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

12. Dezember 2015

Zusammenfassung

Für den eiligen Leser sollen auf etwa einer halben, maximal einer Seite die wichtigsten Inhalte, Erkenntnisse, Neuerungen bzw. Ergebnisse der Arbeit beschrieben werden.

Durch eine solche Zusammenfassung (im engl. auch Abstract genannt) am Anfang der Arbeit wird die Arbeit deutlich aufgewertet. Hier sollte vermittelt werden, warum der Leser die Arbeit lesen sollte.

Inhaltsverzeichnis

1 Einleitung

Sicherheit im Auto - ein Thema, dessen Relevanz kaum geleugnet werden kann. Hunderte Millionen Automobile verkehren auf den Straßen unseres Planeten, zwangsläufig ergeben sich hierbei gefährliche Situationen. Nicht zuletzt um Autos sicherer zu gestalten, wurden in der Vergangenheit bereits zahlreiche computergestützte Systeme entwickelt, die anfangs vor allem das Ziel hatten, den Fahrer in Notsituationen bei der Handhabung des Fahrzeuges unterstützen (wie beispielsweise das 'Elektronische Stabilitätsprogramm' ESP oder Mercedes' Intelligent Drive). Angetrieben von der Digitalisierung in vielen Bereichen unseres Alltags folgten weitere Systeme, komplexe Infotainment-Systeme, in das Auto integrierte Navigationsgeräte, oder gar Einpark- und Spurhalteassistenten, die dem Nutzer vor allem höheren Komfort bieten sollen. Doch auch hier ist oftmals eine sehr enge Kopplung an sicherheitskritische Komponenten (Bremsen, Lenkung, etc.) gegeben.

Betrachtet man die Kommunikation zwischen den digitalen Systemen und der physischen Hardware eines Autos, so fällt schnell auf, dass die meisten Botschaften unverschlüsselt über ein Bus-System ausgetauscht werden. Die Literatur zeigt, dass durch den Zugriff auf den Bus häufig auch an dere Komponenten erreicht werden können und sich die meisten Komponenten eines Automobils somit von außen, oft vollkommen unbemerkt, steuern lassen können. Die Auswirkungen auf die Sicherheit der Insassen liegen auf der Hand. ...

2 Hauptteil

2.1 CAN-Bus

2.1.1 Funktion und Aufbau

Alle im folgenden erörterten Probleme ergeben sich aus Schwachstellen in der Zentralen Kommunikationsschnittstelle eines Autos, dem sogenannten CAN-Bus.

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 gebundenen Systeme werden Electronic Control Units genannt, kurz ECUs.

Die übliche Funktionsweise eines CAN-Busses lässt sich gut am Beispiel des Toyota Prius erklären [**?**]: ECU's 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.

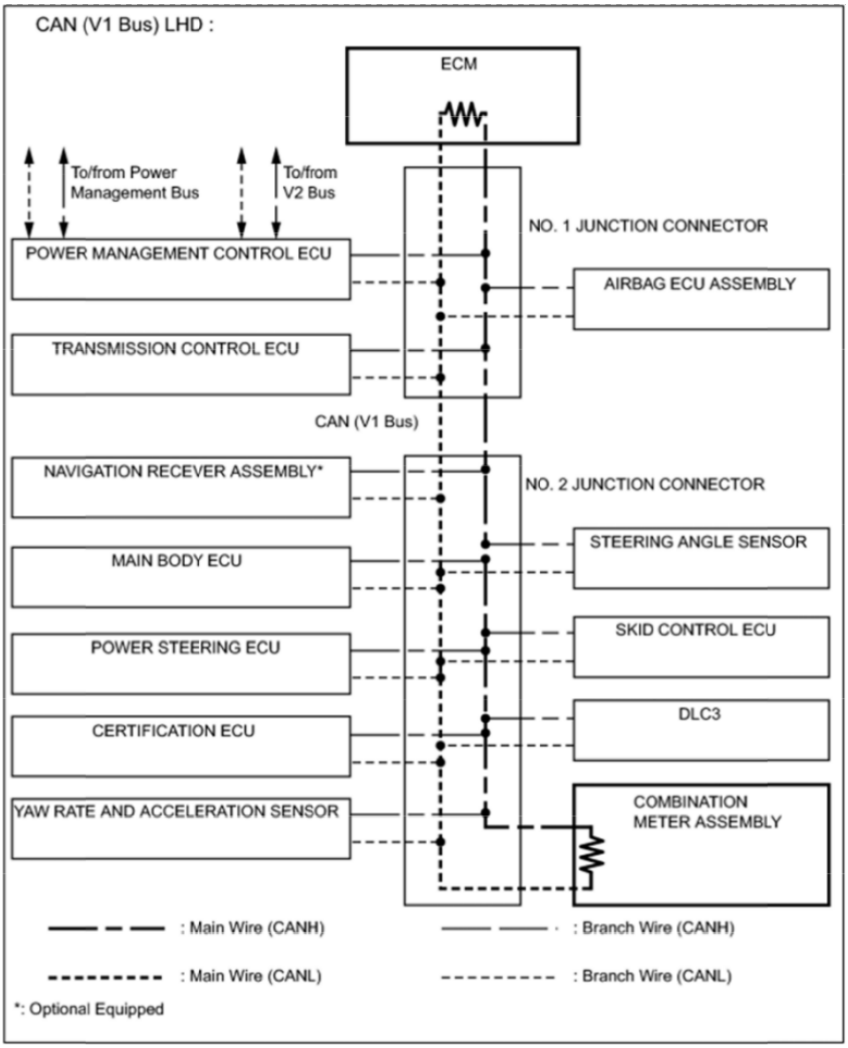

Abbildung 2.1: Aufbau eines CAN-Busses [**?**]

Hier 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 Payload, also der relevanten Daten oder dem Befehl. In diesem Falle sind es 8 Bytes, also die Sequenz (00 00 00 00 3C 18 C0 FF).

Obiger Befehl würde sich auch in folgender Schreibweise darstellen lassen können, auch wenn hierbei durch die fehlenden Bytes ein potentieller Informationsverlust auftritt:

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

In diesem Paket wird lediglich eine Payloadlänge von 4 definiert. Um Nachrichtenintegrität sicherzustellen, wird häufig eine Prüfsumme erstellt. Diese wird üblicherweise als letztes Byte der Payload 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 Payloadlänge von 08. Die Payload eines solchen Befehls hat die Struktur:

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

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

$$\text{Speed}(mph) = 0.0065 * (CCDD) - 67$$
$$RPM = .25 * (AABB) - 24$$

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 Checksum hinzugenommen wird, nach der Formel:

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

Dementsprechend ist der letzte Byte *CC*:

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

Nach diesem Beispiel sollte ersichtlich sein, dass es relativ einfach ist ein CAN-Bus Paket zu erstellen. Da alle Geräte am selben Bus sitzen und jedes Gerät an alle ECU's sendet, ergibt sich daraus eine relativ drastische Problematik. Für den Fall, dass es einem Angreifer gelingt Zugang zu einem der ECU's am Bus zu bekommen, ist er in der Lage Befehle an alle Geräte am Bus zu senden, eingeschlossen kritische Systeme, wie Bremsen, Motor oder Fahrhilfe.

Im folgenden Abschnitt wird auf die Schwachstellen und die Unsicherheit des CAN-Busses eingegangen.

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, Pakete zu designen, welche von den ECUs als valid angesehen werden, allerdings kann ein Angreifer durch die Broadcast-Architektur des CAN-Busses beliebig viele hiervon ohne Probleme für eine spätere Analyse mitschneiden. Selbstverständlich braucht man hierfür Zugriff auf den Bus selbst, die Analyse kann aber z.B. auch an einem baugleichen Modell geschehen, welches der Angreifer besitzt oder mietet. Da der abgehörte Traffic auf dem Bus unverschlüsselt ist, kann der Mittschnit nun benutzt werden um die API der ECUs zu rekonstruieren: Hierfür muss lediglich nach Mustern im Paketstrom gesucht werden, welche in Abhängigkeit zu dem

Zustand des Auto stehen. Durch dieses Verfahren wurde beispielsweise die vorher 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 Kompromitierbarkeit von an dem CAN-Bus befindlichen Geräten zunehmend.

2.2 Ausgewählte Angriffsvektoren

Nicht alle Systeme, TCUs[1] oder ECUs[2], die über den CAN-Bus kommunizieren, sind als Angriffsvektor geeignet. Entscheidend für einen Angriff aus der Ferne (ohne pyhsischen Zugriff) ist eine Verbindung zur Außenwelt. 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 den Standards 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. Mittels eines via Bluetooth gekoppelten Gerätes gelang es den Forschern wegen Sicherheitslücken der Bluetooth-Stack-Implemenatation, 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

1. '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.)
2. 'Electronic Control Units': Ähnlich wie TCUs, allerdings können TCUs 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

dar, mittels Brute-Force-Methode gelang es den Forschern teilweise innheralb 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 bspw. ein 3G-Modem kann das Automobil so mit dem Internet verbunden sein, oder im Falle eines schweren Unfalls selbständig eine SMS an Rettungskräfte senden.

2.2.3 Nachrüstbare TCUs

Zusätzlich zu den unterschiedlichen Medien- und Diagnosesystemen, die bereits in die Fahrzeuge durch den Hersteller integriert sind, gibt es verschiedenste nachrüstbare Gerätschaften, die dazu dienen können, den Funktionsumfang eines Automobils zu erweitern (komplexere Mediennutzung 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 beim Hersteller kommunizieren zu können, sei es um z.B. 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 (Bspw. Progressives *Snapshot*). Andere TCUs zielen direkt darauf ab, das Fahrverhalten zu verbessern, indem sie über Smartphones der Fahrzeughalter in Echtzeit z.B. Hinweise für eine motor- und benzinschonenderes Beschleunigungs- und Bremsverhalten geben (Bspw. Automatic Labs *Automatic*. Häufig finden TCUs auch im Flottenmanagement Anwendung, wo sie zur Lokalisierung und Überwachung der Geschäftfahrzeuge dienen. Delphis *Connect* 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 jugendlichen 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 Herausfinden der Telefonnummer der TCU, die sie für Mobilfunkdienste besaß, selbst mit einer simplen SMS einen Update-Vorgang an einem beliebigen Server anzustoßen. Aufgrund einer nicht vorhandenen Authentifizierung des Servers und Signatur der Updates konnten einfach beliebige eigene Dateien auf die TCU heruntergeladen werden und mit entsprechenden Befehlen ausgeführt werden. Im Endeffekt erlangten die Angreifer so die Möglichkeit, sich mit Root-Zugriff 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.
Ähnliche Fehler bei dem Entwurf derart sicherheitsrelevanter Systeme stellen also eine große Gefahr für den Fahrzeughalter dar. Dies wird noch dadurch verstärkt, dass der Fahrzeughersteller selbst nicht in der Lage ist, diese Fehler zu beheben, da es sich um fremde Systeme handelt, hier muss der jeweilige Hersteller selbst aktiv werden und dafür sorgen, dass alle sich im Umlauf befindenen TCUs eventuelle Updates erhalten.

2.3 Beispiele

2.3.1 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 Kollisionswarnungssystem, 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 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.

Miller und Valasek konzentrieren sich zuerst auf Schwächen der eingebauten Hotspot–Funktionalität. Sofern der Fahrzeughalter ein Abonnements beim Hersteller abschließt, bietet das Fahrzeug an, einen Wlan-Hostpot für die Fahrzeuginsassen bereitzustellen. In dieses Netzwerk einzudringen benötige im Allgemeinen ein übliches Vorgehen beim Einbrechen in Wlan-Netze, Miller und Valasek konnten aber Schwächen bei der zufälligen Generierung der Netzpasswörter entdecken; das Passwort wird stets als Funktion eines Zeitpunkts berechnet, dieser kann zusätzlich stark eingeschränkt werden, da sich das System, so lange keine Uhrzeit über GPS oder Mobilfunk empfangen wurde, auf eine Standarduhrzeit einstellt, und dies immer eine gewisse Zeit nach Systemstart benötigt, um sich zu aktualisieren.

Mit Verbindung zum Wlan-Netzwerk konnten Miller und Valasek mithilfe üblicher Netzwerkanalysetools diverse offene Ports identifizieren, die sich zum Angreifen anbieten. Insbesondere der Port 6667 erwies sich als interessant, da hier nicht, wie von der Port-Nummer zu erwarten, ein IRC-Server lief, 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 Hauptangriffspunkt des Systems.

Über diese Lücke im System erwies es sich als trivial Code auszuführen, mit sehr simplen Skripten kann ein Angreifer nun u.a. GPS-Daten auslesen, um den Standort des Fahrzeugs festzustellen, er kann die Geschwindigkeit der Lüfter verändern, Lautstärke (Beispielcode s.u.) und Bass des Radios verändern, zusätzlich den Radiosender ändern, bestimmen, was auf dem integrierten Display erscheint und eigene Bilder rüberspielen.

```
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. U.a. 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 das Fahrzeug stets eine IP-Adresse in einem von zwei Adressbereichen zugeordnet bekommt 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 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 nun 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 dazu in der Lage ist. Zusätzlich kann dieser Chip über das Mediensystem geupdatet werden. Dies nutzen Miller und Valasek aus: Sie laden über das Multimediasystem eine eigene Firmware auf besagten Chip, welche es ihnen ermöglicht CAN-Befehle auf die CAN-Netzwerke zu versenden. Dieses Update birgt dabei ein Risiko für den 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 von Werkstätten genutzt werden, daher auch nur bei geringer Geschwindigkeit von den ECUs akzeptiert werden. Im Gegensatz zu CAN-Befehlen anderer Hersteller nutzt der Jeep keine herkömmlichen Algorithmen, um Prüfziffern zu berechnen, sondern einen eigenen, etwas komplizierteren (Details siehe [**?**]), welcher von Miller und Valasek aber durch Reverse-Engineering herausgefunden werden konnte. Mit diesem Algorithmus waren Miller und Valasek nun endgültig in der Lage eigene CAN-Befehle zu erstellen.

Über ”normale” CAN-Befehle waren sie z.B. in der Lage den Blinker des Autos zu setzen, die Türen zu öffnen und zu schließen und die Drehzahl, die auf dem Tachometer angezeigt wird, zu manipulieren. Bis auf die Berechnung der Prüfsumme sehen die CAN-Befehle ähnlich aus, wie zuvor in Abschnitt 2.1 beschrieben, erwähnter Befehl zur Manipulation der angezeigten Drehzahl sieht z.B. wie folgt aus:

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

Wobei ”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. ”07 47” steht dabei für die konkrete Drehzahl.

2.4 Sicherheitsmaßnahmen

Interfaces der ECUs sind meist viel ausführlicher und offener als notwendig. Z.B. ist es unnötig, 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 notwendig wären. In der von [**?**] untersuchten TCU waren die Dienste ebenfalls aktiviert und das Gerät 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 müssen 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 da: Die Nachrichtenschlüssel würden auf den ECUs lägen und könnten wie bisher 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 die Komplexität einer Attacke durch diese zusätzliche Hürde enorm und kann in manchen Fällen die Attacke 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 hier auf die offenen Schnittstellen der Diagnose- und Reflashing-Services eingegangen. Eine generelle Restriktion vor 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?

Die Nachrüstung von Drittkomponenten nehme eine besondere Position in der Diskussion um IT-Sicherheit ein: Solange sich diese an den Bus verbinden lassen, könne schädliche Software jederzeit über selbstentwickelte Komponenten eingschleust werden. Ein generelles Verbot von Drittkomponenten sei unvorstellbar, daher lautet der Vorschlag hier, die Geräte über einen Filter an Bus zu schließen, das nur den Verkehr von authorisierten Paketen zulässt.

Ein simple, doch umso wichtigere Maßnahme wäre ein Angreifer-*Erkennungs*mechanismus im CAN-Netzwerk. Miller und Valasek schlagen 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), 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 mehr 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 (z.B. durch Fälschen der Telefonnummer) leicht umgangen werden und nicht zuletzt gelang es den Forschern ebenfalls, diese Listen zu bearbeiten. Sie schlagen 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 mit einem für jedes Gerät individuellen Schlüssel vorgenommen werden.

3 Schluss/Fazit

In Anbetracht der zahlreichen Sicherheitslücken in modernen Automobilen stellt sich heraus, dass es keineswegs unmöglich ist die Systeme eines Autos zu kompromittieren. Hin man die fatalen Ergebnisse (z.B. Auslösen oder gar Ausschalten der Bremsen, Abschalten des Motors), die sogar ohne jeden physischen Kontakt zum Auto möglich sind,

Nichtsdestotrotz wird uns erneut aufgezeigt wie angreifbar die von uns benutzte Hardware ist und es stellt sich die Frage, ob in Zukunft die Sicherheit in Automobilen eine größere Rolle spielen wird. Speziell im Ausblick auf autonom fahrende Fahrzeuge.

In jedem Falle muss die Autoindustrie ihren Fokus mehr in Richtung IT-Sicherheit bewegen. Es müssen hier lediglich einfache Sicherheitsstandards eingehalten werden, welche bereits weit verbreitet sind. Zum Beispiel sollte niemals ein für alle Fahrzeuge des selben Modells gültiger SS Private-Key auf den jeweiligen Fahrzeugen mitgeliefert werden. Ebenfalls ist die Technologie des CAN-Busses überholt und sollte durch ein einfaches Addressierungsverfahren und eine Authentifizierung für neue Geräte um ein vielfaches schwieriger zugänglicher gemacht werden.

Die Software auf den TCUs selbst müsste besser getestet und nach außen hin abgeschirmt werden. Es gibt zu viele Nachlässigkeiten, so sollte zum Beispiel niemals die D-Bus Software, welche als Hilfsmittel für interne Prozesskommunikation ein kritischer Angriffspunkt ist, über einen Port von außerhalb direkt erreichbar sein. Auch müssen mitgelieferte Passwörter oder Hashes mit einem nicht nachvollziehbaren Zufallsalgorithmus erstellt werden.

Diese Änderungen sind nicht kompliziert oder Hightech-Problematiken. Es sind ganz einfach Fahrlässigkeiten des Herstellers, der auf IT-Security keinen Wert legt.

Es bleibt zu hoffen, dass diese Nachlässigkeit niemals zu größeren Problemen führt.