Simulation of In-Vehicle Network CAN (In)Security

Nur Fadhilah Ajwa, Nur Nadiah Ali Azmi, Hafizah Mansor
Department of Computer Science, Kulliyyah of ICT, International Islamic University Malaysia, Selangor, Malaysia.
*Corresponding author hafizahmansor@iium.edu.my

Abstract— In-vehicle network security is becoming one of the leading research fields in the cybersecurity area. Modern automobiles consist of Electronic Control Units (ECUs), microcontrollers that control the operations of a vehicle. These ECUs are mainly interconnected through an in-vehicle Controller Area Network (CAN), a message-based protocol that allows communication between different ECUs. The vulnerabilities of CAN which have no encryptions, authentication and integrity checking attributes are widely known but continuous research is made on the security and exploitation of CAN. This is because, the safety and security of vehicles and passengers have become more concerning due to the increase of external and internal vehicle communications. Our paper aims to discuss related works of CAN bus vulnerabilities and security through literature review. This paper conducts simple cyberattacks against CAN bus by data collection, data analysis and attack experiment on the Instrument Cluster Simulator (ICSim) with the help of CAN network analysis tool, CANalyse. CAN packets were collected and analysed and the ID for a number of car functions in ICSim were determined. Attack experiments resulted in sniffing and replay attacks against CAN buses were valid. Finally, from these findings, the paper proposes recommended security measures of CAN bus which are network segmentation, cryptography-based method, and Intrusion Detection System (IDS).

Keywords— CAN bus, in-vehicle network, CAN bus security, cybersecurity.

I. Introduction

In-vehicle security has been a sensitive issue that affects transportation users and manufacturers as cyberattacks targeting in-vehicle networks can result in threats affecting passenger safety. An automobile is equipped with more than 70 electronic control units (ECUs) that are controlled via several networks which are Controller Area Network (CAN), Local Interconnect Network (LIN), FlexRay and a few other networks [1]. CAN is the most popular protocol among all the in-vehicle networks and is mostly used in all types of vehicles in the market. CAN bus is a broadcast network communication protocol that is commonly used in-vehicle communication protocol compared to other network protocols as CAN bus offers advantages such as cost-effective wiring, immunity to electrical interference, self-diagnosing as well as error correction based on network protocol characteristics [2]. It was developed by Robert Bosch in the 1980s that provides up to 1 Mbps in a classical CAN interface. CAN Flexible Data rate (FD) interface is able to achieve 5 Mbps with 64-byte payload.

The principles of CAN are categorized in three features which are bus topology, multi-master and arbitration of transmission right. CAN is widely used in bus topology. Two or more ECUs are connected to a communication line. Besides that, each node directly transmits messages on the CAN bus when needed, making it easier to add CAN messages and nodes, making the CAN bus a multi-master. If more than one node transmits a message on the CAN bus simultaneously, the CAN-ID determines the transmission authority. After arbitration, the CAN message with the highest priority message is sent first. As a result, messages with a lower priority are transmitted until a message with a higher priority has been sent [1].

Engines and body control modules such as gears, speed, brakes, and others are all critical parts of a vehicle that connect to the CAN bus controller. The CAN protocol consists of the classical CAN and CAN FD protocols, but both protocols are defined and standardised under the ISO 11898 series [1].

CAN itself lacks security support which exposes it to the risk of being attacked or harmed. Due to its characteristics that have maximum of 1 Mbps data transfer rate and maximum of 8 bytes payload of a message, security methods designed for consumer products are difficult to be used directly by CAN. CAN protocol has great weaknesses. CAN packets are sent to all nodes physically and logically. CAN bus network is not segmented and the traffic on the CAN bus is not encrypted. As broadcast mechanism is used in the CAN bus protocol, a malicious component on the network can easily snoop on all communications or send packets to all nodes in the CAN network. In addition, CAN protocol is vulnerable to denial-of-service (DoS) attacks as it makes other CAN nodes to back off owing to CAN's priority-based arbitration scheme that allows a node to maintain a "dominant" state on the bus. On top of that, CAN packets do not contain authenticator fields that make any component send an identical packet to any other components. It makes any single component able to control all other components since they lack a defence mechanism [3]. As a final point, it has limited bandwidth and payload, hence CAN buses are unable to provide strong access control. The high-speed CAN bus has a data rate of 500 Kbit/s and the payload is only up to 64 bits. Since the length and responses are too short, the adversaries can crack the key of an ECU within eight days through a brute-force attack [2].

CAN bus is a broadcast network allowing the capture of all the messages through the network [10]. Since broadcast data is not encrypted, attackers can get one's hand on the desired data, which will lead to breaching of privacy. Denial of Service (DoS) and injection attacks are common attacks for cars. As a result, false meter readings, brake function disabled, non-functional lights, gears and other various parts of vehicles can be controlled effectively by the attackers.

This paper aims to investigate works on CAN bus and its vulnerabilities by directly communicating with the CAN bus of an Instrumental Cluster Simulator (ICSim) [11] through a tool designed by Kartheek Lade called CANalyse tool [8]. Other than that, we aim to analyse the log files of the CAN bus of ICSim to identify the CAN IDs for a number of car functions. Lastly, using the CANalyse tool, we intend to conduct simple attacks such as replay and sniffing attacks on ICSim. Hence, through the observation from the outcome of these experiments is to provide security measure recommendations for the CAN bus. The contribution of this paper is to prove the vulnerability of the CAN bus through the simulation attacks.

II. RELATED WORK

Review and discussion of the related work is presented in this section. We have studied a few research works over the years related to in-vehicle network CAN security. The researchers use many methods to document information about in-vehicle network CAN security. Common methods are practical CAN security evaluation tools, direct and indirect attacks on CAN buses, risk analysis and more.

Zhang et al. conducted their research by introducing a CAN security evaluation tool called CANsec [2]. To evaluate the security risk of the in-vehicle CAN relevant to the purpose of this tool, the tool simulates malicious attacks according to major attack models. The authors started by defining six vulnerabilities of CAN, which are no encryption, no authentication, no integrity checking, broadcast transmission, priority-based arbitration, and limited bandwidth and payload based on the analysis of security characteristics of CAN. Then, they further introduced the four basic attack vectors which are eavesdrop, replay, impersonation and injection attacks. Eleven evaluation vectors that can build an attack model for the target assets and simulate the actual attack scenarios were proposed from the mentioned attack vectors and seven major assets in the CAN network. The experiment to evaluate the performance of the tool was conducted without information from the manufacturer and the structure of the CAN bus in a Ford vehicle was found. Moreover, they conducted replay and fuzzy attacks on the Ford vehicle. The research found that the replay attacks were valid on CAN buses and abnormal displays appeared on the dashboard of the vehicle under the fuzzy attacks. It is also discovered that the instrument panel has a defence mechanism for handling message conflicts but also can introduce the risk of DoS. They concluded that their tool could evaluate the security of the in-vehicle CAN. The strength of the research is the tool, CANsec, supports 11 evaluation vectors that target various assets of in-vehicle networks, which comprehensively evaluates the security of CAN bus. Other than that, reversing the CAN traffic is a key feature of CANsec that other tools do not have. CANsec also can monitor the change in vehicular status, log the evaluation activity, and allow users to configure the evaluation flexibly after selecting the evaluation vectors.

Next, a study by Koscher et al. highlighted an experimental security analysis of modern automobiles by focusing on analysing two 2009 automobiles of the same make and model [3]. Using the same model cars is for differential testing and to validate that their results were not tied to one individual vehicle. They introduced carShark as an injection tool and to sniff the CAN bus to demonstrate and examine the insecurity of the underlying system structure. This research described two types of vectors where one might gain access to a car's internal network which are through physical access such as OBD-II port and numerous wireless interfaces. In this study, attack methodology was introduced by authors which are packet sniffing and targeted probing, fuzzing and reverse engineering. The authors found that the attacks mentioned were easy to execute and the existing automotive systems during the time of the research are very fragile. Many kinds of attacks were conducted, and they successfully proved that unsafe conditions can be created as they were able to directly manipulate a few tested safety-critical ECUs. The authors concluded that all communications can be easily snooped or packets can be sent to any other nodes on the network of CAN with malicious components. The findings showed that CAN protocol is extremely vulnerable to denial-of-service attacks. Any component can indistinguishably send a packet to any other element. The research has the advantage as the carSHARK is a custom CAN bus analyser and packet injection tool that manages to inject several attacks to check for vulnerabilities. However, carSHARK is an outdated tool and probably is not relevant for the latest car to be used for injection of attacks. More recent research with the latest tool must be conducted to see changes in the result.

The third research paper emphasized the risk, threats, and vulnerabilities of a CAN bus network analysis using Failure Modes and Effects Analysis (FMEA). Mansor et al. imitated the communications between the wheel rotation and the odometer on the instrument panel cluster (IPC) of a car [4]. Three different setups of experimental attack with two different nodes, i.e., original and attacker nodes were executed. In the first basic setup, only the two nodes are involved without any security implementation. The second setup included Message Authentication Code (MAC), and the last setup included MAC and Advanced Encryption Standard (AES). The result from the experiment of basic setup shows that the CAN bus network is vulnerable to attacks such as sniffing, denial of service, message manipulation, and many others because the IPC node accepted the attacker node messages. Meanwhile, the second setup shows that the MAC introduced authenticity and integrity to the network but sniffing attacks still could happen. The third setup shows that AES introduced confidentiality to the network adding another security property to the earlier result before. However, the same message can still be sent by the attacker by replaying the message. The strength of this research through the experiment and methodology is, it widely covers the aspect of in-vehicle CAN network security. This research not only analysed the threats, vulnerabilities, and attacks but also the potential failure modes in the general life pattern of a vehicle and the substances in question. However, this method might fail or give contrasting results for different ECUs and operations of a real car.

Palanca et al. proposed a DoS attack that does not involve the transmission of full CAN messages as it overwrites the recessive bits and generates a transmission error [5]. The authors mentioned that its execution is undetectable via frame-level analysis. To test the efficiency of the attack, an on-bench attack test is executed with an imitation of the CAN network. The test is implemented with two different settings in which the first setup involves two nodes exchanging messages with each other without the attacking device. Moreover, an attacking device is connected to the CAN bus and tries to send the CAN frames to the target for the second setup. The authors also executed reliability measurements of the attacking device. Additionally, on-vehicle testing was executed by implementing the attack on an unmodified 2012 Alfa Romeo Giulietta through the OBD II port. The research is concluded by proposing and comparing possible countermeasures for detecting and preventing such attacks, such as network segmentation, network topology alteration, diagnostic port access control, and more. The findings of the research are that the on-bench attack test resulted in the attacking devices managed to correctly terminate the target frames in which the receiving nodes were not able to retrieve the message for both cases. The reliability measurement resulted in 99.9974% accuracy. Furthermore, the result of on-vehicle testing is that the parking sensors immediately stopped working altogether. The authors concluded that the CAN network can be disrupted even with simple tools when any person has physical access to the network. The authors suggested that non-vulnerable protocols need to be used to prevent this kind of attack in automotive networks. The strength of this research is it is an undetectable attack method with frame-level analysis. This research also provides clear evidence of the vulnerability of CAN buses.

The work by Payne is the closest reference to what we are trying to achieve in our research [6]. The paper described the implementation of a hands-on ethical car-hacking and demonstrated a replay attack on a simulated controller area network (CAN). The author introduced an open-source toolkit for car hacking called Instrument Cluster Simulator (ICSim), which relies on Linux tools. The steps for installation, implementation and running the ICSim software were explained by the author. CAN network sniffer (cansniffer)included in the CAN utility software (can-utils) that can work with the ICSim was used to view packets on the virtual CAN network interface and observe the communication between ECUs in a simple CAN bus. Additional tools in can-utils to capture and replay the CAN bus packets were used to successfully conduct replay attacks on the ICSim software. The paper also introduced the reverse engineering CAN bus messages method focused on the turn signals. Complete control of the turn signals in a simulated vehicle was achieved.

Based on the literature review conducted above, it is concluded that a variety of methods are used to study and evaluate the security of in-vehicle network CAN. Vehicle exploitation through CAN bus can be inferred as the common objective that is to be achieved in previous works. And thus, many were able to prove the vulnerabilities of the in-vehicle network CAN through vehicle exploitation.

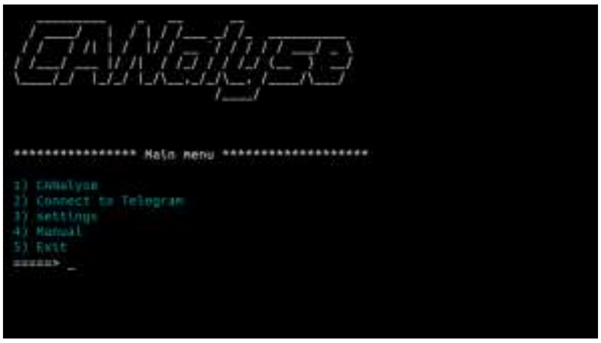
III. METHODOLOGY

Methods including the tools used in our research is discussed in this section. After the literature review, the experiment is designed to be able to prove the vulnerability of the CAN bus.

A. Data Collection and Data Analysis

Instrument Cluster Simulator (ICSim) [11] is a simulator for some of the main car functions which are turn signals, power doors and speedometer operated through the CAN bus. It runs on Linux with simple commands [7].

CANalyse is a tool built to analyse the log files of CAN traffic to find out unique data sets automatically connected with simple user interfaces such as Telegram [8]. It is also able to exploit the vehicle by recording, analysing and replaying the network traffic or log files through a Telegram-bot. The interface of CANalyse is shown in Figure 1.



```
****************** Main menu ********************

1) CANalyse
2) Connect to Telegram
3) settings
4) Manual
5) Exit
=====> _
```

Fig. 1 Interface of CANalyse Tool

The process of data collection is conducted by making a connection between the virtual CAN interface, vcan0 of ICSim, with CANalyse. After the connection is established, CAN frames are immediately collected and displayed with the candump command from the can-utils tool [11] of ICSim as shown in Figure 2. It is a Linux specific set of utilities that enables Linux to communicate with the CAN network on the vehicle.

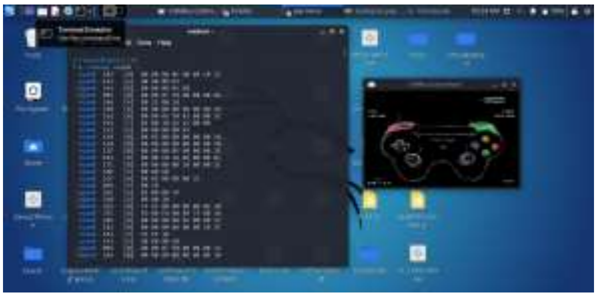


Fig. 2 CAN frames of ICSim traffic through CANalyse Data Analysis

Then, data analysis was conducted by analysing and identifying the CAN IDs for certain car functions provided in the simulated vehicle, ICSim. Using the ICSim controller, we created a CAN traffic to be recorded and then analysed to identify the CAN IDs by simply sending commands from Telegram-bot.

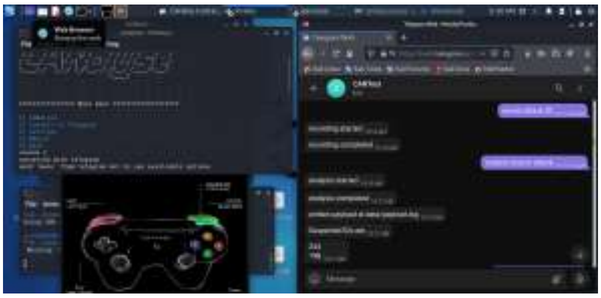


Fig. 3 Identified CAN IDs for door function in ICSim from Telegram-bot.

The process is repeated to discover the CAN IDs for the car functions in ICSim as shown in Figure 3. Table 1 shows the IDs associated with the three (3) car functions in ICSim obtained from the analysis.

Table 1. Identified CAN IDs

ID	Functions
188	Turn signal (right/left)
244	Speedometer
19B	Power doors (lock/unlock)

B. *Attack Experiment*

There are several types of attacks against the CAN bus that can be generated with access from either inside or outside the vehicle which are replay, injection, fuzzing, sniffing and DoS attacks [9].

A. Sniffing attack

During the data collection process, CAN sniffing attack is reflected using the cansniffer command. The data on the CAN bus of ICSim is possible to be dumped in a log file, read and further analysed because there are no authentication mechanisms, encryption and the broadcast transmission in CAN. CAN sniffing attack is considered a passive attack as long as it is not used to manipulate messages.

B. Replay attack

Replay attack is an attack that is executed by replaying the CAN data recorded in log files or replay specific CAN IDs on the traffic. Using CANalyse, we replayed the analysed specific CAN IDs of the functions of the turn signal and speedometer, excluding the power door.

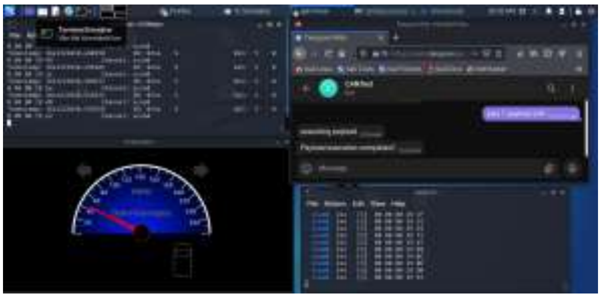


Fig. 4 Result on IC Simulator speedometer after replay attack

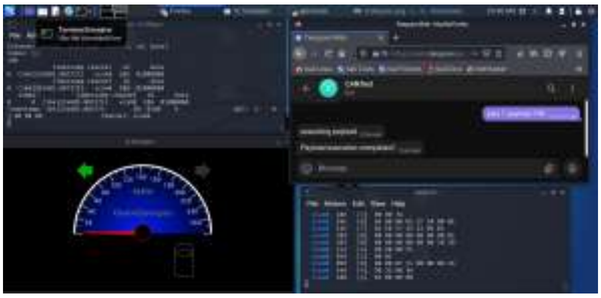


Fig. 5 Result on IC Simulator left turn signal after replay attack

However, we find that CANalyse is not able to show the result of replaying a recorded network traffic and works only by specifying the CAN ID. Thus, we tried to execute the replay attack on ICSim using can-utils tool. This replay attack is done via the cansend command. The command can send specific can ID and its corresponding payload.

After replaying the network traffic recorded in the log files using can-utils, the IC Simulator shows the replayed network traffic. The security vulnerability of the CAN bus broadcasting mechanism is revealed as the replay attack is successfully executed. The speedometer (CAN ID - 244) and the turn signal (CAN ID - 188) were displaying the expected repetitive actions as replayed in the network traffic. The speedometer movement changes its values according to the payload when messages with CAN ID – 244 were sent as shown in Figure 4, and the turn left signal is turned on when a message with CAN ID - 188 is sent as shown in Figure 5. CAN replay attack may be considered passive if it only involves replay of a particular CAN ID but can be severe if the network traffic is replayed repeatedly as the driver may lose control of the vehicle.

From these experiments, it validates the known vulnerabilities of in-vehicle CAN networks even in a simulated environment. However, we are not yet able to show experiments of other mentioned attacks against CAN bus in this paper.

IV. SECURITY MEASURES RECOMMENDATION

The result from the attack experiment proves the insecurity of CAN buses. Hence, we will explain the security measures for CAN buses in this section.

Encryption, authentication and redesign of the CAN protocol are the categorized security solutions for CAN bus [8h]. The recommended security measures are network segmentation, cryptography-based methods, and Intrusion Detection System (IDS).

Network segmentation is the easiest protection mechanism as it is separating CAN network into multiple subnetworks. Segmentation allows you to control who can access a particular subnetwork to reduce the damage of being attacked by limiting the spread of the attack. The benefit is to limit the access to the end-user. Although network segmentation increases security, it is not cost-effective and difficult to maintain.

Aside from that, cryptography-based methods can authenticate and ensure data integrity with MAC mechanism and privacy protection through symmetric and asymmetric cryptosystems. There is an existing method in CAN bus which is checksum calculation, Cyclic Redundancy Code (CRC) that checks any changes in the CAN frame during transmission. One of the proposed solutions is to replace the CRC field with MAC signature to improve authentication and integrity of the in-vehicle network data. However, this method might be costly and requires changes of CAN bus protocol. In addition, better security can be provided with a combination of encryption and MAC.

Moving on to the next recommendation which is in-vehicle intrusion detection system (IDS). IDS is a key approach that detects malicious attacks inside vehicle networks. IDS can be signature based or anomaly-based systems. Implementation of IDS relies on a constant CAN traffic behavior as IDS can be passive because the ECUs inside the vehicle have a fixed interval to generate CAN messages even if no change occurs. Another approach makes use of the physical layer characteristics of ECU, such as signals and voltage profile, and compares changes in these characteristics to detect faulty operation in cars. IDS based on signature is based on detecting a predefined list of attack signatures, it needs to update the database signatures when new attacks occur. In spite of the fact that signature-based IDS is efficient and does not involve modification of CAN protocol, extracting attack signatures in real time can be challenging and may suffer from high latency due to increased processing time [8h].

C. CONCLUSIONS

This paper explained the overview of CAN bus, then moving to methodology that involves data collection and analysis, and the experiment attack and concluded with security measure recommendations. From data collection and analysis and experiment attack, we were able to conclude that in-vehicle CAN networks are indeed insecure. A vehicle can be easily exploited and exposed to malicious consequences if the attacker gets access to the CAN bus.

Even when using a simulated vehicle environment, the absence of security measures of CAN bus is validated as ICSim is easily exploited. However, we find that the car functions available in the simulated vehicle, ICSim software are too limited, and the executed attacks are not enough to further prove on the vulnerabilities of CAN bus. In the future, we aim to use a real vehicle or ICSim, and execute other available attacks against CAN bus with other CAN IDs such as brake, anti-lock braking system (ABS) or electronic brake-force distribution (EBD) to provide comprehensive insight on in-vehicle CAN network and its vulnerabilities.

On the other hand, it is concluded that security measures need to be implemented on the CAN bus in the future to prevent any vehicle issues that are harmful to the people as proposed from many research works.

REFERENCES

[1] Ueda, H., Kurachi, R., Takada, H., Mizutani, T., Inoue, M., & Horihata, S. (2015). Security authentication system for in-vehicle network. SEI technical review, 81, 5-9.

[2] Zhang, H., Meng, X., Zhang, X., & Liu, Z. (2020). CANsec: a practical in-vehicle controller area network security evaluation tool. Sensors, 20(17), 4900.

[3] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., & Savage, S. (2010, May). Experimental security analysis of a modern automobile. In *2010 IEEE symposium on security and privacy* (pp. 447-462). IEEE.

[4] Mansor, H., Markantonakis, K., & Mayes, K. (2014, June). CAN bus risk analysis revisit. In IFIP International Workshop on Information Security Theory and Practice (pp. 170-179). Springer, Berlin, Heidelberg.

[5] Palanca, A., Evenchick, E., Maggi, F., & Zanero, S. (2017, July). A stealth, selective, link-layer denial-of-service attack against automotive networks. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 185-206). Springer, Cham.

[6] Payne, B. R. (2019). Car Hacking: Accessing and Exploiting the CAN Bus Protocol. Journal of Cybersecurity Education, Research and Practice, 2019(1), 5.

[7] Bella, G., & Biondi, P. (2018, September). Towards an Integrated Penetration Testing Environment for the CAN Protocol. In International Conference on Computer Safety, Reliability, and Security (pp. 344-352). Springer, Cham.

[8] Lade, K. (2020). CANalyse. Retrieved Jan 6, 2022 from https://github.com/KartheekLade/CANalyse

[9] Aliwa, E., Rana, O., Perera, C., & Burnap, P. (2021). Cyberattacks and countermeasures for in-vehicle networks. ACM Computing Surveys (CSUR), 54(1), 1-37.

[10] Bozdal, Mehmet & Samie, Mohammad & Aslam, Sohaib & Jennions, I.K.. (2020). Evaluation of CAN Bus Security Challenges. Sensors. 20. 16-17. 10.3390/s20082364.

[11] Smith, C. (2017). ICSim: Instrument Cluster Simulator for SocketCAN. Retrieved Jan 6, 2022 from https://github.com/zombieCraig/ICSim