Universität Hamburg
Fachbereich Informatik

IT-Sicherheit und Schutz vor Malware im Automobil

am Arbeitsbereich Sicherheit in Verteilten Systemen (SVS)

Arne Beer, Stefan Grusche, Joshua Stock

11. Januar 2016

Abstract

In modernen Automobilen werden immer häufiger computergestützte Systeme integriert. Diese dienen hauptsächlich dem Zweck, die Sicherheit von Insassen und fremden Verkehrsteilnehmern zu verbessern, indem dem Fahrzeugführer zunehmend die Kontrolle aus der Hand genommen wird. Einhergehend mit dieser Integration erhalten jene Komponenten immer häufiger Zugriff auf elementare Bestandteile des Fahrzeugs, beispielsweise auf Lenkung und Bremsen.

Um das potentielle Sicherheitsrisiko, das (teil-)automatisiertes Fahren in Kombination mit einer üblicherweise hohen Vernetzung birgt, zu diskutieren, wurde zunächst der CAN-Bus als zentrale Netzkomponente untersucht. Anschließend wurde auf die verwundbarsten Teilsysteme eines vernetzten Automobils eingegangen, bevor die detaillierte Vorgehensweise einer Cyber-Attacke auf ein Fahrzeug dargestellt wurde. Abschließend wurden mögliche Maßnahmen, um vorhandene Sicherheitslücken zu schließen, aufgezeigt und diskutiert.

Inhaltsverzeichnis

1 Einleitung

Mit dem Ziel das Sicherheitsrisiko im Straßenverkehr zu minimieren, wurden und werden zahlreiche computergestützte Systeme entwickelt. Einerseits greifen diese direkt in Notsituationen ein, um Fahrzeuginsassen sowie anderen Verkehrsteilnehmer ggf. das Leben zu retten (zum Beispiel durch automatisierte Notfallbremsungen). Andererseits können sie schon im Vorfeld das Eintreten derartiger Notsituationen zu verhindern versuchen (zum Beispiel Spurhalteassistenten). Angetrieben von der allgemeinen Digitalisierung unseres Alltags finden auch zunehmend komplexere Multimediasysteme Einzug in moderne Automobile. Diese integrieren neben typischen Unterhaltungsanwendungen heutzutage auch Navigationssysteme, Möglichkeiten für WLAN-Hotspots oder gar Einparkassistenten, die dem Nutzer vor allem einen höheren Komfort bieten sollen. Gerade derartige Assistenten benötigen oft direkten Zugriff auf sicherheitskritische Funktionen, wie Lenkung oder Bremsen, und nehmen dem Fahrer so die alleinige Kontrolle über das Fahrzeug.

Genannte Systeme verfügen in den meisten Fällen über verschiedene Kommunikationsarten. Nicht nur die Bereitstellung von Online-Multimediadiensten (zum Beispiel Musikstreaming), sondern auch der Austausch von Daten mit dem Hersteller verlangen eine digitale Verbindung zur Außenwelt. Da nicht nur Sicherheits-, sondern oft auch Multimediasysteme Zugriff auf fahrzeuginterne Netze zwischen kritischen Komponenten besitzen, existiert ein besonderes Schutzbedürfnis sämtlicher im Fahrzeug befindlicher Systeme.

Diese Arbeit dient dem Überblick über den aktuellen Forschungsstand im Bereich der automobilen Sicherheit, wobei der Fokus auf dem Komprimittieren von Fahrzeugen über integrierte Computersysteme liegt. Wir konzentrieren uns auf Angriffe, die ohne physische Verbindung zum Fahrzeug stattfinden und gegebenenfalls über große Distanzen ausgeführt werden können.

Einleitend werden wir uns mit den Grundlagen des sogenannten „CAN-Busses“ beschäftigen. Wir erklären die Funktion des Busses und wie die Nachrichtenübertragung im Detail abläuft. Zusätzlich untersuchen wir die Sicherheit dieser Art des Nachrichtenaustauschs. Wir stellen eine Auswahl wichtiger Angriffsvektoren vor, die ein Angreifer, der nicht in physischem Kontakt zum Fahrzeug steht, nutzen kann, um ein Automobil zu komprimittieren und zeigen anhand von Forschungsergebnissen Miller und Valaseks [**?**], dass diese auch außerhalb der Theorie von Bedeutung sind. In Anbetracht des aktuellen Sicherheitsstandes in der Automobilindustrie fassen wir abschließend verschiedene Vorschläge zur Verbesserung der IT-Sicherheit aus diversen Arbeiten zusammen.

2 Hauptteil

2.1 CAN-Bus

2.1.1 Funktion und Aufbau

Die meisten IT-Sicherheitslücken in Fahrzeugen bauen auf Schwachstellen in der zentralen Kommunikationsschnittstelle eines Autos, dem sogenannten CAN-Bus, auf.

Der „Controller Area Network“ Bus ist ein klassisches Binary Unit System, welches die meisten Systeme eines PKWs miteinander verbindet. Hierbei ist anzumerken, dass in vielen Autos mehrere CAN-Busse verbaut sind, welche unabhängig voneinander agieren können. Die an den CAN-Bus beziehungsweise die CAN-Busse gebundenen Systeme und Komponenten werden „Electronic Control Units“ genannt, kurz ECUs[1], eine spezielle Art von TCUs sind „Telematic Control Units“[2].

Die übliche Funktionsweise eines CAN-Busses lässt sich gut am Beispiel des Toyota Prius erklären [**?**]: ECUs kommunizieren untereinander über den Bus, wobei jedes gesendete Paket an jede andere ECU am selben Bus gesendet wird („Broadcast“). Die ECUs selbst entscheiden anschließend anhand der Struktur der Pakete, ob die Informationen für sie bestimmt sind und verarbeiten diese, falls dies der Fall ist.

1. „Electronic Control Units“können Geräte verschiedenster Art sein und müssen keine Verbindungen zur Außenwelt aufweisen. ECUs sind entscheidende Bauteile eines Automobils und ihre korrekte Funktionsweise sind unverzichtbar für einen geregelten Betrieb. Beispiele für ECUs wären Parkassistenz-Systeme oder Lenkwinkelsensoren. Weitere Beispiele siehe Abb. 2.1
2. „Telematic Control Units“: Steuergeräte für Telematikdienste. Sind in der Regel direkt an den CAN-Bus angeschlossen und zeichnen sich häufig durch viele ausgehende Verbindungen aus (Bluetooth, Mobilfunk, GPS, etc.)

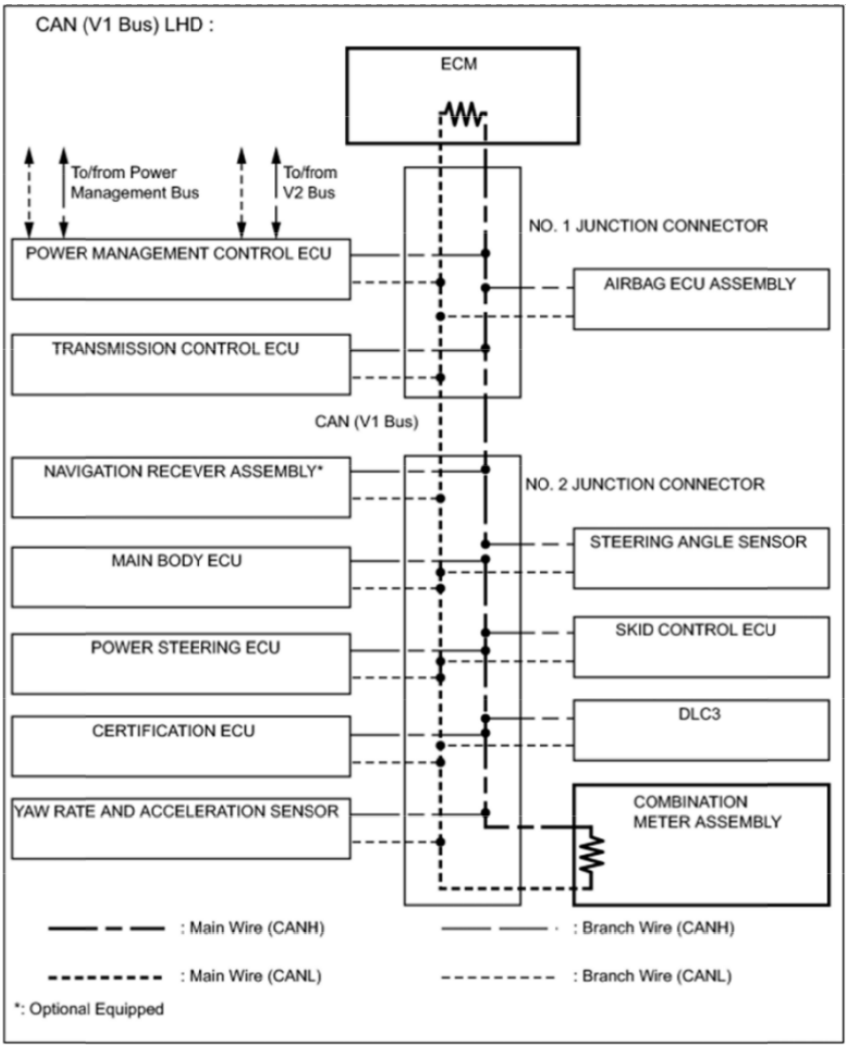

Abbildung 2.1: Schematischer Aufbau eines CAN-Busses [**?**].

Nachfolgend ein Beispiel für ein Paket, das auf einem CAN-Bus kommuniziert wird, in hexadezimaler Schreibweise:

```
00 B4 08 00 00 00 00 3C 18 C0 FF
```

Das Paket kann wie folgt interpretiert werden: Die ersten 2 Byte, in obiger Schreibweise durch 00 *B*4 repräsentiert, definieren den Identifikator (’ID’) der adressierten ECU. Die nachfolgenden 4 Bit definieren die Länge der nachfolgenden Nutzdaten, also der relevanten Daten oder dem Befehl. In diesem Falle sind es 8 Byte, also die Sequenz (00 00 00 00 3C 18 C0 FF).

Obiger Befehl ließe sich auch in folgender Schreibweise darstellen, auch wenn hierbei durch die fehlenden Byte Informationsverlust auftreten kann:

```
00 B4 04 3C 18 C0 FF
```

In diesem Paket wird lediglich eine Nutzdatenlänge von 4 Byte definiert. Um Nachrichtenintegrität sicherzustellen, wird häufig eine Prüfsumme erstellt. Diese wird üblicherweise als letztes Byte der Nutzdaten an das Paket gehängt und wird durch einen vom Hersteller definierten Algorithmus generiert. Im Folgenden beispielhaft ein Befehl an das Geschwindigkeitstacho eines Ford Escape: Die Addresse ist wie vorher 00 *B*4 mit einer Nutzdatenlänge von 08. Die Nutzdaten strukturieren sich wie folgt:

```
AA BB 00 00 CC DD 00 00
```

AABB definiert die RPM, während *CCDD* die Geschwindigkeit angibt. Die Werte werden nach folgenden Formeln berechnet:

$$\text{Speed}(mph) = 0.0065 \cdot (CCDD) - 67$$
$$RPM = 0.25 \cdot (AABB) - 24$$

Dementsprechend würde ein Befehl mit ca. 100km/h und 2000RPM folgendermaßen aussehen:

```
1F 40 00 4E BC 00 00
```

Der komplette Befehl hätte die Form:

```
00 B4 08 1F 40 00 4E BC 00 00
```

Nehmen wir an, dass noch eine Prüfsumme angefügt wird, nach der Formel:

$$\text{Checksum} = (IDH + IDL + Len + Sum(\text{Data}[0] - \text{Data}[Len - 2])) \& 0xFF$$

Dementsprechend ist wäre das letzte Byte *CC*:

```
00 B4 08 1F 40 00 4E BC 00 CC
```

Der simple Aufbau der Pakete ist jedoch nur einer der Gründe, weshalb Angriffe auf den CAN-Bus vergleichsweise einfach sind.

2.1.2 Sicherheit

Wie bereits erwähnt, stellt die Kommunikation über den CAN-Bus das größte Sicherheitsproblem dar: Bei den meisten Autos verläuft diese vollkommen unverschlüsselt. Außerdem kommunizieren die ECUs ohne jegliche Authentifizierung. Deshalb können schadhafte Pakete auf den Bus gelegt werden und im Regelfall akzeptiert und verarbeitet es die betroffene ECU problemlos.

Ferner sind die Protokolle und Programmierschnittstellen („APIs“), mit denen die ECUs kommunizieren, nicht öffentlich einsehbar und nur dem jeweiligen Hersteller bekannt. Zwar erschwert dies den Entwurf valider Pakete, allerdings kann ein Angreifer durch den Broadcast-Verkehr der Pakete auf dem CAN-Bus beliebig viele ebensolcher ohne Probleme für eine spätere Analyse mitschneiden. Selbstverständlich wird dabei Zugriff auf den Bus selbst benötigt, die Analyse kann aber zum Beispiel auch an einem baugleichen Modell geschehen, welches der Angreifer besitzt oder mietet. Da der abgehörte Verkehr auf dem Bus unverschlüsselt ist, kann der Mitschnitt nun benutzt werden, um die API der ECUs zu rekonstruieren: Hierfür muss lediglich nach Mustern im Paketstrom gesucht werden, welche in Abhängigkeit zum Zustand des Autos stehen. Durch dieses Verfahren wurde beispielsweise die oben gezeigte API des Geschwindigkeitstachos rekonstruiert.

Durch das Senden und Empfangen Aller an Alle entsteht ein weiteres Problem. Mit jedem Gerät, welches an dem Bus hängt, wird ein neuer Angriffsvektor bereitgestellt. Solange ein Angreifer physikalischen Zugriff auf den Bus hat, kann er also einen eigenen Empfänger am Bus befestigen oder die Firmware vorhandener ECUs manipulieren. Durch die zunehmende Vernetzung des

Automobils oder technisch beabsichtigte Bequemlichkeiten wie Firmware-Updates per CD/USB oder gar Funk/Internet, steigt die Kompromittierbarkeit von an dem CAN-Bus befindlichen Geräten zunehmend.

2.2 Ausgewählte Angriffsvektoren

Nicht alle Systeme, TCUs oder ECUs, die über den CAN-Bus kommunizieren, sind als Angriffsvektor geeignet. Entscheidend für einen Angriff ohne physischen Zugriff auf den Bus oder die Systeme selbst ist eine von außen erreichbare Verbindung. Neben vielen Geräten mit eingeschränkter Möglichkeit des unerlaubten Fremdzugriffes taten sich die folgenden drei Vektoren als besonders schwerwiegende Sicherheitslücken auf:

2.2.1 Bluetooth

Eine Bluetooth-Schnittstelle zählt mittlerweile zu der Standardausstattung eines modernen Autos. Ihre Hauptfunktion ist meist die Verbindung eines Mobiltelefons zum Media-System des Autos, um beispielsweise Anrufe über eine integrierte Freisprecheinrichtung entgegenzunehmen, aber auch, um Anrufe aus dem Auto über das Mobiltelefon zu starten oder Musik vom Mobiltelefon direkt über das Mediasystem des Autos wiederzugeben. In dem PKW, den Checkoway et al. in [**?**] betrachten, befindet sich das Bluetooth Modul in der Telematik-Einheit. Checkoway et al. konnten durch Reverse-Engineering das Unix-ähnliche Betriebssystem analysieren, inklusive der Bluetooth-Stack-Implementation. Den Forschern gelang es wegen Sicherheitslücken in ebendieser Implementation, einen Kommandozeileninterpreter („Shell“) auf der Telematik-Einheit zu öffnen.

Sie unterscheiden zwischen zwei Angriffsmöglichkeiten: Zum einen werden *indirekte* drahtlose Kurzstrecken-Attacken aufgeführt, bei denen ein bereits über Bluetooth mit der Telematik-Einheit gekoppeltes, beliebiges Android- oder iOS-Smartphone benötigt wird, zum anderen beschreiben sie *direkte* drahtlose Kurzstrecken-Attacken. Für letztere muss die MAC-Adresse des Bluetooth-Moduls im Auto bekannt sein, diese kann allerdings mit wenig Aufwand durch verschiedene „Sniffing“-Software herausgefunden werden. Um nun eine Bluetooth-Kopplung zwischen Automobil und Angreifer zu erzwingen, muss lediglich eine PIN, die sich erst nach Neustart des PKWs ändert, herausgefunden werden. Auch diese stellte kein großes Hindernis dar, mittels Brute-Force-Methode gelang es den Forschern teilweise innheralb von 15 Minuten, sich ohne jede Benutzerinteraktion mit der Telematik-Einheit des Autos zu verbinden und die Bluetooth-Sicherheitslücken auszunutzen.

Miller und Valasek, die sich in [**?**] mit dem 2010 Ford Escape beschäftigten, konnten zwar keine Möglichkeit finden, ein Gerät ohne Benutzer-Interaktion mit dem Auto zu koppeln, sehen den Bluetooth Stack jedoch trotzdem durch seine Größe als „eine der größten und machbarsten Angriffsflächen eines modernen Automobils“.

2.2.2 Mobilfunk

Eine weitere ausgehende Verbindung der Telematikeinheit ist oftmals der Mobilfunk. Durch ein eingebautes 3G-Modem kann das Automobil so mit dem Internet verbunden sein, oder im Falle eines schweren Unfalls selbständig eine SMS an Rettungskräfte senden.

Über die Telematikeinheit können jedoch auch Kommandos übertragen werden. Durch das Mitschneiden des 3G-Verkehrs gelang es den Autoren von [**?**], Teile des verwendeten Protokolls zu rekonstruieren. Durch Umschalten der Telematikeinheit-Software in den Debug-Modus konnten sich die Forscher alle ein- und ausgehenden Daten in Bitform anzeigen lassen. In Kombination mit den aufgezeichneten Daten war es ihnen möglich, die Paketstrukturen zu rekonstruieren. Die Software des Systems war darauf eingestellt, dass empfangene Pakete niemals größer als 100 Byte sind. Diese Sicherheitslücke machten sich die Forscher zunutze, um ein größeres Paket einzuschleusen, welches zu einem Buffer-Overflow führte. Dadurch wurde schädlicher Code in den Speicher des Systems geschrieben, teilweise direkt auf den Callstack des Programms. Eine vorhandene Sicherheitsmaßnahme, die nach wenigen Sekunden eine Authentifizierung des Absenders verlangt, konnte durch einfache Simulation der Authentifzierung leicht umgangen werden.

2.2.3 Nachrüstbare TCUs

Zusätzlich zu den unterschiedlichen Medien- und Diagnosesystemen, die bereits durch den Hersteller in die Fahrzeuge integriert sind, gibt es verschiedenste nachrüstbare Gerätschaften, die dazu dienen können, den Funktionsumfang eines Fahrzeugs zu erweitern (fortgeschrittene Medienanwendungen etc.) oder dem Fahrzeughalter detaillierte Informationen über den Zustand des Autos zu geben. Besonders interessant ist hierbei die Sparte der TCUs, *Telematic Control Units*, die verschiedenste Funktionen und Anwendungszwecke besitzen können. Die meisten dieser TCUs verfügen dabei über einen GPS-Sensor, oft unterschiedliche Beschleunigungssensoren und eine Mobilfunk und/oder Internetanbindung, um entweder mit den externen Systemen des TCU-Herstellers kommunizieren zu können, sei es um zum Beispiel aufgezeichnete Daten zu Auswertungszwecken zu übertragen oder die eigene Soft- oder Firmware zu aktualisieren, oder um sich mit Endgeräten der Fahrzeuginsassen bzw. des Fahrzeughalters zu verbinden.

Ein prominentes Einsatzfeld nachgerüsteter TCUs ist die Aufzeichnung des Fahrverhaltens durch Versicherungen, die basierend auf den aufgezeichneten Daten ihre Beiträge berechnen. Eine in diesem Umfeld angewandte TCU ist zum Beispiel das Modul „Snapshot“ der Firma Progressive. Andere TCUs (beispielsweise „Automatic“ von Automatic Labs) zielen direkt darauf ab, das Fahrverhalten zu verbessern, indem sie über Smartphones der Fahrzeughalter in Echtzeit zum Beispiel Hinweise für eine motor- und benzinschonenderes Beschleunigungs- und Bremsverhalten geben. Häufig finden TCUs auch im Flottenmanagement Anwendung, wo sie zur Lokalisierung und Überwachung der Geschäftfahrzeuge dienen. „Connect“ des Herstellers Delphi ist eine besonders funktionsreiche TCU, die neben bereits erwähnten Anwendungen wie der Lokalisierung auch detaillierte Informationen über die Gesundheit des Fahrzeugs ausgibt, für Eltern die Möglichkeit bietet, ihren Kindern einen Bereich zur Nutzung des Fahrzeugs festzulegen, als Schlüsselersatz zum Öffnen, Schließen und Starten des Autos dient und über 4G-Mobilfunk einen Internet-Hotspot für die Fahrzeuginsassen bereitstellen kann.

TCUs von Fremdherstellern werden üblicherweise direkt an den OBD-Port des Fahrzeugs angeschlossen und werden auf diese Weise direkt in das interne Netzwerk integriert, können im

Allgemeinen Nachrichten vom CAN-Bus mitlesen und beinhalten oft auch selbst die Funktionalität, Nachrichten auf den CAN-Bus zu schreiben. Dies ist dabei die größte Gefahr einer nachgerüsteten TCU: eine oft umfangreiche Internet- oder Mobilfunkfähigkeit und dabei theoretisch uneingeschränkter Zugriff auf sicherheitskritische Systeme bietet Angreifern eine optimale Angriffsfläche, die bei einem erfolgreichen Eindringen von außen u.U. eine volle Kontrolle über das Fahrzeug bedeutet.

In [**?**] haben Foster et al. anhand einer TCU zur Fahrstilanalyse für Versicherungszwecke aufgezeigt, wie ein unsicherer Entwurf des Update-Protokolls letztendlich zur Übernahme des Geräts und damit zum Zugriff auf das Fahrzeug führen konnte. Im beschriebenen Fall waren sie in der Lage, nach Ausfindigmachen der Telefonnummer der TCU, die sie für Mobilfunkdienste besaß, mit einer einfachen SMS einen Update-Vorgang an einem beliebigen Server anzustoßen. Aufgrund einer nicht vorhandenen Authentifizierung des Servers und einer ebenso fehlenden Signatur der Updates konnten beliebige eigene Dateien auf die TCU heruntergeladen werden und mit entsprechenden Befehlen ausgeführt werden. Im Endeffekt erlangten die Angreifer die Möglichkeit, sich mit Root-Berechtigungen in die TCU einzuwählen und erhielten somit freie Verfügung über das komplette System. Die Verbindung zum CAN-Bus bot dann an, selbst Nachrichten zu verschicken. In einer Proof-of-Concept-Attacke zeigten Foster et al. abschließend, dass sie u.a. in der Lage waren, die Scheibenwischer und einzelne Bremsen zu betätigen.

Fehler dieser Natur bei dem Entwurf derart sicherheitsrelevanter Systeme stellen also eine große Gefahr für den Fahrzeughalter dar. Diese wird noch dadurch verstärkt, dass der Fahrzeughersteller selbst nicht in der Lage ist, Fehler zu beheben. Da es sich um fremde Systeme handelt, muss hier der jeweilige Hersteller selbst aktiv werden und dafür sorgen, dass alle sich im Umlauf befindenen TCUs eventuelle Updates erhalten.

2.3 Praxisbeispiel: 2014er Jeep Cherokee

Im Folgenden stellen wir anhand der Forschungsergebnisse von Miller und Valasek aus [**?**] dar, wie ohne direkten Zugang verschiedene Funktionen auf diversen Systemen eines konkreten Fahrzeugs, in diesem Fall eines 2014er Jeep Cherokees, ausgeführt werden können und so eine direkte Gefahr für den Straßenverkehr geschaffen werden kann.

Der Jeep Cherokee besitzt einige Systeme, die für einen Angreifer von großem Interesse sein können. So ist u.a. ein adaptiver Tempomat, der von sich aus Fahrzeugbremsen auslöst, ein Kollisionswarnsystem, welches das Fahrzeug ebenfalls zu einem kompletten Halt bringen kann, ein Spurassistent, der, wenn auch minimal, in die Lenkung eingreifen kann, und einen Einparkassistenten, welcher wiederum komplette Kontrolle über das Lenkrad erhält, integriert. Des Weiteren ist ein Multimediasystem mit WLAN- und Mobilfunkverbindung eingebaut, welches gleichzeitig an beide vorhandenen CAN-Netzwerke im Auto angeschlossen ist. Aus genau diesem Grund ist dieses System das Hauptaugenmerk von Miller und Valaseks Arbeit, da über ein Kompromittieren der Medieneinrichtung potentiell die größte Kontrolle über das Auto erlangt werden kann. Die Autoren konzentrierten sich zunächst auf Schwächen der eingebauten Hotspot–Funktionalität; sofern der Fahrzeughalter ein Abonnement beim Hersteller abschließt, bietet das Fahrzeug an, einen WLAN-Hostpot für die Fahrzeuginsassen bereitzustellen. Das Netz an sich wies keine unüblichen Schwachstellen auf, Miller und Valasek konnten aber Schwächen bei der zufälligen Generierung der Netzpasswörter entdecken: Das Passwort wird stets durch eine Funktion berechnet, die auf dem Zeitpunkt des Aufrufens beruht. Dieser kann zusätzlich

stark eingeschränkt werden, da sich das System, solange keine Uhrzeit über GPS oder Mobilfunk empfangen wurde, auf eine Standarduhrzeit einstellt.

Mit Verbindung zum WLAN-Netzwerk konnten Miller und Valasek mithilfe üblicher Netzwerkanalysetools diverse offene Ports identifizieren, die sich potentiell zum Ausführen von Angriffen anbieten. Insbesondere der Port 6667 erwies sich als interessant, da sich dahinter nicht, wie üblich, ein IRC-Server versteckte, sondern eine Variante von D-Bus über IP. D-Bus ist ein Framework, welches der Interprozesskommunikation dient: Es ermöglicht Prozessen, Funktionen an anderen Prozessen aufzurufen. Zusätzlich stellte sich heraus, dass sich ein Angreifer ohne Authentifizierung mit diesem System verbinden konnte. Mithilfe der D-Bus-Bibliotheken und eines Debugging-Tools konnten Miller und Valasek ohne Probleme aufrufbare Funktionen identifizieren, darunter eine, die vom Nutzer übergebene Shell-Kommandos ausführt: eine deutliche Sicherheitslücke und der attraktivste Angriffspunkt des Systems.

Über diese Lücke erwies es sich als trivial, Code auszuführen. Mit sehr simplen Skripten kann ein Angreifer nun unter anderem GPS-Daten auslesen, um den Standort des Fahrzeugs festzustellen, er kann die Geschwindigkeit der Lüfter, die Lautstärke (Beispielcode: siehe unten) und den Basspegel des Radios verändern, zusätzlich den Radiosender wechseln, bestimmen, was auf dem integrierten Display erscheint und eigene Bilder einspielen.

```
1 require "service"
2
3 params = {}
4 params.volume = tonumber(arg[1])
5 x=service.invoke("com.harman.service.AudioSettings", "setVolume",
      params)
```

Listing 2.1: Beispielskript zum Ändern der Lautstärke.

Das Eindringen in ein WLAN-Netz, obwohl kein direkter physischer Zugriff zum Fahrzeug benötigt wird, kann nur aus begrenzter Entfernung, etwa 30m, erfolgen. Aus diesem Grund änderte sich nun der Fokus von Miller und Valasek auf die Mobilfunk-Fähigkeiten des Multimediasystems. Unter anderem für mobilen Internetzugang besitzt der Jeep eine 3G-Verbindung zum Netz eines Mobilfunkanbieters (Im konkreten Testfall handelte es sich um den amerikanischen Anbieter Sprint). Es konnte beobachtet werden, dass innerhalb des Netzes des Mobilfunkanbieters einerseits dem Fahrzeug stets eine IP-Adresse in einem von zwei Adressbereichen zugeordnet wird und andererseits auch in dieser Umgebung der Zugang zum D-Bus-System für fremde Geräte offen ist. Dies bedeutet, dass, sofern sich der Angreifer im Netz des gleichen Anbieters wie das Fahrzeug befindet, die zuvor beschrieben Attacken nicht nur in direkter Umgebung, sondern über das ganze Land (beziehungsweise der Reichweite des Netzbetreibers) ausgeführt werden können. Zudem konnten Miller und Valasek über IP-Analysen feststellen, dass diverse andere Fahrzeuge, die dasselbe Mediensystem integriert haben, im gleichen Adressbereich kommunizieren und durch ähnliche Angriffe gefährdet sind.

Ein weiterer Teil von Miller und Valaseks Arbeit konzentriert sich auf das Einschleusen gefälschter CAN-Befehle. Das Multimediasystem kann CAN-Befehle nicht direkt auf den CAN-Bus schreiben, aber es kann mit einem Chip kommunizieren, der genau diese Fähigkeit besitzt. Zusätzlich kann die Firmware des Chips über das Mediensystem aktualisiert werden. Dies nutzen Miller und Valasek aus: Sie laden über das Multimediasystem eine modifizierte Firmware auf besagten Chip, welche es ihnen ermöglicht, CAN-Befehle auf die CAN-Netzwerke zu versenden. Dieses Update birgt dabei das Risiko, als Angreifer bemerkt zu werden, da das

Multimediasystem und der zum CAN-Netzwerk verbundene Chip zuerst in einen „Update mode“ überführt werden müssen. Dies kann nur über einen Neustart der Systeme geschehen, welcher unter Umständen von Fahrzeuginsassen bemerkt wird.

Mit der Möglichkeit, eigene CAN-Befehle an das Auto zu schicken, widmeten sich Miller und Valasek der Frage, wie diese Nachrichten aufgebaut sind. Hierfür machten sie von Diagnosegeräten Gebrauch, die in gewöhnlichen Werkstätten benutzt werden. Der Jeep nutzt intern zwei Arten an CAN-Befehlen, einerseits „normale“, die im herkömmlichen Betrieb des Autos ständig verschickt werden, andererseits diagnostische Befehle, die nur zu Wartungszwecken existieren, daher auch nur bei geringer Geschwindigkeit von den ECUs akzeptiert werden. Zur Fälschung dieser Befehle ist es notwendig, den benutzen Algorithmus zur Berechnung der Prüfziffer zu kennen. Die Rekonstruktion des vom Jeep benutzten Vorgehens erwies sich als untypisch aufwändig, da Prüfziffern nicht auf herkömmliche Arten berechnet werden, sondern ein eigener Algorithmus zum Einsatz kommt. Reverse-Engineering eines erworbenen Parkassistentmoduls brachte aber auch diesen zum Vorschein.

Mit einem vollständigen Verständnis der CAN-Befehle waren Miller und Valasek zum Beispiel in der Lage, über „normale“ CAN-Befehle den Blinker des Autos zu setzen, die Türen zu öffnen, zu schließen und die auf dem Tachometer angezeigte Drehzahl zu manipulieren. Bis auf die Berechnung der Prüfsumme sehen diese CAN-Befehle ähnlich wie zuvor in Abschnitt 2.1 beschrieben aus, erwähnter Befehl zur Manipulation der angezeigten Drehzahl hat zum Beispiel folgende Form:

```
01 FC 08 07 47 4C C1 70 00 45 48
```

Dabei ist „`01 FC`“ die ID der adressierten ECU ist, „`08`“ die Länge der folgenden Daten festlegt und somit „`07 47 4C C1 70 00 45 48`“ die eigentlichen Daten der Nachricht inkl. Prüfziffer sind. „`07 47`“ steht dabei für die konkrete Drehzahl. Auch der Parkassistent und das „collision prevention system“, welches das Fahrzeug bei Gegenständen im Fahrtweg zum Halt bringen kann, werden über „normale“ CAN-Befehle gesteuert. Beide Systeme schalten sich jedoch ab, falls sie widersprüchliche Befehle empfangen, sodass zum Beispiel ein einfacher Bremsbefehl das Fahrzeug nicht stoppt. Dies wird umgangen, indem die eigentlichen Systeme, die die korrekten CAN-Befehle versenden, über den Diagnosemodus abgeschaltet werden. Diagnostische CAN-Befehle können darüber hinaus beispielsweise den Motor abschalten oder die Bremsen außer Kraft setzen. Somit sind diese trotz der niedrigen Geschwindigkeiten, während denen diese akzeptiert werden, sehr gefährlich.

2.4 Sicherheitsmaßnahmen

Die Literatur beschäftigt sich nicht nur mit dem Auffinden und Ausnutzen von Sicherheitslücken, es werden stets auch mögliche Wege beschrieben, diese zu schließen. Die Autoren von [**?**] bemängelten, dass Interfaces der ECUs meist viel ausführlicher und offener sind als notwendig. Zum Beispiel ist es nicht unbedingt erforderlich, Bluetooth-Kopplungen ohne Interaktion, oder unbemerkte Datenverbindungen zum Auto zuzulassen. PassThru (Werkstätten-) Geräte sollten über verschlüsselte Verbindungen arbeiten, um das Interface intransparenter zu machen.

Außerdem wurden viele Attacken erst möglich, da auf den ECUs *telnetd*, *ftp* und *vi* vorinstalliert waren, ohne dass sie im gewöhnlichen Betrieb benötigt werden. In der von [**?**] untersuchten

TCU waren die Dienste ebenfalls aktiviert, die Telematikeinheit war durch eine aktive WAN-Verbindung (via 2G-Modem) sogar von Google und Shodan indiziert und somit sehr offen zugänglich.

Da sich Technik ständig ändert, wird es nie 100% sichere Schnittstellen geben, schreiben die Autoren von [**?**] Allerdings müssten sich parallel zur Technik auch die Sicherheitsmaßnahmen, um sie zu schützen, weiterentwickeln. Man könne außerdem den CAN-Zugriff einschränken, beispielsweise bestehe kein Bedarf für das Bluetooth-Modul, Nachrichten auf den CAN-Bus zu schicken.

Eine Verschlüsselung der Nachrichten auf dem CAN-Bus selbst stellt keine große Verbesserung der Sicherheit dar: Die Nachrichtenschlüssel müssten auf den ECUs liegen und könnten wie sätmliche andere Daten extrahiert werden.

Eine vielversprechendere Idee wäre eine grundlegende Neustrukturierung der Netzwerkarchitektur im Auto. In Modellen, bei denen die Forscher mehr Aufwand betreiben mussten, um von außen Zugriff auf sicherheitskritische Komponenten zu bekommen, sind ECUs mit ausgehenden Verbindungen meist nicht direkt mit diesen verbunden, sondern über eine Brücke. Auch die Brücke kann überwunden werden, allerdings steigt dadurch die Komplexität einer Attacke durch diese zusätzliche Hürde enorm und kann deren Erfolg in manchen Fällen sogar verhindern.

Auch die Autoren von [**?**] weisen auf Netzwerkbrücken als kritische Angriffskomponenten hin. In ihrem Versuchsauto konnten sie eine Methode entwickeln, um durch die Telematik-Einheit vom Niedriggeschwindigkeitsnetz des Autos das Hochgeschwindigkeitsnetz anzugreifen. Da *sämtliche* ECUs direkt oder indirekt an das Hochgeschwindigkeitsnetz angeschlossen werden, könnte es schwierig werden, *alle* Geräte und deren Verbindungen gegen diese Art von Angriffen zu sichern.

Außerdem wird in [**?**] auf die offenen Schnittstellen der Diagnose- und Reflashing-Services eingegangen. Eine generelle Restriktion nach der Auslieferung würde die Sicherheit zunächst erhöhen, jedoch bliebe Werkstätten und Besitzern, die an ihrem Auto Komponenten nachrüsten wollen, der Zugriff ebenfalls verwehrt. Die Einführung von Zugriffskontrollen sei schwierig: Welchen Werkstätten soll Zugriff gewährt werden, welchen nicht? Und auf welcher Basis sollten jene Entscheidungen gefällt werden?

Das Nachrüsten von Drittkomponenten nehme laut [**?**] eine weitere entscheidende Position in der Diskussion um IT-Sicherheit ein: Solange sich diese an den Bus verbinden lassen, könne schädliche Software jederzeit über selbstentwickelte Komponenten eingeschleust werden. Ein generelles Verbot von Drittkomponenten sei unvorstellbar, daher lautet der Vorschlag hier, die Geräte über einen Filter an den Bus zu schließen, der nur den Verkehr von autorisierten Paketen zulässt.

Ein simple, jedoch umso wichtigere Maßnahme wäre ein Angreifer-*Erkennungs*mechanismus im CAN-Netzwerk. Miller und Valasek schlagen in [**?**] folgende zwei Möglichkeiten vor: Einerseits könne man die gewöhnliche Rate der Nachrichten, die eine ECU sendet, drastisch erhöhen. Würde ein Angreifer nun über diese ECU eigene Nachrichten versenden (in der Praxis werden diese in einer noch höheren Rate als der regelmäßige Verkehr abgeschickt), wäre ein Angriff deutlich zu erkennen. Durch Erkennung und entsprechende Alarmsignale wird die Angriffsfläche des Automobils zwar nicht kleiner, jedoch könnte das Wissen über einen Angriff ein erster Schritt zu erhöhter Sicherheit sein. Die Autoren von [**?**], die mittels der nachrüstbaren TCU und einer SMS einen modifizierten Update-Prozess starten konnten, schlagen ebenfalls diverse

Maßnahmen vor, um die digitale Angriffsfläche auf Automobile zu verkleinern: Der Code für Updates sollte beispielsweise vom Hersteller signiert werden und nur dann ausgeführt werden, wenn er als authentisch anerkannt wird. Außerdem fanden die Forscher zwar eine Black- und Whitelist für SMS-Absender in der TCU vor, diese könne allerdings (zum Beispiel durch Fälschen der Telefonnummer) leicht umgangen werden und nicht zuletzt gelang es den Forschern ebenfalls, diese Listen zu bearbeiten. Sie schlagen daher vor, die SMS-Administration entweder vollkommen zu deaktivieren, oder durch umfassende Authentifizierung sicherzustellen, dass Update-SMS tatsächlich vom Hersteller kommen, um auch diese Schnittstelle Angreifern möglichst unzugänglich zu machen.

Als weitere Sicherheitslücke wird die Auslieferung der TCUs inklusive öffentlichem *und* privatem Schlüssel für das SSH-Benutzerkonto auf dem Update-Server mit Administratorrechten genannt. Die lokale Speicherung des privaten Schlüssels sei nicht nötig und dadurch könne ihn jeder, der sich Zugriff auf das Dateisystem des Gerätes verschafft, lesen. Den Zweck einer Selbst-Authentifizierung des Gerätes auf dem Update-Server konnten die Autoren ohnehin nicht erkennen - wenn dies tatsächlich notwendig sei, sollte es jedoch mit einem für jedes Gerät individuellen Schlüssel vorgenommen werden.

3 Schluss/Fazit

Das Kompromittieren der Systeme eines Autos ist zusammenfassend nicht nur möglich, sondern auf diverse Wege erreichbar, in vielen Fällen sogar ohne vorzeitiges Bemerken der Insassen. Die in der Literatur erzielten Ergebnisse reichen von der Manipulation der angezeigten Tacho-Geschwindigkeit über Änderung der Musikwiedergabe-Lautstärke, Auslösen der Bremsen bis zur kompletten Abschaltung des Motors. All dies kann ohne physischen Kontakt zum manipulierenden Auto geschehen.

Möglich wird dies einerseits über die (historisch gewachsene) Kommunikation der Systeme über den CAN-Bus, andererseits durch ungewissenhafte Implementation von ECUs beziehungsweise deren Schnittstellen oder schlicht durch fehlende Sicherheitsmaßnahmen wie Authentifzierung und Sicherstellen der Nachrichtenintegrität.

In jedem Fall muss die Autoindustrie ihren Fokus mehr in Richtung IT-Sicherheit verschieben. Die Einhaltung grundlegender und weit verbreiteter Sicherheitsstandards könnte bereits viel zum Schutz vor Malware beitragen. Zum Beispiel sollte niemals ein für alle Fahrzeuge des selben Modells gültiger SSH-Private-Key auf den jeweiligen Fahrzeugen mitgeliefert werden. Der Einsatz des CAN-Busses, dessen Technologie vor über 30 Jahren entwickelt wurde, könnte ebenfalls überdacht werden. Stattdessen wäre eine andere Netzwerktopologie als Weiterentwicklung möglich, in der nicht über Broadcast kommuniziert wird, sondern die Nachrichten direkt an die empfangende Komponente adressiert werden. Das Sicherstellen der Nachrichtenintegrität könnte beispielsweise über einen Nachrichtenauthentifizierungscode („MAC“) realisiert werden.

Die Software auf den TCUs müsste besser getestet und nach außen hin abgeschirmt werden. Es sind viele Nachlässigkeiten zu erkennen, zum Beispiel sollten einzelne Komponenten niemals über offene Ports für Angreifer von außerhalb direkt erreichbar sein. Auch müssten mitgelieferte Passwörter oder Hashes mit einem nicht nachvollziehbaren Zufallsalgorithmus erstellt werden.

Der Trend der Automatisierung in der Automobilbranche ist noch lange nicht gestoppt. Daher bleibt zu hoffen, dass in den kommenden Jahren Nachlässigkeiten behoben werden und die Schließung von Sicherheitslücken in Angriff genommen wird.