Integration realer Angriffe in simulierte Echtzeit-Ethernet-Netzwerke

Sandra Reider, Philipp Meyer, Timo Häckel,
Franz Korf und Thomas C. Schmidt

Department Informatik,
Hochschule für angewandte Wissenschaften Hamburg, 20099 Hamburg
{sandra.reider|philipp.meyer|timo.haeckel|franz.korf|t.schmidt}
@haw-hamburg.de

Zusammenfassung. Ethernet wird zunehmend Bestandteil moderner Fahrzeugnetze und bildet die aussichtsreichste Technologie für künftige Hochgeschwindigkeits-Backbones im Auto. 'Connected Vehicles' öffnen gleichzeitig ihre internen Fahrzeugnetzwerke nach außen und ermöglichen so eine Vielzahl neuer Angriffe, für die neue Sicherheitskonzepte entwickelt werden müssen. Sicherheitskonzepte und -mechanismen vor ihrer Einführung in einer Simulationsumgebungen zu testen, ist flexibel, schnell und kostengünstig. In dieser Arbeit stellen wir ein Konzept vor, mit dem realer Angriffsverkehr aufgezeichnet und in eine Simulationsumgebung eingespielt werden kann. Dieses evaluieren wir am Beispiel eines DoS-Angriffs und können zeigen, dass die erwarteten Auswirkungen des abgespielten Angriffs in der Simulation wiedergegeben werden.

1 Einleitung

Aktuelle Fahrzeuge setzen Fahrfunktionen mit Sensoren, Aktoren und Steuergeräten um, die meist über bewährte Bussysteme (z. B. CAN, Flexray) kommunizieren. In zukünftigen Fahrzeugnetzwerken werden aufgrund der zunehmenden Anzahl an Steuergeräten und des erhöhten Bandbreitenbedarfs immer häufiger Ethernet-Technologien eingesetzt werden [1]. Die Kommunikation vieler Steuergeräte (z. B. Antiblockiersysteme) unterliegt dabei Echtzeitanforderungen, welche für die Funktionssicherheit des Autos garantiert werden müssen [2]. Neue Szenarien, wie die Verkehrsvernetzung (Car-to-X), die Einführung von Internet im Infotainmentbereich und die verstärkte Anbindung an Cloud- und IoT-Dienste, erfordern die Öffnung des Fahrzeugnetzwerks nach außen.

Angriffe auf Fahrzeuge erfolgen häufig über Schwachstellen in Steuergeräten. Der Zugang zu einem Steuergerät kann dabei über jede der Außenschnittstellen erfolgen (z.B. Mobilfunk, Bluetooth, Diagnoseport), nachdem ggfs. Gateways bzw. Autorisierungsbarrieren überwunden wurden [3]. Von einem Steuergerät aus kann versucht werden, manipulierte Nachrichten über das Netzwerk zu verschicken und andere Steuergeräte ebenfalls zu kompromittieren. Das Netzwerk

nimmt daher eine zentrale Rolle in der Erkennung und Unterbindung solcher Angriffe auf Fahrzeugkomponenten ein. Für zukünftige Fahrzeugnetzwerke ist es folglich von essentiellem Vorteil, wenn neue Sicherheitskonzepte gefunden und validiert werden können.

Ziel dieser Arbeit ist es ein Konzept umzusetzen, mit dem reale Angriffe in eine Simulationsumgebung für Echtzeit-Ethernet-Netzwerke integrieren werden können. Diese Methodik soll es ermöglichen, Sicherheitskonzepte anhand verschiedener realistischer Angriffsmuster zu replizieren und im Detail zu untersuchen. Wir stellen ein Konzept zur Wiedergabe realer Angriffsverkehrsdaten in einem simulierten Netzwerk vor. Am Beispiel eines Fahrzeugnetzwerks mit realem Datenverkehr zeigen wir, wie die Auswirkungen eines Angriffs in der Simulation untersucht werden können, und dass die Ergebnisse den erwarteten Folgen des Angriffs entsprechen.

Im Abschnitt 2 werden Grundlagen und verwandte Forschungsergebnisse vorgestellt. Abschnitt 3 stellt das Konzept und dessen Umsetzung vor. In Abschnitt 4 werden die Auswirkungen des Konzepts mithilfe eines Fallbeispiels untersucht und diskutiert. Abschließend folgen Zusammenfassung und Ausblick in Abschnitt 5.

2 Grundlagen und verwandte Arbeiten

Bei der Nutzung von Ethernet-Topologien in internen Netzwerken moderner Fahrzeuge muss sichergestellt werden, dass die Echtzeitanforderungen sicherheitskritischer Anwendungen erfüllt werden. Der vielversprechendste Kandidat, um diese in zukünftigen Bordnetzen umzusetzen, ist Time-Sensitive Networking (TSN) [4]. TSN ist eine Sammlung verschiedener IEEE Substandards, die Echtzeitanforderungen für Ethernet sicherstellen und unter anderem für den Einsatz in Fahrzeugnetzwerken entwickelt wurden. Es ermöglicht den Transport von unterschiedlichen Verkehrsklassen mit ihren speziellen Servicegarantien in derselben Infrastruktur. Diese konkurrierenden Verkehrsklassen sind nach Prioritäten geordnet, wobei 0 die niedrigste und 7 die höchste Priorität ist.

Checkoway et al. haben die Angriffsoberfläche moderner Autos untersucht und dargestellt, wie das interne Autonetzwerk über eine Vielzahl von Schnittstellen angreifbar ist [3]. Darunter fallen bspw. Bluetooth- und Mobilfunk-Schnittstellen, über die mit einem Laptop oder Smartphone Zugang zu den Telematik-Systemen des Fahrzeugs erlangt werden kann. Nach erfolgreicher Manipulation eines Steuergeräts über eine solche Schnittstelle, können von diesem über das interne Netzwerk weitere Steuergeräte angegriffen werden. Auf diese Weise können sicherheitskritische Funktionen manipuliert und somit nicht nur die Informations-, sondern auch die Funktionssicherheit des Fahrzeugs gefährdet werden.

Miller und Valasek haben gezeigt, dass es auch ohne physischen Zugang zu einem Fahrzeug möglich ist, dessen sicherheitskritische Funktionen zu manipulieren [5]. Sie haben sich über das Infotainment-System Zugang zu einem 2014 Jeep Cherokee und dessen CAN-Bus-System verschafft und damit die Kontrolle

über diverse Fahrzeugfunktionen erhalten. Unter anderem haben sie das Soundsystem und die Scheibenwischer, aber auch sicherheitskritische Steuergeräte der Lenkung und Bremsen kontrolliert. Um die Lenkung und Bremsen kontrollieren zu können, muss der Diagnosemodus aktiviert werden, was nur bei geringen Geschwindigkeiten möglich ist. Im Diagnosemodus können u. a. das Parkassistenz- und das Kollisionsvermeidungs-System ausgeschaltet werden. Die Lenkung und Bremsen können dann mit CAN-Nachrichten kontrolliert werden, mit denen diese Assistenzsysteme das Fahrzeug steuern. Aber auch bei hohen Geschwindigkeiten konnten sie bspw. den Motor ausstellen. Die Angriffe erfolgten dabei ohne physischen Zugang zum Fahrzeug. Dadurch wurde aufgezeigt, dass neue Sicherheitskonzepte entwickelt werden müssen, um vor Angriffen zu schützen, die die zunehmende Internetanbindung moderner Fahrzeuge ausnutzten.

Der Wechsel von CAN-Bus auf Ethernet als Kommunikationsmedium erfordert eine Neuplanung der Kommunikationsarchitektur. Dies ermöglicht es die Sicherheit des neuen Fahrzeugnetzwerks bereits im Design zu berücksichtigen und dadurch die Entwicklung und Einführung neuer Sicherheitskonzepte zu begünstigen. Mundhenk hat in seiner Dissertation verschiedene Konzepte vorgestellt, mit denen die Sicherheit von interner Fahrzeugnetzwerken sowohl beim Neudesign als auch bei existenten Netzwerkarchitekturen verbessert werden kann [6].

Wir haben in [7] gezeigt, dass das Verhalten von Fahrzeugnetzwerken in einer Simulation analysiert werden kann, und die ereignisbasierte Simulationsumgebung vorgestellt, mit der wir dies bereits erfolgreich umsetzen. Dies ist schneller, unkomplizierter und günstiger als ein reales Fahrzeug zu verwenden.

3 Integration realer Verkehrsmuster in der Simulation

Die Simulation kann auch verwendet werden, um die Auswirkungen von Angriffen auf ein simuliertes Fahrzeugnetzwerk zu untersuchen und dadurch Sicherheitskonzepte zu evaluieren. Sie muss dafür sowohl eine realitätsgetreue Modellierung des Autonetzwerks beinhalten als auch realistische Angriffsmuster umsetzen. In der Simulation ist es möglich, Angriffe zu erkennen, die in einem realen Netzwerk schwer sichtbar sind. Ein Lauschangriff kann bspw. erkannt werden, wenn bei einem Steuergerät Datenpakete aufgezeichnet werden, die nicht für dieses bestimmt sind. Insgesamt können in der Simulation Verlust, Injektion, Manipulation, Neuordnung, Umleitung und Änderung des Zeitverhaltens von Paketen gemessen werden. Die im simulierten Netzwerk gemessenen Auswirkungen des Angriffs können mit den erwarteten Folgen verglichen werden. Wird z. B. ein Denial-of-Servcie(DoS)-Angriff in das Netzwerk eingespielt, führt dies in der Regel zu Verlusten, vergrößerten Latenzen und Häufung von Paketen.

3.1 Simulationsumgebung

Als Simulationsumgebung dient dieselbe Kombination aus Frameworks, die bereits in [7] verwendet wurde. Sie ist in Abbildung 1 dargestellt und baut auf dem diskreten Ereignissimulator OMNeT++ [8] auf. Dieser wird mit dem Open-

Source-Framework INET [9] um Standardprotokolle (z. B. Ethernet, IP, TCP) erweitert. Zur Simulation fahrzeuginterner Netzwerke werden unsere Open-Source-Frameworks CoRE4INET, FiCo4OMNeT und SignalsAndGateways verwendet [10, 11]. CoRE4INET erweitert das INET-Framework um Protokolle für die Echtzeit-Ethernet-Kommunikation, wie beispielsweise TSN. FiCo4OMNeT baut direkt auf OMNeT++ auf und stellt CAN-Bussimulationsmodelle bereit. SignalsAndGateways bietet unter anderem Gateways, die CAN-Nachrichten in Ethernet-Frames verpacken und somit die Kommunikation von CAN-Steuergeräten über ein Ethernet-Backbone ermöglichen.

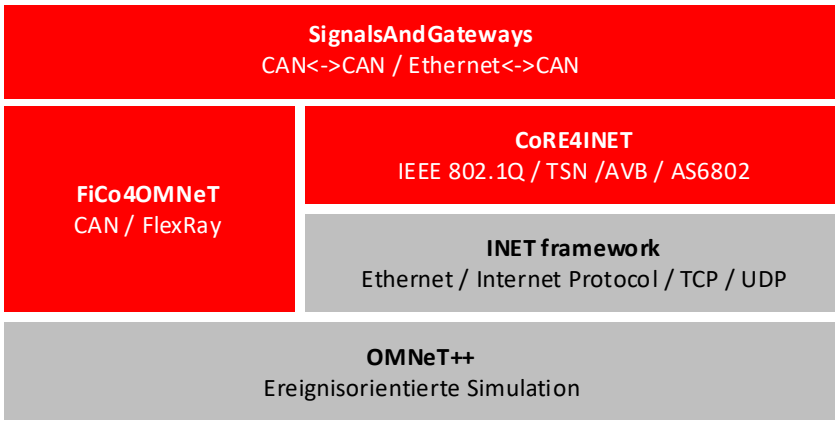


Abb. 1. Die verwendeten, aufeinander aufbauenden Frameworks.

3.2 Paketgenerator für aufgezeichneten Ethernet-Datenverkehr

Die Qualität der Simulationsergebnisse hängt direkt von den verwendeten Stimuli ab. Diese werden von Paketgeneratoren erzeugt und in das simulierte Netzwerk gesendet. Paketgeneratoren können bspw. in Form eines realen Steuergeräts an die Simulation angebunden sein oder anhand bestimmter Spezifikationen Datenpakete erzeugen. Das Einspielen von zuvor aufgezeichnetem Datenverkehr erhöht die Flexibilität der Simulation und die Reproduzierbarkeit der Ergebnisse. Aufgrund ihrer weiten Verbreitung werden für das Einspielen pcapng-Dateien verwendet. Dies ermöglicht einen universellen Einsatz der Angriffssimulation, sowie die Verwendung von Angriffsmustern aus externen Quellen.

Wir haben in CoRE4INET ein Modul zum sequentiellen Einlesen von pcapng-Dateien sowie einen Paketgenerator implementiert, der die eingelesenen Ethernet-Frames in die Simulation einspielt. Das Modul bietet dem Paketgenerator eine Schnittstelle, über die Ethernet-Frames und deren Sendezeiten abgefragt werden können. Die pcapng-Datei wird auf Anfrage des Paketgenerators blockweise eingelesen und kann unter anderem Section Header (SHB), Interface Description (IDB), Enhanced Packet (EPB) und Simple Packet (SPB) Blöcke enthalten.

Dem SHB wird entnommen, ob die folgenden Blöcke in der Byte-Reihenfolge des lesenden Systems aufgezeichnet wurden. Im IDB ist die Information enthalten, welche Auflösung die Zeitstempel in den EPBs haben. Ein EPB enthält einen aufgezeichneten Ethernet-Frame und dessen Sendezeit. Im SPB sind Ethernet-Frames ohne Zeitstempel enthalten, wodurch er für das realitätsgetreue Einspielen von Angriffsmustern ungeeignet ist. Alle weiteren Blöcke enthalten keine für die Simulation relevanten Daten und werden deshalb übersprungen. Der Zeitstempel des ersten eingelesenen EPBs wird als Initialzeit gespeichert. Die Differenz zu dieser ergibt die Simulationszeit der nachfolgenden Frames. Wenn die Sendezeit abgefragt wird, wird ein neuer EPB eingelesen und der enthaltene Ethernet-Frame in einer Liste gespeichert. Dadurch kann der Paketgenerator immer genau eine Zeit und den zugehörigen Frame oder mehrere Zeiten am Stück und anschließend alle zugehörigen Frames abfragen.

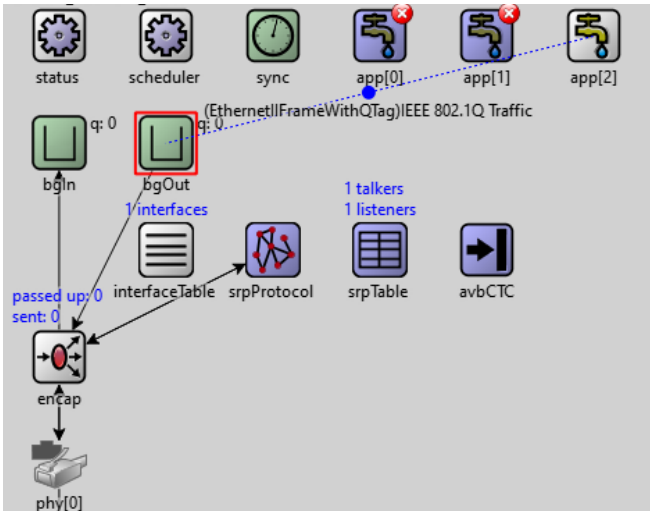


Abb. 2. Darstellung eines TSN Steuergeräts in der Simulation.

Bei der Konfiguration des Paketgenerators kann optional eine Startzeit festgelegt werden, zu der der erste Ethernet-Frame eingelesen wird. Alle weiteren Frames werden zu den Zeiten gesendet, die von der pcapng-Schnittstelle erhalten werden. Sobald der entsprechende Simulationszeitpunkt erreicht wird, wird der entsprechende Frame von der pcapng-Schnittstelle geholt. Die Headerfelder des Frames, wie bspw. die Zieladresse, werden von dem Paketgenerator überschrieben. Dafür kann bei der Konfiguration festgelegt werden, an welche Adresse des simulierten Netzwerkes die Frames gesendet werden sollen. Wenn bei der Konfiguration eine Priorität für die eingelesenen Frames gesetzt wurde, wird zusätzlich ein VLAN-Tag und die Priorität hinzugefügt. Anschließend wird der Frame an den Ausgangspuffer des Steuergeräts weitergeleitet.

Der Paketgenerator kann als eigenständiges Steuergerät ins Netzwerk eingebunden oder in ein bestehendes Steuergerät integriert werden. Wenn in bestehenden Steuergeräten bereits von anderen Anwendungen Datenverkehr erzeugt wird, kann er diese entweder ersetzen oder zusätzlich zu ihnen agieren. In Abbildung 2 ist beispielhaft ein Steuergerät während der Simulation zu sehen, in das der Paketgenerator integriert wurde. Die bisherigen Anwendungen app[0] und app[1] des Steuergeräts wurden deaktiviert und stattdessen sendet der Paketgenerator app[2] den aufgezeichneten Datenverkehr an den Ausgangspuffer. Von dort werden sie abhängig von ihrer Priorität an den Puffer des physischen Ports weitergeleitet, von dem sie in das Netzwerk gesendet werden.

4 Fallbeispiel eines DoS-Angriffs im Fahrzeugbordnetz

Wir führen an einem simulierten Fahrzeugnetzwerk zwei Durchläufe unter gleichen Bedingungen (Netzwerkkonfiguration, Simulationsdauer) durch. Dabei wird der Datenverkehr eines Ethernet-Steuergeräts aus zwei zuvor aufgezeichneten Dateien eingespielt. Im ersten Durchlauf wird der Datenverkehr des normalen Betriebs ohne Angriffe eingespielt, um eine Referenz zu erhalten. Im zweiten Durchlauf wird ein Angriff eingespielt, um dessen Auswirkungen auf das Netzwerk zu untersuchen. Während beider Durchläufe werden Statistiken über die Netzwerk-Kommunikation (z. B. Anzahl und Latenz von Paketen) aufgezeichnet und diese anschließend verglichen. So werden verschiedene Auswirkungen von Angriffen auf das Netzwerk in der Simulation gemessen. An der Veränderung der Anzahl gesendeter und empfangener Pakete kann beispielsweise Paketverlust/-injektion erkannt werden.

4.1 Simuliertes Fahrzeugnetzwerk

Als Grundlage für die realitätsgetreue Abbildung eines Fahrzeugnetzwerks dient dessen Kommunikationsmatrix. Diese enthält alle verbauten Steuergeräte sowie die zwischen ihnen ausgetauschten Nachrichten. Die Spezifikation kann durch verschiedene Netzwerkarchitekturen abgebildet werden. In aktuellen Fahrzeugen wird eine domänenbasierte Architektur verwendet. Die Steuergeräte sind dabei nach ihrer Funktionalität in Domänen gruppiert (z. B. Fahrwerk, Infotainment). Innerhalb einer Domäne sind die Steuergeräte über CAN-Busse verbunden. Ein zentrales Gateway wickelt die domänenübergreifende Kommunikation ab.

Basierend auf der Kommunikationsmatrix haben wir das Fahrzeugnetzwerk eines Mittelklassenfahrzeugs von einer Domänenarchitektur in eine moderne Ethernet-Zonenarchitektur [13] überführt. Diese teilt die Steuergeräte anhand ihrer physischen Position im Fahrzeug in Gruppen ein. Die Gruppen bilden Zonen (z. B. vorne links), die durch ein Ethernet-Backbone verbunden sind. Innerhalb der Zonen sind die Steuergeräte weiterhin an domänenspezifischen CAN-Bussen angeschlossen. Das resultierende Netzwerk ist in Abbildung 3 zu sehen und wurde von uns bereits in einer vorangegangenen Arbeit verwendet [14].

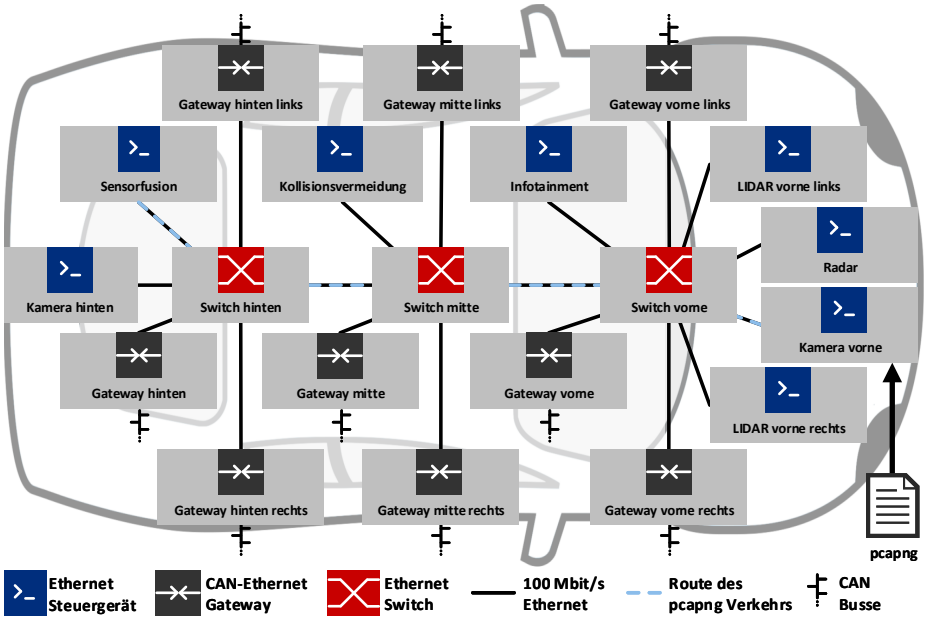


Abb. 3. Die Topologie des verwendeten Fahrzeugnetzwerks.

Das Ethernet-Backbone verfügt über 100 Mbit/s Links und verbindet die Steuergeräte mit drei Switches, die sich vorne, mittig und hinten im Fahrzeug befinden. An jedem Switch sind wiederum drei Gateways angeschlossen, die die Steuergeräte des linken, mittleren und rechten Bereichs im Auto verbinden. Dadurch entstehen insgesamt neun Zonen im Fahrzeugnetzwerk. Hinter jedem der neun Gateways befinden sich domänenbasierte CAN-Busse, an denen die CAN-Steuergeräte angeschlossen sind. Diese sind in der Abbildung nicht explizit dargestellt. Zwischen den einzelnen Gateways werden insgesamt 208 CAN-Nachrichten mit verschiedenen Prioritäten über das Ethernet-Backbone ausgetauscht. Zusätzlich zu den Gateways haben wir eine Auswahl an Steuergeräten direkt an das Backbone angebunden. Dazu gehören insbesondere ein Radar, das über Zeitschlitzverfahren Nachrichten an die Kollisionsvermeidung sendet, und je zwei Kameras und Laserscanner (LIDAR), die ihre Rohdaten mittels Netzwerkströmen, für die Bandbreite reserviert ist, an die Sensorfusion im hinteren Teil des Autos versenden.

4.2 Aufzeichnung der Angriffsmuster und Netzwerkkonfiguration

In die Simulation werden Aufzeichnungen von realen Angriffen eingespielt, um möglichst realitätsnahe Angriffsmuster zu erhalten. Diese erzeugen wir im Ethernet-Netzwerk eines Prototypfahrzeugs unter Verwendung von etablierten Werkzeugen für Penetrationstests, indem wir am Interface eines Kamerasteuergerätes den ausgehenden Datenverkehr aufzeichnen. Es wurden zwei pcapng-Dateien er-

zeugt, von denen eine den regulären Datenverkehr enthält und die zweite einen generierten DoS-Angriff. Der DoS-Angriff wurde über den in Listing 1 dargestellten Kommandozeilen-Befehl mit dem Paket-Injektor *T50* [12] generiert und sendet einhunderttausend minimale UDP-Pakete in das Netzwerk. Der Injektor hat dabei im Mittel 67834 Pakete pro Sekunde generiert. Die –threshold option legt die Anzahl der generierten Pakete fest, darauf folgen die IP-Adressen des Empfängers (Sensor Fusion, 10.0.11.6) und des Senders (Kamera, 10.0.11.5), sowie das Protokoll (UDP) und der Ausgangsport (1112), über den die Pakete versendet werden sollen.

Listing 1. Der t50-Kommandozeilenbefehl zum Erstellen des DoS-Angriffs.

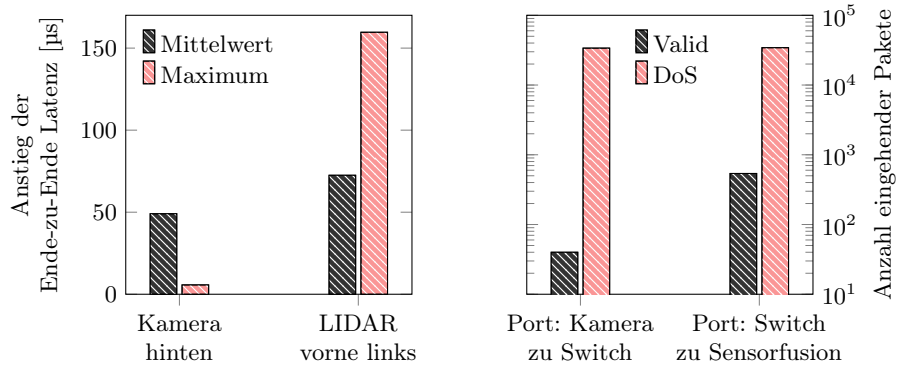
```
1 t50 --threshold 100000 10.0.11.6 --saddr 10.0.11.5 --protocol
    UDP --dport 1112
```

Als Grundlage für die Simulation wird das oben beschriebene Autonetzwerk konfiguriert. In das Kamerasteuergerät am vorderen Switch wird der Paketgenerator integriert, der die aufgezeichneten Nachrichten aus den pcapng-Dateien in das Netzwerk einspielt. Die Nachrichten werden mit Priorität 6 an das Sensorfusionssteuergerät am hinteren Switch gesendet und beeinträchtigen damit den gesamten Datenverkehr mit gleicher oder niedrigerer Priorität im Ethernet-Backbone. Die Simulationszeit entspricht bei beiden Durchläufen 500 ms. Die Konfiguration des restlichen Netzwerks ist für beide Durchläufe identisch.

4.3 Ergebnisse

Durch den DoS-Angriff werden innerhalb der 500 ms Simulationsdauer durchschnittlich 33917 zusätzliche UDP-Nachrichten ohne Payload von der Kamera am vorderen Switch zu der Sensorfusion am hinteren Switch gesendet. Die Gesamtanzahl der gesendeten Pakete zwischen der Kamera und dem vorderen Switch, sowie dem hinteren Switch und der Sensorfusion ist in Abbildung 4(b) graphisch dargestellt und zeigt den erwarteten Anstieg an Datenpaketen. Von der Kamera werden innerhalb der 500 ms regulär 40 Pakete gesendet, mit dem DoS-Angriff 33814. Dadurch steigt die Auslastung des Links zwischen der Kamera und dem vorderen Switch von 0.88% auf 39.66% an. Die Sensorfusion erhält im normalen Betrieb 539 Datenpakete, davon 40 mit Priorität 6. Während des DoS-Angriffs erhöht sich die Anzahl auf 34304, wovon 33805 mit Priorität 6 gesendet wurden. Die Auslastung des Links zwischen hinterem Switch und Sensorfusion erhöht sich dadurch von 13.09% auf 51.87%.

Wir erwarten, dass die Ende-zu-Ende Latenzen aller Nachrichten, die mit gleicher oder niedrigerer Priorität auf denselben Pfaden (in Abb. 3 gestrichelt) gesendet werden, zunimmt. In Abbildung 4(a) sind die Anstiege der mittleren und maximalen Ende-zu-Ende Latenzen der Nachrichten dargestellt, die jeweils mit Priorität 5 vom linken LIDAR und von der hinteren Kamera an die Sensorfusion gesendet werden. Der LIDAR und die Kamera, von der der Angriff ausgeht, sind beide am vorderen Switch angebunden. Dadurch konkurrieren die Nachrichten an drei Links. Es ist erkennbar, dass dies die Latenzen erhöht. Der



(a) Zwei Beispiele für die Erhöhung der mittleren und maximalen Ende-zu-Ende Latenzen durch den DoS-Angriff.

(b) Log. Darstellung der Menge eingehender Pakete, von der vorderen Kamera zur Sensorfusion an zwei Ports.

Abb. 4. Auswirkungen des DoS-Angriffs im simulierten Netzwerk.

Datenverkehr der hinteren Kamera wird über den hinteren Switch direkt an die Sensorfusion gesendet und konkurriert entsprechend nur an einem Link mit den Datenpaketen des DoS-Angriffs. Auch bei diesem ist eine leichte Erhöhung der Latenzen erkennbar.

Insgesamt ist erkennbar, dass der DoS-Angriff in der Simulation die zu erwartenden Auswirkungen bzgl. einer verstärkten Auslastung der Links und einer Vergrößerung der Ende-zu-Ende Latenzen zeigt. Durch den aufgezeichneten DoS-Angriff wird die Bandbreite nicht so stark genutzt, dass es zu Paketverlusten kommt. Da die Auswirkungen eines eingespielten Angriffs in der Simulation sichtbar sind, ist es möglich mit einem aufgezeichneten Angriffsmuster die Sensibilität verschiedener Echtzeit-Ethernet-Netzwerke auf diesen Angriff zu vergleichen und die Wirksamkeit neuer Sicherheitskonzepte zu testen.

5 Zusammenfassung und Ausblick

Die Integration realer Angriffs-Stimuli in eine Netzwerksimulation ermöglicht es, das Schadenspotenzial von Angriffen zu bestimmen und daraus Einsichten über Architektur und Design künftiger Netzwerke im Auto abzuleiten sowie die Wirksamkeit vorgesehener Sicherheitsmechanismen zu testen.

An einem realitätsnahen Beispiel basierend auf realen Fahrzeugdaten konnten wir demonstrieren, wie das vorgestellte Konzept realen Verkehr als Stimulus in eine Echtzeit-Ethernet-Simulation integriert. Wir haben erfolgreich den zuvor aufgezeichneten Angriffsverlauf eines DoS-Angriffs in unser Fahrzeugnetzwerk eingespielt und die erwarteten Veränderungen in der Anzahl gesendeter Datenpakete, der Linkauslastung und der Ende-zu-Ende Latenz identifiziert.

Zukünftig wollen wir den Funktionsumfang des Paketgenerators erweitern, um zusätzlich zu pcapng-Dateien auch pcap-Dateien einlesen und Daten ggfs.

auch periodisch abspielen zu können. Die Möglichkeiten der Angriffsverkehrseinspielung in Netzwerksimulationen wollen wir insbesondere nutzen, um neue Techniken der Anomalieerkennung auf ihre Wirksamkeit zu untersuchen. Hierbei stehen zeit- und flussbasierte Erkennungsmechanismen [15] zuvorderst auf unserer Agenda.

Danksagung Diese Arbeit wurde im Rahmen des SecVI-Projektes vom Bundesministerium für Bildung und Forschung gefördert.

Literaturverzeichnis

1. S. Brunner, J. Roder, M. Kucera, T. Waas: Automotive E/E-Architecture Enhancements by Usage of Ethernet TSN, 2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES), IEEE Press, Jun. 2017, S. 9-13.
2. T. Steinbach, H. Lim, F. Korf, T. C. Schmidt, D. Herrscher, A. Wolisz: Beware of the Hidden! How Cross-traffic Affects Quality Assurances of Competing Real-time Ethernet Standards for In-Car Communication, 2015 IEEE Conference on Local Computer Networks (LCN), IEEE Press, Okt. 2015, S. 1-9.
3. S. Checkoway, D. Mccoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno: Comprehensive Experimental Analyses of Automotive Attack Surfaces, Proceedings of the 20th USENIX Security Symposium, vol. 4, USENIX Association, Aug. 2011, S. 77-92.
4. IEEE 802.1 Working Group: IEEE Standard for Local and Metropolitan Area Network-Bridges and Bridged Networks, IEEE, Standard Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014), Jul. 2018.
5. C. Miller, C. Valasek: Remote Exploitation of an Unaltered Passenger Vehicle, Black Hat USA, vol. 2015, Aug. 2015, S. 91.
6. P. Mundhenk: Security for Automotive Electrical/Electronic (E/E) Architectures, Cuvillier, Aug. 2017.
7. P. Meyer, F. Korf, T. Steinbach, T. C. Schmidt: Simulation of Mixed Critical Invehicular Networks, Recent Advances in Network Simulation, Springer, Mai 2019, S. 317-345.
8. OMNeT++-IDE, `https://omnetpp.org` (abgerufen am 24.09.2020)
9. INET-Framework, `https://inet.omnetpp.org` (abgerufen am 24.09.2020)
10. CoRE-Simulationsumgebungen, `https://sim.core-rg.de` (abgerufen am 24.09.2020)
11. T. Steinbach, H. D. Kenfack, F. Korf, T. C. Schmidt: An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy, SIMUTools 2011 – 4th International OMNeT++ Workshop, März 2011, S. 375-382.
12. Kali Tools: T50, `https://tools.kali.org/stress-testing/t50` (abgerufen am 24.09.2020)
13. T. Steinbach: Ethernet-basierte Fahrzeugnetzwerkarchitekturen für zukünftige Echtzeitsysteme im Automobil, Springer Vieweg, Okt. 2018.
14. M. Cakir, T. Häckel, S. Reider, P. Meyer, F. Korf, T. C. Schmidt: A QoS Aware Approach to Service-Oriented Communication in Future AutomotiveNetworks, 2019 IEEE Vehicular Networking Conference (VNC) (IEEE VNC2019), IEEE Press, Dez. 2019.
15. P. Meyer, T. Häckel, F. Korf, T. C. Schmidt: Network Anomaly Detection in Cars based on Time-Sensitive Ingress Control, Proc. of the IEEE 21th Vehicular Technology Conference: VTC2020-Fall, IEEE Press, Okt. 2020.