Cyber Security and Resilience of smart cars

Good practices and recommendations

DRAFT FOR COMMENT

2.0

TLP AMBER

SEPTEMBER 2016

About ENISA

The European Union Agency for Network and Information Security (ENISA) is a centre of network and information security expertise for the EU, its member states, the private sector and Europe's citizens. ENISA works with these groups to develop advice and recommendations on good practice in information security. It assists EU member states in implementing relevant EU legislation and works to improve the resilience of Europe's critical information infrastructure and networks. ENISA seeks to enhance existing expertise in EU member states by supporting the development of cross-border communities committed to improving network and information security throughout the EU. More information about ENISA and its work can be found at www.enisa.europa.eu.

ISBN: xxx-xx-xxxx-xxx-x | doi:xx.xxxx/xxxxxx

Contents

Executive Summary

Over the last few years, there have been a number of publications on attacks targeting automotive systems, and in particular smart cars. This report defines smart cars systems providing *connected, added-value features in order to enhance car users experience or improve car safety*. It encompasses use cases such as telematics, connected infotainment or intra-vehicular communication. The report excludes Car-to-car and car-to-infrastructure use cases, as well as autonomous vehicles.

An attack on a smart car would threaten the safety of passengers and other citizens. These threats are already having a big impact on car manufacturers, with millions of cars being recalled because of their vulnerability, not to mention the effects of the widespread media coverage of the issues.

The objective of this study is to identify good practices that ensure the security of smart cars against cyber threats, with the particularity that smart cars security shall also guarantee safety. The study lists the sensitive assets present in smart cars, as well as the corresponding threats, risks, mitigation factors and possible security measures to implement. To obtain this information, subject matter experts were contacted to gather their know-how and expertise. These exchanges led to three categories of good practices: *Policy and standards, Organizational measures, and Security functions*.

The protection of smart cars depends on the protection of all systems involved (cloud services, applications, car components, maintenance and diagnostic tools, etc.). However, the challenge resides mostly today in the security of car components and aftermarket products, where security functions have to be implemented in spite of several kinds of limitations: for example, security requirements may conflict with safety requirements. Furthermore, the very large number of interfaces to secure may lead to planning and cost issues; eventually, the long life of cars may create the need for dedicated security requirements.

The impact of attacks on a smart car has far-reaching consequences in terms of safety. The risk to the driver, their passengers and other users of the road makes it a matter of national and European interest. For this purpose, the following recommendations have been developed:

- **Improve cyber security in smart cars.** The industry actors should establish the good practices that effectively enhance the security of their products.
- **Improve information sharing amongst industry actors.** Information sharing helps industry actors challenge the relevance of their security mechanisms according to field information. Communities for information sharing already exist, and we recommend pursuing this effort.
- **Improve exchanges with security researchers and third parties.** Industry actors should enhance their contacts with third parties, especially from the security domain.
- **Clarify liability among industry actors.** Living in a heavily-tiered environments, industry actors should define processes to clarify their respective liability in case security issues arise.
- **Achieve consensus on technical standards for good practices.** The good practices listed in this report are meant as an input for a standardization effort, rather than being directly applicable to a specific car design. The details of the security requirements should defined in the context of standards.
- **Define an independent third-party evaluation scheme.** The existing safety standards for automotive systems only marginally address security, and we recommend to define an independent evaluation scheme.
- **Build tools for security analysis.** Industry actors can directly improve their security testing skills by building tools for security testing and security monitoring.

1. Introduction

Smart Cars integrate Internet of Things components to bring added-value services to drivers and passengers. These components communicate with each other and with the outside of the car (other cars, external services).

Over the last few years, there have been many publications on attacks on automotive systems. A few of them have been particularly under the eye of media, resulting in reputational damage for car manufacturers, especially since several attacks were demonstrated as cheap and easy, as in the example of a teenager unlocking and starting remotely a connected car[1] with only $15 of simple electronics gear.

Beside reputational damage, the cost of cyber security is becoming an issue for car manufacturers.[2] In the past years, vulnerabilities were found and resulted in an ever increasing number of recalls:

- Charlie Miller and Chris Valasek made a spectacular proof-of-concept of remote attack by taking control of a Jeep and sending it off-the-road[3], requiring *1.4 million cars* to be recalled;
- Security researchers hacked the BMW ConnectedDrive[4] and managed to remotely unlock cars, with even more industrial impact than the Miller/Valasek hack (*2.2 million cars* had to be recalled);
- More recently, even more vehicles (including most Volkswagen cars produced since 1995) have been shown vulnerable to an attack on remote keyless entry[5], thus once again increasing the size of impacted fleet. This last issue marked a steep progression of the number of potentially affected cars, which is in the order of magnitude of 100 million vehicles[6].

These threats have an impact not only on the security but on the safety of the passengers and of other citizens.

The objective of this study is to identify the good practices to ensure the security of smart cars against cyber threats, with the particularity that Smart Cars security shall also guarantee safety.

1.1 Objectives and scope

This study presents an analysis of the current situation in smart cars and considers the key factors in play, including: how connectivity changed the security model of cars, how the heavily-tiered car ecosystem can manage these issues, and how can security be integrated in existing, safety-oriented, product lifecycles. Therefore, the following objectives have been set:

- Review and analyse the architecture and interfaces of smart cars;
- Study the car ecosystem actors and lifecycles;
- List the main threats applicable to smart cars;

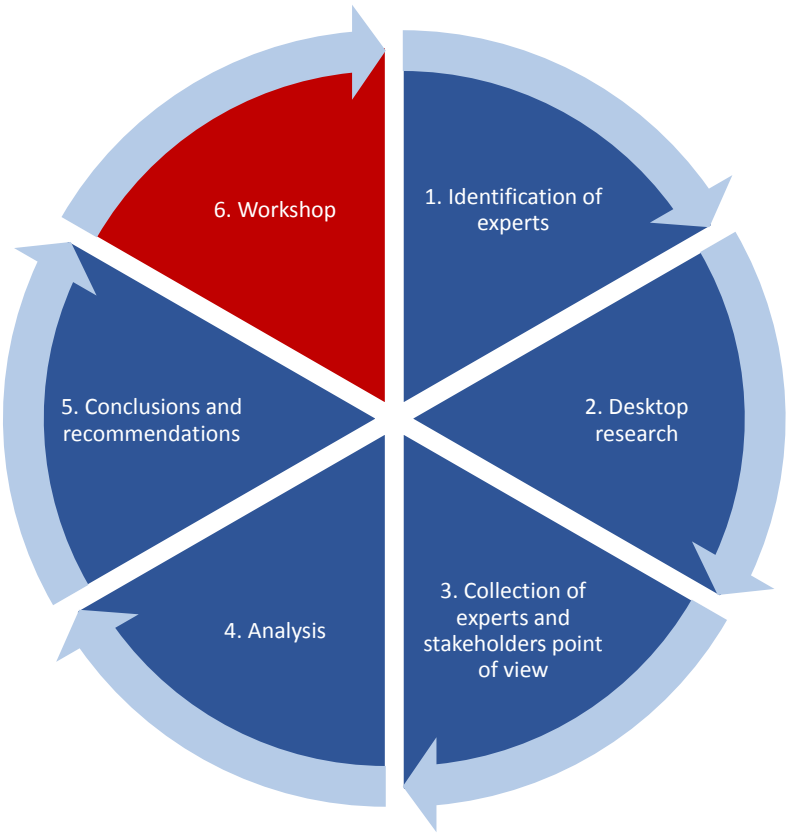
[1] http://www.forbes.com/sites/leoking/2015/02/23/14-year-old-hacks-connected-cars-with-pocket-money/
[2] Anthony Foxx, Secretary, U S Department of Transportation and Mary Barra, the chairwomen and CEO of General Motors Company, stress the importance of these issues in a keynote talk at the Billington Cyber summit 2016 https://www.youtube.com/watch?v=F-sPC2qHkq8
[3] http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/
[4] http://www.heise.de/ct/artikel/Beemer-Open-Thyself-Security-vulnerabilities-in-BMW-s-ConnectedDrive-2540957.html
[5] http://arstechnica.com/cars/2016/08/hackers-use-arduino-to-unlock-100-million-volkswagens/
[6] The affected company producing around 10 million vehicles a year.

- Collect good security practices;
- Analyse, in relation to the identified good security practices, gaps in current implementations;
- Explore limiting factors, impairments, constraints and potential incentives for the target audience to deploy these measures.

1.2 Methodology

This study was carried out using a five-step methodology (shown in Figure 1) which begins at the initial information gathering from official sources and experts in the field and ends in the development of a report summarizing the findings and the recommendations to the target audience.

Figure 1: Methodology used to carry out the study



1. **Identification of experts**: the first step was to identify the experts in the field of smart cars security. In order to obtain varied and well-balanced results, experts were selected from Manufacturers, tier-1 and tier-2 suppliers, aftermarket product suppliers, academics, and other actors, such as consulting companies, test and certification companies and governmental actors.

2. **Desktop Research**: initial research of already published documents in order to get as much information about communication dependencies as possible. This notably allowed to:
   - Identify the assets and threats specific to smart cars through desktop research and interviews with stakeholders in the smart cars domain;
   - Identify good practices to secure the critical assets (business and societal) from cyber threats
   - Analyse the most feared attack scenarios

- Present the good practices in a practical way by showing how to overcome the selected end-to-end attack scenarios.

3. **Collection of experts and stakeholders point of view**: we engaged stakeholders through interviews to understand the current status of security and their challenges. For that purpose, we developed a questionnaire to understand the challenges and needs of car manufacturers and their suppliers;
4. **Analysis**: the fourth step was to analyse all the data obtained, including the results of the interviews, gathering initial conclusions.
5. **Conclusions and recommendations**: the last step was to further analyse and contrast these results with the experience of the consortium and external sources.

The study was eventually validated with the stakeholders, through a review phase and a face-to-face validation workshop. We also stayed updated with regard to the C-ITS Platform[7] run by DG MOVE[8], to synergize efforts. Moreover input from the CARSEC[9] expert group was used to finalize the deliverable.

1.3 Target Audience

This report provides information on smart cars security including lifecycle (including the security maintenance in the field) and business perspective (not focusing only on technical measures). Therefore, the target audience is mostly **Car manufacturers, Tier 1 and Tier 2 suppliers, and Aftermarket suppliers.**

1.3.1 Car Manufacturers

Car manufacturers design new cars and select their equipment according to marketing considerations. Regarding manufacturing of the car itself, their role is mainly limited to the assembly of the various car components provided by their suppliers. They provide to their supplier functional, safety and security requirements for the components as well as qualification of the products.

They also have to take into account security, safety and privacy by design, especially since aftermarket components may be added to the vehicle later by the user.

1.3.2 Tier-1 and Tier-2 suppliers

Car manufacturing is a heavily tiered ecosystem. Car manufacturers integrate components provided by suppliers, which are labelled as “Tier-1”. While driving system are usually a prerogative of the manufacturer itself, Tier-1 suppliers may be in charge of manufacturing most of the components directly facing the final user. From entertainment systems to car seats, a large part of the car cost may be associated to components manufactured by Tier-1 suppliers.

While Tier-1 suppliers have direct contractual relationships with car manufacturers to provide car components, the ecosystem also includes suppliers labelled as “Tier-2”. Tier-2 suppliers only have contractual relationships with Tier-1 suppliers. They produce, for example, plastics, mechanical parts, molds, electronic components or software.

Also some Tier-2 suppliers may also become Tier-1, for instance Operative System (OS) providers for the multimedia system have direct contact with the car manufacturer to allow more control, customization or

[7] See http://ec.europa.eu/transport/themes/its/c-its_en.htm
[8] See http://ec.europa.eu/transport/index_en.htm
[9] See https://resilience.enisa.europa.eu/carsec-expert-group

monetization on the applications, or also secure components providers in order to propose personalization or Over The Air (OTA) management services.

1.3.3 Aftermarket suppliers

Customers can also buy aftermarket products from other vendors; for example smart dongles used on the OBD-II port, providing additional features to their car. More traditional aftermarket products may include media players or third-party GPS.

1.4 Structure of this document

This document contains the following sections:

- **Key aspects of the smart cars.** This section details the typical architectures found in smart cars, as well as the relationships between main actors of the ecosystem. It eventually lists the sensitive *assets* of smart cars;
- **Threats.** This section elaborates on the assets by listing the *main threats* on smart cars. *Sample attacks* taken from the state-of-the-art are given as illustration of the way these threats can lead to car compromising. Eventually, a few significant attacks are further detailed into *Attack scenarios*, to clarify the different steps necessary for an attack, as well as the expected attack potential required for such attack;
- **Key findings.** This section describes the *good practices* able to mitigate the aforementioned attacks. It also puts these good practices in perspectives by describing the current *gaps and challenges* for their implementation, as well as the *constraints and incentives* for the actors of the ecosystem;
- **Recommendations** intended to overcome gaps and challenges in the implementation of good practices;
- **Glossary and abbreviations**

Further details are given in appendix:

- Appendix A details the calculation of attack potentials used in the attack scenarios,
- Appendix B gives further details on the good practices.

2. Key aspects of the smart cars

2.1 Definition

In this study, we define Smart Cars as systems providing connected, added-value features in order to enhance car users experience or improve car safety.
This study excludes car-to-car and car-to-infrastructure use cases, as well as autonomous vehicles. While V2X[10] is not addressed as a use case, V2X interfaces will be taken into account in the report, whenever their existence has an impact on the assets or threats to be considered.

It encompasses use cases such as:

- Telematics, used for example in the context of fleet management or geo-fencing;
- Connected infotainment, which provides an integrated multimedia offer with potential added value services (such as the access to an application store) and can access driving information (such as speed) as well as control non-essential functions (such as air conditioning);
- Intra-vehicular communication, where the infotainment connections can be shared with user devices, typically by creating a hotspot within the vehicle.

2.2 EU Policy context

From a regulation point of view, few initiatives are specific to smart cars:

- The European Parliament voted in 2015 to mandate the implementation of the eCall[11] system in cars commercialized after April 2018;
- More generally, since smart cars consist of cyber-physical components, they are concerned by:
  - The General Data Protection Regulation[12], replacing the *Data Protection Directive*[13];
  - The *Network and Information Security Directive (NIS)*[14].

Other initiatives have been launched, independently from these regulations. In particular, the EU Commission launched the AIOTI[15] Alliance in 2015, in order to enhance the dialogue between actors of the Internet of Things (IoT). An AIOTI workgroup is specifically dedicated to Smart Mobility, which includes IoT use cases pertaining to the car industry.

A 2015 report[16] from the AIOTI Smart Mobility workgroup may be used as an introduction to other initiatives in Europe on this topic:

- The European Technology Platform for Road Transport Research (ERTRAC)
- Research and Development initiatives funded via Horizon 2020
- The C-ITS Deployment Platform

[10] The notion of V2X encompasses Vehicle-to-infrastructure (V2I), Vehicle-to-vehicle (V2V) and Vehicle-to-pedestrian (V2P) use cases.
[11] See https://ec.europa.eu/digital-single-market/ecall-time-saved-lives-saved
[12] See http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2016.119.01.0001.01.ENG
[13] See http://eur-lex.europa.eu/legal-content/en/ALL/?uri=CELEX:31995L0046
[14] See https://ec.europa.eu/digital-single-market/en/news/network-and-information-security-directive-co-legislators-agree-first-eu-wide-legislation
[15] See https://ec.europa.eu/digital-single-market/alliance-internet-things-innovation-aioti
[16] See http://ec.europa.eu/newsroom/dae/document.cfm?action=display&doc_id=11822

- The Electronic Components and Systems for European Leadership (ECSEL)
- The Important Project of Common European Interest (IPCEI)
- Main SDO, Alliances & Open Source initiatives
- FIWARE
- An exploration of national or company initiatives

While most of them are strongly related to autonomous driving, several have to take into account cybersecurity issues already present in today's cars.

2.3 Typical architecture and assets

We describe in this section the typical architecture of smart cars, and list the assets that can be distinguished within such architectures. The architecture of subnetworks and protocols may vary from a vehicle to another, therefore Figure 2 provides a high-level overview of such systems.

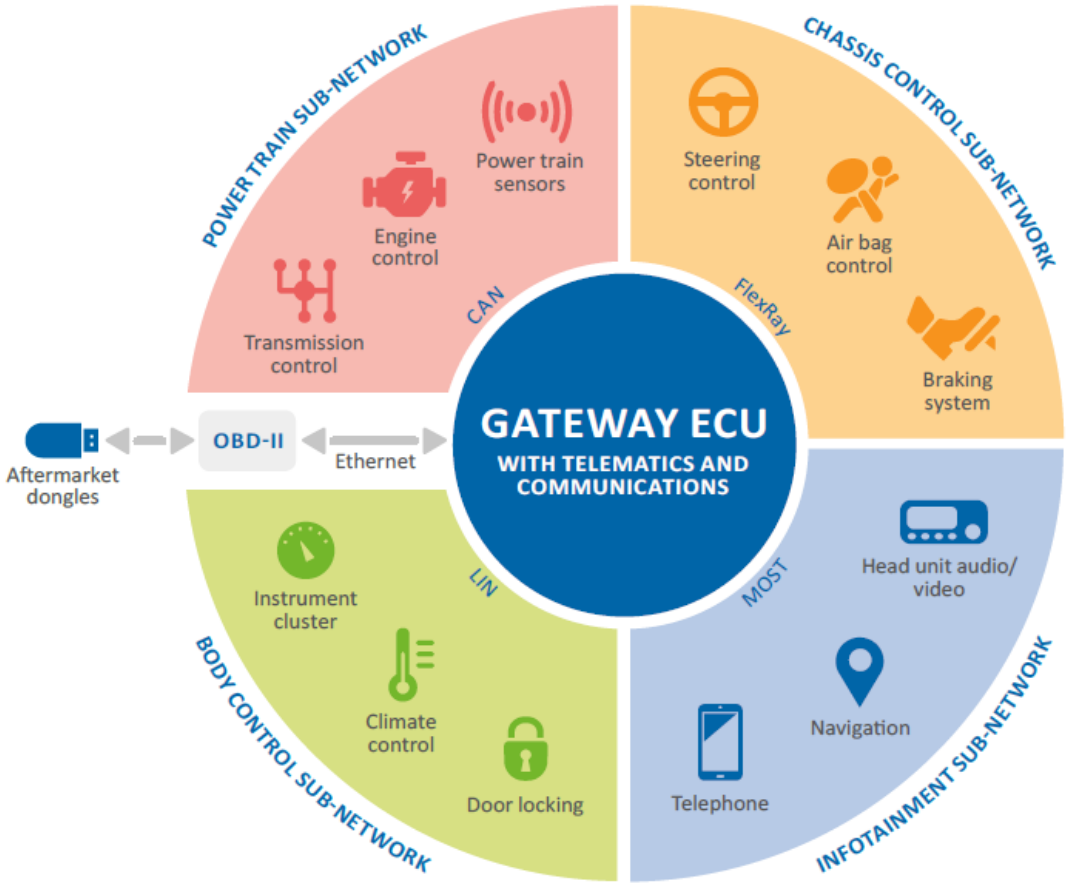


Figure 2 : High-level architecture of a smart car

Most car architecture distinguish between different domains, interconnected by a central gateway, as shown in Figure 2. Domains correspond to different, or sometimes independent, features of the car. All these components may cause risks, should they be compromised. The impact of these risks may vary between safety, security or privacy concerns. For this reason, components of a smart car are described as assets and require appropriate protection. Figure 3 hereafter lists these assets.

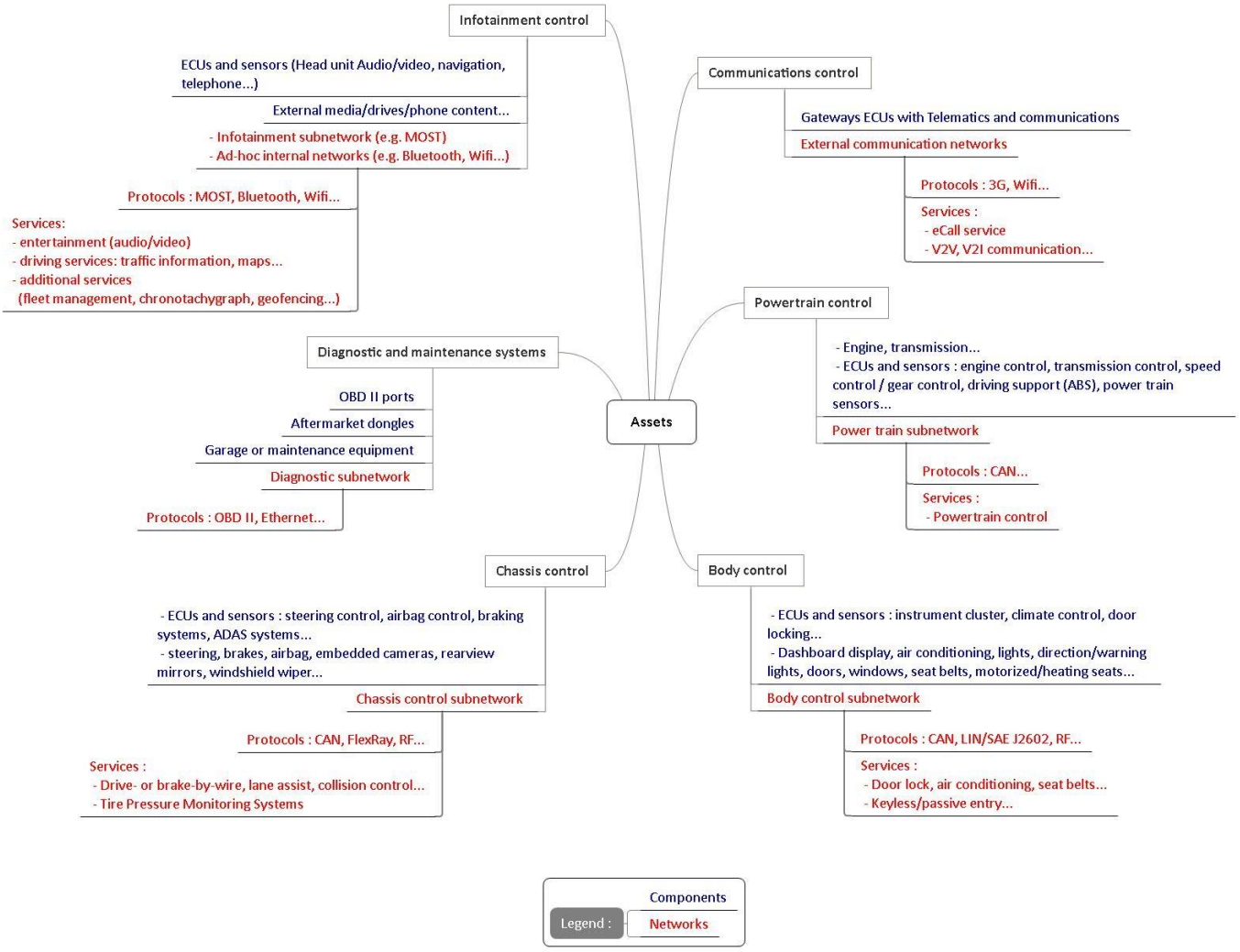


Figure 3 : Smart cars assets

We distinguish the components according to the following categories:

- Powertrain control
- Chassis control
- Body control
- Infotainment control
- Communications control
- Diagnostic and maintenance systems

2.3.1 Powertrain control

This domain is in charge of the chain between the energy source of the car and its transformation into propulsion.

ECUs and sensors

Modern cars are composed of many embedded electronic control units (ECU) that control mechanical or electronic systems of the vehicle[17]. While ECUs are different from a domain to another, here are a few general explanations on the ECUs and TCUs architecture:

- As other IoT systems, automotive devices often rely on the ARM platform for applicative processors. Processor for other usage may come from many origins. However due to constrained operating environment in automotive (temperature, humidity, lifetime), specific declinations of processors, not *commercial-grade* but *automotive-grade*, are used.
- For increasing security, and in particular for vehicular communications, these systems may also rely on a Trusted Platform Module (TPM), a smart card core or a Hardware Security Module (HSM)[18].
- ECU/TCU applications may be written directly in assembly or rely on a specialized real-time operating system, such as QNX (Blackberry), VxWorks (Wind River Systems), Integrity (Green Hills Software), Automotive Grade Linux or also Linux Genivi.

Subnetwork

The powertrain subnetwork typically relies on the CAN protocol.

CAN, an ISO standard since 1993, is by far the most well-known and popular bus, to which most of the ECUs of the vehicles are connected. There may be several CAN buses in a vehicle, interconnected by a gateway, to isolate the most critical functions (such as powertrain management) from the less critical (such as multimedia). The traffic on this internal network varies from a solution to another; in some instances the network can support several hundreds of messages per second[19]; CAN bus is a prominent example and has been thoroughly studied by many researchers[20].

CANs, as other protocols described in this report, faces issues related to bandwidth, scalability or security; protocols such Ethernet or MOST, introduced in 2008, are perceived as potential solutions to some of these issues, but remain expensive. These protocols are, today, still limited to a subpart of the network (multimedia, assisted driving…). Ethernet, however, is progressively developed[21] and may be used to replace protocols such as CAN in future cars.

Other components

This domain includes physical systems such as internal combustion or electrical engines, as well as the transmission, drive shafts, and wheels.

2.3.2 Chassis control

This domain is in charge of the control of the vehicle frame with regard to its environment.

[17] Such as: powertrain, brake, suspension, airbag
[18] Embedded in the processor itself
[19] See for example Hacking a Tesla Model S: What we found and what we learned, Kevin Mahaffey, Lookout
[20] Most notably in *Adventures in Automotive Networks and Control Units*, Valasek/Miller
[21] See http://standards.ieee.org/findstds/standard/802.3bw-2015.html

ECUs and sensors

ECUs are similar than those found in the powertrain domains (see section 2.3.1). They allow the control of functions such as steering control, airbag control, braking systems, or ADAS systems.

Subnetwork

The subnetwork typically relies on the CAN protocol, but also on protocols such as CAN (see section 2.3.1), FlexRay, or RF (e.g. for Tire Pressure Monitoring Systems). FlexRay, introduced in 2008, is faster than CAN and designed for drive-by-wire applications.

Other components

This typically includes the steering and brakes, but also airbag, embedded cameras, rearview mirrors, or even windshield wiper.

2.3.3 Body control

The body control is in charge of the body, which means most of the time the passenger's compartment and trunk.

ECUs and sensors

ECUs are similar than those found in the powertrain domains (see section 2.3.1). They allow passengers to control various functions such as instrument cluster, climate control, or door locking.

Subnetwork

The subnetwork typically relies on the CAN (see section 2.3.1), LIN/SAE J2602 (for door lock, air conditioning, seat belts…), or RF protocols (Keyless/passive entry systems). LIN, a value-oriented variant of CAN introduced in 2002, is based on a single wire, has simpler controllers and offers lower bandwidth.

Other components

This typically includes the dashboard display, air conditioning, but also the lights, direction or warning lights, the doors, windows, seat belts, and even motorized or heating seats.

2.3.4 Infotainment control

This domain is generally separated from the remainder of the body. It includes navigation services, communications (telephone, etc.) as well as entertainment services (head unit audio/video).

ECUs and sensors

ECUs are similar than those found in the powertrain domains (see section 2.3.1). They allow passengers to control various functions such as the Head unit for audio/video content, but also navigation, or interactions with the user's telephone. Services offered through this domain can vary greatly, for example:

- entertainment services (audio/video)
- driving services such as traffic information, maps...
- additional services such as fleet management, chronotachygraph, geofencing...

These services drive the architecture selection of infotainment ECUs:

- For infotainment systems, operating systems from the mobile industry may also be used in ECUs (Windows CE, Android, Tizen or WebOS)
- QNX is also used in systems dedicated to the integration of users' smartphones into the vehicle systems. For example, it is used in Apple Carplay and Android Auto technologies, which allows the end-user to get the display of a mobile phone mirrored to the infotainment display, and grant access to its mobile applications.

Subnetwork

The subnetwork typically relies on protocols such as MOST, but also on ad-hoc networks using Bluetooth or Wifi. Infotainment systems rely on wireless connectivity provided either by an embedded UICC or by an end-user device (smartphone) connected by Bluetooth or with a USB cable.

Other components

External media that are directly connected to the infotainment components, such as drives or phones, should also be considered as an asset.

2.3.5 Communications control

This domain, contrarily to the previous ones, is not a subnetwork, but more frequently a set of communication features offered by a Telematics control unit (TCU), acting as a gateway.

Gateways ECUs with Telematics and communications

The gateway provides both the connectivity and most of the security protections intended for the communications (firewalling, authentication features…). It collects data from the various ECUs using one of the vehicle data buses and provides Internet remote connectivity through an embedded GSM module or using driver's smartphone for instance. This unit is generally also coupled with a GPS to obtain vehicle positioning information. A number of use-cases that are leveraging TCU connectivity are:

- Remote diagnostic of a breakdown
- Crash reporting and emergency warning (eCall, that will be mandatory in Europe in 2018)
- Stolen vehicle tracking or geo-fencing
- Remote engine start
- Fleet management, for instance for rental car companies (for example for trip tracking or diagnosis)
- Insurance, for pay-as-you-drive insurance plans
- "smart driving assistant" (e.g. for fuel efficiency or to improve driving habits)
- Inform driver of battery charge for electronic vehicle

External communication networks

The TCU typically provides 3G or Wifi connectivity to provide several kinds of services, for example eCall, but also V2X communication. Other protocols are possible, as shown in Figure 4, which gives an example of external interfaces found in a smart car. These typically include interfaces intended for long range communication, as well as wired or wireless interfaces intended for local use.

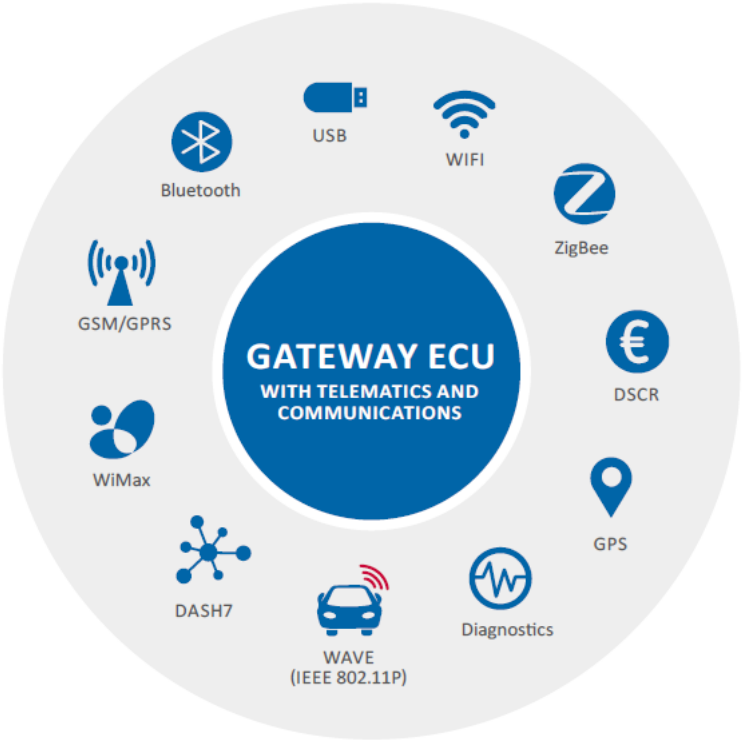


Figure 4 : An example of external interfaces of a smart car

Besides wired protocols such as USB or diagnostics, TCUs often provide various wireless protocols, as detailed hereafter.

Long-range wireless protocols

Telematics also rely on wireless connectivity[22] provided either by an embedded UICC. Mobile protocols such as GSM/GPRS/3G/4G may be used in a variety of context, but the most prominent are the eCall service and the capacity of providing OTA updates to car component firmwares. Smart cars also use GPS as part of their localization features.

Intra-vehicle wireless protocols

Bluetooth and Wifi are frequently provided as a protocol of choice for intra-vehicular communication, although the state-of-the-art suggests possible alternatives, such as ZigBee, Passive RFID, UWB or 60 GHz mm Wave[23]. Usually, communication costs for the TCU are supported by the car manufacturer, whereas they are supported by the end-user for the infotainment. Wireless protocols are also used in two different contexts:

- Near-range to relatively long-range protocols can be used for communication with sensors, for example DASH7, used for Tire Pressure Monitoring Systems (TPMS)

[22] Such as: 2G, 3G, 4G

[23] See Connected Vehicles: Solutions and Challenges - IEEE - Ning Lu, Nan Cheng, Ning Zhang, Xuemin (Sherman) Shen, Fellow, Jon W. Mark

- Wifi or Bluetooth connection may be used, but mostly to communicate with smartphones, using dedicated protocols[24]. Wearables might be the next types of components to benefit from such interfaces to the vehicles[25].

Inter-vehicle, or Vehicle-to-infrastructure wireless protocols

Inter-vehicle communications use a specific band allocated for ITS communication (5.9 GHz Band, called DSRC). Such communications typically use protocols such as

- WAVE (Wireless Access in Vehicular Environments), which is a mode of operation used by IEEE 802.11-compliant devices to operate in the DSRC band;
- DSRC (Dedicated Short Range Communications), *not to be mistaken with the DSRC Band*, which is a standard based on IEEE 802.11a;
- IEEE 802.11p, which is based on the same ASTM Standard E2213-03 as DSRC.

The state-of-the-art also suggests possible alternatives, such as DSA[26] or WiMAX for V2I communication[27].

Protection of communication typically rely on a PKI deployed specifically for this purpose. Work in the European Union on this matter is coordinated under the Connected and automated driving (C-ITS) deployment platform[28], which aims at harmonizing the PKI and trust model for the European Union.

Other components

External media that are directly connected to the infotainment components, such as drives or phones, should also be considered as an asset.

2.3.6 Diagnostic and maintenance systems

Diagnostic and maintenance systems are external systems interfaced with the car through a dedicated port. We also include aftermarket dongles in this category, since they use the same interfaces. It should however be noted that they do not necessarily provide maintenance or diagnostic features.

OBD II ports and **Garage or maintenance equipment**

Various maintenance and diagnostic equipment can be plugged on cars via the OBD-II[29] ports. They can be standalone equipment, such as handheld scanners, or comprised of applications running on a PC or tablet.

Aftermarket dongles

Aftermarket telematics components such as "smart dongles" also have OBD-II connectivity, as well as external bluetooth or cellular connectivity. They are often built upon the same set of components as the competition (SoC, sensor packages, CAN transceiver chip...). They may also include debugging interfaces (for

[24] Sor example Mirrorlink, CarPlay or Automotive Link
[25] See http://www.surewise.com/car-warranty/articles/how-wearable-tech-influences-smart-cars/
[26] See Connected Vehicles: Solutions and Challenges - IEEE - Ning Lu, Nan Cheng, Ning Zhang, Xuemin (Sherman) Shen, Fellow, Jon W. Mark
[27] See A Comparative Study between 802.11p and Mobile WiMAX-based V2I Communication Networks, Ikbal Chammakhi Msadaa, Pasquale Cataldi and Fethi Filali
[28] http://ec.europa.eu/transport/themes/its/c-its_en.htm
[29] The OBD-II interface is also called a “diagnostics plug”, and is available on all vehicles sold in Europe since 2001).

example via mini-USB), configured to emulate a network adapter (i.e., once connected, the TCU appears as a device on the network).

Diagnostic subnetwork

The subnetwork diagnostic is usually performed directly on the CAN bus (see section 2.3.1), through the OBD-II port.

2.3.7 Security, safety and privacy concerns

Assets are related to safety in several ways:

- Compromising *powertrain* or *chassis* ECUs and networks may obviously cause a vehicle to behave in an unexpected way, for example if an attacker illegitimately compromises ignition steering, brakes, speed or gear control, or even driving support (ABS);
- Compromising *body* ECUs and networks Systems that may increase harm to the passengers, should they malfunction:
  - Airbag or safety belts,
  - Door force-lock used for child protection,
  - The windshield wiper,
  - Alerts in the vehicle, dashboard display, notably speed, collision or lane departure warning...
  - Air conditioning,
  - Motorized or heating seats,
  - Rear view mirrors as well as automated windows or roof...

  These systems may also cause a disturbance on surrounding vehicles, for example if there is a disruption in headlights, or direction/warning lights;
- *Infotainment* ECUs and networks may also cause safety issues: incorrect navigation data may lead the car to unsafe areas, and a disturbance of the audio in the entertainment system may distract the driver

More specifically, the networks of the car can be specifically targeted and cause the same safety risks:

- Internal networks (for example the CAN bus, but it also includes wireless networks such as TPMS): a disruption or integrity breach on these networks may result in a loss of control of a vehicle;
- Cell connection of the car may also have adverse impacts on safety, for example in the case of a spoofed firmware update triggered by SMS;
- Local network (e.g. Wi-Fi, BT) and connection to user phones theoretically leads only to the entertainment components of the vehicle. But as the study shows, the lack of isolation between entertainment and driving systems might result in safety-related vulnerabilities from these entry points. This reasoning might also be extended to other local connections such as a wireless keyfob;
- Vehicle-to-infrastructure and Vehicle-to-vehicle communications, which could lead to accidents, were they disrupted or spoofed[30];
- The disruption of eCall, or other alert or alarms, may eventually cause additional concerns at an accident scene.

[30] While these functionalities are out of scope of this study, we still need to consider them as potential entry points for an attacker

Additional security concerns are found in several ways:

- An attacker may get an unauthorized access to functions not intended for users (fleet management, chronotachograph, geofencing...). This typically evokes **fraud** situations, but this may also cause the vehicle systems to malfunction and cause hazardous situations;
- **Trade secrets** may be at risk in several systems: TCU/ECU firmware, which might be sensitive with regard to the competition. Some industry actors, in particular, may be wary of the possibility of device cloning (for example the cloning of aftermarket products);
- More generally, **intellectual property** may also be threatened: Smart car applications, or infotainment application or media, which might be sensitive with regard to fraud (use of application copies obtained through unofficial stores, unauthorized copies of paid premium content…)

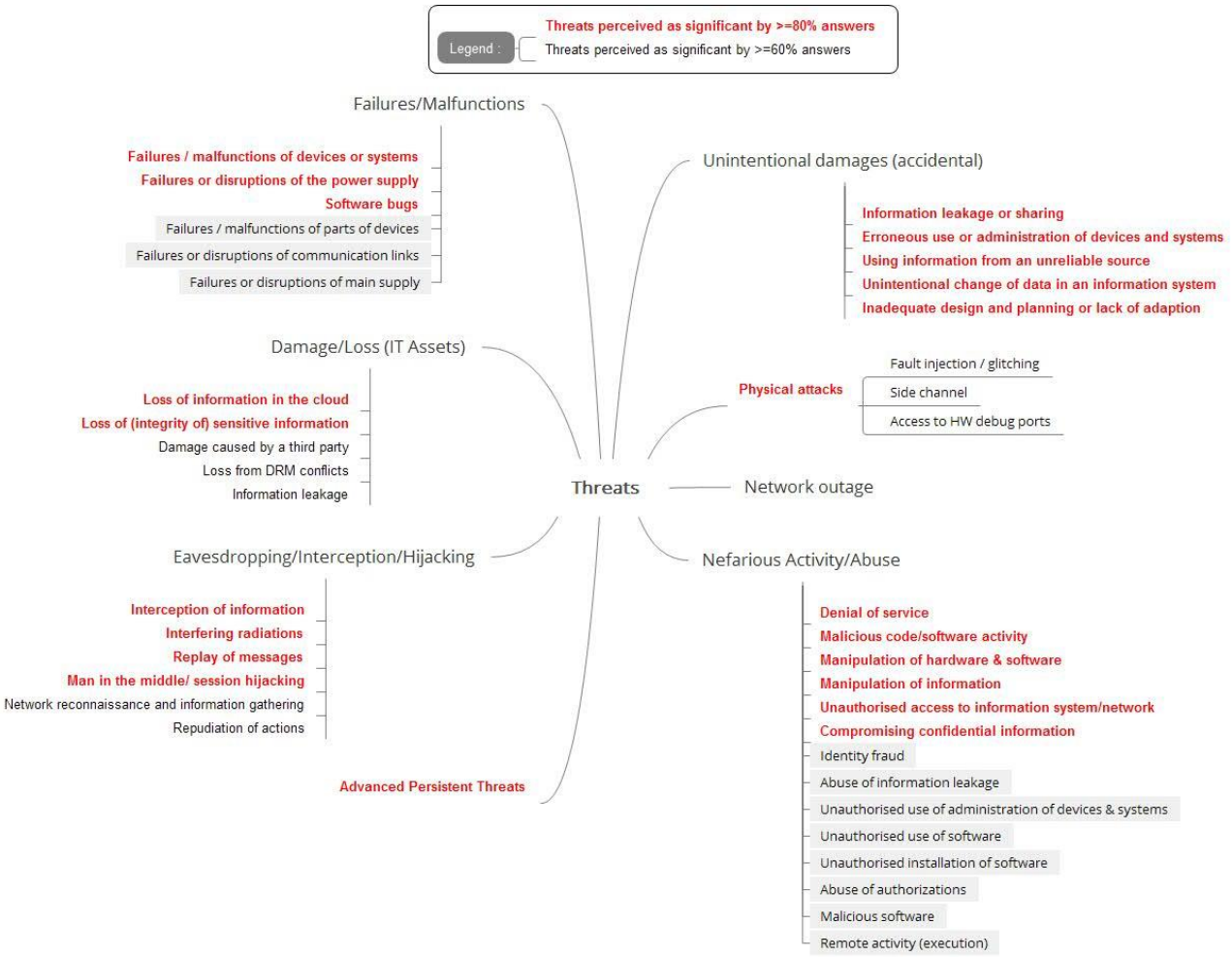
Data confidentiality and privacy are eventually at risk as well. For example, compromising embedded cameras may lead to privacy issues for the driver and passengers.

3. Threats

3.1 Main threats

This study builds upon the threats described in ENISA's previous work[31]. This set of threats has been compared with other available threat analyses[32] during the stocktaking phase of this study. While the presentation and categories of threats differ from analysis to analysis, the outcome of this comparison showed that the content remains the same, that nearly all threats found in ENISA's report are retained. The list of threats was discussed with experts during the interview phase, to focus on a restricted group of significant threats, as shown in Figure 5:

Figure 5: Interviews: Main threats as perceived by interviewees



[31] Cyber Security and Resilience of Intelligent Public Transport - Good practices and recommendations, December 2015

[32] Notably Characterization of Potential Security Threats in Modern Automobiles (NHSTA, 2014)

3.2 List of threats

CATEGORY	THREAT	VARIANTS AND DETAILS	ASSETS AFFECTED
Physical attacks	Side channel, fault injection, glitching, access to HW debug ports…	This may typically consist in several scenarios : tampering with the ECUs or TCUs (to recover keys or access physical debug interfaces); using the device electro-magnetic emanations or power usage to leak information (side-channel); use light, power or other means to alter the device behavior and ultimately gain access to protected data (glitch, fault injection). Physical attacks arise from a well-identified attack vector (physical manipulation of devices). They might lead to various types of risks, including the categories described hereafter as *Nefarious Activity/Abuse* or *Eavesdropping/Interception/Hijacking*.	**ECUs and sensors** (privileged debug interfaces of the ECUs, causing a cascading impact on all assets)
Unintentional damages (accidental)	Information leakage or sharing	This may typically concern administration errors in back-end services or errors when storing data intended for diagnostic in garages, for example.	Mostly IP-sensitive firmware of the **ECUs and sensors**, as well as private data transmitted over **subnetworks**
	Erroneous use or administration of devices and systems	Unintentional damages (accidental) may result from insufficiently trained personnel (for example when using diagnostic equipment), or from an incorrect OTA update pushed by the backend services.	**ECUs and sensors**, causing a cascading impact on all assets
	Using information from an unreliable source	Unintentional damages may cascade from ill-defined trust relationships: for example, trusting a third-party cloud provider with poor data protection, or failing to notify a Tier developer that the data they will store is sensitive.	**All assets**
	Unintentional change of data in an information system	Unintentional damages (accidental) may result from insufficiently trained personnel (for example when using diagnostic equipment), or from an incorrect OTA update pushed by the backend services.	**ECUs and sensors**, causing a cascading impact on all assets
	Inadequate design and planning or lack of adaption	Unintentional damages (accidental) may result from insufficiently trained personnel (for administration, design, operation…) causing for example incompatibilities between components, or lack of adaptation to the changing threat landscape (the use of vulnerable cryptography is an example of this).	**All assets**
Disasters and Outages	Network outage	A Network outage (for example from the ISP) may result in a denial of service for sensitive operations, such as OTA fixes for critical bugs or vulnerabilities. This is also true for internal networks failures. More generally, any design relying too much on connectivity exposes the vehicle to potential issues in case of outages. Vehicles should be designed to offer a usable degraded mode of operation in case of outage. See also Failures/ Malfunctions	**All assets**
Damage/ Loss (IT Assets)	Loss of information in the cloud	Sensitive data may be lost due to attacks or accidents when stored by third-party cloud service providers	Sensitive data stored by cloud service providers (these data do not appear

			on the asset list, but may typically be related to **infotainment control**)
	Loss of (integrity of) sensitive information	The (integrity of) sensitive data may be lost due to IT components wear and tear, causing potential cascading issues (in case of a key alteration, for example) See also Failures/ Malfunctions	**All assets**
	Damage caused by a third party	Sensitive data may be lost or compromised due to physical damages in cases of a traffic accident or theft.	Private data transmitted over **subnetworks**
	Loss from DRM conflicts	User data (traffic- or travel-related services, audio/video entertainment…) may be deleted due to DRM issues	Private data transmitted over **subnetworks**
	Information leakage	Private or sensitive data (such as payment information, driving habits…) may be leaked when the car is sold to another user.	Private data transmitted over **subnetworks**
Failures/ Malfunctions	Failures / malfunctions of (parts of) devices or systems	See Damage/ Loss (IT Assets) - Loss of (integrity of) sensitive information	-
	Failures or disruptions of the power/main supply	A failure of power supply has obvious safety issues besides security issues. However, security causes additional constraints. Typically, some security functions (for example anti-tampering mechanisms) should rely on separate and trusted power sources, to avoid both accidental security failures and potentially exploitable flaws for an attacker	**All assets**
	Software bugs	The presence of software bugs is a basis for potential exploitable vulnerabilities. The lack of a software measure for the Mean-Time-Between-Failure also implies that software bugs are more likely to happen than Hardware failures over the lifetime of a car.	**All assets**
	Failures or disruptions of communication links	See Disasters and Outages - Network outage	**All assets**
Eavesdropping/ Interception/ Hijacking	Interception of information / Interfering radiations	See physical attacks	**All assets**
	Replay of messages	If internal networks are not sufficiently protected against replay, potential attackers have an easy access to a wide range of dangerous commands, such as steering, braking…	Sensitive data transmitted on **subnetworks**
	Man in the middle/ session hijacking	A large set of interfaces means that, assuming a poor protection of the session, there are many incentives for an attacker to impersonate a distant user: - Impersonating an app store, or service provider, may lead to financial abuse;	**All assets**

		- Impersonating backend systems may help the attacker in downloading a rogue firmware on the vehicle; - Impersonating another vehicle on a V2V session may trigger dangerous behaviours; - Impersonating a legitimate keyfob may lead to theft; - etc. The same notion can also be applied to internal network, for example to perform a MitM on the CAN bus[33].	
	Network reconnaissance and information gathering	Information on car networks can be obtained in many ways (looking for successive MSISDN numbers for OTA updates, looking for vulnerable devices on Shodan, war driving for vulnerable protocols such as ZigBee or Wifi…)	Wireless **External communication networks** or **subnetworks**
	Repudiation of actions	The liability of the driver being possibly engaged in accidents/assurance/professional contexts, there is an incentive to compromise data related to the car usage such as driving habits or localization. This is simply the extension of existing fraud schemes, for example on tachographs.	Data related to **powertrain control**, **Chassis control** or **infotainment control**
Nefarious Activity/ Abuse	Denial of service	The denial of service is not only to be understood as a particular form of network outage. A denial of service may also be triggered on internal network by flooding a CAN bus, or by provoking faults on an ECU via a malicious payload. The potential impact of such an attack depends on the targeted ECU, but may lead to unexpected behaviours from driving systems	**All assets**
	Manipulation of hardware & software, Manipulation of information	Changing the firmware of a component, or otherwise altering its configuration data, is an essential steps of many attacks. The risk is emphasized when there are no measures to protect the authenticity of critical data or components, such as a secure boot. Manipulation of hardware also allows to perform a man-in-the-middle (for example, cutting the CAN bus or isolating a given ECU[34])	**All assets**
	Unauthorised access to information system/network	The type of threat attracting the most the attention of media[35] is the case where a remote attacker can take the control of an ECU (or impersonate an ECU on an internal subnetwork) and take the control of a car by sending driving-related commands (steering, braking…).	**All assets**
	Compromising confidential information	While information leak may be accidental (See *Damage/ Loss (IT Assets) - Information leakage*), there are also incentives for attackers to deliberately compromise private data or sensitive data such as keys	**All assets**

[33] See https://www.blackhat.com/us-16/briefings.html#canspy-a-platform-for-auditing-can-devices
[34] See CANSPY: a Platform for Auditing CAN Devices, Arnaud Lebrun, Jonathan-Christofer Demay
[35] See a recent example : https://www.wired.com/2016/03/thousands-trucks-buses-ambulances-may-open-hackers/

	Identity fraud	The simplest case of identity fraud is the cloning of a keyfob. This may however be completed by other cases, such as fraud, for example if a user wants their car to display another identity when communicating: - with road infrastructures such as toll systems; - with manufacturer backend[36].	**All assets**
	Unauthorised use of administration of devices & systems, Unauthorised use of software, Unauthorised installation of software	A user may try to access unauthorized functions for various reasons: they might want to circumvent DRMs on applications or media, or get an unauthorized access to features (geofencing, tachograph… See *Eavesdropping/ Interception/ Hijacking - Repudiation of actions*), or they might simply want to tune the vehicle for comfort or performance purpose. Outside vehicles, manufacturers may also be confronted to garages using unauthorized or unlicensed professional tools and software. This threat also includes the notion of cloning, for example when an attacker copies the firmware of an existing device, in order to commercialize it without authorization.	**All assets**
	Abuse of authorizations, Abuse of information leakage	A disgruntled employee (backend services, garage) may use their authorizations to perform malicious actions. A slightly different scenario would be for an infotainment application to abuse its authorizations (for example, to mine private data or perform surveillance activities) The impact of such threats is enhanced in cases where the system itself leaks data due to a poor security design.	**All assets**
	Malicious software, Malicious software activity	The integration of infotainment and mobile ecosystems may cause an increase of potential malicious software introduced by the user. Malicious software may provide a first step for attackers in a multi-step attack, to get in driving systems via the infotainment subnetwork. Malicious software may also be a first step to gain access to professional systems (e.g. garages or backend), thus potentially gaining a privileged access on a large set of vehicles. It has to be noted that these ties to the mobile and PC ecosystems also means that attackers may recycle well-known attacks paths (generic linux/android/windows) to eventually affect smart cars[37].	**All assets**
	Remote activity (execution)	All external interfaces may be subject to code injection, which may ultimately result in code execution in case of insufficient component robustness.	**All assets**

[36] See http://www.securityfocus.com/archive/1/538862?_sm_au_=icV3HHS2mMF57J6r
[37] It was notably the main point of http://blog.crysys.hu/2015/10/hacking-cars-in-the-style-of-stuxnet/

Advanced Persistent Threats (APT)	-	Some security researchers[38] consider smart car attacks as similar to Advanced Persistent Threats, or advanced enterprise threats, especially because the attackers have to move "laterally into multiple systems". This risk is also relevant for infrastructures (backend systems, or even V2X infrastructures). Such attacks typically use several types of methods and entry points, therefore can be a mix of every other threat described in this table.	**All assets**

3.3 Threat Modeling

This study chooses not to define any specific threat agents (script kiddies, government agencies…) except when it gives useful information on the attacker motivation. Instead, the study will focus on the notion of attack potential. Attack potential is described in the Common Criteria[39] and further refined in the Common Criteria Evaluation Methodology[40]. In Common Criteria, a product evaluated to achieve a given assurance level is supposed to resist attackers with a predetermined attack potential. During the vulnerability assessment, if evaluators detect a potential vulnerability, they will calculate the attack potential required to exploit such a vulnerability. If the attack is exploitable with a potential lower than what the product is expected to resist, the product will fail the associated evaluation task. This attack potential is typically built upon several measures or estimations:

- Time taken to identify and exploit;
- Specialist technical expertise required;
- Knowledge of the [product] design and operation;
- Window of opportunity;
- IT hardware/software or other equipment required for exploitation.

In practice, this means than an evaluated product *may* be vulnerable to some attacks, but that these attacks require *more expertise (or resources, or motivation) than the targeted resistance can handle*. For example, Common Criteria certificates follow a scale of EAL (Evaluation Assurance Level) where a higher EAL means, through the AVA_VAN assurance requirements, that the product is expected to resist stronger attackers:

- A hardware product at an EAL2 level may resist to script kiddies using simple software exploitation frameworks
- The same product evaluated at EAL4+ may be expected to resist an attack by experts using sophisticated equipment such as lasers or Focused Ion Beams.

When performing a threat assessment prior to a certification, computing an attack potential for a threat will help decide:

- Which certification level may provide assurance that the threat is covered;
- Whether some attack scenarios will be "too strong" to be addressed by the expected certification.

[38] See https://blog.lookout.com/blog/2015/08/07/hacking-a-tesla/
[39] Common Criteria for Information Technology Security Evaluation - Part 3: Security assurance components - September 2012 - Version 3.1 Revision 4
[40] Common Methodology for Information Technology Security Evaluation - Evaluation methodology - September 2012 - Version 3.1 Revision 4

While this study will not try to calculate an accurate attack potential for the attack scenarios hereafter, it aims at giving a hint at the differences of potential required depending on the scenario. This is also intended to be a hint to future certification efforts.

3.4 Sample attacks

The Table 1 hereafter lists a sample of attacks showing how previous threats can be related to existing research and exploitation paths:

Table 1 : Sample attacks

THREATS	ATTACK	LESSONS LEARNED
Network reconnaissance and information gathering, **Unauthorised access to information system/network**	**Remote attack (see section 3.5.1 for more details on how this kind of attack can be performed)** First introduced in 2011[41], remote attacks on cars (via internet) have been widely exposed in the press due to the work of Charlie Miller and Chris Valasek[42]. This type of attack typically included attempts to craft messages on the CAN bus to change the behaviour of the vehicle.	Lack of communication protection (from the point of view of the discovery and the lack of authentication); lack of Identification, authentication and authorization for actions accessible remotely.
Network reconnaissance and information gathering, **Unauthorised access to information system/network**	**Remote attack (see section 3.5.1 for more details on how this kind of attack can be performed)** In a variation of previous attacks, the access gained remotely can be used for other purposes, for example force the geofencing of the vehicle, as exposed a in more recent example of remote attack[43]	Lack of communication protection (from the point of view of the discovery and the lack of authentication); lack of Identification, authentication and authorization for actions accessible remotely.
Malicious software, **Unauthorised installation of software**	**Persistent vehicle alteration (see section 3.5.2 for more details on how this kind of attack can be performed)** Researchers compromised libraries used by garages to control diagnostic tools, in order to allow the installation of malicious firmware on cars	Lack of libraries authentication and lack of integrity checks for external components on diagnostic equipment Use of vulnerable cryptographic functions
Manipulation of hardware, **Man in the middle**, replay of messages	**Persistent vehicle alteration (see section 3.5.2 for more details on how this kind of attack can be performed)** Researchers with a physical access to the vehicle performed a man-in-the-middle by inserting an unauthorized component directly on the CAN bus, then proceeded to drop/alter/replay messages.	Direct CAN access is easier than may manufacturers might think. Lack of protections in the CAN protocol allow to perform a man-in-the-middle, even if timing constraints makes the exploitation non-trivial in practice

[41] See Comprehensive Experimental Analyses of Automotive Attack Surfaces, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno

[42] As an example : https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/, https://www.wired.com/2015/07/patch-chrysler-vehicle-now-wireless-hacking-technique/, https://www.wired.com/2016/08/jeep-hackers-return-high-speed-steering-acceleration-hacks/, and https://www.wired.com/2015/08/uber-hires-hackers-wirelessly-hijacked-jeep/

[43] See for example http://jcarlosnorte.com/security/2016/03/06/hacking-tachographs-from-the-internets.html

Man in the middle, **Inadequate design and planning or lack of adaption**	**Theft (see section 3.5.3 for more details on how this kind of attack can be performed)** Researchers recently presented a correlation-based attack on remote keyless entry systems concerning millions of cars ("most VW Group vehicles manufactured between 1995 and [2016]"[44]). In this case, the researchers claim that the attack could explain theft cases found in the wild. This follows a long history of attacks on keyless entry (including notably the RollJam[45] attack) and start systems[46], all of which relying on cheap hardware and short exploitation time. Attacks exploiting vulnerable cryptography on these systems are not new, with examples as far as 2005[47].	Vulnerable (implementation of) cryptography
Unauthorised use of administration of devices & systems	**Theft (see section 3.5.3 for more details on how this kind of attack can be performed)** Thefts have been shown to use, in the wild, administration equipment to defeat keyless entry and start systems[48]. These equipment were initially intended for locksmiths and car dealers.	Identification, authentication and authorization is needed for access to privileged functions, especially for maintenance equipement.
Information leakage, Abuse of information leakage	**Surveillance (see section 3.5.4 for more details on how this kind of attack can be performed)** Researchers devised an experimental setup to validate their cost analysis estimation of a surveillance attack performed by a mid-range attacker using dedicated hardware. The attack uses ITS communication interfaces[49].	Surveillance is possible in practice for a mid-range attacker; interfaces (e.g. ITS interfaces) lack the pseudonymity measures allowing to mitigate the attack

3.5 Attack scenarios

The threats described previously give a very high-level view of the potential issues facing smart cars. Some examples of attacks scenarios are introduced hereafter to show in more details the variety of attacks that can potentially target smart cars. Additionally, they provide an introduction to the categories of good practices allowing to cover these threats. We consider several categories of attacks, which are described as scenarios hereafter: **Remote attacks, Persistent vehicle alteration, Theft and Surveillance.** These scenarios are detailed in the next sections.

3.5.1 Remote attacks (threatening passengers safety)

[44] See Lock It and Still Lose It—On the (In)Security of Automotive Remote Keyless Entry Systems - Flavio D. Garcia, David Oswald, Timo Kasper, Pierre Pavlidès
[45] See https://www.wired.com/2015/08/hackers-tiny-device-unlocks-cars-opens-garages/
[46] See Relay attacks on passive keyless entry and start systems in modern cars, Aurélien Francillon, Boris Danev, Srdjan Capkun, Department of Computer Science, ETH Zurich
[47] See Security analysis of a cryptographically enabled RFID device, Bono, S. C., Green, M., Stubblefield, A., Juels, A., Rubin, A. D., and Szydlo
[48] See http://fortune.com/2016/08/06/houston-car-hackers/
[49] See Connected Vehicles: Surveillance Threat and Mitigation, Jonathan Petit, Djurre Broekhuis, Michael Feiri, Frank Kargl and slides https://www.blackhat.com/docs/eu-15/materials/eu-15-Petit-Self-Driving-And-Connected-Cars-Fooling-Sensors-And-Tracking-Drivers.pdf

555

Table 2 : Attack scenario 1 - remote attack

ATTACK SCENARIO

TYPE OF ATTACK	DESCRIPTION	ASSET AFFECTED
Remote, via functional interfaces	This attack exploits vulnerabilities in external functional interfaces, related to telematics or infotainment. Connected ECUs may be used in a variety of functional uses, all of which can be an entry point for such attacks[50]. The scenario could typically follow these steps: first **identify a vulnerable car**, then **gain access to internal services (e.g. on a TCU)**, and eventually, from the access gained onto the vehicle, **obtain an access to vehicle systems.**	In a first step, **External communication networks** are targeted. Ultimately, all **ECUs and sensors** may be compromised

CRITICALITY	LIKELIHOOD
High	Unlikely[51]

CASCADING EFFECTS	STAKEHOLDERS INVOLVED
Vehicle (safety) systems disruption may result in an accident, possibly involving other vehicles.	All stakeholders providing, or operating external interfaces (ISPs, manufacturers, Tiers, aftermarket and app providers, cloud service providers). ECU manufacturers also concerned, since this scenario exploits the lack of ECU self-protection.

RECOVERY TIME AND EFFORTS	GOOD PRACTICES
Even if vulnerabilities may be fixed by an OTA update, and even if the vehicle does not seems physically damaged, it is likely than a physical inspection will be needed to ensure that safety is maintained.	✓ General good practices apply (Policy and standards, organizational measures) ✓ In terms of security functions, **Communication protection** is obviously needed to mitigate these attacks, and well as **Identification, authentication and authorization** for all actions accessible remotely. These functions are supported by **Cryptography**, **Security Audit**, and software **self-protection.**

CHALLENGES AND GAPS
Insecure design or development, Safety and security process integration.

556

557 This kind of attack is very much in the public eye[52]. While the attack is not at all trivial to realize, and
558 therefore rated as "unlikely", it can have devastating consequences. The type of attacks described by this
559 scenario could be roughly described as in the Figure 6 hereafter[53]:

560

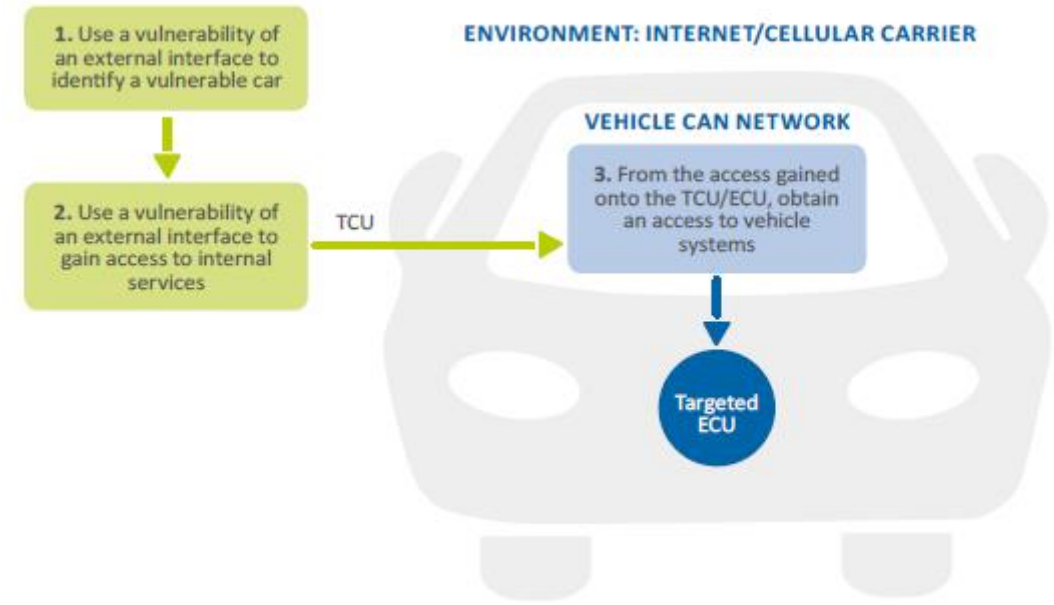
[50] for example insurance, fleet management (trip tracking, diagnosis, surveillance of stolen vehicles,...), "smart driving assistant" (e.g. for fuel efficiency), geo-fencing , remote engine start, crash reporting...

[51] The TVRA method only defines three grades of likelihood (*likely*, *possible* and *unlikely*). *Unlikely* corresponds to vulnerabilities requiring a *high* or *beyond high* attack potential for exploitation. Therefore, it does not mean that the likelihood is *practically* unlikely, but that only motivated and skilled attackers are considered.

[52] in particular due to the work of Charlie Miller and Chris Valasek, but also other studies such as *Fast and Vulnerable: A Story of Telematic Failures* by Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage

[53] These steps are very similar to what is described in previous research, in particular *A Survey of Remote Automotive Attack Surfaces, Chris Valasek and Charlie Miller*, or *Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage*

561 **Figure 6 : Remote attacks threatening passengers' safety**



562

563 Example:

564 • As a first step, the attacker may know that a given vehicle model has a vulnerable SMS link, and knowing
565 its MSIN, enumerates MSINs in hope that all numbers have been sequentially assigned, thus discovering
566 other vulnerable vehicles[54].
567 • The cost to identify such a vulnerability only relatively high, because the attack surface of a smart
568 car is very large: if the direct IP connectivity of a car is well-protected, the attacker can move to
569 another entry point such as SMS.
570 • Interestingly, the use of dedicated components to compromise cellular connection ("stingrays" or
571 "fake BTS") was studied[55] but is dismissed by newer studies describing it as an unnecessary complex
572 entry point compared to other methods, especially internet-based attacks[56].
573 • As a second step, they exploit the lack of authentication for SMS update, and upload a crafted firmware
574 to the TCU[57].
575 • This is an extreme example: the attacker will typically gain an access of some sort on a TCU. The
576 usefulness of this access will however be different it consists in a simple session, a highly privileged
577 session, or the capacity to update a malicious firmware, which has consequences on what is possible
578 as a third and last step.

[54] Alternatively (in the example of Miller/Valasek), the attacker discovers the TCU on Shodan because the carrier supports direct IP

[55] R. Ofir and O. Kapora. A remote attack on an aftermarket telematics service. http://argus-sec.com/blog/remote-attack-aftermarket-telematics-service, Jul. 2014.

[56] Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage

[57] Alternatively (in the example of Miller/Valasek), the attacker may try to directly communicate with ECUs because it contains non-diversified SSH credentials (that may have been extracted by a previous physical attack on another vehicle).

- As a last step, their crafted firmware is able to communicate legitimately on the CAN bus, allowing to communicate with the driving systems
  - The range of consequences may vary from the mildly disruptive (such as horn activation) to life-threatening situations, such as brake disconnect, engine halt or air bag activation.

Exploiting the complete scenario will require several vulnerabilities to be exploited in sequence, and should not be regarded as an easy task. In particular, sending crafted messages on the CAN bus is not a trivial way to trick an ECU into performing a malicious action[58].

3.5.2 Persistent vehicle alteration (by the legitimate user or by the use of diagnostic equipment)

[58] See *A Survey of Remote Automotive Attack Surfaces, Chris Valasek and Charlie Miller*,

588

Table 3 : Attack scenario 2: persistent vehicle alteration

ATTACK SCENARIO

TYPE OF ATTACK	DESCRIPTION	ASSET AFFECTED
Local, via functional or diagnostic interfaces	In the case of an alteration by the legitimate user, the scenario could consist in getting a **direct connection to car components**, then **trying to persistently alter the behaviour of a given ECU.** The objective may be for example vehicle tuning, bypass of the geofencing on a corporate vehicle … The user may also use diagnostic equipment, which may also be used by other categories of attackers, for example in a garage. The steps would then consist in **obtaining a legitimate or illegitimate access to diagnostic equipment**, then **exploiting a vulnerability in the diagnostic equipment to persistently alter the behaviour of an ECU.** In a garage context, such an attack may be related to business intelligence as much as an attack on the vehicle itself	The primary targets are the **OBD II ports** from the **Diagnostic and maintenance systems**. In the case of an alteration by the user, assets related to the access control functions of the **ECUs and sensors**[59] are targeted. In the case of a garage attack, IP and Trade secrets, but also private data stored on **ECUs and sensors**, or transiting on **subnetworks**[60]. In both cases, **Powertrain control** and vehicle safety systems from **Chassis or body control** may be hit indirectly

CRITICALITY	LIKELIHOOD
High	Possible to Unlikely[61]

CASCADING EFFECTS	STAKEHOLDERS INVOLVED
The immediate effect can range to data leak to the disruption of vehicle systems. This may result in an accident, possibly involving other vehicles, while data leak has less critical consequence, but may result in brand damage.	All stakeholders providing ECUs, diagnostic equipment or aftermarket dongles (car manufacturers, Tiers, aftermarket providers). Garages are also concerned, since attacks performed via diagnostic equipment are likely to use garages as an entry point.

RECOVERY TIME AND EFFORTS	GOOD PRACTICES
Even if vulnerabilities may be fixed by an OTA update, and even if the vehicle does not seems physically damaged, it is likely than a physical inspection will be needed to ensure that safety is maintained.	✓ General good practices apply (Policy and standards, organizational measures) ✓ In terms of security functions, **Communication protection** is obviously needed to mitigate these attacks, and well as **Identification, authentication and authorization** for all actions accessible via diagnostic interfaces. Physical **self-protection** also contribute to reduce the attack surface for local attacks. ✓ These functions are supported by **Cryptography**, **Security Audit**, and software **self-protection.**

CHALLENGES AND GAPS
Insecure design or development (especially for the access control to maintenance tools), safety and security process integration

589

590

59 The assets primarily targeted are mostly related to **access control**, especially access to functions not intended for users (fleet management, chronotachograph, geofencing...). Studies give example of privileged services than can be

This category includes for example cases where the legitimate user tries to modify the behaviour of their vehicle, as summarized in Figure 7. This may include:

- Attempts to "tune" the vehicle driving characteristics, for example to enhance performance. The car hacker's handbook, for example, is advertised to users as mean to perform "car mods" or "discover undocumented features"[62];
- Attempt to bypass monitoring services such as geofencing or fleet management. This is, in a way, the extension of existing situations such as tachograph fraud[63] [64].

Other attackers than the legitimate user of the car may also want to alter the behaviour of the vehicle. For example, the attacker may be a garage employee using diagnostic equipment:

- Attacks in garages may be related to business intelligence (aiming at gaining sensible information on competitors technical implementations)
- Attacks inside or outside garages may be related to organized crime. This may for example be used as a threat on the garage or users (on the model of ransomware). As for today, the presence of financial incentives is not yet very frequent (for example payment information accessible in entertainment systems). This example looks very much like existing scenarios targeting point-of-sale terminals, where malware such as memory scrappers can be installed by employees. The situation is however slightly different in garages, since:
  - The turnover in garages is not the same as large shopping centers that heavily rely on temporary work; this creates less opportunity for attackers;
  - The incentive in point of sale is not as strong (attacks on points-of-sale directly allow to obtain payment-enabling data)

compromised because static keys were discovered by a memory dump (for example SSH keys). Other targeted assets are the **driving systems**, especially in cases where the user tries to modify the performance of their vehicle. Vehicle safety systems may also be at risk due to accidental side effects of the attack. Modified traffic on the CAN bus may for example trigger denials of service on the bus, or otherwise cause dangerous situations to arise on vehicle systems
60 **IP and Trade secrets** may be targeted. In a context related to organized crime, the assets are more likely to be **vehicle safety systems**, **driving systems** or **private data** (especially payment data)

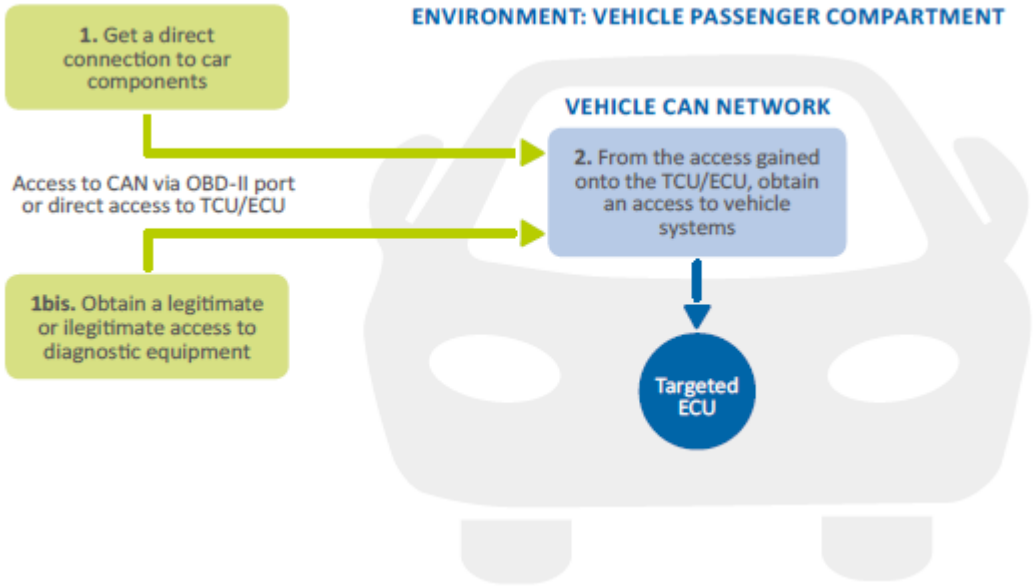
[61] The TVRA method only defines three grades of likelihood (*likely*, *possible* and *unlikely*). *Unlikely* corresponds to vulnerabilities requiring a *high* or *beyond high* attack potential for exploitation. Therefore, it does not mean that the likelihood is *practically* unlikely, but that only motivated and skilled attackers are considered.

[62] See Car Hacker's Handbook by Craig Smith

[63] http://www.euro-controle-route.eu/site/en/info/tacograph/fraud/

[64] See for example, for the example of tachographs: https://www.tispol.org/content/2016/02/02/07/31/technology-used-tachograph-fraud-becoming-more-complex-and-sophisticated-0

619 **Figure 7 : Persistent vehicle alteration scenarios**



620

621 **Main scenario**

622 • Example 1: The attacker connects to the CAN bus, for example by identifying the appropriate pins on
623 the OBD II port. They may then plug a cheap CAN sniffer on these pins (step 1). The step 2 will consist
624 in trying to analyse the CAN traffic and then alter the car behaviour via crafted packets (the car
625 hacker's handbook gives the example of a spoofed speed transmitted to the tachograph[65])
626 • Example 2: As a first step, an attacker directly connects an ECU (using a JTAG port on the board). The
627 step 2 will consist in exploiting the JTAG debug capacities by uploading a crafted firmware[66];
628 • Other examples of attacks may consist in glitching, memory dump, etc [67]…
629

630 **Alternate scenario**

631 • Example 1: As a first step, the attacker may buy a black market diagnostic equipment[68], or reverse
632 engineers a legitimate equipment, or even access a legitimate equipment (rogue garage employee).
633 The second step may then consist in modifying an ECU by injecting a crafted firmware, or simply a
634 previous, vulnerable version of the firmware.
635 • Example 2: Instead of trying to access the diagnostic equipment itself, the attacker may try to
636 compromise the laptop that interfaces with this equipment69. In that case the skills may not be much
637 more than being able to reverse a DLL and exploiting bad digital signature implementations, which are

[65] See Adventures in Automotive Networks and Control Units, Valasek/Miller. The document highlights the fact that analyzing and crafting CAN packets is not an easy task.
[66] See Car Hacker's Handbook by Craig Smith, which reminds that it requires to obtain, and then reverse-engineer a firmware, which is not trivial
[67] See for example Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage
[68] See http://fortune.com/2016/08/06/houston-car-hackers/
[69] It was notably the main point of http://blog.crysys.hu/2015/10/hacking-cars-in-the-style-of-stuxnet/

skills frequently found in “black hat” communities related to DRM, point-of-sale, malware creation... Even in that case, the attacker may need car-specific skills, for example to be able to craft a working firmware. This also assumes that the attacker will perform their attack remotely through a malware, which makes the whole attack more difficult by an order of magnitude.

3.5.3 Theft scenario

Table 4 : Attack scenario 3 - Theft

ATTACK SCENARIO

TYPE OF ATTACK	DESCRIPTION	ASSET AFFECTED
Local	Several possible scenarios, some being more realistic than others: Compromising a local wireless connection (e.g. WiFi), Key fob cloning, Relay attack, Rolling code jam, exploiting the Keyless systems …	**Body control domain and External communication networks** are the primary asset targeted, but ultimately all assets are concerned, in cases chare the vehicle is eventually stolen

CRITICALITY	LIKELIHOOD
Medium	Possible

CASCADING EFFECTS	STAKEHOLDERS INVOLVED
Beyond the theft itself, privacy issues can happen (the same way as the theft of smartphones or tablet may result in private data being accessible by the thief).	Actors providing keyless entry systems (car manufacturers and Tiers), but also Insurance companies, insofar as policies should take into account these scenarios and define appropriate forensic procedures.

RECOVERY TIME AND EFFORTS	GOOD PRACTICES
Assuming that most vulnerabilities are software-based, they may be fixed by an OTA or physical update. Hardware vulnerabilities may cause much higher recovery costs.	✓ General good practices apply (Policy and standards, organizational measures); ✓ In terms of security functions, **Cryptography** is obviously the first coming to mind, since most of these attacks rely on the weak cryptography implemented in remote keyless entry systems; ✓ **Communication protection** is obviously needed to mitigate these attacks, and well as **Identification, authentication and authorization** for all actions accessible, for example, via local wireless entry points (wifi, keyfob…). Physical **self-protection** also contributes to reduce the attack surface for local attacks; ✓ **Security Audit** may help the forensic analysis of such cases, from an insurance point of view, as they are physically undetectable**;** ✓ These functions are supported by **self-protection.** In particular, the design should allow users to fall back to a mechanical lock whenever a vulnerability is found in their keyless entry systems.

CHALLENGES AND GAPS
Insecure design or development

In this scenario we consider the possibility for an attacker to gain physical access to the inside of a vehicle without a legitimate access means.

Standard key fobs and access control devices usually work under the assumption that there is only one level of access, thus gaining access to the inside of a vehicle often entails access to the vehicle main functions : engine start, infotainment unit, trunk opening. While the most plausible risk is the plain and simple theft of

the vehicle or any of the owner's possessions kept inside, such an attack may be a first step towards a more involved attack scenario since access to the inside of the vehicle provides :

- Access to the OBD-II diagnostic port, thus easier access to the CAN/LIN bus;
- Access to the head-up unit physical interfaces (USB, CD/DVD) and assets (navigation data, personal data, access to remote services);
- Easier access to other ECUs (telematics control unit, engine / transmission control unit, gateway).

A complex scenario may involve an attacker that uses one of the means listed above in order to incapacitate (fully or partially) the vehicle operation in such a way that only him/her can restore it remotely, and ask for a ransom. In such a case the main issue is the reproducibility, since such an attack would only be profitable if it has the potential to scale up.

The scenario can use various, very different approaches, such as:

- Compromising a local wireless connection (an proof of concept already exists, where an insecure WiFi connection could be used to ultimately disable the theft alarm[70]
- Key fob cloning: the following techniques may provide this capability:
  - gain access to the key fob secure memory (through reverse engineering or side-channel);
  - compromise the pairing process, for instance by compromising the device used for pairing in the garage;
- use a known vulnerability to get hold of the unique ID from the car's diagnostic port[71].
- Relay attack: this attack has been shown to be effective with PKES (Passive Key Entry and Start) systems[72], where no other action than proximity is required on behalf of the user to open / ignite the car. In such a case it is possible to relay the near-field radio signal over large distances using cheap hardware, from the vehicle to the key fob. This requires the ability to place a radio transceiver near the key fob.
- Rolling code jam: Even if the key cloning scenario is not feasible (by lack of the specific hardware used for pairing the key with the vehicle or reverse engineering / side channel capabilities), it is possible to compromise the rolling code by jamming the radio signal so that the code is not discarded by the vehicle and can be replayed. This attack requires cheap hardware and has been successfully demonstrated for a range of vehicles on the field[73].
- Keyless systems: The smartphone application that controls the opening of the car is compromised, in order to gain illegitimate access to a car. While this mode of access control is far from being common, it may be in a near future[74], which will have the effect of overextending the attack surface with all mobile (applications and OS) vulnerabilities.

Many vulnerabilities used in such attacks are not as technically challenging as in other scenarios, and may use cheap and easy to come by devices[75] that require no high level technical skills for their operation. In

[70] See Hacking the Mitsubishi Outlander PHEV hybrid, Pen Test Partners, https://www.pentestpartners.com/blog/hacking-the-mitsubishi-outlander-phev-hybrid-suv/. This is also part of an ongoing series including https://www.pentestpartners.com/blog/hacking-the-mitsubishi-outlander-working-out-the-protocol/

[71] See http://jalopnik.com/5923802/watch-hackers-steal-a-bmw-in-three-minutes

[72] See Relay attacks on passive keyless entry and start systems in modern cars, Aurélien Francillon, Boris Danev, Srdjan Capkun, Department of Computer Science, ETH Zurich

[73] http://andrewmohawk.com/2016/02/05/bypassing-rolling-code-systems/

[74] http://www.dailydot.com/technology/cars-vulnerable-to-remote-hacking/

[75] Such as the RollJam, for instance: http://thehackernews.com/2015/08/rolljam-unlock-car-garage.html

some cases, though, cryptographic attacks are needed to circumvent the keyless entry protection. Due to hardware limitation, these cryptographic protocols are however weaker than in many other domains, and researchers have shown that attacks can be performed without expensive equipment[76].

3.5.4 Surveillance scenario

Table 5 : Attack scenario 4 - Surveillance

ATTACK SCENARIO

TYPE OF ATTACK	DESCRIPTION	ASSET AFFECTED
Local or remote	There are several different possibilities for surveillance in smart cars. We distinguish between **Targeted Surveillance, Mass surveillance** and **Surveillance on cloud-stored data and services.**	private data stored on **ECUs and sensors**, or in transit through the **subnetworks** or **external communication networks**, notably location-aware content, but also communications or payment data if any

CRITICALITY	LIKELIHOOD
High	Unlikely[77]

CASCADING EFFECTS	STAKEHOLDERS INVOLVED
Cascading effect may include theft of the user identity, for example to perform a financial fraud in a second step. In the case of mass surveillance, consequences are out of scope of this study.	All actors storing or processing private data: car manufacturers, Tiers, aftermarket providers, app providers, cloud service providers, garages…

RECOVERY TIME AND EFFORTS	GOOD PRACTICES
Assuming that most vulnerabilities are software-based, they may be fixed by an OTA or physical update. Hardware vulnerabilities may cause much higher recovery costs.	✓ General good practices apply (Policy and standards, organizational measures). In this case, **privacy regulation**, may notably contribute to reduce the amount of memorized private data in the first place, thus reducing the impact of an attack; ✓ In terms of security functions, **Communication protection** is obviously needed to mitigate these attacks (especially for communication with cloud-based services) as well as **Identification, authentication and authorization**. ✓ These functions are supported by **Cryptography**, **Security Audit**, and software **self-protection.**

CHALLENGES AND GAPS
Insecure design or development (lack of privacy by design in components or protocols), safety and security process integration

[76] See Lock It and Still Lose It—On the (In)Security of Automotive Remote Keyless Entry Systems - Flavio D. Garcia, David Oswald, Timo Kasper, Pierre Pavlidès

[77] The TVRA method only defines three grades of likelihood (*likely*, *possible* and *unlikely*). *Unlikely* corresponds to vulnerabilities requiring a *high* or *beyond high* attack potential for exploitation. Therefore, it does not mean that the likelihood is *practically* unlikely, but that only motivated and skilled attackers are considered.

This scenario gathers considerations regarding the possibilities of surveillance offered by recent cars and vehicles. While there is little public evidence or work for this kind of situations, several potential vulnerabilities and weaknesses have been noticed by proofs of concept by researchers[78].

There are essentially two kinds of plausible surveillance scenarios:

- Targeted surveillance, where a single individual is tracked using a vulnerability in its vehicle systems
- Mass surveillance, where a large number of individuals are tracked through some common vulnerability.

An alternative to both scenarios consist in performing surveillance only on cloud-stored data, instead of focusing on vehicles. This alternative will not be explored in detail here, since ENISA already addressed the issue of cloud security[79].

In the case of targeted surveillance the high investment (in cost and risk) of the attack hints at the following plausible motives: espionage, crime, terrorism, or business intelligence. On the other hand the mass surveillance case involves spying on a large number of vehicles in order to get exploitable data. While the incentives for this are not clear at the moment, and there is not public record of such an exploitation, it is relatively easy to setup such a system by passively sniffing the RF emissions of the vehicles and discriminate between them using unique identifiers.

The associated threat agents may thus be government agencies and criminal organisations, with a high attack potential and strong incentives for targeted surveillance, whereas the scope may be broader for mass surveillance due to the relative easiness of the underlying attacks

Typical attack vectors for targeted surveillance rely on modification of the vehicle software and/or hardware in order to setup the surveillance. Software-based scenarios could typically be found in cases where the attacker has no physical access to the targeted vehicle (therefore is unable to put a physical tracker in the vehicle).

The relevant vulnerable components are then the ECU hardware and software (mainly the infotainment system and navigation unit).

The typical attack vectors for mass surveillance are:

- All wireless emissions: WiFi, Bluetooth and GSM/3G/4G signals can be used to uniquely identify a vehicle. In particular:
  - When the infotainment system provides a WiFi hotspot functionality that broadcasts its SSID;
  - Most TPMS systems broadcast a unique RFID identifier;
  - Using a fake BTS, it is possible to spy on the ICCID of the USIM cards.
  - Car systems can allow fingerprinting[80], quite the same way as browser or device fingerprinting. However, it may be argued that the browser of an infotainment system allows an easier fingerprinting than sensors, which are more difficult to access.

[78] See for example https://www.blackhat.com/docs/eu-15/materials/eu-15-Petit-Self-Driving-And-Connected-Cars-Fooling-Sensors-And-Tracking-Drivers-wp2.pdf

[79] See https://www.enisa.europa.eu/publications#c5=2006&c5=2016&c5=false&c2=publicationDate&reversed=on&b_start=0&c8=Cloud+Computing+Security

[80] See for example Automobile Driver Fingerprinting, Miro Enev, Alex Takakuwa, Karl Koscher, and Tadayoshi Kohno, Proceedings on Privacy Enhancing Technologies ; 2016

- Cloud storages / backed systems, which collect the position of a large set of vehicles. These includes the fleet management systems, localisation-aware services, and navigation systems real-time databases.

Depending on the scenario, the impacts are either financial, or on the privacy personal freedom of the individuals.

It should be noted that surveillance scenarios are facilitated by existing, user-accepted, monitoring features. Several examples come to mind, amongst which:

- The usage of OBD-II dongles to monitor driving habits in exchange of reduced assurance fees[81];
- The accumulation of private information due to the interconnection with social networks.

These user-accepted usages come with entry points, some of them privileged (for example OBD-II dongles), which can be compromised by an attacker. Therefore, reducing the chance of privacy attacks could also benefit from limiting the user-accepted surveillance solutions. European privacy regulation already contributes to limit potential accumulation of private data and abuses of opt-out scenarios[82] - however, they do not address the security risks caused by the introduction of technical components dedicated to user-data collection.

[81] See for example http://www.computerworld.com/article/2684298/once-your-cars-connected-to-the-internet-who-guards-your-privacy.html

[82] Such as the OnStar privacy issues described in http://www.autoblog.com/2011/09/26/gm-onstar-privacy/

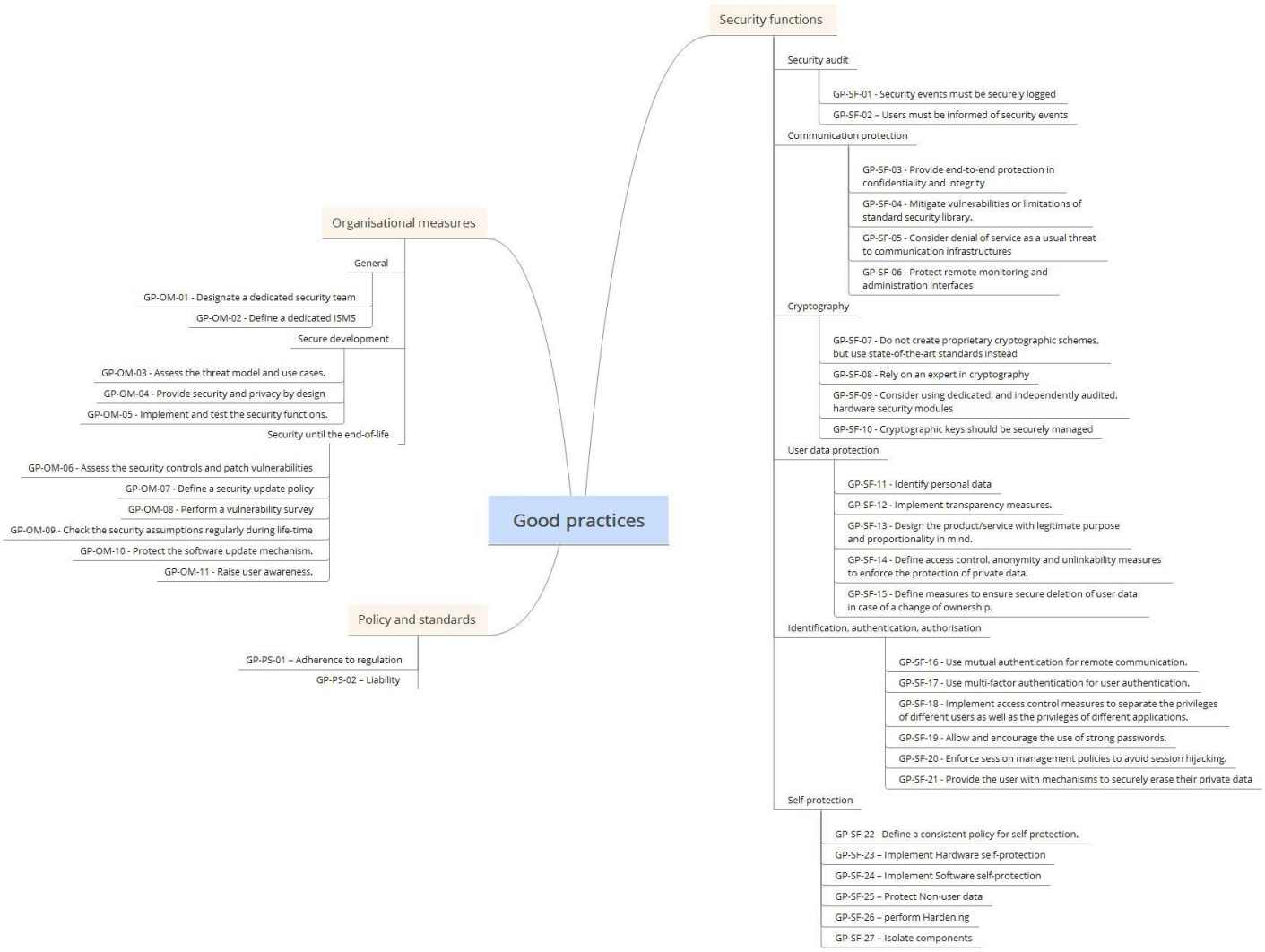
4. Key findings

4.1 Good practices

The Figure 8 hereafter summarizes the good practices identified in this report. The good practices are described in the remainder of this document, and further explained in Appendix B. They are categorized as

- Policy and standards
- Organizational measures
- Security Functions

Figure 8: Summary of good practices



4.1.1 Policy and standards

CATEGORY	GOOD PRACTICES	ASSOCIATED ATTACKS
Policy and standards	**GP-PS-01 – Adherence to regulation.** Industry actors shall, as a first step, adhere to regulation related to security and privacy.	All
	GP-PS-02 – Liability : Car manufacturers should be held liable for damages due to other actors under their control, notably Tiers and garages	All

4.1.2 Organizational measures

CATEGORY	GOOD PRACTICES	ASSOCIATED ATTACKS
Organizational measures – general	**GP-OM-01 - Designate a dedicated security team.** Actors of the smart car industry should rely on specialists, notably for **secure design, penetration testing and risk management**. Expert advice for training and corporate security is also recommended.	All
	GP-OM-02 - Define a dedicated Information Security Management System (ISMS) – Actors of the smart car industry should define an ISMS, possibly inspired from SAE J3061, ISO 27001 or NIST 800-53, and refine it to address the specific needs of their industry, notably the **management of Tier-1 and Tier-2 actors**, and processes to ensure continuous isolation of the components from aftermarket products	All
Organizational measures – secure development	**GP-OM-03 - Assess the threat model and use cases. –** Actors of the smart car industry should perform a threat analysis prior to development possibly inspired **from SAE-J3061 TARA approach** (including EVITA, TVRA, OCTAVE and HEAVENS methods)	All
	GP-OM-04 - Provide security and privacy by design – Actors of the smart car industry should plan their development lifecycles to ensure that security and privacy are taken into account **no later than the design phase**, in order to address the threats identified in the risk assessment.	All
	GP-OM-05 - Implement and test the security functions. Actors of the smart car industry should clearly define appropriate security functions that will be explicitly **implemented and tested** during the development lifecycle. Security functions described in the next section, include Security Audit, Communication protection, Cryptography, User data protection, Identification, authentication, authorization, and Self-protection.	All
Organizational measures – security until the end-of-life	**GP-OM-06 - Assess the security controls and patch vulnerabilities** Actors of the smart car industry should define appropriate assessment procedures to regularly check the effectiveness of their security functions, and patch them whenever needed.	All
	GP-OM-07 - Define a security update policy - Actors of the smart car industry should define an update policy for security patches, taking into **account appropriate timing, conditions, and user awareness for the updates** (to ensure safety during the update), and OTA update mechanisms whenever possible. Manufacturers may have to define whether a vulnerable component can, or should, be put offline when proven vulnerable.	All

CATEGORY	GOOD PRACTICES	ASSOCIATED ATTACKS
	GP-OM-08 - Perform a vulnerability survey - Actors of the smart car industry should perform a vulnerability survey to be proactively able to fix security issues before they can be used in the wild. The vulnerability survey should include developer findings, on-line researches, CERTs advisories, information shared by groups such as CarSec or ISACs, as well as input from customers and security researches. Eventually, vulnerabilities impacting user data should be communicated as transparently as possible, as expressed by the EU Opinion 03/2014 on Personal Data Breach Notification from the Article 29 Working Party	All
	GP-OM-09 - Check the security assumptions regularly during life-time. The devices and services made assumptions to ensure that the security requirements are sufficient (limitations in the usage of the vehicle[83], assumed properties of the environment[84], assumed properties of cryptographic properties [85]…). Vendors and users should be encouraged to check regularly that these assumptions are still valid	All
	GP-OM-10 - Protect the software update mechanism. Vendors should protect the updates (typically via encryption and digital signature) and protect the *application of an update* on the device. Eventually, the update server and infrastructure (including diagnostic tools) should also be protected.	All
	GP-OM-11 - Raise user awareness. Vendors should explain users what actions can contribute to mitigate potential threats, especially how to securely use interfaced systems such as a smartphone.	All

4.1.3 Security functions

This section is structured following the lifecycle of smart cars. Steps are inspired by previous work from NHSTA/NIST[86]

CATEGORY	GOOD PRACTICES	ASSOCIATED ATTACKS
Security functions – Security audit	**GP-SF-01 - Security events must be securely logged** - access to the logs must be documented and protected from disclosure to unauthorized users. The audit trail must be protected from unauthorized access	All
	GP-SF-02 – Users must be informed of security events. HW and embedded systems should provide clear error data that can be leveraged upon by the SW vendors. The user must be informed in case of security errors, updates or compromised data in a device or service they use.	All
Security functions – Communications protection	**GP-SF-03 - Provide end-to-end protection in confidentiality and integrity** using protocols that resist to replay attacks. Favour methods providing forward secrecy whenever possible, for WAN traffic (internet, mobile network) as well as local networks.	Remote attacks, theft, surveillance

[83] For example, users may be advised to remove connectivity features from their entertainment system until a fix has been found for a given vulnerability
[84] For example, vendors should perform a survey to be able to remove a compromised CA from the certificate store.
[85] For example, vendors should check regularly this assumption, for example if a new cryptographic attack puts users at risk unless they use longer keys or change their cryptographic suites.
[86] See National Institute of Standards and Technology cyber security risk management framework applied to modern vehicles

CATEGORY	GOOD PRACTICES	ASSOCIATED ATTACKS
	GP-SF-04 - Mitigate vulnerabilities or limitations of standard security library. Developers must be aware of the vulnerabilities and limitations of the third-party components they use. They should mitigate them whenever possible by patching and by securing the configuration of the communication stacks, which might typically include Bluetooth, Wi-Fi, TLS…	All
	GP-SF-05 - Consider denial of service as a usual threat to communication infrastructures. Vendors and service providers are encouraged to read the ENISA Internet Infrastructure Threat Landscape (for network components) [87].	Remote attacks
	GP-SF-06 - Protect remote monitoring and administration interfaces. Vendors should protect all monitoring and administration interfaces by mutual authentication. And access control mechanisms.	Remote attacks, theft
Security functions - cryptography	**GP-SF-07 - Do not create proprietary cryptographic schemes, but use state-of-the-art standards instead**. If needed, consider getting advice from security experts or your national cybersecurity agency. If no recommendations exist for vendors at a national level, ENISA recommendations should be considered as a reference[88]. This applies also to random number generation, which is a critical part of the cryptographic support, which should meet quality measures on statistical output (for example based upon national requirements[89]). Additionally, consider the expected life duration of the vehicle and find advice on the relevant key size (national recommendations might, in some cases, be based on shorter lifespans than a consumer car)	All
	GP-SF-08 - Rely on an expert in cryptography, notably for interfacing with HW accelerated cryptography or secure elements, or even using or configuring a standard implementation. At least, a third-party code review should be performed to ensure that HW or a standard implementation of cryptography is properly used.	All
	GP-SF-09 - Consider using dedicated, and independently audited, hardware security modules. The standard for independent assessment of security HW should be either FIPS 140-2, or a Common Criteria certification following relevant Protection Profiles. If needed, consider getting advice from security experts or your national cybersecurity agency.	Persistent vehicle alteration
	GP-SF-10 - Cryptographic keys should be securely managed, which means securely generated, distributed (or provisioned), used, stored, and deleted (including revocation). Manufacturers, as well as Tier-1/Tier-2 and aftermarket vendors should consider very carefully the revocation mechanisms associated with their components, especially for OTA provisioning or key management. If needed, consider getting advice from security experts or your national cybersecurity agency.	All

[87] See https://www.enisa.europa.eu/activities/risk-management/evolving-threat-environment/enisa-thematic-landscapes/threat-landscape-of-the-internet-infrastructure
[88] See https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-size-and-parameters-report-2014
[89] See for example A proposal for: Functionality classes for random number generators, Version 2.0 , 18 September 2011, by the BSI, and http:/csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf

CATEGORY	GOOD PRACTICES	ASSOCIATED ATTACKS
Security functions – User data protection	**GP-SF-11 - Identify personal data.** Vendors should identify all data *relating to an identified or identifiable* person. In the case of smart cars, this may especially include location-based data. Consider getting advice from your national data protection agency.	Surveillance
	GP-SF-12 - Implement transparency measures. The interactions with the user (which should not be limited to the *Terms and conditions*) enable to cover the legal transparency requirements.	Surveillance
	GP-SF-13 - Design the product/service with legitimate purpose and proportionality in mind. The actors must ensure that themselves *and their subcontractors or suppliers* do not process user data more than needed, and do not pursue an illegitimate purpose with regard to user data. As a general rule, third party components integrated in the device or third party cloud services should not access user data that have not been anonymized or pseudonymized unless user agreement has been obtained.	Surveillance
	GP-SF-14 - Define access control, anonymity and unlinkability measures to enforce the protection of private data. These measures are typically access control measures, pseudonymity and unlinkability measures (such as ensuring that data is not correlated), and eventually anonymity measures. Anonymity measures may be “one-way” or “non-reversible” (such as truncation or a hash functions) or “reversible” such as encryption.	Surveillance
	GP-SF-15 - Define measures to ensure secure deletion of user data in case of a change of ownership. More generally, a secure factory-reset of the firmware and configuration should be available on the vehicle.	Surveillance
Security functions - Identification, authentication, authorization	**GP-SF-16 - Use mutual authentication for remote communication.** Devices or users connecting to a server must be able to authenticate the server. Reciprocally, servers must be able to authenticate clients and users.	Remote attacks
	GP-SF-17 - Use multi-factor authentication for user authentication. Users should be authenticated by 2-factor authentication whenever possible, including for authentication to cloud services or mobile interfaces, as well as local administration sessions of devices.	Remote attacks, persistent vehicle alteration, theft
	GP-SF-18 - Implement access control measures to separate the privileges of different users as well as the privileges of different applications. In practice, privileged operations should not be readily accessible to normal users. Implementing privilege levels, rings or domains can also be extended to application separation.	Remote attacks, persistent vehicle alteration, theft
	GP-SF-19 - Allow and encourage the use of strong passwords. This concerns all possible uses of passwords: direct device interfaces such as JTAG, but also web, mobile or cloud interfaces. However, the use of passwords in general may cause safety issues for user interactions in a moving vehicle; this good practice is recommended mainly for setup and pairing activities, and especially for administration or diagnostic features.	Remote attacks, persistent vehicle alteration, theft
	GP-SF-20 - Enforce session management policies to avoid session hijacking.	Remote attacks, persistent vehicle alteration, theft
	GP-SF-21 - Provide the user with mechanisms to securely erase their private data For client information in remote infrastructures such as cloud services, data sanitization must be in place. For user data present on vehicles, secure deletion of	Surveillance

CATEGORY	GOOD PRACTICES	ASSOCIATED ATTACKS
	encryption keys may provide enough protection, assuming that data is encrypted in conditions that guarantee long-term confidentiality.	
Security functions – self-protection	**GP-SF-22 - Define a consistent policy for self-protection.** Vendors should challenge every security function of their design, consider how they could be bypassed or weakened, and eventually implement self-protection measures.	Persistent vehicle alteration, theft
	GP-SF-23 – Implement Hardware self-protection: Vendors should define measures to protect hardware against physical attacks or observation. This includes tamper evidence or tamper resistance, and secure design measures.	Persistent vehicle alteration, theft
	GP-SF-24 – Implement Software self-protection: Vendors should define measures to protect existing security functions, typically by validating inputs and outputs, or by separating the capacities of the different software components (levels of trust, virtualization…)	Remote attacks, persistent vehicle alteration, theft
	GP-SF-25 – Protect Non-user data: Vendors should protect data enforcing the security functions, such as keys or configuration data	Remote attacks, persistent vehicle alteration, theft
	GP-SF-26 – Perform Hardening: Vendors should actively reduce the attack surface of the product or device. This includes removing or disabling unused services or interfaces (especially debug interfaces), providing secure configuration by default, as well as integrating malware protection. Some actors may consider intrusion detection systems for internal subnetworks (for example CAN bus monitoring), although this study will not conclude on the merits of these solutions.	Remote attacks, persistent vehicle alteration, theft
	GP-SF-27 – Isolate components: Vendors should reduce the capacity for attackers to jump from a component to another, either by a physical disconnection or by using gateways	Remote attacks, persistent vehicle alteration, theft

4.2 Gaps and challenges

4.2.1 Insecure design or development

Insecure development in today's cars

While the automotive industry has a long-standing expertise in car safety, security issues of connected systems in cars and their potential impact on car safety are not yet properly taken into account, except for few of them[90]. Some studies tried to define a shortlist of the more frequent security issues found amongst manufacturers[91] [92] [93] [94] [95]. After having double checked these shortlists during our own interviews, the following issues seem indeed significant

- No defence in depth strategy during the design of the system (such as a secure boot process, isolation of a Trusted Computing Base, limitation of the number of open ports, self-protection, …);
- No security- or privacy-by-design. For example, telematics schemes may require the car maker to send most of the information exchanged on the CAN bus to a third-party, such as vehicle speed, throttle position, coolant and oil temperature, engine revision status, etc. More information than really needed may be exported outside of the car. While some actors are aware that private data should not be exported without a reason, the same line of reasoning is not always applied to sensitive data;
- Lack of communication protection, on internal as well as external interfaces;
- Lack of authentication and authorization, especially for privileged access to ECUs; for example:
  - no validation or signing of firmware updates,
  - updates happen without server authentication, and even on an arbitrary server,
  - no secure boot,
  - no cellular authentication, or weak authentication mechanisms, or failure to use components that provide authentication functions…;
- Lack of hardening, for example:
  - No data execution prevention or attack mitigation technologies are used on firmware,
  - Public vulnerabilities (DNS proxy, http service...) are left unfixed,
  - ECU services are exposed through different entry points, and even unnecessary communication ports are left open; services such as telnet, web or SSH are sometimes bound to all network interfaces,
  - Weak passwords policies,
  - Misconfiguration (e.g. VPN)
- Lack of diagnosis / response capabilities

[90] https://blog.lookout.com/blog/2015/08/07/hacking-a-tesla/
[91] Tracking & Hacking: Security & Privacy Gaps Put American Drivers at Risk, Ed Markey
[92] Progressive insurance dongle totally insecure, http://www.forbes.com/sites/thomasbrewster/2015/01/15/researcher-says-progressive-insurance-dongle-totally-insecure/
[93] Experimental Security Analysis of a Modern Automobile, Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno
[94] For example in Hacking a Tesla https://blog.lookout.com/blog/2015/08/07/hacking-a-tesla/
[95] Fast and Vulnerable: A Story of Telematic Failures Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage

Security culture

Several sources highlight that actors of the smart car ecosystem come from different domains, leading to different approaches to security, for example that actors having a “deep software experience” are more likely to welcome features such as OTA updates, collaboration with "white hats" or the implementation of bug bounty programs[96].

As already stated, a transparent dialog with security researchers is needed to ensure that the whole community is in a “responsible” disclosure process. The current situation in automotive is very far from this situation, as

- Some findings have been left unpublished due to legal actions between manufacturers and researchers[97], leaving exploitable vulnerabilities in the wild during as long as two years;
- Other researchers have turned to media due to manufacturer’s lack of response[98], thus publishing vulnerabilities for which no fix is planned;
- Some manufacturers do not perform frequent software updates, thus exposing automotive devices to known vulnerabilities (for instance in software frameworks, such as a SSL library or browser library). Such update, even if released in due time by manufacturers, are still seldom deployed Over The Air and may require the car owner to use a USB stick for installing the update or to go a car dealership garage;
- As confirmed by interviews, security functions such as security logs [99] are not regarded as important, while they are essential to security diagnostic in the field.

4.2.2 Liability

Studies show that most users are concerned with cybersecurity issues arising from the integration of connected features in cars. In case a security event happens, they are also likely to blame in equal parts the different actors of the ecosystem such as app stores, app developers and app manufacturers, to take the example of a vulnerability arising from a connected smartphone[100]. Furthermore, there is no chance to enforce a perfect isolation between driving, debug and infotainment (or connected) systems, which means that vulnerabilities from any actor, including aftermarket components, may allow compromising safety-related features of a vehicle. In this context, there is a need to clarify the liability of each actor in case of a security event.

4.2.3 Safety and security process integration

Development processes in place in the car industry take safety issues into account. Despite initiatives to include security in these processes, there is still a **lack of a common standard allowing a complete integration of safety and security in the car development lifecycles**.

[96] For example Hacking a Tesla, https://blog.lookout.com/blog/2015/08/07/hacking-a-tesla/
[97] https://www.theguardian.com/technology/2015/aug/18/security-flaw-100-car-models-exposed-scientists-volkswagen-suppressed-paper
[98] See Hacking the Mitsubishi Outlander PHEV hybrid - https://www.pentestpartners.com/blog/hacking-the-mitsubishi-outlander-phev-hybrid-suv/
[99] Or Security Audit, in the Common Criteria parlance
[100] See for example *Responsibility for Vehicle Security and Driver Privacy in the Age of the Connected Car*, IDC/Veracode, February 2016, IDC #EMEA41026016

The **lack of shared technical standards** for car security is an additional burden for those who try to build secure development processes. Eventually, the complexities of this heavily-tiered ecosystem cause issues in **the supply chain and in the glue code** between components.

Existing initiatives and limitations

The approach in SAE-J 3061[101] tried to address one of the smart cars specifics, which is *a security product that has strong safety requirements and an existing engineering process dedicated to safety*.

It also tried to distinguish between system level and vehicle level issues to define a development method for vehicles, which would both take security into account, and be compatible with the existing lifecycles of the industry. As such, the document is well adapted to smart cars, but still lacks recommendations to address many specifics of this domain. For example, the SAE-J 3061 does not suggest specific remediation to:

- The unusually large attack surface (large number of entry points and variety of attack methods) of smart vehicles[102];
- The combination of easy access for attackers (being a mass-market product) and severe impact (safety consequences on the user and other vehicles)[103];
- The persistence of threats, associated with the relatively long life of the products[104];
- The fact that smart features are not essential to the core features of the car[105].

Several initiatives led to defining guidelines or rules to implement security in the automotive industry (see figure hereafter), and other initiatives[106,107] asked for collaboration on the security topics from the automotive industry. None of them can be considered a standard yet, and the overall standard landscape has yet to achieve the level of completeness and consistency found in domains such as aircraft safety or smartcard security. Figure 9 hereafter gives a sample of existing initiatives, and a sample of initiatives of interest outside of the automotive domain.

[101] See SAE-J 3061 - SURFACE VEHICLE RECOMMENDED PRACTICE - Cybersecurity Guidebook for Cyber-Physical Vehicle Systems

[102] Amongst possible lifecycle adaptation, one may think of the following :
- Adding a dedicated interface design review;
- Adding a dedicated hardening phase during the late integration phases.

[103] This combination implies that smart car security should require a high security assurance. And yet, the SAE-J3061 does not explicitly suggest high assurance certification (for example, Common Criteria EAL4+ security hardware)

[104] This situation should theoretically require smart car manufacturers to reach a consensus on future-proof cryptographic key sizes, which may exceed the usual recommendations of national cyber-security agencies.

[105] As such, a consensus could be reached amongst manufacturers to define an “offline mode” where cars would be functional while deactivating most of the external interfaces, such as the infotainment. Such a mode could be an option when sever flaws have been found and are not yet patched.

[106] https://www.iamthecavalry.org/domains/automotive/5star/

[107] https://www-ssl.intel.com/content/www/us/en/automotive/automotive-security-review-board.html

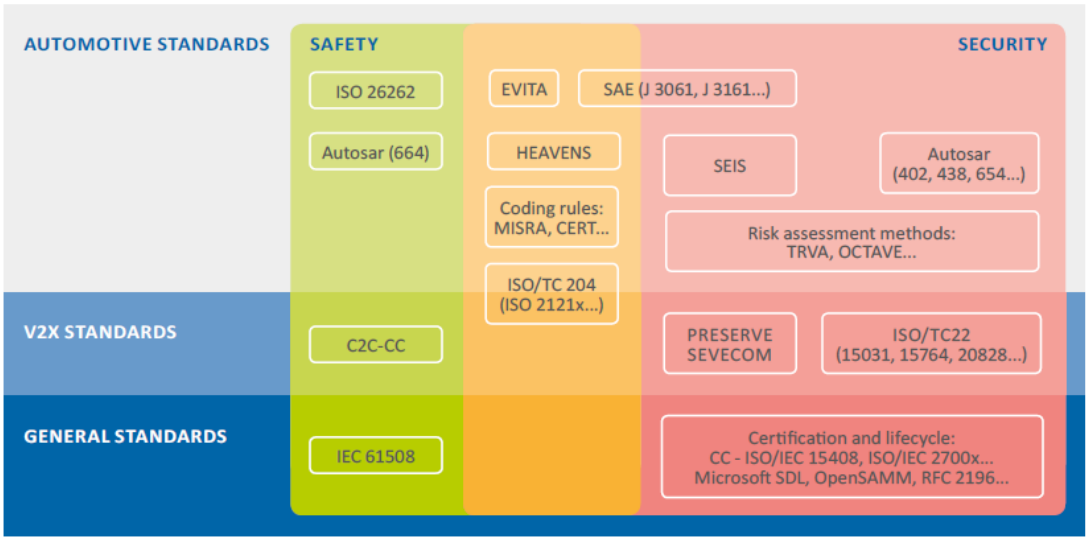


Figure 9 : safety and security standards

At the moment, no certification framework is yet considered a standard for security evaluation or security testing, which would allow detection of vulnerabilities before the product is released. While certification frameworks exist for safety features, for example automatic brake system, most industry actors are still new to the concepts and methods of security certification (for example, the notion of penetration testing).

Other industries (for example airborne systems) eventually defined their own frameworks, for example when facing heavily-tiered environment rendering usual certification standards impractical. Before trying the same approach on automotive products, one should be careful to assess whether these attempts have been successful in practice.

The particular issue of technical standards

The lack of standard ultimately causes additional security issues: for example, several key components of vehicles are still developed with proprietary technologies (the main example coming to mind is the protocols used for CAN communication). This situation makes more difficult to third-party vendors to develop security solutions (for example firewalls or intrusion detection) that could be applied to a large market, hence reducing effectively the cost of security for manufacturers.

Additional issues: supply chain and glue code

Moreover, the heavily-tiered ecosystem of car manufacturing also leads to security integration issues[108]. Eventually, aftermarket products may share the same buses, which also lead to a significant risk[109]. Units such as TCUs can be provided by manufacturers, Tiers or aftermarket providers. Theoretically, all are equally secure or vulnerable, but the ECUs from Tiers or aftermarket providers are more significant from a remediation point of view.

[108] See *Comprehensive Experimental Analyses of Automotive Attack Surfaces*, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno"

[109] See *Experimental Security Analysis of a Modern Automobile* , Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno

As stated by researchers[110], security issues in aftermarket products cannot, by definition, be controlled by manufacturers. In practice, aftermarket vendors are described as fully supportive, but the complexity of the supply chain relationships leads to non-deployed security patches in practice, even when vendors have distributed them (similar issues can be found in smartphones, where security patches on the Android OS are not necessarily cascaded in operators or vendors fine-tuned versions of the OS).

Other studies highlighted the issues caused by integration of SW and HW in the manufacturing, especially the fact that some actors experience with safety issues may cause them to separate software and hardware issues and miss global security vulnerabilities. More generally, the outsourcing model leads to glue code and security flaws due to bad understanding of the security assumptions of third-party code[111]. While acknowledging the effort made by the industry to integrate safety and security approaches, explicit synchronization points should be defined between these activities[112] and between actors of the supply chain.

In the field, this heavily-tiered environment causes additional issues. Security patches need to be validated on the whole supply chain before they can be deployed, which leads to non-deployed security patches in practice, even when the Tier-2 vendor, for example, has developed and distributed them[113] (this issue is, in a way, similar to the issues of a mobile OS security patch not redeployed by OEMs).

4.3 Constraints and incentives

4.3.1 Incentives

Studies consider that the IoT integration into cars cause a leadership crises amongst traditional manufacturers, that are now challenged by actors coming from the software domain[114]. Companies from different domains have different ways to deal with security issues, from disclosure to remediation, which in turn has consequences on the amount of “brand damage” resulting from inevitable cybersecurity issues. In this context, the deployment or non-deployment of cybersecurity measures may have far-reaching consequences.

Other studies point the general perception, by many industry actors, that there is no direct return-on-investment for security[115], which can be attributed to the difficulty to assess the financial impact of hypothetical security flaws.

[110] See Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage
[111] See *Comprehensive Experimental Analyses of Automotive Attack Surfaces*, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno"
[112] See SAE-J 3061 - SURFACE VEHICLE RECOMMENDED PRACTICE - Cybersecurity Guidebook for Cyber-Physical Vehicle Systems
[113] See for example Fast and Vulnerable: A Story of Telematic Failures - Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage
[114] Responsibility for Vehicle Security and Driver Privacy in the Age of the Connected Car, IDC/Veracode, February 2016, IDC #EMEA41026016
[115] See A Summary of Cybersecurity Best Practices, NHSTA

This tends to confirm a widely-accepted consensus that media attention, and more largely, good or bad publicity[116], due to security issues is a main driver to implementing security for industry actors. This consensus was confirmed by the interviews performed during this study, whose results are highlighted in Table 6.

Table 6 : Motivators and incentives, as selected by interviewees (most critical in bold)

CATEGORY	MOTIVATORS/INCENTIVES
Business incentives	**Enabling business opportunities**
	Protecting an organisation's reputation
	Improving efficiencies/reducing-costs
	Protecting intellectual property
Customer incentives	**Protecting users' personal freedom and privacy**
	Protecting physical integrity of customers / users
	Protecting users' confidential information (such as payment data)
	Maintaining data integrity
	Protecting the physical integrity of users' cars, or deter theft
Regulation and infrastructure	**Complying with regulation/legal requirements**
	Protecting the overall transport infrastructure, ensuring continuity of service in a disaster situation

4.3.2 Constraints

Constraints due to the use cases

Some studies point that connected car uses cases, themselves, are inherently insecure. For example, the use of "smart dongles" is often described as a "bad practice" by construction: structural vulnerabilities of the CAN bus have a very deep impact (MiTM, capacity to reflash ECUs, leading to possible actions on brakes, throttle...). The user is only protected by the need for a physical access to the CAN (typically via OBD-II). In this context, a "smart dongle" provides an attacker with the capacity of easily performing a remote attack with the same high impact[117]. Additionally, use cases lead the acceptable cost for some car components. For example, keyless entry systems have an acceptable cost, which implies that they will eventually lack the hardware resources to support state-of-the-art cryptography.

[116] See for example "Automotive Cyber Security: An IET/KTN Thought Leadership Review of risk perspectives for connected vehicles", the IET

[117] See for example Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage

Constraints due to the architecture

Additionally, vehicle systems have very specific issues due to their architecture. In particular, the use of CAN bus (as opposed to Internet-like protocols) would cause:

- More vulnerability to DoS, since arbitration is priority-based [118];
- More issues with network segregation (priority being implicitly derived from safety notions instead of including security properties);
- More vulnerability to reverse engineering due to the small range of valid CAN packets (meaning that few work is needed and a simple fuzzing campaign can have a dramatic impact by itself)[119].

Moreover, in-vehicle systems include a very large number of embedded and *interconnected* components (a typical car contains more than 100 ECUs). Previous studies tend to argue that usual hardening and network isolation issues are insufficient to protect such interconnected systems[120].

It also opens many entry points[121] for an attacker: vulnerabilities in these ECUs may be accessed remotely through[122] multiple possible interfaces, and in some cases including a web browser[123].

Aside of the remote interfaces, various local entry points and diagnostic/test interfaces exist, such as OBD or USB ports, which can also be used to get access to the system, or at least understand how it is designed and which messages are exchanged. Indeed, the legacy bus system (CAN, LIN) offers no protection of the messages. Besides, there is no standard for protection of ECUs (authentication, firmware update), which is left at manufacturers good will. Eventually, many entry points are physically accessible:

- Proprietary connectors[124]: a proprietary implementation does not prevent the tester to find out that it is an Ethernet interface, and to be able to communicate with it,
- Reverse engineering of the firmware (in this example, allows to learn the password rotation scheme and location of the new password in plaintext on the file system)

[118] Experimental Security Analysis of a Modern Automobile, Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno
[119] See *Experimental Security Analysis of a Modern Automobile* , Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno
[120] See *Comprehensive Experimental Analyses of Automotive Attack Surfaces*, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno"
[121] http://www.autosec.org/pubs/cars-usenixsec2011.pdf
[122] http://www.ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf
[123] For example "An unknown 4-pin connector" in Hacking a Tesla
https://blog.lookout.com/blog/2015/08/07/hacking-a-tesla/
[124] For example "An unknown 4-pin connector" in Hacking a Tesla
https://blog.lookout.com/blog/2015/08/07/hacking-a-tesla/

5. Recommendations

Our recommendations aim at enhancing trust within the actors of the ecosystem (car manufacturers, tiers and aftermarket vendors), as well as the trust from citizens in the smart cars available on the market.

5.1 Improve cyber security in smart cars

Recommendation intended for: smart car manufacturers, tiers and aftermarket vendors

The first recommendation of this report is the most obvious one: many vulnerabilities have been found in the last years in the automotive domain. The ever increasing number of recalls due to security issues should be taken into account by industry players. This report gives a possible starting point for the establishment of good practices, and we expect that industry actors will challenge these good practices and **effectively enhance the security of their products**.

By providing a first set of good practices in this report, we hope that the industry should be able to overcome the challenge of *Insecure design or development* identified in section 4.2.1.

5.2 Improve information sharing amongst industry actors

Recommendation intended for: smart car manufacturers, tiers and aftermarket vendors

Information sharing is essential for several reasons. First, transparency between stakeholders is essential to build trust – while some aspects of security implementation generally remain confidential, heavily-tiered environments need to rely on commonly accepted practices to improve integration and avoid “glue code” security flaws. Additionally, transparency contributes to the acceptance of standards, which lower the cost of security implementation by suppliers.

Moreover, information sharing helps industry actors challenge the relevance of their security mechanisms according to field information. Therefore, stakeholders should share and discuss new attack methods found in the wild, in order to help the whole community find countermeasures. Therefore, information sharing will also contribute to overcome the challenge of *Insecure design or development* identified in section 4.2.1.

Eventually, information sharing structures are an efficient way to challenge the skills of security teams by common sessions with other industry players, laboratories, or national agencies.

Communities for information sharing already exist, such as ISACs, or the CarSEC group built by ENISA. This report recommends pursuing this effort.

Last but not least, dedicated CERTs may also play a major role in the detection and remediation of security issues.

5.3 Improve exchanges with security researchers and third parties

Recommendation intended for: smart car manufacturers, tiers and aftermarket vendors

Industry actors should enhance their contacts with third parties, especially from the security domain. The interviews performed during this study showed that:

- Many security companies and academics are interested in the car challenges,

- Many car manufacturers or tiers are reluctant to discuss security matters with third parties, including ENISA.

This situation may hamper security in many ways. First, security researchers might focus on other domains due to the lack of demand, thus leaving the industry without valuable skills and inputs. On top of this, the lack of interaction with researchers might progressively cause the industry to rely on an obsolete vision of security. Therefore, information sharing will also contribute to overcome the challenge of *Insecure design or development* identified in section 4.2.1.

Interactions with third-parties can take multiple forms. The industry should try several approaches and find whichever suits their processes, such as:

- Bug bounties;
- Hosting conferences or hacking contests;
- Creating workgroups with national agencies;

5.4 Clarify liability among industry actors

Recommendation intended for: smart car manufacturers, tiers, aftermarket vendors, insurance companies

This report identified a particular challenge related to liability (see section 4.2.2). In order to address this challenge, industry actors should define processes to clarify their respective liability in case security issues arise. It means, in practice, that they should define:

- Technical processes and criteria to allow a clear detection of the liability for a given security issue;
- Commercial, insurance or legal means to enforce the liability.

Criteria and processes

There are many ways to define criteria and processes to pinpoint liability in cases of security issues. We give hereafter an example of such a process:

- The HW vendor could be rendered "liable" by a certification of the hardware. The HW vendor would be considered liable for any issues occurring in the HW, *provided the OS or runtime environment complies with the HW security guidance*;
- The vendor of the OS or runtime environment could be rendered "liable" by a certification of a composite product (consisting of the runtime environment *and* a given security hardware). The vendor would then be considered liable for any issues occurring in the OS or runtime environment, *provided the applications comply with a specific set of rules*[125]. The notable point here is that the rules are meant to allow an automated verification, typically by code analysis. Such analysis could, for example, be performed when an application is submitted to an app store.

This example typically follows the practice of composite evaluations under the Common Criteria scheme and is applied today in the smartcard environments. While car manufacturers are not expected to directly use a scheme like Common Criteria, a similar approach would contribute to ensure:

- That a given HW is a secure basis for an ECU (security certification of a Tier-2)
- That a given OS is secure when used on a given HW (security certification of a Tier-1 or car manufacturer)
- That *clearly defined, and easily verifiable* rules have been defined for applications, so that they do not threaten then security of the OS (security validation of an aftermarket application)

[125] See http://www.globalplatform.org/specificationform.asp?fid=7828

Enforcing the liability

This report does not condone a particular way to enforce liability. This may be an initiative from insurances, or a legal requirement, or an industry consensus on contractual or commercial commitments. Industry actors may also consider refining their Code of conducts, in order to display clearly the extent of their liability.

5.5 Achieve consensus on technical standards for good practices

Recommendation intended for: industry groups and associations

This reports lists good practices (see section 4.1), which are not meant to be directly applied on a car design. Instead, they are meant as an input for a standardization effort. Industry actors should be aware that a security standard for smart cars should challenge all the categories described in these good practices, in order to be relevant security-wise. The *details* of the security requirements, on the other hand, must be carefully built with regard to actual products, and this report recommends that detailed requirements are defined in the context of a standard. Being able to rely on shared technical standards should contribute to overcome the challenge of *Safety and security process integration* described in section 4.2.3.

5.6 Define an independent third-party evaluation scheme

Recommendation intended for: industry groups and associations

As security awareness increases among car manufacturers, they now include security in the life-cycle of their product:

- Requirements for their products for the design phase,
- Security validation once the product is ready, to check conformity to these requirements and robustness of security functions,
- Security maintenance of the product through updates.

However, the automotive industry mostly assesses security with the same methods as safety (following methods similar to ISO 26262, MISRA or Autosar). These standards marginally address security, and help reducing malfunctions and failures (random and systematic faults), but do not protect against attacks.

This issue is part of challenge of *Safety and security process integration* described in section 4.2.3. In order to overcome this challenge, the industry should define security validation processes that *explicitly address abuse cases and attacks*, which requires a simulation such attacks (in other words, penetration testing).

This requires different skills, and a different mindset as validation testing based on compliance to specifications. For this reason, we recommend to build upon the existing skills and evaluations schemes already in use amongst security professionals.

An example of such a scheme can be found in the initiative led by the Car-to-car communication consortium, which aims at defining a Common Criteria Protection Profile (PP), at the EAL 4 level, for vehicle communication devices. The PP may not address all the categories of good practices of this report. However, the integration of the Common Criteria scheme ensures the security assessment by skilled third-party laboratories, supervised by national cybersecurity agencies, following a standard process.

Some initiatives[126] [127] ask for collaboration on the security topics from the automotive industry, and suggest dedicated security testing from actors skilled in penetration testing.

We suggest that the industry builds on these example to clarify a shared standard for security validation. There is a need to define which method should be used (from basic security checks to penetration testing), the expected amount and depth of testing depending of the component to be tested, and the trust model for these tests (for example, agreeing on a trusted third-party to grant certificates based on security evaluation).

5.7 Build tools for security analysis

Recommendation intended for: industry groups and associations, security companies

Additionally to previous recommendations, industry actors may find other ways to improve their security testing skills. In particular, the development of dedicated tools appears as relevant for several activities.

This report provides a first effort in the definition of tools for:

- Asset identification:
  - See section 2.3 providing a first categorization of assets,
- Threat modelling:
  - See section 3 providing a first categorization of threats,
  - See Appendix A providing example of scenarios and risk ratings formulas according to the TVRA method.

Industry actors should challenge these tools and further contribute on topics where tools provide the most value:

- Security testing, for example by defining fuzzing tools;
- Security monitoring, for example by defining intrusion detection on technologies such as CAN.

[126] https://www.iamthecavalry.org/domains/automotive/5star/
[127] https://www-ssl.intel.com/content/www/us/en/automotive/automotive-security-review-board.html

6. Glossary and abbreviations

ACRONYM	DEFINITION
BTS	Base Transceiver Station
CAN	Controller Area Network
ECU	Electronic control unit
GPS	Global Positioning System
HSM	Hardware Security Module
HUD	Heads-up display
HW	Hardware
IoT	Internet of Things
IoV	Internet of Vehicles
ITS	Intelligent transportation system
MitM	Man-in-the-Middle
MSIN	mobile subscription identification number
OBD	On-board diagnostic
OEM	Original Equipment Manufacturer
OS	Operating System
OTA	Over-The-Air
PKI	Public Key Infrastructure
RF	Radio Frequency
SMS	Single Messaging System
SoC	System-on-Chip
SW	Software
TCU	Telematics control unit
TPM	Trusted Platform Module
V2X	Includes the notions of - Vehicle-to-Vehicle communications - Vehicle-to-Infrastructure communications

7. Appendix A: Detailed risk ratings for the attack scenarios

These scenarios have various levels of likelihood and impact on sensitive assets. To illustrate this, hereafter is an example of risk rating. The rating uses the risk assessment method defined in TVRA[128], but:

- This should not be considered a substitution for a real risk assessment on a car system
- We apply this method to attack scenarios instead of vulnerabilities (in the TVRA sense)

For these reasons, this rating should only be seen as a way to show how threats need to be assessed, in order for manufacturers to define priorities on the security issues that might try to prevent. This is also meant to show that threat assessment methods are follow philosophies than risk assessment methods in a safety context.

The Table 7 hereafter summarizes the ratings for the scenarios selected in this report, while Table 8 gives a rationale to explain the ratings.

[128] ETSI TS 102 165-1 V4.2.3 (2011-03) - Technical Specification - Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis

1083

1084

Table 7 : Risk rating for the scenarios

SCENARIO	RELATED ASSETS	INTENSITY	ASSET IMPACT	TIME	EXPERTISE	KNOWLEDGE OF THE TOE	OPPORTUNITY	EQUIPMENT	ATTACK POTENTIAL	RISK LIKELIHOOD	RISK
1. Attacks threatening passengers safety	vehicle systems, including **vehicle safety systems**	High intensity	High	> 6 months	Expert	Public	Moderate	Standard	High	Unlikely	**Major**
2. Persistent vehicle alteration by the legitimate user	The assets primarily targeted are mostly related to **access control**, especially access to functions not intended for users (fleet management, chronotachograph, geofencing...). Studies give example of privileged services than can be compromised because static keys were discovered by a memory dump (for example SSH keys) Other targeted assets are the **driving systems**, especially in cases where the user tries to modify the performance of their vehicle **vehicle safety systems**	Single instance	High	<= 1 month	Proficient	Restricted	Easy	Standard	Moderate	Possible	**Critical**

	may also be at risk due to accidental side effects of the attack. Modified traffic on the CAN bus may for example trigger denials of service on the bus, or otherwise cause dangerous situations to arise on vehicle systems										
3. Persistent vehicle alteration by diagnostic equipment	**IP and Trade secrets** may be targeted. In a context related to organized crime, the assets are more likely to be **vehicle safety systems**, **driving systems** or **private data** (especially payment data)	Moderate intensity	High	<= 1 month	Expert	Restricted	Moderate	Specialized	High	Unlikely	**Major**
4. Theft	The **vehicle** itself, the **content** of the vehicle (owner's possessions) [and any data accessible through the head-up unit]	Single instance	Medium	<= 1 month	Proficient	Public	Moderate	Standard	Moderate	Possible	**Major**
5.1 Targeted Surveillance	**private data**, notably location-aware content, but also communications or payment data if any	Single instance	Medium	<= 6 months	Expert	Public	Easy	Standard	High	Unlikely	**Minor**
5.2 Mass surveillance	**private data**, notably location-aware content, but also communications or payment data if any	Moderate intensity	Medium	> 6 months	Expert	Public	Moderate	Specialized	High	Unlikely	**Major**
5.3 Surveillance (via cloud)	**private data**, notably location-aware content, but also communications or payment data if any	High intensity	Medium	<= 3 months	Expert	Public	Unnecessary	Standard	High	Unlikely	**Major**

1085

1086

Table 8 : rationale for the rating

SCENARIO	EXPLANATION OF THE RATING
1. Attacks threatening passengers safety	Intensity is considered high, since the attack typically allows to be performed by several agents at a time (exploit kit), or to be performed on several vehicles at a time (sequentially assigned phone numbers). Asset impact is high, since safety is at risk. Time, expertise, knowledge of the ToE and equipment are all rated in a way that reflects existing attacks made by researchers (for example Miller and Valasek). Opportunity is estimated at "moderate": an attacker can work on their own vehicle, which means it still is expensive, and restricts the number of models on which the attacker can work
2. Persistent vehicle alteration by the legitimate user	Intensity is rated as "single instance", since a physical access is required. Impact is rated as high. The attacker may damage their vehicle beyond repair, and may also put their own safety at risk. Expertise is rated as proficient, since the scenario is typically aimed at proficient users trying to tune or modify their own vehicle. Knowledge of the TOE is supposed to be "restricted": online communities are a factor of information-sharing for this public, and information known only by garages may be found is such communities. Time is rated under a month for the same reason. Opportunity is estimated at "easy", since an attacker typically work on their own vehicle (even if one may argue that the vehicle is still, and restricts the number of models on which the attacker can work. Equipment is supposed to be standard.
3. Persistent vehicle alteration by diagnostic equipment	Intensity is moderate because while it needs a vehicle to be accessed via diagnostic equipment, an example of this has been described as repeatable on a wide range of models. Asset impact is high due to the potential safety risk. Time is estimated at under 3 months. Expertise is "expert" because the attacker needs car-specific knowledge (to alter an ECU firmware), as well as they need to know how to reverse a DLL and exploit bad digital signature implementations. Knowledge of the TOE is expected to include "restricted" information, due to the attacker having potentially access to restricted diagnostic tools and data. Opportunity is rated at "moderate" since most of the work is performed on the DLL, which is more readily accessible than the vehicle. Equipment is rated at "specialized" since an access to diagnostic tools will be needed at some point.
4. Theft	Intensity is considered "single instance": while it can be repeated on several vehicles of the same model, there is still needs a physical access for each (since theft is the ultimate goal). Impact is medium (as opposed to safety issues that are considered High). Time is estimated at under 1 month, to reflect the fact that information sharing within criminal networks may contribute to a relatively easy reproducibility of attacks. Expertise is estimated at proficient, since the simplest methods are similar to remote control hacks that are already used today [reference needed]. Opportunity is moderate. Only standard equipment is required
5.1 Targeted Surveillance	Intensity is by definition "single instance". The impact is considered Medium, since safety may only be threatened in a second step. Time is supposed to be inferior to 6 months, since a physical access is possible
5.2 Mass surveillance	Intensity is "Moderate", since it is only repeatable for cars having a given set of vulnerabilities. The impact is considered Medium, since safety may only be threatened in a second step Time is supposed to be superior to 6 months, since a remote exploitation is needed
5.3 Surveillance (via cloud)	Intensity is "High", since it is repeatable for all vehicles using the same cloud services (possibly whole fleets for a leasing company, etc.). The impact is considered Medium, since safety may only be threatened in a second step

	Time, expertise, opportunity and equipment are rated to reflect that the technical domain is widely known to potential attackers (Cloud APIs and interfaces)

1087

8. Appendix B : detailed good practices

8.1.1 Policy and standards

Table 9 summarizes the good practices selected during the interviews.

Table 9: Policy enforcement good practices as selected by interviewees

POLICY AND STANDARDS	DETAILS
Enforce liability	manufacturer for tier-1 and tier-2
Enforce liability	manufacturer for damages due to compromised garage
Adhere to regulation	-

When consulting experts, a few policy enforcement topics were discussed:

- Industry actors should, as a first step, adhere to regulation related to security and privacy. Well aware of the regulation, several experts highlighted the lack of proper cybersecurity regulation for their field;
- Car manufacturers should be held liable for damages due to other actors under their control, notably Tiers and garages;
- Enforcing liability for damages due to aftermarket products was less a consensus amongst interviewees. The measure is practically difficult, thus is addressed in this report under the Gaps and Challenges section (4.2);
- Eventually, liability can only be measured by the compliance to a shared standard and process, which is also lacking today (see 4.2)

8.1.2 Organizational measures

Table 10 hereafter summarizes the good practices selected during the interviews.

Table 10 : Organizational measures as selected by interviewees

ORGANISATIONAL MEASURES	GOOD PRACTICE
Designation of a security team	Design
	Pentesting
	Risk management
	Corporate security
	Training and awareness
Information Security Management System	Define an ISMS (ISO 27001, NIST 800-53, SAE J3061 section 7…)

Designate a dedicated security team. As dealing with cybersecurity issues requires a very narrow set of skills, actors of the smart car industry should rely on specialists for several kinds of activities, notably risk management, secure design, training and awareness, penetration testing and corporate security. Whether this security team should be in-house or a third-party company is not indifferent in some cases; in particular, risk management and corporate security require too much company knowledge to be easily outsourced.

Define a dedicated Information Security Management System (ISMS). Vehicles in the wild cannot be completely protected if the company itself is not able to protect some particularly sensitive assets. For example, if vehicles or components have keys injected during production, the risk of leaking these keys may be more important on the company site than on vehicles themselves. For this reason, an effective ISMS may be of some help. The SAE J3061 describes such an ISMS[129], and references to standards often used to this purpose (ISO 27001 and NIST 800-53).

8.1.2.1 Secure Development or outsourcing

Assess the threat model and use cases. This report gives examples of attack scenarios, along with a risk rating, inspired by the TVRA method, for each scenario. Similar (albeit more detailed) risk assessment is to be expected from any actor involved in smart car components development. The threat analysis itself can follow several possible methods, none of them being a standard. SAE-J3061 describes a TARA (Test And Risk Assessment) phase, which fully supports the EVITA, TVRA, OCTAVE and HEAVENS approaches.

Provide security by design. The security should be taken into account no later than the design phase, in order to avoid unnecessary workarounds, refactoring costs, or worse: leaving vulnerabilities unaddressed because a fix would be unpractical or too expensive. In particular, the secure design should demonstrate how the vehicle security covers the threats identified in the risk assessment. Design should also take into account cybersecurity key principles such as defense in depth or principle of least privilege[130].

Implement and test the security functions. The test phase should also assess how hard it is to bypass the existing security functions, activity which is typically performed by penetration testing. Examples of security controls and measures are described in the next section. These technical measures are sorted using categories loosely adapted from the Common Criteria[131] security certification standard. These categories are:

- **Security Audit**: security events must be logged, and users should be notified whenever needed;
- **Communication protection**: communication should be protected against disclosure, modification, replay and denial of service;
- **Cryptography**: Confidentiality, integrity and authenticity must be protected by using strong and standard cryptography. Keys must be managed securely, and the use of a trust infrastructure (such as PKI) is encouraged;
- **User data protection**: the integrity, confidentiality and authenticity of user data must be protected. Confidentiality protection must be defined with regards to privacy issues;
- **Identification, authentication, authorization**: strong authentication methods must be used, as well as access control mechanisms. Passwords and sessions should be managed accordingly;
- **Self-protection**: HW and SW self-protection measures should be in place to protect previous security functions. Data used to enforce these security functions should be protected, and hardening should be used to reduce the attack surface.

8.1.2.2 Security measures until the end-of-life

Following the good practices described so far shall significantly reduce the risk of having vulnerabilities found in the product, however this risk can never be avoided. Vendors shall not only pro-actively perform a survey

[129] See section 7 of SAE-J3061, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems, January 2016
[130] See for example section 5 of SAE-J3061, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems, January 2016
[131] http://www.commoncriteriaportal.org

for new vulnerability but also provide a secure and reliable device update mechanism to allow fixing vulnerabilities.

Assess the security controls and patch vulnerabilities using appropriate assessment procedures. Determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system.

Define a security update policy. The notion of security update has to be applied to smart cars with several specifics in mind:

- The timing and conditions of the update are different in a vehicle than on a personal computer (users should not be forced to wait for an update before they can start driving. On a similar note, it would be unacceptable to disrupt operations when the vehicle is driving);
- A connected vehicle includes several types of components with different update policies: apps, secure elements and ECUs cannot be updated the same way. While a secure OTA update seems theoretically possible for all components, the need for physical updates might still be present in the next years in many cases;
- Standard are still missing for these operations. While several OTA update framework already exit in several domains, the car community still has to commit on a given, secure process if they want the same channels to be used for manufacturers, Tier-1, Tier-2 and aftermarket developers. Some specific aspects, such as certificate formats, might also need standardization to be fully adaptable to the connectivity constraints of connected vehicles.

Some recommendations apply to the update policy:

- The end-user must be informed of the support period of the device and of the end of support for security fixes.
- A patch may consist of a workaround if the developer did not yet provide a fix.
- When over-the-air updates are not available, a plan for product recalls shall be considered.
- For online services supporting smart cars, a rollback to a secure state must be possible.

Other aspects of the update cannot be addressed directly by this study. For example, applying security updates must be done only when it cannot cause a safety issue, which requires each manufacturer to define appropriate policies. In the same manner, manufacturers may have to think whether a vulnerable component can, or should, be put offline when found vulnerable.

Perform vulnerability survey. Once a device is on the market, the vendor must perform a vulnerability survey and fix security flaws accordingly. The vulnerability survey should include developer findings, on-line researches, CERTs advisories, as well as input from customers and security researches. Eventually, vulnerabilities impacting user data should be communicated as transparently as possible. The EU Opinion 03/2014 on Personal Data Breach Notification from the Article 29 Working Party gives examples of such situations[132]. Manufacturers already move towards using dedicated Security Operations Centers to monitor their infrastructures[133]. While a SOC generally do not delve into in-vehicle vulnerabilities, it may:

- Detect anomalies that are an indication of a vehicle compromise

[132] http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2014/wp213_en.pdf
[133] See for example https://www.sogeti.com/explore/press-releases/Sogeti-sets-up-a-security-operations-center-for-the-Renault-Group/

- Prevent compromising critical functions of the infrastructure, such as remote provisioning or OTA updates

Building a strong security community on a given domain gives many benefits:

- Information sharing groups such as Carsec in Europe, or ISACs, can contribute to raise awareness amongst industry actors;
- CERTs prove useful in informing users of possible vulnerabilities and remediation. While existing CERTs can occasionally play this role for automotive use cases[134], a dedicated CERT might prove more efficient. CERTs might however be better suited for emergency response on infrastructures, such as V2X infrastructures;
- Having a transparent dialog with security researchers, may provide manufacturers with a quicker assessment of their products' possible flaws. It may also "push" the whole community towards more responsible disclosure practices,
- Setting up bounty programs, as already done by several car manufacturers, can also help finding flaws before they are exploited by malicious actors.

A few more recommendations apply:

- A policy for vulnerability handling and disclosure awareness should be defined135.
- Bug bounty programs can also provide an incentive to third-party researchers136 137.
- Known vulnerabilities must be patched138.

Check the security assumptions regularly during life-time. The devices and services made assumptions to ensure that the security requirements are sufficient. Vendors and users should be encouraged to check regularly that these assumptions are still valid. For example: limitations in the usage of the vehicle[139], assumed properties of the environment[140], assumed properties of cryptographic properties [141]…

Protect the software update mechanism. In all cases, the update process requires the vehicle to authenticate the party providing the update, as well as the carrier of this update (for example SMS authentication does not replace the firmware signature, but is used as a complementary countermeasure)

Security updates provide protection against vulnerabilities found during the life of a device or application[142]. However this comes at a cost, since support of this functionality also provides an entry point for an attacker. In particular vendors should:

- Provide automatic and timely security updates[143];

[134] See for example Vulnerability Note VU#615456 - Lemur Vehicle Monitors BlueDriver LSB2 does not authenticate users for Bluetooth access - http://www.kb.cert.org/vuls/id/615456
[135] See The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right
[136] See FTC Careful Connections
[137] See also the global bounty aggregator https://firebounty.com
[138] see Symantec Insecurity in the Internet of things, March 12, 2015 or FTC - Careful Connections
[139] For example, users may be advised to remove connectivity features from their entertainment system until a fix has been found for a given vulnerability
[140] For example, vendors should perform a survey to be able to remove a compromised CA from the certificate store.
[141] For example, vendors should check regularly this assumption, for example if a new cryptographic attack puts users at risk unless they use longer keys or change their cryptographic suites.
[142] see Symantec Insecurity in the Internet of things, March 12, 2015 and Security of Things: An Implementers' Guide to Cyber-Security for Internet of Things Devices and Beyond, NCC group
[143] see Symantec Insecurity in the Internet of things, March 12, 2015 and OWASP I9 | Insecure Software/Firmware

- Protect the updates (typically via encryption and digital signature). The update files must not contain sensitive data [144]. The signature must be verified before the update is applied;
- Protect the *application of an update* on the device. An attacker should not be able to trigger a firmware installation without an authorization;
- Protect the security update interface against attacks;
- Maintain the update server, to avoid attackers using an obsolete domain name to push malicious updates[145].

Raise users' awareness. Vendors should explain users what actions can contribute to mitigate potential threats, especially how to securely use interfaced systems such as a smartphone.

8.1.3 Security functions

This section is structured following the lifecycle of smart cars. Steps are inspired by previous work from NHSTA/NIST[146]

8.1.3.1 Security Audit

Security events must be logged[147], and access to the logs must be documented and protected from disclosure to unauthorized users. Logs are also needed for device integration. Typically, Tier-2 suppliers must give possibility for Tier-1 suppliers to understand security events happening in their products. However logs may also give information to an attacker, which is a serious security drawback. For this reason, the audit trail must be protected[148]

Notifications should be easy to understand and help users find a remediation or workaround. HW and embedded systems should provide clear error data that can be leveraged upon by the SW vendors. The user must be notified in case of security errors, updates or compromised data[149] in a device or service they use. In particular, users must be notified in the case of security events[150]. Notification might vary greatly depending on the type of software considered. Mobile applications notification, messaging such as SMS or e-mail, hardware interfaces such as LEDs, dedicated error messages to a gateway[151]…

---

[144] See OWASP I9 | Insecure Software/Firmware

[145] See Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage

[146] See National Institute of Standards and Technology cyber security risk management framework applied to modern vehicles

[147] See Security of Things: An Implementers' Guide to Cyber-Security for Internet of Things Devices and Beyond, NCC group and see OWASP I8 | Insufficient Security Configurability

[148] Such protection can typically consist in the following practices

- Logs should be anonymous;
- Avoid logging information that would give useful information to an attacker ;
- Access control mechanisms should limit the access to the logs;
- When sent to a remote system, logs should be protected by cryptographic mechanisms

[149] See The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right

[150] see OWASP I8 | Insufficient Security Configurability

[151] Developers should be aware that for some functions, an excess of clarity is a valuable information for an attacker. As a common example, when a login fails, the product should not communicate to the user whether the error is due to a non-existent login or a bad login/password combination. The optimal balance between *not enough* or *too much* clarity is to be assessed during dedicated security testing.

8.1.3.2 Communication protection

Provide end-to-end protection in confidentiality and integrity using protocols that resist to replay attacks. Favor methods providing forward secrecy whenever possible. This should be true even for the communication of already encrypted data[152]; encryption must cover not only WAN traffic (internet, mobile network), but also local network[153].

Mitigate vulnerabilities or limitations of standard security library. Using a standard security library does not mean that the product will automatically be secure. Developers must be aware of the vulnerabilities (due to a flawed implementation) and limitations (vulnerability of the protocol itself) of the third-party components they use. They should mitigate them whenever possible by performing patching[154] and by securing the configuration of the communication stacks[155], which might typically include Bluetooth[156], Wi-Fi[157], TLS[158]…

Consider denial of service as a usual threat to communication infrastructures[159]. This threat should be addressed from the design phase of the infrastructures. On this topic, this study encourages the vendors and service providers to read the ENISA Internet Infrastructure Threat Landscape (for network components)[160] or the GSMA IoT Device Connection Efficiency Guidelines[161].

Protect remote monitoring interfaces. SMS commands should not be protected only by whitelisting[162]. For this reason, privileged commands such as SMS administration commands shall be protected by mutual

---

[152] See OWASP I9 | Insecure Software/Firmware, or Symantec Insecurity in the Internet of things, March 12, 2015. Many protocols use both transport layer and applicative layer protection. The need for applicative layer protection comes from end-to-end protection needs: the transport layer could be exposed if different transport technologies are used during the transmission, therefore needing a dedicated protection:

- In TCP communications, TLS 1.2 is the default choice for securing the transport layer;;
- Applicative layer can be protected by recognized cryptographic means, so as to protect confidentiality and integrity of the payload.

[153] See OWASP I4 | Lack of Transport Encryption

[154] Third-party and open-source libraries need frequent patching: vulnerabilities are regularly found in all most open-source implementations, even those considered as “industry standard”. Communications protection work only as long as firmware updates are available and applied to fix vulnerabilities.

[155] Due to the existence of vulnerabilities in frequently used protocol implementations, configuration of the library is a significant part of the security functionality. Developers should in particular be vigilant to the configuration of cipher suite negotiation and key sizes: allowing weak cipher suites provides an entry point for attacks aiming at downgrading the level of security of the exchanges (See for example CVE-2015-0204 at https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-0204)

[156] See the example of Bluetooth, including Bluetooth 4.0, in *Guide to Bluetooth Security - Recommendations of the National Institute of Standards and Technology - John Padgette, Karen Scarfone, Lily Chen*

[157] See for instance attacks on WEP http://eprint.iacr.org/2007/120.pdf, WPS PIN vulnerability https://www.kb.cert.org/vuls/id/723755 or the Pixie Dust attack on WPS https://passwordscon.org/wp-content/uploads/2014/08/Dominique_Bongard.pdf

[158] SSL and TLS have a long history of security vulnerabilities (see https://tools.ietf.org/html/rfc7457).

[159] See OWASP I3 | Insecure Network Services

[160] See https://www.enisa.europa.eu/activities/risk-management/evolving-threat-environment/enisa-thematic-landscapes/threat-landscape-of-the-internet-infrastructure

[161] http://www.gsma.com/connectedliving/gsma-iot-device-connection-efficiency-guidelines/

[162] The main reasons for this are that:

- phone numbers can be spoofed.
- whitelists are not secret

authentication. More generally, protection of remote monitoring interfaces is crucial since they often provide a highly-privileged entry point into a device. This protection includes access control and authentication mechanisms.

8.1.3.3 Cryptography

Many protection measures rely on cryptographic functions. In a broad definition, cryptography support for security must include:

- Symmetric or asymmetric encryption;
- Message authentication and integrity;
- User/entity authentication;
- Hash functions;
- Digital signature;
- Key management;
- Random number generation.

Do not create proprietary cryptographic schemes, but use state-of-the-art standards instead.[163] Even a home-brewed implementation of a standard is not a good practice when standard implementations are available. If needed, consider getting advice from security experts or your national cybersecurity agency.[164] If no recommendations exist for vendors at a national level, ENISA recommendations should be considered as a reference.[165] This applies also to random number generation, which is a critical part of the cryptographic support. A possible recommendation would be the use of cryptographically secure pseudorandom number generators.[166]

Rely on an expert in cryptography for interfacing with HW accelerated cryptography or secure elements, or even using or configuring a standard implementation. These tasks are difficult for most of developers. If not properly done, the security might be heavily reduced or even completely suppressed. This part should be performed by an expert in cryptography or at least a third-party code review should be performed to ensure that HW or a standard implementation of cryptography is properly used.

Consider using dedicated hardware security modules. HW-based cryptography solutions may help avoiding the incorrect implementation of cryptographic algorithms by software vendors, as well the coexistence of multiple implementations of the same algorithms. They eventually provide implementations that are more resource-efficient. Choosing HW accelerated cryptography means that a reasonable assurance must be obtained on the quality of the HW implementation, since “bad cryptography” on HW will be leveraged on all the SW using these functions[167].

- whitelists may be changed by other SMS commands (administration commands).

[163] See for example see Symantec Insecurity in the Internet of things, March 12, 2015 or Careful connections by FTC

[164] This study will not delve into the detailed requirements for cryptographic algorithms or acceptable keys sizes, since most of national cybersecurity agencies already provide consistent guidance on this topic

[165] See https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-size-and-parameters-report-2014

[166] See examples in http:/csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf

[167] Random number generators are a good example of vulnerable functions with an impact on many features.

- As a general rule, a true random number should be used for key generation, but may not be required for salts, initialization vectors… where a cryptographically secure pseudo-random number may be sufficient.

Eventually, using certified HW may solve most of these issues. In particular, Manufacturers may look for independently audited HW. The standard for independent assessment of security HW would be in that cas either FIPS 140-2 or Common Criteria certification following relevant Protection Profiles. If needed, consider getting advice from security experts or your national cybersecurity agency.

Cryptographic keys should be securely generated, distributed (or provisioned), used, stored, and deleted (including revocation). Badly implemented key management can introduce vulnerabilities that may easily be exploited. Devices without direct user interfaces are particularly vulnerable to PKI compromising. While users of a PC can easily delete or install certificates, such devices rely mostly on remote administration, and sometimes do not even allow end-users to perform such administration tasks. For this reason, Manufacturers, as well as Tier-1/Tier-2 and aftermarket vendors should consider very carefully the revocation mechanisms associated with their components. This is especially true when the mechanisms of key provisioning and management are performed over-the-air[168]. If needed, consider getting advice from security experts or your national cybersecurity agency[169].

8.1.3.4 User data protection

Identify personal data. The interpretation of privacy protection raises many issues, one of them being to successfully identify what can be considered a personal data. The definition according to the EU Directive 95/46/EC includes data *relating to an identified or identifiable* person. In the case of smart cars, however, it may be safe to assume that *most data* related to the user activity are somewhat personal, especially location-based data. This last approach will have to be continued throughout the whole product or service lifecycle. Metadata should be considered as personal data by default, since they are subject to the same threats[170]. Consider getting advice from your national data protection agency.

One may argue that using a cryptographically secure software pseudorandom number generator is more secure than a badly implemented hardware "true random number generator";

- When using hardware claiming a "true random", developers should consider using strong post-processing functions. The functions used for that purpose are typically block encryption or hash functions;

More details on the different categories of random generators can be found in documents from national cybersecurity agencies. See in particular A proposal for: Functionality classes for random number generators, Version 2.0 , 18 September 2011, by the BSI.

[168] Industry players introduced the notion of remote provisioning for mobile communication (See for example GSMA remote provisioning architecture and Security of Things: An Implementers' Guide to Cyber-Security for Internet of Things Devices and Beyond, NCC group). While keys are loaded in SIM cards in protected environment, the "embedded UICCs" rely on remote subscription management systems to obtain key material. The protection of these exchanges is consequently critical and must be assessed accordingly by manufacturers and vendors. Should the keys be leaked, the user and the vendors could be at risk in many ways (loss of control over the device, eavesdropping, credential theft, cloning…). More generally, the notion of confidential key agreement must be considered in IoT in general, and smart cars in particular.

[169] This study will not delve into the detailed requirements for cryptographic algorithms or acceptable keys sizes, since national cybersecurity agencies already provide consistent guidance on this topic

[170] See http://www.lifehacker.com.au/2015/02/why-the-internet-of-things-is-a-problem-for-metadata-retention/

Implement transparency measures. The interactions with the user (which should not be limited to the *Terms and conditions*[171]) enable to cover the legal transparency requirements[172].

Design the product/service with legitimate purpose and proportionality in mind. The design phase of the service or product, where the details of the processing have to be assessed with regards to the explicit and legitimate purposes. The actors must ensure that themselves *and their subcontractors or suppliers* do not process user data more than needed, and do not pursue an illegitimate purpose with regard to user data. As a general rule, third party components integrated in the device or third party cloud services should not access unencrypted user data unless user agreement has been obtained. Access control or anonymity/pseudonymity measures gives assurance that user data is not accessed by these third parties.

Define access control, anonymity and unlinkability measures to enforce the protection of private data. These measures are typically access control measures[173], pseudonymity and unlinkability measures (such as ensuring that data is not correlated[174]), and eventually anonymity measures. Anonymity measures may be

[171] While the Terms and Conditions are a practical support for the vendors, many actors consider that this cannot be considered a good practice. In particular, the user may be lost in a barely-legible legalese instead of being able to make informed choices regarding their privacy. The US FTC gives recommendations on this topic, for example using other supports such as registration emails.

[172] The service or device provider must communicate

- The provider's name and address;
- What data is collected, in layman terms;
- The purpose of processing, explaining notably why the processing is necessary for the performance, to protect the vital interests of the data subject, or for compliance with a legal obligation;
- The recipients of the data;
- How the user can:
  - Access all data processed about him,
  - Require the rectification, deletion or blocking of data that is incomplete, inaccurate or isn't being processed in compliance with the data protection rules.
- And all other information required to ensure the processing is fair;
- The service or device provider must require the consent of the user (or "data subject").

On top of legal requirements, actors might also consider:

- Defining a strict opt-in policy;
- Enabling rectification, deletion or blocking of data without a reason;
- Ensuring data portability.

[173] As a general rule, access to sensitive data should be controlled. For web services and components including virtualization, access control could be completed by data isolation

[174] The typical example is ensuring that the key used to browse the "customer database" is not the same as the key used to browse the "usage analytics database". However the situation is more complicated in practice: in the case of smart cars, for example, network locator is a critical factor of linkability and should be taken into account accordingly. Vendors should also be aware, that unlinkability can also:

- Cause trust issues and reduce attack mitigation capabilities (for example if a user cannot be notified that their device is compromised);
- Cause a conflict with other legal requirements.

There is no one-size-fits-all good practice to balance unlinkability against other desired properties. The right balance must be defined during the design stage by examining the associated risks.

"one-way" or "non-reversible" (such as truncation[175] or a hash functions[176]) or "reversible" such as encryption[177].

Define measures to ensure secure deletion of user data in case of a change of ownership. More generally, a secure factory-reset of the firmware and configuration should be available on the vehicle.

8.1.3.5 Identification, authentication, authorization

Use mutual authentication for remote communication. Devices or users connecting to a server must be able to authenticate the server. Reciprocally, servers must be able to authenticate clients and users. Mutual authentication[178] consists in demonstrating cryptographically to both the client and the server that they are communicating with the expected party. Mutual authentication is generally performed by using Public Key Infrastructures (PKI) and certificates. These methods can be embedded in protocols such as TLS. However using methods such as TLS does not grant a secure mutual authentication, unless:

- There is a certificate for *both the server and the client;*
- Certificate are properly validated (ruling out, for example, the use of self-signed certificates);
- Revocation lists are verified (alternatively, interrogations to an OCSP server);
- *All services* require this authentication step[179]. Which also means that even private URLs accessible on a device must require authentication;
- Certificate pinning is used[180].

As a side note, it must be noted that certificate pinning does not eliminate the need for certificate validation. For example, the pinned certificate can be an intermediate or root Certificate Authority (CA) – which means that the end certificate still has to be verified against the CAs.

Use multi-factor authentication for user authentication. Users should be authenticated by 2-factor authentication whenever possible, including for authentication to cloud services or mobile interfaces[181], as well as local administration sessions of devices. Several methods can be used for multi-factor authentication. As an example, the NIST provides a summary of these methods[182].

---

[175] Truncation is often used in the payment industry to anonymize cardholder data (see https://www.pcisecuritystandards.org/documents/PCI%20SSC%20Quick%20Reference%20Guide.pdf)
[176] Hash functions also have vulnerabilities (see for example https://en.wikipedia.org/wiki/Cryptographic_hash_function). As for other cryptographic operations, robust standard mechanisms should be preferred – vendors are encouraged to contact their national cybersecurity agency if needed.
[177] See The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right, OWASP I5 | Privacy Concerns and OWASP I10 | Poor Physical Security. As a sidenote, encrypted storage can also address authenticity or integrity of user data if combined with the right mechanisms (for example AES-GCM).
[178] see Symantec Insecurity in the Internet of things, March 12, 2015
[179] See See Home Automation Benchmarking by SYNACK, but also Making Smart Locks Smarter (aka. Hacking the August Smart Lock), The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right.
[180] See Home Automation Benchmarking by SYNACK or Making Smart Locks Smarter (aka. Hacking the August Smart Lock). For details on Certificate pinning, see https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning#What_Is_Pinning.3F
[181] see OWASP I2 | Insufficient Authentication/Authorization, I6 | Insecure Cloud Interface, I7 | Insecure Mobile Interface
[182] See NIST Special Publication 800-63-2 – Electronic Authentication Guideline

Implement access control measures to separate the privileges of different users as well as the privileges of different applications. In practice, privileged operations should not be readily accessible to normal users. Reducing access to these services can be achieved either by disabling them (some studies recommend disabling WAN administration, for example, since it provides a remote entry point to privileged services [183]; local administration such as JTAG could also be deactivated by using fuses) or by introducing dedicated access controls. Typically:

- An administrative access should always require authentication, and should ideally require unique credentials for each device[184];
- Not all individual accounts need to have access user data stored in the device or associated services[185];
- User accounts must be unique and separated for both local and distant services[186];
- The device must distinguish between normal users and admin users. The latter only have access to configuration functions[187].

Implementing privilege levels, rings or domains can also be extended to application separation. Some platforms implement such levels in hardware. If such functions are available, vendors are advised to use them[188]. If not, operating systems already provide capacities to implement privilege control. At the firmware / software level, access control must be used to control access rights of *both applications and individuals*. In particular, not all applications need to be root or be executed in kernel land.

Allow and encourage the use of strong passwords. As it is regularly demonstrated, passwords are often a weak point, whether they are weak user passwords or weak default passwords for products internal services. Many devices use strong protection measures that are defeated by the lack of proper password management[189]. This concerns all possible uses of passwords: direct device interfaces such as JTAG, but also web, mobile or cloud interfaces. The usual measures are the following:

- Allow and encourage the use of strong passwords[190], regardless of the presence of a second authentication factor;
- Require the user to change credentials (username, password) at their first login[191];
- Do not use hard-coded or “default” passwords or shared passwords, for instance for remote support accounts;

[183] See for example Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage
[184] See for example Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage
[185] I5 | Privacy Concerns
[186] See The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right
[187] See OWASP I8 | Insufficient Security Configurability
[188] See "Security of Things: An Implementers’ Guide to Cyber-Security for Internet of Things Devices and Beyond, NCC group"
[189] See for example Fast and Vulnerable: A Story of Telematic Failures, Ian Foster, Andrew Prudhomme, Karl Koscher, and Stefan Savage
[190] See I2 | Insufficient Authentication/Authorization and OWASP I1 | Insecure Web Interface; See also see Symantec Insecurity in the Internet of things, March 12, 2015
[191] See OWASP I1 | Insecure Web Interface, OWASP I6 | Insecure Cloud Interface, OWASP I7 | Insecure Mobile Interface

- Do not store/expose passwords in clear text or with weak protection. Adaptative one-way functions such as PBKDF2, scrypt or bcrypt should be preferred[192];
- Use countermeasures against password guessing / account harvesting[193]. Services must be protected against:
  - horizontal guessing (testing a small number of usual passwords on a high number of user accounts);
  - vertical guessing (testing a high number of passwords on a single user account)
  - This typically includes lock-out and delaying measures as well as high password strength / entropy and diversification of passwords across devices. This also includes countermeasures against account discovery or other means used to exploit password recovery functions[194];
- Define options for password control. Typically, in the case of an administrator account, the default option should require strong passwords by default[195.196.]

Password policies are eventually useless if the final user is not fully aware of the threats and good practices. Vendors and service providers should consider raising the awareness of their users whenever possible, for example to support the use of password managers. Examples of simple guidelines can be found in *ENISA Basic security practices regarding passwords and online identities*[197].

Since the use of strong passwords is not acceptable for normal users interactions in a moving vehicle, this good practice is recommended mainly for setup and pairing activities, and especially for administration or diagnostic features.

Enforce session management policies to avoid session hijacking. Session management also contributes to making sure that the authorized user is the one using a given session. Typically:

- Sensitive functions such as administration via web services should require re-authentication.[198]
- No data should be transmitted before authorization.[199]
- Strong (random) session handlers should be used to avoid replay.[200]

[192] See https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet. Hash functions such as MD5, SHA should not be used for password protection, and even SHA256 or SHA3 would lack the additional work factor to be efficient in a password storage context
[193] see Symantec Insecurity in the Internet of things, March 12, 2015
[194] see OWASP I2 | Insufficient Authentication/Authorization
[195] See OWASP I2 | Insufficient Authentication/Authorization and OWASP I8 | Insufficient Security Configurability
[196] An example of policy can be found at https://www.sans.org/security-resources/policies/general/pdf/password-protection-policy. Policies may vary depending on the threat analysis and dimensions (such as password length) also depend on attacker's capabilities, especially the computing power, which grows constantly over time. Vendors are invited to contact their national cybersecurity agency or CERT to stay informed of the state-of-the-art.
[197] See http://www.enisa.europa.eu/media/news-items/basic-security-practices-regarding-passwords-and-online-identities
[198] See OWASP I2 | Insufficient Authentication/Authorization
[199] See The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right
[200] See for example Veracode White Paper – The Internet of Things: Security Research Study, 2015, and also The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right

- The user must know at any time if, and why, they are logged on a particular service, meaning that no passive sign-up for third party services should be performed.[201]

8.1.3.6 Self-protection

Define a consistent policy for self-protection. Self-protection includes all measures taken to enhance the robustness of previously mentioned security functions. Developers should challenge every security function of their design, consider how they could be bypassed or weakened, and eventually implement self-protection measures. The main topics considered here are:

- **Hardware self-protection**: these measures aim at protecting the hardware against physical attacks or observation. They include tamper evidence or tamper resistance, and secure design measures[202]
- **Software self-protection**: software also contributes to protect existing security functions, typically by validating inputs and outputs, or by separating the capacities of the different software components (levels of trust, virtualization…)
- **Non-user data protection**: data used to enforce the security functions should be protected. These measures intend to avoid storing internal keys as cleartext, or any other data that could be used to circumvent the service security
- **Hardening**: hardening consists in reducing the attack surface of the product or device. This includes removing unused services or interfaces (for instance remote shell access to the device, which should not be needed in production), as well as integrating malware protection. Hardening in smart cars is particularly difficult to address, since these systems are behaving both like embedded and networked systems.
  Some actors have advocated that, in the CAN context, intrusion detection should be used on top of firewalls, in the same manner as usual IT systems use both in a *defense-in-depth* approach[203]. Dedicated solutions are already being commercialized, in order to provide CAN bus monitoring in a fashion quite similar to the traditional IDS/IPS systems[204]. This study will not, however, conclude on the respective merits of these solutions.
- **Isolation**: this subset of hardening measures is especially relevant for the car industry. Isolation of components aims at reducing the capacity, for an attacker, to jump from a component to another. This notion is found in the two main paradigms for CAN bus isolation in cars:

---

[201] See The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right

[202] Hardware protection measures are related to:
- threats that are not related to privacy, and where the user itself is the attacker (for example fraud use cases);
- threats to equipment that is not protected by physical measures.

These are also related to attackers with very high skills and motivation profiles (which is for example the model used in smartcards this includes for example:
- Use of tamper-resistant hardware such as Active shields;
- Protection against glitch;
- Protection against fault injection;
- Protection against side channels (for example electromagnetic or power analysis).

Examples can be found for example in Security of Things: An Implementers' Guide to Cyber-Security for Internet of Things Devices and Beyond, NCC group. Even if this level of security cannot be required for all smart home devices, several physical protection measures can be recommended to ensure a better overall security on the device.

[203] See http://www.automotiveitnews.org/articles/572873/car-hacking-can-be-stopped-by-ips-from-argus-cyber/

[204] See for example http://iotbusinessnews.com/2016/06/08/34788-symantec-launches-new-iot-solution-help-carmakers-protect-zero-day-attacks/ or http://www.automotiveitnews.org/articles/572873/car-hacking-can-be-stopped-by-ips-from-argus-cyber/

- Solution 1 : the CAN bus related to driving systems is “airgapped”, that is, completely isolated from the infotainment network and internet
- Solution 2 : Systems are connected, but a gateway is in place to ensure the isolation between networks, typically by access control mechanisms

These two solutions have architectural consequences – for example, the first only allows physical updates, while the second allows OTA updates.

Studies argue that the second solution is gaining momentum, especially now that the eCall regulation requires a SIM-card to be present in all cars, which provides a channel for updates[205].

Most of the self-protection measures must be considered from the early design phases. Only the hardening can be defined as an additional measure that can take place after the design and implementation phases.

Implement HW tamper evidence / tamper resistance. Devices vendors should be aware of tamper evident or tamper-resistant mechanisms[206]. While they are not recommended in any case, vendors should consider using them depending on the level of sensitivity of the assets stored on the device. In particular, even constrained devices could be able to implement some kind of tamper evidence, even if they are not able to implement resistance and response. More details on anti-tamper technologies can be found at different sources, for example Black Hat[207] or ICCC[208] conferences

Implement HW protections at the design level. Hardware design can be used to make the device harder to attack[209].

---

[205] See for example *Responsibility for Vehicle Security and Driver Privacy in the Age of the Connected Car*, IDC/Veracode, February 2016, IDC #EMEA41026016

[206] This includes typically:

- Basic to moderate “Tamper resistance” mechanisms, which will slow an attacker (this typically includes specific sealing methods for the casing, or the use of epoxy to protect components, or the entire board);
- Basic to moderate “Tamper evidence” mechanisms, such as tamper-evident seals or labels, or even switches or sensors (light, power…) that will trigger a tamper response;
- Basic to moderate “tamper response” mechanisms such as sending an alarm to a remote service, logging a security error or erasing sensitive data.

[207] Introduction to Embedded Security, Joe Grand, Black Hat USA 2004

[208] Physical protection: Anti-tamper mechanisms in Common Criteria security evaluations, Epoche & Espri, ICCC Norway 2010

[209] In particular:

- Memory (including memory controller) can include measures such as secure erase and wear levelling, Direct memory access, Non executable memory, …;
- Printed Circuit Board (PCB) design can contribute to security by including blind and buried vias, buried bus lines, or electronic fuses and similar techniques, for example to deactivate JTAG access (other uses can also be considered).
- System on Chip (SoC) design can include some of the previous measures, and can also include pin placement, or the implementation of “system level” features such as HW Virtualization, micro kernels, Secure boot, Trusted Execution Environments…

Security of Things: An Implementers’ Guide to Cyber-Security for Internet of Things Devices and Beyond, NCC group states that "For chips with security features or functionality that may impact security it is important to understand where these are located on the chip’s pin out. It is generally advisable not to use chips where these features are on the outer two rows in high-security environments due to risk of fly wires being used". Some labs consider today that for "grid array" chip carriers, the outer three or four rows might be relatively easy to access for an attacker. In any

Protect the software security functions by reinforcing interfaces and strengthening the application separation at runtime. Software can contribute to self-protection measures for instance for robustness of interfaces against bad inputs[210]. Secure implementation, thoroughly tested, will protect against common attack vectors such as buffer/heap overflows or OWASP's List of the Top Ten Web Vulnerabilities[211]. This typically includes robustness of network interfaces against buffer overflows or fuzzing[212]. Implement trust zones for the execution of applications (and/or ensuring segregation or execution protection), for example by whitelisting applications, or by using Trusted Execution Environments or Secure boot, or SW virtualization[213]…

The default configuration of devices and services should be secured. The operation mode of the device (or service) should be the most secure one by default. A user might arguably want to disable a given security function, but this should be the consequence of a deliberate action from the user, and the user should be warned that this change reduces the security of the solution[214].

Encrypted storage is not only useful to protect user data, but also to protect data that is needed to enforce security on the device[215]. Internal data may be just as sensitive as user data, but are often not protected enough, leading for example, to situations where "hardcoded root credentials, API keys for Amazon Web Services, URLs never meant to be known to end-users, and manufacturing network configurations"[216] can be found in cleartext on devices. As a general rule, configuration data should be encrypted at rest and in transit[217].

Perform hardening to reduce the attack surface: remove unused services or interfaces, integrate dedicated security software, activate memory or control flow protections. For devices that have a complete operating system, several measures can be considered to harden the device, such as ASLR, non-executable memory, process segregation or sandboxing. Another measure is removing unused tools, services and libraries[218]. Unnecessary services should not be present on the device (typically telnet must

---

case, a consensus is needed amongst stakeholders and security labs on this topic, so cybersecurity agencies could provide vendors with clear recommendations.
The ease of access to the components, as well as their removability, can also be considered during the design phases, even if it cannot be the primary physical protection measure.

[210] see Symantec Insecurity in the Internet of things, March 12, 2015

[211] https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

[212] OWASP I3 | Insecure Network Services

[213] See for example Symantec Insecurity in the Internet of things, March 12, 2015, IoT-A - D4.2 - Concepts and Solutions for Privacy and Security in the Resolution Infrastructure

[214] Providing a secure configuration by default means in practice that

- a remote service will use HTTPS by default
- setup processes include the necessary steps to upload any security configuration data such as certificates
- the stronger password policies will be selected by default
- …

[215] See The Internet of Fails: Where IoT Has Gone Wrong and How We're Making It Right and OWASP I10 | Poor Physical Security

[216] See A Primer on IoT Security Research, March 30 2015, Stanislav

[217] see OWASP I8 | Insufficient Security Configurability and See Security of Things: An Implementers' Guide to Cyber-Security for Internet of Things Devices and Beyond, NCC group

[218] see Symantec Insecurity in the Internet of things, March 12, 2015 and The Internet of Fails Where IoT Has Gone Wrong and How We're Making It Right

always be deactivated, but even SSH or FTP can be deactivated in many cases). This type of measures is also applicable at a network level: the device should not leave open ports, especially ports that could be exposed via plug-n-play protocols[219]. The default configuration of the device should be based upon the most secure parameters, and users should be warned if they have the possibility to roll back to less secure parameters. For example multi-factor authentication should be the default configuration. Users should be warned if they want to configure the service to single-factor authentication. Vendors should also consider integrating malware protection to their systems[220], since the smart home ecosystem provides many possible ways for malware to enter a device (mobile, personal computer, device network interfaces…). Eventually, Vendors should consider deactivation or protection of the external interfaces[221], for example:

- protecting the physical debug interfaces such as JTAG/ISP (by password and physical action), or physically deactivate the physical debug access;
- including mitigation to avoid exploitation of interfaces such as I2C/SPI buses or serial interfaces;
- Suppressing or limiting to a local access[222], the administration interfaces.

More generally, vendors should consider their means of protection for:

- BootROM interface;
- Firmware update interfaces;
- Configuration and calibration interfaces;
- Inter-processor IPC;
- USB external interfaces;
- Protection against DMA attacks[223];
- No unnecessary external interfaces should be accessible from the exterior of the device[224].

[219] See Home Automation Benchmarking by SYNACK, or OWASP I3 | Insecure Network Services
[220] see Symantec Insecurity in the Internet of things, March 12, 2015
[221] See for example Veracode White Paper – The Internet of Things: Security Research Study or Security of Things: An Implementers' Guide to Cyber-Security for Internet of Things Devices and Beyond, NCC group
[222] see OWASP I10 | Poor Physical Security
[223] https://en.wikipedia.org/wiki/DMA_attack
[224] see e.g. OWASP I10 | Poor Physical Security

ENISA

European Union Agency for Network
and Information Security
Science and Technology Park of Crete (ITE)
Vassilika Vouton, 700 13, Heraklion, Greece

Athens Office

1 Vass. Sofias & Meg. Alexandrou
Marousi 151 24, Athens, Greece

Catalogue Number

PO Box 1309, 710 01 Heraklion, Greece
Tel: +30 28 14 40 9710
info@enisa.europa.eu
www.enisa.europa.eu

ISBN: xxx-xx-xxxx-xxx-x
doi:xx.xxxx/xxxxxx