European Journal for Security Research (2021) 6:171–187
https://doi.org/10.1007/s41125-021-00077-1

ORIGINAL PAPER

Static Analysis of Controller Area Network Communication for Attack Detection

Jo Laufenberg[1] · **Thomas Kropf**[1] · **Oliver Bringmann**[1]

Received: 30 June 2021 / Accepted: 24 December 2021 / Published online: 19 January 2022

Abstract
The vulnerability of modern cars increases due to their multiple connections to the environment, which offers the possibility of remote attacks in the worst case with fatal outcome. The controller area network (CAN) is still highly used and includes no security features, so intrusion detection systems are a promising approach to secure the communication. The proposed method monitors the CAN communication and uses static checks to differentiate between normal and attack traffic. This enables reliable and comprehensible attack detection and achieves a detection rate up to 100%, generating zero false alarms for the investigated data sets.

Keywords Intrusion Detection System · Controller Area Network · Automotive

1 Introduction

Nowadays, cars include about 100 Electronic Control Units (ECUs) (Miller and Valasek 2014) communicating among each other to increase driving comfort and to fulfill safety standards. The most established communication system between these ECUs is the Controller Area Network (CAN). Advantages of CAN are its proven high functional safety and its cost and complexity savings compared to direct wiring. Even ECUs which are responsible for high critical functionality like braking or airbag systems communicate over CAN.

The disadvantages of CAN are its vulnerability, as there are almost no security mechanisms built in and countermeasures are hard to include due to functionality and manufacturer requirements. Modern cars are highly connected to the outside

✉ Jo Laufenberg
laufenbe@informatik.uni-tuebingen.de

Thomas Kropf
kropf@informatik.uni-tuebingen.de

Oliver Bringmann
bringmann@informatik.uni-tuebingen.de

[1] Department of Computer Science, University of Tübingen, Sand 13, 72076 Tübingen, Germany

world, which leads to multiple remote attack vectors such as mobile broadband communication, Bluetooth, or GPS. Their number will increase even more concerning future technologies such as V2V (Vehicle to Vehicle) and V2I (Vehicle to Infrastructure) communications. The risks assumed from manipulated cars to people and the environment were shown by Miller and Valasek (2015). They are able to control the car through direct physical as well as remote access which in the worst case ends fatally for a human being.

In order to prevent damage, attack detection is an important tool. A reliable detection allows suitable reactions which may range from raising an alert up to setting the car into a fail safe mode or shutting down the engine. For this purpose, Intrusion Detection Systems (IDS) are used. The challenge for IDS operating on CAN increases due to the variety of CAN matrices that are used and not publicly available. A CAN matrix encodes the messages send internally over CAN and varies from manufacturer to manufacturer and even within manufacturers from vehicle line to vehicle line.

2 Related Work

The quality of an IDS can be evaluated on the basis of the following criteria, although not all possible criteria are mentioned here: the detection rate, the false positive rate, the applicability in real systems, which includes resource consumption and detection time, and the adaptability or expandability. In addition, the IDS should of course not represent a security risk for the system. The perfect IDS detects fast all possible attacks, even those which are unknown at the moment, has a false positive rate of zero, performs this tasks successful with a low resource budget, and is applicable in real-time systems.

Many imperfect IDS approaches exist with different strengths and weaknesses which will be described in more detail in Sect. 2.4. First, we introduce CAN and important attacks related to the protocol architecture.

2.1 Controller Area Network (CAN)

CAN was introduced by Robert Bosch GmbH in the early 1980's to reduce the wiring complexity of the automobile, where the possibility of attacks and especially remote attacks are not considered. CAN is a message-based broadcast protocol, where each message is transmitted sequentially and received by every participant on the bus. Only one participant can write to the bus at the same time. If more than one participant wants to write to the bus, the message with the highest priority, which is encoded in the identifier field, wins the arbitration and is allowed to transmit its message. CAN has a maximum signaling rate of 1 MB/s for short networks (smaller than 40 m), while the real bitrate depends on network length and the hardware used, usually around 500 KB/s for high speed CAN.

CAN has four frame types: Data frames which are used to transmit data, remote frames which are used to request data, error frames which are transmitted if a node

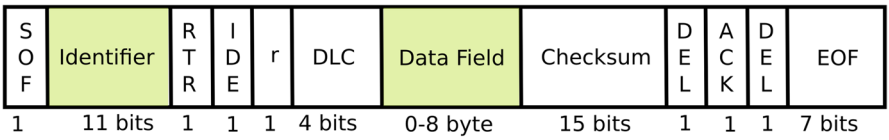

Fig. 1 CAN message format. SOF (Start of Frame), RTR (remote transmission request), IDE (identifier extension bit), r (reserved bit), DLC (data length code), ACK (acknowledge field), DEL (delimiters) and EOF (End of Frame). The green fields contain the actual information transported by the frame (colour figure online)

detects an error, and overload frames which are used to inject a delay between data or remote frames. Each message has a fixed format as shown in Fig. 1. Highlighted in green are the fields carrying the payload, while the other fields are defined by the protocol. The checksum is only used to detect transmission errors. The Identifier (ID) in base frame format has 11 bits, and in extended frame format 29 bits. IDE (Identifier extension bit) denotes which format is used. DLC (Data Length Code) is the number of bytes in the data field with a maximum data payload of 8 bytes in classical CAN. CAN FD introduced in 2012 allows a payload of up to 64 bytes. Considering the basic CAN protocol described above, CAN contains no security features. The resulting vulnerabilities are described in detail by Buttigieg et al. (2017) and Hartzell and Stubel (2017).

2.2 CAN Attacks

The lack of security benefits a range of attacks targeting the CAN, shown by Nie et al. (2017), which can either be executed directly via physical access to the car's OBD-II port (On-board Diagnostics port) or remote via different vectors, short- or long-range as described in Checkoway et al. (2011). As a consequence of a successful attack, the hacker is able to control the lights or the dashboard as well as he is able to control safety-critical functions related to the basic driving behavior of the vehicle like braking, acceleration, or steering.

Some attacks gain information about the architecture and the behavior of the connected ECUs in the car. Therefore, no specific prior knowledge is needed. Every manufacturer uses his own set of identifiers for the same functionality, this is decoded in the CAN matrix and is not even constant for every model. Thus, in-deep knowledge is important to perform the targeted attacks.

2.2.1 Replay Attack

For a replay attack Hoppe et al. (2007), no prior knowledge is needed. Observed messages are saved and replayed while the reaction of the system is monitored. Replaying only one message is possible as well as replaying a sequence of messages.

2.2.2 Fuzzing Attack

For a fuzzing attack either the ID or the payload of the observed messages is randomly manipulated and transmitted to the CAN. By again monitoring the system reaction, the attacker increases their knowledge about the target (Koscher et al. 2010).

2.2.3 Diagnostic Attack

Woo et al. (2015) mentioned that the range of standard message IDs differs from the range used by messages from the diagnostic tools, they identified a range for diagnostic messages from 0×700 to $0 \times 7FF$. The data set from Dupont and Lekidis (2019) includes data from an Opel Astra and shows occurrences of CAN IDs frequently in this range during normal driving, so the range for diagnostic messages depends on manufacturer and model or the concept is questionable.

2.2.4 Spoofing Attack

If the attacker has already some knowledge about the messages and their effects, selected messages with specific values are transmitted to the bus to achieve the desired outcome. If the legitimate ECU is still active and continues to transmit its messages to the bus, the attacker has to deal with the reaction of legal ECUs to conflicting messages. Miller and Valasek (2013) executed a spoofing attack on a Ford Escape, where the dashboard shoes a 'door ajar' alarm, while the door is closed.

2.2.5 Flooding Attack

To ensure that the spoofed messages are considered while the correctly working ECU is still transmitting the correct messages, flooding attacks are used. In these attacks, the frequency of the spoofed message is much higher, usually up to 100 times, than the transmission rate of the correct message (Miller and Valasek 2014).

2.2.6 Denial of Service (DoS) Attack

With a DoS attack the bus is occupied, so that no participant can transmit its messages to the bus. The behavior of the car in this case is unpredictable. The easiest way is to send messages with the highest priority (ID 0×000) as shown by Miller and Valasek (2013).

2.2.7 Suspension Attack

For suspension attacks, the assumption is that one message ID is transmitted only by one ECU. This kind of attack results in missing messages with a specific ID from the compromised ECU, as shown by Taylor et al. (2016). Wang et al. (2018) suppose that IDs are used by more than one ECU, but not at the same time. The data sets used in this paper assume the first case for suspension attacks.

2.2.8 Impersonating Attack

Like in case of suspension attacks the assumption here is a compromised ECU. This ECU stops its message transmission to CAN and manipulated messages are sent from another node using IDs from the compromised ECU. This attack could be seen as a combination of suspension and spoofing attack.

The attacks influence the transmitted CAN messages in different manner: some change the ID itself (fuzzing attack), or the sequence of IDs (replay, fuzzing, flooding, DoS, and suspension attack), some add IDs to the normal traffic (DoS and diagnostic attack), some influence therefore the frequency of IDs indirectly or directly (spoofing attack), and some influence the payload (fuzzing, spoofing, impersonating attack).

2.3 Data Sets

The data set from TU Eindhoven (Dupont and Lekidis 2019) includes recorded CAN traffic via the OBD-II port of Opel Astra and Renault Clio driving in a city. In addition, they build a prototype with a VW instrument cluster, 2 Arduino boards with CAN bus shields and a joystick. The two Arduino boards are programmed to model either a legitimate or a compromised ECU. The legitimate ECU sends its messages regularly, while the compromised ECU launches certain attacks on the CAN bus. Additionally, the dashboard receives inputs from the joystick, which represents the car's throttle. If the joystick is pushed forward, acceleration is desired and CAN messages related to the speedometer are sent resulting in increasing speed displayed on the dashboard.

The attacks in real cars are simulated by manipulating the recorded data, while for the prototype, the corrupt ECU executes DoS and speedometer spoofing attacks. For the real cars, each a diagnostic attack, two fuzzing attacks (one manipulates the ID, the other the payload), a replay, a DoS attack (with ID 0×000), and a suspension attack are simulated.

The data set from Hacking and Countermeasures Research Labs (HCR Lab) (Lee et al. 2018) includes recorded CAN traffic from a Kia Soul during driving in a city. They executed and recorded a DoS attack with ID 0×000, a fuzzing attack with random ID and payload, and two spoofing attacks related to revolutions per minute (RPM) and drive gear information (gear).

2.4 Related Work

Lokman and Othman (2019) and Dupont et al. (2019) summarize a wide range of IDS. IDS are either host-based, where the IDS is implemented in a participating node of the network, or network-based, where the IDS is attached as an additional node to the network. We focus on network-based IDS, where no internal knowledge or manipulation of existing ECUs is necessary and only the traffic of the network is monitored. IDS could be further categorized by the techniques used for intrusion detection.

Specification-based IDS use a set of thresholds and rules to detect abnormal behavior. These specifications depend on the component supplier and the car model, which are not generally available to the public.

Signature-based IDS use signatures of known attacks and compare their actual input to these patterns to detect attacks. Obvious new attack patterns are not always detectable using this method, while known attacks are detected accurately.

Anomaly-based IDS use patterns to compare the actual traffic against. The patterns in this case describe the normal behavior of the traffic. Deviations from the normal behavior are then classified as anomalous. Techniques used to build these patterns are machine learning, statistical, or hybrid approaches. Variance of detection rates are very high and a direct comparison is hard as different data sets are used, ranging from pure synthetical data to real data including real attacks. Statistical approaches have the advantage of reliable and traceable alerts in contrast to machine learning approaches. The best solutions using machine learning achieve an accuracy of 100%, with a false alarm rate that is higher than zero, like Taylor et al. (2016). They consider fuzzing attacks on message payload as well as replay attacks in which the order of the messages is changed for 20 message IDs on data taken from a Subaru Impreza for their experiments. Analysis of the data sets used in this paper show, however, an amount of 27–84 different message IDs in normal traffic. Beneath this, nothing is said about the applicability in real cars, especially about the resources and the time needed for detection. According to the authors, further research should be conducted on extensibility, among other things; in the presented approach, this is not provided.

Weber et al. (2018) use a hybrid approach. In the first step, specification-based static checks are applied, which use the specifications from the manufacturer. These checks are extended in the second step with learning checks, executed on signal time series as produced by sensors. They use CANoe from Vector Informatics together with a synthetic CAN signal showing that their approach works in principle. According to the authors, further evaluations of the approach should be realized with data from real vehicles, also with regard to the performance.

Another hybrid approach is proposed by Tariq et al. (2020). They achieve an accuracy of 99.45% for real car data of two different cars. They consider DoS, fuzzing, and replay attacks. The rule-based approach is based on ID frequency and hamming distance for the payload and reaches an F1-score of 99.9% for the Kia Soul and 98.41% for Hyundai Sonata. Additionally, they measured the time-delay in detection, which is for the rule-based part in average 0.073 s. As they only used a PC for the experiments (without any information regarding the

required memory), a statement regarding the operational capability in real systems is difficult to make.

Song et al. (2016) use the message rate as feature for attack detection. They achieve an accuracy of 100% for real car data but only for injection attacks. The time for detection is stated as 1 ms, without further information about the hardware used. They do not consider fuzzing, suspension, and impersonating attacks. Fuzzing attacks that only affect the payload are undetectable with this approach.

Wu et al. (2018) proposed an approach where the entropy of message IDs in a sliding window is used to detect DoS and injection attacks. They achieve an accuracy of 100% for DoS and 92.3% for injection attacks on real car data with 0.081 ms as response time of attack detection. Since they only use the message IDs for detection, they are not able to detect attacks that alter only the payload.

Zhang et al. (2018) use a two stage approach, where the first stage is a robust rule-based system and the second stage uses a deep neural network for anomaly detection. The rule-based system includes a valid ID rule and a time interval rule. With this they achieve an accuracy between 99.91 and 99.97% and a false positive rate of 0.018–0.09% for real car data with a processing time for each message between 0.53 and 0.61 ms on a ThinkPad T440s notebook. They consider DoS, replay, spoofing, suspension, and fuzzing attacks.

If we reconsider the criteria mentioned before (detection rate, false positive rate, detection time, resource consumption, applicability, and expandability), only detection rate and false positive rate are examined by most of the authors. For this point of view, perfect IDS have a false positive rate of zero, detect all possible attacks with a detection rate of 100%, and prove this on real car data. None of the mentioned approaches fulfills all criteria. Some have (nearly) perfect detection or false positive rates while tested only with simulated data or only with certain attacks. A direct comparison of the mentioned approaches is hard because they consider different attacks and use different data sets from different cars, which vary among others in amount of the used IDs.

The challenge here is to detect as many attacks as possible—while the false positive rate has to be zero—with a system that has real-time ability on embedded systems with limited resources. A method that can not fulfill this criterion is of no avail in practice. As well as a low false positive rate seems to be very good the interesting metric is the alerts per hour or better per day. Car owner which receives warnings every time they drive their car will ignore such alerts or will be frustrated. The rate of messages per second in CAN is between 8 and 17, thus makes up to 62,000 messages per hour. A false positive rate of 0.00001 may already raise one alert per hour.

Our approach shows that a systematic content-driven data analysis is much more effective than standard anomaly analysis approaches. With respect to the false positive rate, our approach is superior to those that use machine learning methods. In contrast to the rule-based approaches mentioned before, our approach achieves a higher detection rate and is able to detect more different attacks. Therefore, additional rules are developed based on analyses of the data sets. The basic assumptions made for this are also described by other authors who work on data sets for other cars. The application of the publicly available

data sets, Dupont and Lekidis (2019) and Lee et al. (2018), allows a comparison in the future, and additionally the data sets originate from real cars, providing the applicability in practice. To show that our approach can be used in real systems, we have created a demonstrator on a FPGA. Such a comprehensive IDS with demonstrated applicability in real systems has not been presented by any author before.

3 Method

First, we analyzed the attack-free dumps and the dumps with attack data for the different data sets. Based on these analyses, in the second step, we developed assumptions about certain properties of the dumps that can be used to distinguish normal dumps from those with ongoing attacks. In the third step, we used these assumptions to design criteria that are used to detect attacks. In order to detect all the mentioned attacks, the designed criteria must consider both the ID of the messages and their payload. The criteria are described in more detail in Sects. 3.1 and 3.2.

With these criteria, we implemented a framework consisting of different python scripts, which in a first step extracts the parameters needed for the criteria from a given attack-free dump. In the second step, the framework is used to explore the dumps containing attacks and reports the detected attacks. The different steps are performed automatically, first loading the complete dump and then following the parameter extraction, executing the attack detection with all criteria at the same time. Further, we designed a demonstrator with a Xilinx Zynq Z7 Board (Zybo), where we implemented the criteria on a FPGA. The Zybo is connected to a CAN bus, as well as a second FPGA, which is able to send CAN messages. The criteria are implemented as hardware-accelerated Assertion Checking Units (ACU), which are guided by software and thus reconfigurable. To visualize the detection of an attack, a LED is activated when an attack is detected. With

Table 1 Analysis of Opel Astra

Attack	FQ	#ID	#AF	Type
Attack-free	2,690,069	84	0	–
Diagnostic	807,009	87	10	Injection
DoS	827,555	85	40,015	Injection
			19,459	Deletion
Fuzz_{ID}	807,009	88	10	Injection
Fuzz_{pay}	806,999	84	10	Modification
Replay	807,026	84	27	Injection
Suspension	806,599	84	400	Deletion

Frame quantity (FQ) describes the amount of messages contained in the data set, #ID denotes the amount of different IDs, the amount of attack frames is denoted as #AF, and Type denotes the type of the manipulation related with the attack

Table 2 Analysis of Renault Clio with FQ the number of messages, #ID the number of different IDs in the data set

Attack	FQ	#ID	#AF	Type
Attack-free	386,567	55	0	–
Diagnostic	115,981	58	10	Injection
DoS	141,927	56	40,001	Injection
			14,045	Deletion
$Fuzz_{ID}$	115,981	59	10	Injection
$Fuzz_{pay}$	115,971	55	10	Modification
Replay	116,002	55	31	Injection
Suspension	115,472	55	499	Deletion

The quantity of attack frames is denoted as #AF, and Type denotes the type of the manipulation related with the attack

Table 3 Analysis of prototype with FQ the number of messages, #ID the number of different IDs in the data set

Attack	FQ	#ID	#AF	Type
Attack-free	100,292	17	0	–
Diagnostic	29,003	20	10	Injection
DoS	65,493	18	40,037	Injection
$Fuzz_{ID}$	29,003	21	10	Injection
$Fuzz_{pay}$	28,993	17	10	Modification
Spoof	146,109	17	5772	Injection
Suspension	28,702	17	291	Deletion

The quantity of attack frames is denoted as #AF, and Type denotes the type of the manipulation related with the attack

Table 4 Analysis of Kia Soul with FQ the number of messages, #Inj the quantity of injected attack frames and #ID denotes the number of different IDs in the data set

Attack	FQ	FQ clear	#Inj	#ID
Attack-free	988,871	988,871	–	27
DoS	3,665,771	2,774,703	587,521	28
Fuzzy	3,838,860	2,947,792	491,847	2048
Spoof-Gear	4,443,142	3,552,074	597,252	27
Spoof-RPM	4,621,702	3,730,634	654,897	27

this we can demonstrate that our approach gets along with the limited resources and is able to react in real-time.

The basic analysis of the data sets is shown in Tables 1, 2, 3 and 4 and covers the number of messages (denoted as FQ (Frame Quantity)), the quantity of different IDs (#ID), and, in case of data which includes an attack, the amount of Attack Frames (#AF) and the type of the manipulation (Type).

In detail the data set from TU Eindhoven contains attack-free data for the Opel Astra with 84 unique IDs (see Table 1), for the Renault Clio with 55 unique IDs (Table 2), and for the prototype the attack-free data contains 17 unique IDs (Table 3), whereat no extended IDs are considered.

For the data set from HCRL, the attack-free data contain 27 unique IDs in standard identifier format with 11 bits and encompass 988,871 messages in total, see Table 4. In case of the data set from HCRL for the Kia Soul, we found that the attack dumps show a gap in time, where the attacks take place before this gap. After the gap follows an attack-free sequence, which is identical for all files. So we divided the files at this point and received therefore a second attack-free sequence. The number of remaining frames in the respective files is called "FQ clear" in Table 4. The second attack-free sequence includes 891,068 messages and also 27 IDs.

In comparison, it is noticeable that the attack-free data of TU Eindhoven comprises significantly more messages than those of the HCRL, even if the generated second data set is included.

In both data sets, the DoS attack is characterized by one additional ID, while the fuzzing attack (related to the ID) is characterized by several additional IDs as can be seen in the Tables 1, 2, 3 and 4. In contrast, spoofing, suspension and fuzzing attacks (related to the payload) do not change the number of IDs.

Overall, it can be observed that only a small number of IDs is used during normal driving, contrary to the 2048 IDs that are possible to encode with 11 bits. Furthermore, it shows that the number of IDs used is influenced by attacks. From this we derive the assumption:

Assumption 1 (Message ID) *Not every possible ID is used during normal driving.*

The research from Woo et al. (2015) also points to this assumption.

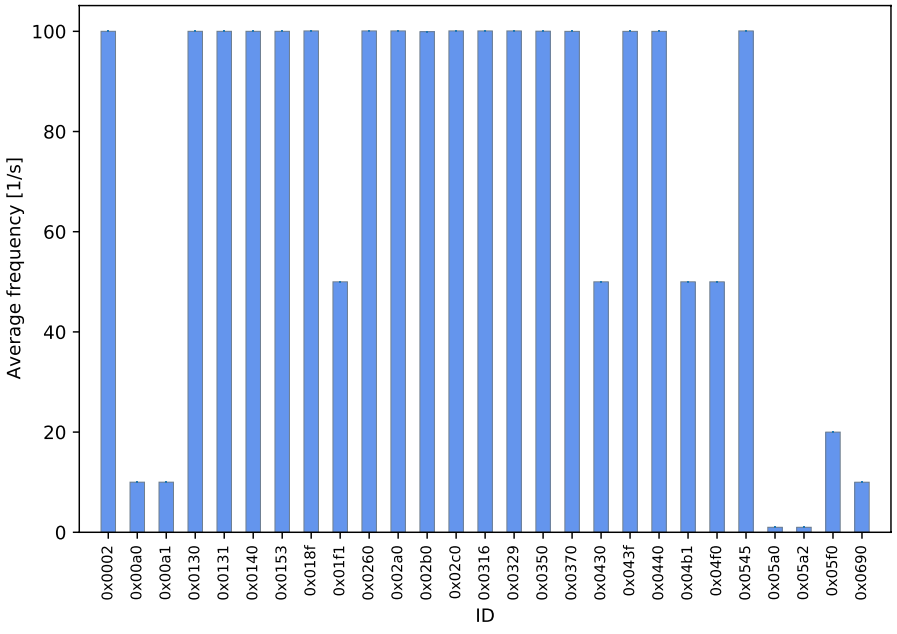

Fig. 2 Analysis of the frequencies of message IDs for the attack-free data from Kia Soul (HCRL)

Thereupon, we analyzed the ID frequencies in the different attack-free data sets. In Fig. 2, this analysis is shown for the data from Kia Soul. A single ID occurs between 507 and 50,689 times in this data set. Remarkably, only five frequencies are found: 100, 50, 20, 10, and 1. The range of this frequencies is very small, in average the overall maximum time delta is 0.09s. For the data set of Opel Astra there are ten frequencies found with the IDs occurring between 276 and 138,277 times, and for Renault Clio six frequencies with occurrences between 64 and 19,255. From this we derive the second assumption:

Assumption 2 (ID Frequencies) *IDs are sent periodically.*

Researches published by Ruth et al. (2012) or Wang et al. (2018) support this assumption.

Because the messages are sent periodically with a certain frequency and are ordered due to the arbitration of the CAN protocol, the messages should also occur in defined sequences. The analysis of the attack-free data sets supports this thesis. Thus, we formulate another assumption:

Assumption 3 (ID Sequences) *IDs appear in specific sequences, according to their content and priority of ID.*

Zhang et al. (2018) confirm this assumption.

The results in Table 5 show that criteria based on the assumptions described above are not enough to detect all attacks included in the data sets. The problem faced by most researchers in this field is that of the lack of a communication matrix. This makes analyses of the content of the messages difficult. But what we can analyze without knowing the meaning of the content of the payload is the length and presence of certain structures in the payload. The data sets from TU Eindhoven and HCRL show in our analysis a fixed length for every ID and some messages show constant values at certain bit positions in the payload. From this we infer the assumption:

Assumption 4 (Payload) *The payload of the messages have an internal structure.*

Hanselmann et al. (2020) split the payload into so-called signals, which promotes the hypothesis of the defined structure. Markovitz and Wool (2017) presented an algorithm to identify different fields in the payload, whereby they distinguish between constant, multi-value, and sensor/counter fields. Constant fields contain one value, multi-value fields contain a limited set of values (lower than possible for the field length), while counter/sensor fields contain values up to the number of values possible for the field length. According to their definition, the values of the fields representing the counters contain certain sequences. Likewise, this can be partially assumed for sensor fields, which must follow certain physical laws. Derived from this is the assumption:

Table 5 Detailed results for the proposed approach

Check								
Attack	ID_M	ID_D	$Freq_T$	$Freq_M$	DL	FB	$\sum$	ND
Opel Astra								
Dia	10	1	0	1	0	0	10	0
DoS	40,015	40,015	40,015	40,015	0	0	40,015	0
$Fuzz_{ID}$	10	0	0	0	0	0	10	0
$Fuzz_{pay}$	0	0	0	0	0	10	10	0
Replay	0	27	27	27	0	0	27	0
Suspension	0	0	400	400	0	0	400	0
Renault Clio								
Dia	10	0	0	0	0	0	10	0
DoS	40,001	40,001	40,001	40,001	0	0	40,001	0
$Fuzz_{ID}$	10	0	0	0	0	0	10	0
$Fuzz_{pay}$	0	0	0	0	0	10	10	0
Replay	0	31	31	31	0	0	31	0
Suspension	0	0	499	293	0	0	499	0
Prototype								
Dia	10	10	0	10	0	0	10	0
DoS	40,037	40,037	40,037	40,037	0	0	40,037	0
$Fuzz_{ID}$	10	1	0	10	0	0	10	0
$Fuzz_{pay}$	0	0	0	0	0	10	10	0
Spoof	0	1072	293	791	0	5772	5772	0
Suspension	0	0	176	291	0	0	291	0
Kia Soul								
DoS	587,521	575,918	551,452	563,414	0	0	587,521	0
Fuzzy	483,491	457,353	431,827	435,363	719	9356	491,847	0
Spoof-RPM	0	653,395	652,577	256,589	0	654,897	654,897	0
Spoof-Gear	0	594,823	593,814	232,405	0	0	594,823	2429

$\sum$ summarizes the amount of different attacks detected with the checks, while ND denotes the attacks that are not detected

Assumption 5 (Payload Sequences) *Values in payload fields occur in specific sequences.*

We decided to divide the assumptions into ID-based and assumptions related to the payload of the messages (data-based). Assumptions 1–3 fall into the ID-based category, while Assumptions 4 and 5 fall into the data-based category. The attack detection criteria resulting from this assumptions are introduced in detail below.

3.1 ID-Based Criteria

From the assumptions described before, we derive different criteria. This subsection address the ID-based criteria, the next subsection the data-based.

Criterion for Assumption 1 (Message ID)

$\mathrm{ID_M}$ *Valid IDs*
A set of IDs is identifiable, which appear during normal driving. This set is specific for one car model.

Criteria for Assumption 2 (ID Frequencies)

Freq_T: *Time-based*
Due to the periodicity of IDs a time range can be identified, in which IDs appear. By reason of arbitration the range is larger than the ID period.
Freq_M *Message-based*
Instead of using time for calculating the range, the number of messages send in between the appearance of messages with the same ID is used.

Criteria for Assumption 3 (ID Sequences)

ID_S: *ID Sequences*
A set of ID sequences is identifiable, which always appear together.
According to Zhang et al. (2018), the number of sequences is too large to be tested for, thus we have designed the following criterion:
ID_D: *Consecutive ID distance*
The distance between consecutive IDs in normal traffic is limited.

3.2 Data-Based Criteria

Starting from the data-based assumptions we derived the following criteria to detect attacks:

Criteria for Assumption 4 (Payload)

DL: Fixed length.
The length of the payload is fixed for every ID.

FB: Fixed bits.
Inside the payload exist fields with constant values.

SC: Signal correlation.
Fields within the payload are related to each other.

An example presented by Verma et al. (2020) is the wheel speeds of a car expressed in individual signals.

Criterion for Assumption 5 (Payload Sequences)

Pay_S: *Sequences in fields*

For parts of the payload (fields) sequences of ranges are identifiable, which appear during normal driving.

For every car model and data set we determine the parameters for the static checks using the attack-free dumps and execute the checks on the dumps containing the different attacks. For this we created a python framework, which automatically determines the parameters and executes the checks.

The data set from HCRL for the Kia Soul contains only a very small attack-free dump. Using the parameters determined from this dump, we classify normal data from other dumps as attack. The observed behavior was expected given the proportions of attack-free data and data containing attacks. Using the second attack-free data set that we acquire by splitting the attack data sets, we also achieve a false alarm rate of zero for Kia Soul.

4 Results

The results were determined individually for each car using our framework. First, we use the attack-free dumps to extract the parameters, and with these parameters, we perform the checks ID_M, ID_D, Freq_T, Freq_M, DL, and FB as described in Sect. 3 on the attack dumps.

To consider the criteria for evaluation and applicability of an IDS, besides the detection and the false positive rate, we implemented a demonstrator on a FPGA, which simulates an embedded system as it could be integrated into a vehicle, as described in Sect. 3 in more detail. With this demonstrator, we can attest the real-time capabilities of our approach and the ability of adaptation. Additionally, we can determine the resource consumption and measure the detection time.

The overall outcome is that we generate zero false alarms, with a detection rate over 99.99%. Every alert generated by our approach indicates an anomaly. The results are depicted in detail in Table 5 and described in the following sections.

4.1 ID-Based Attack Detection

It is unsurprising that we can not detect fuzzing attacks related to the payload with ID-based checks, as in this case only a payload mutation is executed. For the data sets from TU Eindhoven we are able to detect all diagnostic, DoS, fuzzing (ID), spoofing, and replay attacks with the ID-based criteria. Suspension attacks are not detectable with criteria that are related to message content, as a missing message do not have any content. However, since the frequency of the ID changes with missing

messages, the frequency-based criteria are suitable for detection. We detected all suspension attacks included in the data sets with this criteria.

For the Kia Soul data set from HCRL, we detected all DoS attacks and most of the fuzzy and spoofing attacks. Only for the spoofing attack related to the driving gear we could not detect all of the injected messages, but the amount of not detected attack messages is only 0.004%. Related to all attacks this amount is even smaller.

The detection time varies from criterion to criterion. While criteria ID_M and ID_D can decide directly when the message appears whether there is an attack, the frequency-based criteria ($Freq_T$ and $Freq_M$) can only trigger an alarm when the defined range is exceeded or not reached, which is not necessarily the time at which the message with the ID should have appeared. The measured detection time with the time-based criteria for the attacks is between 20 and 30 ms. Similar results are produced for the message-based frequency criterion, where the attacks are detected after 49 up to 65 messages. This corresponds to a range from 29 to 38 ms.

4.2 Data-Based Attack Detection

The data-based criteria complement the ID-based criteria. Their contribution is the additional detection of all fuzzing attacks related to the payload in the data sets of the TU Eindhoven. All injected messages in the spoofing attack in this data set are detected just as well by this criteria.

For Kia Soul the combination of the ID-based with the data-based checks detected all attacks in the fuzzy and the spoofing attack related to RPM.

The detection time for these criteria depends on the required message processing time, which is hardware and software dependent.

4.3 Comparison

Compared to the IDS mentioned in Sect. 2.4, we achieve a higher detection rate, consider more attacks and generate fewer false alarms with our approach.

Dupont et al. (2019) published an evaluation of different approaches based on the same data sets as our approach. The best approaches in this evaluation reach a false positive rate of 0%, or close to 0 and detect every kind of attack for the Kia Soul. The authors state that an attack type is detected if at least one alert has been raised when the IDS is executed on that attack data set, thus it has to be assumed that not all attacks are detected. Additionally, for the data set of TU Eindhoven not every attack type is detected. Especially the attacks that only modify the payload are not detected by these approaches. If we use the metrics from Dupont et al. we are able to detect all types of attacks with a false positive rate of zero for both data sets. As the checks can be executed in parallel and need little resources the system is able to react in real-time. With our demonstrator, we measured the time needed to process the messages and detect the attack. In average the processing takes 30 µs, while the detection time depends, as mentioned before, on the criterion. For criteria, which can decide directly on a message, the detection time is 30 µs, for the frequency-based criteria this is in average 3 ms. The measured processing time of our implementation

is much faster than the 0.53 ms stated by Zhang et al. (2018) as well as we beat the detection time of Tariq et al. (2020), which is stated as 0.073 s.

5 Conclusion

The proposed approach achieves a detection rate of over 99.99% while generating zero false alarms and considering a wide range of attack types. This makes the approach very attractive for practical usage.

Another result of this research is the question: how good are the publicly available and popular data sets of the TU and the HCRL to evaluate the quality of IDS? Since these data sets can be analyzed almost perfectly with so few, simple rules, these data sets are often used for comparison, especially for approaches based on machine learning (Dupont et al. 2019). The data set from TU Eindhoven in particular does not seem to be suitable for machine learning approaches, as they contain very few messages representing attacks in relation to the total data set.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Buttigieg R, Farrugia M, Meli C (2017) Security issues in controller area networks in automobiles. In: 2017 18th international conference on sciences and techniques of automatic control and computer engineering (STA). IEEE, pp 93–98

Checkoway S, McCoy D, Kantor B, Anderson D, Shacham H, Savage S, Koscher K, Czeskis A, Roesner F, Kohno T (2011) Comprehensive experimental analyses of automotive attack surfaces. In: Proceedings of the 20th USENIX conference on security, SEC'11. USENIX Association, USA, p 6

Dupont G, Hartog J, Etalle S, Lekidis A (2019) Evaluation framework for network intrusion detection systems for in-vehicle can. pp 1–6. https://doi.org/10.1109/ICCVE45908.2019.8965028

Dupont G, den Hartog J, Etalle S, Lekidis A (2019) A survey of network intrusion detection systems for controller area network. In: 2019 IEEE international conference of vehicular electronics and safety (ICVES). IEEE, pp 1–6

Dupont G, Lekidis A (2019) Automotive controller area network (can) bus intrusion dataset v2. https://data.4tu.nl/repository/uuid:b74b4928-c377-4585-9432-2004dfa20a5d

Hanselmann M, Strauss T, Dormann K, Ulmer H (2020) Canet: an unsupervised intrusion detection system for high dimensional can bus data. IEEE Access 8:58194–58205

Hartzell S, Stubel C (2017) Automobile can bus network security and vulnerabilities

Hoppe T, Kiltz S, Lang A, Dittmann J (2007) Exemplary automotive attack scenarios: Trojan horses for electronic throttle control system (ETC) and replay attacks on the power window system. VDI BERICHTE 206:165

Koscher K, Czeskis A, Roesner F, Patel S, Kohno T, Checkoway S, McCoy D, Kantor B, Anderson D, Shacham H, Savage S (2010) Experimental security analysis of a modern automobile. In: 2010 IEEE symposium on security and privacy, pp 447–462. https://doi.org/10.1109/SP.2010.34

Lee H, Jeong SH, Kim HK (2018) Can dataset for intrusion detection (OTIDS). http://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset

Lokman SF, Othman AT (2019) Intrusion detection system for automotive controller area network (can) bus system: a review. EURASIP J Wirel Commun Netw 1:184

Markovitz M, Wool A (2017) Field classification, modeling and anomaly detection in unknown can bus networks. Veh Commun 9:43–52

Miller C, Valasek C (2013) Adventures in automotive networks and control units. Def Con 21:260–264

Miller C, Valasek C (2014) A survey of remote automotive attack surfaces. Black Hat USA 2014:94

Miller C, Valasek C (2015) Remote exploitation of an unaltered passenger vehicle. Black Hat USA 2015:91

Nie S, Liu L, Du Y (2017) Free-fall: hacking tesla from wireless to can bus. Brief Black Hat USA 25:1–16

Ruth R, Bartlett W, Daily J (2012) Accuracy of event data in the 2010 and 2011 Toyota Camry during steady state and braking conditions. https://doi.org/10.4271/2012-01-0999

Song HM, Kim HR, Kim HK (2016) Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network. https://doi.org/10.1109/ICOIN.2016.7427089

Tariq S, Lee S, Kim HK, Woo SS (2020) CAN-ADF: the controller area network attack detection framework. Comput Sec 94:101857

Taylor AH, Leblanc S, Japkowicz N (2016) Anomaly detection in automobile control network data with long short-term memory networks. In: 2016 IEEE international conference on data science and advanced analytics (DSAA), pp 130–139

Verma ME, Iannacone MD, Bridges RA, Hollifield SC, Kay B, Combs FL (2020) ROAD: the real ORNL automotive dynamometer controller area network intrusion detection dataset (with a comprehensive CAN IDS dataset survey and guide)

Wang Q, Lu Z, Qu G (2018) An entropy analysis based intrusion detection system for controller area network in vehicles. In: 2018 31st IEEE international system-on-chip conference (SOCC), pp 90–95. https://doi.org/10.1109/SOCC.2018.8618564

Weber M, Klug S, Zimmer B, Sax E (2018) Embedded hybrid anomaly detection for automotive can communication. In: Proceedings of the 9th European congress on embedded real time software and systems (ERTS 2018)

Woo S, Jo HJ, Lee DH (2015) A practical wireless attack on the connected car and security protocol for in-vehicle can. IEEE Trans Intell Transp Syst 16:993–1006

Wu W, Huang Y, Kurachi R, Zeng G, Xie G, Li R, Li K (2018) Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks. IEEE Access 6:45233–45245. https://doi.org/10.1109/ACCESS.2018.2865169

Zhang L, Shi L, Kaja N, Ma D (2018) A two-stage deep learning approach for can intrusion detection. In: Proceedings of the ground vehicle systems engineering and technology symposium (GVSETS), pp 1–11