AUTOMOTIVE CYBERSECURITY: THREAT ASSESSMENT

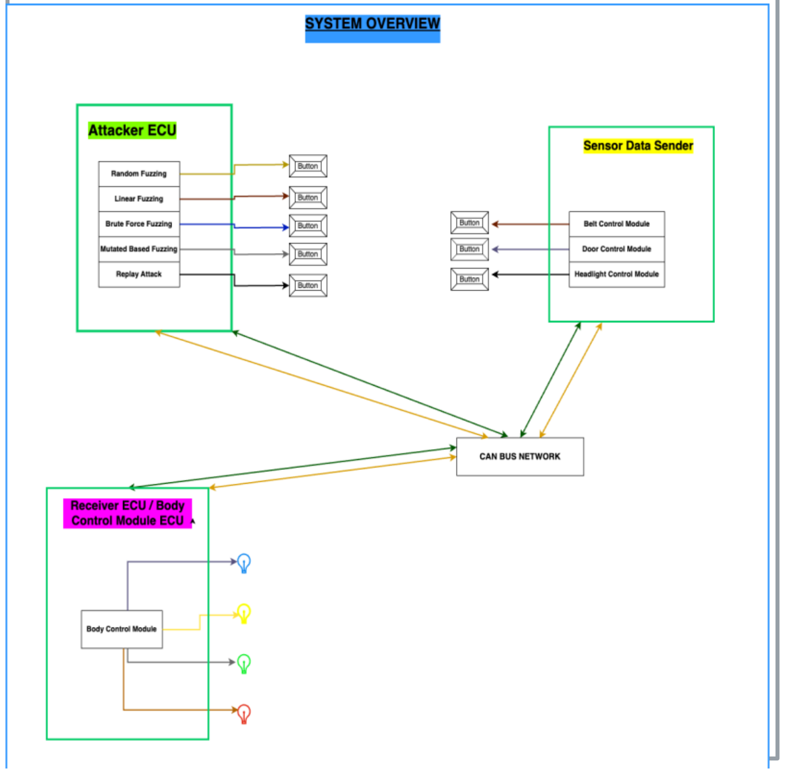
AUTHOR: LINDA MAFUNU

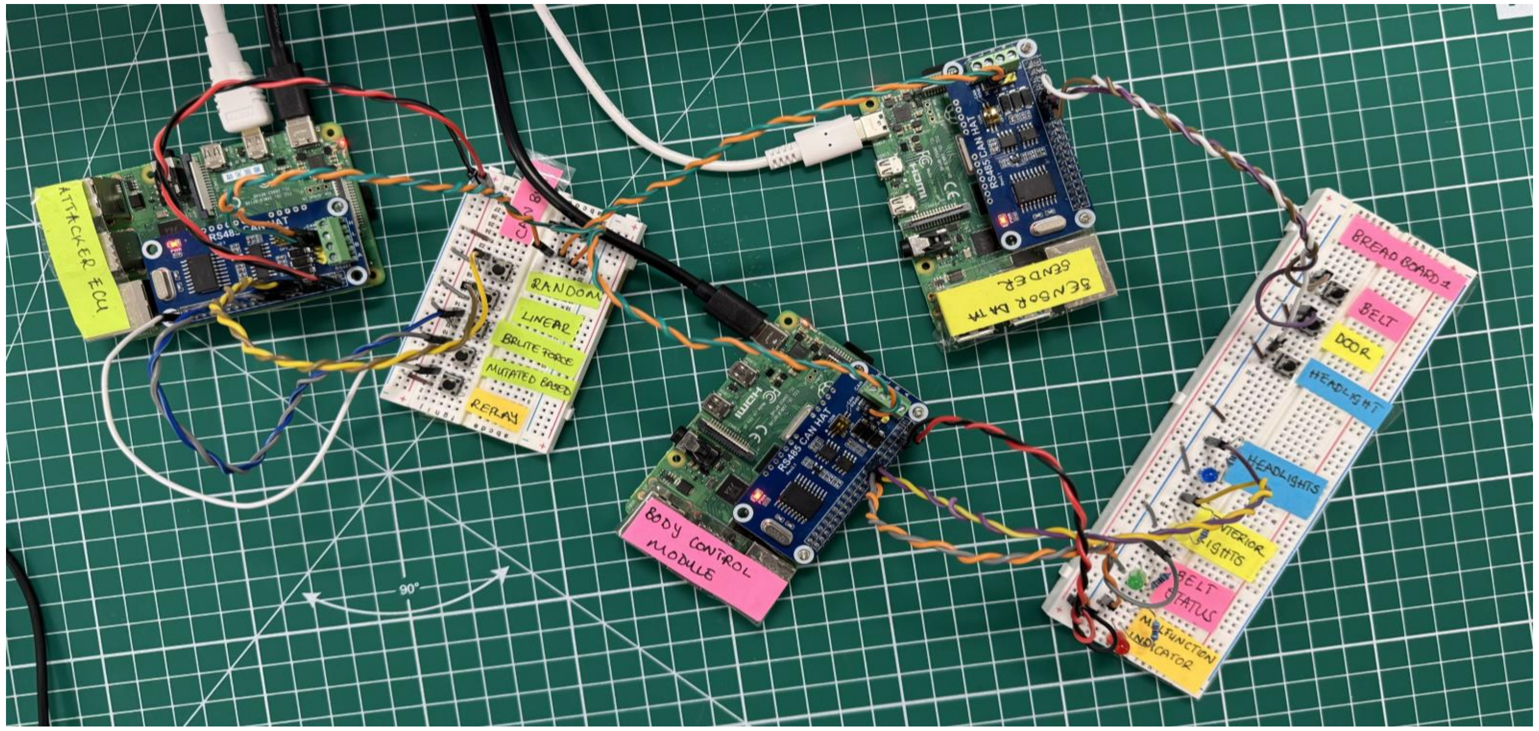
STUDENT NUMBER: 2216686

INTRODUCTION

Objective: To uncover vulnerabilities in the CAN network using multiple fuzzing and replay attacks, while evaluating the effectiveness of Message Authentication Codes (MACs) in defending against these attacks and assessing their impact on message performance in automotive networks.

Scope: This study contributes to the development of secure and efficient automotive cyber security solutions, focusing on balancing security and performance in CAN bus communications.





SYSTEM SET UP

SYSTEM FUNCTIONALITY

- **ECUs:** This setup includes three Raspberry Pi-based modules, each designated as an Attacker, Sender, or Receiver ECU.
- **Sender ECU:** This ECU is responsible for transmitting sensor data from three control modules:
  - **Door Control Module (DCM):** Sends messages based on door lock or unlock status.
  - **Headlight Control Module (HCM):** Sends commands in response to ambient light levels.
  - **Belt Control Module (BSM)**: Transmits status messages based on the door lock state.
- **Attacker ECU:** This ECU initiates various types of fuzzing attacks on the CAN bus, including **Random, Linear, Brute Force, Mutation-Based, and Replay attacks,** designed to test the resilience and security of the CAN network.
- **Receiver ECU**: Acting as the Body Control Module (BCM), this ECU processes incoming CAN messages, checks MAC tags, logs data for message analysis, and tracks latency, ultimately influencing the status of connected output devices (LEDs).
- **Breadboard 1:** Contains buttons that control the Sender ECU's modules for Door, Headlight, and Belt functions, allowing real-time testing and monitoring.
- **Breadboard 2:** Contains buttons that activate each of the five attack types from the Attacker ECU, enabling a range of simulated interference scenarios on the CAN bus.

FUZZING TESTING AND REPLAY ATTACK METHODOLOGY

Fuzzing Techniques:

- Random Input Generation
- Linear Sequence Injection
- Brute Force Attacks
- Mutated-based Fuzzing

REPLAY ATTACK PROCESS

CAPTURE LEGITIMATE CAN MESSAGES

STORE MESSAGES FOR LATER USE

RE-TRANSMIT CAPTURED MESSAGES TO BYPASS VERIFICATION

RESULTS SUMMARY

Test Type	Results	Impact	Performance
Random	A 100% success rate in triggering errors, as the message was classified as illegitimate and did not pass MAC verification.	Negative value delays observed in logs	Buffer overflow if the frequency of sending messages is too high. Risk: Potential Denial Of Service (Dos)
Linear	A 99.9% success rate in triggering errors, as the message was classified as illegitimate and did not pass MAC verification.	Negative value delays observed in logs	Buffer overflow if the frequency of sending messages is too high. Risk: Potential Denial Of Service (Dos)
Brute Force	100 % fail rate in triggering errors for session without Message Authentication, all messages are classified as legitimate but result in no sensor manipulation. With Message Authentication enabled, however, all messages fail verification.	Negative value delays observed in logs.	Buffer overflow if the frequency of sending messages is too high. Risk: Potential Denial Of Service (Dos)
Mutated-Based	Without message authentication enabled, messages were classified as legitimated and successfully triggered sensors With message authentication, all mutated messages were blocked	Negative value delays were observed in logs. Sensors, behavior manipulated	Buffer overflow if the frequency of sending messages is too high. Risk: Potential Denial Of Service (Dos)
Replay Attack	100% successful in bypassing MAC verification	Manipulated sensor behavior	Huge delay between the time messages were captured and replayed on CAN bus.
Baseline Test	No errors were detected. All messages processed correctly	Manipulated sensor behaviors. Minimal delay	Average approximate overhead of 3.59 milliseconds

KEY FINDINGS

Replay Attacks were highly effective against current security measures

Fuzz testing reveals vulnerabilities in message parsing

MAC tags significantly reduce attack success rates

Adding MAC tag increases message transmission delays

SECURITY RECOMMENDATION

1

Implement end to end encryption for CAN messages

2

Adopt dynamic MAC addresses for improved message authentication

3

Enhance message filtering algorithms

4

Implement intrusion detection systems

CONCLUSION

This project demonstrated the effectiveness of implementing MAC tags in CAN bus messages to resist fuzz testing and replay attacks in automotive attacks in automotive networks. The performance impact of adding MAC tags is a little bit significant but necessary for enhanced security.

REFERENCES

- Aliwa, E., Rana, O., Perera, C., & Burnap, P. (2021). Cyberattacks and Countermeasures for In-Vehicle Networks. ACM Computing Surveys, 54(1), 1–37. https://doi.org/10.1145/3431233
- Luo, F., Zhang, X., Yang, Z., Jiang, Y., Wang, J., Wu, M., & Feng, W. (2022). Cybersecurity Testing for Automotive Domain: A Survey. Sensors, 22(23), 9211. https://doi.org/10.3390/s22239211
- Mahmood, S., Nguyen, H. N., & Shaikh, S. A. (2022). Systematic threat assessment and security testing of automotive over-the-air (OTA) updates. Vehicular Communications, 35, 100468. https://doi.org/10.1016/j.vehcom.2022.100468
- RS485 CAN HAT - Waveshare Wiki. (n.d.). Www.waveshare.com. https://www.waveshare.com/wiki/RS485_CAN_HAT
- Smith, C. (2016). The car hacker's handbook: a guide for the penetration tester.
- Thorne, B. (2024, June 23). GitHub - hardbyte/Python-can: The can package provides controller area network support for Python developers. GitHub. https://github.com/hardbyte/Python-can/tree/main
- Werquin, T., Hubrechtsen, M., Thangarajan, A., Piessens, F., & Mühlberg, J. T. (2019, September). Automated fuzzing of automotive control units. In 2019 International Workshop on Secure Internet of Things (SIOT) (pp. 1-8). IEEE.

INFORMATION

School email: 2216686@swansea.ac.uk

Primary email: mafunulinda@outlook.com

Raspberry Pis Username: lindamafunu

Password: swansea2024

Git repository link: here