CAN Security

Simone SODERI

IMT School for Advanced Studies Lucca

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Goal

- Overview of hardware and software interfaces that can be used to connect to the CAN bus
- Presentation of CAN bus security attacks
- Review of some mitigation techniques

Outline

- Security overview
- Attacks to the CAN bus
- Mitigations techniques
- Final remarks on CAN bus security

Outline

- Security overview
- Attacks to the CAN bus
- Mitigations techniques
- Final remarks on CAN bus security

Introduction

- CAN bus is **widespread** in many sectors because of its high reliability and low cost
- CAN bus was designed to operate in **isolated** LANs without any external interaction
- Systems today are **increasingly connected to the Internet**, and, at the same time, attackers are creating increasingly sophisticated attacks.
- Connected to CAN are devices that if deliberately attacked could **cause damage or injury to people.**

Security overview

CAN was not designed with security in mind!

- CAN communication **lacks** both encryption and authentication mechanisms
- It is sensible to integrity, authenticity, and confidentiality attacks.
- There are **attack scenarios in which people's safety may be at risk**.
  - Self-driving vehicles: an adversary could take control of the actuators and potentially destroy the vehicle.

The **inherent vulnerability** of the CAN bus forces us to ask the following questions:

- How can an attacker gain access to the CAN bus?
- What are the attacks he/she can make?
- Are there strategies that can mitigate attacks and improve CAN bus security?

Access to the CAN bus (1/2)

There are four ways to gain access to the CAN bus

- **Using the ODB-II connector**
  - Plugging into the OBD-II connector a **dongle** that contains a **CAN transceiver** (to translate CAN differential voltages into logical 0's and 1's), a **CAN controller** (to convert between a logical bit stream and CAN frames), and a **communication interface** to communicate with something else (e.g., Bluetooth to communicate with a laptop or phone)
- **Through wiring:**
  - Knowing where the CAN bus wires are routed, it's **relatively easy to join the CAN H and CAN L wires** by drilling a hole in a car's panels to get directly to the wiring

Access to the CAN bus (2/2)

- **Through the infotainment system**
  - Vehicle infotainment systems are devices running software on an Operating System. These devices **can be attacked** by means of USB dongles, Wi-Fi, Bluetooth, DAB radio, or the Internet.
- **Through a hijacked electronic control unit (ECU)**
  - ECU can be **hijacked by programming errors in its firmware**, such as vulnerabilities in the diagnostic software stack or in the software that handles sensor data. The most common method is a **buffer overflow attack** that causes the CPU stack to be corrupted with carefully crafted data that will cause the CPU to execute it as malicious code.

Outline

- Security overview
- Attacks to the CAN bus
- Mitigations techniques
- Final remarks on CAN bus security

Types of CAN bus attacks

There are three main types of attacks

- **Authentication attacks**
  - These are where a receiver sees CAN frames with manipulated data as if from a legitimate source but designed to trigger an action (e.g., open the door locks).
- **Protocol attacks**
  - These are where the signal on the CAN TX pin to the transceiver comes not from a CAN controller but from a malicious software that sends carefully timed signals to attack the CAN protocol itself.
- **Denial of Service attacks**
  - These can vary from simple flood attacks to load the bus with otherwise legitimate traffic (causing lower priority frames to be delayed or lost) to subvert the CAN protocol.

Bus Flood Attack

- A *Bus Flood Attack* is a very simple DoS attack:
  - the attacker transmits CAN frames as fast as possible **to saturate the bus bandwidth**
  - this causes **delay of legitimate frames** and failure of parts of the system because some frames do not arrive in time.

Simple Frame Spoofing

Frame spoofing is a type of **authentication attack:** getting a receiver to accept a fake frame as if it came from a legitimate sender.

- If **directly connected** (e.g., via the OBD-II port) this is done by simply queueing the CAN frame through the drivers in the firmware of the connected device.
- If **connected via a hijacked ECU** (e.g., infotainment) this can be done by using the drivers in the device or with new drivers installed as part of the hijacking.

Simple Frame Spoofing: Issues

- **SECURITY ISSUE 1:** With this simple spoofing approach, the frames from the legitimate ECU are also received
  - the **receivers may act on both the legitimate frame and the spoofed one**.

- **SECURITY ISSUE 2:** CAN protocol requires that two **frames with the same ID are not entered into arbitration at the same time**.
  - if this happens no ECU will win the arbitration and a loop will be created
  - **Arbitration Doom Loop**: at each turn of the cycle the **Transmit Error Counter (TEC)** is increased in each of the two controllers that eventually both will go in the **bus-off state.**
    This takes about 4ms on a 500 kbit/sec CAN bus.

Simple Frame Spoofing: Issues

- **SECURITY ISSUE 3:** Legitimate ECU is driven **bus-off**
  - The legitimate ECU may **treat the failure as a wiring fault** and permanently move into a fail- safe state where it refuses to communicate.
  - All the other frames from the ECU will stop, which will cause receivers to detect a fault and they may also decide to move into a fail-safe state.
  - The intent of an attack is to deliberately **push a vehicle into a fail-safe!**

Adaptive Frame Spoofing

- The attacking device **listens to the bus** to see when the legitimate frame is sent and then queues the spoofed frame so that it does not clash.

- If the spoofed frame is sent immediately after the legitimate frame, then it will **overwrite the buffer and the receiver will most likely act on the contents of the spoofed frame**.

- The attacker needs to respond to the frame received interrupt within **a very tight timing window in order to queue a spoofed frame in time** to enter arbitration.

Error Passive Spoofing Attack (1/3)

The **simple spoofing can be detected by monitoring the bus** and looking at the timing of frames:

spoofed frames will be sent more often than expected and traffic analysis can detect anomalies.

Detection can be made much more difficult by a type of spoofing that subverts the CAN protocol itself:

error passive spoofing!

Error Passive Spoofing Attack (2/3)

CAN controller is **Error Passive** if the Transmit Error Counter (TEC) or the Receive Error Counter (REC) is above 127.

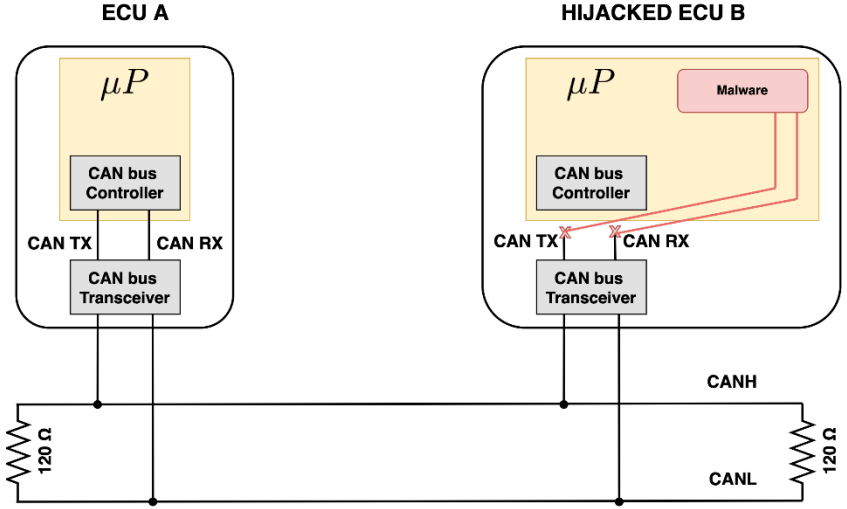
- In this stage the controller **cannot signal errors properly**: a transmitter has to basically stop sending and wait for the bus to become idle.

The attack:

- **Drive** the targeted ECU's CAN controller **into the passive error state**. This is typically done by generating error frames when the targeted ECU is sending one of its CAN frames
- Monitor the bus and see the ID of the targeted frame after it wins arbitration and then step in and **overwrite the data** and CRC fields **with a spoofed payload**.
- Other receivers (ECUs) **do not see the attack**: they simply see a single received frame with a spoofed payload. This also means that traffic analysis will not detect this attack.

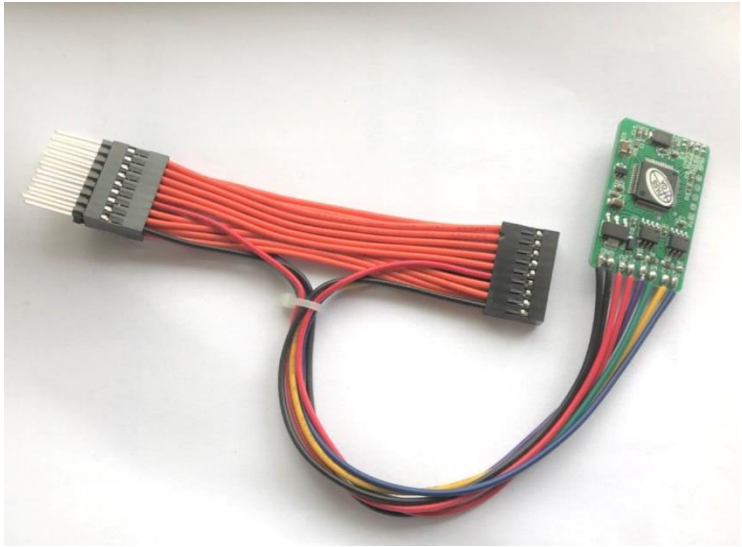
Error Passive Spoofing Attack (3/3)

- This attack cannot be done through a CAN controller: it requires low-level access to the bus to spoof the CAN protocol itself. This means that the **CAN TX and CAN RX pins on the transceiver must be under direct control of the attacker**
- A microcontroller with an on-chip CAN controller typically **multiplexes the functions of its pins**. The two pins used for the CAN controller signals to the transceiver **can be controlled as general purpose I/O pins by a malware**.



Wire-cutting Spoofing Attack

- The attacker has **physical access** to the CAN bus and can **cut wires to partition the bus**
- He/she can **spoof frames** to one of the partitions by emulating the other partition by gatewaying other frames and generating spoofed frames directly.
- This type of attack is used today **to spoof odometer readings**
  - Although the odometer is outputting correct values, the dashboard display shows a reduced mileage and is inserted into a cut wiring harness.



Bus-off Attack (1/2)

- The **Bus-off Attack** is where a targeted ECU is driven offline: all the other ECUs continue to operate but the targeted one is removed.

- It is a low-level protocol attack driving the CAN TX pin to force the Transmit Error Counter (TEC) above 255 and the ECU's CAN controller automatically goes bus-off.

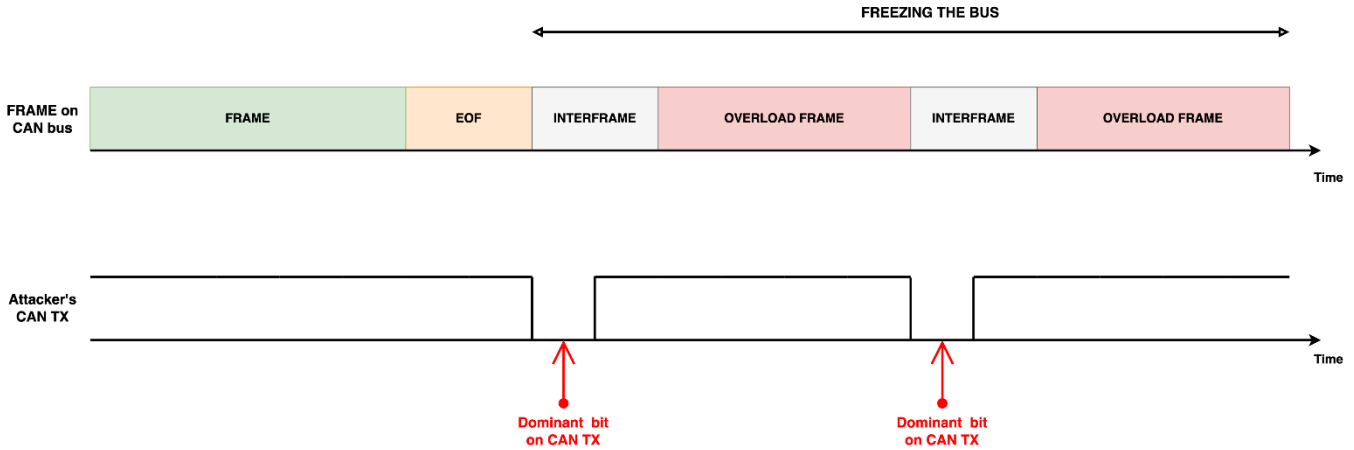
Bus-off Attack (2/2)

- It might be a simple DoS attack on a fleet of vehicles:
  - instead of trying to hijack the instrument cluster to display a Check Engine light **it is probably easier to simply take the engine management ECU off** the CAN bus and trigger the instrument cluster to see a failure and display a warning.
- Some ECUs will try to recover automatically, requiring the attack to be repeated.
- Other ECU set a flag in its internal non-volatile memory to stay offline.
  - If the purpose of the attack is to cause trouble this would count as success.

Freeze Doom Loop Attack (1/2)

- The Freeze Doom Loop Attack is a low-level attack that exploits a **legacy feature** of the CAN protocol.
- It effectively **freezes bus traffic for an arbitrary time** and could be used to delay a specific CAN frame to increase its latency or to generally remove bandwidth from the bus.
- The attack differs from others by being very difficult to detect:
  - the error counters are not increased, and the only symptom is that **frames arrive later than otherwise expected.**
  - If no timing analysis was done to calculate worst-case latencies of frames, then **the attack will look like an intrinsic transitory timing error**

Freeze Doom Loop Attack (2/2)



- The CAN protocol defines a dominant bit in the first bit of the inter-frame space (IFS) as a controller signalling an overload condition. It effectively freezes bus traffic for an arbitrary time and could be **used to delay a specific CAN frame to increase its latency or to generally remove bandwidth from the bus**.

- The Freeze Doom Loop Attack works by asserting a dominant bit on the CAN TX pin at the first IFS bit then monitoring the error recovery and again asserting a dominant bit in the IFS field at the end of the error recovery. This can be repeated an arbitrary number of times, in **effect freezing the bus for as long as desired**.

Outline

- Security overview
- Attacks to the CAN bus
- Mitigations techniques
- Final remarks on CAN bus security

Mitigation techniques

There are several main techniques for mitigating attacks

- **Intrusion detection**
  - This is a technique where the traffic on the bus is inspected for abnormal behaviors. Without hardware support it cannot generally prevent an attack but even so has a use in intelligence gathering and in post-incident forensics.
- **Security gateway**
  - This is a hardware approach using a device with two (or more) CAN bus interfaces. The gateway copies only legitimate traffic between the trusted bus (typically a vehicle control network) and an untrusted bus that contains a device that is potentially compromised.
- **Encryption**
  - This is generally a software technique (sometimes with hardware assistance) where an ECU protects its CAN bus traffic using cryptographic methods. Only receivers with a key can decrypt a message and verify its legitimacy. There are several issues around practical use of encryption for protecting CAN.
- **CAN security hardware**
  - These approaches use a hardware device included on a PCB that monitors the CAN signals to and from the CAN bus and provides various levels of protection.

IDS for CAN bus: software

- The goal of an IDS is to detect when a likely attack is occurring and to take some action.
- **Software IDS** is a software application with a standard CAN controller
  - there is little direct action that can be taken to prevent or halt an attack and is generally too late to prevent an attack.
- Software IDS can **collect data for post-incident analysis.** This could be very important for preventing a repeat attack on other systems.

IDS for CAN bus: hardware

- An IDS augmented by CAN security hardware does have the possibility of **mitigating attacks before they can cause harm**

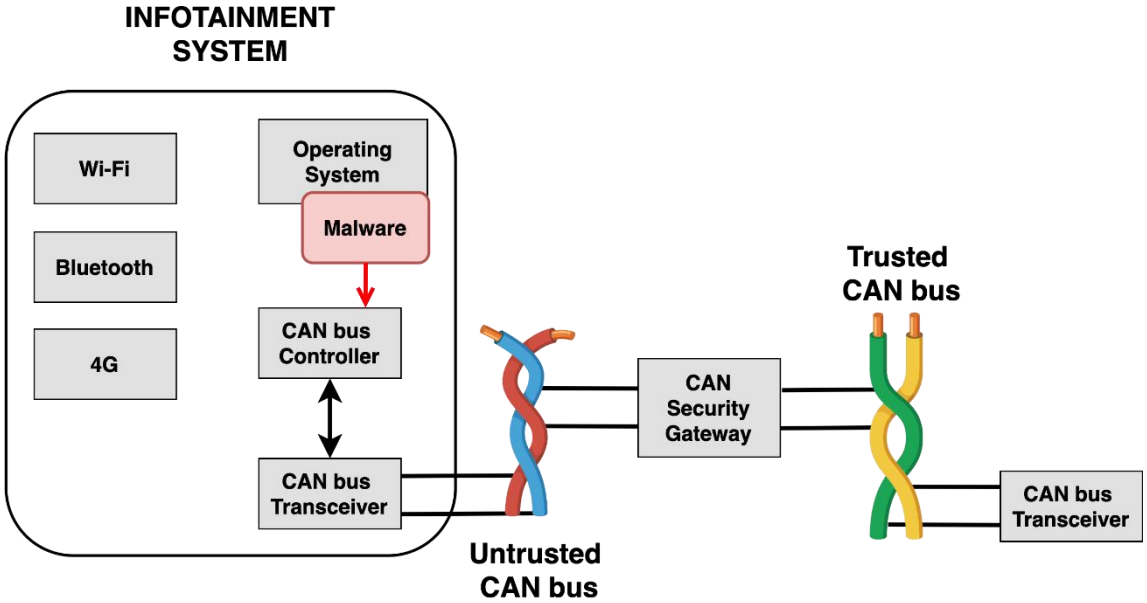
- **Hardware IDS** might consist of
  - a CAN controller with support for **generating interrupts** before the frame has been received allows the IDS to decide if the frame is spoofed and to inject an error frame to prevent receivers seeing it.

CAN Security Gateway (1/2)

- A CAN security gateway copies legitimate traffic back and forth between an **untrusted side** and a **trusted side.**
  - The trusted and untrusted sides are CAN buses.

- **A security gateway can protect the trusted bus in the following ways:**
  - **Protects from low-level protocol attacks:** the access to the trusted bus is only via CAN controller hardware in the gateway there is no opportunity to take direct control of the CAN TX pin and attack the CAN protocol on the trusted bus.
  - **Protects from denial-of-service attacks:** attempts to flood the bus will fail because the gateway can refuse to forward traffic outside of a predefined rules.
  - **Protects from spoofing attacks:** only predefined frames are permitted through the gateway.

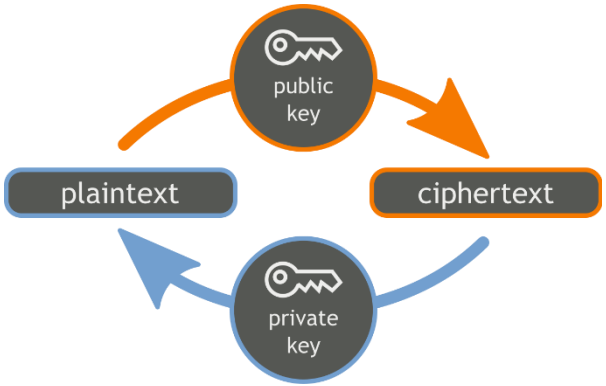
CAN Security Gateway (2/2)



- The CAN security gateway may be required to transmit **messages received from the untrusted bus across to the trusted bus.**
  - For example, frames containing new firmware for over-the-air (OTA) download functions. A security gateway should ensure these frames can only reach the control bus if genuine.

Encryption for CAN bus

- Encryption provides two protections for messages:
  - confidentiality;
  - authentication.
- Encrypting CAN frame payloads for confidentiality makes it harder to determine the sequences of exchanged frames but does not prevent an adversary with good resources from reverse-engineering communications



Encryption for CAN bus (1/2)

- Encrypting CAN frame payloads for **confidentiality** makes it harder to determine the sequences of exchanged frames but does not prevent an adversary with good resources from reverse-engineering communications

Encryption for CAN bus (2/2)

- Authenticating CAN frames is much more important: spoofing attacks rely on the receiver not knowing if the message is authentic (i.e., from a legitimate source).
- The cryptographic **authentication** process is generally as follows:
  - The sender computes a message authentication code (MAC) from the details of the message, using a MAC algorithm and a secret key.
  - The MAC is attached to the message, and both are sent together.
  - The receiver performs the same computation on the message with the same secret key. If the result matches the received MAC, then message must have come from a sender who knows the secret key.

Encryption for CAN bus: caution!

- Encryption is limited in what it can achieve:
  - it can only mitigate message content and sequence attacks. While this does prevent re-wiring spoofing attacks;
  - it cannot address denial-of-service attacks;
  - it does not detect or prevent the Flood Attack, Freeze Doom Loop Attack and Bus-Off Attack.

Encryption for CAN bus: issues

- There are several issues for implementing encryption on CAN:
  - **Bandwidth.** The bus **load is increased** due to including a MAC with every message.
  - **Performance.** How long it takes to generate and authenticate messages and how long it takes to start up and begin communications are **critical in a real-time control network**.
  - **Key distribution.** Both ends of communication **must share the same key.** These keys need to be generated securely, put into devices, kept secret and potentially replaced later with new ones.
  - **Resisting attacks.** There are specific attacks on **poorly-implement encryption systems**.

Outline

- Security overview
- Attacks to the CAN bus
- Mitigations techniques
- Final remarks on CAN bus security

CAN bus Security Remarks (1/2)

- **Prevent double receive errors.**
  - The software should include a sequence number in the CAN frame so that an automatic retransmission by the hardware can be detected and spurious frames discarded. A single bit that is toggled by the CAN drivers is sufficient.
- **Perform timing analysis.**
  - Key to detecting and preventing bus flood attacks is knowing the worst-case real-time behavior of legitimate bus traffic.
- **Do not allow ODB-II or infotainment direct access to the control CAN bus in a vehicle.**
  - These are the two biggest threats to CAN bus security. They should be behind a security gateway.
- **Use hardware interlocks for critical operations.**
  - Do not allow OTA downloads or disable airbag commands in a vehicle to be activated without a hardware interlock that a physically present human must activate.

CAN bus Security Remarks (2/2)

- **Use hardware to guard the CAN TX pin.**
  - The low-level protocol attacks depend on direct access to the CAN TX pin. Use an external CAN controller connected via SPI, a microcontroller with a pin multiplexer that can be permanently locked to the internal CAN controller.
- **Use an IDS to log suspicious traffic.**
  - At a minimum there should be evidence collected for post-incident analysis.
- **Use a hardware-assisted IDS to prevent attacks.**
  - An IDS that can destroy CAN frames before they are received can guard ECUs from spoofing attacks.
- **Use encryption to protect against re-wiring attacks.**
  - For situations where an attacker is motivated to re-wire a CAN bus then encryption with authentication should be used to protect critical messages (such as odometer readings).

CAN Security

Simone SODERI

IMT School for Advanced Studies Lucca

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

CAN Security
In depth-material

Simone SODERI
IMT School for Advanced Studies Lucca

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

Vehicle Hacking hardware

- Vehicle hacking via OBD-II requires a physical connection to the car.

The basic hardware to hack the CAN bus:

- A laptop (with Linux) is necessary for sniffing and crafting packets
- CAN to USB interface or Arduino CAN bus shield
- OBD-II pigtail cable

Vehicle Hacking software

- Linux O/S (even if vehicle hacking can be performed on Windows too)
- SocketCAN package
  - to make CAN communication like the ordinary use of TCP/IP sockets (vcan0 interface)

```
sudo modprobe vcan
sudo ip link add dev vcan0 type vcan
sudo ip link set up vcan0
```

  - provides useful applications and utilities aka **can-utils**
    - **candump**
    - **cansend**
    - **cansniffer**
- **Wireshark** with SocketCAN can decode CAN bus protocol
- **python-can** library provides CAN support for Python

CAN Security
In depth-material

Simone SODERI
IMT School for Advanced Studies Lucca

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it