

Adapted Development Process for Security in Networked Automotive Systems

Karsten Schmidt
Audi Electronics Venture GmbH

Peter Tröger
Hasso Plattner Institute

Hans-Martin Kroll
FTM, Technische Universität München

Thomas Bünger
Hasso Plattner Institute

Florian Krueger
Audi AG

Christian Neuhaus
Hasso-Plattner-Institut

ABSTRACT

Future automotive systems will be connected with other vehicles and information systems for improved road safety, mobility and comfort. This new connectivity establishes data and command channels between the internal automotive system and arbitrary external entities.

One significant issue of this paradigm shift is that formerly closed automotive systems now become open systems that can be maliciously influenced through their communication interfaces. This introduces a new class of security challenges for automotive design. It also indirectly impacts the safety mechanisms that rely on a closed-world assumption for the vehicle.

We present a new security analysis approach that helps to identify and prioritize security issues in automotive architectures. The methodology incorporates a new threat classification for data flows in connected vehicle systems.

CITATION: Schmidt, K., Tröger, P., Kroll, H., Bünger, T. et al., "Adapted Development Process for Security in Networked Automotive Systems," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 7(2):2014, doi:10.4271/2014-01-0334.

INTRODUCTION

Future road vehicles will be connected with other vehicles and information systems to improve overall system functionality, driver safety, mobility and comfort. The more vehicles become externally connected, the more innovative features can rely on such external data input. However, these systems still must fulfill the established safety standards.

Safety is the primary design goal in automotive design and engineering. The necessary design procedures, analysis approaches, and evaluation methods are widely established in the automotive domain and follow commonly agreed principles.

In contrast to this, security had a less prominent role in automotive systems in the past. Classic security problems targeted here were theft protection, odometer integrity and secure remote unlocking [34].

Allowing vehicle systems to communicate with other systems that are not within their physical boundaries imposes a previously non-existing security problem. Any external communication with the vehicle system must be considered as a potential security threat, which may impact the system functionality or any of the safety properties. When internal

systems are directly or indirectly opened for external access, a safety mechanism that was designed with a closed-world assumption must be reevaluated with security in mind.

Another problem is the continuously changing set of vulnerabilities for a given vehicle, which also increases with the rising dependence on software-implemented functionality. Any progress in cryptographic research, any new vulnerability assessment of widely sold hardware components, or any new attack capability due to advanced technologies (e.g. wireless senders) may compromise the security of an existing and sold automotive system.

A first approach to target this problem might be to re-use knowledge from the field of computer systems security. However, Czerny [7] and others already identified that IT security and automotive security have significant differences. This is mainly reasoned by the close link between a vehicle system and its operational environment, while IT systems can be physically and virtually isolated. Examples are potentially tampered sensors or the physical access to the vehicle for an attacker. The computer world also implements sophisticated security update mechanisms, which allow the vendors to fix vulnerabilities quickly before their exploitation. This is, so far, not given in most of the established automotive design. There are also significant differences between IT and vehicle hardware, such as in the amount of computational resources or external bandwidth being available [35]. This makes vehicle security a different challenge in comparison to the IT world.

Facing the upcoming need for better security assessment, this paper proposes a new analysis approach tailored to the requirements of the automotive industry. The presented methodology involves a model-based analysis of data flow and qualitative assessment of threats. The methodology is added to the toolkit that is available in an ISO 26262-aligned development process. For evaluation of our proposed methodology, we discuss a practical automotive engineering use case.

The paper starts with a short definition of relevant terminology and a discussion of some existing work in this area. We then present the new methodological approach for security analysis and discuss the initial application in a car2x use case.

TERMINOLOGY

Safety and security research are known to be distinct fields in academia, although they share a lot of principles, concepts and mathematical foundations.

Safety is a critical aspect in automotive and other industries. It is commonly understood as one attribute of the more general notion of *system dependability*, which describes the trustworthiness of a system for the user, expressed through a set of non-functional attributes [22]. The most relevant attributes are:

- **Safety:** Protection against catastrophic consequences for humans or environment.
- **Reliability:** Capability to operate without failure for a continuous time period.
- **Availability:** Capability to operate at any given point in time.
- **Maintainability:** Level of support for fault removal and failure prevention.

Automotive research and standards typically use the notion of *functional safety*, which denotes the correct functioning of safety-related system components. Within the context of this paper, we interpret this as the *reliability of the safety mechanisms*, and use the safety definition given above.

Classic dependability research distinguishes between faults, errors and failures. A *fault* event represents the initial root cause for a dependability problem. Activated faults lead to an *error* state of the system, which then may lead to a perceptible deviation from the specified system configuration, a *failure* event. Some domains also use the *error* as synonym for a failure event. The concept of a system is recursively defined, meaning that the failure of a sub-system can be the originating fault for the overall system.

The standard for safety aspects in automotive systems is the ISO 26262 specification [14]. It defines the term *hazard* as potential source of *harm*, where harm is restricted to physical injury or damage to human life. The prevention of such hazards, which means more generally the prevention of safety failures, is the general goal in a safety-oriented system design.

Security is an abstract term that describes the resilience of a system in the face of intentional attacks. These acts can be active (e.g. manipulation, misuse) or passive (e.g. eavesdropping). Security comprises several distinct security properties. The most prominent taxonomy for them is the CIA-triad [3], combining *confidentiality*, *integrity* and *availability*. An alternative taxonomy is given by [24].

In the security domain, an *attacker* targets a particular *asset* of the system, such as interfaces, applications, or communication channels. Typical assets in the automotive domain are *Electronic Control Units* (ECUs), theft protection implementations for vehicles or components, vehicle bus components, or pay-per-use service implementations. Typical examples for attacks on such assets are: their replacement; the removal of their security parts; the eavesdropping, altering, injection, or replay of their internal communication data; or the modification of their firmware [34]. Attackers can be classified as clever outsiders, knowledgeable insiders, funded organizations, and funded organizations with insider knowledge [1].

A security *threat* is a description of who might attack an asset with a certain goal and probability. In contrast, *vulnerability* describes a specific weakness in security that could be exploited as part of an attack. Vulnerabilities do not directly

define threats, and threats do not impose a specific kind of vulnerability being exploited [18]. Both concepts must be considered in proper security analysis, but since vulnerabilities are easier to reason about, it is more efficient to target them first.

In order to have a clear separation of different security threats in our chosen problem domain, we distinguish car2x scenarios into *car-to-car*, *car-to-infrastructure* (car2i) and *car-to-mobile* (car2m) communication.

Risk is typically defined as combination of *severity* and the *probability* of occurrence. The failure probability of a component can be stochastically defined with established methods, however, the probability of an intended, malicious attack is difficult to grasp. There are neither sufficient statistics about security attacks against vehicles, nor accepted stochastic models for these kinds of attacks.

One solution for this lack of mathematical tools is the work by Wolf and Ward et al. [33], who proposed the concept of an attack potential value to express the difficulty of an attack. The *attack potential* is defined by the ISO 15408 [13] as part of a vulnerability assessment and should consider the elapsed time, specialist expertise, knowledge of the system, window of opportunity, and required equipment. The standard also defines multiple levels for each factor and numeric values for each level. These can be used to calculate the total attack potential. Wolf and Scheibel [36] slightly modify these values and give additional examples from the automotive domain. The *attack probability* is given by the reciprocal of the attack potential. A very difficult attack with a high attack potential corresponds to a low probability, whereas a high probability would be an easy attack with a low attack potential.

RELATED WORK

Most authors agree that security and safety are distinct concepts with some overlap. The main difference from an analytical point of view is the level of quantifiability.

Safety hazards can be investigated based on historical field and service data, accelerated lifetime testing, reliability modeling, and a general application of probabilistic methods. Influencing factors, such as environmental conditions, can mostly be described with mathematical models. This makes safety analysis a complicated and work intensive, but a mathematically and conceptually solid approach in the assessment of vehicle systems.

In contrast, security is often qualitatively described in terms of abstract threats. Related work introduces qualitative ranking schemes for vulnerabilities, threats, and potential security-reasoned damages [10]. Some work exists on attack and damage potential quantification, i.e. by Wolf and Scheibel [36].

Threats have a close relationship to timing and the precise chaining of different vulnerability exploitations. There are also some unique influence factors, such as the knowledge needed by the attacker, the required physical or virtual system access for the attacker, or the need for special equipment to break into the system. Related work from the automotive domain commonly agrees that a secure system design may not remain secure after several years due to newly discovered vulnerabilities, the increase in computational power, or the simplification of attack tools [35].

Some authors argue that security and safety control mechanisms may even work against each other. One example is the abuse of compute-intensive security mechanisms to bring safety-critical control systems down. A useful comparison criterion in the related work is the understanding of *failure impacts*, both in the safety and security world. Czerny [7] declares the following classes of negative consequences from a security failure:

- Safety properties of the vehicle are violated (e.g. disabled airbags).
- Privacy violation of the vehicle owner (e.g. leaked position data).
- Operational capabilities of the vehicle are harmed (e.g. broken radio).
- Financial revenue of the vehicle manufacturer or other parties is negatively impacted (e.g. warranty fraud).

This classification matches with other work [33], and consequently makes any system that has to be evaluated for safety conditions also relevant for security evaluation.

In 2011, Weimerskirch identified [34] four specific countermeasures for the commonly agreed security failures: electronic immobilizer in the vehicle key, vehicle unlock with rolling codes, component identification between the Electronic Control Units (ECUs), and secure software download with digital signatures.

The European EVITA project¹ conducted research on the architecture of on-board networks that protect security-related components against tampering. The European OVERSEE project² is focusing on an in-vehicle runtime environment for arbitrary software components, which is intended to ensure both isolation and real-time behavior. The secure implementation of car-to-car and car2i networks is investigated in the European PRESERVE project³. SeVeCom⁴ focused on the definition of proper security requirements and software designs for secure vehicle communication.

One typical concept in secure automotive architectures is a single *Hardware Security Module (HSM)*, sometimes also called *security anchor*, which implements security-enabling

1. <http://evita-project.org>

2. <https://www.oversee-project.com>

3. <http://www.preserve-project.eu>

4. <http://www.sevecom.org>



functionality such as encryption, secure key storage, secure hashing, and cloning prevention [35]. Due to negative experiences with closed proprietary encryption schemes [9], the industry meanwhile moved to standard cryptography algorithms in such systems [34]. Another relevant aspect is the proper management of digital certificates and keys [26].

Even though these projects and standards can contribute architectural blueprints for secure systems, they still demand the proper modeling and ranking of relevant safety and security failures in a new vehicle design. Such a model must be applicable within the ISO 26262 processes. Ward et al. provided a detailed discussion of how the ISO 26262 analysis methods could be extended for threat analysis and ranking [33], including the consideration of safety effects originating from security issues. One of the main contributions is an extension of ISO 26262 severity classification by a safety component. Burton et al. proposed another extension of ISO 26262 in 2012 [4]. It extends the V-model of the standard by an explicit consideration of security in each of the steps.

SAFETY AND SECURITY ANALYSIS

The ISO 26262 standard for safety-relevant systems structures the analysis process in automotive design. The standard provides best practices and marks the state of the art for designing safety-critical systems. It covers all functional safety aspects of the development process and acts as a guide for the entire lifecycle. A central concept is the *Hazard Analysis and Risk Assessment (HARA)*, which results in the assignment of *Automotive Safety Integrity Levels (ASILs)*. The different ASILs guide the selection of appropriate requirements and development techniques in order to reduce the overall risk to an acceptable level. Several specific analysis techniques are established under the organizational umbrella of ISO26262.

One of the most commonly used approaches for qualitative or quantitative dependability investigation is *Fault Tree Analysis (FTA)* [12]. It allows a deductive analysis in failure space, starting from the unintended failure event down to the basic fault events representing the root causes of the problem.

A fault tree only contains the basic events that are significant for the investigated top-failure event. For this reason, the FTA must be combined with a prior identification of interesting failure events. Typically, qualitative approaches in failure space, such as the *Preliminary Hazard Analysis (PHA)* or the *Failure Mode and Effect Analysis (FMEA)* [19], are used in this phase. From the available approaches, FMEA gained the widest adoption due to its small demand for initial system information or probability information.

The commonly accepted security analysis methods are used to verify that the security properties of a product meet all requirements. The methods can be applied in different stages of the development process. At the beginning, results of the design phase can be analyzed to derive security properties from the system's basic structure. In later stages, technical

details of algorithms or an implementation of security mechanisms can be analyzed. *Qualitative methods* seek to systematically discover and enumerate security threats while *quantitative methods* derive numeric or probability measurements for security properties.

One example are attack trees [27], which are logical models that analyze the logical connections of individual attacks. Individual attacks (basic events) are represented as the leaves of the tree. Logical connections with AND and OR gates determine how security events propagate through the tree towards the security failure event, which represents a violation of the analyzed security property.

The qualitative analysis of attack trees allow to determine a minimal cut-set, i.e. a minimal set of attacks that leads to a violation of a security property. If probability information of the basic events is available, the probability of a security violation can be calculated.

Attack trees have been extended to attack-defense trees [20] for including mitigation and defense strategies. A different approach for modeling the interrelations of individual attacks is an attack graph [28, 17]. Starting from a network layout, attack possibilities are modeled in a finite-state machine. Analyses can then derive attack vectors that violate security properties.

Another example for a qualitative security approach is Microsoft's STRIDE [29], a structured approach for discovery and enumeration of threats present in a software system. In a first step, the system under examination is described in a data flow model, which represents entities in the system and data flows between them. For each element of this model, applicable threats are denoted following the STRIDE acronym: (S)poofing, (T)ampering, (R)epudiation, (I)nformation disclosure, (D)enial of service, and (E)levation of privilege. The result of this enumeration is a list of threats that can be amended by mitigation strategies and relevance.

An approach intended for later phases in the development is *penetration testing* [2], which analyzes the security of a product by attempts to circumvent its security mechanisms. The result of penetration testing is a list of discovered vulnerabilities. If penetration testing is carried out repeatedly with tool support, improvement metrics can be used to track progress in the system protection. However, since penetration testing is applied late in the development life cycle, the related repair activities are known to be costly. Various tools such as Nessus [16], NMap, or Metasploit support automated software penetration testing. A similar method is *fuzzing* [32]. In a black-box testing approach, systems are exposed to randomly generated input that intentionally deviates from intended use in content and frequency.

Information flow models describe how information is processed in computer programs. Security requirements for information confidentiality and integrity can be specified formally in such models [8], to describe permitted and forbidden information