Hardware Trojan enabled Denial of Service Attack on CAN-BUS:

The new electrical vehicles have some new functionalities such as smart features, better connectivity, and autonomous behavior. For these functionalities naturally, more ECUs are connected using CAN. For this reason, the security of the vehicle is become a major concern.

Hardware Trojan is a novel threat that enables attackers to use backdoor access to implement their attacks. It is an event-triggered activation attack. This can neither be detected nor prevented.

ATTACK:

Security not only depends on the software but also on the hardware. But nowadays, due to economic interest, it is compromised and hardware security threats such as Hardware Trojan are emerging.

Hardware Trojan is nothing, but a malicious modification of hardware implemented at the hardware level, and this is not active until a rare event. This is why it is hard to detect. It is easy to implement, and the result is catastrophic.

Trojanised CAN Controller is attached to a node, and it masquerades as another node. Since the CAN has broadcast and no authentication. So, it is easy to take control of the system. There is no main computer in CAN for communication, So CAN relies on itself to communicate. It has a special rule called “BIT STUFFING RULE” which means after five consecutive bits of the same logic level, the next bit should be the compliment. HT messes it by sending the same bit (Logic zero) six times consecutively. Logic zero is dominant so it overwrites Logic one. It will cause an error frame. Usually, other nodes will discard this message, and then the error counter increases. Once, the error counters more than 255, the node will go to BUS-OFF state.

A diagram of a error

Description automatically generated

Implementation and Demonstration of Threat Over CAN Bus for Automotive Embedded System Equipped with ADAS:

**OVERVIEW:**

ECUs in the EVs are connected to the outside world via on-board navigation or entertainment. They play a vital role in ADAS, which significantly reduces the driver’s stress load and makes the drive safe. At this stage, internal and external communication pass through the same hardware and the rules defined by software do not apply to the hardware. This opens a backdoor for the hackers.

**Vehicle Investigation:**

Using a CANalayzer tool make a physical connection to the CAN bus. Then read the frames that are transmitted by the CAN Bus. Then record the CAN trace (CAN log1).

**Creating CAN Database File:**

This is done in two steps:

1. CAN bus Log1 is uploaded to service can2sky.com which decodes it and generates a CAN DBC1 file.
2. Some specific actions like acceleration and Cruise Control are performed on the vehicle and they record CAN bus Log2.

The file is analyzed to identify the byte of frames where there is a signal change related to acceleration and cruise control. Thus, a new custom CAN DBC2 file is created.

The last step is to fusion these two files.

1. **CAN Bus Load by Injection Attack:**

**Principle:**

The principle of this attack is to send hazard frames into the CAN bus with a very low periodicity to load the bus. This will lead to bus saturation and a total stop of the vehicle which can lead to a very dangerous situation especially when a high driving speed.

**Device used:**

ECU, Dashboard with vector panel, 7-inch touchscreen car radio, Harness grouping the 12V power supply, CAN bus with 120 ohm termination resistor.

A diagram of a car

Description automatically generated

**Implementation:**

In this experiment, a simple frame of 1ms is sent on a 250 Kbit/s CAN bus, and at 51% of the total CAN load the bus can be overloaded and the ECU will be automatically restarted. Which is very dangerous.

1. **Cruise Control Acceleration Attack:**

The cruise control system maintains the car at a desired speed. Once the driver activates the cruise control via a button, this system is capable of taking over the accelerator. So, now instead of the driver, the attacker can speed up and down the car. As a result, when the attacker accelerates or decelerates the car it can lead to a very dangerous situation.

A STEALTH, SELECTIVE, LINK-LAYER DENIAL-OF-SERVICE ATTACK AGAINST AUTOMOTIVE NETWORKS:

Overview:

Unlike traditional attacks, selective denial-of-service doesn’t need to transmit complete frames. As a result, this would be undetectable via frame analysis. In modern vehicles, all CAN Buses are vulnerable due to CAN protocol weakness. The attack is low-cost and accessible.

Devices and Tools used:

Arduino Uno that served as a microcontroller, MCP2551 CAN Transceiver, ODBLink SX USB-to-ODBII Cable, CANtact 1.0.

Implementation:

The attacker physically connects the well-directed error flags like the ODB-II dongle to the car’s ODB-II port, forcing other nodes to enter a bus-off state. The attack can be done by attaching and hiding the attacking device anywhere in the car’s internal network. Additionally, the attack can be executed without any physical interaction with the target vehicle.

A diagram of a computer

Description automatically generated

Mitigations:

To mitigate these security risks, a proposed solution involves a novel detection mechanism based on the measurement of current flow on the CAN bus. This mechanism aims to identify the addition of rogue devices to the vehicle's internal network by detecting changes in the bus load caused by the presence of unauthorized nodes. While this solution can address physical tampering, it may not protect against remotely compromised nodes. However, it provides a practical approach to enhance the security of CAN bus networks in vehicles.