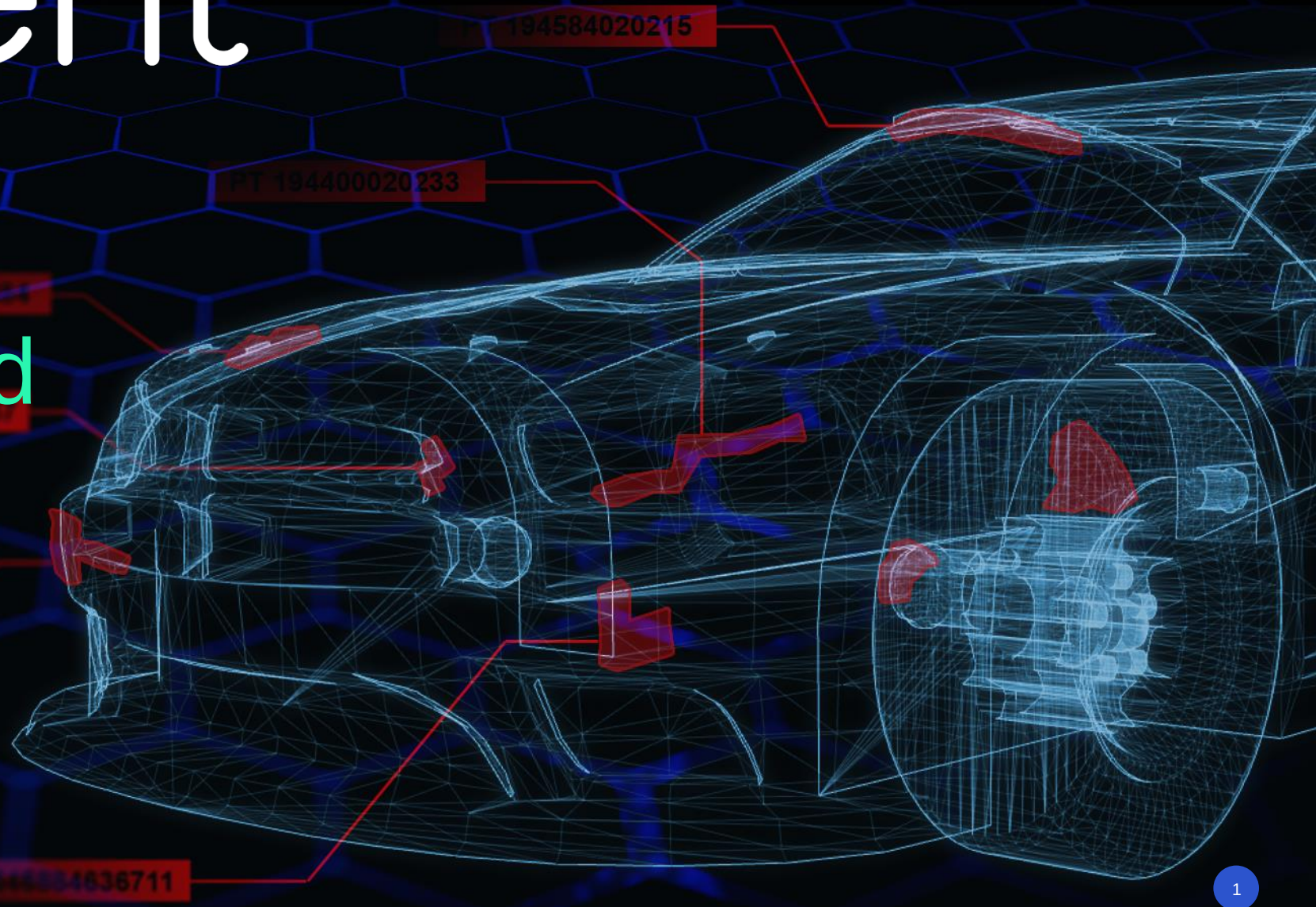
Security of EV vehicles and infrastructure



Aleksander Gorkowienko
Senior Consultant in Cybersecurity

About myself

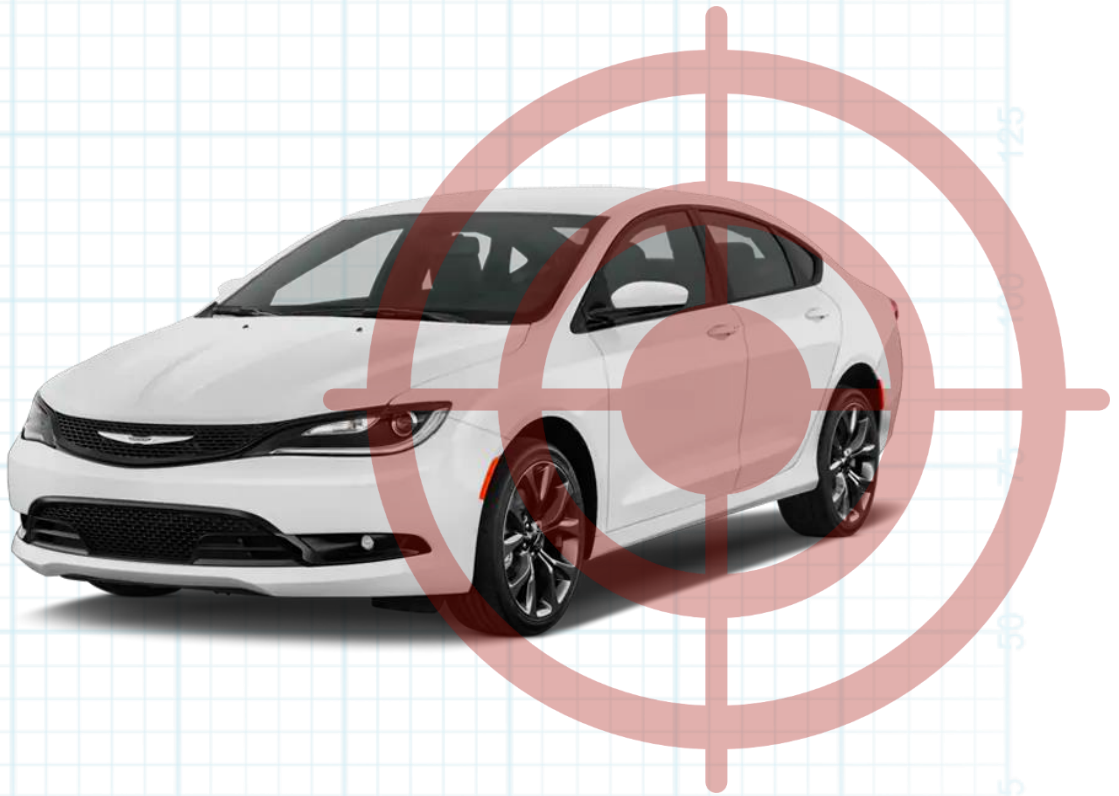
Aleksander Gorkowienko

- Cybersecurity advocate, practitioner and researcher with more than 20 years of experience
- A part of the Spirent SecurityLabs team (UK branch)
- Specialised in security of automotive and industrial systems, IoT and telecommunication
- Speaker at various international cybersecurity conferences
- Running complex security projects around the globe
- Experimenting with AI applied to cybersecurity
- Every day learning something new

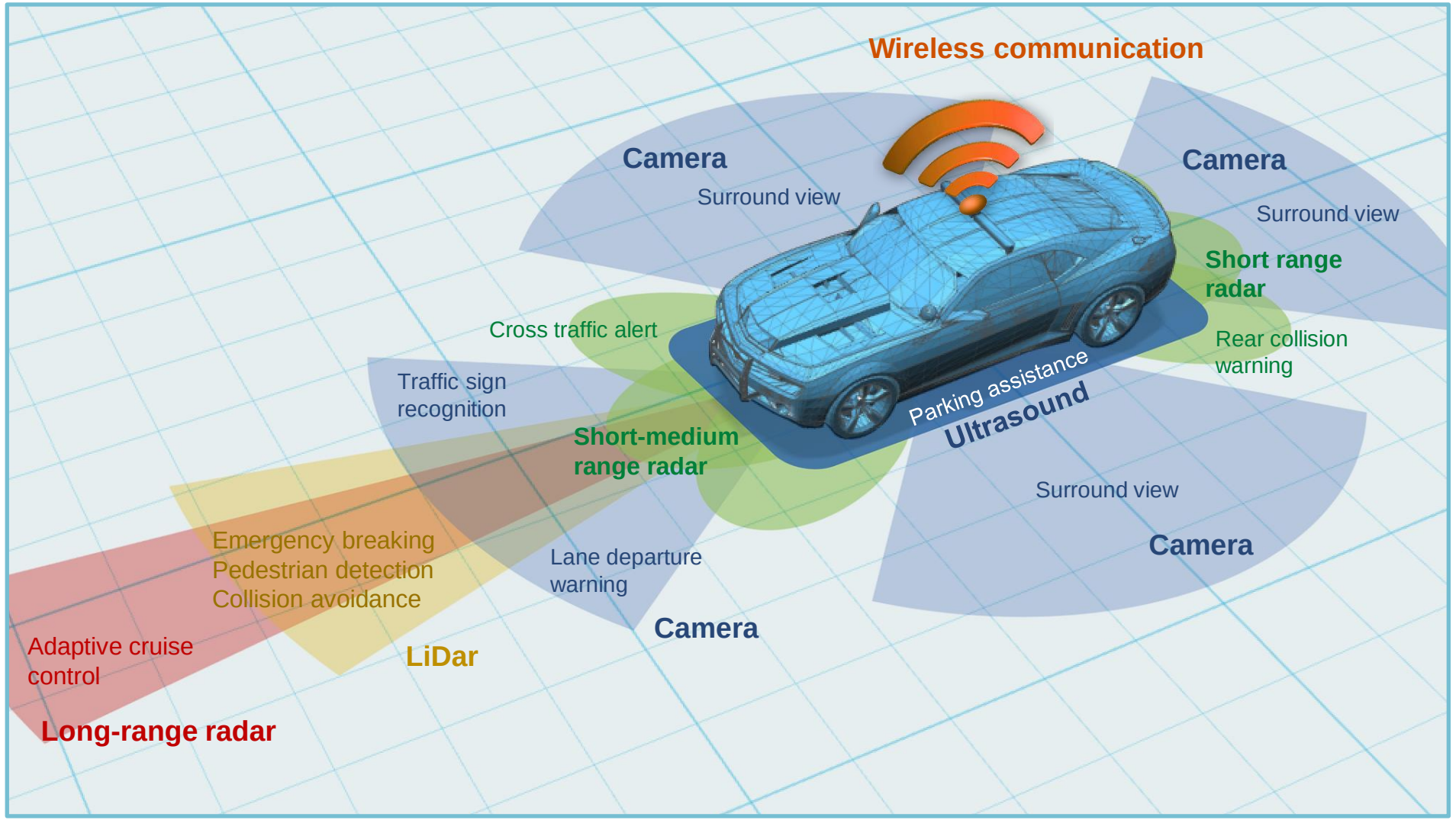
What is so special about EV cars and infrastructure?

What EV vehicle is?

- The whole EV car is 100% “fly-by-wire”. Car is full of electronic components (ECUs): sensors, actuators, processing units, etc. → multiple points of failure.
- Highly advanced data-driven energy management
- Realtime two-way connectivity to automotive cloud → someone always has real-time access to your car
- Car generates an enormous amount of data (lots of it: sensitive). Some of it processed locally, some is send to the automotive cloud, typically without your knowledge.
- Most connected cars utilize mobile apps to control various functions → mobile apps = perfect source of security weaknesses



Modern EV cars are full of automation and sensors



Typical ADAS sensors used in modern smart and self-driving cars

EV charging ecosystem is complex

Key elements:

- Power grid
- EV charging station
- Electric vehicle
- User/driver
- IT infrastructure
  - Industrial systems
  - Cloud systems
  - Communication protocols

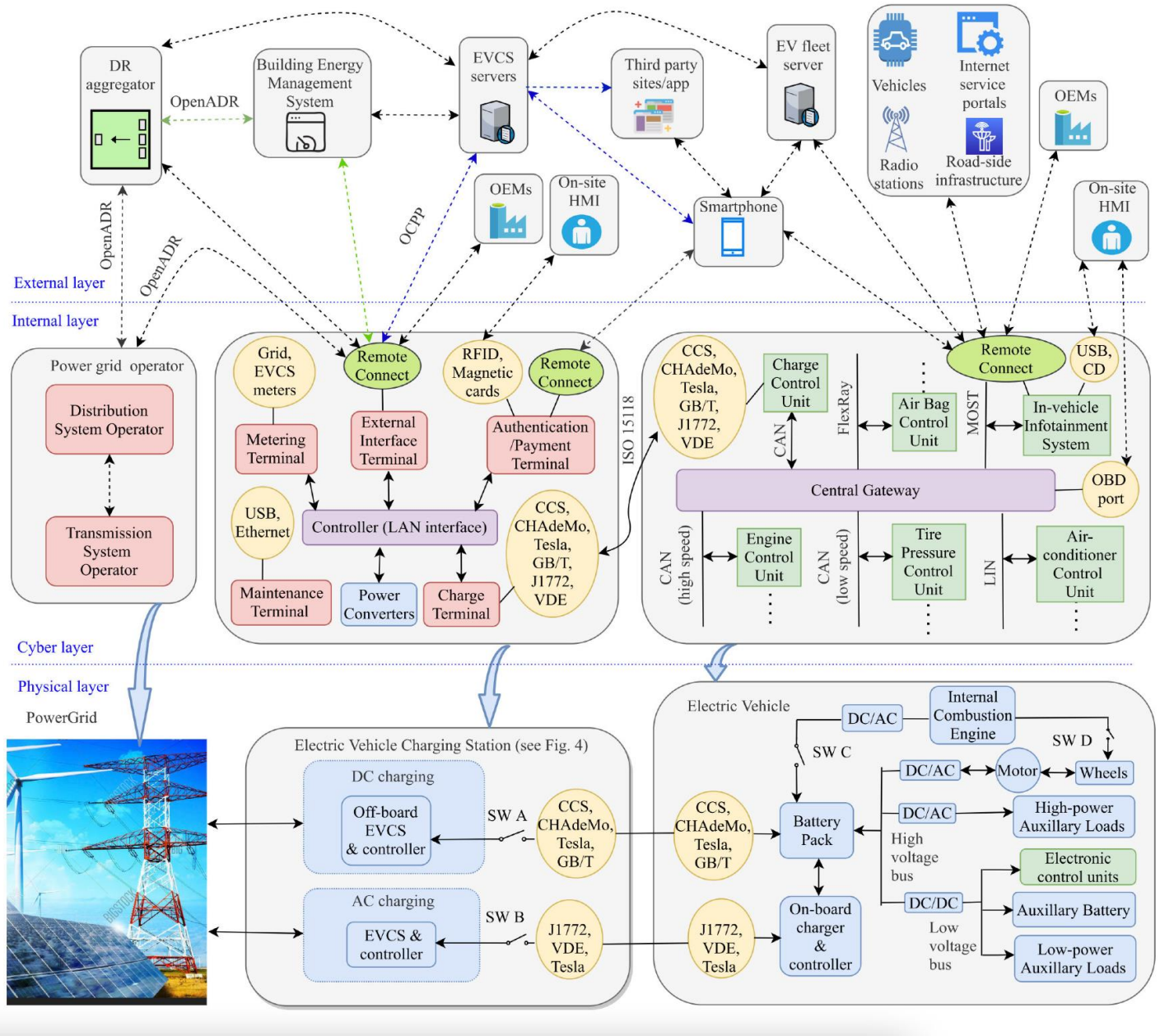


Image source: Cybersecurity of Smart Electric Vehicle Charging: A Power Grid Perspective (2020)

EV chargers vary by manufacturer and charger speed

- There is still no universal standard for EV charging connectors and plugs. Instead, automakers have largely converged on a small set of different designs.
- Level 1 and level 2 (AC) are nearly universal (J1772 connector). **Draws 1.5 kW – 20 kW.**
- Level 3: DCFS (DC fast charger): CCS charger (modified J1772 connector), CHAdeMO charge port. **Draws 50kW – 350 kW.**
- Tesla… well, is Tesla – it has all proprietary sockets, so adapters are needed on the road. ☺

Image source: MUO

Fun fact: a “dedicated” medical issue related to EVs!

It’s called **Range Anxiety**

Range anxiety is the fear of running out of power before reaching your intended destination and being unable to recharge the battery.

Range anxiety is an emotion that, surprisingly, all EV owners experience sooner or later, especially when they are new drivers.

A traditional vehicle can be filled up on almost any highway, but EV charging stations aren't yet as common. Therefore, the possibility of running out of power in the middle of nowhere and being unable to recharge the car is a valid concern.

Image by: luis_molinero on Freepik

Hacking EV cars and infrastructure

What hackers can do with your car?

A quick answer:

Turn your shiny new car…

A quick answer:

Turn your shiny new car… into…

A quick answer:

Turn your shiny new car… into… **this.**

“Doors and windows” to the smart car

Remote access

- Bluetooth
- WiFi
- RFID
- TPMS
- LTE
- 5G

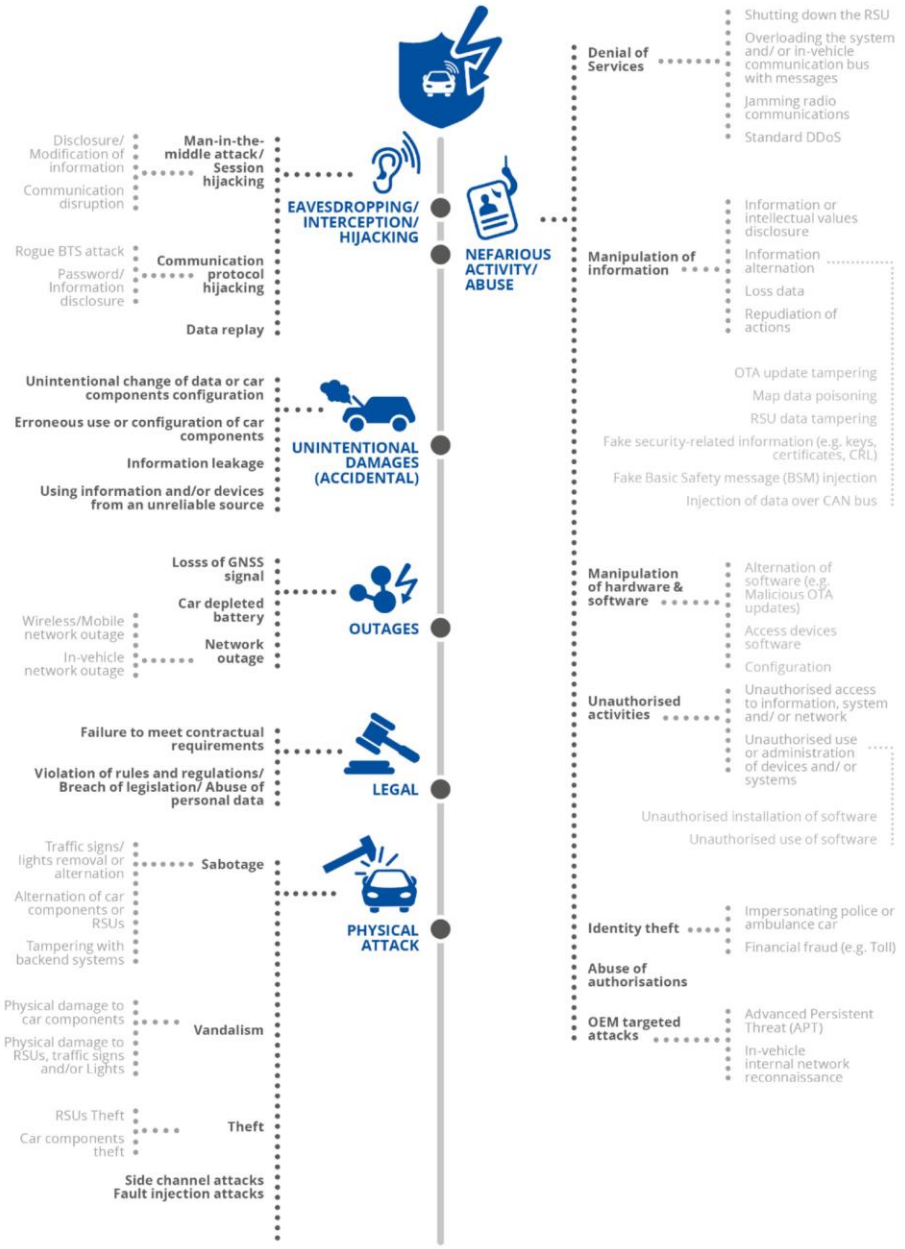
Short range communication

Long range communication

Direct physical access

- Multimedia ports
- OBDII port
- Debug ports

Threat taxonomy according to ENISA



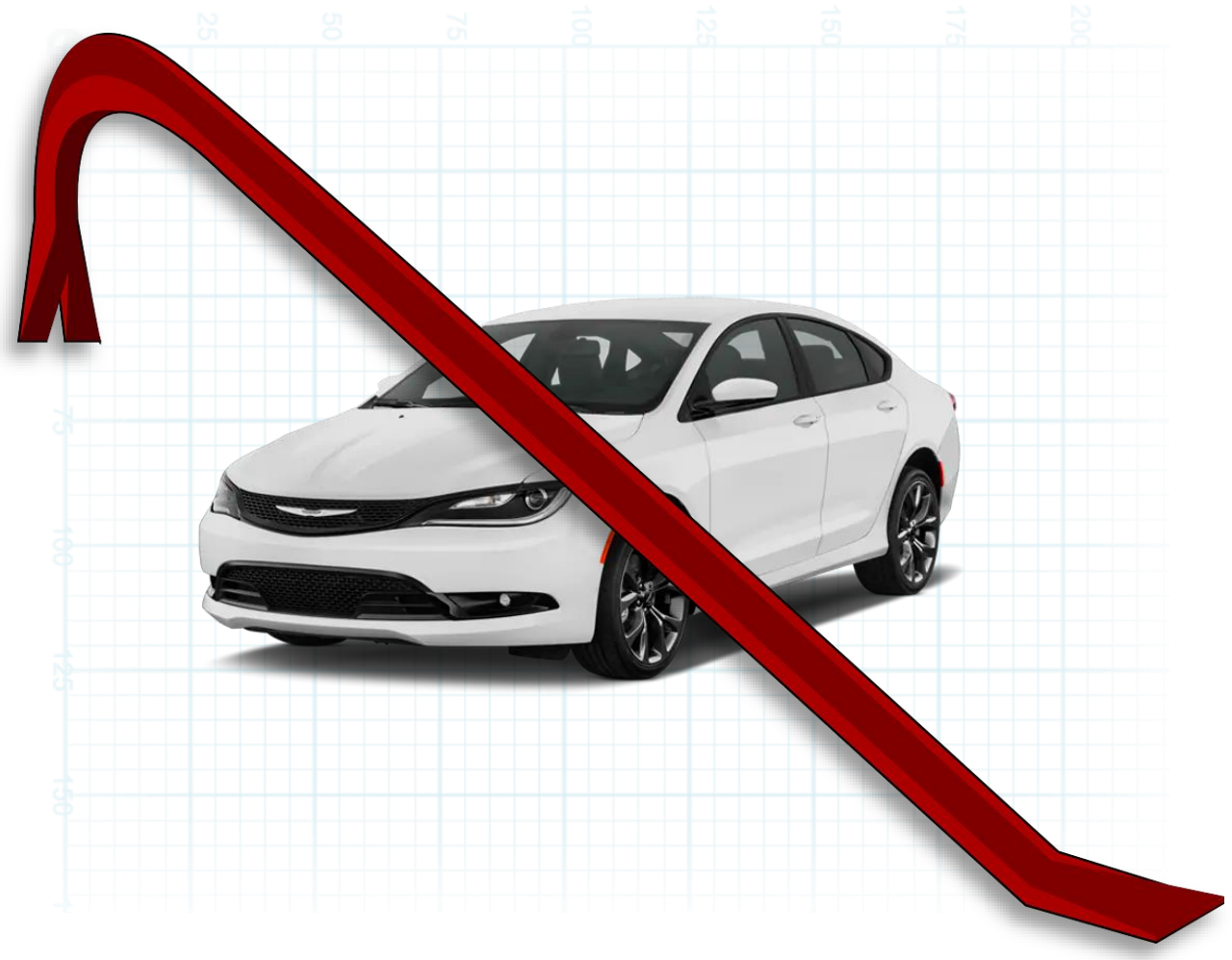
Threats to modern smart and EV cars

Standard ”menu” of threats:

- Unlock and steal a vehicle
- Remotely take over a vehicle
- Remotely stop and shut down a vehicle (denial of service)
- Spy on vehicle occupants, steal their sensitive data
  - Access GPS data and track a vehicle
  - Circumvent safety systems and cause crash/pre-crash conditions etc.
  - Install malware on the vehicle

Threats specific to self-driving cars:

- Circumvent autonomous navigation system and e.g. stop or change the route of the vehicle
- Smart sensor spoofing
- Circumvent car’s AI
- Change the computer’s logic and priorities in crash conditions

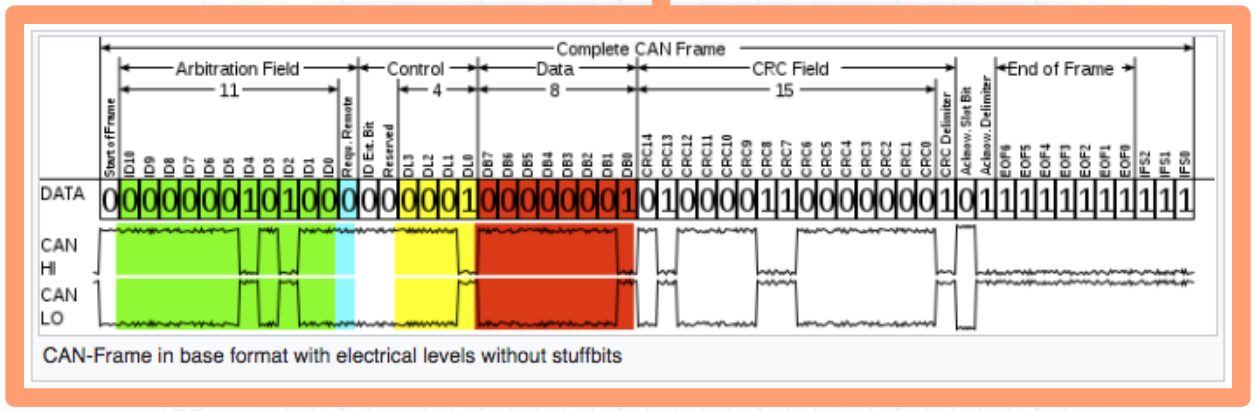


CAN Bus and car hacking

The **Controller Area Network (CAN)** bus is a standard developed by Bosch and Intel in 1983.

We use the CAN Bus version which was released in the 1990's.

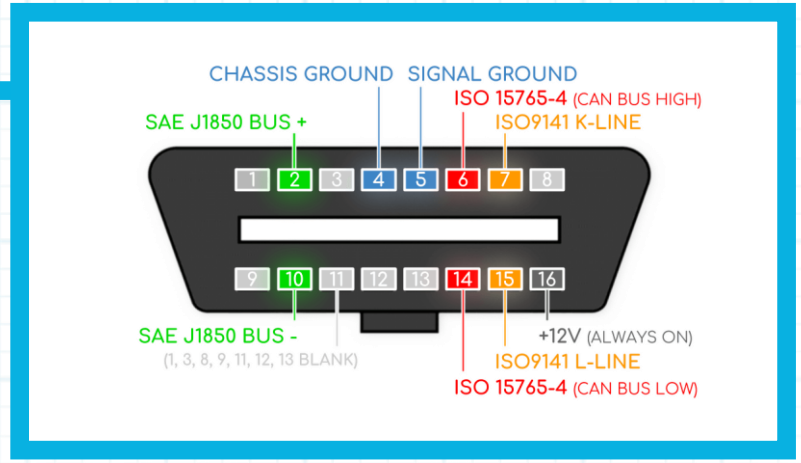
CAN is a serial communications protocol that allows distributed real-time communication and control between various vehicle components like: brakes, power steering, windows, A/C, airbags, cruise control, infotainment systems, doors, battery and recharging systems etc.



CAN-Frame in base format with electrical levels without stuffbits

OBD II (on-board diagnostics) and CAN bus are the entry point in monitoring and also (partially) are controlling a car (pretty much any car since 1996!).

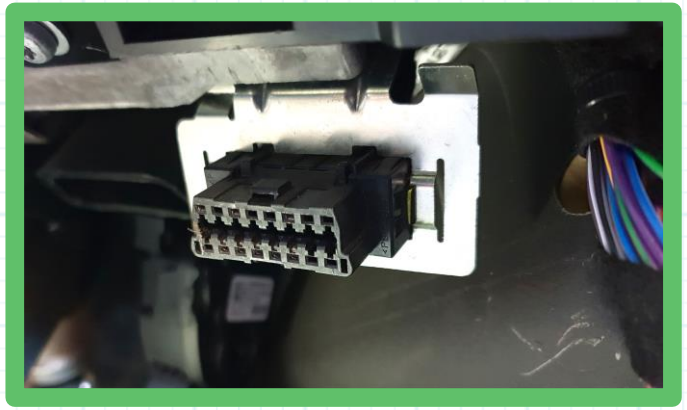
Smart cars and self-driving cars use this protocol!



CAN Bus and car hacking

CAN is a **broadcast serial bus standard** for connecting (electronic control units) ECUs. All of them are connected to the same “internal network” meaning **there is no central computer**.

When an ECU sends a message, **every other ECU on the bus receives it** and can choose to respond to it or ignore it.



ODB II socket

WiFi to ODB II interface
(connected to my own car, btw.)

Hacker’s way of thinking:

- If we can reach CAN Bus **– we can potentially “talk” to all electronics in the car**.
- Consider the simplest approach first: secretly connect your ODB II wireless device to the ODB port in the victim’s car and access it over WiFi or Bluetooth. **The car is immediately under your control!**

Fun fact about the CHAdeMO interface

- CHAdeMO is a DC charging protocol enabling V2G operations for electric vehicles, i.e. charging and discharging of their internal battery.
- **CHAdeMO lacks secure communication features** and it relies on CAN communication.
- Vehicles that plug to chargers with CHAdeMO **expose their CAN bus to the (untrusted) charger**.
- Security concerns on the CHAdeMO are mostly related to the **unencrypted communication through the CAN bus**, and in particular to the ability of a malicious agent to control or program other car ECUs exposed by the CAN bus.
- In 2019, CHAdeMO announced that it is co-developing the next generation ultra-fast EV charging standard called ChaoJi. Currently ChaoJi is *not implementing any security for the communication* and will still communicate using the CAN bus (however, this can be changed in the future).

CHAdeMO electric vehicle connector

Image source: Wikipedia

So what about the EV chargers?

Security of chargers

Public EVSE devices offer a range of **methods for authenticating** a charging session, such as:

- Radio Frequency Identification (RFID) tags
- Smart phone
- Near Field Communication (NFC)
- Credit card chip/swipes.

These methods link the EV operator's account information (i.e., owner or driver) to the charging session for billing and tracking purposes. *However, in some cases you just need to plug it in*.

"Plug-and-charge" functionality is developed in ISO 15118-20 that allow the vehicle to automatically authenticate over the charging cable. This is done by using a **public key infrastructure (PKI)** that uniquely identifies each vehicle.

generation and storage of cryptographic materials has been the source of significant debate within the industry

Security of chargers

The existing public charging model is based on **unattended self-service**, meaning that charging stations may be in **remote locations without physical security**.

To charge an electric vehicle, it is often needed to communicate with Electric Vehicle Supply Equipment (EVSE).

There are several **components of an EVSE that are susceptible to exploitation**, such as:

- The communication channel between the vehicle and charging station
- The mobile app that may be required
- Malicious firmware updates to the EVSE
- EVSE remote management interfaces
- Wireless communication
- EV operator interfaces
- Cloud services
- Charger maintenance ports (such as: WiFi, USB, RS485 or Ethernet)
- Insecure user interfaces and hidden menues

Categories of EVSE vulnerabilities

- EV-to-EVSE interface vulnerabilities
- EV operator interface vulnerabilities
- EVSE Internet interface vulnerabilities
  - Web services
  - Internet-accessible EVSE services
  - Communications to backend servers and/or automotive cloud
  - EVSE maintenance interface and linked hardware/software vulnerabilities

Vulnerabilities found (so far)

- Unauthorised access to sensitive information about users and the charging device configuration
- Insecure API (e.g., having authorisation issues), allowing account takeover and remote control of all chargers
- EV charger firmware is not signed
- Weak credentials in use
- Some chargers internally use Raspberry Pi compute module → no secure boot, easy extraction of stored data, easy manipulation and hacking
- Hardcoded sensitive data and credentials
- Running an unauthorised code on EV charger
- Unencrypted communication (cleartext protocols in use)
- Mobile apps can be easily reverse-engineered to reveal weaknesses
- Some EVSE chargers could be located on the public internet or mobile network using Shodan
- Insecure remote management interfaces
- SQL injection
- RFID cloning

Vulnerabilities found (so far)

- Unauthorised access to sensitive information about users and the charging device configuration
- Insecure API (e.g., having authorisation issues), allowing account takeover and remote control of all chargers
- EV charger firmware is not signed
- Weak credentials in use
- Some chargers internally use Raspberry Pi compute module → no secure boot, easy extraction of stored data, easy manipulation and hacking
- Hardcoded sensitive data and credentials
- Running an unauthorised code on EV charger
- Unencrypted communication (cleartext protocols in use)
- Mobile apps can be easily reverse-engineered to reveal weaknesses
- Some EVSE chargers could be located on the public internet or mobile network using Shodan
- Insecure remote management interfaces
- SQL injection
- RFID cloning

Impact:

- Functional impact
- Financial and privacy impact
- Safety impact
- Power system impact

Vulnerabilities found (so far)

- Unauthorised access to sensitive information about users and the charging device configuration
- Insecure API (e.g., having authorisation issues), allowing account takeover and remote control of all chargers
- EV charger firmware is not signed
- Weak credentials in use
- Some chargers internally use Raspberry Pi compute module → no secure boot, easy extraction of stored data, easy manipulation and hacking
- Hardcoded sensitive data and credentials
- Running an unauthorised code on EV charger
- Unencrypted communication (cleartext protocols in use)
- Mobile apps can be easily reverse-engineered to reveal weaknesses
- Some EVSE chargers could be located on the public internet or mobile network using Shodan
- Insecure remote management interfaces
- SQL injection
- RFID cloning

Consequences:

Theft of electricity (account compromise and charging the cost to innocent legitimate users)

Theft of sensitive information about EV car users/drivers (PII), telemetry data and information about the infrastructure

DoS by preventing legitimate users from charging (”Denial of charging”)

Multiple high-power EV (level 3) chargers can be switched on and off remotely → de-stabilizing the power grid

EV chargers can be forced to act as a part of a botnet

Preventing the disaster

Follow the industry best practices

Image from: ENISA Good Practices for Security of Smart cars

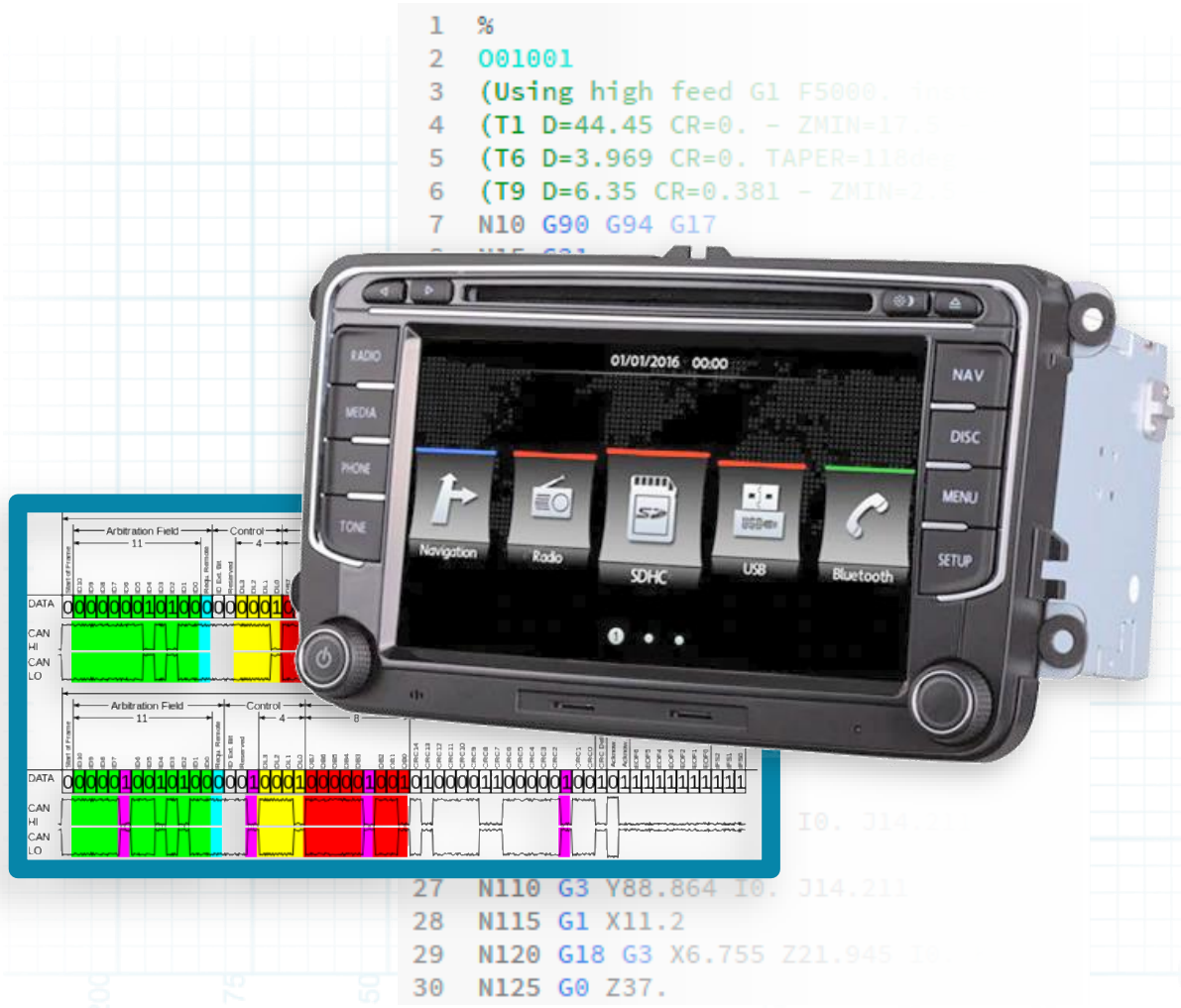
Recommendations for improving security of EV infrastructure

- Follow the industry best practices regarding the hardware and software development
- Follow the government regulations that are applicable to the EV charging infrastructure
- Security principles should be applied to the whole EV charging ecosystem
- Strong encryption of the communication is mandatory; PKI implementation should be considered
- It should be well considered which sensitive data should be collected and/or processed; the rest could be anonymised
- Procurement an use of third-party components should be logged into a Digital Bill of Materials (DBOM).
- The same concept should be applied for the software written for every component ranging from firmware to the UI through a Software Bill of Materials (SBOM).
- Publicly exposed devices should be tamper-proof and sufficiently protected from local and remote attacks
- Security testing and assurance as well as certifications and compliance test are essential

Image from: Freepik by frimufilms

Follow ISO/SAE 21434

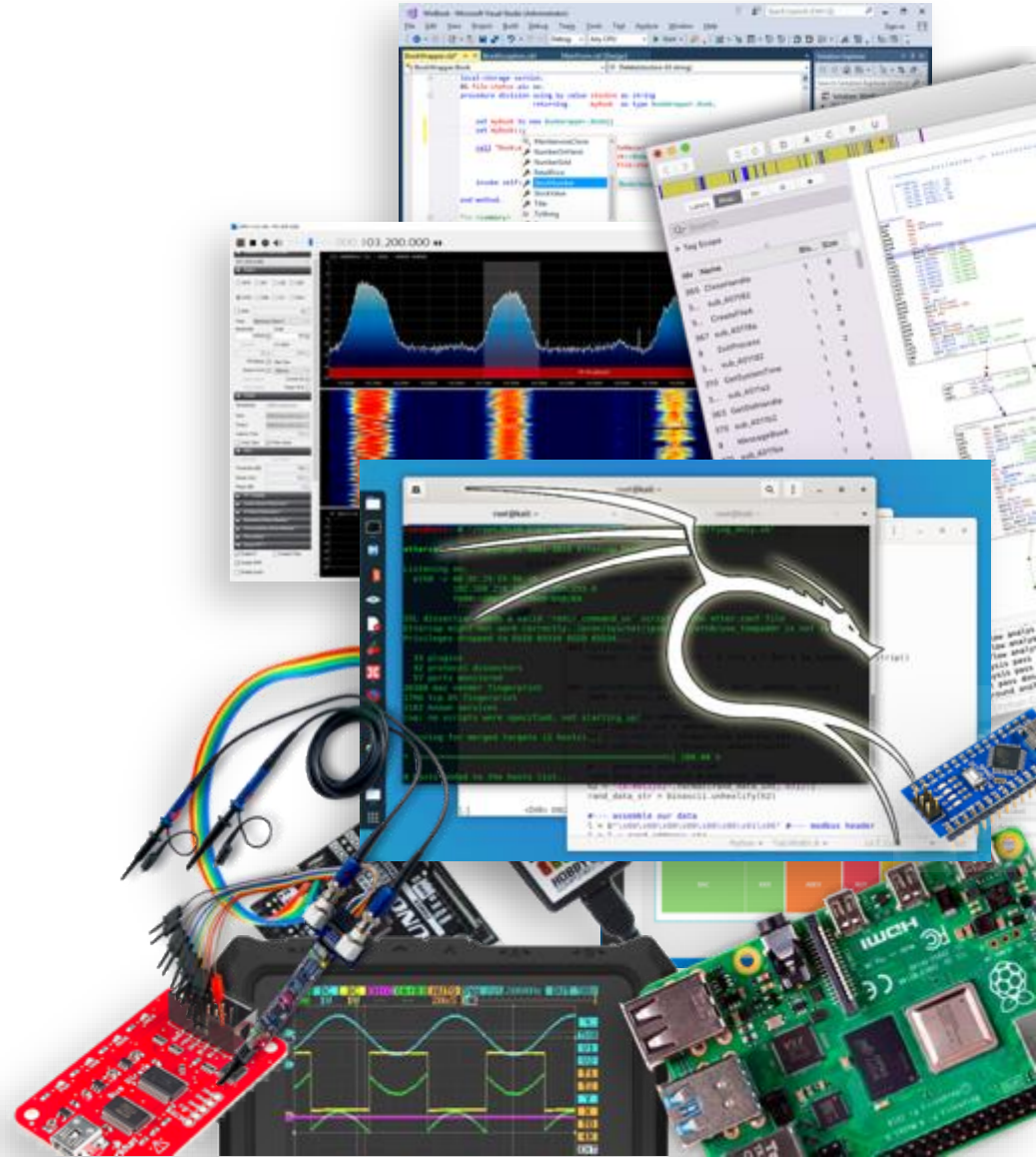
- ISO/SAE 21434 ” Road Vehicles – Cybersecurity Engineering” is an automotive standard that focuses on **managing cybersecurity risks in every stage of the lifecycle of a vehicle** and **across the entire supply chain**.
- With the volume of embedded software and increased connectivity in vehicles, compliance with ISO 21434 is essential for ensuring the security of EV software.
- The ISO standard covers all software devices within the vehicle, as well as connectivity to external systems.
- A few notes for software developers:
  - Adopt security culture: make security a priority
  - Choose the right programming language
  - Follow the secure programming design patterns
  - Add security and fuzzing to your CI/CD



Thorough independent security assessment is mandatory

Assessment steps are driven by the area of testing, methodology and our experience

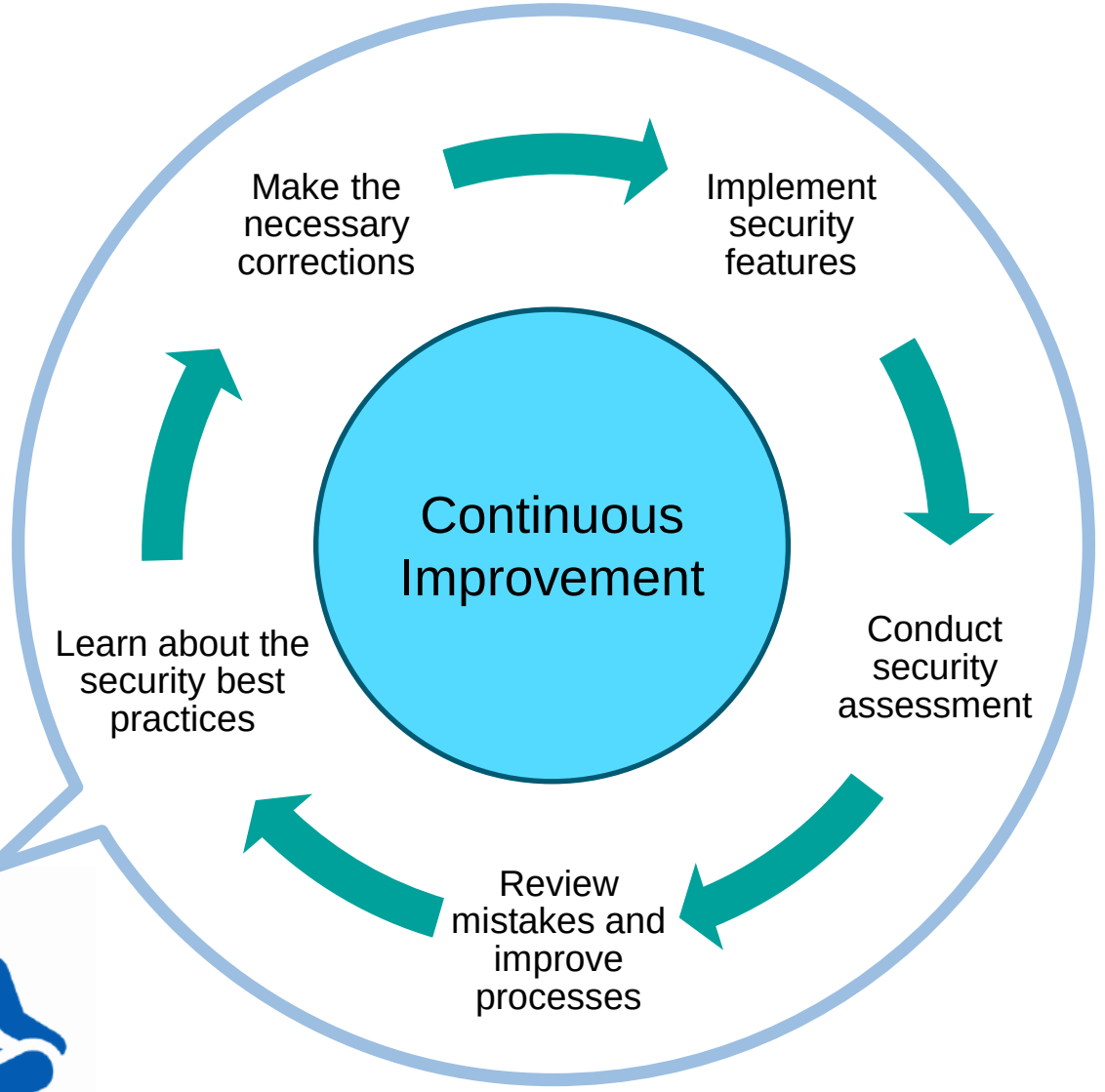
- **Fingerprinting** automotive systems in scope (port scanning, non-invasive services probing, passive and active reconnaissance)
- **Identifying security weaknesses** in processes, systems, configurations or protocols in use
- **ECU security:** hardware, software, fuzzing
- Issues with **authentication and authorization**
- **Automotive protocols:** CAN, LIN, I2C, Automotive Ethernet
- **Security of all interfaces** (wired, wireless, diagnostic, GUI, CLI)
- **Firmware update** mechanisms (e.g. OTA)
- Security of **wireless communication** (WiFi, Bluetooth, BLE, proprietary protocols, etc.)
- **Internet of Vehicle (IVI) communication:** CAN bus, ECUs, Sensors, Actuators, T-box
- **Source code review**
- **Stress-testing (spoofing and jamming)** your RF and PNT equipment in scope
- Researching security features of the hardware in scope and **writing own exploits**



Takeaways: what to do to prevent the disaster?

Change your mindset first

- **Security testing is** a **mandatory** part of lifecycle of any modern car or its components
- Security is a **continuous process**, where an organization is learning and improving their processes and the security posture all the time *(very much related to "The Toyota Way")*
- A system can be called "secure" only in a specific moment in time. It cannot be "always secure", therefore **regular testing is imperative**
- **Adopt zero-trust mindset**: ensure sufficient authentication is everywhere
- **Eliminate unnecessary public IP** addresses
- Have always **software and firmware up-to-date**.



securitylabs

Aleksander Gorkowienko
e: aleksander.gorkowienko@spirent.com
m: +44 (0) 7974431025