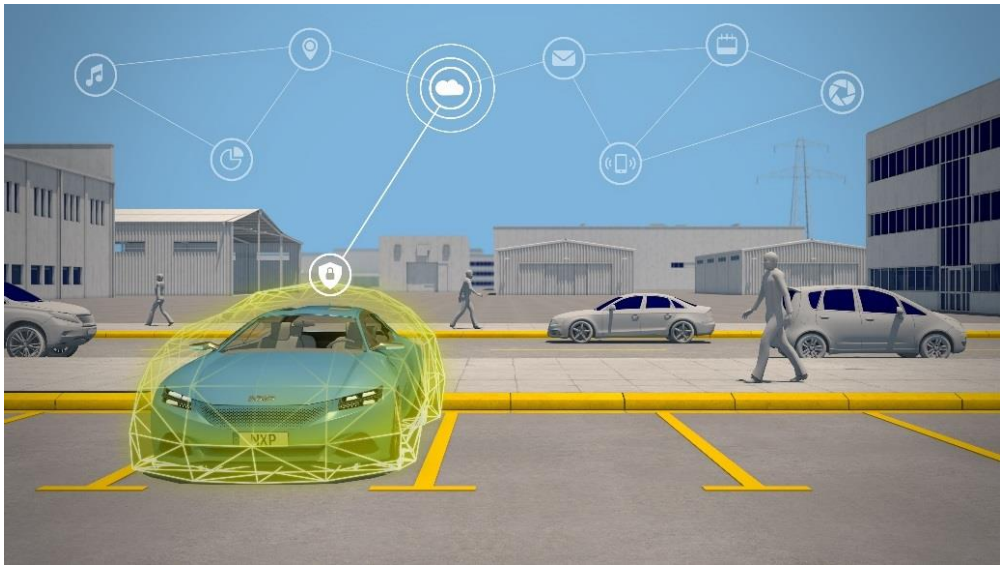
Cybersecurity for ECUs:
Attacks and Countermeasures

Whitepaper

Today's vehicles are, already, an attractive target for hackers. The trend towards autonomous driving only further increases the need for securing the vehicle and its electronic systems (ECUs) against cyberattacks. In this whitepaper, we zoom into the security aspects and needs of individual ECUs inside the vehicle's E&E architecture and we provide guidance on the level of protection that is needed for various ECUs, depending on their function in the vehicle network.

Authors:

Timo van Roermund, Andreas Bening, Fabrice Poulard
NXP Semiconductors



Contents

Introduction

Cybersecurity for vehicles is, currently, a hot topic. The need for security was initially driven by the steep increase in electronic features and the introduction of wireless connectivity, which made the vehicle an attractive target for hackers. More recently, the trend towards autonomous driving is further 'fueling the fire' – the more autonomous vehicles become, the more users must be able to trust their vehicle and its electronic systems. In other words, these systems must do what they're meant to do – also under extreme conditions and when under attack.

In our previous whitepapers, we already elaborated on the need to secure modern vehicles [1] and on our approach to address the security concerns in vehicle electrical and electronic (E&E) architectures in a defense-in-depth approach, using the 4+1 security framework [2].

In this whitepaper, we zoom into the security aspects and needs of individual ECUs inside such vehicle E&E architecture. We start by explaining some of the basic concepts, such as *assets* and *attack surface*, and by providing a high-level overview of the security functions and countermeasures that can be found in modern electronic control units (ECUs). In the second part of this whitepaper, we explore ECU security in more detail. We introduce attack categories, to bring structure into the wide range of attacks that can be mounted against ECUs. Per attack category, we then sketch some example countermeasures. Finally, we provide guidance on the level of protection that is needed for various ECUs, depending on their function in the vehicle network.

Cybersecurity for ECUs

In this chapter, we investigate the cybersecurity needs for Electronic Control Units (ECUs). We explain the basic concepts of *assets* and the *attack surface*, including its individual *attack vectors*. We also sketch what kind of countermeasures can be applied, and where.

Purpose of this chapter is to introduce the basics of cybersecurity for ECUs. The next chapter will explore the topic in more detail.

What is ECU hacking?

In short, hacking an automotive ECU means gaining unauthorized access to that ECU, to extract data (information) from it and/or to impact its operation (functions and features). There are many reasons why someone may want to do this. For example, a car owner may want to unlock certain features (e.g. extra horse power), researchers may want to demonstrate a lack of security in a vehicle model or system, or criminals may want to extort people by installing ransomware that disables (part of) the vehicle. A more comprehensive list of actors and their motivations can be found in our whitepaper "Secure Connected Cars for a Smarter World" [1].

Assets

In the security world, *assets* are defined as the parts of a system which represent a certain value, for example to their owner(s), their user(s), or the manufacturer(s). These parts can be tangible (e.g. the devices and their components) or intangible (e.g. code and data).

Globally seen, the main assets in an ECU are its functions and features, which can be split into two parts: the application (software code, hardware intellectual property (IP)), and its parameters (configuration and data).

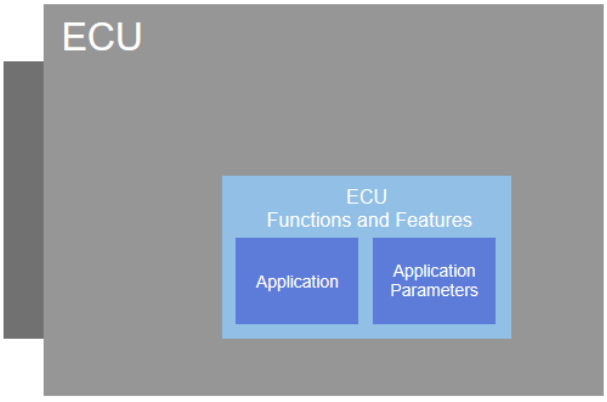


Figure 1: ECU Assets

A compromise of the assets results in loss; e.g. a financial loss, or a loss of trust (in the system). To prevent this, the assets should be protected. Depending on the type of assets, the protection may need to prevent against (unauthorized) access, use, disclosure, modification, theft, etcetera.

Attack surface

So, how can a hacker gain access to the ECU assets? The simple answer is: by taking advantage of vulnerabilities, i.e. security weaknesses, in an ECU. In other words, hackers identify and exploit vulnerabilities in an ECU – and the 'route' by which the attack was carried out, is called the attack vector:

> ***Attack vector:*** *one path or means by which a hacker can gain (unauthorized) access to a system (in our case the ECU) and its assets.*

So, an *attack vector* is *one way* in which a hacker gains access to an ECU and its assets. For example, a ECU may provide a test/debug port or channel that is protected using a weak password. A hacker may guess the password, and bypass the protection, thereby being able to read out (and possibly replace) the ECU's code and data or use internal test functionalities that are not meant to be used in field.

In practice, systems have multiple attack vectors, which together form the attack surface:

> ***Attack surface:*** *The sum of the different paths (the 'attack vectors') where an unauthorized user (the 'attacker') can try to attack a system.*

We will now explore the attack surface of ECUs in more detail.

ECU Interfaces and channels

A typical ECU has one or more interfaces (see Figure 2). These interfaces can be divided into two categories, namely *user channels* and *restricted channels*. User channels are being used to communicate to the ECU in normal operation mode, e.g. communication interfaces to the in-vehicle network or (wireless) communication interfaces to external networks and devices. Restricted channels on the other hand are being used for specific, and typically privileged, operations such as device debug, test and maintenance.

An attacker could try to use any of these interfaces as attack vector. He could, for example, try to listen into the communication (eavesdropping) or to manipulate it (replay messages, insert fake messages etc.). It is therefore important to consider all interfaces/channels and the protocols used thereon as being part of the attack surface.

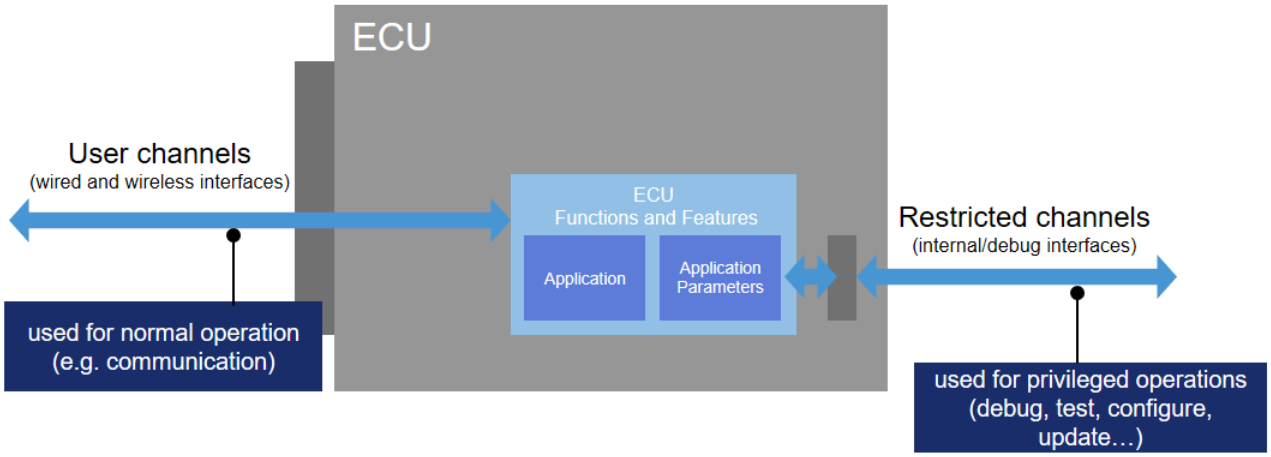


Figure 2: ECU interfaces

OBD-II

One interface that deserves specific attention, is the on-board diagnostics connector (OBD-II)[1]. It was originally designed as a normal user port to enable emission checks by regulators like the EPA and CARB. Over time, the function of the port was extended to provide a wealth of detailed diagnostics on the internals of the vehicle network that are used for inspection and maintenance. In its current form, it can be misused by hackers to manipulate the vehicle network and/or to extract data from it. The SAE Data Link Connector Vehicle Security Committee (TEVDS20) is therefore currently working on specifications (J3138 and J3146) that will help to turn the diagnostics port into a (more) restricted port with limited capabilities.

Implementation

The attack surface of ECUs is larger than only the interfaces. Information can also leak via unintended channels which are typically called 'side channels'. These side channels can be used to observe the internal behavior of the ECU or a component of it. Typical state of the art side channel attacks include timing analysis, static and dynamic power analysis (SPA/DPA), electromagnetic analysis (EMA) and photo emission analysis.

[1] The official name for the diagnostics port is Data Link Connector (DLC)

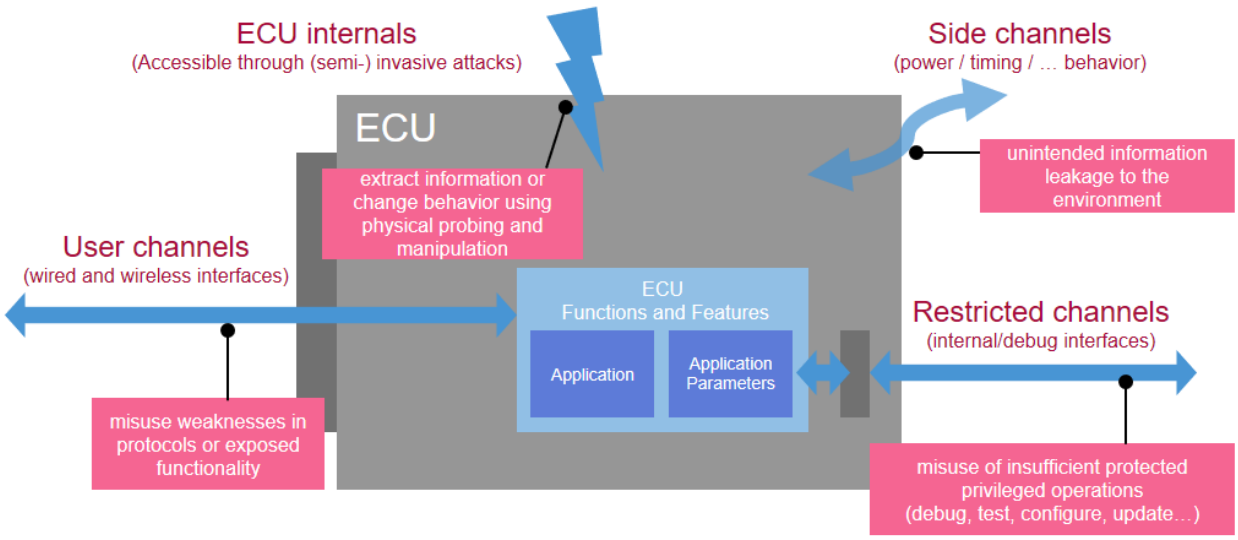


Figure 3: The attack surface for an ECU

Furthermore, hackers may try to gain access, either directly or indirectly, to the device's internals. This could be achieved e.g. by executing an *invasive attack* on the ECU or its components. Prominent invasive attacks that an attacker may mount include (micro) probing and reverse engineering of the ECU or its components. The latter category applied on ICs generally requires deep and advanced know-how and tools, and thus, is harder to achieve in practice. Nevertheless, it may be a first step that allows a hacker to identify vulnerabilities that can later be exploited remotely. Figure 3 summarizes the attack surface of an ECU.

Attack vectors at various layers

Attack vectors can exist at various levels of an ECU. An attacker can attempt to analyze and manipulate the ECU through its external interfaces, but also through interfaces that exist at various layers inside the ECU. For example, he can use connector(s) and test pins on the printed circuit board (PCB), the pins and pads of an IC, or the interfaces (APIs) of the software that runs on the microcontrollers and processors. All these interfaces together provide the attacker with a large set of potential attack vectors to choose from.

The same holds for semi-invasive and invasive attacks, which can also be mounted at various layers inside an ECU. For example, a hacker could apply electromagnetic analysis (EMA) at printed circuit board (PCB) level, or at a more fine-grained level by measuring the activity in a co-processor inside an IC. He may also want to physically tamper with the PCB, e.g. by replacing or removing some of the components on it. Or he could even choose to go one step further and tamper with the IC, e.g. by using a Scanning Electron Microscope (SEM) to reverse engineer (part of) the IC design, or by applying fault injection using a Laser or a Focused Ion Beam (FIB).

Exploiting vulnerabilities

Hackers will, generally, take the 'path of least resistance' to achieve their objectives. In other words, they will use the attack vectors that provide them the easiest way to e.g. extract assets (information, crypto keys, …) or to modify the system's operation.

Figure 4 shows a few typical hacker objectives for using various types of attacks. Let's now see how a hacker goes about achieving his objectives.

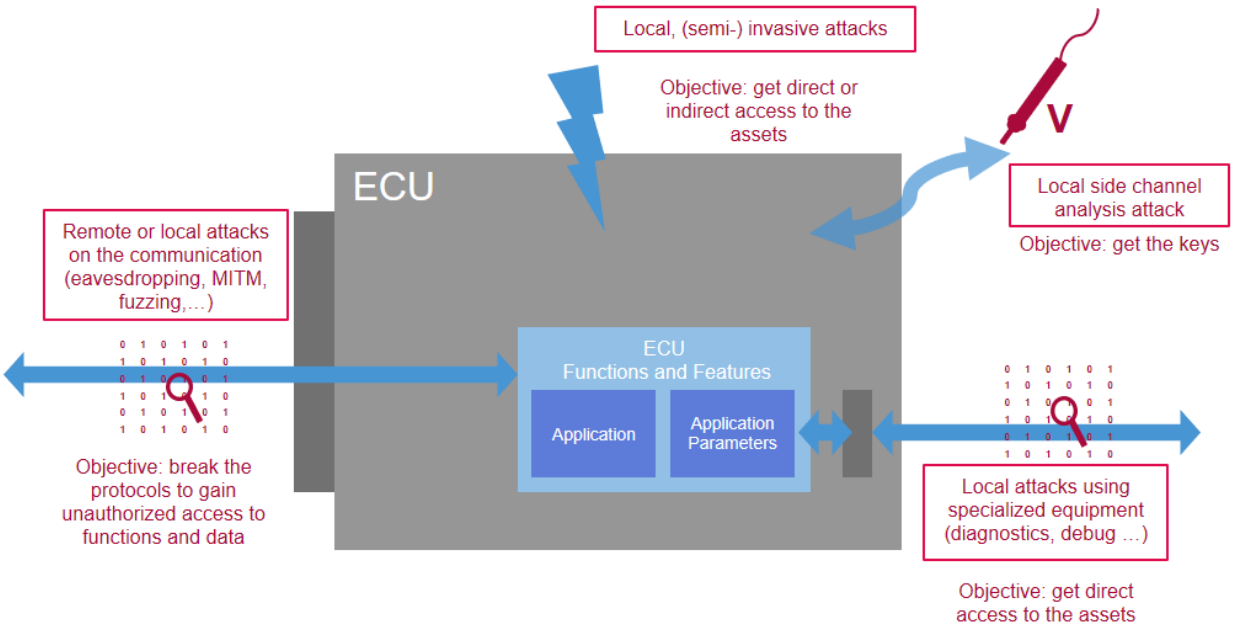


Figure 4: Attack Vectors and Related Objectives

Attack phases: identification & exploitation

Every successful attack consists of two phases: the *identification* phase is the first phase, in which a hacker identifies a vulnerability in a system. In the second phase, the *exploitation* phase, the hacker exploits this vulnerability, to achieve his goal of gaining access to the assets he was after.

The identification phase is typically the one that takes most time. Especially when the hacker doesn't have a lot of upfront knowledge about the targeted system, it may be a tedious process of trial and error to find vulnerabilities. The exploitation phase is often much easier.

Both phases may be local, or remote, but it is also possible to have a combination of both. For example, an attacker may initially have local access to the device and then identify a vulnerability that can be remotely exploited on (all) similar devices.

Let's make this more tangible using two (hypothetical) examples:

- Let's assume a vehicle model would use shared secret (AES) keys for over-the-air (OTA) software updates. In other words, every vehicle uses the same secret AES key to verify and/or decrypt new images before they are applied. In case an attacker would be able to extract such key using a local attack (e.g. using side-channel analysis), it could be used to remotely replace the firmware of multiple vehicles.
- A vehicle that is equipped with Vehicle-to-X (V2X) communication has several ECC key pairs that are used to authenticate the messages it broadcasts. When a vehicle identity (ECC private key) would be extracted using a local attack, it could be used to send false messages to other cars, which seem to be valid (trustworthy) as the key that was used to sign the message, is valid. As such, it may be possible to disrupt traffic or to takeover an identity and its associated permissions.

Exploit chains

Most security hacks, whether targeted at cars or consumer goods like smartphones, link up several smaller exploits. The Jeep hack of 2015 is a famous example: after (physical) reverse engineering the vehicle, Miller and Valasek linked up exploits that took advantage of weaknesses in the external network, the TCU and the programming interface of a device on the CAN network, to finally take full remote control over the vehicle.

A similar term that is also sometimes used in this context, is the term *kill chain*. It was originally used as a military concept related to the structure of an attack. More recently, Lockheed Martin adapted this concept to information security, using it as a method for modeling intrusions on a computer network.

Security functions and countermeasures

We already mentioned that we need to protect the assets from being compromised, to prevent losses. We've also elaborated that there are many possible attack vectors, and many techniques, that a hacker may use to achieve his objectives. It is usually impossible to protect against all possible attacks. And this may also not be needed, as some attack vectors may be unlikely to be exploited in practice, e.g. because the costs for doing so are too high in comparison to the value of the asset. One must therefore make a risk assessment to determine which assets to protect, and against what kind of attacks.

Based on this assessment, one needs to apply appropriate countermeasures at various levels of the system. Figure 5 shows a high-level overview of the countermeasures that can be applied at various levels in the ECU.

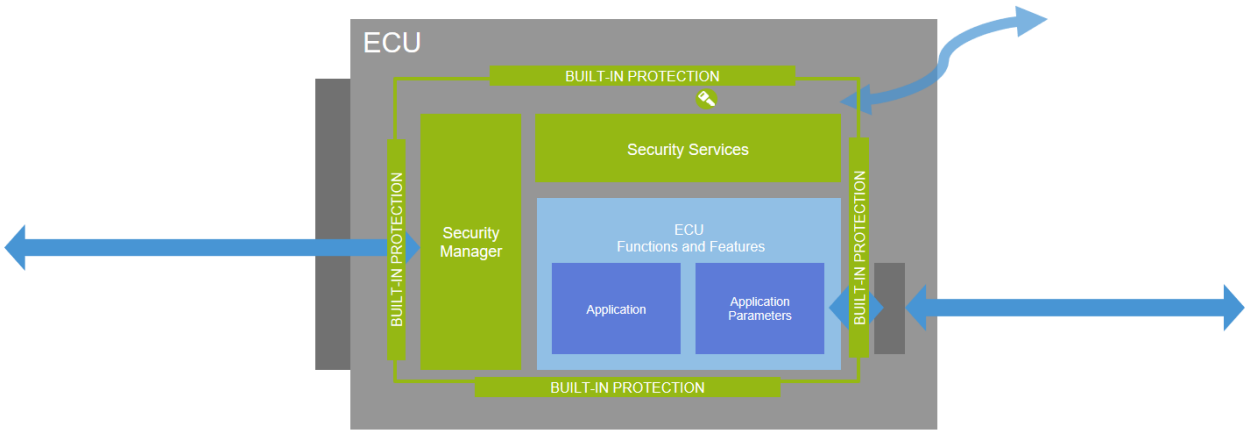


Figure 5: Protecting the assets

The security manager

The security manager is the first line of defense to prevent malicious access to the ECU functions and features. This system component operates a high-level and feature-rich security stack within the application space, that links the ECU with in-vehicular and/or extra-vehicular networks. It operates the high-level security protocols that supports the overall security architecture defined by the car maker.

For instance, it authenticates the CAN frames received and sent-out on the CAN bus; it runs the Transport Layer Security (TLS) to connect with a security back-end; it manages the security policies within the ECU; it secures the download of external code and data to be used by the application.

The security services

The security services form a lower-level system component that runs the underlying primitives required at the level of the security manager, in an execution context separated from the application. On top of performing the necessary cryptographic operations, it provides a set of services to manage the cryptographic keys, to enforce hardware countermeasures such as the secure boot mechanism, or to operate built-in self-tests.

Moreover, it isolates the cryptographic keys from the security manager and the rest of the application: this is the only component within the system being able to read-out and use the keys in the cryptographic primitives, thereby protecting those keys from application-level software attacks.

The built-in protections

Built-in protections are a third security component that wraps around the security manager and the security services to further protect them against specific security threats. They take many forms and integrate in various points within the system they protect. They can be very simple, such as a password protection that restrict access through the debug or test port, or very complex, such as sensors detecting voltage glitch on the power

line, shields that protect the device from invasive micro-probing, or specific circuits that limit the leakage of cryptographic operations to the outside.

Built-in protections are distributed within the ECU and are not necessarily always operating within the chip that runs the ECU functions and features. They can, for instance, operate in companion chips, adding specific protection to the data to be processed by the application *before* it reaches it. CAN frame filtering or Ethernet frame inspections are two examples of this kind of protections.

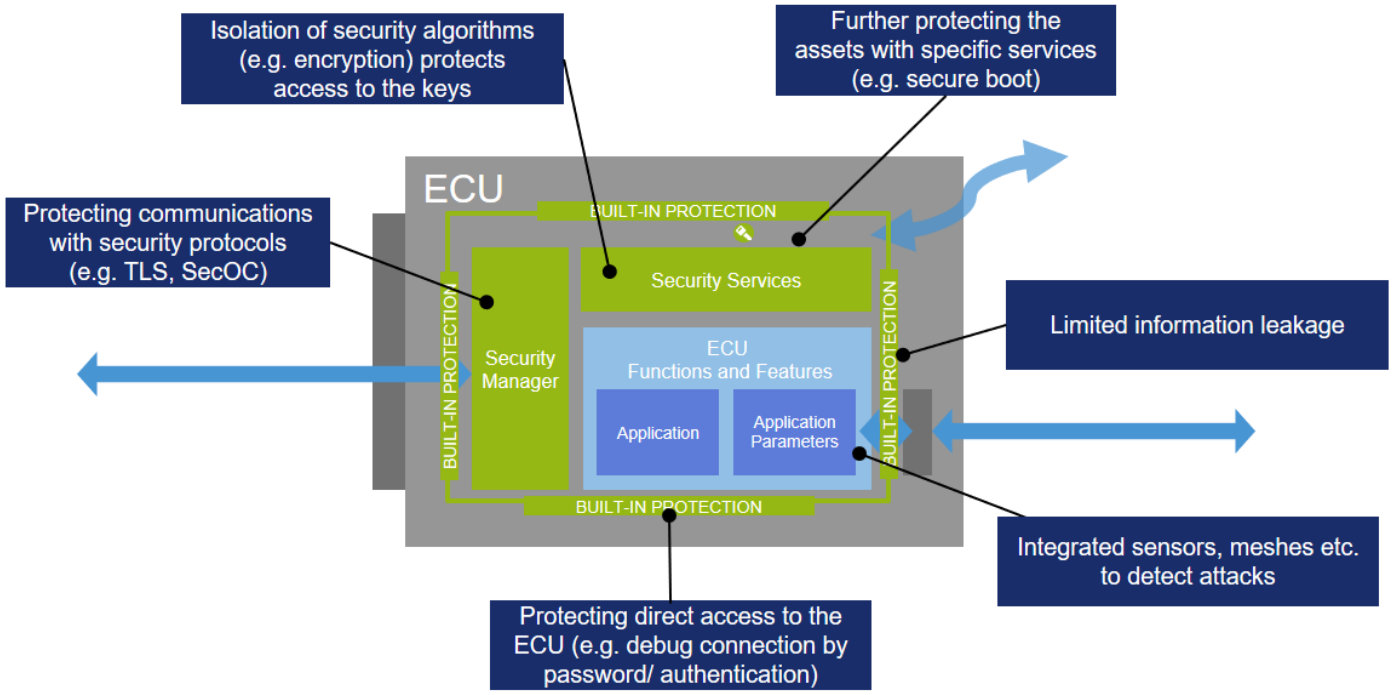


Figure 6: Examples for asset protections

Summary

In this chapter, we provided a high-level introduction to the protection of *assets* in ECUs, against cyber-attacks. We introduced the concepts of *attack vectors*, the resulting *attack surface*, and we've provided a high-level overview of some of the *countermeasures* that can be applied in an ECU. The following chapter explores these items in more detail in combination with the possible attack entry points.

Detailed Exploration

In the previous chapter, we touched upon some of the attacks that can be mounted against ECUs. In this chapter, we will explore the various attacks and attack vectors in a more structured manner – by dividing them into multiple attack categories, and by sketching at which levels in a system (ECU) they apply.

Level of access

What a hacker can do to a system, depends amongst others on the level of access that a hacker has, to the system. We distinguish here between the following four access levels:

1. Remote access – through the vehicle's external interfaces
2. Local access, up to the ECU – through its user and restricted interfaces
3. Local access, up to the board (PCB) level – through internal wiring, test points, etc.
4. Local access, at (or inside) the IC

Figure 7 depicts these access levels. The more (fine-grained) access a hacker has, the more he can, typically, do. However, the 'cost' for an attack/hack is, typically, tied to the level of access and know-how needed to perform them. For example: the cost (effort, time, know-how, equipment) needed for attacking an IC is higher than the cost for attacking an ECU. Therefore, it is more likely to see attacks at external (wireless) interfaces, than at the PCB or even IC level. However, it may still be worth to perform attacks at these levels, e.g. when the assets the hacker is after are of high value, or, when an attack exploitation scales to multiple devices.

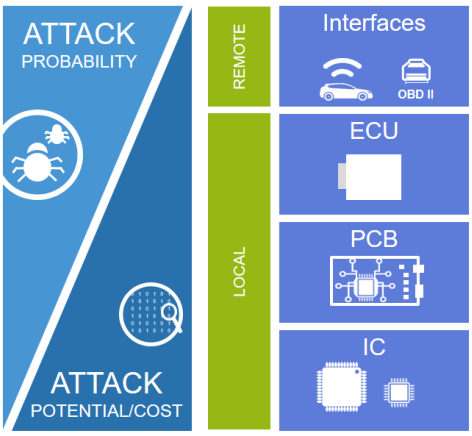


Figure 7: Access Levels

Attack classes

Let's now have a look at what attacks are possible at the various levels in an ECU. To structure this, we identified 5 main classes of attacks, which are also depicted in Figure 8:

- Attacks on communication
- Attacks on exposed functions
- Non-invasive attacks
- Semi-invasive attacks
- Invasive attacks

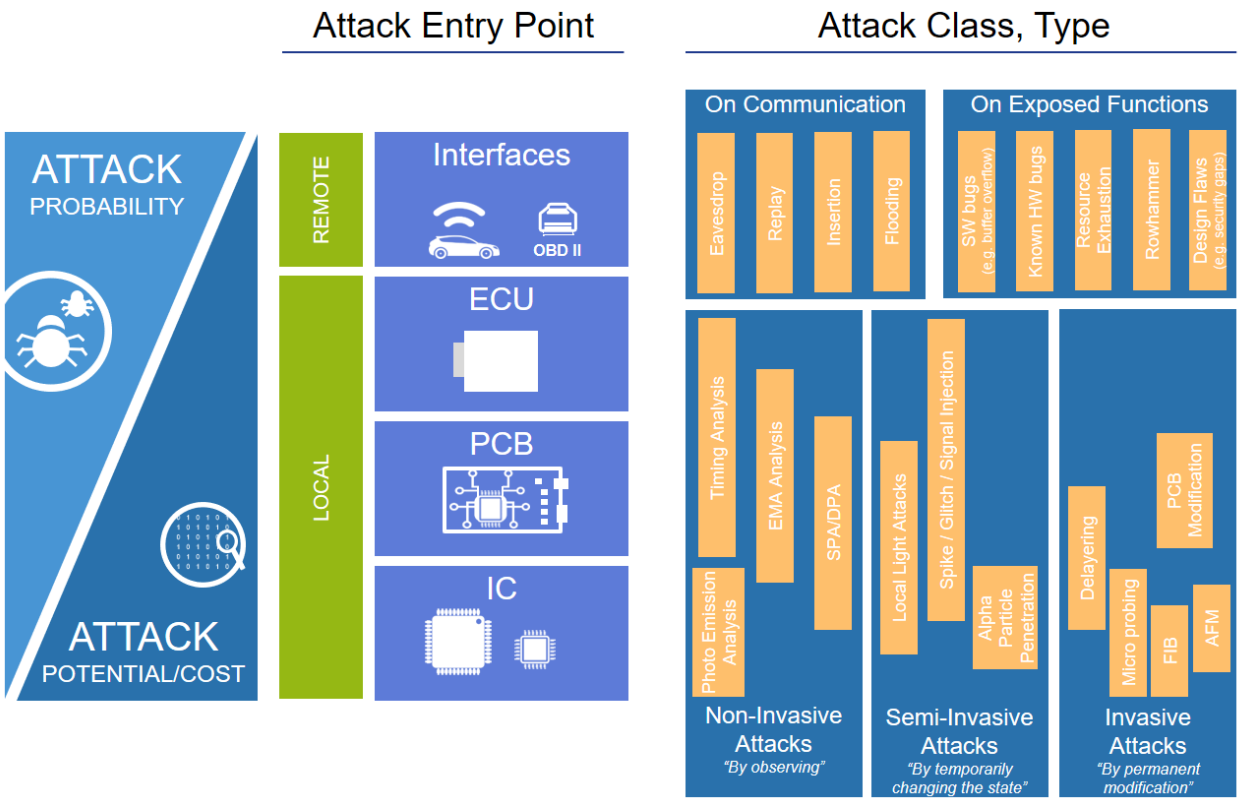


Figure 8: Attacks at various levels of an ECU

We will now have a more detailed look on each of these five attack classes.

Attacks on communication

These attacks can be applied by (passively) listening into, or actively participating in the communication between components in a system. For example, eavesdropping (i.e. listening into private communication), message flooding (a denial-of-service attack that exhausts the bandwidth in a channel and thereby blocks legitimate traffic) and various Main-in-the-Middle (MITM) attacks, such as message replay, insertion, or blocking.

Attacks on exposed functions

Another category consists of attacks that target those parts of the implementation that are exposed through the interfaces. For example, one could try to send messages in such a way that a protocol handler is unable to process them correctly: a weak implementation may not be able to properly handle messages that are received earlier or later than expected. Or, a weak implementation may have difficulties handling messages that are received 'out of order'.

Also, a hacker may try to manipulate the data (header, or payload) of messages, which may trigger unintended behavior in the implementation. For example, a weak implementation may not validate data before processing.

When the payload of a message is used as input to a program or a function, such maliciously crafted messages may trigger for example buffer overflows or stack overflows in the implementation.

Non-invasive attacks

An ECU, or its components, may leak information via unintended channels, called side channels, which can be used to observe the internal behavior of an IC or system. This category of attacks is therefore also known as side channel analysis (SCA) attacks. Examples of such attacks are timing analysis, static and dynamic power analysis (SPA/DPA), electromagnetic analysis (EMA) and photo emission analysis. For a good overview of the history on SCA, and its impact on cryptographic modules, see [3].

Semi-invasive attacks

The class of semi-invasive attacks encompasses those attacks that temporarily change the state of the ECU, or its components. For example, a hacker may introduce extra charge inside an IC using a flash light or laser beam (light attacks), thereby flipping bits from 0 to 1, or vice versa. Alternatively, a hacker can introduce glitches in the supply voltage, or on the input signals, which may 'ripple through' the design and cause instabilities, ultimately resulting in changed and unintended behavior.

Invasive attacks

This last category is the type of attacks that (permanently) change the design of the system. In other words, they (permanently) change the implementation, for example, through changing the interconnects on a PCB, or even by changing the circuit inside an IC using a focused ion beam (FIB).

Terminology

Figure 9 shows some other terms that are also often used in relation to the types of attacks that were elaborated in the previous section.

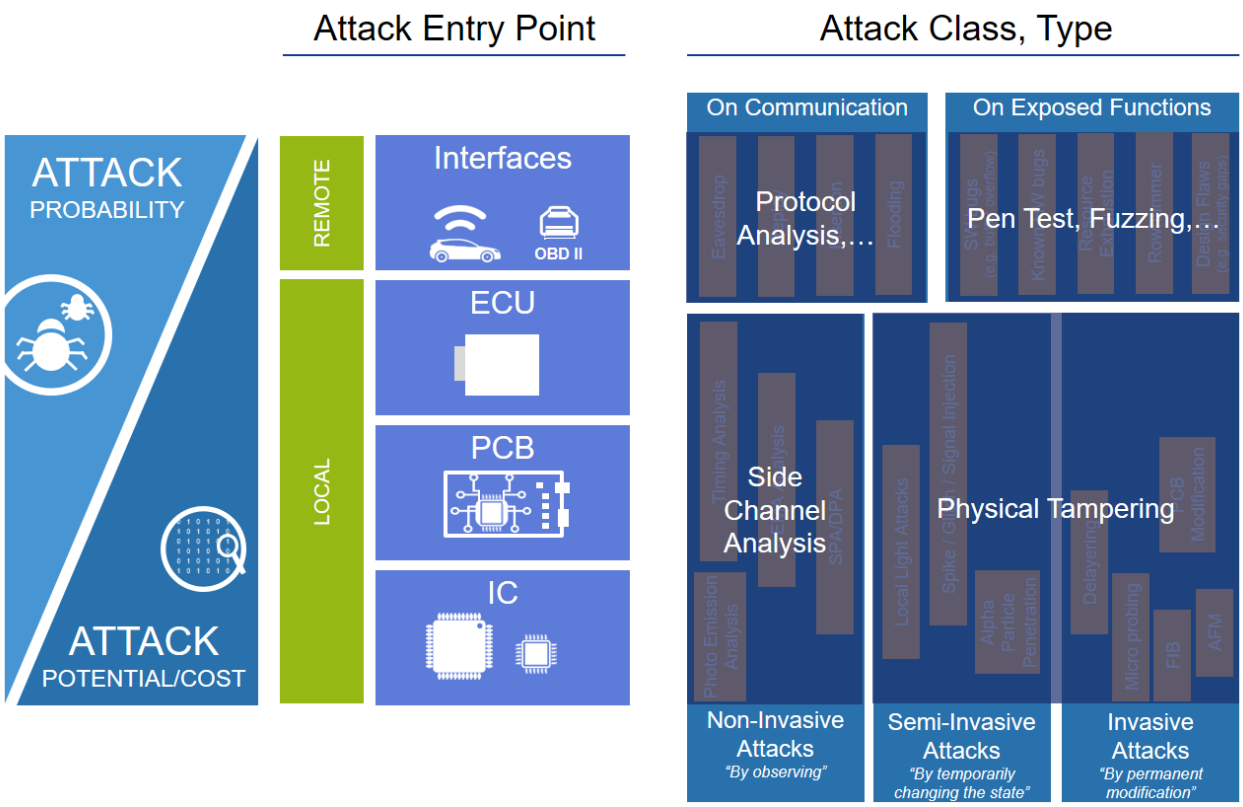


Figure 9: Attack Types – Terminology

Likelihood of attacks

How likely (or probable) it is that we see a certain attack in real life, depends on the balance between the cost that is needed to perform the attack, and the benefits it brings – in other words, it depends on the expected Return on Investment (ROI) for the hacker, for that specific attack, and to that specific system.

Cost (Attack Potential)

In Common Criteria, an international standard for computer security certification, the *attack potential* for a certain attack on a Target-of-Evaluation (ToE) expresses how much 'potential' an attacker needs to perform a certain attack – an easy attack requires low 'potential', whereas a difficult attack requires high 'potential'. This attack potential is expressed in terms of:

- Elapsed Time – i.e. how much time does an attacker need to successfully mount this attack on the ToE?

- Expertise – i.e. is basic know-how sufficient, or does one need to be an expert?
- Knowledge of the ToE – i.e. how much does one need to know about the design and operation of the ToE?
- Opportunity – i.e. what level of access does one need to the ToE, to how many instances of the same device, and for how long?
- Equipment – i.e. what (advanced) equipment is required?

All five factors need to be evaluated, for both phases of an attack (identification and exploitation), to come to a quantification of the required attack potential.

Benefits

The benefits depend on the value of the assets. Also, it depends on whether the exploitation of an attack is 'scalable', i.e. whether it can easily be applied to multiple devices.

In general, it is hard to quantify the benefits for specific attacks. For example, what is the value of stolen data? Or the value of publicity, e.g. for researchers?

As such, it is also hard to quantify the likelihood of an attack. Nevertheless, we can make some generic statements to get a view on the likelihood that we see certain attack types in the field.

Access entry points

Let's have a closer look at the level of access on the ECU that is required for various attack types.

These first two categories of attacks, those on communications, or on exposed functions, can be performed on (or through) the interfaces of the ECU, but potentially also on interfaces of its components. In the case of wireless interfaces, it may also be possible to perform such attacks remotely, and potentially, on multiple devices at the same time (i.e. the attack is 'scalable'). As such, the benefits of such attacks are, potentially, large. And thus, the likelihood that we see such attacks in the field, is high.

The latter three categories of attacks, on the other hand, typically require local access to the device. Clearly, this is the case for physical modifications (invasive attacks) and for semi-invasive attacks such as light attacks. Also, most side channel attacks (non-invasive attacks) require proximity to the device. The only notable exception may be timing analysis attacks, which may in certain cases be possible remotely (see [4]).

So, in general, the latter three categories of attacks are more difficult to perform. This does, however, not mean that they are, in practice, 'too difficult' (or, too costly). Certain types of attacks, e.g. glitch attacks and timing and power analysis attacks, are getting more and more easy to perform, also for 'newbies'. For example, devices like the ChipWhisperer[2] (cost range: ~ 200$) are significantly lowering the bar for hackers to apply such attacks. This makes such attacks more easily profitable, and therefore more likely, as the value of the compromised asset more easily outweighs the cost of such attack. And the likelihood increases even further, when information that can obtained using such attack, could be used to target a larger fleet.

[2] See https://newae.com/tools/chipwhisperer/

Countermeasures

Now that we understand the types of attacks that can be applied to an ECU, it's time to move on to possible countermeasures. Figure 10 shows several example countermeasures that can be applied. It is important to note that it is, generally, not possible to apply a 1:1 mapping between the attack examples depicted in Figure 8, and the countermeasures presented here: some countermeasures may cover multiple attack types whilst resistance against certain attacks can only be achieved when multiple countermeasures are applied. Also, this is not an exhaustive list of countermeasures, but rather an indication (by example) of the types of countermeasure that can be applied per category of attacks.

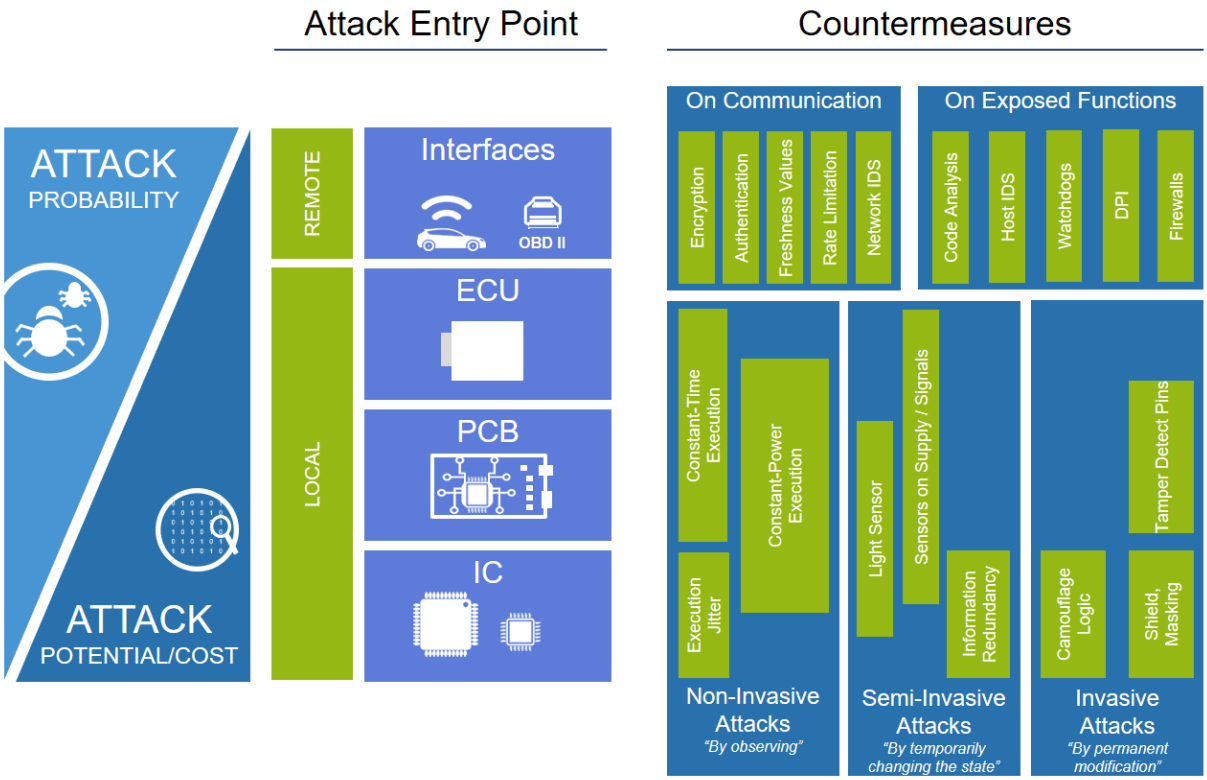


Figure 10: Countermeasures (examples only)

Devices connected to external networks – either directly, or indirectly via e.g. a gateway – could be attacked remotely and thus require protection against such remote attacks. The type of countermeasures shown in Figure 10, that can be applied against attacks on the communication or on functions exposed via interfaces, are similar the ones we know from the IT security world.

Many (embedded) connected devices are residing in an unmonitored and hostile environment. This is clearly the case for vehicles that are parked in public spaces, or for vehicles that are brought to a workshop that (illegally) tunes the vehicle's engine performance. As such, the opportunity for attackers is larger (see section 'Likelihood of attacks'), which increases the need to provide additional protection against more advanced, local attacks.

Protection levels

This section provides an overview of the protection levels for vehicle ECUs, as well as the protection levels that are offered by typical semiconductor products that are available in the market today. Note that we only indicate which *types* of attacks may be in scope. What is not depicted, is the *level of resistance* against such attacks, as needed by applications, or as offered by solutions.

Security needs, per vehicle domain

The exact scope of attacks to be considered for a particular ECU as well as the resistance levels against these attacks, is strongly dependent upon the assets that are implemented on that ECU. We can nevertheless give an indication of the required security level for an ECU, depending on the system domain it is part of.

Let's start with having a look at NXP's five system domains for vehicles, interconnected by the vehicle network, as depicted in Figure 11:

- Experience – i.e. all systems that enhance the user experience, for example navigation and infotainment systems
- Body & Comfort – i.e. systems such as automatic climate control, automatic windscreen wipers, electronic seat adjustment, etcetera
- Powertrain & Vehicle Dynamics – i.e. the engine control unit, ABS, ESP, etc.
- Driver Substitution – i.e. the systems that autonomously operate the car
- Connectivity – the communication units that provide the (wireless) connections to the outside world, over cellular connections, WiFi, Bluetooth etc.
- Vehicle Network – in particular, the part of the network (including active components such as gateways) which interconnects all the five domains

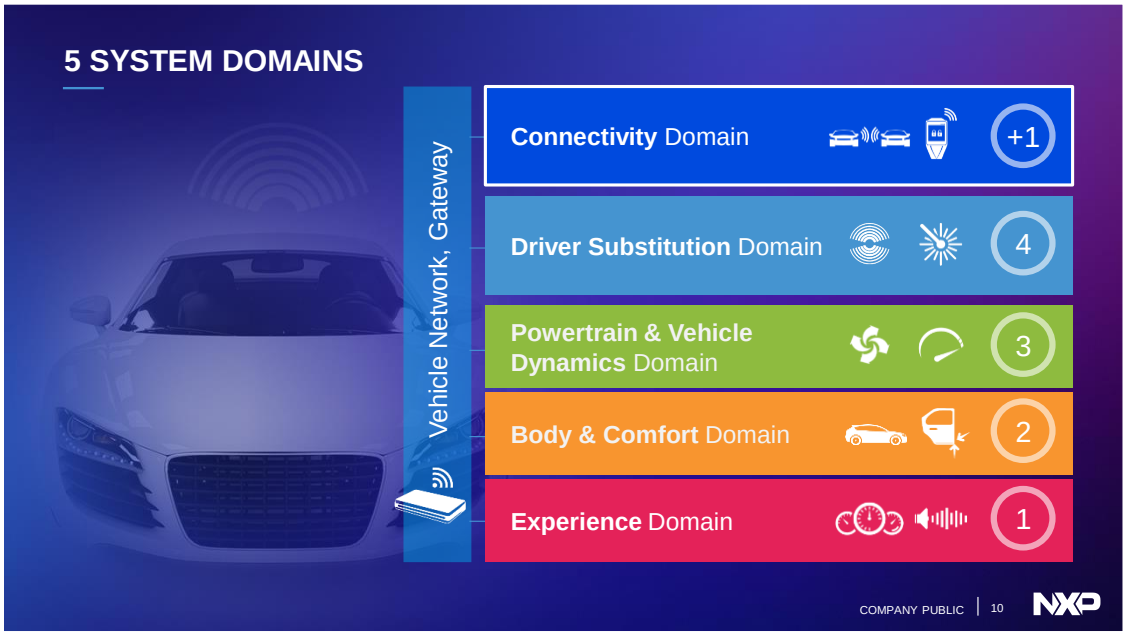


Figure 11: Five system domains in vehicles

Clearly, these five domains have different criticalities. But the systems (ECUs) *inside* a single domain have similar criticalities. And therefore, we can give generic guidance on the types of attacks that may be in scope for each of these domains. Such guidance is provided in Figure 12:

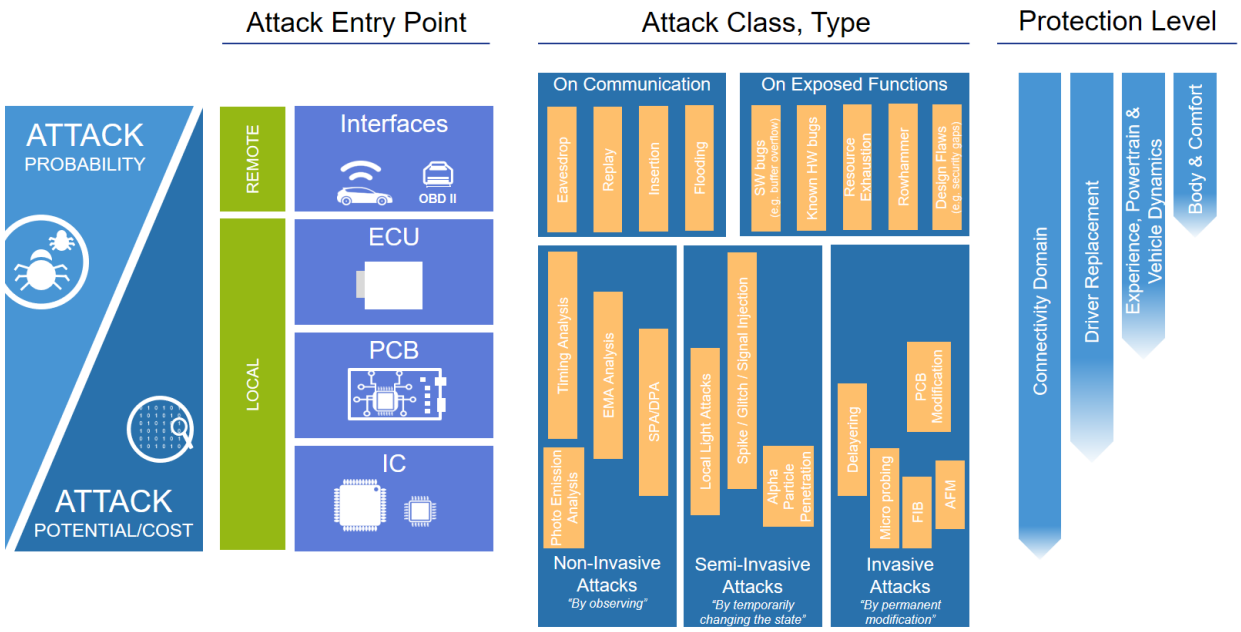


Figure 12: Required protection level – per vehicle domain

As can be seen, the highest level of protection is needed for (some of the systems in) the connectivity domain. Main reasons for the high security needs are that:

- There is a *high incentive*: this domain forms the entry point to the vehicle – and vice versa, it provides access to (valuable) cloud services and infrastructure. Hence, hackers will, generally, have a high interest to attack these systems, e.g. to extract the credentials (crypto keys) and gain (unauthorized) access to map updates or other cloud services.
- There is a *large(r) opportunity*: these systems (modems) are generally mounted close to the perimeter of the vehicle – for example, in a plastic 'shark fin' on the rooftop. Hence, they are more easily accessible – a hacker may not need to have access to (i.e. unlock) your vehicle to gain access to these systems. This opens the door for local (and more advanced) attacks.

A similar level of protection is needed for a gateway, as it provides, by nature, access into all five domains. Being a central component in the vehicle, it is furthermore used to implement central control functions such as managing firmware updates for the rest of the vehicle network – i.e. for ECUs inside the five domains. If such central control function would be compromised, the entire vehicle network may be affected.

Second highest requirements are there for the Driver Substitution domain. These are large and complex systems that (partially or sometimes even fully) take over control over the vehicle. As such, the safety of the passengers fully depends on the correct functioning of these systems. It speaks for itself that this leads to high security requirements.

Next are then the Experience domain and the Powertrain and Vehicle Dynamics domain. The risk of an attack against these systems may be slightly lower than for the Driver Substitution domain, but it is still significant. For systems in the Experience domain this is because they may be directly connected to external devices, such as smartphones (via Bluetooth or WiFi), which increases their attack surface. For the systems in the Powertrain and Vehicle Dynamics domain the attack surface is smaller (no direct, wireless connection). Nevertheless, the risk is significant as these systems are safety-critical: they contribute to the overall safety of the vehicle, and therefore, the safety of the passengers depends on the correct functioning of these systems.

Incorrect functioning of the systems in the last domain, Body & Comfort, is inconvenient, but not necessarily safety-critical. Hence, the security requirements for these domains can be relatively relaxed.

The above is a guidance, but not a blueprint, for ECU security. As mentioned, the applications or functions that are implemented on the ECU ultimately determine which attacks are in scope for a specific ECU, and what resistance level is required against these attacks. Also, the fact that other countermeasures have (or have not) been applied elsewhere in the vehicle's E&E architecture may influence the actual requirements per ECU.

Security targets, per product type

Now that we have a better feeling of the level of protection that is needed, let's have a look at what is offered by typical products that are available on the market today. We can, globally, distinguish between three types of products with increasing levels of protection:

- Standard microcontrollers and processors
- Microcontrollers and processors with increased attack resistance
- Secure Elements

These product types, and their approximate level of protection, are depicted in Figure 13.

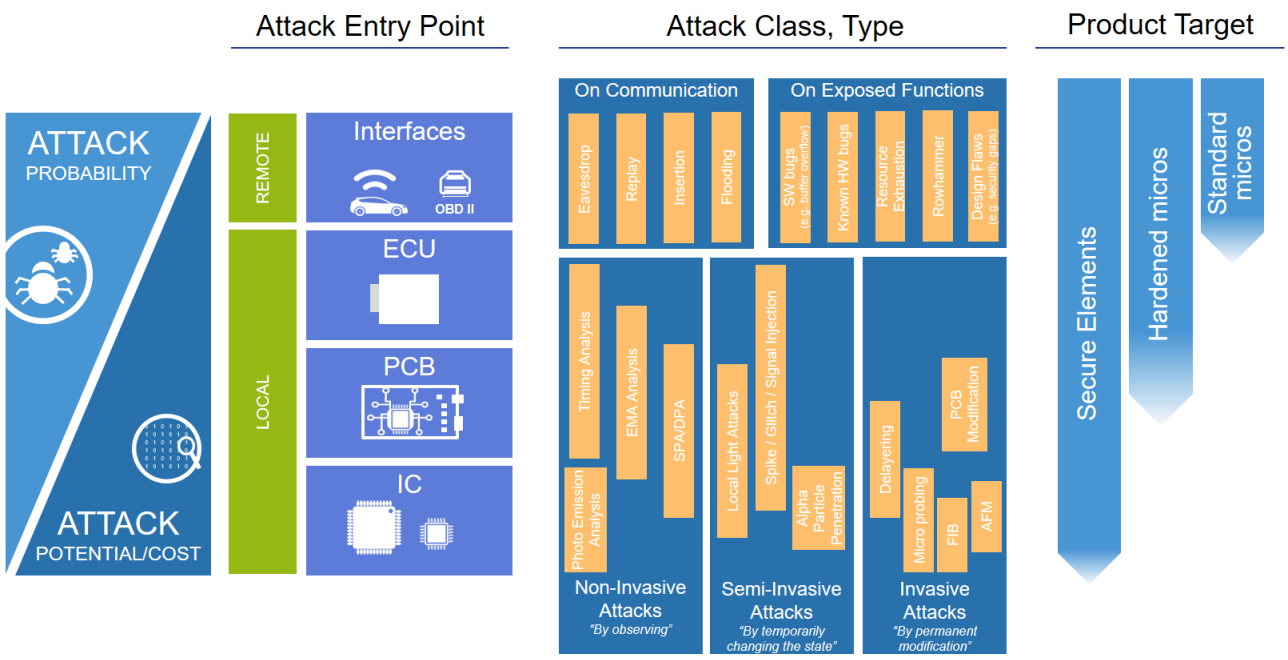


Figure 13: Targeted protection level – for different product types

Standard microcontrollers and processors, i.e. most of the general-purpose micros that are available in the (automotive) market today, even if some of these micros may feature a dedicated (e.g. SHE-compatible) hardware security subsystem, target, at best, remote attacks. In other words, they assume that a hacker has no local access to a system and therefore, do not protect the system against attacks such as side channel analysis or physical tampering.

There are certain microcontrollers and processors that offer protection (i.e. are hardened) against certain local attacks. For example, products in NXP's S32 microcontroller family implement countermeasures against certain side channel and fault injection attacks that are becoming easier to perform today.

Secure Elements (SE), finally, are highly tamper-resistant devices that are using technology that originates from the smart card industry. They have traditionally been used for applications like payment, banking and electronic identification (eID) and are now also considered for M2M authentication including e.g. V2X and car-to-cloud connectivity. As this M2M authentication is used to provide trust in the other entity, and when successful, may result e.g. in providing the other entity with access to the vehicle or cloud service, this function and the secrets involved need to be well protected.

Conclusions

There are many ways in which someone can try to attack an ECU. To bring structure into this attack landscape, we have introduced a framework that clusters these attacks in the following five attack groups: attacks on communication, attacks on exposed functions, non-invasive attacks, semi-invasive attacks and invasive attacks. This categorization of the attacks makes it easier to reason about the attacks in terms of which protection level is in scope for a certain ECU, and how to protect against such (classes of) attacks.

One of the crucial factors that determines the attacks in scope, is the level of access that an attacker has to the ECU. For example, several attack classes are (generally) out of scope when local (physical) access to the ECU is not possible. On the other hand, when a hacker has fine-grained access to the ECU and its components, the list of potential attacks rapidly expands.

The NXP system domain model divides ECUs in the vehicle in five categories: experience, body & comfort, powertrain & vehicle dynamics, driver substitution and connectivity. All ECUs inside one domain implement a similar type of function inside the vehicle network – and the criticality (from a security and safety point-of-view) for the various ECUs within a single domain is therefore rather similar. As such, we can use this system domain model to provide a guidance for the level of protection needed for ECUs.

By bringing both categorizations together, we can finally provide generic guidance on the types of attacks that may be in scope for a specific ECU – as well as the type of silicon solutions that provide the right security features, functions and levels to protect against such attacks.

NXP has leveraged its strong leadership positions in the automotive and smart card markets to build an industry-leading portfolio of automotive security solutions that provide the right security basis for securing *any* ECU – regardless of the specific security requirements for that particular ECU. This portfolio comprises microcontrollers and processors with integrated security subsystems, secure elements as well as security solutions for CAN and Ethernet networks.

Definitions

Attack surface	The sum of the different points (the "attack vectors") where an unauthorized user (the "attacker") can try to attack a system.
Attack vector	The path or means by which a hacker can gain (unauthorized) access to a system.
Common Criteria	A framework that provides customers assurance that a product's security attributes can be trusted and that the customer's security needs are protected. As the basis for the international standards ISO/IEC 15408 and ISO/IEC 18045, Common Criteria provides assurance that the process of specification, implementation and evaluation of products has been conducted in a rigorous, standard, achievable, repeatable and testable manner at a level that is commensurate with the target environment for use.
Cyber-physical system	A system of collaborating computational elements controlling physical entities
Cybersecurity	All processes and mechanisms by which digital equipment, information and services are protected from unintended or unauthorized access, change or destruction. It can for example be a combination of physical security, information security, standards, legislation, policies, and risk mitigation strategies.
DPA	Differential Power Analysis
E&E architecture	The Electrical and Electronic Architecture of a vehicle. In other words, the electric and electronic features of vehicles, implemented using ECUs that are inter-connected using In-Vehicle Networks.
ECU	Electronic Control Unit. An embedded system that controls certain functions in a vehicle.
EMA	ElectroMagnetic Analysis
EVITA	EVITA was a project co-funded by the European Union within the Seventh Framework Programme for research and technological development. Its objective was to design, verify, and prototype an architecture for automotive on-board networks where security-relevant components are protected against tampering and sensitive data are protected against compromise.
HIS	Herstellerinitiative Software (German for 'OEM software initiative') is an interest group consisting of the car manufacturers Audi, BMW, Daimler AG, Porsche and Volkswagen. This group created the SHE specification.
HSM	In the Automotive context, a Hardware Security Module is a security unit, typically integrated in a microcontroller, which can be used to protect software (secure boot, secure firmware update) and data (secure storage, secure communication). It typically consists of a programmable microcontroller, one or more hardware accelerators (e.g. AES, SHA2) and dedicated storage for crypto keys. The HSM specification can be interpreted as the successor of the SHE specification.

	This specification should not be confused with the definition that is commonly used outside the Automotive domain. There, this term is used for a tamper-proof physical computing device that is often used to safeguard and manages digital keys for and provides crypto-processing to a mission-critical infrastructure such as a public key infrastructure or an online banking application.
IDS/IPS	Intrusion Detection (and Prevention) Systems are systems that monitor network and/or system activities for malicious activity and report it (or attempt to block it). The detection mechanisms can vary widely, from simple network protocol analysis to advanced statistical anomaly detection based on machine learning.
Kill-chain	Originally used as a military concept related to the structure of an attack; consisting of target identification, force dispatch to target, decision and order to attack the target, and finally the destruction of the target. Conversely, the idea of "breaking" an opponent's kill chain is a method of defense or preemptive action. More recently, Lockheed Martin adapted this concept to information security.
MITM	Man-In-The-Middle
Physical attack	Attacks that can only be executed by an attacker with physical access to a system or an IC. Examples of physical IC attacks are fault injection attacks, micro probing, chip delayering, reverse engineering and side channel analysis.
PII	Personally Identifiable Information (PII) or Personal Data is information with the specific property that its disclosure or revelation conflicts (to a greater or lesser extent) with people's need for privacy. Such information is thus relating to one or more identified or identifiable natural persons and is describing one or more factors specific to their personal, physical, physiological, mental, economic, cultural or social identity or to their behavior, interests or whereabouts.
Remote attack	Attacks that are executed at a distance, typically via a network, by sending messages to exploit weaknesses in a system's design or its implementation (e.g. software bugs).
Restricted channel	An (intended) communication channels that is used to perform specific, and typically privileged, operations such as device test and maintenance.
SCA	Side Channel Analysis. A class of non-invasive attacks, in which the behavior of an IC or system is observed. Examples are timing analysis, static and dynamic power analysis (SPA/DPA), electromagnetic analysis (EMA) and photo emission analysis. Usually physical access is needed, although there have also been real-life examples of timing attacks against networked devices.
SE	A Secure Element is a tamper-resistant platform (typically a one chip secure microcontroller) capable of securely hosting applications and their confidential and cryptographic data (e.g. key management) in accordance with the rules and security requirements set forth by a set of well-identified trusted authorities. The management of applications on a Secure Element is usually done in accordance with the GlobalPlatform Card Specification.

SHE	The Secure Hardware Extension (SHE) is an on-chip extension to any given microcontroller. It is intended to move the control over cryptographic keys from the software domain into the hardware domain and therefore protect those keys from software attacks. It consists of a state machine, an AES accelerators and dedicated storage for crypto keys. It is not meant to replace highly secure solutions like TPM chips or smart cards, i.e. no tamper resistance is required by the specification. The SHE specification was created in 2008 by the HIS consortium and can be interpreted as a subset of the newer HSM specification.
Side channel	A path through which, unintendedly, information about the behavior of an IC or system leaks to the environment. An observer may use this information to learn more about the internals of a system or device, and eventually, may also be able to extract information (such as crypto keys) through such channels.
SPA	Static Power Analysis
Tamper-resistance	Resistance to tampering the device by normal users or systems or others with physical access to it. It ranges from simple features like screws with special heads to complex devices (e.g. ICs) which can withstand even the most sophisticated attacks.
Target Of Evaluation (TOE)	The product or system that is the subject of the evaluation
User channel	An (intended) communication channels that is used during normal operation of a system/device.

References

[1]	"Secure connected cars for a smarter world"; Timo van Roermund, NXP Semiconductors; Dec. 2016. Available at https://www.nxp.com/automotivesecurity.
[2]	"A multi-layer vehicle security framework"; Andy Birnie and Timo van Roermund, NXP Semiconductors; May 2016. Available at https://www.nxp.com/automotivesecurity.
[3]	"Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing"; YongBin Zhou and DengGuo Feng, Chinese Academy of Sciences; 2005. Available at: https://www.iacr.org/cryptodb/data/paper.php?pubkey=12722
[4]	"Remote Timing Attacks are Practical"; David Brumley and Dan Boneh, Stanford University; May 2003. Available at: http://crypto.stanford.edu/~dabo/abstracts/ssl-timing.html

About the authors

Timo van Roermund is leading the security team in NXP's Automotive business unit. He has deep expertise in applied security for embedded devices, such as Vehicle-to-X communication systems, in-vehicle networks, Internet-of-Things appliances, mobile phones and wearable devices. He is a regular speaker at international conferences and he actively contributes to security standards and industry consortia such as the Auto ISAC. Timo received the MSc degree in computer science and engineering from the Eindhoven University of Technology.

Andreas Bening is a system security architect in the Innovation Center Crypto & Security within NXP's business unit Security and Connectivity. He has extensive expertise in embedded systems, applied security and semiconductors with more than 20 years of R&D experience. Andreas contributes actively in the definition and creation of state-of-the-art and fit-for-purpose security solutions for connected devices. Andreas received his Dipl.Ing. degree in information and communication engineering from the Ilmenau University of Technology.

Fabrice Poulard is a system security architect within NXP's Automotive business unit. He has more than 20 years of experience in the semiconductor industry and embedded security, and more than 10 years in the marketing & development of cyber-security solutions in the automotive space. In his role in NXP, Fabrice contributes in defining cost-effective, comprehensive and easy-to-use solutions that can foster cyber-security adoption within the car industry. Fabrice holds a MSc in computer science (Paris University XI) and a MSc in system engineering (Paris University VI).

About NXP

NXP Semiconductors N.V. (NASDAQ:NXPI) enables secure connections and infrastructure for a smarter world, advancing solutions that make lives easier, better and safer. As the world leader in secure connectivity solutions for embedded applications, NXP is driving innovation in the secure connected vehicle, end-to-end security and privacy and smart connected solutions markets. Built on more than 60 years of combined experience and expertise, the company has 31,000 employees in more than 33 countries. Find out more at www.nxp.co

How to Reach Us:

Home Page: **www.nxp.com**
Web Support: **www.nxp.com/support**

USA/Europe or Locations Not Listed:

NXP Semiconductor
Technical Information Center, EL516 2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.nxp.com/support

Europe, Middle East, and Africa:

NXP Halbleiter Deutschland GmbH Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.nxp.com/support

Japan:

NXP Japan Ltd.
Yebisu Garden Place Tower 24F,
4-20-3, Ebisu, Shibuya-ku,
Tokyo 150-6024, Japan
0120 950 032 (Domestic Toll Free)
www.nxp.com/jp/support/

Asia/Pacific:

NXP Semiconductor Hong Kong Ltd. Technical Information Center
2 Dai King Street
Tai Po Industrial Estate Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@nxp.com
www.nxp.com

www.nxp.com/automotivesecurity

Date of release: November 2017