Real Vehicle-Based Attack Dataset for Security Threat Analysis in a Vehicle

Yeji Koh, Yoonji Kim, Munkhdelgerekh Batzorig, and Kangbin Yim(✉)

Department of Information Security, Soonchunhyang University, Asan, South Korea
{julysnowflake,rladbswl7,munkhdelgerekh,yim}@sch.ac.kr

Abstract. Modern Connected and Autonomous Vehicles (CAVs) are equipped with an increasing large number of Electronic Control Units (ECUs). These ECUs transmit Control Area Network (CAN) protocol in data exchange via an in-vehicle network. Since CAN message uses broadcast transmission, it becomes hinder to detect the intrusion. Thus, building an intrusion detection system (IDS) and intrusion prevention system (IPS) to IVN is essential, and most effective method for those system is effective usage of deep learning (DL) or machine learning (ML); Nevertheless, building an IDS or IPS based on DL or ML is required huge amount of data of normal state and during the attack. Consequently, there is a high demand for an accurate normal and attack dataset for IVN protection. Therefore, we propose accurate environment configuration for attack data collection from the vehicle.

1 Introduction

As the development of the modern automobile industry, the number of electronic control units (ECUs) that maintains the vehicle has increased rapidly, increasing the complexity of communication interfaces. Controller Area Network (CAN) protocol is more efficient than Conventional Universal Asynchronous Receiver/Transmitter (UART) methods. Thus, CAN protocol is the most frequently used protocol in IVN for transmitting and receiving data from ECUs. Since CAN protocol uses broadcast transmission as to communicate with other nodes, it is hinder to apply an authentication mechanism to IVN. Thus, if an attacker unauthorized access through the CAN bus and manipulates data, it is impossible to distinguish between the error message and the actual message, which can lead a great risk [1]. The possible attack models through the CAN bus are same as follows:

Injection Attack. Through OBDII and ECU systems, it enters the network inside the vehicle and attacks by injecting attack data. The CAN network cannot identify whether the received frame is legitimate because it does not have an authentication mechanism.

DoS (Denial of Service) Attack. Since the CAN protocol uses broadcast transmission, there is no source or destination address, and all messages are received synchronously to one ECU. To avoid this collision, CAN protocol uses ID field to calculate priority of order of the messages. If message has low ID value, it represents a higher priority, if there

L. Barolli (Ed.): BWCCA 2022, LNNS 570, pp. 137–146, 2023.
https://doi.org/10.1007/978-3-031-20029-8_13

is a frame with the highest priority on the CAN bus, other nodes cannot send message frames to the IVN until high priority message is served. Thus, the attacker always sets the highest priority frame (0x00) to be transmitted to the in-vehicle network to attack in a form in which other nodes are placed in the standby state [2].

Fuzzing Attack. Fuzzing attacks are similar to indiscriminate attacks in the form of randomly injecting ID values and data into CAN networks. Attack data can be easily injected using the characteristics of the CAN network, where authentication is not performed on the transmission and reception nodes, which can cause an unexpected situation while driving the vehicles [3].

Replay Attack. The attack is executed when the vehicle injects a message that causes a specific function or behavior different from the current state into the message cycle of the existing state [4].

Spoofing Attack. A spoofing attack is a hacking attack that an adversary approaches on a network pretending to be an authorized address of the system to gain unauthorized access [5].

The ECU transmits data by sending and receiving CAN messages thorough the CAN network. If adversary such as Dos, Bus-off, and Fuzzing are activated due to the vulnerability of the CAN, data on the CAN cannot be read or written, resulting in serious consequences that damage to the ECU or even threaten the safety of the driver. In this paper, using the vulnerability of CAN, we select a total of three attack models for real vehicles: DOS, Fuzzing, and Replay to show the effect on the vehicle injected with the attack message, and collect attack datasets generated through the attack. The collected attack datasets can be used to develop Intrusion Detection System (IDS) for vehicles and machine learning for vehicles to prevent security threats in preparation for future attacks on actual vehicles. The paper is organized as follows. Section 2 explains similar research works about possible attacks on IVN. Section 3 explains methods and configuration of environment for CAN attack datasets collection, and tools for inject data into vehicle internal networks. Section 4 provides detailed reactions of the vehicle when data is injected. Finally, the conclusions and future works are drawn in Sect. 5.

2 Related Work

In the vehicle, various ECUs are integrated into the CAN bus system to exchange CAN messages. CAN does not apply security for various attacks, so several vulnerabilities are found, and Tianxiang Huang et al. analyze attack models for CAN buses and design and develop ATG (Attack Traffic Generation Tool) to explain key features. ATG can define the attack types, content, set timer-based execution and build an effective function GUI using wxPython library. Types of attacks include Dos, Fuzzing, Spoofing, etc. DoS attack send messages every 0.8 ms from 2 s to 4 s, and a series of random messages from 8 s to 40 s for fuzzing attacks. ATG that the proposed method can share and reuse attack scenarios for CAN bus security testing and supports CAN database conversion capabilities.

In addition, it is possible to migrate to CAN networks as well as other network communication protocols such as CAN-FD, Flex-Ray, and Ethernet. The authors construct the attack via connection with On-Board Diagnostics (OBD) using real cars and inject CAN messages directly to show the speedometer decreasing to zero. We confirm that the recently released OBD ports of vehicles have a limited type of CAN network used for analysis and attack, and in this paper, we test and prove whether attack injection for various CANs is possible using the EDA (CAN gateway ECU Direct Approach) method [6].

Kazuki Iehira et al. propose a method of transitioning to the bus-off state by intentionally setting the message transmitted by the ECU to an error based on a spoofing attack method in which the receiving and transmitting ECU does not detect anomalies in the applied ECU using a bus-off attack. The CAN message is a broadcasting method, and since there is no transmission address and only the CAN ID exists as the destination address, spoofing attacks are easily possible. Three methods are presented, and the results are shown: bus-off attack using bit errors, bus-off attack using stub errors, and bus-off attack using one frame. This paper consisted of attack hardware and target ECU for CAN bus experiments and used Field-Programmable Gate Array (FPGA) to inject spoofing messages. The spoofing target was the engine speedometer of a stationary vehicle, and in the case of simple spoofing, the speedometer's pointer fluctuates, but the proposed method of spoofing using bus-off attacks in this paper did not detect errors and no anomalies in the vehicle used in the experiment. They validate the attack in a simulated environment consisting of attack hardware and ECU and evaluate the impact of spoofing attacks effectively implemented on real cars [7].

Cyber-attacks on vehicles are likely to cause casualties. Shahida Malik and Weiqing Sun conduct an analysis of cyberattacks on connected and autonomous vehicles and simulate cyberattacks based on the analysis to show the actual damage impact. They used CARLA 0.9.6 for simulation and controlled the environment differently depending on the scenario. They simulated a total of 10 significant and high-priority attack scenarios and recorded important functions, including collision, position at intersections, speed, gear, position, orientation, and acceleration. They apply scenario that wireless update attacks, cyber terror attacks, mobile app attacks, OBD dongle attacks, etc., it shows results such as stop the car engine or potentially stop the car moving through Bluetooth and disable emergency support services. We confirm that these attacks are fully simulated, and we evaluate the quality of the attack dataset by injecting and testing attack messages in direct-driving vehicles, verifying their impact on the vehicle, and collecting attack datasets [8].

3 Method of Generate Attack CAN Dataset Based on Actual Vehicle

3.1 Method for Collection CAN Dataset in Vehicle

In this paper, we used to EDA (CAN gateway ECU Direct Approach) method to collect CAN dataset in vehicle [9]. EDA is method to access the CAN gateway ECU through ECU's tapping line and collect actual vehicle internal network data by using CAN dataset collection tools (Fig. 1).

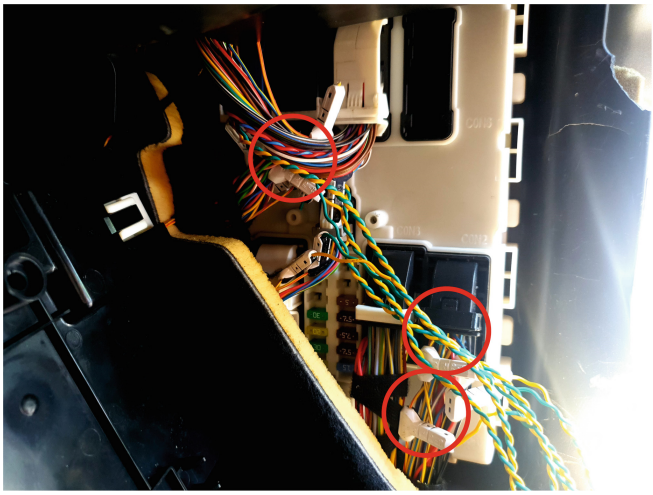

Fig. 1. Shows the actual vehicle tapping into the CAN gateway module using the EDA method. The CAN network connection is for K-CAN, K-CAN2 and PT-CAN, and each CAN is responsible for body control, multimedia function control, etc.

Mostly Much internal network research of vehicle uses OBDII port to access the network. However, because there are located just the diagnostic CAN data in OBDII port, it hard to check entire CAN bus of the vehicle. So, limits of check and research to diverse vehicle functions always exist.

On the other hand, the EDA method can collect more diverse function data of vehicle than collection data using OBDII port. By using this method, we could collect more accurately performed of detailed analysis and attack to target CAN data for a specific function. Also, it is possible to collect more accurate and larger amount of data than the data of OBDII port because EDA method can access to entire CAN bus network. In terms of vehicle internal network research, this method makes the result more effective and accurate than previous research.

3.2 Vehicle Internal CAN Network Attack Tool

In order to analyze the threat of the vehicle's internal network, high-quality attack data is required for researchers. The high-quality attack data means data that is difficult to detect because the attack data is similar to the actual driving data. If the attack data is similar to the actual driving data, it is difficult to find anomalies in contrast to the actual data, making it difficult to detect them with Intrusion Detection System (IDS).

To improve the existing IDS of the vehicle, we developed more sophisticated internal network attack tools to generate effective attack data. In this paper, we implement attack tools against Fuzzing, DoS, and Replay attacks. Figure 2 below is a schematic diagram of the attack process by injecting an attack data packet into an actual vehicle.

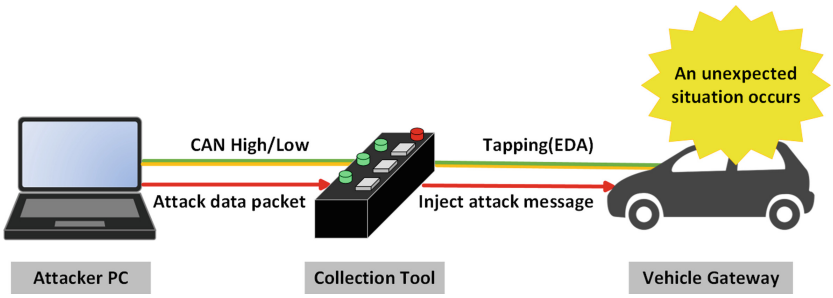

Fig. 2. Actual vehicle-based attack dataset injection process

In this paper, we developed an internal CAN network attack tool based on the characteristics of each attack. The purging attack tool is set to inject arbitrary data periodically, allowing attackers to specify the range of arbitrary data values, the range of IDs, the number of injections, and the length of arbitrary data. If the attacker runs the purging tool, a random dataset is injected and malfunctioning.

When the DoS attack tool is executed, a large number of high-priority IDs are injected within a short time. As a result, the operation of the data packet having a low priority is blocked.

3.3 Collection of Attack CAN Data Packet Based on Actual Vehicle

Using our own developed attack tools, we collected attack CAN data packets based on real vehicles. In order to collect the IVN attack data set of the real vehicle, line tapping was performed on the CAN gateway module using the EDA method. When collection begins, the vehicle's CAN data packet is collected through the tapping line of the CAN gateway module. In addition, the attack data packet is injected into the CAN network of the real vehicle using the tapping line.

The CAN network attack based on the actual vehicle is performed during driving. In the case of a purging attack, basic functions and eventuality functions are activated through random value data packet injection. Dos attacks inject attack data packets to activate certain eventuality features.

In this study, attacks were carried out during driving on CAN networks mainly related to vehicle control and visual parts. In preparation for unexpected situations, attacks on body-related functions took place only at rest. If an attacker attacks a large amount of data while collecting driving data packets, a collision between the collection data and the attack data may occur. In this case, an appropriate attack packet may not be generated. Therefore, to avoid this problem, we made a scenario in which attacks are performed at regular intervals for a short time.

The attack scenario is as follows (Table 1):

Table 1. Actual vehicle-based CAN internal network attack scenario

Attack	Scenario
Fuzzing	- Inject 100 data every 2 min for 3 s - Inject 500 data every 2 min for 3 s - Inject 100 data every 2 min for 5 s - Inject 500 data every 2 min for 5 s
DoS	- Inject 5,000 data every 2 min for 2–3 s - Inject 10,000 data every 2 min for 2–3 s
Replay	- Replay pre-collected injection data for 5–6 s

All attack scenarios were carried out only in certain sections where the risk of accident rate was low. To prevent collision between each data packets, we use to scenarios in which 100 to 500 data are injected for 3 to 5 s at 2-min intervals in Fuzzing attack. In case of DoS attack, 5,000 to 10,000 data inject for 2 to 3 s at 2-min intervals. And Replay attack inject pre-collected data for 5 to 6 s at 5-min intervals. By injecting CAN attack data packets based on actual vehicle, the vehicle's functions such as emergency light and warning light on the instrument panel were activated. And function that could be contacted by the manufacturer in an emergency situation was successfully attacked (Fig. 3).

Fig. 3. Perform RDC initialization via fuzzing attack on real vehicle

Collecting CAN network attack data from real-world vehicles allows us to identify potential attack risks in real-world situations rather than data collected from virtual environments or stationary vehicles. You can also attack features that cannot be performed on a simple test bed. It is possible to analyze data related to more functions by analyzing data collected during an actual vehicle attack. And it can have an effective impact on security threat detection and analysis studies on vehicle interior networks.

4 Evaluation

In this section, we describe the reactions of vehicle functions when we inject the attack datasets into the vehicle. We were able to develop our own program for each attack using P-Code library that provides from peak-system [10].

4.1 Result of DoS Attack

The nature of DoS attack is slow down or shut down a functions or network, making it inaccessible to its intended node. We accomplished this attack by flooding IVN with high priority ID message which is 0x00 ID. As result, we were able to check functions like frequency of signal light, dashboard information etc., are slowing down, and could check time offset of messages that collected from the vehicle.

```
from PCANBasic import *
import time
def DoS_Attack():
    time_offset = 0.03
    DoS_attack_data = (0x00, 0x04, 0x81, 0x00, 0xC0, 0x02,
0x20, 0x4D)
    DoS_attack = TPCANMsg()
    DoS_attack.ID = int(0x000)
    DoS_attack.LEN = 8
    DoS_attack.MSGTYPE = PCAN_MESSAGE_STANDARD
    DoS_attack.DATA = DoS_attack_data
    for i in range(0, 10000):
        CAN.Write(CAN_BUS, DoS_attack)
        time.sleep(time_offset)
CAN = PCANBasic()
CAN_BUS = PCAN_USBBUS2
CAN.Initialize(CAN_BUS, PCAN_BAUD_500K, 2047, 0, 0)

while True:
    DoS_Attack()
```

4.2 Result of Fuzzing Attack

The main purpose of fuzzing attack is to exploit the possible vulnerabilities or function messages. We initiated fuzzing attack by injecting messages with random ID and Data. We have developed program for this attack. With this attack, we were able to inject data as scenario follows, and same time we could save injected data into another text file for data labeling. As mentioned in Sect. 3, we collected attack dataset while driving in highway, and at the same time, we had injected data into the vehicle, because of this, we carried out the risk that threatening to driver's life, because of the risk, we have only injected data to K-CAN which mainly assigned for controllers of the vehicle's dashboard. After the fuzzing attack, we found following function's data, and store in excel file (Table 2).

Table 2. Vehicle function findings after fuzzing attack

Function	Description
Speed in dashboard	Change the speed measurement in dashboard
RPM	Change RPM measure in dashboard
Gear (PARK)	Change the gear to parking mode
Gear (Drive)	Change the gear to drive mode
Signal Light (Left)	Change the status of turn left signal light to on
Signal Light (Right)	Change the status of turn right signal light to on
Emergency Light	Change the status of emergency light to on
Open door	Open all doors
Front ultra-sonic wave	Give a signal of detected to radar sensor. (Give Max, Medium and Low)
Back ultra-sonic wave	Give a signal of detected to radar sensor. (Give Max, Medium and Low)

4.3 Result of Replay Attack

Main intention of replay attack is fraudulently delays or resends it to misdirect the receiver into doing what the adversary wants. Thus, adversary needs information about the system, to initiate a replay attack. After injecting fuzzing attack successful, we have obtained some message formats that assigned for certain functions. Based on this information, we could initiate replay attack by injecting those messages, and control vehicle's some functions. For replay attack on the vehicle, we have developed attack tool with user interface using P-CAN library of python. Replay attack program has main two roles. The first one is to inject the data from list that gained functional messages after fuzzing attack. The other one is to save injected time of messages for later data labelling (Fig. 4).

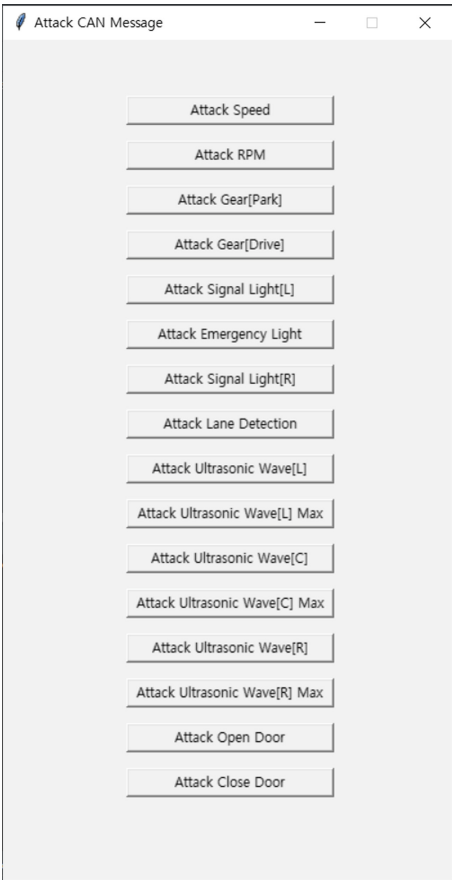

Fig. 4. Display of the tool for replay attack to IVN

5 Conclusion and Future Work

In modern automobiles, the number of ECUs mounted inside the vehicle is increasing exponentially for the safety and convenience of drivers. However, the need for vehicle security has also increased as vehicle attack methods with serious consequences that threaten the driver's life continue to expand. Therefore, an accurate analysis of the type of attack that occurred is required, and it is important to recognize that there is a high possibility of an attack on the vehicle, and to anticipate and prepare for an attack on the vehicle.

In this paper, DoS, Fuzzing, and Replay attacks were performed on actual vehicles driving with 7 scenarios. As a result of the experiment, the vehicle window was opened, the emergency SOS button was activated, and navigation reboot etc., this resulted in an unexpected situation in which the driver could be embarrassed while driving. In addition, we collected CAN data from the ECU by selecting the existing EDA method and made attack tools using the characteristics of each attack. In real-world vehicles,

attack messages were injected into the CAN bus to generate high-quality attack datasets that can later be used as materials for vehicle security. In the future, we will apply our attack dataset into IDS or IPS system, and compare performance with other public attack dataset.

Acknowledgments. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A4A2001810) and This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-01343, Regional strategic industry convergence security core talent training business).

References

1. Hong, S.: Research on countermeasures of controller area network vulnerability. J. Converg. Inf. Technol. **8**(5), 115–120 (2018)
2. Biron, Z.A., Dey, S., Pisu, P.: Real-time detection and estimation of denial of service attack in connected vehicle systems. IEEE Trans. Intell. Transp. Syst. **19**(12), 3893–3902 (2018). https://doi.org/10.1109/TITS.2018.2791484
3. Mukherjee, S., Shirazi, H., Ray, I., Daily, J., Gamble, R.: Practical DoS attacks on embedded networks in commercial vehicles. In: Ray, I., Gaur, M.S., Conti, M., Sanghi, D., Kamakoti, V. (eds.) ICISS 2016. LNCS, vol. 10063, pp. 23–42. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49806-5_2
4. Chandrasekaran, S., Ramachandran, K.I., Adarsh, S., Puranik, A.K.: Avoidance of replay attack in CAN protocol using authenticated encryption. In: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6 (2020). https://doi.org/10.1109/ICCCNT49239.2020.9225529
5. Yang, Y., Duan, Z., Tehranipoor, M.: Identify a spoofing attack on an in-vehicle CAN bus based on the deep features of an ECU fingerprint signal. Smart Cities **3**, 17–30 (2020). https://doi.org/10.3390/smartcities3010002
6. Huang, T., Zhou, J., Bytes, A.: ATG: an attack traffic generation tool for security testing of in-vehicle CAN bus. In: Proceedings of the 13th International Conference on Availability, Reliability and Security (2018)
7. Iehira, K., Inoue, H., Ishida, K.: Spoofing attack using bus-off attacks against a specific ECU of the CAN bus. In: 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE (2018)
8. Malik, S., Sun, W.: Analysis and simulation of cyber attacks against connected and autonomous vehicles. In: 2020 International Conference on Connected and Autonomous Driving (MetroCAD). IEEE (2020)
9. Koh, Y., Kim, S., Kim, Y., Oh, I., Yim, K.: Efficient CAN dataset collection method for accurate security threat analysis on vehicle internal network. In: Barolli, L. (ed.) IMIS 2022. LNCS, vol. 496, pp. 97–107. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-08819-3_10
10. https://documentation.help/PCAN-Basic/PCAN-Basic_Documentation.html