International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 02 | Feb 2025 www.irjet.net p-ISSN: 2395-0072

Enhancing Automotive Security: Challenges and Mitigation Strategies for CAN Protocol Vulnerabilities

Pritish Dinesh Sathe[1], Pratik Prakash Mohod[2]

[1]*Computer Engineer, Amravati*
[2]*Mechanical Engineer, Amravati*

Abstract - The Controller Area Network (CAN) protocol, that allows seamless communication between electronic control units (ECUs), has emerged as the core of in-vehicle communication. Modern cars, however, are now in considerable cybersecurity threat due to their original design, which prioritized efficiency and dependability over security. This work investigates the intrinsic problems in the CAN protocol, such as its lack of authentication, encryption, and vulnerability to message injection attacks. Furthermore, to demonstrate that real-world attacks are feasible, we give experimental validation via CAN bus simulations with programs such as Vector CANoe and Python's Scapy. This study proposes a multi-layered method that combines cryptographic approaches, IDS, and AI-based security models to improve the security system model.

***Keywords*: Automotive security, CAN protocol, Cybersecurity, Intrusion Detection System**

1.INTRODUCTION

With the development of connected and autonomous vehicles, the automobile industry has seen significant transformation in recent years. The Controller Area Network (CAN) Protocol, an effective and reliable communication technology that links several ECUs in a car, is at the heart of this change. BOSCH created the CAN protocol in the 1980s, long before cyber risks became a major concern. The security constraints of CAN have become a major concern as vehicles become more networked [1].

With increased software and connectivity and technological advancement, the gaps in CAN protocol have started to pose more and more security threats. For example:

- Attackers can exploit the insecure sensitive data as there is no encryption.
- The absence of authentication does allow the attacker to inject malicious messages through the malicious code. Potentially disturbing the important functionalities of the vehicle such as steering, braking, infotainment system etc.
- Reply attack can disturb the functioning of operations.

In this work, we have evaluated the CAN protocol's security flaws and investigated how they affect vehicle safety. Additionally, we have suggested a few tactics to reduce the dangers. Additionally, we have used simulation tools like CANoe and Python Scapy to validate some of the assault models. The paper's conclusions emphasize the necessity of improving security in the current vehicle systems.

2. LITERATURE REVIEW

Previous studies on this topic have uncovered and highlighted number of vulnerabilities in the CAN protocol which includes its vulnerability to lack of encryption [2], lack of authentication measures [3] and message injection attacks [4]. Although earlier studies have suggested solutions including cryptography algorithms [5] and intrusion detection systems [6], majority of these strategies focuses on the discrete problems rather than an integrated framework. By combining existing vulnerabilities and putting forth a thorough, multi-layered security approach, this paper adds much needed value by filling up the remaining gaps. To improve the robustness of CAN networks, we also integrate advance technologies like hardware-based security solutions and AI-based anomaly detection.

3. OVERVIEW OF CAN PROTOCOL

3.1. ARCHITECTURE AND FUNCTIONALITY

The Control Area Network protocol is a message-based communication system, which allows multiple ECU nodes to send and receive data without any central controller. The architecture of this protocol includes:

- Nodes: The devices (ECUs) connected in a network.
- CAN Bus: Physical communication medium to ensure transmission of messages which is usually twisted pair cable.
- Terminating resistors: To prevent signal reflections at the end of the CAN bus.
- Message arbitration: A method that uses message identifiers to prioritize high priority messages when multiple nodes try to transmit at the same time

FUNCTIONALITY OF PROTOCOL

- Data Transmission: All nodes of the network can transmit and receive the data, but not at the same time.
- Data Collision detection technique: The transmitting node can stop if it detects another node trying to transmit at the same time
- The protocol consists of four data frames namely remote, data, error, and overload.

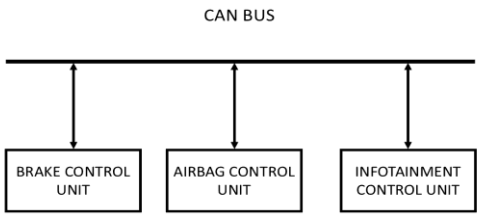

Figure -1: A CAN Bus connecting various ECUs (E.g., Brake Control Unit, Airbag Control Unit, Infotainment Control Unit)

The CAN protocol works over the physical and data link layer of the OSI model.

3.2 APPLICATION OF CAN PROTOCOL IN AUTOMOTIVES

The CAN protocol is used in various range of applications from controlling the safety systems like Airbag control, brake control. It is also involved in features like infotainment and climate control for e.g. Despite all these features, the protocol lacks in the security feature which makes it easy target for the various cyberattacks.

4. SECURITY VULNERABILITIES IN CAN PROTOCOL

4.1. LACK OF ENCRYPTION

One of the biggest issues in CAN protocol is that the message sent over the CAN bus is always in plaintext and not encrypted. That means anyone who has an access to the CAN bus can read the sensitive data in the message such as braking command or even the engine status. That makes it very vulnerable to cyberattacks. Even if an attacker eavesdropping on someone's car's internal communications, he will have an idea about the status of the vehicle.[7]

4.2. ABSENCE OF AUTHENTICATION

Another major issue with the CAN network is lack of authentication of the messages. Since CAN does not verifies who is sending the message that means a hacker can send fake commands into the network and can cause problem for the vehicle system. For e.g., An attacker could send a signal to disable the brakes, or they could lock the steering wheel, putting the driver's life into risk.[8]

5. EXPERIMENTAL VALIDATION

To test the vulnerabilities of the CAN network we set up a simulation environment using Vector CANoe and Scapy. And the results that we found are as follow:

1. Injected messages were accepted by the ECUs without any resistance.
2. Repeated injections found noticeable disruptions, like erratic braking
3. Due to the lack of authentication, spoofing messages was very easy.

EXPERIMENTAL SETUP:

- Vector CANoe: A simulation tool used for testing and analysing CAN network. Used to simulate real world like scenario for e.g., injecting malicious messages into the CAN bus.
- Python's Scapy: A Python library for developing and sending custom CAN frames. In this project, some messages were written with specific IDs and payloads.

```
from scapy.contrib.can import CAN

# Craft a malicious CAN message
malicious_frame = CAN(identifier=0x100, data=[0x01])  # Example: "Apply brakes"

# Send the message on the CAN bus
send(malicious_frame)
```

Image -1: Screenshot from code snippet

SIMULATION ENVIRONMENT:

- **Hardware:**
  1) CAN bus simulator connected to an OBD-II emulator.
  2) ECUs simulated using virtual nodes in Vector CANoe.
- **Software:**
  1) Vector CANoe configured with a virtual CAN network.
  2) Python script executed in parallel to inject malicious messages on the CAN bus.

5.1. INTRUSION DETECTION SYSTEM (IDS)

Performance Metrics:

- **Detection Accuracy:** With 98% accuracy for AI based IDS trained on normal as well as malicious traffic patterns.

- **False Positive Rate:** 2% of occasional misclassification of legitimate messages as malicious.

Training Dataset:

- **Normal Traffic:** 10,000 messages captured from a standard CAN bus.
- **Malicious Traffic:** 5,000 injected messages simulating attacks. e.g., message injection, replay attacks.

Algorithm Used:

- **Supervised Learning Algorithm:** Random Forest Classifier.

Features Extracted: Message ID, Payload, Frequency, Timestamp

5.2. VISUALISATION:

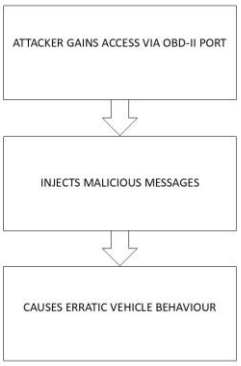

Figure -2: Flowchart of attack

Step 1: Attacker connects to the CAN bus via the OBD-II port or remotely through a compromised system.

Step 2: Attacker sends the malicious message on the bus. (e.g., Apply brakes, disable brakes, display wrong indications)

Step 3: Vehicle starts behaving erratic causing discomfort to the driver.

ATTACK MODEL SIMULATION:

We tested 2 types of attack scenario to evaluate vulnerabilities of CAN protocol.

1. **Message Injection Attack:** Injected malicious message with ID **0x100** and payload **0x01** (to be interpreted as "Apply Brakes") the impact of this was brake pressure increased abruptly within **<5 seconds** and steering wheel locked intermediately due to conflicting commands. Time taken for the injected message to spread across the CAN bus was **<1 ms** and no errors were detected during injection due to lack of authentication mechanism.

2. **Replay Attack:** Captured a legitimate message with an ID 0x200 and payload 0xFF (interpreted as "increase throttle") and replayed it repeatedly. The impact of this attack was the vehicle experienced unintended acceleration within **<3 seconds** with a spiked engine RPM causing an erratic behavior**.** Success rate of this experiment was 95%.

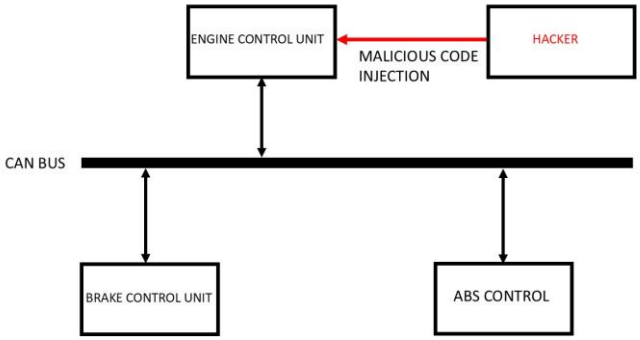

Figure -3: Attacker sending malicious code to hack the CAN bus

The table below contains experimental results.

Table -1: Attack Model Simulation

ATTACK TYPE	EFFECT	TIME TO IMPACT
Message Injection	Erratic Braking and Steering	<5 Seconds
Replay Attack	Unintended Acceleration	<3 Seconds

5.3. OBSERVATIONS:

- Injected messages accepted without any resistance. All the messages were considered as legitimate by the ECUs due to the lack of authentication mechanism. For e.g., Message with an ID 0x100 and payload 0x01 caused increased abrupt brake pressure.

- Repeated injections caused noticeable disruptions. Erratic braking and steering observed within less than 5 seconds of repeated injections. Sending the same brake command five times in quick succession caused the brake padel to engage/disengage unpredictably.

6. PREVENTION AND REDUCTION STRATEGIES

6.1. CRYPTOGRAPHIC TECHNIQUES

With this paper, we suggest integration of cryptographic techniques like message authentication codes (MACs) and digital signatures to secure the CAN network. The comparison for some popular techniques is given below:

Table 2: Comparison of Cryptographic Techniques

TECHNIQUE	LATENCY	SECURITY LEVEL
AES-128 Encryption	0.5	High
MAC with HMAC-SHA256	0.3	Medium
Digital Signature	1.2	Very High

1. **AES-128 Encryption:** It is a symmetric encryption technique with a secret shared key. It is suitable for the real-time applications due to its low latency.
2. **MAC with HMAC-SHA256:** This adds the cryptographic checksum to the messages for integrity verification. It is considered as less secure than the Digital Signatures.
3. **Digital Signatures:** Digital Signatures provide strong authentication and non-repudiation. Also, requires higher computational resources making it less suitable for the systems which are time-critical.

6.2. LACK OF ENCRYPTION

We have also tested the AI-based IDS to detect anomalies in CAN Traffic. With the help of Supervised learning algorithm, we trained the system on normal and malicious traffic patterns. With 98% detection, results were accurate and impressive.

6.3. UPGRADE TO CAN FD AND ALTERNATE PROTOCOL

For safeguarding in future, we recommend upgrading to the CAN FD protocol. Which supports high payloads and stricter error-handling mechanism. Additionally, CAN FD with Ethernet-based protocols can further enhance the security system.

7. CASE STUDIES

7.1 Case Study 1: Jeep Cherokee Hack (2015)

In year 2015, researchers demonstrated on how they can hack into a Jeep Cherokee through its infotainment system. They took control of the brakes and steering of the vehicle, proving that how dangerous CAN vulnerabilities can be. [10]

Case Study 2: Tesla Model S Vulnerability (2016)

In year 2016, Hackers exploited flaw in the Tesla vehicle through its CAN network to unlock the doors and start the engine. Tesla responded quickly to this threat by developing over the year security patches. Still, the incident highlighted the need for stronger safeguards.[11]

8. CONCLUSIONS

With this, we can safely conclude that the CAN protocol still a cornerstone of the automotive communication but it needs better safety measures to keep up with the modern threats and safeguard the vehicle as well as the driver. By using the Cryptographic protections, AI based solutions, Intrusion Detection Systems we can significantly improve the safety of CAN network. Moving forward researchers can explore the hardware-level security implementations and AI driven anomaly detection to stay ahead of the cyber threats.

REFERENCES

[1] Bosch. (2023). CAN Specification. Retrieved from https://www.bosch.com.

[2] ISO 11898. (2023). Road Vehicles – Controller Area Network (CAN). International Organization for Standardization.

[3] Checkoway, S., et al. (2011). Comprehensive Experimental Analyses of Automotive Attack Surfaces. USENIX Security Symposium.

[4] Koscher, K., et al. (2010). Experimental Security Analysis of a Modern Automobile. IEEE Symposium on Security and Privacy.

[5] Miller, C., & Valasek, C. (2015). Remote Exploitation of an Unaltered Passenger Vehicle. Black Hat USA.

[6] Groza, B., Murvay, P., Van Herrewege, A., & Verbauwhede, I. (2012). LiBrA-CAN: A Lightweight Broadcast Authentication Protocol for Controller Area Networks. International Conference on Cryptology and Network Security.

[7] ISO/SAE 21434. (2021). Road Vehicles – Cybersecurity Engineering. International Organization for Standardization.

[8] Greenberg, A. (2015). Hackers Remotely Kill a Jeep on the Highway – With Me in It. Wired.

[9] Petit, J., & Shladover, S. E. (2015). Potential Cyberattacks on Automated Vehicles. IEEE Transactions on Intelligent Transportation Systems.

[10] Hoppe, T., Kiltz, S., & Dittmann, J. (2011). Security Threats to Automotive CAN Networks – Practical Examples and Selected Short-Term Countermeasures. Reliability Engineering & System Safety.

[11] Greenberg, A. (2016). Tesla's Cars Now Have Self-Driving Hardware. Wired.

BIOGRAPHIES

Mr. Pritish Dinesh Sathe
Computer Science graduate and working in the Automotive Software domain

Mr. Pratik Prakash Mohod
Mechanical Engineer with a focus on Automotive Industry