After going over recent studies on different attacks possible on CAN and its variants, it is clear that as the number of ECU’s increase the system becomes more and more vulnerable to attacks as a large number of nodes have created a network with essentially no one central host.

The researchers have not only researched potential attacks but possible solutions as well to mitigate the vulnerabilities. Implementing those on CAN Networks can potentially improve the security but have not been implemented on a large scale.

Going forward we’ve discussed different attacks on CAN Networks.

Five years ago, the number of ECUs within a vehicle was about 70 ECUs. This number has doubled nowadays. The CAN protocol, along with other communication protocols, encounters mainly three security issues such as backward compatibility, replay attacks, and message authentication. In [1] LCAP - Lightweight Controller Area Protocol has been discussed. It differs from CAN as the main purpose of the LCAP is to keep messages authenticity and integrity for all messages traveling over the CAN bus. However, it is highlighted that the LCAP has a vulnerability related to replay attacks, specifically in channel requests. The attackers can exploit this by periodically replaying channel request messages, leading to a Denial of Service (DoS) attack, blocking normal data messages. A malicious node can record messages on the bus, identify the channel request message through trial and error, and replay it periodically. Controlling messages take priority over data messages so periodically sending the control messages for a replay attack will block the network pipeline to keep sending controlling messages, and data messages don't get passed through.

The attack is quite severe as it affects controls in an automotive system. A delay or failure of those could cause major issues in a vehicle. The detection of this attack is also quite difficult but can be prevented by adding a nonce to the messages. A nonce or a timestamp will prevent a replay attack.

The simulation of this attack is conducted using the Vector CANoe simulation, a well-known automotive network simulator. The proposed solution includes using nonce in messages to prevent replay attacks and improve system security. [2][3]

Another form of replay attack was implemented in [4]. One is to replay the entire message and the other one is to replay only part of the frame excluding the identifier. In the first attack, the entire message is stored by a node and then transmitted again when the CAN bus is idle. In this attack all nodes will receive faulty data. The only way to detect this attack is when ID is compared and the original source node detects receiving a message with the same ID as theirs. They can create an error frame when a replay attack is detected.

In the second method, partial frames are replayed and ID frames are replaced. This attack is harder to detect as the source uses its own ID. Since it’s a replay attack the way to prevent this is also through adding timestamps or nounces. [5][6]

As mentioned in [7], [8], [9], and [10], CAN does not offer security services such as encryption or data frame authentication. This means that eavesdropping and replay attacks in CAN are possible.

Other than replay attacks, a common attack researched was through a diagnostic malicious App. When a malicious app is used for attack, it enables the attacker to perform a long-range wireless attack where the attacker or attacking equipment need not be in the close range of the vehicle to perform the attack. [11] The user will install a self diagnostic app to monitor status information after installing an OBD2 scan tool on the vehicle and then pairing it with his/her smartphone by Bluetooth. When the driver installs on his/her smartphone the malicious self-diagnostic app distributed by an attacker, the attacker can launch the actual attack. The attacker can obtain status information of the vehicle from the malicious self-diagnostic app and use it to inject malicious data into the in-vehicle network. As per this research vulnerabilities of in-vehicle CAN are weak access control, no encryption and no authentication.

After the malicious app is installed on the victim’s device, it transmits data frames of the in-vehicle CAN to the attacker’s server using the smartphone’s mobile communication network. This would force control of an ECU to the in-vehicle CAN via the malicious app. The target vehicle would have a physical malfunction caused by the abnormal control data that was transmitted from the attacker’s server.

Another prominent attack with malicious apps is discussed in [12]. Apps for vehicles can be easily forged/repackaged and redistributed. Using a vehicle diagnostic device, The attacker can get a CAN data frame that can drive a specific ECU mounted in the target vehicle. Furthermore, the attacker can download a vehicle application that is distributed/sold in the app market and repackage it in a desired form. The diagnostic app will provide many services to the victim while they are driving and can use this service to start an attack.

An attacker identifies vulnerabilities in a vehicle's wireless communication module (ELM327), creates a malicious code based on the analysis, and inserts it into a legitimate vehicle app. The tampered app is then distributed through app markets. A victim unknowingly downloads and installs the malicious app, which is designed to exploit the ELM327 module in their vehicle. When the victim runs the app and starts driving, the malicious code executes harmful actions. As countermeasures to such attacks, app obfuscation and a whitelist-based firewall were proposed. Car manufacturers must install the security feature in the OBD-II interface, because we cannot depend on an external device to counter threats to the vehicle itself.

Apart from attacks such as DOS or Replay attacks, there are other attacks that CAN is vulnerable to. One of those attacks is a bus-off attack, which can cause a victim ECU to disconnect itself from the CAN bus and, subsequently, an attacker can masquerade as that ECU. A limitation of the bus-off attack is that it requires the attacker to achieve tight synchronization between the transmission of the victim and the attacker’s injected message. [13]

The bus-off state is a state of error in a CAN controller in which the node is disconnected from the bus communications which means it can neither transmit nor acknowledge frames. A node that is in the bus-off state can only rejoin the network after observing 128 occurrences of the bus-free signal of 11 consecutive recessive bits. When a node recovers from bus-off, it resets its counter and starts from the initial error-active state.

The attack's success depended on meeting a few conditions: matching message IDs with the victim and synchronization with the victim message. The attack is achieved by identifying a unique message that precedes the victim. However, if no unique preceded ID exists, fabricating and injecting unique preceded IDs to interfere with the victim's transmission was proposed. It can still be detected by an IDS. Alternatively, the schedule-based bus-off attack doesn't rely on unique preceded IDs for synchronization. It targets instances of the victim message facing blocking or interference, regardless of the preceding message's ID. The attacker gains knowledge about message sets on the CAN bus before launching the attack, allowing them to identify opportunities for successful synchronization

REFERENCES

[1]P. Noureldeen, M. A. Azer, A. Refaat and M. Alam, "Replay attack on lightweight CAN authentication protocol," 2017 12th International Conference on Computer Engineering and Systems (ICCES), Cairo, 2017, pp. 600-606, doi: 10.1109/ICCES.2017.8275376.

[2] A. Van Herrewege, D. Singelee, and I. Verbauwhede. CANAuth - A Simple, Backward Compatible Broadcast Authentication Protocol for CAN bus. Escar 2011, Embed. Secure. Cars, 2011.

[3] A. Hazem and H. a H. Fahmy. LCAP - A Lightweight CAN Authentication Protocol for Securing In-Vehicle Networks. Escar 2012, 2012.

[4] P. Thirumavalavasethurayar and T. Ravi, "Implementation of Replay Attack in Controller Area Network Bus using Universal Verification Methodology," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 1142-1146, doi: 10.1109/ICAIS50930.2021.9395871.

[5] M. R. Ansari, W. T. Miller, C. She and Q. Yu, "A low-cost masquerade and replay attack detection method for CAN in automobiles," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4

[6]W. A. Farag, “Cantrack: Enhancing automotive can bus security using intuitive encryption algorithms,” in Modeling, Simulation, and Applied Optimization (ICMSAO), 2017 7th International Conference on. IEEE, 2017, pp. 1–5

[7]M. Wolf, A. Weimerskirch, and T. Wollinger, “State of the art: Embedding security in vehicles,” EURASIP J. Embedded Syst., vol. 2007, no. 5, p. 1, 2007.

[8] T. Hoppe and J. Dittman, “Sniffing/replay attacks on CAN buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy,” in Proc. Conf. Embedded Syst. Security, 2007, pp. 1–6.

[9] T. Hoppe, S. Kiltz, and J. Dittmann, “Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures,” Rel. Eng. Syst. Safety, vol. 96, no. 1, pp. 11–25, Jan. 2011.

[10] K. Koscher et al., “Experimental security analysis of a modern automobile,” in Proc. IEEE Security Privacy. Symp., Oakland, CA, USA, 2010, pp. 447–462.

[11] S. Woo, H. J. Jo and D. H. Lee, "A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN," in IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 993-1006, April 2015, doi: 10.1109/TITS.2014.2351612.

[12] Yousik Lee, Samuel Woo, Jungho Lee, Yunkeun Song, Heeseok Moon, Dong Hoon Lee, "Enhanced Android App-Repackaging Attack on In-Vehicle Network", Wireless Communications and Mobile Computing, vol. 2019, Article ID 5650245, 13 pages, 2019. <https://doi.org/10.1155/2019/5650245>

[13]S. Hounsinou, M. Stidd, U. Ezeobi, H. Olufowobi, M. Nasri and G. Bloom, "Vulnerability of Controller Area Network to Schedule-Based Attacks," 2021 IEEE Real-Time Systems Symposium (RTSS), Dortmund, DE, 2021, pp. 495-507, doi: 10.1109/RTSS52674.2021.00051.