**A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN**

This attack discusses the vulnerabilities in modern vehicle networks, particularly the Controller Area Network (CAN), which lacks proper security measures to authenticate and encrypt messages. This vulnerability, coupled with the emergence of connected cars that link in-vehicle networks to external networks (e.g., 3G/4G mobile networks), opens up the possibility for long-range wireless attacks.This attack is implemented using a real vehicle and a malicious smartphone application, showcasing how an attacker can remotely control a vehicle's functions by injecting CAN frames.

**Attack Model and Methodology:**

**Preliminary Phase:** The attacker utilizes diagnostic tools to acquire CAN data frames capable of controlling the vehicle's ECUs. This phase is crucial for gathering the necessary commands to manipulate the vehicle's functions remotely.

**Malicious App Distribution:** A crafted malicious self-diagnostic app, which appears legitimate, is uploaded to app markets. This app, once installed on the driver's smartphone, allows the attacker to control the vehicle through the smartphone by injecting malicious CAN data frames.

**Actual Attack Execution:** Exploiting the infected smartphone, the attacker carries out a long-range wireless attack, demonstrating the ability to control the vehicle's functions such as engine shutdown or dashboard manipulation from any location, provided there's mobile network connectivity.

**Security Vulnerabilities Identified:** The in-vehicle CAN lacks essential security features like encryption and data frame authentication, allowing for potential eavesdropping and replay attacks.The use of automotive diagnostic tools without proper security measures poses a significant risk, as these tools can control the ECUs.

A list of information on a white background

Description automatically generated

**Fig Tools used for attack**

REPLAY ATTACK

Replay attacks, are a type of Man-in-the-Middle (MITM) attack, where previously captured messages are retransmitted or delayed in their transmission. This can be particularly hazardous in vehicle environments, where such attacks can influence the functionality of safety-critical systems like braking. The insidious nature of replay attacks lies in their ability to be executed even when messages are encrypted, bypassing the traditional security measures that ensure message confidentiality. An example provided illustrates the potential danger an adversary captures a braking system's message and retransmits it at a critical moment to prevent the actual braking signal from being correctly processed, potentially leading to a collision.

**Experimental Scenario Involving Distance Measurement**

This system comprises a distance sensor and a control unit, highlighting the process through which an adversary can exploit the CAN network to execute a replay attack. The core components involved in this setup are:

Distance Sensor (M16 – Solid State LiDAR): Used for detecting, locating, and estimating the distance between the vehicle and potential obstacles on the road.

Control Unit with Microcontroller (LPC-1768): Acts as the brain of the collision avoidance system, processing data from the distance sensor and making decisions based on programmed safety thresholds.

Adversary System (Raspberry Pi with Kali Linux and MCP 2515 CAN Transceiver): Represents the attacker's toolkit, capable of recording and replaying CAN messages.

**Specifics of the Replay Attack Scenario**

Distance Recording: The adversary records the CAN messages transmitted by the distance sensor under normal driving conditions. For example, the sensor might transmit a message indicating a distance of 100 meters from an object when it is far away.

Critical Distance for Collision Avoidance: The control unit is programmed to initiate braking when the detected distance falls below 30 meters, a safety threshold to prevent collisions.

Execution of the Replay Attack: As the vehicle approaches an object and the actual distance falls below the critical 30 meters, the adversary replays the previously recorded message indicating a safe distance of 100 meters.

A diagram of a system

Description automatically generated A diagram of a computer program

Description automatically generated

Fig Record Transmit Message Scenario Fig: Replay / Playback Message Scenario

Impact of the Replay Attack: The control unit, deceived by the replayed message, perceives the object as being safely distant and thus fails to activate the braking system, leading to a potential collision.

This scenario effectively demonstrates how replay attacks can manipulate critical safety functions in vehicles by exploiting the lack of message freshness and authentication in the CAN protocol. The adversary's ability to record and replay messages pertaining to crucial parameters like distance underscores the necessity for robust countermeasures that ensure both the integrity and authenticity of CAN messages.

The inherent vulnerabilities of the CAN protocol, such as its lack of built-in encryption and authentication mechanisms, make it susceptible to various forms of cyberattacks, including replay attacks. Despite the development of countermeasures like authenticated encryption and the use of timestamps or counters to ensure message freshness, the practical implementation of these solutions varies across the automotive industry.

**Paper 3**

**Message Injection Attacks**

**Overview of the Attack**

Message Injection involves unauthorized introduction of messages onto the CAN network. Since the CAN protocol lacks message authentication, any device connected to the network can broadcast messages that appear legitimate to other ECUs (Electronic Control Units).

**Execution of the attack**

An attacker can use a device connected to the OBD-II port or exploit a vulnerable ECU to inject malicious messages. Tools for this can range from simple microcontrollers equipped with CAN interfaces (e.g., Arduino with a CAN shield) to more sophisticated custom devices designed for stealth and remote control.

**Attack Implications on the Vehicle**

Injected messages can trigger unintended actions from ECUs, such as engaging brakes, disabling safety systems, or manipulating dashboard displays. The simplicity of message injection makes it a common attack vector for disrupting vehicle functionality or creating a precursor state for more complex attacks.

**Resources Required**

Hardware: A basic microcontroller or computer with a CAN interface.

Software: Custom scripts or publicly available hacking tools designed for CAN networks.

Access: Physical or remote access to the vehicle’s CAN network, often via OBD-II port.

**Feasibility of the Attack at present times**

Message injection attacks remain possible primarily due to the lack of built-in authentication and encryption within the CAN protocol.

**Message Modification Attacks**

**Overview of the Message Modification Attack**

Message Modification involves altering legitimate CAN messages in transit between ECUs without necessarily injecting new traffic. This requires intercepting communications, a task made possible by a Man-in-the-Middle (MITM) approach.

**Execution**

A MITM device, placed inline with the CAN bus, reads legitimate messages, modifies their content, and forwards them to the intended recipient. This technique can bypass some of the inherent safety checks in CAN, such as Cyclic Redundancy Checks (CRC), by maintaining the original message structure but altering the data payload.

**Implications**

Modification attacks can subtly alter vehicle behavior in dangerous ways, such as falsifying sensor readings or command signals, leading to unsafe driving conditions. The complexity and subtlety of these attacks make detection and mitigation challenging.

**Resources Required**

Hardware: A sophisticated MITM device capable of real-time message interception and modification, usually involving two CAN interfaces and processing capabilities.

Software: Advanced software to analyze, modify, and forward CAN messages according to the attacker’s objectives.

Access: Physical access to the vehicle’s CAN network to install the MITM device.

**Feasibility of the on the present times**

These attacks are facilitated by the ability to perform a Man-in-the-Middle (MITM) intervention. They are more complex than injection attacks due to the need for intercepting and altering communications without detection. The physical nature of CAN network connections provides an opportunity for installing MITM devices.

**Denial of Service (DoS) Attacks**

**Overview**

DoS attacks aim to render the CAN network non-functional by overwhelming it with traffic or exploiting vulnerabilities to disrupt communication.

**Execution**

Two primary methods exist: flooding the CAN bus with high-priority messages to monopolize bandwidth or inducing a fault condition that triggers protective shutdown of communication by ECUs. The former can be accomplished with a simple CAN interface, while the latter may require a more sophisticated approach to exploit specific vulnerabilities.

**Implications**

A successful DoS attack can incapacitate critical vehicle functions, including safety systems, leading to a complete operational halt. This attack can be used as a standalone threat or to facilitate other direct attacks by disabling security measures.

**Resources Required**

Hardware: A device with CAN communication capability; sophistication varies by attack method.

Software: Tools to generate and send CAN messages at high rates or to exploit specific vulnerabilities for fault induction.

Access: Physical or potentially remote access to the CAN network, depending on the attack vector.

**Feasibility of DOS Attack on the recent times**

DoS attacks on CAN networks are possible due to the protocol's inability to prioritize critical messages over malicious traffic. The non-preemptive nature of CAN communication allows attackers to flood the network with high-priority messages, effectively silencing legitimate traffic.