

Seminar Kognitive Automobile

Sicherheit in Kognitiven Automobilen

Seminararbeit
von

Martin Thoma

Fakultät für Informatik
Institut für Anthropomatik
und
FZI Forschungszentrum Informatik

Betreuender Mitarbeiter: Dipl.–Inform. Ralf Kohlhaas

Sommersemester 2015

Kurzfassung

Moderne Automobile verfügen über eine Vielzahl von Assistenz- und Fahrsicherheitssystemen. Diese Systeme haben Schnittstellen, welche das Ziel von Angriffen sein können. In dieser Seminararbeit wird der aktuelle Stand der IT-Sicherheit kognitiver Automobilität untersucht. Dabei wird auf mögliche Angriffe sowie Möglichkeiten zum Schutz eingegangen.

Inhaltsverzeichnis

1	Einleitung	1
2	Standards und Verordnungen	3
3	Angriffe	5
3.1	CAN-interne Angriffe	5
3.2	CAN-externe Angriffe	5
3.3	Buffer Overflow Angriffe	6
3.4	Sicherheitslage in Automobilen bis August 2015	7
4	Verteidigungsmaßnahmen	9
4.1	Buffer Overflows	9
4.2	Code-Qualität	10
A	Literaturverzeichnis	11

1. Einleitung

Kognitive Automobile sind, im Gegensatz zu klassischen Automobilen, in der Lage ihre Umwelt und sich selbst wahrzunehmen und dem Fahrer zu assistieren oder auch teil- bzw. vollautonom zu fahren. Diese Systeme benötigen Zugriff auf Sensoren und Aktoren, um ihre Aufgabe zu erfüllen. So benötigt ein Auto mit Antiblockiersystem beispielsweise die Drehzahl an jedem Reifen und die Möglichkeit die Bremsen zu beeinflussen; für Einparkhilfen werden Sensoren benötigt, welche die Distanz zu Hindernissen wahrnehmen sowie Aktoren, die das Auto lenken und beschleunigen können. Weitere dieser Systeme sind Spurhalteassistent, Spurwechselassistent und Fernlichtassistent.

Als immer mehr elektronische Systeme in Autos verbaut wurden, die teilweise sich überschneidende Aufgaben erledigt haben, wurde der CAN-Bus entwickelt [16]. Über ihn kommunizieren elektronische Steuergeräte, sog. *ECUs* (engl. *electronic control units*). Diese werden beispielsweise für ABS und ESP eingesetzt.

Der folgende Kapitel geht auf Standards wie den CAN-Bus und Verordnungen, die in der Europäischen Union gültig sind, ein. In Kapitel 3 werden Angriffsziele und Grundlagen zu den Angriffen erklärt, sodass in Kapitel 4 mögliche Verteidigungsmaßnahmen erläutert werden können.

2. Standards und Verordnungen

Für den Automobilbereich existieren viele Standards und Verordnungen. In diesem Kapitel wird eine Auswahl vorgestellt, die Fahrzeuge der Klassen M_1 und N_1 betrifft. Das sind Fahrzeuge zur Personenbeförderung „mit mindestens vier Rädern und höchstens acht Sitzplätzen außer dem Fahrersitz“ sowie „für die Güterbeförderung ausgelegte und gebaute Kraftfahrzeuge mit einer zulässigen Gesamtmasse von 3,5 Tonnen“[7].

In der EU wurde mit [8] die OBD-Schnittstelle verpflichtend für Fahrzeuge der Klasse M_1 und N_1 mit Fremdzündungsmotor ab 1. Januar 2004. Die EU-Direktive führt weiter die in der ISO DIS 15031-6 Norm aufgeführten Fehlercodes als Minimalstandard ein. Diese müssen „für genormte Diagnosegeräte [...] uneingeschränkt zugänglich sein“. Außerdem muss die Schnittstelle im Auto so verbaut werden, dass sie „für das Servicepersonal leicht zugänglich [...] ist“.

Der Software-Zugang ist mit J2534 der Society of Automotive Engineers standardisiert [15]. Dieser Standard stellt sicher, dass unabhängig vom OBD-Reader Diagnosen über das Auto erstellt und die ECUs umprogrammiert bzw. mit Aktualisierungen versorgt werden können.

Um die Daten bereitzustellen, werden verschiedene elektronische Komponenten über den CAN-Bus vernetzt. Dieser ist in ISO 11898 genormt.

Weiterhin wurde in der EU mit [9] beschlossen, dass ab 1. November 2012 alle PKWs für Neuzulassungen ein System zur Reifendrucküberwachung (engl. *tire pressure monitoring system*, kurz *TPMS*) besitzen müssen. Ab 1. November 2014 müssen alle Neuwagen ein solches System besitzen. Da sich die Räder schnell drehen ist eine kabelgebundene Übertragung der Druckmesswerte nicht durchführbar. Daher sendet jeder Reifen kabellos ein Signal, welches von einem oder mehreren Sensoren im Auto aufgenommen wird.

Mit [10] wird für Fahrzeuge, die ab dem 31. März 2018 gebaut werden das eCall-System, ein elektronisches Notrufsystem, verpflichtend. Dabei müssen dem eCall-System „präzise[-] und verlässliche[-] Positionsdaten“ zur Verfügung stehen, welche über das globale Satellitennavigationssystem Galileo und dem Erweiterungssystem EGNOS geschehen soll. eCall soll über öffentliche Mobilfunknetze eine „Tonverbindung zwischen den Fahrzeuginsassen und einer eCall-Notrufabfragestelle“ herstellen können. Außerdem muss ein Mindestdatensatz übermittelt werden, welcher in DIN EN 15722:2011 geregelt ist. Diese Funktionen müssen im Fall eines schweren Unfalls automatisch durchgeführt werden können.

3. Angriffe

Eine Reihe von elektronischen Systemen wurde zum Diebstahlschutz entwickelt [26, 30, 14]. Allerdings passen sich auch Diebe an die modernen Gegebenheiten, insbesondere Funkschlüssel [18], an. Außerdem gehen diese Systeme von einem klassischen Angreifer aus, der sich ausschließlich auf der Hardware-Ebene bewegt.

Im Folgenden werden zunächst Möglichkeiten von netzwerk-internen Angreifer, d.h. Angreifern welche physischen Zugang zum CAN-Bus haben, aufgelistet. Es folgt eine Beschreibung wie Angreifer ohne direkten physischen Zugriff auf das Auto sich mit dem CAN-Bus verbinden können. Viele Angriffe nutzen sogenannte Buffer Overflows aus. Diese werden in Abschnitt 3.3 erklärt. Abschließend folgt ein Abschnitt über konkrete, der Öffentlichkeit bekannt gewordene Sicherheitslücken.

3.1. CAN-interne Angriffe

Koscher et al. haben in [17] zwei nicht näher spezifizierte Autos der selben Marke und des selben Modells untersucht. Sie waren in der Lage über den CAN-Bus etliche Funktionen des Autos, unabhängig vom Fahrer, zu manipulieren. Das beinhaltet das deaktivieren und aktivieren der Bremsen, stoppen des Motors, steuern der Klimaanlage, Heizung, Lichter, Manipulation der Anzeigen im Kombiinstrument sowie das Öffnen und Schließen der Schlösser. Durch moderne Systeme wie eCall kann der Angreifer sich sogar einen Kommunikationskanal zu dem Auto aufbauen. Dies setzt allerdings voraus, dass der Angreifer sich bereits im auto-internen Netzwerk befindet.

3.2. CAN-externe Angriffe

In [5] wurde an einer Mittelklasselimosine mit Standardkomponenten gezeigt, dass der Zugang zum auto-internen Netzwerk über eine Vielzahl an Komponenten erfolgen kann. So haben Checkoway et al. CD-Spieler, Bluetooth und den OBD-Zugang als mögliche Angriffsvektoren identifiziert.

Bei dem Angriff über den Media Player haben Checkoway et al. die Tatsache genutzt, dass dieser am CAN-Bus angeschlossen ist und die Software des Mediaplayers über eine CD mit einem bestimmten Namen und Dateiformat aktualisiert werden kann. Außerdem wurde ein Fehler beim Abspielen der Audio-Dateien genutzt um einen Buffer Overflow zu erzeugen. Es wurde gezeigt, dass dieser genutzt werden kann um die Software des Media-Players zu aktualisieren. Dafür muss nur eine modifizierte Audio-Datei, welche immer noch abspielbar ist, auf der CD sein.

Der von Checkoway et al. durchgeführte Angriff via Bluetooth benötigt ein mit dem Media Player verbundenes Gerät. Dieses nutzt dann strcpy-Befehle, bei denen die Puffergrößen nicht

überprüft wurden um bei der Bluetooth-Konfiguration aus um beliebigen Code auf der Telematik-Einheit des Autos ausführen zu können. Daher ist das Smartphone des Autobesitzers ein Angriffsvektor.

Die Bluetooth-Verbindung kann jedoch auch ohne ein verbundenes Gerät für Angriffe genutzt werden. Dazu muss der Angreifer genügend Zeit in der Nähe des Autos verbringen um die Bluetooth-MAC-Adresse zu erfahren. Damit kann er ein Anfrage zum Verbindungsaufbau starten. Diese müsste der Benutzer normalerweise mit der Eingabe einer PIN bestätigen. Checkoway et al. haben für ein Auto gezeigt, dass die Benutzerinteraktion nicht nötig ist und die PIN via Brute-Force, also das Ausprobieren aller Möglichkeiten, innerhalb von 10 Stunden gefunden werden kann. Allerdings kann dieser Angriff parallel ausgeführt werden. Es ist also beispielsweise möglich diesen Angriff für alle Autos in einem Parkhaus durchzuführen. Bei einem parallel durchgeführten Brute-Force-Angriff ist der erste Erfolg deutlich schneller zu erwarten.

Die standardisierte und von Automechanikern zu Diagnosezwecken genutzte OBD-Schnittstelle stellt einen weiteren Angriffspunkt dar. Für die verschiedenen Marken gibt es Diagnosewerkzeuge, wie z.B. NGS für Ford, Consult II für Nissan und der Diagnostic Tester von Toyota. Diese dedizierten Diagnosegeräte werden allerdings über PCs mit Aktualisierungen versorgt. Modernere Diagnosewerkzeuge sind nicht mehr bei der Diagnose vom PC getrennt, sondern werden direkt, über ein Kabel, W-LAN oder Bluetooth, mit einem PC verbunden. Daher stellt die Diagnose- und Aktualisierungstätigkeit von Automechanikern einen weiteren Angriffsvektor dar. Wenn der Mechaniker ein Diagnosegerät benutzt, welches ein W-LAN aufbaut, so können Angreifer sich mit diesem verbinden und selbst Aktualisierungen durchführen. Außerdem wurde von Checkoway et al. gezeigt, dass auch das Diagnosegerät selbst so manipuliert werden kann, dass es automatisch die gewünschten Angriffe ausführt.

Wie in Kapitel 2 beschrieben wird eCall-System ab 2018 in Europa verpflichtend eingeführt. Dieses nutzt das Mobilfunknetz zur Kommunikation. Checkoway et al. haben gezeigt, dass Telematik-Einheiten von außerhalb des Autos angewählt werden und die Software auf diese Weise aus beliebigen Entfernungen aktualisiert werden kann. Dazu wurden zahlreiche Schwachstellen der Telematik-Einheit von Airbiquitys Modem aqLink genutzt. Dieses Modem wird unter anderem von BMW und Ford eingesetzt [1, 2].

Ein Angriff auf die Privatsphäre ist mit TPMS möglich. In [23] wurde gezeigt, dass TPMS-Signale zur Identifikation von Autos genutzt werden können. Die Identifikation eines vorbeifahrenden Autos ist aus bis zu 40 m Entfernung möglich.

Genauso stellt das Mikrofon, welches wegen eCall ab 2018 in jedem Auto sein muss, eine Möglichkeit zum Angriff auf die Privatsphäre dar.

3.3. Buffer Overflow Angriffe

Dieser Abschnitt erklärt anhand eines einfachen Beispiels wie Buffer Overflow Angriffe durchgeführt werden.

Um Buffer Overflow Angriffe zu verstehen, müssen Grundlagen der Struktur eines Prozesses im Speicher bekannt sein. Diese werden im Detail in [25] erklärt.

Buffer Overflow Angriffe nutzen die Tatsache aus, dass bestimmte Befehle wie beispielsweise `gets` Zeichenketten in einen Puffer schreiben, ohne die Größe des Puffers zu beachten. `gets` erhält als Parameter einen Zeiger auf die Startadresse des Puffers. Wenn der Benutzer eine längere Eingabe macht als der Puffer erlaubt, so wird in nachfolgende Speicherbereiche geschrieben. Dies kann an folgendem, aus [3] entnommenem und leicht modifiziertem Beispiel beobachtet werden:

```
simple.c
1  #include <stdio.h>
2
3  int main(void) {
4      char buff[10];
5      int pass = 0;
6
7      printf("Enter password: ");
8      gets(buff);
9
10     if (strcmp(buff, "correct")) {
11         printf("Wrong Password\n");
12     } else {
13         printf("Correct Password\n");
14         pass = 1;
15     }
16
17     if (pass) {
18         printf("Password protected code. pass=%i\n", pass);
19     }
20
21     return 0;
22 }
```

Kompiliert man dieses Programm mit `gcc -O0 -fno-stack-protector -g simple.c -o simple`, so kann mit der Eingabe von 16 Zeichen die Variable `pass` überschrieben werden.

Allerdings ist es nicht nur möglich interne Variablen zu überschreiben, sondern sogar beliebigen Code auszuführen. Dies wird in [21] mit einem sehr ähnlichem Beispiel gezeigt und im Detail erklärt. Dabei wird nicht beliebiger Text in den Puffer geschrieben, sondern sogenannter *Shellcode*. Unter Shellcode versteht man Assemblerbefehle, welche in Opcodes umgewandelt wurden.

3.4. Sicherheitslage in Automobilen bis August 2015

Heutzutage ist nicht nur die Hardware von Automobilen durch Diebe und andere Angreifer gefährdet, sondern auch die Software. Die folgenden Beispiele zeigen, dass Angriffe auf die IT in

Automobilen nicht nur im akademischen Rahmen auf einige wenige spezielle Modelle durchgeführt werden, sondern dass auch modellübergreifende Angriffe möglich sind.

Die Art der Angriffe ist nicht neu. So sind wurde mit dem Morris-Wurm bereits 1988 ein Stack Overflow Angriff durchgeführt [24], Replay-Angriffe wurden 1994 beschrieben [29] und seit Beginn der Entwicklung von Viren werden bekannte Lücken in veralteter Software ausgenutzt. Allerdings ist die Software in kognitiven Automobilen komplexer als in herkömmlichen Automobilen, die Menge der eingesetzten Software ist größer und die Möglichkeiten zur Einflussnahme durch Aktoren sind gewachsen. Daher bieten kognitive Automobile eine größere Angriffsfläche als herkömmliche Automobile.

- 2010 hat ein ehemaliger Angestellter mehr als 100 Autos über ein Fernsteuersystem, welches Kunden an fällige Zahlungen erinnern soll, die Hupen aktiviert [22].
- 2010 wurde mit [17] auf mögliche Probleme in kognitiven Automobilen hingewiesen. Mit [5] wurde 2011 gezeigt, dass mindestens ein Modell in einer bestimmten Konfiguration unsicher ist.
- 2015 wurde von Charlie Miller und Chris Valasek gezeigt, dass das Unterhaltungssystem Uconnect von Fiat Chrysler benutzt werden kann um Autos aus der Ferne zu übernehmen. Wegen dieses Softwarefehlers hat Fiat Chrysler 1,4 Millionen Autos zurückgerufen [11].
- 2015 wurde eine Sicherheitslücke in BMW's ConnectedDrive bekannt. Diese hat es dem Angreifer erlaubt, das Auto zu öffnen [27]. Dieter Spaar hat dabei mehrere Sicherheitslücken aufgedeckt: BMW hat in allen Fahrzeugen die selben symmetrischen Schlüssel eingesetzt, Teildienste haben keine Transportverschlüsselung verwendet und ConnectedDrive war nicht gegen Replay-Angriffe geschützt. Bei Replay-Angriffen nimmt der Angreifer Teile der Kommunikation zwischen dem BMW-Server und dem Auto auf und spielt diese später wieder ab. Dies könnte beispielsweise eine Nachricht sein, die das Auto entriegelt.
- In [31] wurde 2015 gezeigt, dass einige Dutzend Automodelle von Audi, Ferrari, Fiat, Opel, VW und weiterer Marken eine Sicherheitslücke in den Schlüsseln haben, welche es Autodieben erlaubt nach nur zweimaligem Abhören der Kommunikation des Originalschlüssels mit dem Auto eine Kopie des Schlüssels anzufertigen.
- Die Sicherheitsfirma Lookout hat 2015 Fehler in der Software des Tesla Model S gefunden, welche Root-Zugang zu internen Systemen erlaubt hat [19]. Drei der gefundenen sechs Sicherheitslücken sind veraltete Softwarekomponenten.
- 2015 wurde auch ein Angriff auf das OnStar-System von GM bekannt, durch welchen der Angreifer beliebige Autos mit dem OnStar-System öffnen und den Motor starten konnte [28]. Die Art des Angriffs, die bisher noch nicht detailliert beschrieben wurde, betrifft laut [12] auch die iOS-Apps für Remote von BMW, mbrace von Mercedes-Benz, Uconnect von Chrysler und Smartstart von Viper.

4. Verteidigungsmaßnahmen

Wie bereits in Abschnitt 3.4 beschrieben, sind die Arten der Angriffe nicht neu. Daher sind auch die Verteidigungsmaßnahmen nicht spezifisch für den Automobilbereich, sondern allgemeiner softwaretechnischer Art.

Es gibt etliche Automobilhersteller, -marken und -modelle. Für viele Modelle gibt es unterschiedliche Konfigurationen und wiederum zahlreiche Optionen für Zubehör wie beispielsweise das Autoradio oder Navigationssysteme. Dies macht allgemeine Aussagen über die Sicherheit und Verteidigungsmaßnahmen von Automobilen schwierig. Allerdings stellen Standards und Verordnungen sicher, dass Teile der relevanten Infrastruktur in Automobilen gleich sind, sodass Angreifer diese fahrzeugübergreifend nutzen können.

Der CAN-Bus ist eine große Schwachstelle der IT-Sicherheit in Autos. Über ihn müssen viele ECUs kommunizieren und einige, wie das Autoradio, werden nicht als sicherheitskritisch wahrgenommen. Gleichzeitig sind sicherheitskritische ECUs an dem selben CAN-Bus angeschlossen. Daher ist es wichtig die Nachrichten, welche über den CAN-Bus empfangen werden, zu filtern. Die Informationen müssen auf Plausibilität geprüft werden.

Alle von Checkoway et al. beschriebenen Angriffe basieren zum einen auf Reverse-Engineering, also der Rekonstruktion der Software-Systeme und Protokolle, zum anderen auf Fehlern in der Software. Das Reverse-Engineering wurde in einigen Fällen laut Checkoway et al. stark vereinfacht, da Debugging-Symbole in der Software waren. Diese können und sollten entfernt werden.

Außerdem sollten laut Checkoway et al. die Diagnosegeräte Authentifizierung und Verschlüsselung wie beispielsweise OpenSSL nutzen.

4.1. Buffer Overflows

Gegen Buffer-Overflow-Angriffe können zum einen Sprachen wie Java oder Rust verwendet werden, welche die Einhaltung der Bereichsgrenzen automatisch überprüfen. Des Weiteren kann anstelle der C-Funktion `strcpy()` die Funktion `strncpy()` verwendet werden, welche die Anzahl der zu schreibenden Zeichen begrenzt [6].

Ein weiteres Konzept zum Schutz vor Buffer-Overflow-Angriffen sind Stack Cookies [4]. Stack Cookies sind Werte die auf den Stack, direkt nach den Puffer geschrieben werden. Bevor der Sprung zurück in die aufrufende Funktion durchgeführt wird, wird die XOR Operation auf den Stack Cookie und die Rücksprungadresse ausgeführt. Der so errechnete Wert wird mit dem erwarteten Wert verglichen. Falls es eine Abweichung gibt wird nicht die RET Operation ausgeführt, sondern in eine Sicherheitsroutine gesprungen, die diesen Fall behandelt.

4.2. Code-Qualität

Code Reviews können solche Sicherheitslücken aufdecken [13]. Code Reviews können teilweise automatisch mit Werkzeugen zur statischen Code Analyse durchgeführt werden [20].

Eine weiterer wichtiger Stützpfeiler für sichere Software sind schnell ausgelieferte Sicherheitsaktualisierungen. Dazu gehört laut [19] unter anderem ein System zum mobilen versenden von Aktualisierungen an Autos mit Mobilfunkverbindung.

A. Literaturverzeichnis

- [1] Airbiquity signs telematics deal with bmw, Oktober 2006.
- [2] Airbiquity link highlights ford's telematics strategy, 2008.
- [3] H. Arora. Buffer overflow attack explained with a c program example, Juni 2013. Verfügbar unter <http://www.thegeekstuff.com/2013/06/buffer-overflow/>.
- [4] B. Bray. Compiler security checks in depth. MSDN, Februar 2002.
- [5] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *Proceedings of the 20th USENIX Conference on Security, SEC'11*, pages 6–6, Berkeley, CA, USA, 2011. USENIX Association.
- [6] C. Eckert. *IT-Sicherheit*. Oldenbourg Wissenschaftsverlag GmbH, 2012.
- [7] Europäischer Rat. Richtlinie des rates 70/156/ewg, Februar 1970.
- [8] European Parliament, Council of the European Union. Richtlinie 98/69/ec des europäischen parlaments und des rates, Oktober 1998.
<http://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:31998L0069>.
- [9] European Parliament, Council of the European Union. Verordnung (eg) nr. 661/2009 des europäischen parlaments und des rates, Juli 2009.
- [10] European Parliament, Council of the European Union. Verordnung (eu) 2015/758 des europäischen parlaments und des rates, April 2015.
- [11] S. Gallagher. Fiat chrysler recalls 1.4 million cars over remote hack vulnerability. *arstechnica*, Juli 2015. Verfügbar unter <http://arstechnica.com/security/2015/07/fiat-chrysler-recalls-1-4-million-cars-over-remote-hack-vulnerability/>.
- [12] A. Greenberg. Your bmw or benz could also be vulnerable to that gm onstar hack, August 2015. Verfügbar unter <http://www.wired.com/2015/08/bmw-benz-also-vulnerable-gm-onstar-hack/>.
- [13] M. Howard. A process for performing security code reviews. *Security Privacy, IEEE*, 4(4):74–79, Juli 2006.
- [14] S. Hwang. Wireless car security system, Juli 15 1997. US Patent 5,648,754.

- [15] S. International. Recommended practice for pass-thru vehicle programming, Dezember 2004.
- [16] U. Kiencke, S. Dais, and M. Litschel. Automotive serial controller area network. Technical report, Robert Bosch GmbH, Februar 1986.
- [17] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, SP '10, pages 447–462, Washington, DC, USA, 2010. IEEE Computer Society.
- [18] D. Lee. Keyless cars 'increasingly targeted by thieves using computers', Oktober 2014.
- [19] K. Mahaffey. Hacking a tesla model s: What we found and what we learned, August 2015. Verfügbar unter <https://blog.lookout.com/blog/2015/08/07/hacking-a-tesla/>.
- [20] G. McGraw. Automated code review tools for security. *Computer*, 41(12):108–111, Dezember 2008.
- [21] Mixer. Writing buffer overflow exploits - a tutorial for beginners. Verfügbar unter <http://www.eecis.udel.edu/~bmiller/cis459/2007s/readings/buff-overflow.html>.
- [22] K. Poulsen. Hacker disables more than 100 cars remotely, März 2010. Verfügbar unter <http://www.wired.com/2010/03/hacker-bricks-cars/>.
- [23] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. 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.
- [24] L. Seltzer. The morris worm: Internet malware turns 25, November 2013. Verfügbar unter <http://www.zdnet.com/article/the-morris-worm-internet-malware-turns-25/>.
- [25] A. Silberschatz, P. B. Galvin, and G. Gagne. *Operating System Concepts*. John Wiley & Sons. Inc, 2005.
- [26] H. Song, S. Zhu, and G. Cao. Svats: A sensor-network-based vehicle anti-theft system. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages –, April 2008.
- [27] D. Spaar. Auto, öffne dich! Sicherheitslücken bei BMWs ConnectedDrive, 2015.
- [28] T. Stevens. Gm issues fix for onstar hack, Juli 2015. Verfügbar unter <http://www.cnet.com/news/ownstar-onstar-hack/>.
- [29] P. Syverson. A taxonomy of replay attacks [cryptographic protocols]. In *Computer Security Foundations Workshop VII, 1994. CSFW 7. Proceedings*, pages 187–191, Juni 1994.

- [30] V. Turner. Automotive vehicle anti-theft and anti-vandalism and anti-carjacking system, Dezember 14 1999. US Patent 6,002,326.
- [31] R. Verdult, F. D. Garcia, and B. Ege. Dismantling megamos crypto: Wirelessly lockpicking a vehicle immobilizer. In *Supplement to the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 703–718, Washington, D.C., 2015. USENIX Association.