profil gehört. Dazu werden auch die Reference Data, die Informationen über bekannte Angriffssignaturen und Profile vom Normalverhalten speichern, verwendet. Sobald die Anomaly Detection Engine eine Anomalie feststellt, wird ein Alarm ausgelöst. Anschließend werden entweder ein Administrator oder ein System benachrichtigt, die auch wenn notwendig Gegenmaßnahmen einleiten können. Weiterhin gehört zum Aufbau des ANIDS auch die Configuration Data, in denen Zwischenergebnisse abgelegt werden können sowie ein Security Manager.



Quelle: https://ieeexplore.ieee.org/document/6524462 [vgl. BBK14, Seite 9]

Abb. 4 In der Abbildung sind die Methoden für die Network Anomaly Detection in verschiedene Kategorien unterteilt. Einige Kategorien enthalten auch noch Unterklassen.

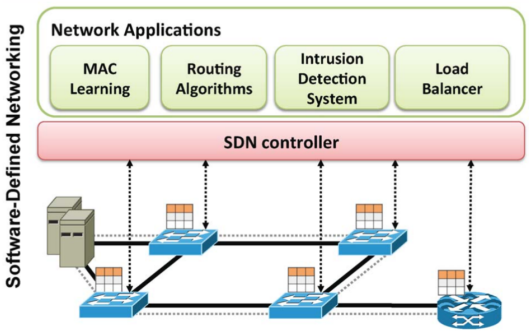
4.3 Network Anomaly Detection Methoden

Grundsätzlich ist das Network Intrusion Detection Problem ein Klassifizierungsproblem [vgl. BBK14, Seite 7]. Dabei liegen als Input verschiedene Daten vor, die sowohl numeri-

sche als auch binäre oder kategorische Werte enthalten können. Für diese Daten muss dann mit Hilfe eines Distanzbestimmungsverfahren (z.B euklidische Abstand für Zahlenwerte) die Nähe zueinander festgestellt werden. Anschließend kann das System mit Hilfe von Trainingsdaten trainiert werden. Dabei können die Datenobjekte vorab als Anomalie bzw. keine Anomalie gekennzeichnet werden, um gelabelte Trainingsdaten zu erhalten. Abhängig davon ob gelabelte Trainingsdaten vorhanden sind, kann das Anomaly Detection System in drei verschiedenen Modi agieren. Die Modi sind supervised, semi-supervised and unsupervised. Im supervised Modi sind Trainingsdaten für das normale Verhalten sowie auch für die Anomalien vorhanden. Beim semi-supervised sind nur Trainingsdaten für das normale Verhalten vorhanden und beim unsupervised Modi sind überhaupt keine Trainingsdaten vorhanden und es wird davon ausgegangen das normale Instanzen deutlich häufiger vorkommen als Anomalien. Methoden, die für Network Anomaly Detection eingesetzt werden können, sind in Abbildung 4 dargestellt [vgl. BBK14, Seite 9]. Es gibt die Klassen Statistical, Classification-based, Clustering and Outlier-based, Soft computing, Knowledge-based und Combination learners.

5 Software Defined Networking

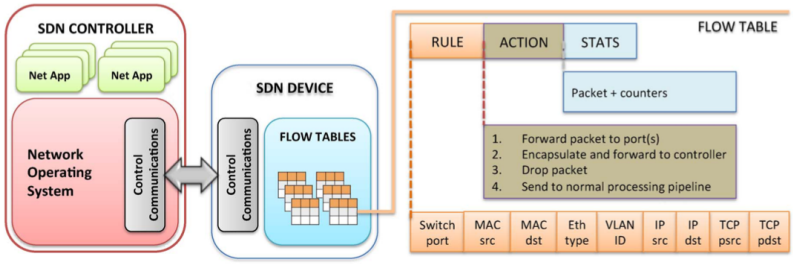
Software Defined Networking (SDN) ist ein neues aufkommendes Netzwerkparadigma [siehe KRV+15]. Es ist entstanden, weil traditionelle IP-Netzwerke sehr komplex und schwierig zu konfigurieren sind. Zum Beispiel müssen Netzwerkadministratoren jedes Netzwerkgerät individuell mit low-level und herstellerspezifischen Befehlen konfigurieren, um bestimmte Regeln (Policies) im Netzwerk zu etablieren. Auch das dynamische Verändern dieser Regeln bei Fehlern oder Änderungen der Netzwerklast ist fast nicht umsetzbar. Daher möchten Software Defined Networks das Netzwerk „programmierbar“ machen, indem es Steuerungslogik von den Routern und Switchen wegnimmt und eine logische zentralisierte Steuerung des Netzes einführt. So kann in einem SDN zwischen der Ebene Data Plane, die nur für das Weiterleiten von Paketen verantwortlich ist, und der Ebene Control Plane, die für die Steuerung der Switche verantwortlich ist, unterschieden werden. In Abbildung 5 stellen die Switche das Data Plane dar und der SDN Controller inklusive der verschieden Applikationen die Control Plane.



Quelle: Kreuz et al.: Software-Defined Networking: A Comprehensive Survey [Seite 6]

Abb. 5 Die Abbildung zeigt einen einfachen Aufbau eines SDNs mit Control Plane und dem darunterliegenden Data Plane.

In einem SDN steuert der Controller die Switche im Netzwerk durch ein wohldefinierte API (Open southbound API). Das wohl bekannteste Beispiel für diese API ist OpenFlow. Mit Hilfe dieser Api können in einem SDN Devices Einträge in einer Flowtabelle geschrieben werden (siehe Abbildung 6). Anhand dieser Tabelleneinträge entscheidet das Device wie es den ankommenden Traffic behandeln soll. Dabei besteht ein Tabelleneintrag aus 3 verschiedenen Abschnitten. Zuerst wird definiert für welche Pakete eine bestimmte Regel gelten soll. Zum Beispiel kann eine spezifische Regel für alle Pakete, die auf einem Port des Switches eingehen formuliert werden oder für eine bestimmte Zieladresse. Der zweite Abschnitt der Regel beschreibt dann welche Aktion mit diesem Traffic durchgeführt werden soll. Aktionen können zum Beispiel das Weiterleiten zu einem bestimmten Port, das Wegwerfen des Paketes oder das Schicken des Paketes zum zentralen Controller sein. Der dritte Abschnitt des Tabelleneintrages ist für das Zählen der Pakete, die die Regel erfüllen, verantwortlich. Diesen Daten können dann auch für das Monitoring und die Analyse des Systems benutzt werden.



Quelle: Kreuz et al.: Software-Defined Networking: A Comprehensive Survey [Seite 11]

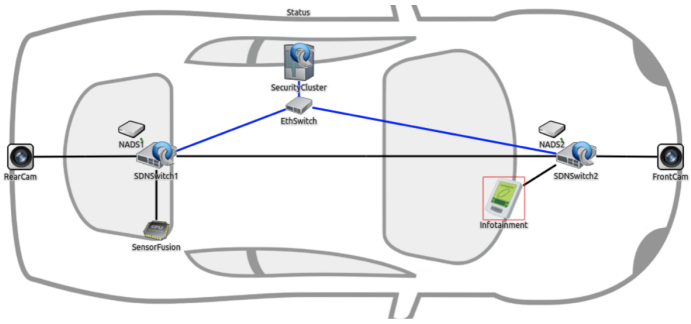
Abb. 6 Die Abbildung demonstriert wie SDN-Switche anhand von Flowtabellen vom zentralen Controller aus programmiert werden und ankommende Pakete verarbeiten.

5.1 Vor- und Nachteile von SDN

Ein großer Vorteil von SDN liegt in den Kosten für die Switche. Denn SDN-Switche benötigen im Vergleich zu klassischen Switchen nur geringe Mengen an Rechenleistung und Software, da sie einfach nur Pakete anhand von Regeln in einer Tabelle verarbeiten müssen. Ein weiterer Vorteil ist die Zentralisierung, die es zum Beispiel ermöglicht optimale Routen durch das Netzwerk zu bestimmen und das dynamische Verteilen von Ressourcen um QoS Anforderungen besser erfüllen zu können. Weiterhin kann ein SDN auch als Firewall zum Internet fungieren und Pakete herausfiltern oder unbekannte Pakete zum SDN Controller sowie höheren Sicherheitsmechanismen wie einem Network Intrusion Detection System weiterleiten. Ein Nachteil vom SDN ist, dass der zentrale Controller einen Single Point of Failure darstellt oder die erhöhte Netzlast durch das Verschicken von Steuerungsoperationen für die Switche.

5.2 SDN im Kontext Auto

Abbildung 7 demonstriert wie ein SDN in ein Fahrzeug integriert werden könnte. In diesem Beispiel sind zwei SDN-Switche im Netzwerk vorhanden, die über einen weiteren Switch mit dem SecurityCluster (dem zentralen Controller) verbunden sind. Weiterhin befindet sich ein Infotainment Steuergerät mit nicht-sicherheitskritischer Kommunikation innerhalb des Netzwerkes. Die komplette Kommunikation der Infotainment Komponente mit anderen Steuereinheiten gelangt zuerst zu einem SDN-Switch. Der SDN-Switch kann anhand der Flowtabelle entscheiden, ob der Traffic weitergeleitet, weggeworfen oder für die weitere Analyse zum SecurityCluster weiterleitet werden soll. Wenn nun zum Beispiel durch ein Anomaly Detection System festgestellt wird, dass die Infotainment Komponente übernommen wurde, kann das SecurityCluster durch einen neuen Eintrag in der Flowtabelle die komplette Kommunikation oder einzelne Flows der Infotainment Komponente mit anderen Komponenten deaktivieren. Dann wird der am SDN-Switch ankommende Traffic der Infotainment Komponente direkt verworfen. Zuvor sind die SDN-Switche so eingestellt das nur Komponenten, die miteinander kommunizieren müssen, auch miteinander kommunizieren dürfen (Whitelist).



Quelle: https://secvi.inet.haw-hamburg.de [siehe Sec] (Demo für den Automotive Ethernet Congress 2019 im Februar in München)

Abb. 7 Die Abbildung zeigt eine beispielhaften Einsatz von SDN im Fahrzeug.

6 Zusammenfassung und Ausblick

In der aktuellen domänenbasierten Architektur des Fahrzeuges fehlen wichtige Sicherheitsmechanismen, um eine hohe Informationssicherheit (Security) garantieren zu können. Die zunehmende Kommunikation mit der Umgebung (z.B. Internet oder Fahrzeug-Ad-hoc-Netze) und die wachsende Anzahl an Schnittstellen zur Außenwelt erfordern neue Sicherheitsmechanismen im Fahrzeug. In dieser Ausarbeitung wurden die Konzepte Network Intrusion Detection und Software Defined Networking aus der klassischen IT-Security als mögliche Bausteine für das gesamtes Sicherheitssystem des Autos vorgestellt. Außerdem wurde ihr Einsatz im Kontext des Autos beschrieben.

Im weiteren Verlauf meines Masterstudiums möchte ich mich tiefer mit Network Intrusion Detection befassen. Dazu möchte ich zuerst im Grundprojekt meinen Fokus auf statistische Verfahren sowie Klassifizierungsverfahren legen. Anschließend im weiteren Verlauf auch noch die weiteren Verfahren wie Clustering etc. genauer betrachten. Als Ergebnis meiner Masterarbeit soll am Ende ein Anomaly Detection System für einen Teil des Fahrzeugprototypen aus dem SecVi-Projekt entstehen.

Literatur

AUT. AUTOSAR. Specification of lin interface.

BBK14. M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys Tutorials*, 16(1):303–336, First 2014.

BCK+12. C. Buckl, A. Camek, G. Kainz, C. Simon, L. Mercep, H. Stähle, and A. Knoll. The software car: Building ict architectures for future electric vehicles. In *2012 IEEE International Electric Vehicle Conference*, pages 1–8, March 2012.

Bos. Bosch. Can specification.

CMK+11. Stephen Checkoway, Damon Mccoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive Experimental Analyses of Automotive Attack Surfaces. *USENIX Security*, 2011.

COO. MOST COOPERATION. Most specification.

Dav15. Andy Davis. Broadcasting your attack: Security testing dab radio in cars. *Blackhat Conference*, 2015.

Eck14. Claudia Eckert. *IT-Sicherheit Konzepte - Verfahren - Protokolle. 9., aktualisierte und korr. Aufl.* Oldenbourg, 2014.

Fle. FlexRay. Flexray communications system protocol specification.

HAF+09. O. Henniger, L. Apvrille, A. Fuchs, Y. Roudier, A. Ruddle, and B. Weyl. Security requirements for automotive on-board networks. In *2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST)*, pages 641–646, Oct 2009.

KRV+15. D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, Jan 2015.

Lev18. Ziv Levi. Automotive ethernet: The future of in-car networking? *electronicdesign*, 2018.

MV15. Charlie Miller and Chris Valasek. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015:91, 2015.

RMM+10. Ishtiaq Rouf, Rob Miller, Hossen Mustafa, Travis Taylor, Sangho Oh, Wenyuan Xu, Marco Gruteser, Wade Trappe, and Ivan Seskar. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In *Proceedings of the 19th USENIX Conference on Security*, USENIX Security'10, pages 21–21, Berkeley, CA, USA, 2010. USENIX Association.

Rö18. Jürgen Röder. Die zukunft der vernetzten fahrzeugarchitektur - the future of in-vehicle data management. *VDI Wissensforum*, 2018.

Sec. Projekt SecVI. Security for vehicular information - research project for secure automotive network architectures.

SNA+13. I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaâniche, and Y. Laarouchi. Survey on security threats and protection mechanisms in embedded automotive networks. In *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, pages 1–12, June 2013.